



SURFACE VEHICLE RECOMMENDED PRACTICE

J3061™

JAN2016

Issued

2016-01

Cybersecurity Guidebook for Cyber-Physical Vehicle Systems

RATIONALE

To provide a cybersecurity process framework and guidance to help organizations identify and assess cybersecurity threats and design cybersecurity into cyber-physical vehicle systems throughout the entire development lifecycle process.

- Defines a complete lifecycle process framework that can be tailored and utilized within each organization's development processes to incorporate cybersecurity into cyber-physical vehicle systems from concept phase through production, operation, service, and decommissioning.
- Provides high-level guiding principles.
- Provides information on existing tools and methods.
- Provides the foundation for further standards development.

TABLE OF CONTENTS

1.	SCOPE	5
1.1	Purpose	6
1.2	When to Apply a Cybersecurity Process.....	6
2.	REFERENCES.....	6
3.	DEFINITIONS AND ACRONYMS	8
4.	RELATIONSHIP BETWEEN SYSTEM SAFETY AND SYSTEM CYBERSECURITY	17
4.1	Analogies between System Safety and System Cybersecurity Engineering.....	18
4.2	Unique Aspects of System Safety and System Cybersecurity	18
5.	GUIDING PRINCIPLES ON CYBERSECURITY FOR CYBER-PHYSICAL VEHICLE SYSTEMS (CPS).....	20
5.1	Know Your System's Cybersecurity Potential.....	20
5.2	Understand Key Cybersecurity Principles.....	20
5.3	Consider the Vehicle Owners' Use of the System	21
5.4	Implement Cybersecurity in Concept and Design Phases.....	21
5.5	Implement Cybersecurity in Development & Validation	21
5.6	Implement Cybersecurity in Incident Response	22
5.7	Cybersecurity Considerations When the Vehicle Owner Changes	22

SAE Technical Standards Board Rules provide that: "This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user."

SAE reviews each technical report at least every five years at which time it may be revised, reaffirmed, stabilized, or cancelled. SAE invites your written comments and suggestions.

Copyright © 2016 SAE International

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of SAE.

TO PLACE A DOCUMENT ORDER: Tel: 877-606-7323 (inside USA and Canada)

Tel: +1 724-776-4970 (outside USA)

Fax: 724-776-0790

Email: CustomerService@sae.org

SAE WEB ADDRESS:

<http://www.sae.org>

SAE values your input. To provide feedback

on this Technical Report, please visit

http://www.sae.org/technical/standards/J3061_201601

6.	CYBERSECURITY PROCESS OVERVIEW	22
6.1	Motivation for a Well-Defined and Well-Structured Process.....	22
6.2	Process Framework	23
6.2.1	Overall Management of Cybersecurity.....	23
6.2.2	Concept Phase.....	25
6.2.3	Product Development.....	26
6.2.3.1	Product Development: System Level.....	27
6.2.3.2	Product Development: Hardware Level	28
6.2.3.3	Product Development: Software Level.....	30
6.2.4	Production, Operation & Service.....	33
6.2.5	Supporting Processes	33
6.3	Milestone and Gate Reviews	33
7.	OVERALL MANAGEMENT OF CYBERSECURITY.....	36
7.1	Cybersecurity Culture.....	37
7.2	Measuring Conformance to a Cybersecurity Process	37
7.3	Identifying and Establishing Communication Channels	38
7.4	Developing and Implementing Training and Mentoring	38
7.5	Operation and Maintenance Activities	39
7.5.1	Incident Response Process	39
7.5.2	Field Monitoring Process.....	39
8.	PROCESS IMPLEMENTATION	39
8.1	Applying a Cybersecurity Process Separately with Integrated Communication Points to a Safety Process	40
8.2	Applying a Cybersecurity Process in Conjunction with a Safety Process Tailored after ISO 26262.....	43
8.3	Concept Phase.....	44
8.3.1	Feature Definition.....	44
8.3.2	Initiation of Cybersecurity Lifecycle.....	44
8.3.3	Threat Analysis and Risk Assessment.....	44
8.3.3.1	Identifying Cybersecurity Goals	45
8.3.4	Cybersecurity Concept.....	45
8.3.5	Identify Functional Cybersecurity Requirements	45
8.3.6	Initial Cybersecurity Assessment.....	46
8.3.7	Concept Phase Review	47
8.4	Product Development: System Level.....	47
8.4.1	Initiation of Product Development at the System Level (Planning).....	48
8.4.2	System Level Vulnerability Analysis	48
8.4.3	Refine Functional Cybersecurity Concept into Technical Cybersecurity Concept.....	48
8.4.4	Specify Technical Cybersecurity Requirements	48
8.4.5	System Design	49
8.4.6	Feature Integration and Testing	49
8.4.7	Verification / Validation of Cybersecurity Technical Requirements	49
8.4.8	Final Cybersecurity Assessment / Cybersecurity Case	50
8.4.9	Final Cybersecurity Review	50
8.4.10	Release for Production.....	51
8.5	Product Development at the Hardware Level	52
8.5.1	Background	52
8.5.2	Initiation of Product Development at the Hardware Level.....	52
8.5.3	Hardware Level Vulnerability Analysis.....	52
8.5.4	Specification of Hardware Cybersecurity Requirements	53
8.5.5	Hardware Cybersecurity Design	53
8.5.6	Hardware Level Integration and Testing	53
8.5.7	Hardware Level Vulnerability Testing and Penetration Testing.....	54
8.5.8	Verification / Validation of Hardware Cybersecurity Requirements	54
8.5.9	Refine Cybersecurity Assessment.....	54

8.6	Product Development at the Software Level	55
8.6.1	Initiation of Product Development at the Software Level (Planning)	55
8.6.2	Specification of Software Cybersecurity Requirements	56
8.6.3	Software Architectural Design.....	56
8.6.4	Software Vulnerability Analysis	57
8.6.5	Software Unit Design and Implementation.	58
8.6.6	Software Implementation Code Reviews	58
8.6.7	Software Unit Testing.....	59
8.6.8	Software Integration and Testing	60
8.6.9	Verification/Validation to Software Cybersecurity Requirements.....	60
8.6.10	Software Vulnerability Testing and Penetration Testing	60
8.6.11	Refine Cybersecurity Assessment.....	60
8.7	Production, Operation and Service	61
8.7.1	Production	61
8.7.1.1	Planning	61
8.7.2	Operation, Service (Maintenance and Repair)	61
8.7.2.1	Field Monitoring.....	61
8.7.2.2	Incident Response	62
8.7.2.3	Execution and Maintenance of an Incident Response Process (15)	64
8.8	Supporting Processes (16)	65
8.8.1	Configuration Management.....	65
8.8.2	Requirements Management.....	66
8.8.3	Change Management.....	66
8.8.4	Documentation Management.....	67
8.8.5	Quality Management.....	68
8.8.6	Requirements for Distributed Development (with suppliers).....	68
9.	NOTES	69
9.1	Revision Indicator.....	69
APPENDIX A	Description of Cybersecurity Analysis Techniques	70
APPENDIX B	Example Templates for Work Products	90
APPENDIX C	Examples Using Identified Analyses	92
APPENDIX D	Security & Privacy Controls Description and Application.....	102
APPENDIX E	Vulnerability Databases and Vulnerability Classification Schemes	108
APPENDIX F	Vehicle Level Considerations.....	112
APPENDIX G	Current Cybersecurity Standards and Guidelines that may be useful to the Vehicle Industry	115
APPENDIX H	Vehicle Project Awareness	122
APPENDIX I	Security Test Tools of Potential use to the Vehicle Industry.....	127
Figure 1	Relationship between safety-critical and Cybersecurity-critical systems.....	17
Figure 2	Relationship between system safety and system Cybersecurity engineering elements	17
Figure 3	Overall Cybersecurity process framework	24
Figure 4	Concept phase activities	25
Figure 5	Relationships between product development at the system, hardware, and software levels	26
Figure 6	V diagram for product development at the system level.....	27
Figure 7	Product development: system level	28
Figure 8	V diagram showing product development at the HW level and its relationship to product development at the system level.....	29
Figure 9	Product development: hardware level.....	30
Figure 10	V diagram for product development at the software level in relation to product development at the system level	31
Figure 11	Product development: software level	32
Figure 12	Possible milestones and gate reviews	34
Figure 13	Gate review activities	36

Figure 14	Concept phase activities with potential communications paths between Cybersecurity and safety activities.....	40
Figure 15	Product development at the system level activities with potential communications paths between Cybersecurity and safety activities.....	41
Figure 16	Product development at the hardware level activities with potential communications paths between Cybersecurity and safety activities	42
Figure 17	Product development at the software level activities with potential communications paths between Cybersecurity and safety activities	43
Figure 18	Determining functional Cybersecurity requirements	46
Figure 19	Example incident response team data sources (15).....	62
Figure 20	Example incident response process	64
Figure 21	Phases of the OCTAVE method (21).....	76
Figure 22	OCTAVE allegro roadmap (22).....	76
Figure 23	Workflow of the HEAVENS security model.....	78
Figure 24	Generic attack tree.....	88
Figure 25	Attack tree	98
Figure 26	Data flow diagram of on-board diagnostics (OBD) use case.....	99
Figure 27	Relationships between faults, weaknesses, vulnerabilities and exploits	109
 Table 1	 Example incident handling checklist (15).....	65
Table 2	EVITA severity classes	71
Table 3	Rating of attack potential and attack probability	72
Table 4	Cybersecurity risk graph for privacy, financial, and operational Cybersecurity threats	72
Table 5	Controllability classifications of safety-related threats	73
Table 6	Portion of risk graph for safety-related threats	73
Table 7	Column headings for EVITA method applied at feature level using THROP	74
Table 8	Correlation of OCTAVE phases (21) and process steps to NIST SP 800-30 (16)	77
Table 9	Mapping between STRIDE threats and security attributes	79
Table 10	Applying the TL parameters to estimate threat level	81
Table 11	Estimating the threat level (TL)	82
Table 12	Impact level parameter – safety	83
Table 13	Impact level parameter – financial	83
Table 14	Impact level parameter – operational.....	84
Table 15	Impact level parameter – privacy and legislation	85
Table 16	Estimating impact level (IL).....	85
Table 17	Security level based on threat level and impact level	86
Table 18	Examples of deriving Cybersecurity requirements	86
Table 19	OCTAVE's allegro worksheet 10, information asset risk worksheet.....	90
Table 20	OCTAVE's allegro worksheet 10, risk mitigation section.....	91
Table 21	Example spreadsheet of EVITA risk assessment at feature level	93
Table 22	Example of OCTAVE's allegro worksheet 10, information asset risk worksheet - ECU firmware.....	94
Table 23	Example of OCTAVE's allegro worksheet 10, risk mitigation section – ECU firmware	95
Table 24	Example of attack tree structure for “malicious intentional vehicle disable”	97
Table 25	Threats associated with the OBD use case	100
Table 26	Risk rating of the OBD use case based on the HEAVENS methodology	100
Table 27	Asset, threat, security attribute and security level for the OBD use case	101
Table 28	Sample list of potential security control families & controls for vehicle industry	103
Table 29	Sample list of potential privacy control families & controls for vehicle industry	107
Table 30	Example on abstraction levels concerning software security issues	109
Table 31	Examples of dictionary and terminology sources for vulnerability databases	110
Table 32	Examples of Vulnerability Databases	110
Table 33	Examples of Vulnerability Classification Schemes	111
Table 34	Cybersecurity standards and guidelines that may be useful to the vehicle industry	115
Table 35	Vehicle Cybersecurity-relative research projects (2004 to present)	122
Table 36	Sample categories of security test tools	127

1. SCOPE

This recommended practice provides guidance on vehicle Cybersecurity and was created based off of, and expanded on from, existing practices which are being implemented or reported in industry, government and conference papers. The best practices are intended to be flexible, pragmatic, and adaptable in their further application to the vehicle industry as well as to other cyber-physical vehicle systems (e.g., commercial and military vehicles, trucks, busses). Other proprietary Cybersecurity development processes and standards may have been established to support a specific manufacturer's development processes, and may not be comprehensively represented in this document, however, information contained in this document may help refine existing in-house processes, methods, etc.

This recommended practice establishes a set of high-level guiding principles for **Cybersecurity** as it relates to **cyber-physical vehicle systems**. This includes:

- Defining a complete lifecycle process framework that can be tailored and utilized within each organization's development processes to incorporate Cybersecurity into cyber-physical vehicle systems from concept phase through production, operation, service, and decommissioning.
- Providing information on some common existing tools and methods used when designing, verifying and validating **cyber-physical vehicle systems**.
- Providing basic guiding principles on Cybersecurity for vehicle systems.
- Providing the foundation for further standards development activities in vehicle Cybersecurity.

The appendices provide additional information to be aware of and may be used in helping improve Cybersecurity of feature designs. Much of the information identified in the appendices is available but some experts may not be aware of all of the available information. Therefore, the appendices provide an overview of some of this information to provide further guidance on building Cybersecurity into cyber-physical vehicle systems. The objective of the overviews is to encourage research to help improve designs and identify methods and tools for applying a company's internal Cybersecurity process.

Appendices A-C - Describe some techniques for Threat Analysis and Risk Assessment, Threat Modeling and Vulnerability Analysis (e.g., Attack Trees) and when to use them.

Appendices D-I - Provide awareness of information that is available to the Vehicle Industry.

Appendix D - Provides an overview of sample Cybersecurity and privacy controls derived from NIST SP 800-53 that may be considered in design phases.

Appendix E - Provides references to some available vulnerability databases and vulnerability classification schemes.

Appendix F - Describes vehicle-level considerations, including some good design practices for electrical architecture.

Appendix G - Lists current Cybersecurity standards and guidelines of potential interest to the vehicle industry.

Appendix H - Provides an overview of vehicle Cybersecurity-related research projects starting from 2004.

Appendix I - Describes some existing security test tools of potential interest to the vehicle industry.

Refer to the definitions section to understand the terminology used throughout the document.

1.1 Purpose

Just as with system safety, Cybersecurity should be built in to the design rather than added on at the end of development. Building Cybersecurity into the design requires an appropriate lifecycle process from the concept phase through production, operation, service, and decommissioning. This document provides a complete lifecycle process framework that may be tailored to a company-specific process. The process framework described in this document is analogous to the process framework described in ISO 26262 Functional Safety Road Vehicles (1). These two processes are different, but are related and require integrated communications in order to maintain consistency and completeness between an organization's safety process outputs and their Cybersecurity process outputs. An organization is free to maintain separate processes with appropriate levels of interaction between the two processes, or to attempt to directly integrate the two processes. The Cybersecurity process framework described in this document can be tailored to either application (integrated or separate) by individual organizations.

1.2 When to Apply a Cybersecurity Process

For systems that may be considered Cybersecurity-critical cyber-physical vehicle systems, an initial short assessment of potential threats related to, for example, operation, privacy (e.g., PII, eavesdropping), financial, reputation, and an initial estimate of risks can be made to determine if the system being considered should follow a Cybersecurity process. The quick assessment version of a threat analysis and risk assessment to determine applicability of a Cybersecurity process may consist of a short brainstorming and discussion meeting to consider potential threats associated with the feature, and to consider whether the potential risks of the threats may be high enough to warrant following a Cybersecurity process. The risk assessment portion of the initial assessment may be based on experience or expert judgment rather than on a rigorous assessment process. Potential brainstorming could come from knowledge gained by hacker chatter and conferences, previous experiences, checklists, etc. Examples of issues that might be considered in determining the risk include an estimate of the magnitude of the impact, from financial, safety, privacy, or operational aspects, and whether an attack may involve a fleet of vehicles or a single vehicle.

For potential safety-related vehicle features, it is recommended that an initial short assessment of potential safety-related threats be performed to determine if there are any potential high-risk safety-related threats. If the initial assessment indicates that high risk safety-related threats may exist, then a Cybersecurity process should be applied. If the initial assessment does not identify any high risk potential safety-related threats, the Cybersecurity process may not need to be applied with respect to the low-risk safety-related threats; it is up to an organization to determine what is considered low risk and whether low risk safety-related threats need to be addressed. To help ensure that all potential safety-related threats are considered, the Cybersecurity experts should communicate with the safety experts. Note that the basis of decision for following the process is on the identified potential risk of the identified safety-related threats rather than on whether a corresponding potential hazard is ASIL rated (A, B, C, or D). This is because the threat risk for a safety-related threat may be low, while the corresponding hazard may be assessed a high ASIL; there is no direct correspondence between an ASIL rating and the potential risk associated with a safety-related threat.

2. REFERENCES

1. ISO 26262 Part 8 First Edition, "Supporting Processes, Road Vehicles – Functional Safety", 11-15-2011.
2. Barbara J. Czerny. "System Security and System Safety Engineering: Differences and Similarities and a System Security Engineering Process Based on the ISO 26262 Process Framework", SAE Technical Paper 2013-01-1419, SAE World Congress and Exhibition, 2013.
3. B. Potter. 'Microsoft SDL Threat Modelling Tool'. In: *Network Security* 2009.1 (2009), pp. 15–18. ISSN: 1353-4858.
DOI:[http://dx.doi.org/10.1016/S1353-4858\(09\)70008-X](http://dx.doi.org/10.1016/S1353-4858(09)70008-X).
URL: <http://www.sciencedirect.com/science/article/pii/S135348580970008X> (cit. on p. 37).
4. Iván Arce, Kathleen Clark-Fisher, Neil Daswani, Jim DelGrosso, Danny Dhillon, Christoph Kern, Tadayoshi Kohno, Carl Landwehr, Gary McGraw, Brook Schoenfeld, Margo Seltzer, Diomidis Spinellis, Izar Tarandach, and Jacob West. "Avoiding the Top 10 Software Security Design Flaws", IEEE Computer Society, 2014.

5. Global Alliance, Global Automakers, "Consumer Privacy Protection Principles for Vehicle Technologies and Services", November 12, 2014.
6. NIST, SP 800-88, Revision 1, "Guidelines for Media Sanitization", December, 2014.
7. ISO/IEC 15408 "Information Technology – Security Techniques – Evaluation Criteria for IT Security", (3 Parts).
8. NIST, Version 1, "Framework for Improving Critical Infrastructure Cybersecurity", February 12, 2014.
9. NIST SP 800-53, Revision 4, "Security and Privacy Controls for Federal Information Systems and Organizations", April 2014.
10. FIPS Pub 199. "Standards for Security Categorization of Federal Information and Information Systems", February 2004.
11. ISO (International Organization for Standardization). "ISO 12207 - Systems and Software Engineering - Software LifeCycle Processes", 2008.
12. ISO (International Organization for Standardization). "ISO/IEC 27001: - Information technology - Security techniques - Information security management systems - Requirements". International Organization for Standardization. 27 January 2015.
13. ISO (International Organization for Standardization). "ISO/IEC 27002: Information Technology - Security Techniques. Code of Practice for Information Security Controls" 2008.
14. ISO (International Organization for Standardization). "ISO/IEC 29119: The International Software Testing Standard", September 10, 2014.
15. NIST 800-61 Revision 2, "Computer Security Incident Handling Guide", August 2012.
16. NIST Special Publication 800-30 Revision 1, "Guide for Conducting Risk Assessments", September 2012.
17. Chrysler Corporation, Ford Motor Company, General Motors Corporation, QS 9000 Third Edition, "Quality System Requirements", October 1998.
18. ISO/TS 16949:2009 "Quality Management Systems" December 2008.
19. Ruddle, Alastair, Ward, David, *et al*, EVITA Project Deliverable D2.3: "Security requirements for automotive on-board networks based on dark-side scenarios" Version 1.1, 30 December 2009.
20. EVITA deliverable D2.1: "Specification and evaluation of e-security relevant use cases", 2009.
21. Woody, Carol. "Applying OCTAVE: Practitioners Report." Software Engineering Institute, May 2006.
22. Caralli, Richard, James Stevens, Lisa Young, and William Wilson. "Introducing OCTAVE Allegro: Improving the Information Security Risk Assessment Process." Software Engineering Institute, May 2007.
23. M. Islam *et al.*, *Deliverable D2 Security models*. HEAVENS Project, Deliverable D2, Release 1. Dec. 2014.
24. ISO (International Organization for Standardization). *Road vehicles - Functional safety* ISO 26262:2011. (cit. on pp. 17, 30, 40, 42, 44).
25. BSI-Standard 100-4, Version 1.0, 2009, Federal Office for Information Security (BSI), Germany.
26. Automotive Industry Action Group (AIAG), "Potential Failure Mode and Effects Analysis (FMEA)", 2008.
27. "Privacy Impact Assessment Guideline", 2011, Federal Office for Information Security (BSI), Germany.

28. ISO (International Organization for Standardization). *Road vehicles - Functional safety - Part 3: Concept phase* (ISO 26262-3:2011). ISO 26262-3:2011. 2011 (cit. on p. 18).
29. Schneier, Bruce, "Secrets and Lies – Digital Security in a Networked World", Wiley, ISBN 978-0-471-45380-2.
30. Amer Ajiaz1, Bernd Bochow2, Florian Dötzter3, Andreas Festag4, Matthias Gerlach2, Rainer Kroh5 and Tim Leinmüller5 , "Attacks on Inter Vehicle Communication Systems – an Analysis".
31. http://www7.informatik.uni-erlangen.de/~dulz/fkom/06/Material/11/Security/NOW_TechReport_Attacks_on_Inter_Vehicle_Communications.pdf.
32. Avizienis A, Laprie J-C, Randell B, Lanwehr C, "Basic Concepts and taxonomy of Dependable and Secure Computing", IEEE Transactions on independent and Secure Computing". January-March 2004.
33. MITRE Corporation, "Common Weakness Enumeration, A Community Developed Dictionary for Software Weakness Types ",2006 -2014, <http://cwe.mitre.org/>.
34. MITRE Corporation, "Common Vulnerabilities and Exposures ", 1999-2014, <https://cve.mitre.org/cve/index.html>.
35. Security Focus, 'BugTraq', 2010, <http://www.securityfocus.com/archive>.
36. NIST, 'National Vulnerability Database', <http://nvd.nist.gov/>.
37. MITRE Corporation, 'Common Weakness Scoring System', 2006-2014, http://cwe.mitre.org/cwss/cwss_v1.0.1.html.
38. NIST, 'Common Vulnerability Scoring System', <http://nvd.nist.gov/cvss.cfm>.

3. DEFINITIONS AND ACRONYMS

3.1 API

Application Programming Interface

3.2 ASF - Application Security Frame

Threat categorization tool that determines threats based on system break down methodology.

3.3 ASIL - Automotive Safety Integrity Level

A means of classifying hazards in ISO 26262.

3.4 ATTACK POTENTIAL

The likelihood that a potential attack can be successfully carried out.

3.5 ATTACK SURFACE

The different points (the "attack vectors") where an unauthorized user (the "attacker") can try to enter data to or extract data from an environment.

3.6 ATTACK TREE ANALYSIS (ATA)

An analysis method to determine the potential paths that an attacker could take through the system to lead to the top level threat.

3.7 BLACK BOX TESTING

Testing where nothing formal is known about the system being tested. No specifications, hardware information, software code, etc. are provided during the testing.

3.8 CAN - Controller Area Network

A serial communication network. The following standards provide the specifics associated with the CAN protocol and some of its automotive variants: SAE J1939, SAE J2411, ISO 11898, ISO 15765-2.

3.9 CERT C

Secure coding standard for C, C++, Java and Pearl. Developed by the CERT coding initiative team.

3.10 CPU - Central Processing Unit

The part of a computer system (a microcontroller) that performs the computer's main functions and controls parts of the system.

3.11 COMMON VULNERABILITY ENUMERATION (CVE™)

CVE is designed to allow vulnerability databases and other capabilities to be linked together, and to facilitate the comparison of security tools and services.

3.12 COMMON VULNERABILITY SCORING SYSTEM (CVSS)

Scoring system for vulnerabilities.

3.13 COMMON WEAKNESS ENUMERATION (CWE™)

The CWE is a “formal list of software weakness types” hosted by MITRE cooperation.

3.14 COMMON WEAKNESS SCORING SYSTEM (CWSS™)

Is a scoring system that may help stakeholders concerned with software security - to assess and – where applicable - prioritize reported software weakness.

3.15 CYBER-ATTACK

An assault on system Cybersecurity that derives from an intelligent act, i.e., an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade Cybersecurity services and violate the Cybersecurity policy of a system.

3.16 CYBER-PHYSICAL SYSTEM (CPS)

A system of collaborating computational elements controlling physical entities.

3.17 CYBER-PHYSICAL VEHICLE SYSTEM (CPAS)

Vehicle embedded control systems where there exists a tight coupling between the computational elements and the physical elements of the system and the environment around the system.

3.18 CYBERSECURITY

Measures taken to protect a cyber-physical system against unauthorized access or attack.

3.19 CYBERSECURITY ASSESSMENT

An assessment of the level of Cybersecurity of a feature that will be refined throughout the development process and provides appropriate arguments and evidence to support Cybersecurity claims at each stage of development. The Cybersecurity assessment is reviewed at each of the major milestones.

3.20 CYBERSECURITY CASE

The final Cybersecurity Assessment after all milestone reviews have been completed and before the feature can be released for production. The Cybersecurity case provides the final argumentation and evidence that the feature as designed and developed satisfies its Cybersecurity goals.

3.21 CYBERSECURITY CONCEPT

Developed in the concept phase to describe the high-level Cybersecurity strategy for the feature. The Cybersecurity concept will be refined to a technical Cybersecurity concept during product development at the system level.

3.22 CYBERSECURITY CONTROLS

The management, operational, and technical controls (e.g., safeguards or countermeasures) prescribed for a feature to eliminate potential vulnerabilities or to reduce the likelihood that a vulnerability will be exploited.

3.23 CYBERSECURITY-CRITICAL

A system where losses could occur in cyber-physical systems due to vulnerabilities in the system that could be exploited either directly or indirectly by an outside entity.

3.24 CYBERSECURITY GOALS

The goals for achieving Cybersecurity for the feature determined from the threat analysis and risk assessment results. These are the highest level Cybersecurity requirements that will drive the development and refinement of functional and technical Cybersecurity requirements.

3.25 CYBERSECURITY MECHANISM

Technical Cybersecurity control added to the feature to eliminate potential vulnerabilities or to reduce the likelihood that a vulnerability will be exploited.

3.26 CYBERSECURITY POTENTIAL

The level of risk or likelihood that something may happen.

