AUTO-ISAC
AUTOMOTIVE CYBERSECURITY BEST PRACTICES

SECURITY DEVELOPMENT LIFECYCLE

Best Practice Guide
Version 1.3

Traffic Light Protocol: WHITE.

This information may be shared in public forums.

AUTO-ISAC
Automotive Information Sharing and Analysis Center

Version History

This is a living document, which will be periodically updated under the direction of the Auto-ISAC Best Practices Working Group. We will track any updates in the table below.

Version Notes:

Version	Revision Date	Notes
v1.0	20 March 2019	
v1.1	01 July 2019	Performed periodic continuity and consistency refresh across all Best Practice documents
v1.2	19 August 2019	Changed from TLP **Amber** to TLP **Green** for release to industry stakeholders via request on Auto-ISAC website
v1.3	02 February 2020	Changed from TLP **Green** to TLP **White** for release to the public via request on Auto-ISAC website

Contents

This Guide does not prescribe or require specific technical or organizational practices.
These are voluntary and aspirational practices, which may evolve over time.
Please see Section 1.2 for more information.

1.0 Introduction

1.1 Best Practices Overview

This Best Practice Guide is one in a series of seven Guides intended to provide the automotive industry with guidance on the following Key Cybersecurity Functions defined in the Automotive Cybersecurity Best Practices Executive Summary:

1. Incident Response
2. Collaboration and Engagement with Appropriate Third Parties
3. Governance
4. Risk Assessment and Management
5. Awareness and Training
6. Threat Detection, Monitoring and Analysis
7. **Security Development Lifecycle**

These guides offer greater detail to complement the high-level Executive Summary. This Guide aligns with the "Security Development Lifecycle" function and can be used by companies, as appropriate for their unique systems, processes, and risks.

1.2 Purpose

The purpose of this Guide is to assist the automotive industry stakeholders with designing cybersecurity into vehicle components, subsystems, and systems through the use of a security development lifecycle (SDL). This Guide describes key considerations for successful SDL execution.

This Guide provides guidance without being prescriptive or restrictive. These best practices are:

- **Not Required**. Companies have autonomy and can decide which of these practices to select and can adopt these practices based on their respective risk landscapes and organizational structures.
- **Aspirational**. These practices are forward-looking and voluntarily implemented over time, as appropriate.
- **Living**. Auto-ISAC plans to periodically update this Guide to adapt to the evolving automotive cybersecurity landscape.

1.3 Scope

The scope of this guide includes the vehicle ecosystem, from vehicle hardware and software to the connected services that interact with the vehicle. This ecosystem is shown in Figure 1 and depicts the entities involved and the connections among them. Note that, although vehicle manufacturers, suppliers, and commercial fleet operators are not responsible for designing each extended element (e.g. third-party services) of this ecosystem, vehicle designs generally can account for the insecurities introduced by interactions with these extended elements.

1 *This Guide does not prescribe or require specific technical or organizational practices. These are voluntary and aspirational practices, which may evolve over time. Please see Section 1.2 for more information.*

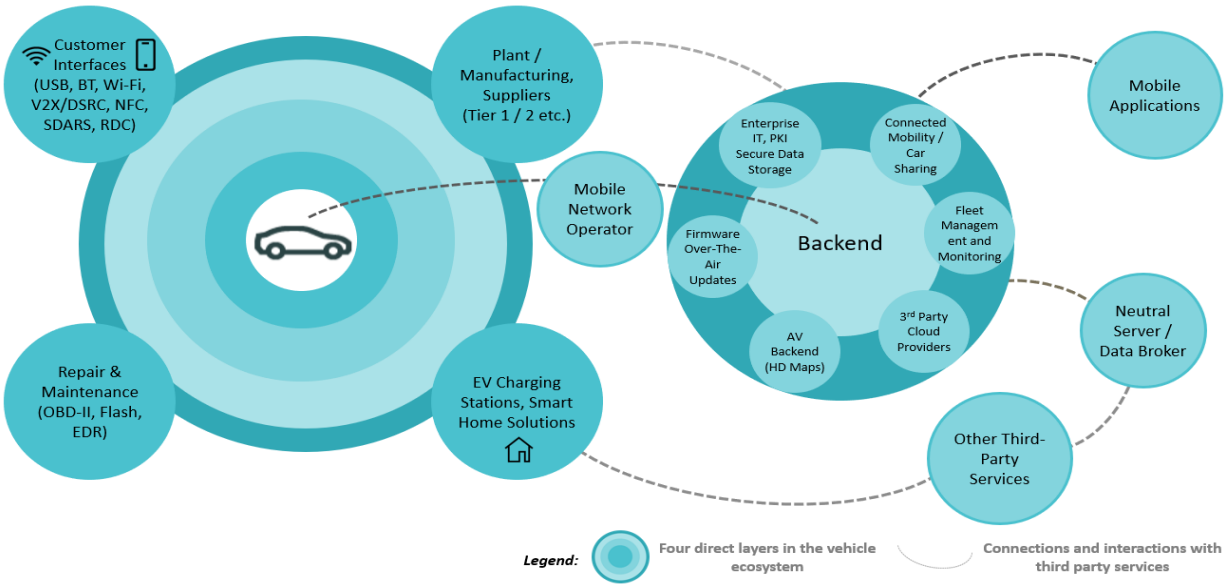

FIGURE 1 – OVERVIEW OF THE AUTOMOTIVE CYBERSECURITY ECOSYSTEM

The best practices in this document provide guidance for the design and development phases of the overall vehicle lifecycle and may be applicable or have implications in later Post-Production phases as shown in Figure 2. Security decisions that are implemented during these phases may include risk mitigation for other phases of the lifecycle, such as the ability to patch vulnerabilities that emerge during operations or functions that enable incident response. However, this Guide does not extend to designing security processes for the full vehicle lifecycle, such as secure manufacturing. Although the details and nuances of the vehicle lifecycle vary across Auto-ISAC members, the best practices described in this guide apply broadly across the industry.

FIGURE 2 – VEHICLE LIFECYCLE PHASES

1.4 AUDIENCE

This Guide was written for use by light-duty and heavy-duty vehicle OEMs, light-duty and heavy-duty vehicle suppliers, and commercial vehicle companies (e.g. fleets, carriers). It may also provide insights for other stakeholders across the connected vehicle ecosystem.

Within these organizations, the primary audience includes cybersecurity managers and senior executives.

This Guide does not prescribe or require specific technical or organizational practices.
These are voluntary and aspirational practices, which may evolve over time.
Please see Section 1.2 for more information.

1.5 Authority and Guide Development

The Auto-ISAC Best Practices Working Group (BPWG) oversaw work on this Guide, with support from Booz Allen Hamilton vehicle cybersecurity Subject Matter Experts (SMEs) who facilitated the Guide's development. The Working Group is comprised of over 150 representatives from Auto-ISAC Member organizations.

The Working Group also coordinated with several external stakeholders while developing this Guide, including NHTSA, ISA/SAE and DHS.

1.6 Governance and Maintenance

The Auto-ISAC Best Practices Standing Committee is responsible for the maintenance of the Guide, which will undergo periodic refreshes to incorporate, as appropriate, lessons learned, new policies, updated or new engineering standards, and the like.

This Guide will be rolled out in phases and marked with the appropriate Traffic Light Protocol (TLP) classification:

- **First 3 months after publication:** **TLP Amber** - available exclusively to Auto-ISAC Members
- **3 to 9 months after publication:** **TLP Green** - released by request to industry stakeholders
- **9 months after publication:** **TLP White** - released to the public via the Auto-ISAC website (www.automotiveisac.com), subject to Board of Directors confirmation

This Guide was developed while ISO and SAE were still in the process of jointly developing the *ISO/SAE 21434 Road Vehicles – Cybersecurity Engineering* Standard. After *ISO/SAE 21434* is published, the Standing Committee plans to review and update this Guide as appropriate.

2.0 Security Development Lifecycle Overview

2.1 Security Development Lifecycle Considerations

Principles of the automotive Security Development Lifecycle (SDL) help ensure that appropriate cybersecurity protections are identified in the early stages of design (e.g. during vehicle electrical architecture planning), when implementation costs are lower and there is time to consider design interactions that might affect cybersecurity.

The SDL applies to all OEMs and suppliers that design and develop vehicles or vehicle components, whether hardware or software. This is true across all styles of development, including both traditional waterfall ("V-model") development cycles and development cycles based on iterative Agile methodologies, in addition to programs use elements of each in hybrid models.

Some of the key considerations that drive security protection decisions in the Security Development Lifecycle are listed in Table 1.

This Guide does not prescribe or require specific technical or organizational practices.
These are voluntary and aspirational practices, which may evolve over time.
Please see Section 1.2 for more information.

TABLE 1. KEY SECURITY CONSIDERATIONS IN THE AUTOMOTIVE SDL

Safety-Critical Systems	Driver occupant and road user safety takes precedence in the system design. For safety-critical systems, reliable operations are required during extreme events (e.g. accident, evasive maneuver, extreme weather).
Shared Components	Frequently, there is a high level of reuse (same/similar ECUs in various makes and models); hence, vulnerabilities in vehicles in the field (older model year) may also affect new vehicle models that are still in development.
Long Lifecycle	On average it takes an OEM approximately four years to develop a new product. A vehicle may then be in production for several years.
Long Consumer Usage Lifecycle	Even after a vehicle is no longer in production, OEMs may support vehicle security for some extended period due to continued driver user. Average life expectancy for a passenger vehicle is 8 or more years, and for commercial vehicles, it can be much longer. Maintenance of security during this long period can be challenging.
Highly Complex Systems	Today's vehicles have many Electronic Control Units (ECUs) each with a combination of hardware, software and firmware, and these can be connected via multiple internal vehicle networks (e.g. CAN, Flexray, Ethernet) and external interfaces (e.g. Wi-Fi, Bluetooth, Near-Field Communications (NFC)). The amount of software (~100 million lines of code) on modern vehicles is on the rise and modularization of vehicles with multiple operating systems only adds to the complexity. There are strict real-time requirements (guaranteed/deterministic/real-time approach), and strict availability requirements (fast boot times, low latency, etc.) for automotive systems.
Highly Constrained Operational Parameters	The environment often has limited microprocessor computing power, data storage, and network bandwidth, making the addition of post-production countermeasures challenging. Additionally, vehicles need to be able to be repaired and calibrated at certified dealerships, OEM service centers and 3rd party repair shops alike. Vehicle owners may not perform regular maintenance and software/firmware updates. Furthermore, vehicle ownership changes often and is difficult to track. The population of vehicle owners is both diverse and not defined or constrained by the OEMs.
Complex Supply Chain	There are multiple tiers of suppliers, from around the globe, involved in product development and multiple vehicle ECUs are developed concurrently by Tier 1 suppliers and OEMs.

2.2 GUIDE STRUCTURE – THE AUTOMOTIVE SDL

The structure of this guide is based on leading cross-industry security development lifecycle processes. The automotive SDL described in Figure 3 is adapted to guide software and hardware development according to the organizational needs of OEMs and suppliers, as well as project or program needs. This includes key interactions that take place through OEM-supplier collaboration around the design process. For example, the automotive SDL considers that:

- Training occurs continuously to support the simultaneous development of multiple components (and often connected services), for a specific vehicle model and for a specific model year.
- Cybersecurity requirements for a vehicle electric architecture are defined by OEM well before specific, component-based cybersecurity requirements are developed. However, Tier 1 and Tier 2 companies may do initial development on systems and components prior to a contract with an OEM. After contract signing, the supplier's work is also related to, but decoupled in time from, OEM design and integration activities.
- Lessons learned are fed back into future electrical architecture requirements and component requirements.

This Guide does not prescribe or require specific technical or organizational practices.
These are voluntary and aspirational practices, which may evolve over time.
Please see Section 1.2 for more information.

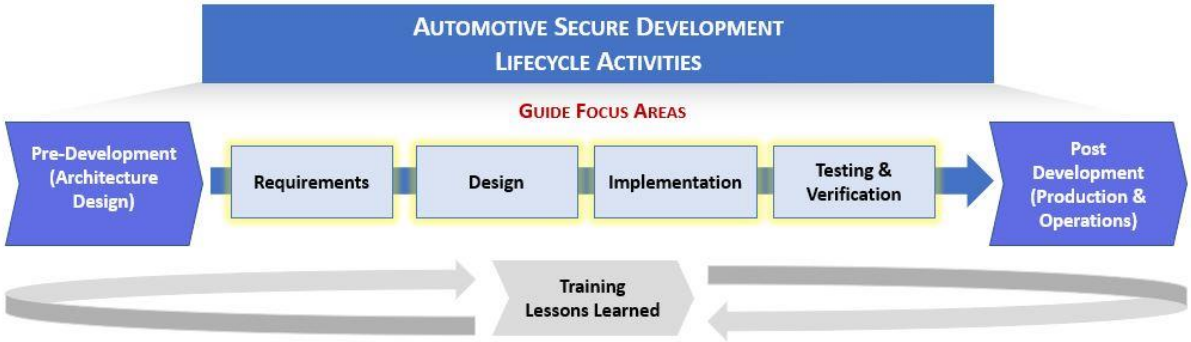

FIGURE 3 – Notional Automotive SDL

Within this guide, each of the focus areas of the automotive SDL are broken down into four subsections: stakeholders, inputs, processes, and outputs. The outputs from one phase feed into the next phase as inputs, creating a holistic approach that can help organizations at different levels of cybersecurity capability apply a structured approach to build or refine SDL processes. This Guide describes key considerations for companies in the automotive industry looking to adopt security development lifecycle best practices.