3.27 CYBERSECURITY PROGRAM PLAN

Defines responsibilities for planning and overseeing the Cybersecurity activities.

3.28 CYBERSECURITY REVIEW

A review, possibly conducted by a small team of technical reviewers, to assess the technical aspects of the work products during the various stages of the development process.

3.29 DIS

Draft International Standard

3.30 DOD

Department of Defense

3.31 DREAD - Damage Reproducibility Exploitability Affected users and Discoverability

Threat categorization tool that determines threats based on system break down methodology.

3.32 DVD - Digital Video Disc

A device with high storage capacity of information.

3.33 ECU - Electronic Control Unit

A module that provides a function to the vehicle.

3.34 ETHERNET

A serial communication network.

3.35 ETSI

European Telecommunications Standards Institute

3.36 EVITA - E-safety Vehicle Intrusion protected Applications

A project initiated by the European Community from 2007 through 2013. Primarily to design, verify and prototype Cybersecurity building blocks for vehicle on-board networks.

3.37 FEATURE

System or an array of systems to implement a function at the vehicle level to which a Cybersecurity process for cyber-physical vehicle systems is applied.

3.38 FAULT TREE ANALYSIS (FTA)

A deductive analysis technique that starts with a top-level hazard and works down to identify potential single and multi-point failure combinations that can cause the hazard to occur.

3.39 FLEXRAY

A serial communication network.

3.40 FUZZ TESTING

A software testing technique that can be used to find potential security flaws.

3.41 GREY BOX TESTING

Testing where partial information is known about the feature being tested. For example, some feature specifications are provided but the product source code was not provided. Thus, some ad hoc methods are still required to try to determine the vulnerabilities.

3.42 GPS

Global Positioning System, used for navigation.

3.43 HACKER

A person who illegally attempts to gain access to or gains access to a system with the intent to gain something or to cause losses from a stakeholder perspective; e.g., fame, financial, terrorist attack.

3.44 HACKER CHATTER

On line blogs or conventions, etc. where hackers hold conversations about what they try to do.

3.45 HACKER INTRUSION

An unauthorized access.

3.46 HAZOP

Hazard and Operability Analysis, in the context of functional safety, is a structured and systematic technique for identifying potential hazards of a feature; the method uses guidewords and brainstorming to attempt to identify potential hazards.

3.47 HEAVENS

HEAling Vulnerabilities to ENhance Software Security and Safety

3.48 HMI

Human Machine Interface

3.49 HSM - Hardware Security Module

A physical computing device that safeguards and manages digital keys for strong authentication and provides cryptoprocessing.

3.50 HM

Hardware Module

3.51 IEC - International Electrotechnical Commission

Group authoring industry standards.

3.52 INCIDENT

An attack on the system that may have or may not have been successful.

3.53 ISAC - Information Sharing and Analysis Centers

A central repository for security-related information. The group's purpose is to share each organization's information about Cybersecurity attacks and vulnerabilities among all the members.

3.54 ISO/TS

International Organization for Standardization / Technical Specifications

3.55 I/O

Input / Output

3.56 IT - Information Technology

Resource management activity.

3.57 ITS

Intelligent Transportation System

3.58 JTAG

Port on a microprocessor used to extract data or code from an ECU.

3.59 LIN - Local Interconnect Network

A serial communication network.

3.60 MALICIOUS ACTORS

A person or persons with the intent to identify and exploit vulnerabilities within a feature to achieve access to the feature for personal or group gain; the gains may be for fame, financial, malicious intent, etc.

3.61 MISRA C

A software development standard for the C programming language developed by Motor Industry Software Reliability Association (MISRA). It aims to facilitate code safety, portability, and reliability in the context of embedded systems.

3.62 MOST - Media Oriented Systems Transport

A serial communication network (fiber optic or electrical).

3.63 NIST - National Institute of Standards and Technology

The U.S. Department of Commerce pulls groups of people together to draft standards in various areas of technology.

3.64 NVD - National Vulnerability Database

Contains more than 57000 vulnerabilities entered by NIST.

3.65 OBD-II - On Board Diagnostics Connector

Access point in vehicle to be able to communicate to modules and pull diagnostic information.

3.66 OCTAVE - Operationally Critical Threat, Asset, and Vulnerability Evaluation

A threat analysis and risk assessment method for assessing risk in existing enterprise information systems.

3.67 OEM - Original Equipment Manufacturer

A large vehicle manufacturer corporation.

3.68 PENETRATION TESTING

A testing method with the intent to penetrate a feature in order to identify unknown vulnerabilities and to determine vulnerabilities that are not adequately protected. Penetration testing uncovers critical issues and demonstrates how well the feature is protected. Combined with a comprehensive Cybersecurity program, penetration tests can help you reduce the risk of a breach.

3.69 PII - Personally Identifiable Information

Is information that can be used on its own or with other information to identify, contact, or locate a single person, or to identify an individual in context.

3.70 QS

Quality System. (e.g., QS 9000)

3.71 RAM

Read Access Memory

3.72 ROM

Read Only Memory

3.73 RISK ANALYSIS METHOD

A process for analyzing the potential risk of identified threats with respect to the severity of the possible outcome of an attack and the likelihood that a potential attack can be successfully carried out (the attack potential).

3.74 SAE

Society of Automotive Engineers

3.75 SAFETY-CRITICAL SYSTEMS

A system that may cause harm to life, property, or the environment if the system does not behave as intended or desired.

3.76 SCADA

Supervisory Control and Data Acquisition is a system operating with coded signals over communication channels so as to provide control of remote equipment.

3.77 SDL - Security Development Lifecycle

The Security Development Lifecycle (SDL) is a software development process that helps developers build more secure software and address security compliance requirements while reducing development cost.

3.78 SEI

Software Engineering Institute at Carnegie Mellon.

3.79 STAKEHOLDER

A group, organization or member that can affect or can be affected by an organizations actions.

3.80 SQL

Structured Query Language is a special-purpose programming language designed for managing data held in a relational database management system (RDBMS).

3.81 STRIDE

Threat modeling technique by Microsoft. Stands for Spoofing identity, Tampering with data, Repudiation, Information disclosure, Denial of service, and Elevation of privilege.

3.82 SW

Shorthand for Software.

3.83 SYSTEM

A collection of hardware and software to perform a function or functions in a vehicle.

3.84 SYSTEM CONTEXT

To define the interfaces between the system's hardware and software, the key data flows, storage and processing within the system.

3.85 TECHNICAL PHASE REVIEW

A technical phase review is used at the end of a development phase to review the technical activities performed and the technical documents created during the development phase; for example, a Concept Phase Technical Review would be performed at the completion of the concept phase of development to help ensure that the technical activities and documents generated during the concept phase are correct, consistent, complete, etc. The technical phase review is best performed by a technical review board consisting of a small team of technical experts (3-4). The work products being reviewed should be provided to the review board members at least 2-3 weeks prior to the review board meeting.

3.86 TECHNICAL CYBERSECURITY CONCEPT

The high-level Cybersecurity strategy defined into engineering terms.

3.87 TECHNICAL REVIEW

A review of the technical correctness, technical completeness, and technical consistency of the work products developed as part of a Cybersecurity process. A technical review may be performed as each individual analysis activity or work product is completed, or at set gate review points throughout the development lifecycle. This review may be done by a technical review board of a small number of (e.g., 3-4) technical experts and the work product or analysis activity being reviewed should be provided to the review board members enough in advance (e.g., two to three weeks) of the technical review to allow the review board ample time to review the work products.

3.88 TARA - Threat Analysis and Risk Assessment

An analysis technique that is applied in the concept phase to help identify potential threats to a feature and to assess the risk associated with the identified threats. Identifying the potential threats and assessing the risk associated with these threats, allows an organization to prioritize follow-on Cybersecurity activities associated with the threats so efforts and resources can be focused on the highest priority threats.

3.89 TATRC

Telemedicine and Advanced Technology Research Center.

3.90 TIER 1

A supplier sourced directly by a vehicle manufacturer for a given product. The supplier has a direct business agreement with the vehicle manufacturer.

3.91 THREATS

A circumstance or event with the potential to cause harm, where harm may be with respect to financial, reputation, privacy, safety, or operational.

3.92 THROP - Threat and Operability Analysis

An analysis technique that applies guidewords to identified primary functionality of a feature to identify potential threats associated with the feature. A THROP parallels a HAZOP except that it considers potential threats rather than potential hazards.

3.93 TRUST BOUNDARY

A boundary where program data or execution changes its level of "trust". An example of an execution trust boundary would be where an application attains an increased privilege level (such as root). A data trust boundary is a point where data comes from an untrusted source.

3.94 TVRA

Threats, Vulnerabilities and Risks (TVR) of a system to be analyzed. A TARA method.

3.95 Unauthorized Access

If a user attempts to access an area of a system they should not be accessing. When attempting to access that area, they would be denied access.

3.96 USB - Universal Serial Bus

A means to store and communicate information to others.

3.97 VALIDATION

The assurance that a product, service, or system meets the needs of the customer and other identified stakeholders. It often involves acceptance and suitability with external customers. Contrast with verification.

3.98 VERIFICATION

The evaluation of whether or not a product, service, or system complies with a regulation, requirement, specification, or imposed condition. It is often an internal process. Contrast with validation.

3.99 VULNERABILITY ANALYSIS

Vulnerability analysis techniques attempt to identify and classify potential Cybersecurity vulnerabilities or holes in the software and hardware of the feature being developed that may be exploited by an attacker.

3.100 WELL DEFINED AND WELL STRUCTURED (WDWS) PROCESS

Establishes a repeatable, structured method to systematically identify and assess threats, and vulnerabilities that could be exploited to achieve a threat, and appropriate Cybersecurity Controls to design in to the system during development to protect against the identified vulnerabilities.

3.101 WHITE BOX TESTING

Testing where full information is known about the feature being tested. Detailed specifications and source code is available during the testing.

4. RELATIONSHIP BETWEEN SYSTEM SAFETY AND SYSTEM CYBERSECURITY¹

System safety (beyond regulatory requirements) is the state of a system that does not cause harm to life, property, or the environment. System Cybersecurity is the state of a system that does not allow exploitation of vulnerabilities to lead to losses, such as financial, operational, privacy, or safety losses. A safety-critical system is a system that may cause harm to life, property, or the environment if the system does not behave as intended or desired. A Cybersecurity-critical system is a system that may lead to financial, operational, privacy, or safety losses if the system is compromised through a vulnerability that may exist in the system. All safety-critical systems are Cybersecurity-critical since a cyber-attack either directly or indirectly on a safety-critical system could lead to potential safety losses (Figure 1). Not all Cybersecurity-critical systems are safety-critical since cyber-attacks on Cybersecurity-critical systems can result in losses other than safety losses; namely, privacy, operational, or financial.

The two domains are also related in that there is some overlap between the elements of system safety engineering and the elements of system Cybersecurity engineering, but the elements are not identical between the two engineering disciplines (Figure 2). An example of a Cybersecurity-critical system that is not safety-critical is an entertainment system that obtains personal information from the driver. If this system is compromised, it may lead to financial or privacy losses to the driver, however, it most likely would not cause physical harm to the driver; thus, the system is Cybersecurity-critical, but not safety-critical. An example of a system that is both Cybersecurity-critical and safety-critical is a steering assist system. A steering assist system is safety-critical since if it exhibits malfunctioning behavior it could lead to potential harm to the vehicle occupants. The steering assist system is also Cybersecurity-critical since if the system is compromised by an attacker and a malicious intentional steering maneuver is injected, this could also lead to potential harm to the vehicle occupants; in Cybersecurity, this would be analyzed as a potential safety loss.

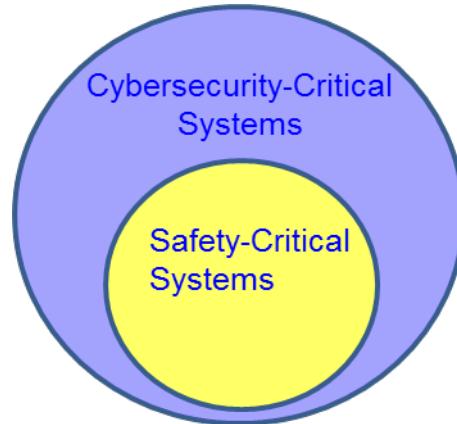


Figure 1 - Relationship between safety-critical and Cybersecurity-critical systems

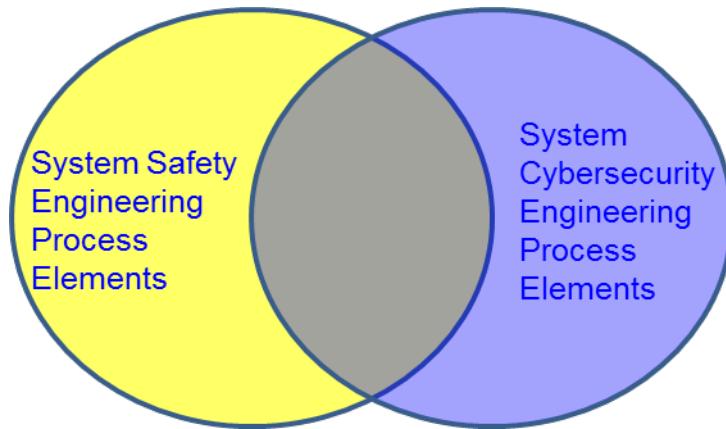


Figure 2 - Relationship between system safety and system Cybersecurity engineering elements

1. The information in this chapter was taken from: SAE Technical Paper 2013-01-1419 – System Security and System Safety Engineering: Differences and Similarities and a System Security Engineering Process Based on the ISO 26262 Process Framework (2).

4.1 Analogies between System Safety and System Cybersecurity Engineering

The objectives of system safety and system Cybersecurity engineering are analogous to each other. The goal of both engineering activities is, if possible, to build safety into the design, or to build Cybersecurity into the design, rather than attempting to add safety and Cybersecurity on to an existing design. The systems engineering aspect is important in both system Cybersecurity and system safety. However, in Cybersecurity, there can be a tendency to only consider the problem to be one of simply adhering to best practices (e.g., authentication and cryptography), and ignoring the system engineering aspects. This section uses the term “system” in front of Cybersecurity to stress a systems approach to Cybersecurity as described in this Recommended Practice.

The process elements of system safety and system Cybersecurity engineering are also analogous to each other. In the concept phase of a system safety engineering process, a hazard analysis and risk assessment is performed. Analogous to this in a system Cybersecurity engineering process, a threat analysis and risk assessment is performed during the concept phase. In the requirements phase of a system safety engineering process, safety requirements are derived and refined from the safety goals identified in the hazard analysis and risk assessment. Likewise, in the requirements phase of a system Cybersecurity engineering process, Cybersecurity requirements are derived and refined from the Cybersecurity goals identified in the threat analysis and risk assessment. In the design phase of a system safety engineering process, detailed hazard analysis is performed on the highest risk hazards to help identify controls or safety mechanisms to help eliminate the potential hazards, or to help mitigate the consequences should a potential hazard occur. Likewise, in the design phase of a system Cybersecurity engineering process, detailed vulnerability analysis is performed on high risk identified threats to help identify Cybersecurity controls to apply to help reduce the likelihood of a successful attack. Analogies between process elements continue through product development and verification/validation. Though there are many similarities among the process elements between system safety and system Cybersecurity engineering, the underlying application of the process elements may be unique between the two engineering disciplines.

Suggestions for maintaining consistency between safety & Cybersecurity include:

- Build appropriate checkpoints into the product lifecycle of both processes.
- Use a **Risk Analysis Method** to help guide the corporation to address the threats of highest riskBuild Cybersecurity awareness and interface or communication points into existing processes and forums.
- Identify, establish and distribute the various communication paths between safety and Cybersecurity.

4.2 Unique Aspects of System Safety and System Cybersecurity

System safety and system Cybersecurity are unique from each other. Whilst system safety focuses on analyzing the system for potential hazards so that safety mechanisms can be identified and integrated into the design to address the causes of the potential hazards and to reduce the risk associated with those potential hazards, system Cybersecurity considers potential threats posed by an attacker whose goal is to cause harm, wreak havoc, gain financial or other benefits, or simply to gain notoriety.

Though there are analogies between “hazard analysis and risk assessment” and “threat analysis and risk assessment,” the response to identified hazards is different from the response to identified threats. Since potential threats involve intentional, malicious, and planned actions, they are more difficult to address than potential hazards. Addressing potential threats fully, requires the analysts to think like the attackers, but it can be difficult to anticipate the exact moves an attacker may make. Predicting an attacker’s moves, however, helps the analysts to know what Cybersecurity controls are appropriate for protecting against the attacker’s possible actions. In system safety, analysts can more readily identify the potential hazards, identify the potential causes and take appropriate action to mitigate the potential consequences, or to eliminate the potential hazards all together. The reason is that causes of potential hazards can be determined based on experience, knowledge of the system, components, and interactions, etc., and potential causes may be unique for particular systems, but are not unknown. In addition, with respect to safety, statistics may be used for claiming an acceptable level of risk. However, this may not be the case for Cybersecurity. For Cybersecurity, statistics or statistical analysis techniques may need to be employed in some manner due to the large amount of unknown information likely to be part of a Cybersecurity analysis, however, these techniques are likely fundamentally different than the empirical techniques that can be used based on experience in safety-related systems.

A further difference between hazard analysis and risk assessment and threat analysis and risk assessment is that additional factors are considered with respect to threats and the risks associated with threats, that need not be considered with respect to hazards and the risks associated with hazards. These additional factors to consider in assessing the risks of potential threats include the knowledge required by an attacker (proprietary or publically available), the experience level of an attacker, the access to the system that is required by an attacker, an attacker's need for special equipment, etc. None of these factors need be considered in assessing the risks of potential hazards.

Another unique aspect with respect to system Cybersecurity and system safety is that with system Cybersecurity a broader focus is considered. In system safety the focus is on safety-critical systems, whereas in system Cybersecurity, both safety-critical and non-safety-critical systems are considered. System Cybersecurity considers both safety-critical and non-safety-critical systems since threats may be non-safety-related (e.g., financial, privacy, operational), it may be possible to access a safety-critical vehicle system from non-safety-critical vehicle systems (e.g., in-vehicle entertainment systems). In addition, any safe state identified as part of the system safety engineering analyses needs to be considered with respect to Cybersecurity to assess whether that safe state could be exploited by an attacker. Even if exploiting safe states seems acceptable and would not lead to a safety threat, it should also be evaluated to determine if it could result in denial of service of another feature. It is possible that a safe state may have some potential risk associated with it; these safe states could be exploited by an attacker and may lead to a safety threat that would need to be analyzed from a Cybersecurity perspective. Thus, traditional hazard controls are not sufficient as Cybersecurity controls and as stated, may be used against a system by causing a denial of service or a potential safety-related threat to occur.

With respect to detailed hazard analysis and vulnerability analysis, the analysis techniques may be analogous to each other, but the methods and goals are unique. For example, a detailed hazard analysis technique may utilize **Fault Tree Analysis (FTA)**. Similarly, in system Cybersecurity, a detailed threat analysis technique may utilize **Attack Tree Analysis (ATA)**. Though the methods are analogous to each other, they are unique. In Fault Tree Analysis the analyst identifies potential causes of the top hazard event and looks for single-point and multi-point random hardware failures that can lead to the top-level hazard. Attack Tree Analysis is not concerned with single-point and multi-point random hardware failures, but rather with determining potential paths that an attacker could take through the system to lead to the top level threat. The underlying goals are synonymous – in FTA the goal is to identify single and multi-point random hardware failures that may lead to the top hazard so safety mechanisms can be added to detect and mitigate potential causes, and in ATA the goal is to identify potential vulnerabilities that could be exploited to lead to the top level threat so that Cybersecurity controls can be identified to eliminate the vulnerabilities or to make them more difficult to exploit.

In the implementation and verification/validation stages, static code analysis used in system safety is used to help identify bugs that directly affect primary functionality. In system Cybersecurity, static code analysis is used to identify potential Cybersecurity vulnerabilities in the code. Valid or correct code from a safety perspective may still have Cybersecurity vulnerabilities.

Finally, some verification/validation methods for Cybersecurity are different and more difficult than verification/validation methods used in system safety. For example, in system safety fault injection tests are performed to verify that the identified faults are detected and that the appropriate response occurs, however, in Cybersecurity there is no particular fault that can be injected to see if the system vulnerabilities are closed. Cybersecurity relies on attack (vulnerability) testing or penetration testing vs. fault injection testing. In penetration testing, a pseudo attacker(s) attempts to identify and exploit vulnerabilities in the system. This is clearly not as straightforward as fault injection testing. Penetration testing is also not intended to confirm that the correct response (i.e., specific Cybersecurity controls added to the design) has been made relative to a potential vulnerability. Alternatively, traditional structured testing of the effectiveness of Cybersecurity Controls (which can confirm that the design meets its requirements) is not sufficient to address black-hat out-of-the-box types of attack. Cybersecurity may use both traditional structured testing and penetration testing to address the unpredictability of attacker methods.

A more general difference between Cybersecurity and safety is that Cybersecurity risks evolve over time as attacker's motivations and capabilities change. This makes Cybersecurity especially difficult, since it involves defense against techniques that may not be understood at the time the system is created.

It is also possible for safety and Cybersecurity to conflict with each other in some cases. For example, Cybersecurity countermeasures can conflict with safety requirements and vice versa. Systems engineering principles consider the overall integration of requirements for the system that includes integrating Cybersecurity and safety requirements. To help maintain consistency and completeness between safety and Cybersecurity, various communications points between safety and Cybersecurity should be identified and established. Also appropriate checkpoints or gate reviews should be added in the product lifecycle between safety and Cybersecurity.

5. GUIDING PRINCIPLES ON CYBERSECURITY FOR CYBER-PHYSICAL VEHICLE SYSTEMS (CPS)

The Guiding Principles for Cybersecurity presented in this section are intended to work for a wide variety of companies in the vehicle industry. Since each company is likely to have its own internal processes by which it manages its product development, this Recommended Practice provides a set of guiding principles with respect to Cybersecurity that can be applied by any organization within a company. The following guiding principles are tailored for cyber-physical vehicle systems Cybersecurity from Microsoft's ***Security Development Lifecycle (SDL)*** guiding principles (3), and IEEE's **Avoiding the Top 10 Software Security Design Flaws** (4). In addition to these guiding principles, identify legislation or regulatory requirements that may be applicable with respect to Cybersecurity.

5.1 Know Your System's Cybersecurity Potential

It is very important to understand what the potential Cybersecurity vulnerabilities are for your system (e.g., attack surfaces that can be identified by conducting the appropriate vulnerability analyses). The concept phase for system development should consider what defense to use for these potential vulnerabilities. For example:

- Will there be any Sensitive data and/or Personally Identifiable Information (PII) stored on, or transmitted by, your system that could make your system a target?
- What role does your system have (if any) in the safety-critical functions of a vehicle?
- What communications or connections will your system have with entities that are external to the vehicle's electrical architecture?
- Can your system be used as a "stepping stone" to an attack on another system?
- Can information about your system (e.g., timing, power consumption) be used to mount a side channel attack?
- Conduct the appropriate Threat Analysis and Risk Assessment.

5.2 Understand Key Cybersecurity Principles

The following lists some key principles relative to Cybersecurity for cyber-physical systems.

- Protect Personally Identifiable Information (PII) and Sensitive data: One potential reference source that provides guidance on how to do this can be found in the Auto Alliance and Global Automakers Consumer Privacy Protection Principles (5). PII stored on the vehicle should be protected, and access to stored PII data should be controlled and limited:
 - Utilize conservative default access settings for customers' data.
 - Obtain appropriate consent from the responsible body before collecting or transferring any data.
 - Prevent **unauthorized access** by protecting customers' data stored in access control lists.
- Use the principle of "Least Privilege" - All components run with the fewest possible permissions.
- Apply "Defense in Depth", particularly for the highest risk threats. This means that threat mitigation should not rely on only a single Cybersecurity Control while leaving other vulnerabilities in the system that could be exploited if the primary Cybersecurity control is penetrated.

- Prohibit changes to calibrations and/or software that have not been thoroughly analyzed and tested.
- Prevent vehicle owners from intentionally or unintentionally making unauthorized changes to the vehicle's systems that could introduce potential vulnerabilities. Some ways vehicle owners may introduce vulnerabilities include:
 - "Tuners" who change calibration settings and/or software to get different powertrain performance features,
 - Software from devices such as **DVD's**, Bluetooth-paired phone's, etc. that may attempt to install itself via the vehicle's entertainment systems, without informing the user or telling the users about possible risks. The installed software may not have malicious intent, however, it may have vulnerabilities that may be exploited.

5.3 Consider the Vehicle Owners' Use of the System

Consider how your system will be used by the owner of the vehicle your system will be in.

- Minimize data collection. Collect the minimum amount of personal data that is required for a particular purpose, and use the least sensitive form of that data (e.g., User name is less sensitive than social security number).
- Enable user policy and control. Enable owners to manage privacy settings on their vehicle systems, provide authorization where applicable, and update/revoke authorization when they wish. Also enable manufacturers to manage privacy settings for their operations.
- Protect the storage, usage, and transfer of PII. Ensure that data usage complies with uses communicated to the OEM and the vehicle owners.
- Provide appropriate notice about data that is collected, stored, or shared so that owners can make informed decisions about their personal information.
- Develop appropriate material for dealerships, customer assistance help lines, websites, and owner's manuals. The goal of this material is to set customer expectations relative to data privacy, and to inform them of the capabilities and limitations of the systems, as well as promote general cybersecurity practices.

5.4 Implement Cybersecurity in Concept and Design Phases

- Design the feature with Cybersecurity in mind, starting in the concept phase of the development lifecycle. Engineers should consider Cybersecurity when defining the requirements that are to be met for the system and feature(s).
- Analyze threats (i.e., initiated external or internal to the system) to determine what will be faced by the system. For the determined threats, identify any vulnerabilities and determine the appropriate Cybersecurity controls.
- Implement Cybersecurity analysis (and management tools) that enable engineers to determine and configure the optimal Cybersecurity level for the system.

5.5 Implement Cybersecurity in Development & Validation

- Have status reviews to assess whether design work is on track to meeting the Cybersecurity requirements. For any Cybersecurity requirements that are at risk of not being met, work with appropriate stakeholders to develop a plan to resolve the open issues.
- Conduct testing to confirm the requirements that were established for Cybersecurity have been met in the feature.
- Ensure that any risks associated with the mechanism for doing re-flashes to the feature/vehicle software are minimized or eliminated.