The References and Resources listed in Appendix B were used in the development of these Best Practices and can be used as additional sources of information. To outline these organizational best practices this guide describes:

- **Stakeholders** that will be engaged during each phase
- **Inputs** for each phase of the automotive SDL
- **Processes** that organizations complete to help ensure security
- **Outputs** from each process that can be used to support subsequent phases

2.3 REFERENCES TO OTHER AUTO-ISAC BEST PRACTICE GUIDES

The automotive SDL brings together and relies on the Best Practices described in all six previous Guides. Throughout this Guide, references are provided to other relevant Best Practice Guides and key information is contextualized within the principles of the automotive SDL. A summary of how each Guide affects the automotive SDL follows:

- **Governance** – Helps to ensure that product cybersecurity is appropriately staffed and integrated into the product development process, including the recognition and enforcement of product cybersecurity practices, timelines, and authority.
- **Risk Assessment and Management** – Underpins all automotive SDL activities by focusing on assets that may benefit from additional security focus. It also informs development of future architecture and decisions on any deviations from the baseline.
- **Collaboration and Engagement with Appropriate Third Parties** – Provides guidance on how to enhance security through collaborating with third parties, as appropriate throughout the development process.

This Guide does not prescribe or require specific technical or organizational practices.
These are voluntary and aspirational practices, which may evolve over time.
Please see Section 1.2 for more information.

- **Incident Response** – Supports response to incidents post-production and helps internalize lessons learned into cybersecurity requirements. The results of IR also feed into design, implementation and testing phases of the secure development lifecycle, and vice versa (e.g. respond to vulnerabilities discovered during development to in-the-field vehicles, as appropriate).
- **Threat Detection, Monitoring and Analysis** – Serves as a source of threat awareness and discovery with the same benefits as the Incident Response.
- **Awareness and Training** – Reduces the introduction of vulnerabilities by developing stakeholder security skills. Facilitates compliance with cybersecurity requirements by improving the ability to answer supplier questions, respond to and analyze vulnerabilities, and create future cybersecurity requirements.

3.0 Best Practices for Pre-Development Considerations

The first activity in the automotive SDL is to identify any pre-development considerations. These considerations may include existing system architectures that constrain future design decisions. Additionally, organizations may want to define the types of cyber risks that are acceptable and unacceptable for the final product. General guidance on risk appetite can be found in the Auto-ISAC *Risk Assessment and Management* Best Practice Guide. Lessons learned during previous design cycles may also inform requirement considerations for the development of new products.

4.0 Best Practices for Design and Development

4.1 Requirements Design

The assignment of appropriate cybersecurity requirements to components, sub-systems, and systems is fundamental to the successful implementation of the automotive SDL. This section discusses an approach to deriving cybersecurity requirements from cybersecurity design principles. The intent during this phase is to develop a comprehensive superset of all required cybersecurity specifications that can be tailored to a component based upon its features (we generically refer to this process as 'tailoring').

Identifying automotive cybersecurity goals is the first step to deriving cybersecurity requirements. For example, a goal could be to offer products that do not allow for unauthenticated electronical manipulation of vehicle control systems, or to reduce vehicle theft numbers to zero. These seemingly simple goals drive efforts to identify control systems, associated attack surfaces, access control mechanisms, and so on. Combined with parallel efforts to support additional goals, a superset of requirements is developed that can be tailored to specific vehicle electrical architectures and their connected components. For example, refer to the security objectives described in the EVITA project for the architecture design process[1].

[1] https://www.evita-project.org/Publications/AEHR10.pdf

Cybersecurity design principles also shape cybersecurity requirements, providing overarching best practices shaped by the collective input of many cybersecurity experts. The process for defining requirements can be described as shown in Figure 4.

FIGURE 4 – OVERVIEW OF REQUIREMENTS PROCESS

4.1.1 Stakeholders

The primary stakeholders involved in this phase include OEMs and Tier 1 and 2 suppliers. Usually, Tier 2 suppliers have minimal interaction with OEMs. However, when a Tier 1 supplier sources a Tier 2 supplier's services or products, the considerations in the section "OEM Role" may apply to that Tier 1 supplier. This approach supports supply chain security efforts. The respective roles of the stakeholders and nature of their interactions are further described below.

OEM Role

OEMs play a major role during this phase by creating the cybersecurity requirements and use cases and by tailoring cybersecurity requirements to the product being developed. OEMs select Tier 1 suppliers through a sourcing process and hold meetings with the chosen suppliers, as appropriate, to review cybersecurity requirements, timing, and key deliverables.

Supplier Role

Suppliers support OEMs in clarifying requirements, developing lower-level requirements, and enhancing use cases. During this phase suppliers may begin to make decisions on software and hardware for a component. Suppliers can surface concerns as needed with the OEM to confirm understanding of operating environments and help to ensure appropriate mitigating controls exist.

If a supplier identifies a gap in security or privacy requirements from an OEM, they can notify the OEM of these additional requirements, resulting in a change to requirement specifications or transfer of risk from the supplier to the OEM. A supplier should not withhold information about gaps in the OEM security requirement specification.

4.1.2 Inputs

Cybersecurity principles are used as references when developing cybersecurity requirements. These principles are based upon anticipated cybersecurity threats, attack surfaces, and architectural features (e.g. network interfaces, bandwidth, topology, latency, cryptographic acceleration).

Secure Design Principles

Secure design principles are a set of common patterns which can help practitioners recognize common security pitfalls and avoid the introduction of vulnerabilities. A cybersecurity best practice is to apply relevant cybersecurity principles within the automotive SDL.

This Guide does not prescribe or require specific technical or organizational practices.
These are voluntary and aspirational practices, which may evolve over time.
Please see Section 1.2 for more information.

Many publications have discussed secure design principles, e.g. the "Framework for Improving Critical Infrastructure Cybersecurity" released by NIST in 2018[2]. These principles can be applied by the automotive industry during all phases of the automotive SDL, including activities within phases, such as: software architecture/development, hardware architecture/development, system integration, test, verification and validation, risk management, and vulnerability management. The principles influence early and later decisions during testing to guide remediation and improve understanding of identified vulnerabilities resulting from non-adherence to one or more principles.

Some of these principles are listed below, with automotive cybersecurity examples.