5.6 Implement Cybersecurity in Incident Response

- Revise (or create) an Incident Response Process that comprehends both the tracking of, and the response to, Cybersecurity incidents. This Incident Response Process should emphasize the importance of responding promptly to reports of Cybersecurity vulnerabilities/incidents, and of communicating information about security updates to affected users and stakeholders. These Incident Response processes need to be documented and published.
- Determine how software and/or calibration updates will be made if there is an incident. For example, if a secure Over-the-Air (OTA) mechanism is available, that method could be used to make authorized modifications of calibrations and/or software.

5.7 Cybersecurity Considerations When the Vehicle Owner Changes

The vehicle owner changes when a vehicle owner sells their vehicle, when a vehicle is totaled in an accident and goes to a salvage yard, when a dealer's demo vehicle needs to be prepared for sale to a customer, etc. To plan for when the vehicle owner changes:

- Determine whether there are any systems on the vehicle that have software or Customer Personal Information that might need to be erased to protect the customer and/or to protect the organization (e.g., immobilizer, cell phone pairing).
- Provide a method to remove personal information from vehicle systems when change of ownership and/or end of vehicle life occurs. This should be described in the Vehicle Owner's/Operator's Manual or in instructions for vehicle service providers. See NIST Special Publication 800-88, "Guidelines for Media Sanitization" (6), for information on methods for cleansing the storage media.

6. CYBERSECURITY PROCESS OVERVIEW

6.1 Motivation for a Well-Defined and Well-Structured Process

As with system safety, Cybersecurity should be built in to the system rather than added on at the end of development. Attempting to add Cybersecurity on to an existing system, or using an ad-hoc approach to identify and implement Cybersecurity Controls can lead to:

- Unneeded Cybersecurity Controls that require valuable limited resources (cost and engineers) to identify, develop, and implement,
- Incorrect Cybersecurity Controls,
- Incomplete or inconsistent Cybersecurity Controls,
- Unintentional insertion of additional and unknown vulnerabilities.

No system can be guaranteed to be 100% secure. However, following a well-defined and well-structured process provides a means to help reduce the likelihood of a successful attack. A ***well-defined and well-structured (WDWS)*** process establishes a repeatable, structured method to systematically identify threats, vulnerabilities that could be exploited to achieve a threat, and appropriate Cybersecurity Controls to design in to the system during development to protect against the identified vulnerabilities. A WDWS process provides guidance throughout the entire life-cycle, from concept phase through production, operation, and service.

Reducing the likelihood of a successful attack can be likened to reducing the likelihood of potential hazards. To reduce the likelihood of potential hazards from occurring, the vehicle industry applies the principles of system safety engineering in the design and development of ***safety-critical*** systems. In a similar way, to reduce the likelihood of successful ***cyber-attacks*** on vehicles, the vehicle industry may apply principles of system Cybersecurity engineering to the design and development of Cybersecurity-critical cyber-physical vehicle systems.

6.2 Process Framework

Figure 3 shows an overall Cybersecurity engineering process framework for cyber-physical vehicle systems that considers the entire lifecycle from concept phase through production, operation, and service. The lifecycle shown in the figure is analogous to the process framework from ISO 26262 Functional Safety Road Vehicles standard (1). This analogous lifecycle is chosen to allow organizations with safety processes based on ISO 26262 to use a common framework between Cybersecurity and safety to facilitate:

- Development of a tailored Cybersecurity process by capitalizing on aspects of an organization's existing safety process that are common to both Cybersecurity and safety, for example, the supporting process procedures and templates,
- Maintaining consistency between Cybersecurity and safety given the interrelationships between the two areas.

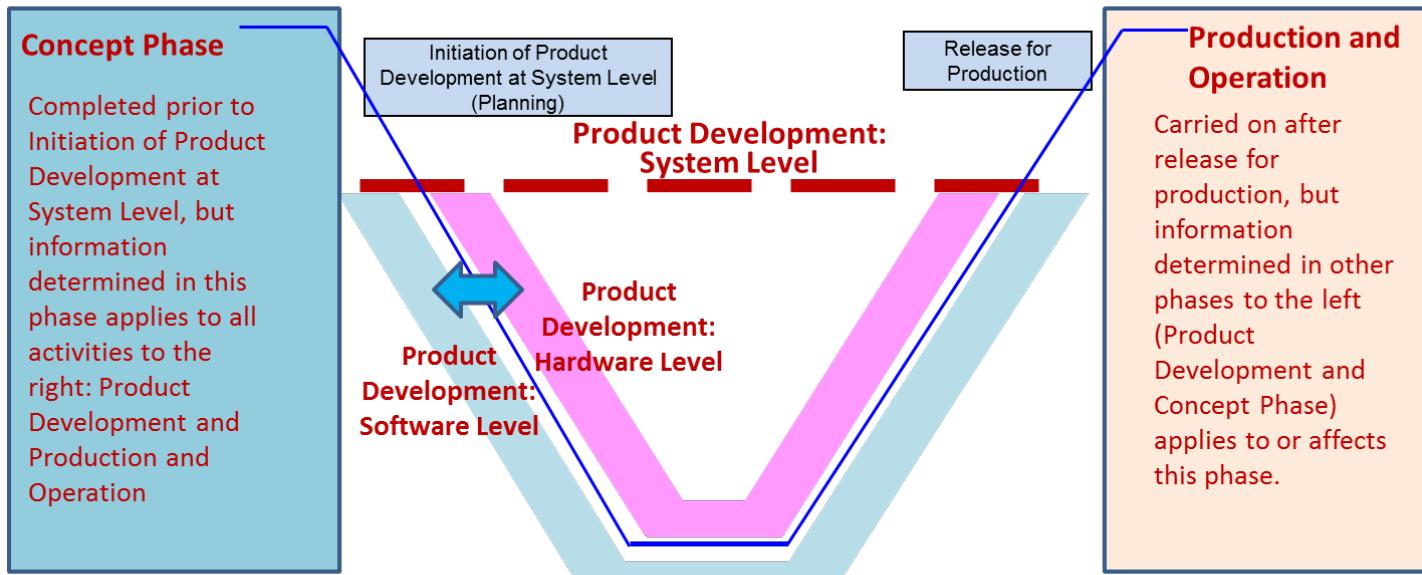
The process framework consists of the management of Cybersecurity, the core Cybersecurity engineering activities, and the supporting processes. The core Cybersecurity engineering activities include concept phase activities, activities for product development at the system, hardware, and software levels, and production, operation, and maintenance activities. Supporting process activities include activities that are applicable across different life-cycle phases, such as configuration management, change management, etc.

6.2.1 Overall Management of Cybersecurity

Management of Cybersecurity consists of two aspects: 1.) the overall management of Cybersecurity; and 2.) management of Cybersecurity activities within specific stages of the development lifecycle. Part of the overall management of Cybersecurity includes:

- Creating, fostering, and sustaining a Cybersecurity culture that supports and encourages effective achievement of Cybersecurity within the organization,
- Establishing methods to help ensure compliance to an adopted Cybersecurity engineering process,
- Identifying and establishing needed communication channels with respect to Cybersecurity, both internally and externally,
- Development and implementation of training and mentoring to achieve a competence in Cybersecurity for cyber-physical vehicle systems,
- Incorporating an expanded field monitoring process that includes monitoring **hacker chatter** (including online and at conferences where potential attacks/vulnerability conversations may occur), reporting unsuccessful attacks, etc.,
- Incorporating an incident response process is important and should include an attack incident reporting procedure, and attack incident investigation, resolution, and action procedures.

Management of Cybersecurity



Supporting Processes

Figure 3 - Overall Cybersecurity process framework

In the concept phase, management of Cybersecurity activities may include assigning a Cybersecurity manager to oversee the Cybersecurity activities, and to be responsible for planning and overseeing the Cybersecurity activities, including developing a **Cybersecurity Program Plan**.

During product development, management of Cybersecurity activities may include:

- Beginning a preliminary **Cybersecurity assessment** that will be refined throughout the development process and reviewed at major milestones and will culminate into the final Cybersecurity assessment (**Cybersecurity case**),
- Identifying and overseeing reviews to confirm that the appropriate activities are performed correctly.

Any open issues with respect to Cybersecurity would be recorded and appropriate follow-up action stated. If open Cybersecurity issues were contained in a previous Cybersecurity assessment, these should be addressed in the updated assessment. The final Cybersecurity issue assessment is the Cybersecurity Case. In the Cybersecurity Case, any open Cybersecurity issues would be resolved, or a rationale would be included stating why the open issue is acceptable, and the final arguments and evidence to support the claims would be provided.

6.2.2 Concept Phase

Figure 4 shows the flow of activities during the concept phase. The **feature** definition activity is to define the feature being developed, including identifying the boundaries and the Cybersecurity perimeter, and identifying the external dependencies and assets. Defining the feature clarifies the boundaries and scope for the future analysis activities. A well-defined scope helps to bound future analysis activities so the analyses can be completed more efficiently and effectively.

The initiation of the Cybersecurity lifecycle includes development of the Cybersecurity Program Plan that describes the activities to be carried out as part of the Cybersecurity lifecycle. The **Threat Analysis and Risk Assessment (TARA)** activity is used to identify and assess the potential threats to the system and to determine the risk associated with each identified threat. The TARA results will drive future analysis activities by focusing future analyses on the highest risk Cybersecurity threats.

Cybersecurity goals are determined for the highest risk potential threats. At the highest level, Cybersecurity goals may be the inverse of the potential threat; for example, if a potential threat is malicious braking, the highest level Cybersecurity goal may be to prevent or reduce the likelihood of malicious braking from occurring, or mitigate the potential consequences of malicious braking. Once the Cybersecurity goals are determined for the highest level threats, a **Cybersecurity Concept** can be developed to describe the high-level Cybersecurity strategy for the feature.

From the Cybersecurity Concept and the Cybersecurity goals, the high-level Cybersecurity requirements can be identified and derived. These high-level Cybersecurity requirements can then be refined further in the product development stages. At the end of the concept phase a preliminary Cybersecurity Assessment may be performed to assess the state of Cybersecurity that is proposed for the feature.

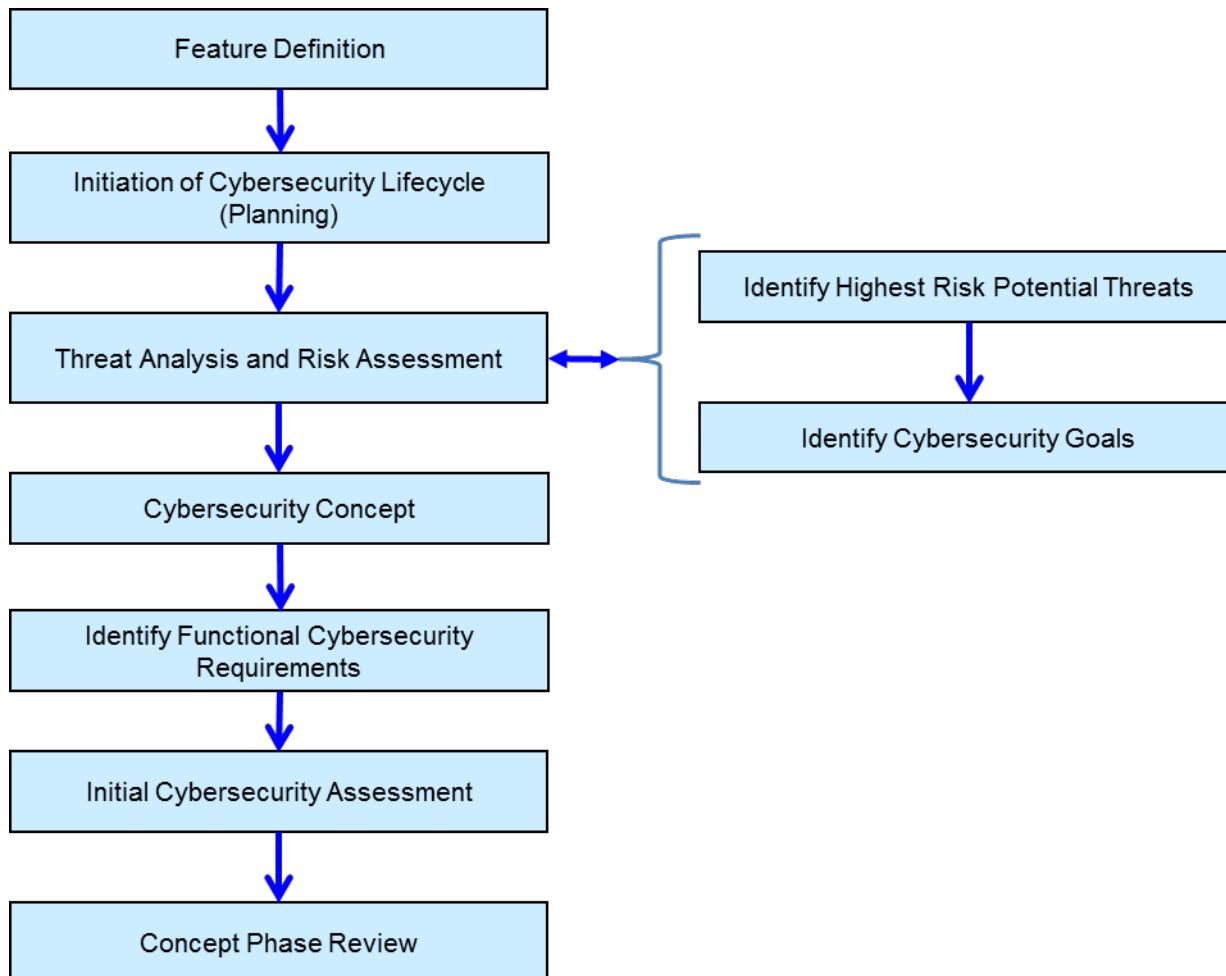


Figure 4 - Concept phase activities

6.2.3 Product Development

The Product Development phase of the lifecycle consists of product development at the system level, product development at the hardware level and product development at the software level.

Figure 5 shows an overview of the product development phase, and the relationships between product development at the system-level design phase and product development at the hardware and software levels.

NOTE: Iteration occurs throughout many phases of the lifecycle; however, these iterations are not depicted to avoid over-complicating the diagrams.

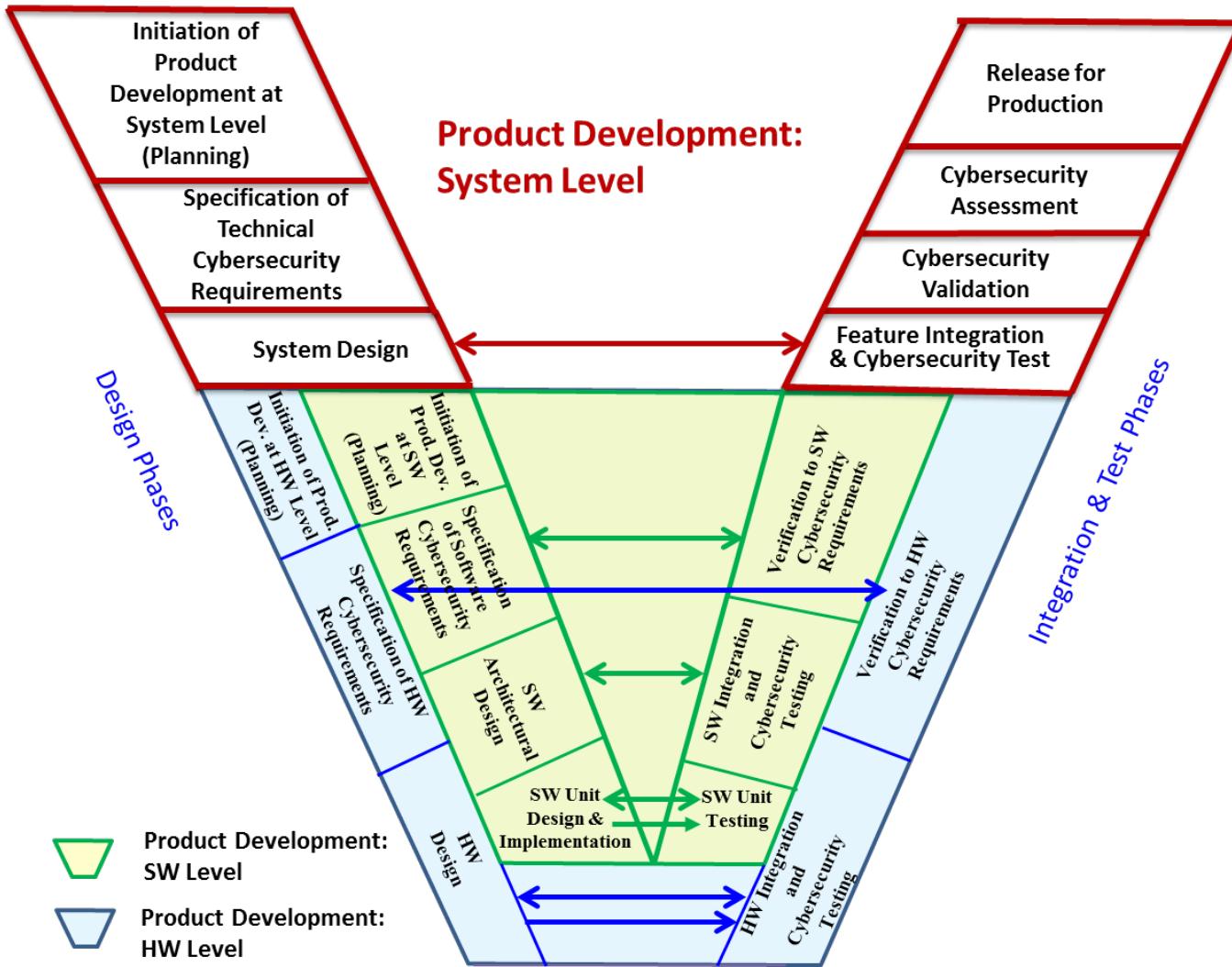


Figure 5 - Relationships between product development at the system, hardware, and software levels

6.2.3.1 Product Development: System Level

Figure 6 shows a V diagram for product development at the system level. During product development at the system level, the Cybersecurity Concept is refined into a **Technical Cybersecurity Concept** (i.e., the high-level Cybersecurity strategy described in the Cybersecurity Concept is refined into engineering terms). To help refine the Cybersecurity Concept into the Technical Cybersecurity Concept, a system level threat analysis or vulnerability analysis may be performed if there is significant new information available. Technical Cybersecurity requirements are then derived and refined from the high-level Cybersecurity requirements and the technical Cybersecurity strategy.

A **System Context** (hardware/software interface document) may be created to define the interfaces between the system's hardware and software, the key data flows, and storage and processing within the system. Using the System Context, the system-level technical Cybersecurity requirements are then allocated to hardware and software or to both. Once the technical Cybersecurity requirements have been allocated to hardware and/or software, the activities at the Product Development: Hardware Level (6.2.3.2) and Product Development: Software Level (6.2.3.3) can begin (see Figure 6).

Upon completion of the product development activities at the hardware and software level, hardware and software integration and testing is performed (Figure 6 and Figure 7). Vulnerability and Penetration testing may be performed on the integrated system, and verification/validation of the Cybersecurity technical requirements is performed. A final Cybersecurity assessment is performed resulting in the final Cybersecurity case. A final Cybersecurity review can then be completed prior to release for production.

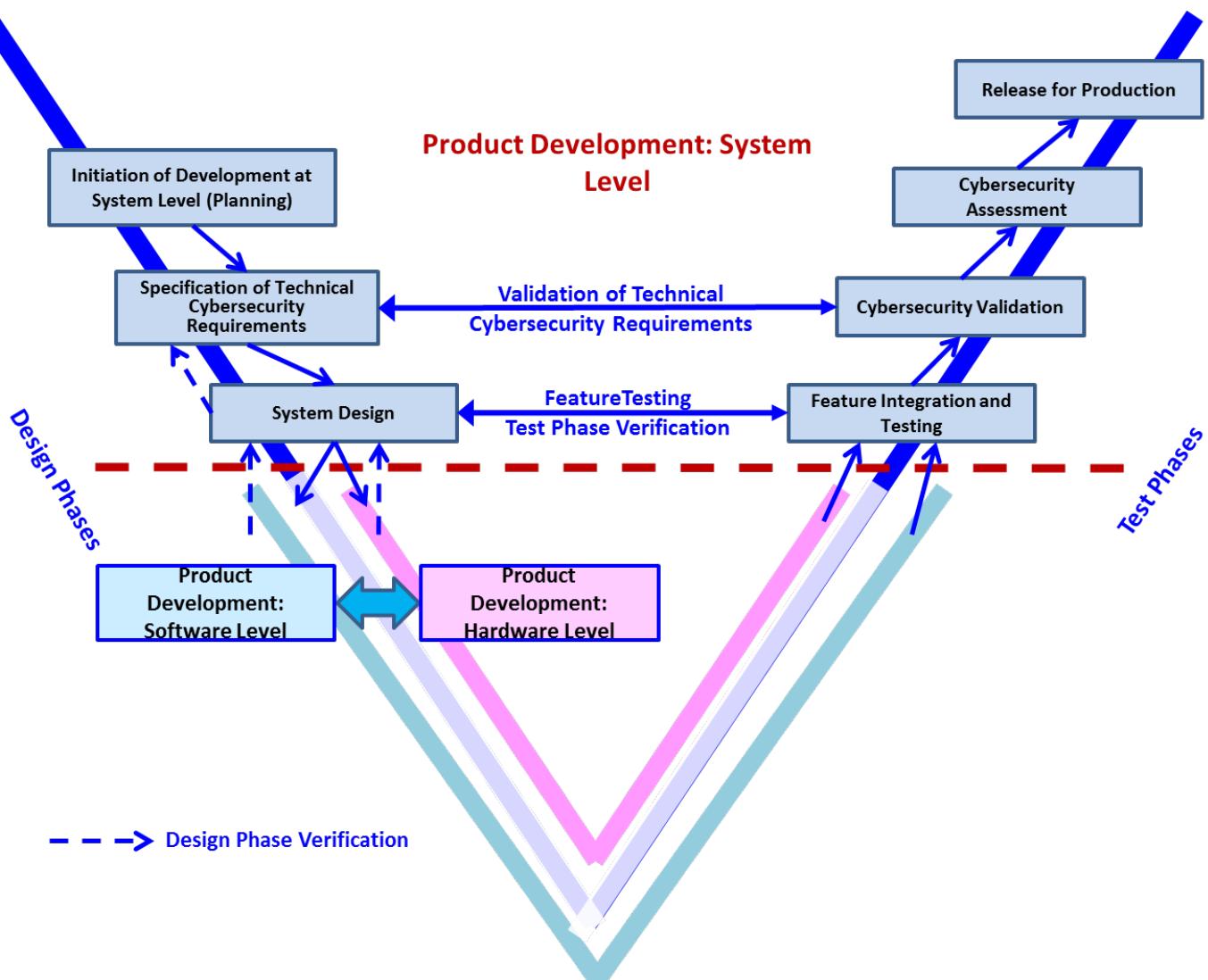


Figure 6 - V diagram for product development at the system level

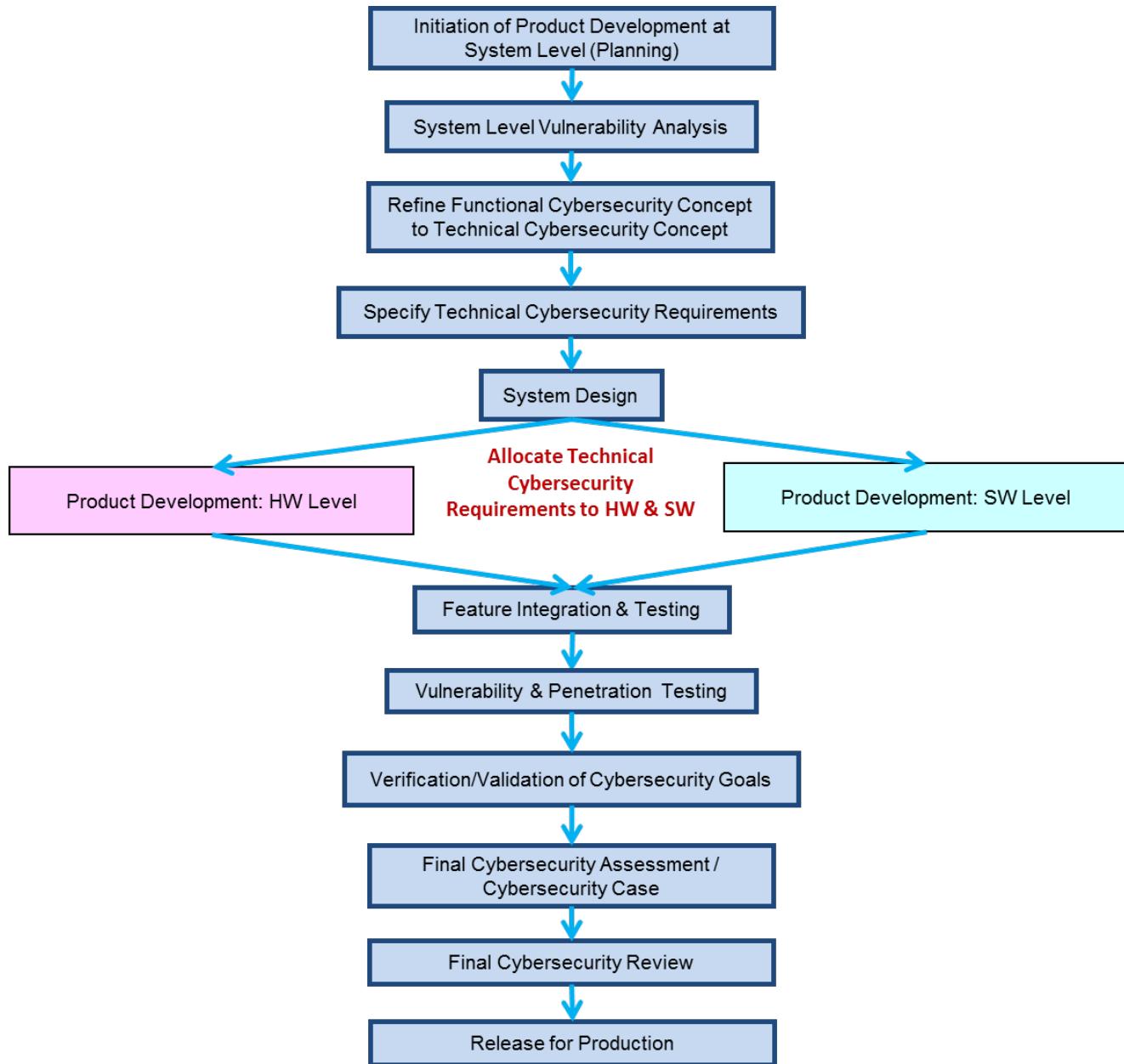


Figure 7 - Product development: system level

6.2.3.2 Product Development: Hardware Level

Figure 8 shows a V diagram for product development at the HW level in relation to product development at the system level. Figure 9 shows the flow of activities for product development at the hardware level. Hardware Cybersecurity requirements would be specified from the Cybersecurity requirements allocated to hardware during the system level development. If applicable, the Technical Cybersecurity Concept could be refined at this stage. Following hardware design, a vulnerability analysis would be performed to help identify potential vulnerabilities in the design and to help identify potential Cybersecurity Controls to address the potential vulnerabilities. Following hardware integration and testing, vulnerability and penetration testing may be applied to the hardware design. A Cybersecurity assessment is then performed and the preliminary Cybersecurity assessment is refined.