- **Defense in Depth** – Layers multiple different controls, so that if one control fails, security remains intact, like the defenses of a castle (e.g. a gateway limiting access to and among ECUs, ECU hardening in case the gateway is compromised, or signed code to prevent unauthorized ECU reprogramming in case ECU hardening is compromised).
- **Secure by Default** – Usually used in the context of configurations which come from the factory in a secure state but could be modified by the user (e.g. a vehicle's Wi-Fi access point defaults to a secure configuration from the factory but may be modified by the customer). The system may also warn the user if a change in configuration reduces security.
- **Open Design (Avoiding Security by Obscurity)** – Security is not based upon secrecy of implementation. Instead, use techniques like verification of signed code that are inherently secure even if their use is widely known.
- **Fail Securely** – When a component fails it defaults to a secure configuration (e.g. if a gateway stops working, then it defaults to denying all traffic rather than allowing all traffic).
- **Economy of Mechanism** – Designs are kept simple to shrink attack surface (e.g. a component implements only required protocols and unused functionality is removed).
- **Separation of Duties (Functions)** – This involves separating functional duties (e.g. Wi-Fi is not included on a steering module).
- **Separation of Privileges** – A single privilege is broken into multiple pieces (e.g. workflow with multiple approvers is followed to provision a certain vehicle component).
- **Principle of Least Privilege** – Limiting privileges to the least necessary to perform a required operation (e.g. internet facing entities have no CAN bus privileges, services that do not require root access do not run under the root account).
- **Zero Trust Model** – Since most components can be individually compromised, trust is limited (e.g. validate inputs).
- **Continuous Update** – A secure system deployed today may not remain secure forever. Note that other principals like defense in depth, when applied well, may mitigate or reduce the impact of components found to have vulnerabilities later.
- **End-to-End Security** – This relates to safeguarding information in an information system from point of origin to point of destination[3].

A security architecture based on these security principles may provide layered cybersecurity defenses, with the goal of making it not worth an adversary's time or resources to attempt

[2] https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf
[3] https://csrc.nist.gov/glossary/term/end_to_end-security

vulnerability exploitation and attack development. OEMs and suppliers may use a risk-based decision making when balancing security against stability, determinism, availability, future analysis/testability, and vehicle performance. One way that these may be balanced is by classifying risk profiles with a score, which can then be mapped against appropriate countermeasures and defense in depth techniques.

Results of Programmatic Risk Assessment

Organizations may use risk analysis to inform their security goals, which inform security requirements, which then inform the security architecture of the design and eventually a secure design and development process. Risk analysis and requirements definition are related activities that are usually done in parallel and iteratively.

OEMs aim to clearly communicate their defense requirements to their suppliers. These discussions with suppliers may lead to clearer cybersecurity requirements for selected assets. OEMs may also choose to adapt their cybersecurity requirements to the asset that is being protected. An example of a methodology for doing this is to group technical components (e.g. infotainment system, blind spot monitoring) into high/medium/low risk categories, as defined in the Auto-ISACs *Risk Assessment and Management* Best Practice Guide. Then those risk categories can be used to inform the amount of security rigor that needs to be applied to the associated component. However, independent of this high/medium/low risk grouping methodology, are basic cybersecurity principles that all ECUs meet.

Industry Standards

With advancements in technology, the criteria for compliance among automotive engineers and designers is likely to evolve. It is important for companies to review changes periodically and adapt requirements appropriately for compliance. There are also industry standards such as *SAE J1962*, which defines the diagnostic connection, and *ISO 26262*, which provides a system of steps for managing functional safety and regulating product development and *ISO/SAE 21434*. These and other applicable standards[4] such as *J3101* and *J3061* can inform product requirements.

Other external factors can also influence requirements, such as infrastructure challenges with corresponding cyber challenges in support of certain safety critical functions.

4.1.3 Processes

Threat Modeling

Threat modeling is a structured representation of all the information that affects the security of an application to identify, communicate, and understand threats and mitigations within the context of protecting that application. More details on this topic can be found by referencing *OWASP and Microsoft STRIDE*. As such, threat modeling is one of the most direct and valuable approaches to securing products using the automotive SDL. It directly leverages a project's designs and guides the design decisions in a security-centric way. Once the various threats to all the elements of the designs are known, decisions about those threats can be made early in the project to inform the security requirements and security designs. Threat modeling is an iterative process that has

[4] https://www.asam.net/index.php?eID=dumpFile&t=f&f=2062&token=36695ba6f676ea928763525a33dcbf34e2217eee

relevance in nearly every stage of a project. The requirements phase could include multiple rounds of threat modeling.

It is important to understand and manage risk during each phase, so companies will benefit from a strong understanding of potential threats and vulnerabilities. Different techniques can be employed to understand the risk and threat landscape and incorporate security into the design process. Although systems can have significant differences, threat modeling is a generic strategy which can be used to identify potential cyber risks against any kind of system.

Threat modeling fits into the larger categories of threat analysis and risk assessment. Threat modeling helps identify threats which are then analyzed. A threat could be as simple as someone pretending to be someone else, or as advanced as a multiple-link attack chain pivoting methodically through several layers of the target system.

Threat modeling may be used in every phase as necessary to manage and mitigate risk. For more information on risk analysis and mitigation approaches refer to the Auto-ISAC *Risk Assessment and Management* Best Practice Guide.

4.1.4 Outputs

Security Requirements Documentation and Traceability

Identified requirements must be clearly documented. Requirement documentation may be a hierarchy of documents at general/concept, feature, function, interface, protocol levels. Such documents typically contain all the requirements for a given product or feature (e.g. use an explicit library or TLS version, or implement certain OS/library patch levels) and can be an all-encompassing reference for those involved in designing the product. The documents typically are limited to requirements only and do not provide specific solutions for security requirements.

4.2 DESIGN

Secure design benefits from well-defined requirements. However, even with the best requirements, software and hardware design can be challenging and benefits from clear goals. Some best practices to consider using in the design phase include:

- **Ensuring all requirements** are clear and testable.
- **Having a good understanding** of threats and risks to a system. This will help the design team to focus on including secure solutions.
- **Utilizing a system architecture** that can help mitigate the identified threats and risks.
- **Embracing cybersecurity principles** and using design techniques in line with these principles.

Knowing the full set of features that products must meet is critical at this stage to ensure security has covered the full scope of the product. This includes functions relative to donor systems, update methods and planned diagnostics.

4.2.1 Stakeholders

The primary stakeholders involved in this phase include OEMs and Tier 1 and 2 suppliers. Usually, Tier 2 suppliers have minimal interaction with OEMs. However, when a Tier 1 supplier sources a Tier 2 supplier's services or products, the considerations in the section "OEM Role" may apply to

that Tier 1 supplier. This approach supports supply chain security efforts. The respective roles of the stakeholders and nature of their interactions are further described below.

OEM Role

OEMs own the overall system design and conduct paper-based design reviews with stakeholders to ensure designs meet their requirement specifications. OEMs also perform risk assessments.