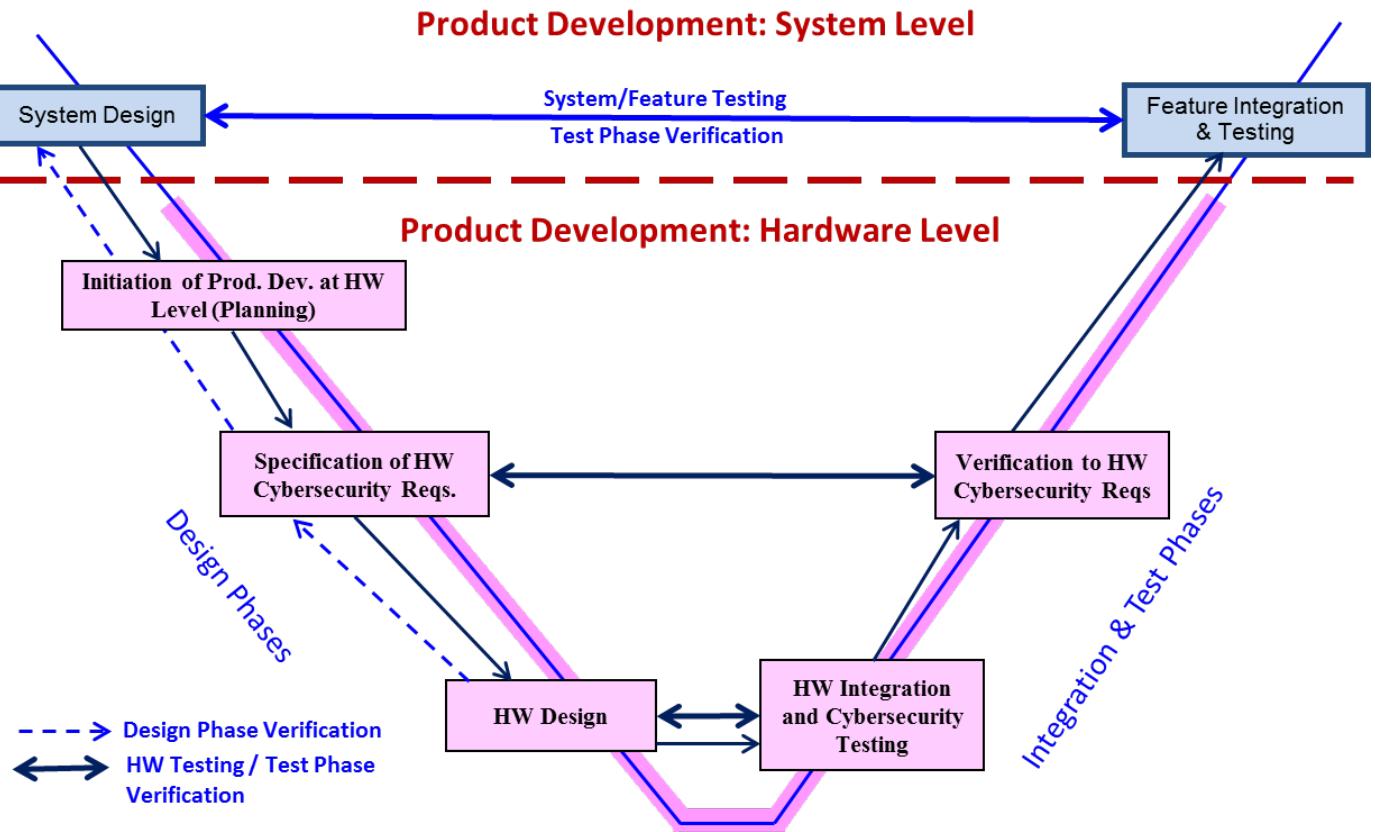


Figure 8 - V diagram showing product development at the HW level and its relationship to product development at the system level

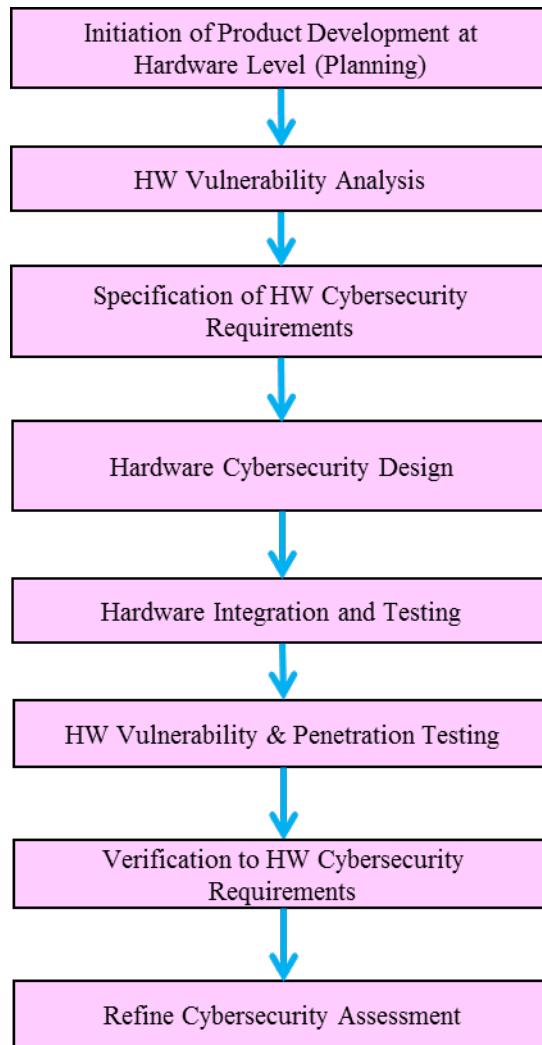


Figure 9 - Product development: hardware level

6.2.3.3 Product Development: Software Level

Figure 10 shows a V diagram for product development at the software level in relation to product development at the system level. Figure 11 shows the flow of activities for product development at the software level. Software Cybersecurity requirements would be specified from the Cybersecurity requirements allocated to software during the system level development. If applicable, the Technical Cybersecurity Concept could be refined at this stage. Following software architectural design, a vulnerability analysis may be performed to help identify potential vulnerabilities in the software architectural design and to help identify potential Cybersecurity Controls to address the potential vulnerabilities. Following software unit design and implementation, a software level vulnerability analysis may be performed, followed by software unit testing and software integration and testing. The software Cybersecurity requirements are verified after software integration, and vulnerability and penetration testing may be performed on the software. A Cybersecurity assessment is then performed and the previous Cybersecurity assessment is refined.

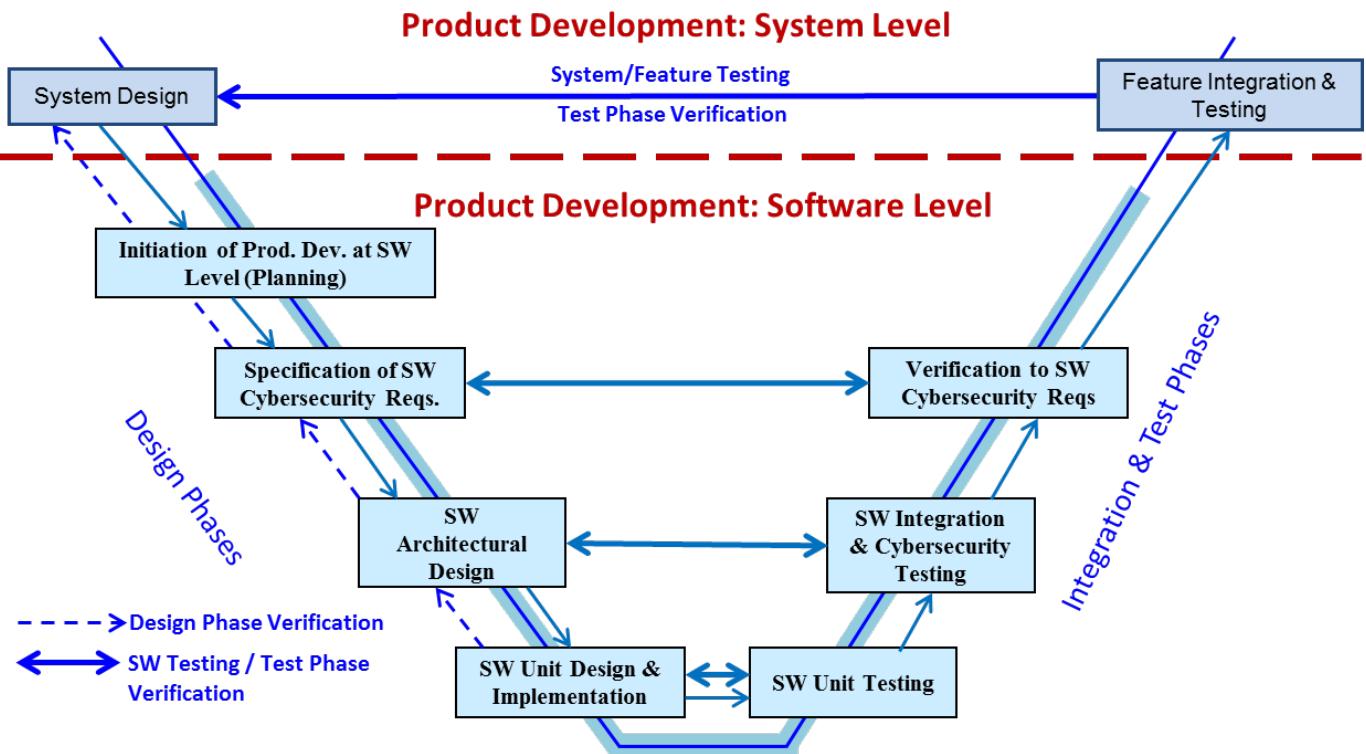


Figure 10 - V diagram for product development at the software level in relation to product development at the system level

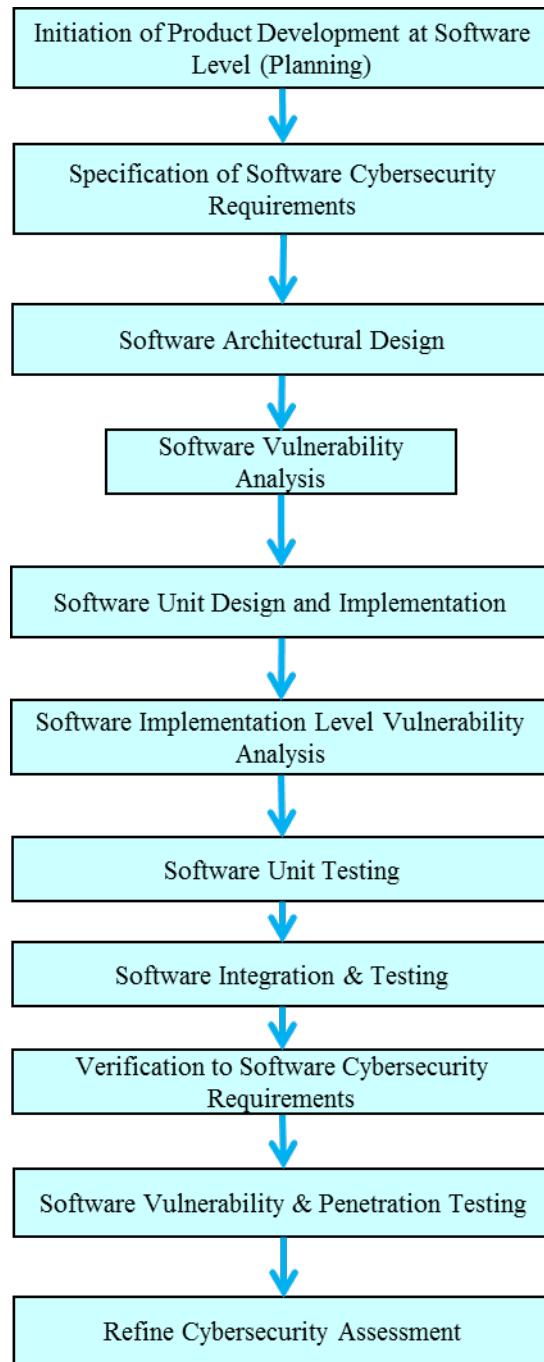


Figure 11 - Product development: software level

6.2.4 Production, Operation & Service

Activities in the production phase include production planning with respect to any Cybersecurity-related requirements that may impact the production process, including requirements relative to having specific portions of the manufacturing process secure. Cybersecurity-related production requirements may be included in the existing production plan. Cybersecurity requirements for the system may affect the specific process by which software will be flashed in the manufacturing facility onto ECU's, for example by requiring that the tools used for flashing be secure.

The operation phase includes both operation and service. Service includes normal maintenance activities and repair. Any requirements specific to Cybersecurity during operation, should be recorded within the appropriate documents (e.g., Vehicle Owner's Operating Manual). With respect to service, any maintenance and repair activities that have the potential to adversely affect Cybersecurity should have been identified in earlier lifecycle phases, and appropriate procedures should have been specified on how to maintain Cybersecurity during maintenance and repair; for example, maintenance procedures, and repair procedures. Service that could affect Cybersecurity includes re-flashing ECU's, connecting to the on-board diagnostics port, telematics system updates, vehicle/cloud computing interfaces, etc.

The operation phase also includes performing and maintaining the field monitoring process that was defined and established in the overall management of Cybersecurity activities, and fulfilling the incident response procedure that was also defined and established in the overall management of Cybersecurity activities.

6.2.5 Supporting Processes

Some of the supporting process activities should be identical to activities applied as part of a system safety engineering process tailored to be consistent with ISO 26262. It is recommended that these processes be integrated into a company's existing product development process. These include: configuration management, documentation management, change management, etc. Other supporting process activities used in ISO 26262 may be tailored to be specific to Cybersecurity. These include: management of Cybersecurity requirements, requirements for dealing with distributed development, etc. The distributed development requirements are designed to help ensure the following:

- That a supplier is capable of developing and producing Cybersecurity-critical features according to a customer organizations internal Cybersecurity process,
- That the appropriate communication channels are established and maintained between the supplier and customer,
- That the Cybersecurity work products are agreed to,
- That appropriate reviews are established at key milestones with customer access to work products,
- That any changes that could affect Cybersecurity are evaluated and agreed to,
- That the final Cybersecurity case is reviewed and agreed to,
- That any Cybersecurity issues that the supplier may become aware of are reported to the customer in a timely manner, etc.

6.3 Milestone and Gate Reviews

Figure 12 shows the gate reviews that may be performed at each major milestone. These include the Concept Phase Review at the completion of the concept phase activities, the Requirements Review that may occur in stages (for example, functional requirements review and technical requirements review, including review of requirements allocated to hardware and software at the system level), the Design Review, Implementation Review, and Verification and Validation Review.

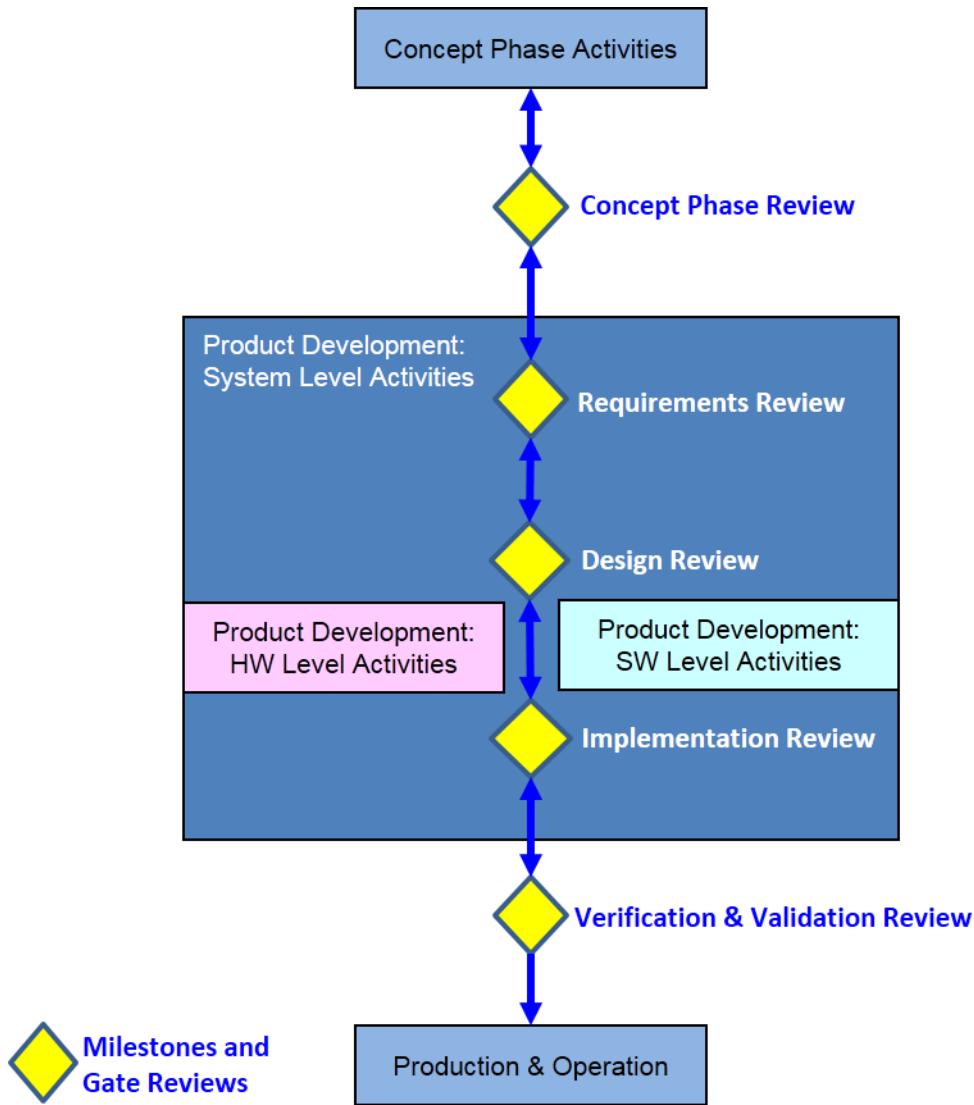


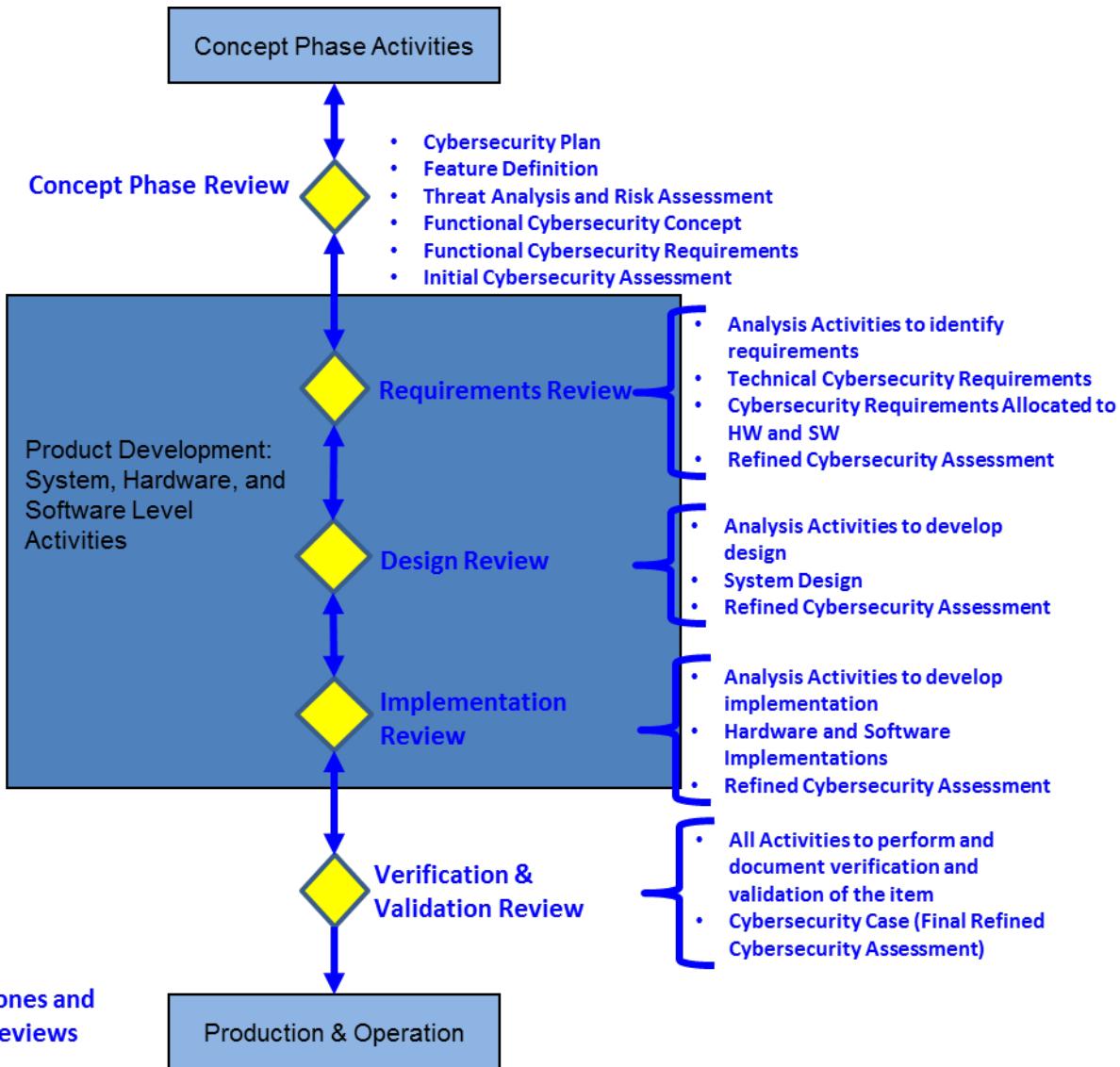
Figure 12 - Possible milestones and gate reviews

Figure 13 shows the tasks that may be reviewed at each of the major milestones. In the Concept Phase Gate Review, the review may include review of the Cybersecurity plan, the system definition, the threat analysis and risk assessment results (including the identified threats, threat classifications, and the Cybersecurity goals), the Functional Cybersecurity Concept, the refinement of the Cybersecurity goals into functional Cybersecurity requirements, and the Preliminary Cybersecurity Assessment. The Requirements Review may cross lifecycle phases and includes review of the functional Cybersecurity requirements, the technical Cybersecurity requirements refined or derived from the functional Cybersecurity requirements, and the technical Cybersecurity requirements allocated to hardware and software. It also includes reviews of analysis activities that were performed in identifying the requirements. The Design Review includes the analysis activities and results that affect the design, and the system design. The Implementation Review includes the analysis activities and results that affect the implementation, and the implementations at the hardware and software levels. The verification and validation review helps ensure that the system was designed and developed according to the requirements, and that the Cybersecurity Controls are appropriate and work as intended; note that in some cases, it may not be feasible or possible to verify and validate all of the Cybersecurity Controls. Common criteria methods may be considered to ensure comprehensive testing is conducted (7).

The gate reviews are intended to help ensure that appropriate activities have been performed and completed correctly and consistently before the next stage of development begins. These reviews may be conducted by a small (e.g., 3-4 person) team of technical experts that should ideally be independent of the feature development. To maintain consistency and completeness across the feature development, it is recommended that this same 3-4 person team participates in all of the reviews throughout the system development. The results of each review may be a pass, or a conditional pass (rework required). A gate review should be completed successfully prior to exiting the gate and moving on to the next phase.

There are two possible ways an organization may choose to complete the technical reviews. One is to use a technical “gate” review concept as described above, where the review is considered a gate to the next phase and is performed at the end of the development phase being completed, and the second approach is to perform a review of the activities completed during each development phase as each activity is completed. One advantage of the first approach (the gate review approach) is that the reviewers can follow the progression of definitions, descriptions, analyses, etc., for one review and this may make it easier to identify potential incomplete, inconsistent, or incorrect aspects across the activities when completing the reviews at the same time. A second potential advantage is that the technical experts are only required for a single meeting as opposed to multiple review meetings. Potential disadvantages of this approach are that issues may not be caught immediately and may propagate to the next activity, a greater amount of time is required during a given period to review multiple documents and the single meeting will require more time since more activities are being reviewed at one time. Providing the results of the activities well ahead of the review (e.g., at least two weeks ahead) may help to alleviate the first potential disadvantage by providing the reviewers with a period of time to review the results.

Potential advantages of the second approach, reviewing the results as each activity is completed, include that issues may be caught sooner and repaired before they propagate, less time is required to prepare for the review by reviewing a single documented result than would be required to review multiple results, and the review meetings for single documents will also be shorter. The primary disadvantage of this approach is that the continuity between results may be lost when the results are reviewed individually as opposed to collectively. As a result, in order to be thorough, the reviewers would be required to re-review previous results in order to check for completeness, consistency, and correctness across the results. This eliminates the advantage of less preparation time. It is up to an organization to determine the best approach to follow with respect to reviews. It may be beneficial as an organization is coming up to speed with Cybersecurity and a Cybersecurity process, to start out using the second approach and transition to the first approach as experience is gained in Cybersecurity.