Supplier Role

Suppliers are accountable for their designs and for providing evidence to demonstrate that their designs meet OEM requirements (functional and security). In addition, suppliers are accountable for the requirements for solutions developed by sub-suppliers and can make appropriate design decisions to mitigate risks in their own products. Furthermore, they are accountable for their designs and for providing evidence to demonstrate that their design meets the customer's requirements (functional and security).

4.2.2 Inputs

Security Requirements

The input is a comprehensive set of security requirements from the requirements phase as discussed in section 4.1.4. The design team can make sure that the security requirements are testable and organized in the manner that is also traceable. These form the basis for design along with appropriate cybersecurity principles.

4.2.3 Processes

Requirements Decomposition

Organizations can break down requirements into smaller manageable and testable parts during the design process. This usually happens if some part of the security requirements will be used in another context. It also helps to break down requirements if additional requirements are expected to be derived from them. This step can provide a layered approach to the automotive SDL and better readability and accountability.

Designing Security Controls

Security control designs can draw upon lessons learned and other intelligence from sources such as incident response and vulnerability management processes, industry best practice guides, and historical issues found within the automotive industry or similar industries (e.g. medical, aviation, IoT). A few security control measures that will help organizations in the design phase include defining cybersecurity policies for design, access controls, key management, etc. Organizations should consider appropriate controls that can be used during design phase to strengthen cybersecurity of the system.

Security Design Reviews

Security design reviews help ensure that a system’s design elements reflect the security requirements, applicable security standards, and secure design principles. An overall security review of system design can also help identify other vulnerabilities outside of the system (e.g. randomness of seeds not strong enough for challenge-response authentication between to ECUs). An approved security design review can be used as a gate for containing a phase.

Design reviews typically serve as the exit gate for the design phase of the process, and security design reviews would be precursor and input to the design review. Issues identified during the

security design review are evaluated for risk treatments (see the Auto-ISAC *Risk Assessment and Management* Best Practice Guide). Organizations may also consider lessons learned as appropriate. During OEM security design reviews, OEMs can confirm supplier assumptions about the context of their components within the security architecture and understand controls to achieve defense in depth. A security review could also be conducted after requirements analysis.

What needs to be reviewed during a security design review is typically agreed upon in advance, and can include, but is not limited to, the following:

- Design documentation detailing all interfaces and security-related functions
- Test plan documentation
- Requirements traceability matrix
- Documentation of any identified risks
- High-level milestone schedule for subsequent OEM/supplier interaction (e.g. security reviews, integration events, or interim test events).

4.2.4 Outputs

Test Plans

During the design phase, a high-level test plan can be constructed to identify the best means of verifying the security requirements (e.g. design review, manual code review, automated code analysis component/unit testing, bench and vehicle penetration testing). These test plans can focus on detailing the methodology used to perform the test, with special attention to the tools and any unique build components or infrastructure support. Most companies may include the following in their test plans:

- Project scope
- Objectives
- Target market
- Assumptions
- Testing cycle start/end dates
- Major roles and responsibilities/overall resources
- Testing environment
- Deliverables
- Major risks and mitigation strategies
- Defect reporting and mitigation
- Test cases
- Testing schedule and recurring tasks (e.g. review configuration security)
- Major test tooling to be purchased and/or created

Security testing may require specialized personnel and tools that may not exist in a traditional quality control team. It is important that these security tests are made to be verifiable.

The threat modeling that was described earlier could be used to help directly derive the security test plan. A threat model typically describes what needs to be protected, the attacks that may take place, and the conditions under which such attacks are likely. A security test plan thus focuses on testing these identified conditions and attacks. At a minimum, a test plan report

typically includes the following elements. For more details on this refer to existing standards by *NIST, SAE* and *ISO*.

- Identifies the best security verification methods (e.g. design review, manual code review, automated code analysis, component/unit testing, bench and vehicle penetration testing).
- Identifies tools needed to perform the test, including special build components and infrastructure support.
- Includes an evidence sheet with details of software, hardware level, date, pass/fail status, notes on failures or unexpected behavior, person running the test and approver, and others as necessary.
- Supports transparent traceability to the cybersecurity requirements defined during the requirements phase and, in turn, transparent traceability to the goals from the risk analysis.

Requirements Traceability Matrix

At this point of the process, the requirements, design elements, and high-level test cases can all be identified. To ensure that each requirement is implemented, and that there are high level plans to verify that design meets requirements, it is important to begin to build out a traceability matrix that will serve to lay out the bidirectional traceability of requirements, design elements, and test cases. The practice of creating this matrix allows the development and testing teams to identify any gaps in the implementation or verification of requirements early in the development process. As things change and evolve during development, this matrix can be revisited and updated to ensure that no gaps have arisen out of any changes.

4.3 IMPLEMENTATION

A focus on security in the requirements and design phases drives secure vehicle development, integration, and production. While these foundational phases (requirements and design) are central to building an overall secure product, mistakes can occur during the implementation of those security focused requirements and designs. These mistakes may result from the people, processes, or technologies involved in the implementation process.

Using secure implementation best practices can help ensure the effort put into requirements analysis and secure design is not lost during implementation.

4.3.1 Stakeholders

The primary stakeholders involved in this phase include OEMs and Tier 1 and 2 suppliers. Usually, Tier 2 suppliers have minimal interaction with OEMs. However, when a Tier 1 supplier sources a Tier 2 supplier's services or products, the considerations in the section "OEM Role" may apply to that Tier 1 supplier. This approach supports supply chain security efforts. The respective roles of the stakeholders and nature of their interactions are further described below.

OEM Role

Implementation is carried out on multiple levels, with OEMs taking ownership of the overall security requirements and policies for a vehicle and therefore managing and monitoring the implementation process. This means that OEMs may work with multiple Tier 1 and Tier 2 suppliers during implementation to ensure that a completed system adheres to the OEM's security policies and requirements. OEMs provide requirements to suppliers with which a fully integrated vehicle

should comply, and review supplier provided test plans. Then, the final system is typically tested at the fully integrated level to ensure compliance with defined security requirements.

Supplier Role

Implementation for suppliers means working to fulfill the OEM's security requirements by integrating them into their product design and provided test plans. A supplier-developed system should meet the OEM's requirements and expectations, per any contractual agreements at a minimum. Testing the developed systems at the component level helps to ensure compliance with all requirements, which includes providing security test procedures/plans to the OEM before testing and reporting security test results upon completion of testing. Additionally, the system should be compliant with all relevant cybersecurity regulatory requirements, including shipment or export regulations. Taking into consideration the security of the environment in which a product will be packaged or integrated can help ensure that installation and operation occurs as expected.

4.3.2 Inputs

Inputs to the implementation process include the outputs from the design process, such as requirements traceability matrix (see section 4.2.4), as well as leveraging knowledge of other existing implementation processes and taking steps to secure the development environment in which product integration occurs.