**Figure 13 - Gate review activities**

7. OVERALL MANAGEMENT OF CYBERSECURITY

As presented in Chapter 4, Cybersecurity Process Overview, part of the overall management of Cybersecurity includes:

- Creating, fostering, and sustaining a Cybersecurity culture that supports and encourages effective achievement of Cybersecurity within the organization,
- Establishing methods to help ensure compliance to an adopted Cybersecurity engineering process,
- Identifying and establishing needed communication channels with respect to Cybersecurity, both internally and externally,
- Development and implementation of training and mentoring to achieve a competence in Cybersecurity for cyber-physical vehicle systems,

- Incorporating a field monitoring process that includes monitoring hacker chatter (including online and at conferences where potential attacks/vulnerability conversations may occur), media articles, reporting unsuccessful attacks, etc.,
- Incorporating an incident response process that includes an attack incident reporting procedure, and attack incident investigation, resolution, and action procedures.

This chapter will describe these concepts in more detail.

7.1 Cybersecurity Culture

A Cybersecurity culture within an organization is an organizational culture that:

- Places high priority on Cybersecurity.
- Sets forth the organizational goal to stay ahead of potential threats.
- Makes addressing highest risk threats a priority.
- Works to create, foster, and sustain a Cybersecurity culture within the organization by:
 - Building a dedicated organizational structure to:
 - Develop and follow global Cybersecurity processes, practices, tools, and methods,
 - Train employees in the proper way to design and think about Cybersecurity,
 - Inform management on attack motivations and the potential detrimental effects to an organization if an attack is successful,
 - Develop a business case for Cybersecurity vulnerability resolution and product introduction (e.g., defined threats, rationale, benchmarking [if applicable], company impact, pros/cons, implementation implications [which groups/products are affected], and timing) and to
 - Champion Cybersecurity topics within the product development organization.

7.2 Measuring Conformance to a Cybersecurity Process

The Cybersecurity process should be followed throughout the product development lifecycle. With this process, threats that have been identified as having higher risk may require additional or more rigorous analysis. This process is described below.

The goal is a holistic process whereby an organization monitors “daily events” and warns and notifies those with a need-to-know, determines threats of greatest risk, analyzes data, plans upgrades, maintains systems, ensures compliance to standards and regulations, responds to Cybersecurity issues, and is always monitoring new technologies.

Based on level of risk, there will be more or less rigor for process implementation. Some activities may include:

- Ensuring conformance to a company’s internal Cybersecurity process and requirements.
- Holding technical reviews at each of the development milestones.
- Performing audits to ensure process is followed.
- Submitting performance and verification/validation plans.

7.3 Identifying and Establishing Communication Channels

The organization structure should be clearly identified so appropriate communication channels are used. The organization should know how to communicate:

- New media articles and events to those appropriate groups and organizations.
- Processes for an individual (both internal and external) to report incidents.
- Processes for companies to report incidents as appropriate to their suppliers and/or vehicle manufacturer.
- Processes (and dedicated groups) to address and respond to government, media, public, and internal inquiries.
- Internal organization structure within the company so employees know who to go to about any type of Cybersecurity question or issue.

7.4 Developing and Implementing Training and Mentoring

Empowering the employees in your organization to recognize common threats can be beneficial to the organizations Cybersecurity health. Cybersecurity awareness and training teaches employees to understand vulnerabilities and threats to business operations and products.

- Employees should realize that all products and processes have vulnerabilities.
- A regimented threat analysis and risk assessment should be performed to direct where resources should be spent. This regimented threat analysis and risk assessment should occur early in the development of new features/products because it is more expensive to fix any issues late in the product lifecycle.
- Mentoring should occur from an experienced Cybersecurity engineer.
- Share Cybersecurity best practices and organizational insights on what really works.
- Encourage an open exchange of ideas and questions in a non-threatening environment.
- Continually emphasize the critical nature of data and product Cybersecurity and that it's the employee's responsibility to be aware and build this in to their designs.
- Incorporate Cybersecurity awareness into existing processes and forums, as appropriate.
- Provide ongoing training on what the Cybersecurity process is and how it should be implemented.
- Provide ongoing technical Cybersecurity training sessions to improve individual engineering competency.
- Inform employees of Cybersecurity design and evaluation tools as well as make them aware of industry standards and best practices that are available.
- Fold in learning of vulnerabilities from previous products into new and future products.

7.5 Operation and Maintenance Activities

7.5.1 Incident Response Process

An organized method should be planned to handle any Cybersecurity incident(s) affecting a vehicle, vehicle fleet, or vehicle manufacturer/supplier infrastructure. A team should be formed that will initially review a reported incident to determine potential impact and accuracy. If the reported incident is deemed to be accurate and has a high-level of importance, the team should investigate the issue to determine more detail, bring the correct teams together to determine overall impact and finally determine changes necessary, if any, to mitigate/eliminate the issue. Incident response should be monitored for timeliness and appropriate closure. Detailed logging of the event, capture/collection of forensics data and the prevention measures taken should be generated.

7.5.2 Field Monitoring Process

A plan and method should be available for potential communication paths for reporting a Cybersecurity incident. Field monitoring is needed once the system, vehicle, vehicle fleet and/or vehicle manufacturer/supplier infrastructure is available to the public. Notification may come from customers, law enforcement, insurance companies, media, suppliers, etc. There should be clear and easy instructions on how to go about reporting an incident to the vehicle manufacturer, should one occur. A team should be in place to retrieve these notifications.

The goal is a holistic process whereby an organization monitors “daily events” and warns and notifies those with a need-to-know, determines threats of greatest risk, analyzes data, plans upgrades, maintains systems, ensures compliance to standards and regulations, responds to Cybersecurity issues, and is always monitoring new technologies.

NOTE: In 2015, an **Automotive Information Sharing and Analysis Center (ISAC)** was initiated. The scope is related to light duty on-road passenger vehicle electronics and associated networks. This can be another valuable source for quick information about Cybersecurity incidents that are related to the Automotive Industry.

8. PROCESS IMPLEMENTATION

The Cybersecurity process described in this document may be applied separately from a system safety engineering process with integrated communication points between the two processes, or the Cybersecurity and system safety process may be integrated more tightly. It is left to the organization to decide how best to implement and apply both processes. Some advantages and disadvantages of both types of implementation are described in this section. Even a third hybrid approach could be taken where there are some shared processes and steps with safety and some that are unique to Cybersecurity only. Regardless of which implementation is chosen by an organization, if a Cybersecurity process is tailored from an organization's existing safety process and the processes are analogous to each other (share a common framework), then the Cybersecurity process can be developed by leveraging work that has already been done in the safety process development. For example, the supporting processes developed for a safety process can be easily tailored and adapted to a Cybersecurity process. Likewise, templates developed for a safety process can be tailored for the Cybersecurity process; for example, the safety plan from a safety process can be easily tailored to a Cybersecurity plan for a Cybersecurity process. In addition, a field monitoring and incident response process developed for safety could be revised as needed to work for Cybersecurity. There are many other possible activities and templates from a safety process that can be tailored and adapted into a Cybersecurity process to facilitate creation, implementation, and application of a Cybersecurity process within an organization that has an existing safety process.

8.1 Applying a Cybersecurity Process Separately with Integrated Communication Points to a Safety Process

The Cybersecurity process may be developed, implemented, and maintained separately from an organization's safety process. Some advantages to this approach are that though the two domains (Cybersecurity and safety) have analogous activities from a process perspective, the activities are different and can impact different domains (e.g., Cybersecurity often impacts infotainment, which is typically a domain not impacted by safety), and require different types of expertise. Given that developing systems in each domain according to a well-structured and well-defined process may be resource intensive, it may be advantageous to keep the two activities separate with separate technical experts working according to their respective process. A possible disadvantage to this approach may be increased resource requirements since separate resources would be required for each domain. However, since all safety-critical systems are Cybersecurity-critical, and Cybersecurity vulnerabilities may lead to a violation of a safety goal, it is necessary to maintain consistency and completeness between Cybersecurity and safety.

Figure 14 through Figure 17 show both the activities of each phase of the product development lifecycle for a well-structured and well-defined Cybersecurity process, and the possible communication links between the Cybersecurity activities and safety activities in an analogous safety process. Another important aspect is that, in addition to the communication between the teams as shown in the figures, it is essential that synchronization of both teams to the design process occur. For instance, in both cases, all requirements should be resolved before the start of design verification so that requirements-based testing can proceed.

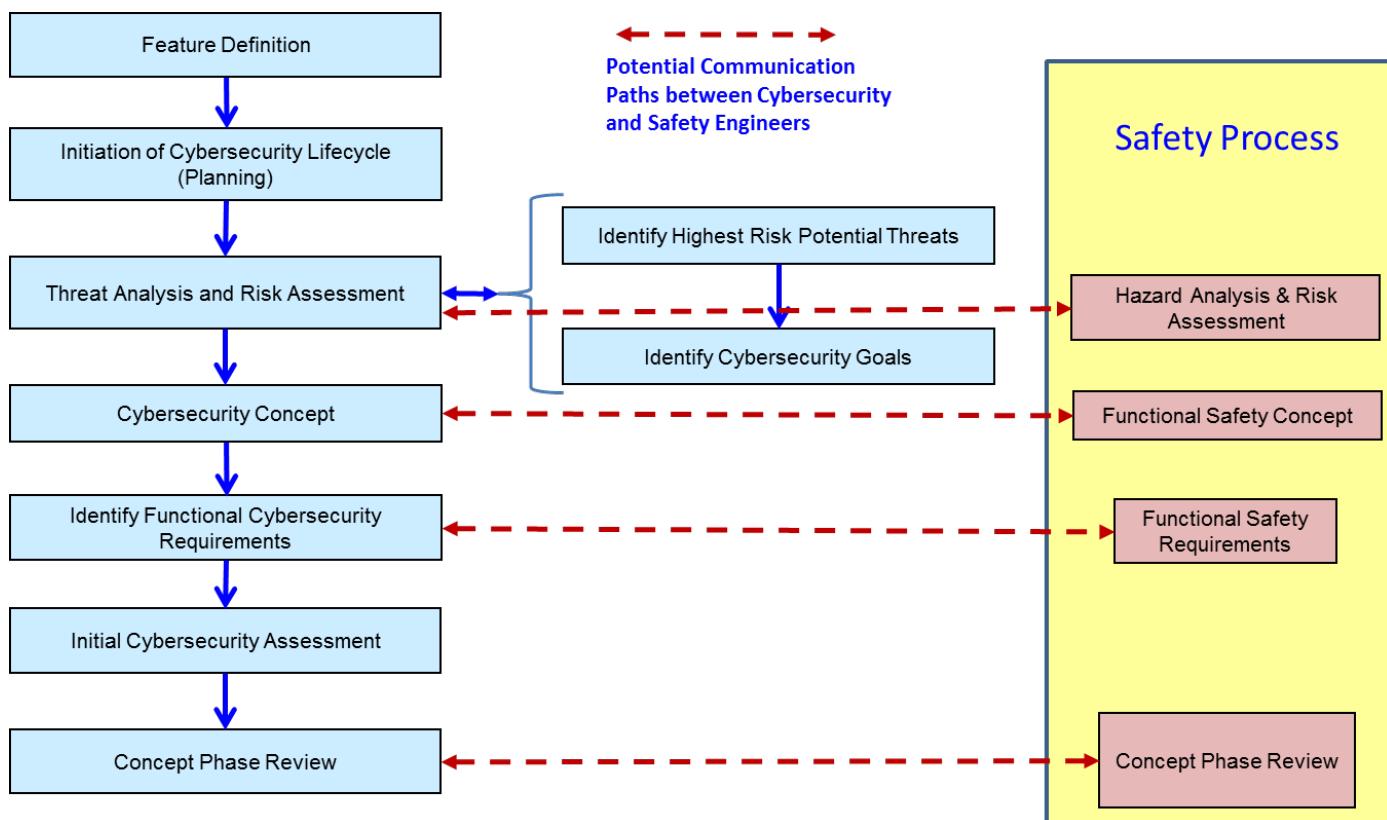


Figure 14 - Concept phase activities with potential communications paths between Cybersecurity and safety activities

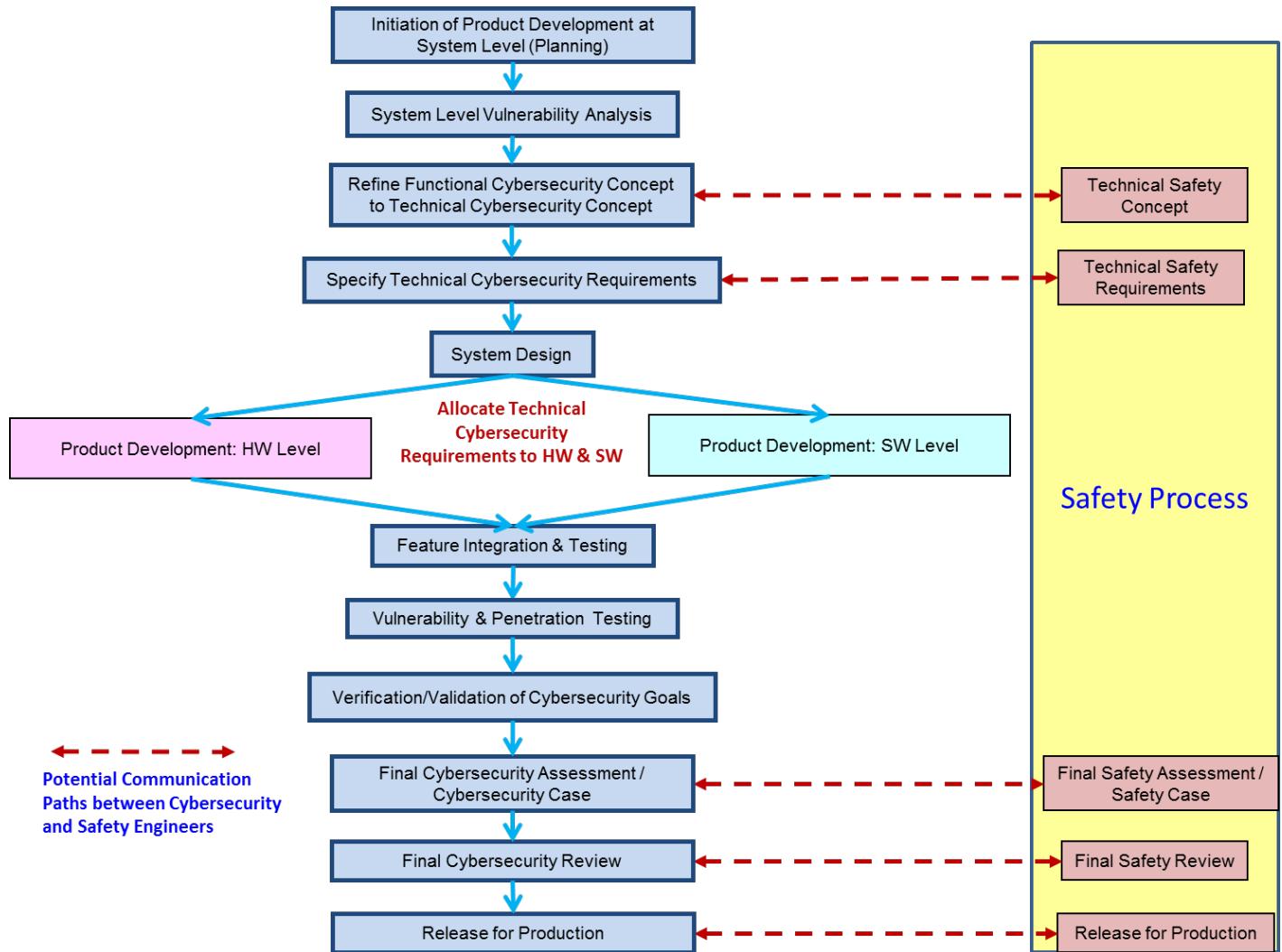


Figure 15 - Product development at the system level activities with potential communications paths between Cybersecurity and safety activities

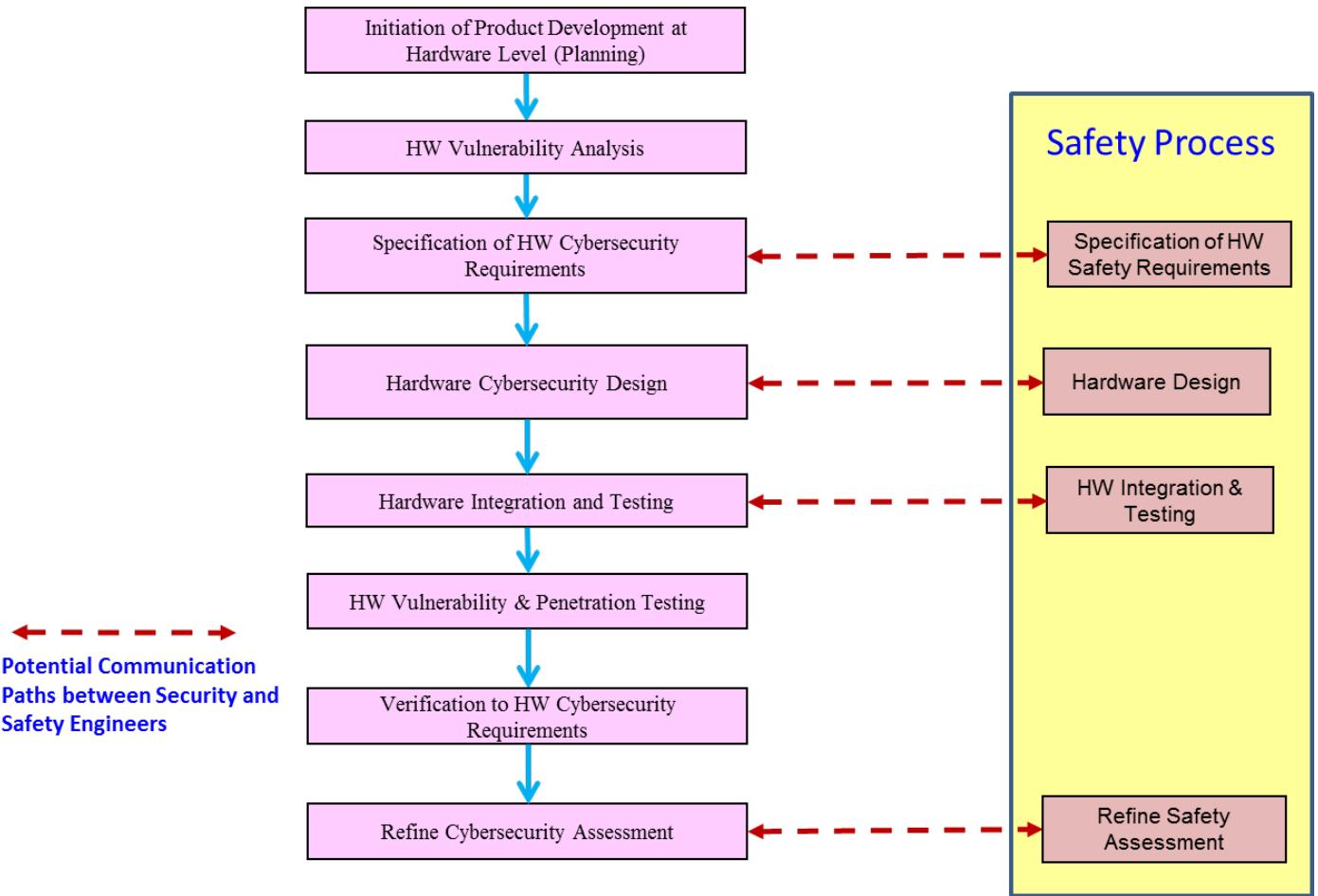


Figure 16 - Product development at the hardware level activities with potential communications paths between Cybersecurity and safety activities

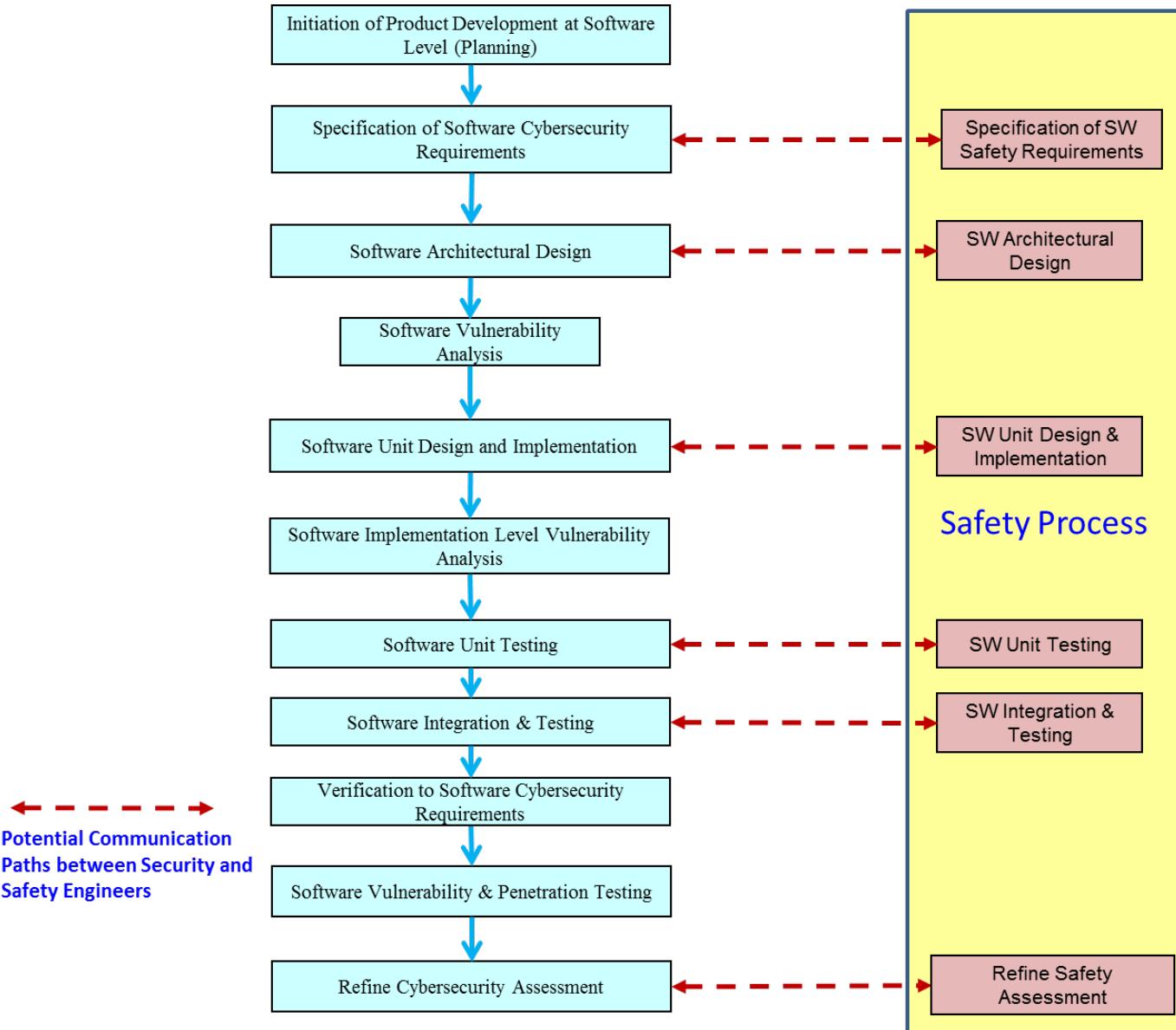


Figure 17 - Product development at the software level activities with potential communications paths between Cybersecurity and safety activities

8.2 Applying a Cybersecurity Process in Conjunction with a Safety Process Tailored after ISO 26262

This type of application is a tightly integrated Cybersecurity and safety process. Since the Cybersecurity process described in this recommended practice is based on the ISO 26262 process framework, tightly integrating the two processes would simply mean to include the Cybersecurity activities described in this document for each product lifecycle phase, with the corresponding activities for each product lifecycle phase described in the safety process. The integration of these activities may be done by keeping the Cybersecurity and safety activities separate, but performing these activities in conjunction with each other and with the same team, or parallel activities may be done by developing an integrated technique that covers both safety and Cybersecurity at the same time. An example of this is to develop a technique to perform both a hazard analysis and risk assessment, and a threat analysis and risk assessment at the same time using a single integrated template and method. A tightly integrated process for Cybersecurity and safety has the advantage of a common resource set, thus, requiring fewer additional resources. However, since both activities require different technical expertise and both activities are resource intensive, it may not be feasible to expect a single team of experts to have the skills to perform both Cybersecurity and safety tasks simultaneously.

8.3 Concept Phase

Figure 4 shows the concept phase activities. Each of the activities shown in the boxes in the figure will be described in this section.

8.3.1 Feature Definition

The feature definition defines the system being developed to which the Cybersecurity process will be applied. The feature definition identifies the physical boundaries, Cybersecurity perimeter, and trust boundaries of the feature, including the network perimeter of the feature. This feature definition is important since it defines the scope of the feature. Analysis activities performed on the feature are restricted to the described scope and perimeters defined in the feature definition.

8.3.2 Initiation of Cybersecurity Lifecycle

This is the beginning of the Cybersecurity lifecycle of the feature and includes developing the Cybersecurity plan (i.e., activities that will be carried out as part of the Cybersecurity lifecycle for the feature, who is responsible for the activities, the start dates, end dates, status, etc.). The Cybersecurity plan may be a simple spreadsheet, or it may be a plan in MS Project, etc. The important part of this phase is that the project is planned with respect to designing and developing the feature in the context of a Cybersecurity process.

If there is a modification to a feature that was previously developed according to the organization's Cybersecurity process, an impact analysis can be performed to determine what aspects of the feature are affected and whether or not the modifications can affect Cybersecurity. In the case of a modified feature, only the modifications that could adversely affect Cybersecurity would follow the tailored process. Modified features developed previously without a Cybersecurity process, should follow the organization's Cybersecurity process.

8.3.3 Threat Analysis and Risk Assessment

Threat Analysis and Risk Assessment (TARA) identifies **threats** and assesses the risk and residual risk of the identified threats. The results of the TARA drive all downstream Cybersecurity activities. Identifying potential threats and assessing the identified threats risk, allows valuable resources to be applied to the highest risk potential threats. In addition, since the focus is on the highest risk potential threats, a TARA facilitates downstream identification of the most valuable Cybersecurity Controls during application of more detailed analysis techniques.