Existing Implementation Processes:

It is often valuable to integrate cybersecurity into existing implementation processes within an organization. However, it remains each independent organization's responsibility to determine whether and how cybersecurity is incorporated into these existing processes. In doing so, the goal is to ensure that contradictions do not occur and to ensure all perspectives are given consideration in the design and implementation of an item.

Implementation Reviews:

Implementation reviews (especially for critical systems) are typically conducted over the course of the development cycle, including during testing and verification, and can include both internal and external (i.e., outside of the direct development team) parties to safeguard against any bias. Security can be an explicit focus area during implementation reviews, along with other areas such as structure, complexity, reliability, etc. Implementation review results can be tracked, and the successful completion of code reviews can be a requirement for advancement into integration or release branches. This can be in a separate system or as part of the Software Configuration Management (SCM)[5] process.

4.3.3 Processes

Secure Software Development:

To help ensure that secure software development practices are at the forefront of developers' minds, teams should align on project and organizational requirements.

Coding Standards:

As applicable, development teams can follow common security coding standards such as MISRA and CERT-C. Organizations such as ISO and SAE refer to these. Some examples may include:

[5] https://www.energy.gov/sites/prod/files/cioprod/documents/scmguide.pdf

This Guide does not prescribe or require specific technical or organizational practices.
These are voluntary and aspirational practices, which may evolve over time.
Please see Section 1.2 for more information.

- Compliance with policies that banned functions, libraries, and APIs should not be allowed in code (e.g. non-secure versions of memcpy, strcpy, sprintf)
- Use of some form of static and dynamic code analysis (agreed to by customers and suppliers), performed at defined intervals, with explicit gates in place at milestones in the process (e.g. build, release) and defined for resolution of any issues that are discovered

Secure Configurations

Configurations need to be implemented securely and following principles as secure by default and least privilege. One example to implement secure configurations is to review planned firewall rules for an ECU, apply the strictest firewall rules in the early stage of a project, review the firewall rules on a periodic basis and document necessary inevitable exceptions. In general, the review of configuration security should be a part of the test plan.

*Secure Codi*ng

Secure coding practices are important to developing robust vehicles. While there are many defined standards for secure coding practices (e.g. *MISRA C:2012*, *CERT-C*, *ISO/IEC TS 17961*), organizations may align on and enforce standards based on project and organizational requirements, while taking into consideration certain aspects that are unique to the vehicle environment (e.g. collaboration/integration with components developed by suppliers/OEMs/open source community, real-time requirements of vehicle systems, right-to-repair requirements).

Secure coding practices begin with the OEM defining expectations of what practices suppliers and internal developers need to follow, as well as what evidence is to be provided to ensure those practices are being followed. Since OEMs often do not have direct line-of-sight into, or ownership of the code, considerations can be put in place to help ensure that the OEM, as the system owner, has enough insight or assurance that secure coding practices have been applied. This may be included in the statement of work, and/or in standard terms and conditions. OEMs may have clearly defined means to audit, perform threat detection and response (TDR), or perform spot checks.

In all cases, care can be taken to help ensure that production software does not include unneeded components (e.g. debuggers or debugging symbols), known vulnerabilities, or default certificates/files/credentials. To minimize development impacts, code can be developed with minimum privileges from the beginning and can manage increases to privileges as needed.

Software Traceability

To mitigate risk, organizations can request a detailed bill of materials (BOM) to be included with all developed components for traceability. This detailed BOM can then be mapped to VINs and ECU serial numbers on specific vehicles. For example, a software BOM can include the following:

- Software identifier, version, revision, and date of release
- PKI information (Certificate Life Cycle Management)
- Vendors or Suppliers

A software BOM can be clearly documented and can include all third-party software in use—ideally there would be an automated process to collect this information from either source code or binary code, and then roll it up in a database to get a picture of the overall system. This software BOM can be cross-checked against known vulnerabilities (e.g. from the CVE database),

and patches can be applied for any vulnerabilities that exceed an agreed-upon risk/severity level. Support contracts with third parties should take this requirement into account.

Secure Hardware Implementation

For more details on this topic refer to *SAE J3101 Hardware Protected Security for Ground Vehicles*.

4.3.4 Outputs

Outputs of secure implementation include verifying and testing that integrated components meet cybersecurity requirements at the hardware and software level. At the hardware level, this can be accomplished through confirmation reviews or assessments and penetration testing. At the software level, this can be accomplished through code reviews, automated code analysis, and penetration testing. Both processes are iterative in nature and typically occur multiple times throughout the implementation process at defined checkpoints and at the end of overall implementation.

4.4 Testing and Verification

This section provides an overview on which cybersecurity tests may be performed during the vehicle development lifecycle and which testing methodology may be appropriate under certain circumstances. While testing proves the implemented systems are working properly, verification is the overarching process to consider whether a system was developed according to the requirements and specifications. This includes checking if security design principles (e.g. “Least Privileges”) have been specified and implemented properly for the target system.

4.4.1 Stakeholders

The primary stakeholders involved in this phase include OEMs and Tier 1 and 2 suppliers. Usually, Tier 2 suppliers have minimal interaction with OEMs. However, when a Tier 1 supplier sources a Tier 2 supplier’s services or products, the considerations in the section "OEM Role" may apply to that Tier 1 supplier. This approach supports supply chain security efforts. The respective roles of the stakeholders and nature of their interactions are further described below.

OEM Role

OEMs perform system-level verification, user acceptance testing, security validation testing and implement remediation as required. OEMs are also responsible for conducting a final (production) risk assessment.

Supplier Role

Suppliers perform security verification testing and provide evidence (reports) to demonstrate that the implementation fulfills the security requirements.

Whenever Tier 1 suppliers integrate solutions offered by Tier 2 suppliers into their products, their role shifts from a contractor towards a principal. To secure the security supply chain, Tier 1 suppliers perform validation on parts or software developed by Tier 2 suppliers. In this case the items mentioned above are provisioned by Tier 2 suppliers to the Tier 1.

This concept has the big advantage that security risks are communicated up and down the supply chain, so that no unknown security issues can be introduced in products a Tier 1 is liable for.

Furthermore, the OEM can perform better risk management efforts, even if there is no contract or agreement between Tier 2 and the OEM itself.

Third-party Role

Third-parties may support OEMs or suppliers in conducting independent validation and security verification testing.

4.4.2 Inputs

Inputs to the testing and verification phase can leverage the outputs from the requirements and implementation phases. These include the implemented system and supporting infrastructure, confirmation review or assessment, and security testing results, such as penetration test results, code reviews, and results from automated code analysis, which can be used in conjunction with an assessment of testing methodologies to determine the correct types of testing for a specific product or organization.

The implementation phase delivers a more detailed implementation plan to fine-tune test planning activities. Test cases for security-relevant functions, either provided by developers or self-written, can be used together with existing or interim test results as inputs to make the test planning and activities more thorough.