Threat Analysis and Risk Assessment consists of three components or steps:

1. Threat Analysis (Threat Identification) - Identification of the potential threats to a system or organization (**stakeholders**).
2. Risk Assessment (Threat Classification) - An assessment and hence, classification, of the risk associated with a particular identified threat.
3. Risk Analysis – the threats are ranked according to level of risk and a determination is made as to whether or not the risk associated with a particular threat is at an acceptable level, or if risk reduction measures are required.

The risk assessment component of a TARA, considers the severity of the possible outcome of a potential attack on the system, and the likelihood that a potential attack can be successfully carried out. The likelihood that a potential attack can be successfully carried out is typically referred to as the "**attack potential**". The attack potential may be defined differently depending on the threat analysis and risk assessment method used. Typically, the attack potential considers a number of different factors, including elapsed time (time to identify a vulnerability, develop an attack, and mount an attack successfully), specialist expertise, knowledge of the system under investigation, level of Cybersecurity countermeasure controls, window of opportunity (access to the system), and equipment required.

If the risk analysis identifies threats that have an unacceptable risk level, then a Cybersecurity process as defined in this recommended practice may be followed to identify risk reduction measures that may be applied to reduce the threat risk to an acceptable level. These risk reduction measures are the **Cybersecurity Controls**. To determine if the risk reduction has been completed to an acceptable level, a reassessment of the threat risk may be done taking into account the Cybersecurity Controls applied to reduce the risk to an acceptable level.

There may be different criteria to determine what an acceptable level of risk is depending on the TARA method used and on what a specific organization deems acceptable or not. An acceptable level of risk may be based on the classification of risk associated with a particular threat. For example, if risk is assessed and classified according to a scale of I to IV where IV is the highest risk and I is the lowest risk, an organization may determine that identified threats with risk classification of I and II are acceptable, while threats with a risk classification of III and IV are not acceptable and require appropriate Cybersecurity Controls to be determined to reduce the risk to the acceptable levels of I and II. It is left to an organization to determine which TARA method is appropriate for their purposes, and to determine what an acceptable level of risk means with respect to the TARA method they have chosen.

The goal of a Threat Analysis and Risk Assessment is to identify potential threats to the feature, to assess the risk associated with each identified potential threat, and to classify the threats and determine if the risk is at an acceptable level or if risk reduction measures are required. The risk classification allows threats to be prioritized so an organization's resources can be focused on the highest risk threats. Appendix A provides an overview of various Threat Analysis and Risk Assessment methods to help an organization determine which method is best for application within their organization.

8.3.3.1 Identifying Cybersecurity Goals

Cybersecurity goals are the highest level Cybersecurity requirements and comprise the goals for achieving Cybersecurity for the feature. Cybersecurity goals are determined based on the results of the TARA. Once the highest risk potential threats are identified, Cybersecurity goals are identified for each of the highest risk potential threats. Cybersecurity goals may be stated in terms of what to avoid, or the inverse of the potential threat. For example, if a potential threat is *Malicious Unintended Steering*, the Cybersecurity goal for this potential threat may be expressed as *Avoid or Prevent Malicious Unintended Steering*. A single potential threat may have multiple Cybersecurity goals, and multiple potential threats may have the same Cybersecurity goals. The Cybersecurity goals along with their associated risk are used to determine the high-level strategy for achieving Cybersecurity of the system.

8.3.4 Cybersecurity Concept

The Cybersecurity concept is a description of the high-level strategy for obtaining Cybersecurity for the feature. At this stage, the Cybersecurity concept may contain the high-level Cybersecurity goals identified during the TARA, the risks associated with each of the Cybersecurity goals, and a potential high-level strategy for satisfying the Cybersecurity goals. The strategy for addressing the Cybersecurity goals may be dependent on the potential risk level of the threat associated with the Cybersecurity goals. An organization may be able to create a template of high-level strategies for the different classifications of potential threats that are identified. Creating a template based on threat risk level would simplify and streamline creation of a Cybersecurity concept. During the next phase of development, product development at the system level, the Cybersecurity concept will be updated and refined to a technical level. That is, the high-level Cybersecurity strategy will be refined from a functional level strategy to a technical strategy.

8.3.5 Identify Functional Cybersecurity Requirements

Once the high-level strategy is determined for satisfying the Cybersecurity goals for the identified threats, the functional Cybersecurity requirements can be determined. Essentially, the Cybersecurity goals identified during the TARA are the highest-level Cybersecurity requirements. These functional Cybersecurity requirements are derived from the Cybersecurity strategy and derived and refined from the Cybersecurity goals. Figure 18 provides a graphical depiction of the flow from the Cybersecurity goals to the functional Cybersecurity requirements.

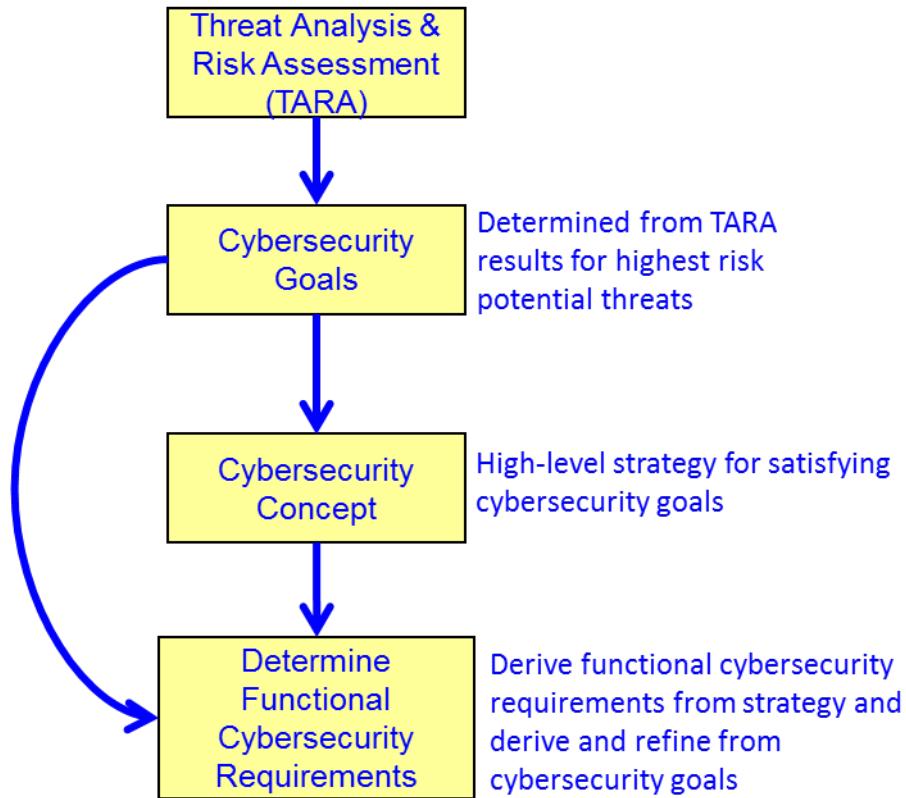


Figure 18 - Determining functional Cybersecurity requirements

8.3.6 Initial Cybersecurity Assessment

The Cybersecurity Assessment describes the current level of Cybersecurity for the feature and is developed in stages throughout the Cybersecurity lifecycle. The final Cybersecurity assessment will be completed in the Production, Operation, and Service phase of the lifecycle, and will become the Cybersecurity case that provides the justification that the feature as designed and developed is “secure” to the required level; i.e., the Cybersecurity goals identified in the TARA in the Concept Phase are satisfied.

At this stage, the interim Cybersecurity assessment may only contain the high-level Cybersecurity goals identified during the TARA, the risks associated with each of the Cybersecurity goals, and any open Cybersecurity issues that may be identified at this early stage. Open Cybersecurity issues at this point may simply be that a threat has been identified and one or more high-level Cybersecurity goals have been identified for addressing the threat, but a strategy to address the threat and to satisfy the Cybersecurity goals for the threat may not be determined yet and requires further analysis. Any open Cybersecurity issues should be addressed in subsequent updates to and refinements of the initial Cybersecurity assessment.

8.3.7 Concept Phase Review

The concept phase gate review may be performed at the completion of the concept phase activities or a review may be performed after each activity is completed (6.3 describes advantages and disadvantages of both approaches). The particular approach chosen is up to each organization. The activities to be reviewed in the concept phase include the:

- Cybersecurity Plan,
- Feature Definition,
- Threat Analysis and Risk Assessment,
- Cybersecurity Concept,
- Functional Cybersecurity Requirements,
- Initial Cybersecurity Assessment.

The review should be technical and should include:

- Verification of the correctness of the feature definition scope, boundaries, and perimeters,
- Verification of the completeness, consistency, and correctness of the TARA, both within the TARA and with respect to the Feature Definition,
- Verification of the completeness, consistency, and correctness of the Cybersecurity Concept, both within the Cybersecurity Concept and with respect to the Feature Definition and the TARA,
- Verification of the completeness, consistency, and correctness of the Functional Cybersecurity Requirements, both within the Functional Cybersecurity Requirements and with respect to the Feature Definition and Cybersecurity Concept,
- Verification of the completeness, consistency, and correctness of the Cybersecurity Assessment, both within the Cybersecurity Assessment and with respect to the TARA, the Cybersecurity Concept and the Functional Cybersecurity Requirements.

Prior to moving on to Product Development at the System Level, a successful completion of the Concept Phase Review should occur. If any issues are identified during the review, they should be corrected prior to moving on to the next stage of development.

8.4 Product Development: System Level

In the realm of Cybersecurity, the system level design is most focused on the integration of software and the electronic hardware of the system. This is because Cybersecurity deals with the movement of signals throughout the system, the storage of data, the software that sends messages, etc. As a result, the packaging of parts of the system, the interior dimensions of module, and other physical features of the system are of minimal interest for Cybersecurity (Note: The exception could be where a physical vulnerability could be used to gain access to the system's communication systems).

Figure 6 shows the activities that occur during product development at the system level. Each of the activities shown in the boxes in the figure will be described in this section.

NOTE: One should recognize that not all systems are developed from scratch and may have existing vulnerabilities that should be taken into consideration when integrating existing components. If your system is being developed using existing HW and/or SW, it may be necessary to take additional steps to help ensure that any existing vulnerabilities are addressed. A future version of this Recommended Practice will provide additional guidance on this topic.

8.4.1 Initiation of Product Development at the System Level (Planning)

The purpose of the Initiation task at the System level is to develop a plan, based on the Cybersecurity concept that resulted from the Concept phase, for how the Cybersecurity activities will be addressed at the System level, and how the integration of the electronic hardware and software activities will be ensured. Finally, the key members of the system-level Cybersecurity team will be identified as a result of this Initiation task.

8.4.2 System Level Vulnerability Analysis

The Cybersecurity team that was identified in the above Initiation task will conduct a vulnerability analysis of the System to identify potential threats. This team can use the Cybersecurity concept from the Concept phase (see 8.3.4) and the Cybersecurity Assessment (see 8.3.6) as two of the inputs to a vulnerability assessment. A vulnerability assessment is designed to find areas where an attack is likely to occur, without necessarily exploiting that vulnerability. The vulnerability assessment starts by cataloging all of a system's resources and assets, and assigns a priority level to each based on its value and importance. Next, the assessment identifies both vulnerabilities and potential threats, and the steps to mitigate or eliminate the most serious threats on the most valuable assets. Appendix A, describes several different methods that could be used for conducting and documenting a Vulnerability Analysis. Your organization needs to determine which of these methods to use. The key is to understand where the vulnerabilities lie for your System, and what the potential impacts could be of those vulnerabilities on the functions of the System.

Infrastructure vulnerabilities may impact other systems in the vehicle, systems in common across a vehicle family, or off-vehicle applications (e.g., IT back-end, intelligent transportation systems, service and maintenance). These vulnerabilities should be communicated to the other systems so that any interactions that need to be addressed to help ensure Cybersecurity can be properly identified and managed.

8.4.3 Refine Functional Cybersecurity Concept into Technical Cybersecurity Concept

In the Concept phase, a Cybersecurity Concept was defined. In this task, that Cybersecurity Concept is analyzed, along with the System-level Vulnerability Analysis, to identify the System Functions that are at most risk relative to a potential Cybersecurity event. This analysis, and the determination of the high priority functions/data for Cybersecurity, will be used to create a Technical Cybersecurity Concept that defines specific technical decisions that will be made at the System level relative to a Cybersecurity design to protect these high-priority functions/data. Examples include:

- Isolation of specific functions. For example, should a calculation for particular function be done on a separate circuit?
- Use of countermeasures (e.g., encryption, decryption).
- Not storing a copy of current **GPS** location on system.
- Defense-in-depth strategy, etc.

8.4.4 Specify Technical Cybersecurity Requirements

Once the Technical Cybersecurity Concept has been defined, the specific system requirements can be identified. In order to do this task, there should be a catalog of which specific functions (e.g., activation of airbags, braking, steering, etc.) will be performed by the system. In addition, a System Context is created to define the interfaces and functions within the system. These include:

- Hardware and software interfaces,
- Data flows,
- Data storage,
- Data processing,
- Functions which support Cybersecurity functionality.

Using this list of functions for the system, and the System Context, the specific technical requirements that should be met to achieve these functions and to support the context are defined.

8.4.5 System Design

The design work for the system level would be conducted using the organization's designated processes, tools and procedures. The intent is to design a system that will meet its requirements, including those for Cybersecurity. The status of the design work on the features that are within the system needs to be tracked to ensure that the feature level designs (hardware and software) will meet their requirements, and that the features will be able to be integrated into the overall system.

8.4.6 Feature Integration and Testing

The ability of the System to perform as intended for Cybersecurity will be assessed based on the results of the feature level testing (e.g., hardware testing, software testing), and the testing of the integration of those features into the system. The integration testing should confirm the correct communications between the features, proper identification of the features, and functioning of countermeasures as appropriate. Update the Cybersecurity Assessment as needed based on the results of the feature integration and testing.

Vehicle-Level: Integration of the systems and testing to confirm proper integration is a critical part of the verification and validation work that is done at the vehicle-level. This vehicle-level integration testing relies on the individual systems having already successfully completed the verification and validation task to confirm that they have met their system-level requirements (including Cybersecurity requirements). The purpose of the vehicle-level integration testing is to confirm that the individually validated systems will work together correctly and that the vehicle will meet its vehicle-level Cybersecurity requirements.

8.4.7 Verification / Validation of Cybersecurity Technical Requirements

The verification/validation of the Cybersecurity technical requirements is done throughout the development using a combination of traditional methods, and also by using testing methods specifically for Cybersecurity. Vulnerability testing, penetration testing, and fuzz testing are critical tools in evaluating the Cybersecurity performance of a system. A Vulnerability test plan can be developed based on the System Level vulnerability Analysis that was described in 8.4.2. The purpose of Vulnerability Testing is to confirm that the requirements that have been given to the features do, when integrated back into the system, provide effective mitigations for the vulnerabilities that were identified.

Vulnerability testing should include:

1. Vulnerability scanning methods used to detect vulnerabilities that could be exploited,
2. Exploratory testing methods used to detect and probe vulnerabilities that can be present in an implementation, and
3. Aggressive testing to attempt to break, bypass, or tamper with the Cybersecurity controls so as to demonstrate the ability to misuse the system or feature.

Vulnerability testing addresses the vehicle from the perspective of a potential adversary, using Cybersecurity analysis and attack methods, and taking advantage of access and vulnerabilities identified for the vehicle.

Penetration testing (Pen Testing for short) involves a simulated attack on the system. These active (external) attacks by an individual (or a number of individuals) provide a realistic approximation of how an actual hacker would attempt to infiltrate and exploit the system in question (see 8.2 for more information), and can be a good way to test how well the Cybersecurity controls work. However, a disadvantage of Pen Testing is that it needs to happen relatively late in the lifecycle (when there is representative system software, and maybe hardware, available) so there is less time to correct the errors.

Fuzz testing may also be done as part of the feature testing. The purpose of fuzz testing is to bombard the feature or system with data and/or signals to see if the feature or system will respond in an undesirable way that could expose a vulnerability that could be exploited. Tools are still being developed for conducting fuzz testing specifically for embedded vehicle systems, but some of the fuzz testing tools that have been developed for use in other industries (e.g., smart phones, websites) might have some applicability for specific systems (e.g., Bluetooth and Wi-Fi connections).

These testing methods - Vulnerability Testing, Pen Testing, Fuzz Testing - can either be conducted in-house or by third party organizations. It is recommended that Vulnerability and Penetration Testing be conducted by Independent Penetration Testing Assessment Teams of individuals or groups who conduct impartial assessments of systems. Impartiality implies that assessors are free from any perceived or actual conflicts of interest with regard to the development, operation, or management of the systems under assessment or to the determination of Cybersecurity control effectiveness. To achieve impartiality, assessors should not: (i) create a mutual or conflicting interest with the organizations where the assessments are being conducted; (ii) assess their own work; (iii) act as management or employees of the organizations they are serving; or (iv) place themselves in positions of advocacy for the organizations acquiring their services. Independent assessments can be obtained from elements within organizations, or can be contracted to public or private sector entities outside of organizations. (Source: NIST 800-53, CA-8, Penetration Testing/Independent Penetration Agent or Team (9)). The results of these tests should be documented, including any new vulnerability that is found.

Prior to production release, it is important to verify and validate that the Cybersecurity technical requirements have been achieved. What were the results of the Vulnerability testing that was done in the previous task? Based on the vulnerability testing, penetration testing, feature level testing, etc., have the defined technical Cybersecurity requirements been met? If any Cybersecurity requirements have not been fully met, they should be documented for use in the final Cybersecurity assessment.

8.4.8 Final Cybersecurity Assessment / Cybersecurity Case

The final Cybersecurity Assessment analyzes how well the Cybersecurity requirements are met by the system in the final stages of development, and addresses any remaining open Cybersecurity issues from the Cybersecurity assessment updates that were made during product development at the hardware level, the software level, and at the integration of hardware and software. The final Cybersecurity assessment is completed prior to release of the system for production. Once completed the final Cybersecurity Assessment becomes the **Cybersecurity Case**. The Cybersecurity Case is akin to the Safety Case in a system safety process. While the Safety Case provides the evidence and argumentation that the system as designed and developed satisfies the identified safety goals, the Cybersecurity Case would provide the evidence and argumentation that the system as designed and developed is “secure” to the required level; i.e., the Cybersecurity goals identified in the TARA in the Concept Phase are satisfied.

A plan of action and milestones should be prepared for closing out issues based on the findings and recommendations of the Cybersecurity assessment/Cybersecurity case. It may be the case that not all of the open Cybersecurity issues need to be eliminated. It is possible that some open Cybersecurity issues may be determined to be allowable by the Cybersecurity personnel. If this is the case, a rationalization should be provided that explains why the associated Cybersecurity risks are acceptable. Note that it is important to provide this rationalization in order to successfully address and virtually “close” the open issues. The system should not be released for production until either all open Cybersecurity issues are closed, or are rationalized as to why they are acceptable and are virtually “closed”.

8.4.9 Final Cybersecurity Review

The final Cybersecurity gate review may be performed at the completion of the product development at the system level phase activities or a review may be performed after each activity is completed (6.3 describes advantages and disadvantages of both approaches). The particular approach chosen is up to each organization. The activities to be reviewed in the product development at the system level phase review include the:

- System level vulnerability analysis and results,
- Technical Cybersecurity Concept,
- Technical Cybersecurity requirements,
- System design,

- Feature integration and testing and results,
- Vulnerability and penetration testing and results,
- Verification and validation of the Cybersecurity goal and the
- Cybersecurity Case.

The review should be technical and should include:

- Verification of the correctness and completeness of the system level vulnerability analysis and the results,
- Verification of the completeness, consistency, and correctness of the Technical Cybersecurity Concept both within the Technical Cybersecurity Concept and as derived and refined from the Functional Cybersecurity Concept,
- Verification of the completeness, consistency, and correctness of the technical Cybersecurity requirements, both within the technical Cybersecurity requirements and as derived and refined from the Technical Cybersecurity Concept and the functional Cybersecurity requirements,
- Verification of the completeness, consistency, and correctness of the system design with respect to the technical Cybersecurity requirements,
- Verification of the completeness, consistency, and correctness of the feature integration and testing and of the results of the feature integration and testing,
- Verification of the completeness, consistency, and correctness of the vulnerability and penetration testing and the results of the vulnerability and penetration testing,
- Verification of the completeness, consistency, and correctness of the verification and validation of the Cybersecurity goals,
- Verification of the completeness, consistency, and correctness of the Cybersecurity Case, both within the Cybersecurity Case and with respect to the final Cybersecurity Assessment.

Prior to moving on to Release for Production, a successful completion of the Product Development at the System Level Phase Review should occur. If any issues are identified during the review, they should be corrected and/or addressed prior to moving on to the release for production.

8.4.10 Release for Production

Once the final Cybersecurity review has been completed and accepted, the system is ready to be released into production. Once in production, an organization's production processes and procedures would be in effect for the duration of the lifecycle. See 8.7 for information on the tasks associated with Cybersecurity during production and during the time when the system/vehicle is being operated in the field and when service of the system/vehicle is needed.

In addition, this is also when Cybersecurity considerations for Change-in-Ownership and/or End-of-Life should be rolled out. Specifically, if the system is storing any Personally Identifiable Information (PII) that the owner would want to erase (e.g., when vehicle is sold), any methods for erasing/sanitizing PII should be described, and instructions should be made available to the owner, the dealerships, and other service providers. See 5.7 for more information on this topic.

8.5 Product Development at the Hardware Level

Figure 9 shows the activities for product development at the hardware level. Each of the activities shown in the boxes in the figure will be described in this section.

8.5.1 Background

Prior to the introduction of automation and wireless interconnections, a vehicle was a simpler, physically isolated machine with mechanical controls. Now automated controls can augment or replace human interaction and digital information can flow on internal wired networks as well as wireless networks that extend beyond the physical vehicle. As sensor and control information is being transmitted, processed, and stored, the modern vehicle begins to look more like an information system where traditional Cybersecurity objectives of integrity, availability, and even confidentiality are applicable. If vehicle data is corrupted or unavailable, it can negatively impact vehicle operation. If manufacturer proprietary information or a user's private information stored within the vehicle network is compromised, then confidentiality can be compromised.

This section focuses on hardware process development as one part of building a secure vehicle system. Hardware can be especially important to ensuring feature Cybersecurity since embedded systems have limited resources and because Cybersecurity functions like cryptographic calculations can be performed many times faster with specially designed hardware than they can be performed in software. Cybersecurity hardware may exist as a special peripheral chip that frees a **CPU** or microcontroller from computationally intensive cryptographic operations. A dedicated hardware security chip or a hardware security module embedded in a microcontroller could host multiple Cybersecurity functions that support the vehicle system as a whole such as cryptographic algorithm acceleration, secure key storage, secure data storage, and secure execution. A hardware security chip can also be implemented in tamper resistant packaging to help detect and deter physical tampering.

8.5.2 Initiation of Product Development at the Hardware Level

Initiation determines and plans the Cybersecurity activities associated with the individual sub-phases of hardware development. During this phase the Cybersecurity and system engineering teams will identify any hardware-related Cybersecurity requirements, including safety, privacy, financial, business, legal or regulatory impacts.

Cybersecurity should ultimately reduce the likelihood and impacts of threats to the vehicle in order to prevent damage to the system, the users, and the business case justification. Cybersecurity management roles and responsibilities should be defined and documented to include identifying the hardware Cybersecurity lead, defining the relationships between Cybersecurity, engineering, hardware / software, safety, and establishing the budget and scope for Cybersecurity evaluations and testing. During initiation, the Cybersecurity team, in conjunction with engineering, safety, and the business community of interest should identify potential threats as outlined in the section which follows.

8.5.3 Hardware Level Vulnerability Analysis

From a safety perspective, a hardware level analysis identifies, quantifies, and prioritizes hazards that can impact safety goals. On modern vehicles, a hardware level analysis from a Cybersecurity perspective should identify, quantify, and prioritize vulnerabilities that can lead to risks that impact Cybersecurity goals or Cybersecurity requirements. One example of a potential hardware Cybersecurity vulnerability would be the unprotected access to a Joint Test Action Group (**JTAG**) port on an ECU that can be used to extract data or code (such as firmware) from an ECU's **ROM** or **RAM**. Another example of a physical / hardware vulnerability, might be easy access to the vehicle bus through the OBD II port, even though access is mandated for emissions testing. The in-vehicle network **OBD II** port allows easy direct physical access to other-than-diagnostic information and control messages in general. Bus access through the OBDII port could also be used to inject false diagnostic message traffic or control messages. Another example of a potential physical / hardware Cybersecurity vulnerability might be a wireless connection intended for entertainment, safety, or diagnostic applications which could also allow unauthorized remote access to vehicle information and controls. In this case the wireless protocol connection exists at the physical layer of the network model prior to any higher level processing in software in the network stack. In order to help identify potential hardware attack surfaces due to hardware Cybersecurity vulnerabilities, one may focus on the physical components and interfaces within the vehicle and the potential threats to these components and interfaces.

8.5.4 Specification of Hardware Cybersecurity Requirements

The first step in specifying hardware Cybersecurity requirements is to review (and update as needed) the system Cybersecurity context which includes identifying hardware interfaces, data flow, data storage, data processing, and those systems which support Cybersecurity functionality. The next step involves understanding how the hardware supports the overall system purpose or mission including the Cybersecurity functions it should perform such as preventing unauthorized access or detecting tampering. Required Cybersecurity functions may be defined in terms of parameters such as performance, effectiveness, or timeliness. For example, when a hardware device detects that tampering has occurred, the device should erase all stored information or log the tampering occurrence it has been designed to protect. Potential constraints on the design should be identified, including internal or external threats, legal / regulatory considerations, and cost.

The hardware Cybersecurity requirements guide the creation of the Cybersecurity design and represent the standard against which the Cybersecurity of the system is measured during the testing phase.

8.5.5 Hardware Cybersecurity Design

The Cybersecurity design focuses on satisfying Cybersecurity requirements developed during the previous phase. The Cybersecurity design may call for the use and integration of existing Cybersecurity solutions to meet the Cybersecurity requirements or entirely new Cybersecurity solutions may need to be developed. The Cybersecurity design should reduce the overall risk from the threats identified during the threat analysis. It is important to measure the potential effectiveness of competing Cybersecurity design options, and to choose the options with the greatest potential to reduce risk. A risk evaluation should be completed to determine the effectiveness of the Cybersecurity design. Hardware Cybersecurity architecture metrics can be devised that help measure or quantify progress toward the Cybersecurity goals.

Hardware Cybersecurity solutions can be implemented in terms of controls that protect against tampering, unauthorized physical access, or reverse engineering. For example, tamper protection should prevent an attacker from accessing some part of the device without being detected. Physical access controls should prevent unauthorized reading or use of some part of the system (such as the internal vehicle bus) for an unauthorized purpose. Anti-reverse engineering controls should prevent an attacker from deciphering and reading proprietary or privacy-related information or algorithms.

A designer could consult a list of recommended Cybersecurity controls such as those found in Appendix D to help ensure that the hardware Cybersecurity design considers standard Cybersecurity protection methods. The Cybersecurity controls listed in Appendix D, which is only a sample list, is organized around different control families. A single control feature could be implemented as a combination of hardware and software so the hardware Cybersecurity design should be coordinated with the software Cybersecurity design. In a subsequent phase, testing should verify whether the controls called out in the design have been implemented and are functioning as intended.

An especially useful Cybersecurity design option would be to incorporate trust anchors such as **HSMs** (Hardware Security Modules) or Secure Hardware Extensions into the hardware design. A trusted hardware component can provide trusted and reliable crypto processing functions such as key generation, key storage, encryption/decryption, and random number generation. The EVITA Project developed specifications for three levels of HSMs (Full, Medium, and Light), which may be embedded in ECUs to enable secure storage and secure communication. The Full version HSM is the most expensive and the most capable. It is suited for communication between ECUs and external systems and includes hardware-accelerated asymmetric encryption/decryption functionality. The Medium version HSM is simpler and cheaper than the Full version. It does not have hardware-accelerated asymmetric encryption/decryption functionality and is best suited for communication with other ECUs within the vehicle using symmetric encryption. The Light version is the cheapest and simplest. It is best suited for communication between ECUs and vehicle sensors or actuators.

8.5.6 Hardware Level Integration and Testing

During the implementation phase, hardware Cybersecurity components are acquired and/or built, integrated, configured, tested, and documented. Hardware level integration and testing combines hardware components, and tests the combination as a group prior to system testing. The integration testing should verify that the grouping of hardware components meets the functional, performance, and reliability needs, prior to the vulnerability and penetration testing which follows.

8.5.7 Hardware Level Vulnerability Testing and Penetration Testing

Cybersecurity vulnerability testing and/or penetration testing helps to determine whether the hardware has been secured against a creative intelligent threat, such as a skilled human attacker. Testing helps determine the amount of residual risk that remains after Cybersecurity controls have been applied. It may not be possible to eliminate all risk; some risk may need to be accepted. The test results and residual risk are documented, and an individual with the authority to accept the residual risk will determine whether the risk is acceptable or whether additional Cybersecurity design work and controls are needed to lower the risk to an acceptable level. The documentation package for this stage may also include a plan of action for addressing the residual risk. For example, a particular risk could be acceptable if a policy and procedures were created that made owners/operators or maintenance personnel aware of the potential risk and provided instructions for avoiding it.

Hardware level vulnerability testing should seek to verify that known vulnerabilities and potential vulnerabilities have been mitigated. The test methodology would check against a list of known hardware vulnerabilities and their recommended mitigations to ensure the corresponding Cybersecurity controls have been implemented and are working properly. Hardware level penetration testing should simulate the actions of an attacker or attackers attempting to circumvent Cybersecurity Controls and gain control over the system. Penetration testing (see A.2 for more information) can be carried out by a range of simulated attackers having novice to expert skillsets and attacks can simulate attackers having increased knowledge of the target system with each attack or having increasingly advanced attack tools, until an attack is successful, which helps define the system's threshold of resistance to attack.

Hardware vulnerability testing and penetration testing may begin as soon as a working prototype is available, and the testing should be repeated at key points during the development lifecycle.

Vulnerability and penetration testing can be performed by personnel on an internal Cybersecurity test team as a baseline evaluation, but a qualified, independent entity (see 8.4.7) should ultimately be engaged for this testing to identify issues that an internal team may miss.

8.5.8 Verification / Validation of Hardware Cybersecurity Requirements

Cybersecurity tests covering all hardware Cybersecurity requirements are conducted to determine if the actual hardware design matches the required results. The hardware Cybersecurity design is traceable to and validated against the hardware Cybersecurity requirements, and the implementation is traceable to and validated against the hardware Cybersecurity design.

8.5.9 Refine Cybersecurity Assessment

The Cybersecurity Assessment for the feature is refined at completion of the product development at the hardware level phase. Any previously open Cybersecurity issues should be examined to determine if the product development activities at the hardware level have resulted in closure of any of the open issues. An explanation of how the issues were closed should be included in the updates to the Cybersecurity Assessment. Issues that cannot be closed will be carried over to the next refinement of the Cybersecurity Assessment. Any new open Cybersecurity issues that have been identified during this phase should be included in the assessment. If potential ways of closing the open issues in later stages of development are known, these recommendations can be included with the respective open issues. If any of the open Cybersecurity issues are deemed acceptable at this stage of development, then a rationale should be provided explaining why the open issue is acceptable and the issue should be virtually “closed” based on the rationale provided. Virtually “closed” issues, do not need to be revisited unless information is introduced at a later stage of development that invalidates the rationale used for virtually “closing” the issue.

8.6 Product Development at the Software Level

Figure 11 shows the activities for product development at the software level. Each of the activities shown in the boxes in the figure will be described in this section.

8.6.1 Initiation of Product Development at the Software Level (Planning)

This section uses the ISO 26262 Part 6 software development process framework to allow for efficient planning of the software development and is meant only as a reference. There are other frameworks that can be used based on project requirements. For development of embedded software, ISO 26262 Part 6 “Product development at the software level” is used as the basis, allowing for efficiency of common processes between development of safety-related software, Cybersecurity-related software, and software without either safety or Cybersecurity implications.

Typical activities to be included in planning of software development include:

- Planning, scheduling and resourcing of the software lifecycle phases;
- Identifying any “off the shelf” or reused software components, and determining any required qualification activities to establish the Cybersecurity capabilities of those elements;
- Identifying any tools that support the software development process, the required confidence in those tools and any guidelines for their application;
- Selection of methods to support software development (see below);
- Selection of the programming and/or modeling languages (see below);
- Planning of software integration and testing (see below).

Selection of Methods

ISO 26262 Part 6 (in common with other parts of that standard) contains extensive tables of methods. The methods are examples of techniques that are to be applied in software development that support achieving associated requirements and the required integrity (robustness). A key feature of the tables in ISO 26262 is that alternative methods to those listed may always be used, and, regardless of the selection of methods (from the tables or elsewhere), a rationale should be provided as to why the chosen methods fulfill the associated requirements. The selection of methods and the rationale are recorded in the documented planning of software development.

The guidance on methods given in ISO 26262 may be used as the starting point to select additional methods to support Cybersecurity integrity / robustness in addition to functional safety.

Selection of Programming and/or Modeling Languages

The programming language(s) are selected for implementing the Cybersecurity-relevant software. Key requirements apply to the selection of the language(s) to minimize the likelihood of the software containing vulnerabilities and these include:

- Languages should have an unambiguous definition both in terms of syntax (the permissible constructs) and semantics (the behavior resulting from the language constructs written in program code);
- Languages should support modularity, abstraction and structured constructs;
- Languages should promote creation of deterministic and analyzable code;
- Languages should support real-time systems and run-time error handling.

Where the language does not inherently support such requirements, they may be imposed on the base language through the use of language subsets, coding guidelines and analysis tools in the development environment. Such subsets and guidelines may be existing public-domain guidance or may be modified for a specific development.

For example, for the “C” programming language, the publicly-available subsets **MISRA C** and **CERT C** provide guidance on avoiding vulnerabilities and unpredictable behavior in the software. The subsets are typically enforced through the use of static analysis tools in the development environment. See the “software unit design and implementation” section below for further discussion of static analysis.

Note that while language subsets and static analysis are traditionally applied to imperative programming languages such as “C”, the requirements are equally applicable to model-based development paradigms and automatic code generation.

Planning of Software Integration and Testing

It is recommended that a strategy for software integration and testing is developed “up front” to ensure efficiencies and avoid repeating tasks; although of course this should be updated as the software is designed and implemented. For example, penetration testing is traditionally viewed as a late-in-the-lifecycle activity to prove the robustness of a product, but test cases can be specified and reused much earlier in the lifecycle during, for example, software unit testing to verify how software units respond to tainted data. Common criteria testing methods may be considered to ensure full comprehensive testing is captured (7).

8.6.2 Specification of Software Cybersecurity Requirements

As with defining the hardware Cybersecurity requirements, the first step in specifying the software Cybersecurity requirements is to review (and update as needed) the system context which includes identifying hardware and software interfaces, data flow, data storage, data processing, and systems which support Cybersecurity functionality. The next step involves understanding how the software supports the overall system purpose or mission including the Cybersecurity functions it should perform such as preventing unauthorized access or detecting tampering. Required Cybersecurity functions may be defined in terms of parameters such as performance, effectiveness, or timeliness. For example, when a piece of software detects that tampering has occurred, the software should record (and if possible report) the tampering, and change the associated Cybersecurity keys for all stored information it has been designed to protect. If the software erases the data that it is protecting, this may be the intended consequences of the attacker; therefore all Cybersecurity measures implemented should minimize and mitigate unintended consequences. Another example is implementing code signing to detect and prevent modified code from being installed or executed on a module or within a system. Potential constraints on the design should be identified, including internal or external threats, legal / regulatory considerations, and business case.

8.6.3 Software Architectural Design

The software architecture should be designed with an analysis of: the data types being used; how the data will flow; how the software will detect errors; and how the software will recover from errors. The desired results are to have the data maintain confidentiality, integrity, and availability (CIA).

FIPS 199 (10) defines CIA as:

- **Confidentiality:** “Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information” A loss of confidentiality is the unauthorized disclosure of information.
- **Integrity:** “Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity” A loss of integrity is the unauthorized modification or destruction of information.
- **Availability:** “Ensuring timely and reliable access to and use of information” A loss of availability is the disruption of access to or use of information or an information system.

The CIA as defined by NIST FIPS 199 provides a framework for understanding potential threats that should be addressed when designing software architecture. For example, if there are significant concerns with sensitive data that will be on the system (i.e., Confidentiality), determine an architecture that would provide capability of handling encryption and/or access control. In a similar fashion, the software designer should look at the other aspects of CIA and determine if there are threats associated with CIA categories that can be addressed in the design of the software architecture such as protecting Cybersecurity-critical data and/or functions.

An analysis of the data flow will help identify where the software can be partitioned and isolated. This will aid in keeping the effects or consequences from propagating to another section.

To help prevent a section of software from failing (ex: non-responsive application, stack/data manipulation), the software should implement error detection and error recovery, including malformed or corrupt input. If an error is non-recoverable, the software should revert to a pre-defined secure/safe state, and notify all dependent software modules that an error has occurred. Those dependent software modules should have use cases designed for this type of failure.

If a failed software module is able to recover at a later point, after it has notified dependent modules of its failure, it should notify dependent software modules that it has recovered and is back in an operational state. The software modules should also have use cases for this type of recovery case.

Logging of errors, failures, and recoveries should also be done. Logging assists in analysis to identify abnormalities, intrusion attempts or gaps in system robustness.

8.6.4 Software Vulnerability Analysis

As part of the software vulnerability analysis, take the analysis of the software Cybersecurity requirements and the data flow from the Software Architectural Design and define where ***trust boundaries*** exist. As data crosses these trust boundaries, define the controls that are required. This is an essential part of completing a threat model, a threat analysis, and an attack tree (reference A.1.7 for more information on attack trees).

Threat Modeling is first and foremost a process to help highlight risks during design. It helps the design team determine where to target added testing and scrutiny during Cybersecurity validation. Threat Modeling answers the need for a methodical thought process in analyzing the Cybersecurity of a system; primarily the focus is the methodology for creating attack goals that can then be used to secure a system.

Threat Modeling accomplishes its goal by focusing on the flow of data and control between components and through entry and exit points in the overall system for given functional use cases of the system. Any place where data and control passes any trust boundaries is given particular concern. Trust boundaries show locations where the level of trust changes (e.g., where untrusted external data comes into the system or safety systems interact with non-safety).

The basic steps in Threat Modeling start with decomposing the application or use case. This involves understanding how the application interacts with external entities and where data is stored and processed. This involves creating use cases to understand how the application is used to understand where a potential attacker could interact with the system. This information is documented in the Threat Model and produces specialized data flow diagrams for the application that show different paths through the overall system, highlighting privilege boundaries.

The second step involves using a threat categorization such as ***STRIDE***, ***ASF***, or ***DREAD*** to identify threats based on the break-down of the system. General threat scenarios can be used here and overlaid on the system specifics to help highlight areas of risk. In some models, severity values for process, control, and data areas are determined while others, notably Microsoft's SDL (3), any entry point or trust boundary is treated as critical for the final step.

The final stage takes the target areas from previous steps and applies controls to identified areas working from highest risk to lowest. Cybersecurity Controls can include reducing or mitigating the risk, accepting the risk, disclosing the risk (e.g., a warning label), or terminating the risk (e.g., change the functional behavior or reduce the feature). The decision on which to implement should include an understanding of both the business costs, and the costs associated with the risk being realized.

8.6.5 Software Unit Design and Implementation

During software design and implementation, good coding practices should be followed. Good coding practices include, but are not limited to:

- Input validation
- Input Sanitization (e.g., **SQL** injection)
- Secure String usage, banned **API** usage (deprecated functions), unsafe functions
- String or Array usage without an explicit length that can cause buffer overflows
- Domain specific: SQL, web, networking compared with CAN usage
- Use of standards where practical (e.g., MISRA C, CERT C)
- Static and Dynamic Analysis

Note that many of the methods (also called “design principles”) recommended in ISO 26262 that help ensure required attributes including robustness in the software unit design and implementation are typically fulfilled through a language subset such as MISRA C or CERT C. Each language subset typically concentrates on a particular area; MISRA C is intended for reliable embedded programming, while CERT C is intended for security. Language subsets may overlap. It is recommended that language subsets covering both safety and Cybersecurity (e.g., MISRA C and CERT C language subsets) be used together.

Experience shows that static analysis is a particularly useful technique in identifying software vulnerabilities in code that would otherwise successfully compile, since while the code may meet the syntactic requirements of the language it still may contain unpredictable or undefined behaviors.

The following sources can be referenced for more information:

- ISO 12207: Systems and software engineering - Software lifecycle processes (11)
- ISO 27001: Information security management (12)
- ISO 27002: Information technology - Security techniques - Code of practice for information security management (13)
- ISO 29119: Software testing standard (14)

8.6.6 Software Implementation Code Reviews

Code reviews should be conducted on the software throughout the software design and implementation phase (8.6.5), especially for new or modified software, during appropriate gate reviews. These reviews should analyze and identify coding methods and constructs that may pose or introduce risk or vulnerabilities into the system. Using language subsets can help avoid constructs that introducing vulnerabilities. Also, tools should be evaluated (e.g., optimizing compilers that may produce undefined behavior) to determine if the use of the tool could introduce or fail to detect a vulnerability. Test cases should be created to assess the risk imposed by the code. If risk exceeds what is deemed permissible, or if the code compromises the Cybersecurity or stability of the system, the code should be rewritten or mitigated.

The code review should also verify that data being passed between methods, functions, classes, modules, etc. are being sent and received as expected and are not mismatched (e.g., Improper units, big/little endian, out of bounds).

Third party libraries should also be analyzed for potential vulnerabilities that could increase the risk of a successful attack. Third party components may contain different versions of the same library. Tracking third party components can be a significant task. Resources and tools do exist to assist with this process. Some resources include:

- NIST National Vulnerability Database (NVD) (35)
- Common Vulnerability Enumeration (CVE™)

Tools and utility examples can be found under Appendix I.

8.6.7 Software Unit Testing

Unit testing verifies that an element of software, for example a subroutine, function, or a class performs as expected. Unit testing isolates the element under test from the rest of the application or system. When conducting unit testing, it is recommended that you start with the lowest element level and work up. Each element should be tested to ensure that the following perform as intended:

- Input
- Output
- Data flow / data dependency chain
- Edge cases
- Error handling
- Exception handling
- Failure modes
- Recovery modes

If an element fails unit testing, corrective action should be taken and the element needs to be retested. Regression testing should be performed as well, to ensure that the element did not adversely affect other elements.

When designing appropriate test cases for unit testing, you should have a good sampling of tests cases from:

- General test data
- Edge cases
- Error handling
- Failure/recovery handling

It is important to note that while edge cases are not routinely encountered, they tend to be a vulnerability source due to lack of adequate testing

8.6.8 Software Integration and Testing

After the unit testing, the software elements need to be integrated together. Testing should include verifying that the integrated software does not result in the integrated software operating in unintended ways, such as sending out a CAN message when one should not be sent.

Testing seeks to determine whether Cybersecurity requirements have been met. Testing should include fuzz testing on all data entry points, including wireless interfaces, **USB** and CAN. This tests the software's robustness and that software modules are communicating as intended. Testing should also include penetration testing to help determine whether the software and the system have been secured against a creative, intelligent threat such as a skilled human attacker. Testing helps determine the amount of residual risk that remains after Cybersecurity controls have been applied. It may not be possible to eliminate all risk; some risk may need to be accepted. The test results and residual risk are documented and an individual with the authority to accept the residual risk will determine whether the risk is acceptable or whether additional Cybersecurity design work and controls are needed to lower the risk to an acceptable level. The documentation package for this stage may also include a plan of action for addressing the residual risk. For example, a particular risk could be acceptable if a policy and procedures were created that made owners/operators or maintenance personnel aware of the potential risk and provided instructions for avoiding it.

8.6.9 Verification/Validation to Software Cybersecurity Requirements

During the implementation phase, software components are acquired and/or built, integrated, configured, tested, and documented. Cybersecurity tests covering all software Cybersecurity requirements are conducted to verify that the actual results match the required results.

The software Cybersecurity design is traceable to, and validated, against the software Cybersecurity requirements, and the implementation is traceable to and validated against the software Cybersecurity design.

8.6.10 Software Vulnerability Testing and Penetration Testing

Section 8.4.7 gives an overview of the testing methods of Penetration testing and Fuzz testing, both of which can be used at the software level

Penetration testing (see A.2), fuzz testing, and static code analysis should be conducted if some level or threshold of risk warrants it. The depth and breadth of that risk can help determine how expansive the penetration and fuzz testing should be.

Fuzz testing should also be conducted if some level or threshold of risk warrants it. Fuzz testing is a software testing technique that can be used to find potential Cybersecurity flaws, reliability problems and stability problems. It works by generating randomizing input to a system in an effort to make it crash, or behave in a way other than what was intended. Fuzz testing has proven to be a cost effective means of discovering potential Cybersecurity flaws in software like buffer overflows, denial of service attacks, and format bugs, all of which can be leveraged by potential attackers to gain unauthorized access to an ECU or network of ECUs.

Vulnerability and penetration testing can be performed by personnel on an independent internal Cybersecurity test team or by outside third party engagement.

8.6.11 Refine Cybersecurity Assessment

The Cybersecurity Assessment for the feature is refined at completion of the product development at the software level phase. Any previously open Cybersecurity issues should be examined and if the product development at the software level has resulted in closure of any of the open issues, the open issues should be closed and an explanation of how the issues were closed should be included. Issues that cannot be closed will be carried over to the next refinement of the Cybersecurity assessment. Any new open Cybersecurity issues that have been identified during this phase should be included in the assessment. If potential ways of closing the open issues in later stages of development are known, these recommendations can be included with the respective open issues. If any of the open Cybersecurity issues are deemed acceptable at this stage of development, then a rationale should be provided explaining why the open issue is acceptable and the issue should be virtually "closed" based on the rationale provided. Virtually "closed" issues, need not be revisited unless information is introduced at a later stage of development that invalidates the rationale used for virtually "closing" the issue.

8.7 Production, Operation and Service

8.7.1 Production

8.7.1.1 Planning

After release of the system for production, the supplier should:

- Provide evidence to the customer that the process capability is being met and maintained properly.
- Review the agreement between the customer and supplier that addresses and defines the Cybersecurity responsibilities for each party.
- Sign a supplier agreement stating they have access to, exchange of, and production monitoring of Cybersecurity-related special characteristics.
- Report Cybersecurity-related events in a timely manner and according to the supplier agreement. If a Cybersecurity-related event occurs, an analysis of that event should be performed.

After release of the vehicle for production, the vehicle manufacturer should:

- Manage end of life Cybersecurity considerations (disposal, PII data / password deletion, etc.).
- Monitor field for Cybersecurity issues.
- Follow an incident response plan for Cybersecurity issues.

8.7.2 Operation, Service (Maintenance and Repair)

8.7.2.1 Field Monitoring

A field monitoring process reviews a variety of sources to identify those field issues and incidents that should be addressed as potentially impacting Cybersecurity. A field monitoring process may monitor data gathered from law enforcement, insurance, media articles, hacker chatter, other vehicle organizations, etc. to determine high areas of risk or what is happening in real time. Here is a model of areas from where Cybersecurity incident information can be gathered and with whom it could be shared.

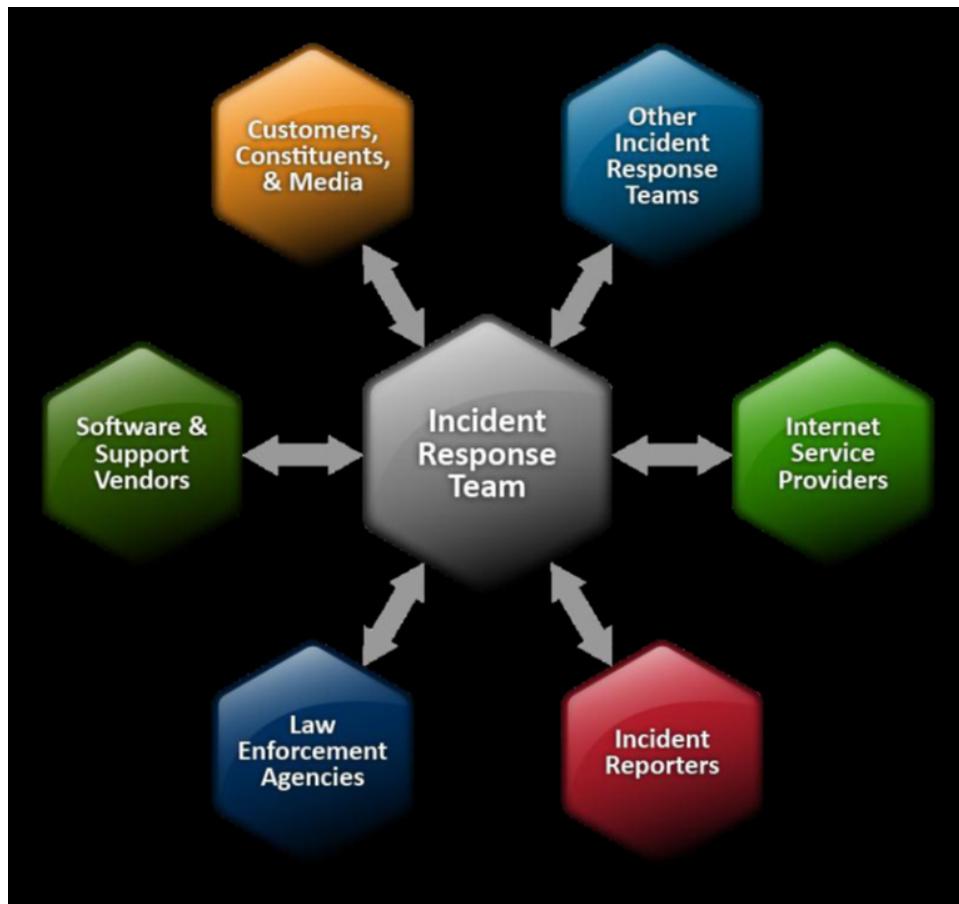


Figure 19 - Example incident response team data sources (15)

A team may be formed that determines the root cause of the incident, identifies corrective actions, assures implementation of corrective actions, and monitors the corrective action in the field to determine if the precautions taken have satisfactorily addressed the root cause of the incident. However, monitoring for similar incidents in the field and not finding any, does not indicate that the potential vulnerabilities have been completely eliminated. The fact that a system is not breached in a certain period of time is not an indicator that a future breach is unlikely, or that there aren't any vulnerabilities that have not yet been exploited.

8.7.2.2 Incident Response

An incident response process responds to Cybersecurity incidents that are reported to an organization or that occur in the vehicle industry. These incidents can be actual attacks on your organization's cyber-physical vehicle systems or attacks on other organization's cyber-physical vehicle systems. It is a process that becomes active once an incident is identified, works to contain (minimizes loss and destruction) the incident, and mitigates a Cybersecurity incident such as malware infections, **hacker intrusions**, data breaches, etc. Regular monitoring for attacks is essential. Establishing clear triage procedures for handling incidents is critical. It is also vital to build relationships and establish suitable means of communication with internal groups (e.g., IT, Human Resources, Public Relations, Legal) and with external groups (e.g., law enforcement, other incidence response teams from other vehicle organizations, or public Information Sharing and Analysis Centers (ISAC) or other similar forums). It may be beneficial to hire an outside Cybersecurity supplier to help determine if that vulnerability has been effectively addressed.

Reporting potential incidents accurately and in a timely fashion is an important part of an effective Cybersecurity incident management process, as the failure to report discovered potential incidents can lead to significant consequences, no matter how well the system is designed or how adequately the response or containment procedures are developed. Such as, other groups not being aware of lessons learned and not addressing a similar vulnerability properly, etc. Formal Cybersecurity event reporting, forensics and escalation procedures should be in place. All employees, contractors, and third party users should be made aware of the procedures for reporting the different types of events and weaknesses that might have an impact on the Cybersecurity of organizational assets.