Cybersecurity-related test planning processes usually do not substantially differ from the existing test planning processes (e.g. customer function validation) of a company.

4.4.3 Processes

Cybersecurity Testing

The actual testing happens in multiple iterations during and after the implementation phase of a product. Cybersecurity testing is an important form of “health check” to demonstrate proper implementation of security requirements and provide input for residual-risk assessments after vulnerabilities have been identified. During the implementation of a vehicle and its ecosystem, cybersecurity testing at planned milestones helps a company to ensure planned safeguards are effective, identify potential vulnerabilities and remediate these before vehicle deployment. The earlier vulnerabilities are identified, the less time and budget is consumed for remediation.

Internal Cybersecurity Sign-off Process

The internal cybersecurity sign-off does not give a guarantee that a product is 100% secure against cyber attacks, but it can give a snapshot that under current conditions a product is robust enough to withstand the attempts of previously assessed attacker profiles.

The internal cybersecurity sign-off process is a way to provide final confirmation to involved stakeholders of a development project that all security-related work has been done completely and diligently. This would provide assurances that cyber-security controls have been properly implemented, common vulnerabilities have been assessed and fixed if needed, and cybersecurity tests have been performed showing that safeguards work as expected. This confirmation can be performed for each component/software development project at planned milestones throughout the product life cycle. The internal cybersecurity sign-off gives the security team a tool to escalate deviations to management and prioritize critical projects. However, security work does not end with the cybersecurity sign-off, because prolonged, public exposure of a system to uncontrolled environments gradually increases the risk of abuse and misuse.

To provide an internal cybersecurity sign-off for OEM senior management (i.e., assurance for the entire vehicle and its ecosystem), a sign-off of every component/software project needs to be collected and aggregated. Relevant information and items within this sign-off process should be the overall test plan, performed functional tests, penetration tests and source code audits, the number of unresolved vulnerabilities and their risk ratings, open tests or test cases, etc. If a component/software project cannot be signed-off due to incomplete work (e.g. missing test results or unfixed vulnerabilities) the overall risk to the vehicle and its ecosystem needs to be assessed.

Ideally, all assessments have been performed before the vehicle is deployed. All sign-off-related documents can be used to track progress of the security team's and developer's work (e.g. in the case when a vulnerability could not be fixed in time and its risk acceptance has been decided on by senior management). Before vehicle deployment, residual risks identified during and after the sign-off process can be treated as explained in the Auto-ISAC *Risk Assessment and Management* Best Practice Guide.

The cybersecurity sign-off process may be tied to an organization's existing development processes like the internal approval processes for the functional safety of a vehicle. This has the benefit that cybersecurity testing will become embedded in the development process, raising overall awareness. It is also a great tool to provide senior management with information if processes are complete and have been exercised with due diligence and due care by all stakeholders.

Based on a final assessment, resulting recommendations for future projects can be communicated to the development project teams, the vehicle architecture team and management to raise the security bar long-term.

4.4.4 Outputs

The most important output of the testing and validation phase is reports of the test results. Based on these reports, further steps can be undertaken. This sign-off document can then be used to show senior management that due care and due diligence has been exercised during a product's development phase. The following process describes how a company may be able to create such a document.

Residual Risk Assessments

The assessment of residual risk can be performed both as part of the Secure Design Lifecycle leading up to the initial start of production, and on a periodic basis during production to account for evolving threats. This applies to known residual risk, as the discovery or publication of new attack methods, or reduction of attack costs due to new or cheaper tools may shift the cost/benefit assessment on risk mitigation over time. The trend for attacks to improve over time can also be considered prior to production, when making the initial decision to accept a residual risk. A company's threat monitoring capabilities as described in the Auto-ISAC *Threat Detection, Monitoring and Analysis* Best Practices Guide may support the identification of new attack trends.

The product's position in the supply chain may enable mitigation of residual risks at a different level. For example, residual risks identified in a component from a Tier 1 or lower Tier supplier may be mitigated at a higher system level. For this reason, it is valuable for suppliers to share their threat models and mitigations with their customers, who can then either accept or develop mitigations for the transferred residual risk. It is helpful that all risk rating methodologies and processes are shared amongst a product's supply chain stakeholders to support the risk rating and treatment work of all involved cyber and risk management teams. Please refer to the Auto-ISAC *Risk Assessment and Management* Best Practice Guide for additional information on how an organization may treat these residual cyber risks.

5.0 Best Practices for Post Development (Production and Operations)

Focusing on security along each phase of the automotive development lifecycle can produce a secure vehicle. However, there are post-development challenges to keep a product secure over its lifespan. Some of these challenges include long lifecycle of vehicles, serviceability, and maintenance. Such challenges can be considered early on and may influence the requirements for, and design of, the vehicle and its components.

Although designing security into a product occurs well before a product rolls off the production line, it is worthwhile to note that maintenance processes offer valuable feedback to the development process. Some of these post-production processes can inform future secure product design include incident response, vulnerability management, and security monitoring. For additional information on each topic, please see the Auto-SAC's *Incident Response Best Practice Guide; Threat Detection, Monitoring, and Analysis Best Practice Guide;* and *Vulnerability Management Best Practice Guide* (tentatively slated for future development). These processes can extend well beyond a single product and have the potential to enhance the overall security posture of an organization. The lessons learned from these challenges can feed into the requirements and design phases of the automotive SDL process to aid in continuous improvements in security.

5.1 Long Vehicle Lifecycles

Automotive systems are in use, on average, for over a decade. This long lifecycle compared to many other consumer internet-connected products can give rise to unique challenges: (1) the security threat landscape has been evolving quickly and may drastically change over the life of a vehicle; (2) vehicles are not under direct control by OEMs; (3) the high amount of product exposure to the public can erode security; and (4) it may be difficult for OEMs and product providers to shut down a system entirely considering safety related issues.

To mitigate these post production challenges associated with the long vehicle lifecycle, automotive companies may engage in several best practices during the automotive SDL process. Both OEM (system designers) and suppliers (product providers) can consider how systems will be maintained if critical elements are found to be vulnerable. For OEMs this could mean building in various forms of agility and defense-in-depth at several layers to adequately respond to changes in the threat landscape over time (e.g. update crypto schemes, enable/disable services) as well as using periodic reviews of carryover designs for new risks arising from application changes. OEMs may also want to employ limited and controlled lifecycle mechanisms, such as

This Guide does not prescribe or require specific technical or organizational practices.
These are voluntary and aspirational practices, which may evolve over time.
Please see Section 1.2 for more information.

key revocation and replacement or lifecycle mode changes (e.g. development, production, and warranty).

Planning for the long-term maintenance of connected vehicle infrastructure or secure decommissioning can prevent bad actors from interacting with older vehicles (e.g. maintaining control of telematics-related domains to prevent domain takeovers). OEMs may also consider ways to help their customers protect their information and remote access both in transfer of ownership and in end-of-life use cases.

For suppliers, this could mean considering the long-term support plans for products as they are built and communicating those support plans to their customers. The types of information that suppliers may communicate to OEMs could include how long the product will be supported for weaknesses or discovered vulnerabilities and how they will ensure maintenance of build environments, development expertise, and resources necessary to provide this type of support.

5.2 SERVICEABILITY AND MAINTENANCE

Contracts govern the duration for which vehicle and component designs are serviceable and maintainable. Offering detailed service procedures for troubleshooting, replacing and updating vehicle components as needed either through service centers or via over the air updates can support these efforts. Updates are changes made to the hardware or software of the appropriate item or system that is deployed in the field (e.g. configuration, software, new or removed capability). Servicing includes the ability to make functional and remedial updates while maintaining revision control, traceability and revocability.

Tools management includes providing secure tool usage for diagnosis and functional testing of the product. Evidence of strategy, calibration level, and list of tools available can be logged and made available where appropriate.

Appendix A: Glossary of Terms

Relevant terms used in this Guide are defined below.

TERM	DEFINITION
Attacker	Individual, group, organization, or government that conducts / has the intent to conduct an attack.
Bill of Materials (BOM)	A list of the raw materials, sub-assemblies, intermediate assemblies, sub-components, and the quantities of each needed to manufacture an end-product.
Confirmation Review/Assessment	This assessment judges whether the available evidence provides enough confidence in the achieved cybersecurity of the item, or of the contribution to the achievement of cybersecurity by the component(s). This is typically performed at regular milestones throughout development, prior to moving to the next stage of development. At a minimum, a cybersecurity assessment can be done to judge whether to proceed to the testing and verification phase.
Impact	Estimate of magnitude of harm to stakeholders originating from a threat and/or attack
Penetration Testing	The practice of testing a system, network or application to find vulnerabilities that a threat actor could exploit.
Risk Profile	An evaluation of a company's risks, including the number of risks, type of risk, and potential effects of risks.
Risk Tolerance	The threshold of risk that an organization or individual is willing to accept without some form of response.
Threat	Any circumstance or event with the potential to adversely impact the vehicle ecosystem through an information system via unauthorized access, destruction, disclosure, or modification of information, and/or denial of service.
Threat Actor	A person or entity posing a threat to the vehicle ecosystem.
Threat Event	An event or circumstance, perpetrated by a threat actor, that has the potential to cause a negative impact to the vehicle ecosystem.
Vehicle Cybersecurity	The activities, processes, and capabilities that protect, detect and respond to cyber occurrences (e.g. remote control, unauthorized access, disruption, manipulation) that actually or potentially result in adverse consequences to a vehicle, connected infrastructure, or information that the vehicle processes, stores, or transmits.
Vehicle Cybersecurity Risk	The likelihood of and potential impact from the exploitation of a vehicle ecosystem cybersecurity vulnerability in a threat event.
Vehicle Ecosystem	The components and infrastructure on or connected to the vehicle (e.g. hardware and software, intellectual property, mobile applications, customer data, vehicle data, supplier/manufacturing networks, applications, processes and organizations that directly or indirectly touch the vehicle and may play a role in vehicle cybersecurity).
Vulnerability	Weakness of an asset or control that can be exploited by one or more threats.

This Guide does not prescribe or require specific technical or organizational practices.
These are voluntary and aspirational practices, which may evolve over time.
Please see Section 1.2 for more information.

Appendix B: Additional References and Resources

The following References and Resources provide additional content and expertise for companies to consider in conjunction with the Best Practices discussed in this Guide.

REFERENCES – DOCUMENTS THAT MAY OFFER ADDITIONAL IMPLEMENTATION GUIDANCE
Threat Modeling: Designing for Security by Adam Shostack
ISO/SAE 21434 - Road Vehicle Cybersecurity Engineering Standard (under development) <link>
NIST SP 800-30 - Guide for Conducting Risk Assessments <link>
SAE J3061 - Cybersecurity Guidebook for Cyber-Physical Vehicle Systems <link>
NIST Cybersecurity Framework <link>
ISO/IEC 15408 – Information Technology - Security Techniques <link>
ISO/IEC 17799 – Code of Practice for Information Security Management <link>
ISO/IEC 27001 – Information Security Management Systems - Requirements <link>
ETSI Cyber Security Technical Committee (TC CYBER) ETSI TR 103 456 – Implementation of the Network and Information Security (NIS) Directive <link>
ISO 31000:2009 – Principles and Guidelines on Implementation <link>
DOT HS 812 073 – NIST Cybersecurity Risk Framework Applied to Modern Vehicles <link>
Capability Maturity Model Integration (CMMI) <link>
SAE International Aerospace Standards - ARP6328, AS6081 and ARP6178 <link>
Supply Chain Response EIA STD 4899B and EIA 933B <link>

RESOURCES – ORGANIZATIONS THAT MAY OFFER ADDITIONAL INSIGHTS
International Organization for Standardization (ISO) <link>
National Institute of Standards and Technology (NIST) <link>
National Highway Traffic Safety Administration (NHTSA) <link>
PMI PMBOK Guide <link>
SAE International <link>
Institute of Risk Management (IRM) <link>
ISA/IEC 62443 Cybersecurity Certificate Programs <link>
Capability Maturity Model Integration <link>

Appendix C: Acronyms

API	Application Programming Interface
Auto-ISAC	Automotive Information Sharing and Analysis Center
BOM	Bill of Materials
BPWG	Best Practices Working Group
BT	Bluetooth
CAN	Controller Area Network
CERT	Computer Emergency Response Team
CVE	Common Vulnerabilities and Exposures
DSRC	Dedicated Short-range Communications
ECU	Engine Control Unit
EDR	Endpoint Detection and Response
EV	Electric Vehicle
EVITA	E-Safety Vehicle Intrusion Protected Applications
IRM	Institute of Risk Management
ISO	International Organization for Standardization
IOT	Internet of Things
IT	Information Technology
MISRA	Motor Industry Software Reliability Association
NFC	Near Field Communication
NHTSA	National Highway Traffic Safety Administration
NIST	National Institute of Standards and Technology
OBD	On-board Diagnostics
OEM	Original Equipment Manufacturer
OS	Operating System
OWASP	Open Web Application Security Project
PKI	Public Key Infrastructure

PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
RDC	Remote Desktop Connection
SAE	Society of Automotive Engineers
SDARS	Satellite Digital Audio Radio Service
SDL	Security Development Lifecycle
SME	Subject Matter Expert
TDR	Threat Detection and Response
TLP	Traffic Light Protocol
USB	Universal Serial Bus
V2X	Vehicle to Everything
VIN	Vehicle Identification Number