Establishing an incident response capability may include the following actions:

- Creating an incident response policy and plan.
- Developing procedures for performing incident handling and reporting.
 - Determining if a threat is real,
 - Root cause analysis,
 - Forensics Analysis,
 - Determine operational impact,
 - Determine commercial impact,
 - Know how to handle sensitive information properly,
 - Document actions taken,
 - Communication,
 - Documenting lessons learned and fold back into new designs.
- Setting guidelines for communicating with outside parties regarding incidents.
- Selecting a team structure and staffing model (e.g., technically competent, informed about vehicle systems, trained)
- Establishing relationships and lines of communication between the incident response team and other groups (both internal and external).
- Determining if an incident requires escalation:
 - If an incident results in public safety concerns,
 - If an incident results in adversely affecting the company's reputation or integrity,
 - If an incident results in financial loss (examples),
 - Vehicle theft
 - Warranty
 - Loss of sales
 - Unauthorized access to features/functions
 - Results in higher insurance costs
 - Fraudulent commercial transactions
 - If an incident results in loss of privacy (examples),
 - Unauthorized personally identifiable information (PII) obtained
 - Unauthorized vehicle tracking

- If an incident results in loss of function or denial of service (examples),
 - Customer dissatisfaction
 - Vehicle will not start
- Determining what services the incident response team should provide.

More details behind building a proper Incident Response Team along with necessary capabilities can be found in NIST 800-61 Computer Incident Handling Response Guide (15).

8.7.2.3 Execution and Maintenance of an Incident Response Process (15)

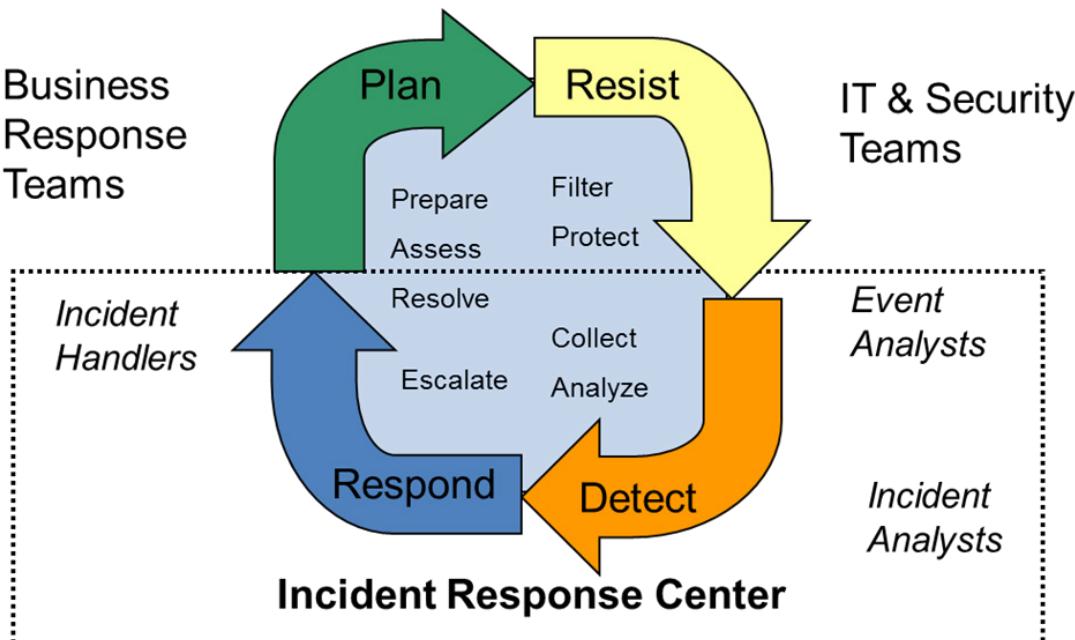


Figure 20 - Example incident response process

An organization should be in place to have teams responsible for detecting and analyzing the data while others escalate (assign priority, alert staff, report timeliness) and resolve/eradicate the issues. Organizations should create written guidelines for prioritizing incidents and they should use the lessons-learned process to gain value from the incidents. Once an organization develops a plan and gains management approval, the organization should implement the plan and review it annually to ensure it is maturing the capability and fulfilling their goals for incident response.

Standard operating procedures, specific technical processes or techniques, checklists and forms should be used by the Incident Response Team. Following standardized responses should minimize errors, particularly those that might be caused by ad hoc incident handling.

The following is an example checklist outline for handling incidents. The checklist provides general guidelines to handlers on the major steps that should be performed; it does not dictate the exact sequence of steps that should be followed.

Table 1 - Example incident handling checklist (15)

		Action	Completed
Detection and Analysis			
1	Determine whether an incident has occurred		
1.1	Analyze the precursors and indicators		
1.2	Look for correlating information		
1.3	Perform research (e.g., search engines, knowledge base)		
1.4	As soon as the handler believes an incident has occurred, begin documenting the investigation and gathering evidence		
2	Prioritize handling the incident based on the relevant factors (functional impact, information impact, recoverability effort, etc.)		
3	Report the incident to the appropriate internal personnel and external organizations		
Containment, Eradication, and Recovery			
4	Acquire, preserve, secure and document evidence		
5	Contain the incident		
6	Eradicate the incident		
6.1	Identify and mitigate all vulnerabilities that were exploited		
6.2	Remove malware, inappropriate materials, and other components		
6.3	If more affected hosts are discovered, (e.g., new malware infections), repeat the Detection and Analysis steps (1.1, 1.2) to identify all other affected hosts, then contain (5) and eradicate (6) the incident for them		
7	Recover the incident		
7.1	Return affected systems to an operational ready state		
7.2	Confirm that the affected systems are functioning normally		
7.3	If necessary, implement additional monitoring to look for future related activity		
Post-Incident Activity			
8	Create a follow-up report		
9	Hold a lessons learned meeting (mandatory for major incidents, optional otherwise)		

8.8 Supporting Processes (16)

The content of this section gives high-level descriptions of the key supporting processes that should be in place as part of the foundation that the Cybersecurity process is built upon. If an organization is not currently using one or more of these supporting processes, successfully implementing a Vehicle Cybersecurity process will be more difficult.

8.8.1 Configuration Management

The purpose of a Configuration Management process is to manage the systems as they proceed in the product development lifecycle:

- To ensure that the principles and general work conditions the system was originally created under can be uniquely identified and reproduced in a controlled manner.
- To ensure the relations and differences between earlier and current versions of the system can be traced.
- Auditing and reporting on a system's configuration baseline.
- To meet applicable prerequisites of relevant lifecycle phases where configuration management of the system is planned.

8.8.2 Requirements Management

The first objective is to ensure the correct definition of requirements with respect to their attributes and characteristics.

The second objective is to ensure consistent management of requirements throughout the entire lifecycle of a system.

- Maintenance of these requirements (update use cases, ensure no contradictions within the requirement itself or with other requirements, maintain a hierarchical structure so no duplicating of requirements occurs, etc.).
- Creation of test procedures to validate that those requirements have been met.
- Plan regular release cadence so that multiple requirements can be bundled for release once approved and proven out in order to reach global distribution.
- Retain traceability from the Cybersecurity goals through the implementation of the requirements, and the verification and validation of the requirements.
- Meet certain authoring attribute and content in each requirement [clear (unambiguous, comprehensible), consistent, complete (comprehensive), feasible, testable, etc.].

8.8.3 Change Management

The objective of change management is to analyze and control changes to systems/products throughout the product lifecycle. The systematic planning, control, monitoring, implementation and documentation of changes made to a given product are required. For this purpose, decision-making processes for change are introduced and established, and responsibilities are assigned to the parties involved, specifically:

- A change history log should be automatically maintained to document what changes are made to a system/product. For each revision provide a date, reason for requested change (and/or Change Request number), and a description of the exact change.
- Take change requests into the “approval board” to get approval of the proposed change. Ensure documentation of the change request includes the request author, date requested, reason for requested change, a description of the exact change, etc.
- Have a team/approval board in place to review revisions and approve requests prior to release.
 - Conduct an impact analysis (all products affected) to be provided when requesting approval. For example, potential impacts on Cybersecurity would be assessed before changes are made.
 - Provide an introduction plan when requesting approval.
 - Identify all parties involved.
 - Assign responsibilities to those involved.
- Develop a plan to implement the approved changes. This needs to comprise both the release of new requirements and changes to existing requirements, and how the introduction of these approved changes will be released, especially if there are global impacts and/or impacts on more than one system/product. See also Configuration Management.
- Not only is it necessary to test the work product to verify the issue has been resolved at the component or system level, but subsequent monitoring of data in the field after the fix has been put in place should also occur to ensure the change had the expected improvement impact.

NOTE: Here change is understood as modification due to: anomalies, removals, additions, enhancements, obsolescence of components, etc.

NOTE: Configuration management and change management are initiated at the same time. Interfaces between the two processes are defined and maintained to enable the traceability of changes.

8.8.4 Documentation Management

The primary objective is to develop a documentation management strategy for the entire system lifecycle in order to facilitate an effective and repeatable documentation management process. For each system, the following documents/artifacts should be comprehended in the Document Management strategy:

- Cybersecurity Plan
- Feature Definition
- System Context
- Threat Analysis and Risk Assessment
- Cybersecurity Concept
- Functional Cybersecurity Requirements
- Cybersecurity Assessment/Cybersecurity Case

It is recommended that these documents be stored securely and only be made accessible to trusted and authenticated parties.

Duplication of information within a document, and between documents, should be avoided to aid maintainability.

NOTE: The documentation can be in the form of a single document containing the complete information for the work product or a set of documents that together contain the complete information for the work product.

The documentation process should be planned in order to make documentation available:

- During each phase of the development lifecycle for the effective completion of the phases and verification and validation activities,
- For the management of Cybersecurity, and
- As an input to the Cybersecurity assessment.

The documents should be:

- Precise and concise,
- Structured in a clear manner,
- Easy to understand by the intended users, and
- Maintainable,
- Organized to facilitate the search for relevant information.

8.8.5 Quality Management

Establish an internal quality management system similar to QS 9000 (17), ISO/TS 16949 (18).

Any Cybersecurity change should follow the normal corporate quality process. That is, development of quality documentation (Design FMEAs, Boundary Diagrams, Quality History Reports, etc.). The incident reports are quality assurance documents that are confidential and should be stored and distributed based on the organization's policies.

- Quality management should be evidence based.
- Any problems arising in the process should be detected as early as possible.
 - Evaluations at each gateway are intended to catch errors.
- Roles and responsibilities should be clear and explicit.
- Knowledge and information about the issue should be documented and shared.
- Regular evaluation should capture lessons learned and lead to continuous improvement.
- As with any monitoring activity it is useful to prioritize the Cybersecurity metrics in terms of their level of importance in ensuring the quality of the process. There is a cost to benefit trade off assessment that can be made in order to achieve high quality outputs in a timely manner.

8.8.6 Requirements for Distributed Development (with suppliers)

In addition, the supplier's ability to develop Cybersecurity systems according to a company's internal process and appropriate risk level should be assessed. The supplier selection criteria should include an evaluation of the supplier's capability to develop and produce the feature as well as their expertise in the Cybersecurity vulnerability domain, if available. The following items should be considered:

- Evidence of the supplier's capability to develop Cybersecurity-critical systems, if available.
- Evidence of the supplier's capability to follow a well-structured Cybersecurity process in development.
- Evidence of the supplier's quality management system.
- Evidence of the supplier's past performance and quality history in developing critical systems.
 - The evidence might not be with respect to Cybersecurity development, given that this is a new area.
- Evidence of the ability of the supplier to provide Cybersecurity support over the lifetime of the feature.

A Development Interface Agreement between the customer and supplier should be developed in order to establish the supplier responsibilities for a given project. Responsibilities and expectations of the selected supplier(s) should apply to all aspects of the relationship. The Development Interface Agreement should include at a minimum:

- The Cybersecurity responsible person from the supplier who will oversee the supplier development and be the main point of contact with the customer.
- An agreement on the Cybersecurity process that will be followed by the supplier.
- An agreement of the Work Products that will be made available to the customer from the supplier.
 - Work Products that will be shared with the customer.
 - Work Products that will only be shown to the customer and reviewed with the customer, but not released to the customer.

- An agreement on timing of the development process.
- An agreement on scheduled technical gate reviews.
 - Where and when they will be held and what will be reviewed.
- An agreement on sharing of knowledge from the supplier of known Cybersecurity breaches or attempted breaches.
- An agreement between the customer and supplier on a process to both report on and respond to Cybersecurity incidents.
 - Both during development and after release for production.
- An agreement that the supplier will provide Cybersecurity support over the lifetime of the item and a process to provide this support.

9. NOTES

9.1 Revision Indicator

A change bar (I) located in the left margin is for the convenience of the user in locating areas where technical revisions, not editorial changes, have been made to the previous issue of this document. An (R) symbol to the left of the document title indicates a complete revision of the document, including technical revisions. Change bars and (R) are not used in original publications, nor in documents that contain editorial changes only.

PREPARED BY THE SAE VEHICLE ELECTRICAL SYSTEM SECURITY COMMITTEE

APPENDIX A – DESCRIPTION OF CYBERSECURITY ANALYSIS TECHNIQUES

Appendix A is provided as a reference to further research and to facilitate design and process improvements. Appendix A is not a comprehensive listing of Cybersecurity analysis techniques.

A.1 OVERVIEW OF THREAT ANALYSIS & RISK ASSESSMENT AND VULNERABILITY ANALYSIS METHODS

This appendix outlines a sampling of security analysis techniques including the methods used by the E-Safety Vehicle Intrusion Protected Applications (EVITA) program, the Threat, Vulnerabilities, and implementation Risks Analysis (TVRA) method, the Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) method, and the HEALing Vulnerabilities to ENhance Software Security and Safety (HEAVENS) method and attack tree information, in general. This is not intended to be a comprehensive list and this document does not, at this time, recommend a specific method. Therefore, it is up to each organization to determine whether to use one of the methods described below, or whether to use a different method. Examples of applying some of these methods are given in Appendix C, Appendix Overview of Threat Analysis & Risk Assessment and Vulnerability Analysis Methods

A.1 describes some of the methods that can be used for Threat Analysis and Risk Assessment (TARA) and Vulnerability Assessment. TARA was described in 8.3.3.

Vulnerability analysis is also known as vulnerability assessment. Vulnerability analysis techniques attempt to identify and classify potential Cybersecurity vulnerabilities or holes in the software and hardware of the system being developed that may be exploited by an attacker. Some identified vulnerabilities may be easier to exploit than others. Therefore, it is beneficial to classify the vulnerabilities to determine which vulnerabilities require the most attention. Once vulnerabilities are identified and classified, the appropriate Cybersecurity Controls can be determined to either eliminate the vulnerability or to make the vulnerability more difficult for an attacker to exploit.

Vulnerability analysis can be performed at different levels; the system level, hardware level, and software level. Some analysis methods, such as attack trees, may be used across different levels.

One concept to keep in mind when doing a Risk Assessment is the principle of “Residual Risk”. ISO/IEC 27001: defines residual risk as “the risk remaining after risk treatment” (12). In other words, after having identified the risks and determined what Cybersecurity controls will be used to mitigate risk, there will still be some remaining risk (residual risk) at some level since it is not possible to eliminate all risk. The Risk Assessment is redone taking into account the known applicable Cybersecurity controls. This can also provide information on how the likelihood of an attack may have changed, and perhaps what changes there are in the severity of an attack. See ISO/IEC 27001 for additional information on the risk assessment.

A.1.1 EVITA Method

EVITA stands for **E**-Safety **V**ehicle **I**ntrusion **P**rotected **A**pplications. The EVITA project was started in 2008 and was co-funded by the European Commission. It consisted of a number of organizations, including MIRA, BMW Group Research and Technology, Bosch, Continental, ECRYPT, Fujitsu, and Infineon. The goal of the project was to design, verify, and prototype an architecture for vehicle on-board networks where Cybersecurity-relevant components are protected against tampering and where sensitive data are protected against compromise. To satisfy the goal, the aim was to infer Cybersecurity functional requirements based on the key methodology from ISO/IEC 15408 (7) and adopting the ISO/DIS 26262 process together with systems engineering practices.

EVITA considered four Cybersecurity objectives:

- Operational – to maintain the intended operational performance of all vehicle and ITS functions,
- Safety – to ensure the functional safety of the vehicle occupants and other road users,
- Privacy – to protect the privacy of vehicle drivers and the intellectual property of vehicle manufacturers and their suppliers,
- Financial – to prevent fraudulent commercial transactions and theft of vehicles.” (19)

For each of the Cybersecurity objectives, the EVITA project considered:

- Threat Identification: Used “dark-side” scenarios and attack trees to identify generic threats and hence generic Cybersecurity requirements.
- Threat Classification: Developed recommendations for classifying threat risk based on severity of the threat outcome and probability of a successful attack.

Risk Analysis: Recommendations for actions based on the resulting risk classification of the threats.

For Threat Identification, developing the dark-side scenarios for EVITA consisted of:

- Identification and classification of possible attack motivations,
- Evaluation of associated attacker capabilities (e.g., technical, financial),
- Attack modelling, comprising:
 - Identification of specific attack goals that could satisfy the attack motivations, and
 - Construction of possible attack trees that could achieve attack goals, based on the functionality identified in the use cases (20) and (19).

The identified attack goals become the top events in the attack trees. The attack trees are then constructed by identifying one or more attack objectives that satisfy the attack goal, followed by identifying one or more attack methods that can be used to achieve the attack objectives. Additional information on attack trees can be found in A.1.7.

The EVITA Threat Classification component of the EVITA Threat Analysis and Risk Assessment method is based on the good practice Risk Assessment method used in the ISO 26262 Hazard Analysis and Risk Assessment method. The severity determination is based on the ISO 26262 classes of severity, but is expanded to consider non-safety-related outcomes and potential consequences on multiple vehicles since severity in ISO 26262 only considers safety-related outcomes and single vehicles. Table 2 shows the severity table used in the EVITA Threat Classification. Text in red shows the extensions to the severity classification beyond those used in functional safety (ISO 26262).

Table 2 - EVITA severity classes

Class	Safety	Privacy	Financial	Operational
S0	No injuries	No unauthorized access to data	No financial loss	No impact on operational performance
S1	Light or moderate injuries	Anonymous data only (no specific driver or vehicle data)	Low-level loss (~\$10)	Impact not discernible to driver
S2	Severe injuries (survival probable) Light or moderate injuries for multiple vehicles	Identification of vehicle or driver Anonymous data for multiple vehicles	Moderate loss (~\$100) Low losses for multiple vehicles	Driver aware of performance degradation Indiscernible impacts for multiple vehicles
S3	Life threatening (survival uncertain) or fatal injuries Severe injuries for multiple vehicles	Driver or vehicle tracking Identification of driver or vehicle, for multiple vehicles	Heavy loss (~\$1000) Moderate losses for multiple vehicles	Significant impact on performance Noticeable impact for multiple vehicles
S4	Life threatening or fatal injuries for multiple vehicles	Driver or vehicle tracking for multiple vehicles	Heavy losses for multiple vehicles	Significant impact for multiple vehicles

The probability of a successful attack in EVITA is based on the concept of “attack potential” used in IT security evaluation, and considers both the attacker and the system. With respect to an attacker, the attack potential considers a number of factors, such as time required for an attacker to determine how to attack a system and to perform a successful attack, expertise required of the attacker, knowledge of the system required, the need for specialist equipment, etc. Each factor has a number of classes each assigned with a numerical value; for example, the classes for attacker expertise and the corresponding numerical values are layman (0), proficient (3), expert (6), and multiple experts (8). The attack potential is also divided into classes based on the ranges of the sum total of the numerical values assigned to each of the factors. The classes of attack potential are: Basic, Enhanced-Basic, Moderate, High, and Beyond High. The attack potential ranges from Basic – meaning the feature is easily attacked, to Beyond High – meaning the feature is extremely difficult to attack. An attack probability is then assigned based on the determined attack potential. Table 3 shows the rating of attack potential and attack probability.

Table 3 - Rating of attack potential and attack probability

Values	Attack potential required to identify and exploit attack scenario	Attack probability (reflecting relative likelihood of attack)
0-9	Basic	5
10-13	Enhanced-Basic	4
14-19	Moderate	3
20-24	High	2
>=25	Beyond High	1

The severity and attack probability are then combined using a “risk graph” approach to identify the risk associated with each threat. This approach is analogous to Table 4 “ASIL determination” of ISO 26262 Part 3 (28). However, unlike the ASIL determination in ISO 26262, there is not a single mapping for Cybersecurity risks since in Cybersecurity, severity is a 4-component vector. In addition, controllability has to be considered for safety-relevant Cybersecurity risks. Table 4 shows the Cybersecurity risk graph for non-safety-related Cybersecurity threats (privacy, financial, and operational). The severity shown in the table, S_i , represents S_p , S_f , and S_o , where S_p represents severity with respect to privacy threats, S_f represents severity with respect to financial threats, and S_o represents severity with respect to operational threats. The determined severity for each potential threat is mapped to the appropriate classification shown in the table, 1 – 4, and the determined attack probability, $A = 1 – 5$. The intersection in the matrix of the severity and attack probability determines the risk, $R0 – R6$, for the potential threat being assessed, where $R0$ represents the lowest risk and $R6$ represents the highest risk.

Table 4 - Cybersecurity risk graph for privacy, financial, and operational Cybersecurity threats

Security Risk Level (R)		Combined Attack Probability (A)				
		A=1	A=2	A=3	A=4	A=5
Non-Safety Severity (S_i)	1	R0	R0	R1	R2	R3
	2	R0	R1	R2	R3	R4
	3	R1	R2	R3	R4	R5
	4	R2	R3	R4	R5	R6

The risk graph for safety-related threats is slightly more complicated, since controllability needs to be determined and included in the safety-related risk graph. The controllability classification is used to assess the potential for a human to act in such a way as to avoid a potential accident associated with a safety-related threat. Controllability is classified as C1 – C4, where C1 means it is possible for a normal human response to avoid an accident and C4 means that a human cannot act in such a way as to avoid the accident. Table 5 shows the controllability classes and their corresponding definitions.

Table 5 - Controllability classifications of safety-related threats

Class	Meaning
C1	Despite operational limitations, avoidance of an accident is normally possible with a normal human response
C2	Avoidance of an accident is difficult, but usually possible with a sensible human response
C3	Avoidance of an accident is very difficult, but under favorable circumstances some control can be maintained with an experienced human response
C4	Situation cannot be influenced by a human response

The risk graph for safety-related threats consists of four sub-matrices; one matrix for each controllability classification. Thus, for safety-related threats, the risk determination is a combination of controllability, severity, and combined attack probability. A portion of the risk graph for safety-related threats is shown in Table 6. Note that the risk levels will be different for the different classes of controllability in Table 6.

Table 6 - Portion of risk graph for safety-related threats

Controllability (C)	Safety-Related Severity (S_s)	Combined Attack Probability (A)				
		A=1	A=2	A=3	A=4	A=5
C=1	...					
C=2	...					
C=3	$S_s=1$	R1	R2	R3	R4	R5
	$S_s=2$	R2	R3	R4	R5	R6
	$S_s=3$	R3	R4	R5	R6	R7
	$S_s=4$	R4	R5	R6	R7	R7+
C=4	...					

Once the risk is determined for each potential threat, the Cybersecurity goals are identified, and the threats can be prioritized based on risk level. The higher the risk level, the more rigor can be applied in implementation of the process.

A.1.2 EVITA Method Applied at the Feature Level using Threat and Operability Analysis (THROP)

The EVITA method as it was developed was applied apart from a particular feature or system. However, the method can be adapted to apply at the feature or system level. The process described in this recommended practice, is applied at the feature level. This section describes how to apply the EVITA method at the feature (or system) level. The method described in this section provides a systematic and consistent way to identify threats relevant to the feature under evaluation. The method is derived from the well-known **HAZOP** (Hazard and Operability Analysis) method that is often used in system safety engineering. Rather than HAZOP, the method is called a **THROP** (Threat and Operability Analysis). A THROP is similar to a HAZOP except that it considers potential threats rather than potential hazards. Akin to HAZOP, the THROP addresses risk from a functional perspective for a particular feature. Threats are defined at the functional level based on the primary functions of the feature being analyzed. The primary functions of the feature that are identified in the feature definition are recorded in a matrix and guidewords are applied to the functions to identify the potential threats. For example, a potential generic threat may be *potential maliciously caused undesired behavior of a feature*.

The steps for performing a THROP are to:

1. Identify the primary functions of the feature (this is done during the feature definition),
2. Apply guidewords to the functions to identify potential threats
 - a. e.g., Malicious unintended “function”, Malicious incorrect (too high, too low, ...) “function”, Malicious loss of “function”, and
3. Determine potential worst-case scenario outcomes from the potential malicious behavior
 - b. e.g., a scenario for a malicious loss of “function” threat, could be loss of ability to start a vehicle.

Once the potential threats have been identified and the potential worst-case scenarios have been identified, the risks of the potential threats can be assessed by applying the EVITA risk assessment method described in 8.1.1. The identified potential threats can then be ranked according to risk level so the focus of further analysis can be on the highest risk threats. Cybersecurity goals can then be determined for the highest risk threats and a unique ID can be assigned and used to identify each Cybersecurity goal. Table 7 shows an example spreadsheet with the column headings that can be used for applying the EVITA method at the feature level using the Threat and Operability Analysis (THROP). An example of using THROP to apply EVITA at the feature level is given in C.1.

Table 7 - Column headings for EVITA method applied at feature level using THROP

Feature: Remote Vehicle Disable					Severity		Attack Potential				Attack Probability		Controllability (Safety)		Risk				Cybersecurity Goal ID	Cybersecurity Goals		
Threat ID	Function	Potential Item Threats	Potential Vehicle Level Threat	Potential Worst-Case Threat Scenario	Financial	Operational	Privacy	Safety	Elapsed Time	Expertise	Knowledge	Window of Opportunity	Equipment Required	Attack Probability Total	Attack Prob.	Controllability (Safety)	Financial	Operational	Privacy	Safety	Cybersecurity Goal ID	Cybersecurity Goals

A.1.3 TVRA

TVRA, which stands for Threat, Vulnerabilities, and implementation Risks Analysis, is a process-driven threat assessment / risk assessment methodology which was developed in 2009 and updated in 2010 by the European Telecommunications Standards Institute (**ETSI**). The current standard ETSI TS 102 165-1 V4.2.x (2010) TISPAN describes how the TVRA is completed in 10 steps to systematically identify unwanted incidents to be prevented in a system. TVRA identifies the assets in the system and their associated weaknesses and threats and determines the risk to the system by modeling the likelihood and impact of attacks on the system's vulnerabilities.

The ten steps of TVRA are outlined as follows:

1. Identify the target system assets and specify the goal, purpose, and scope of the analysis.
2. Identify the objectives and produce a high level statement of the Cybersecurity issues to be resolved.
3. Identify the functional Cybersecurity requirements (derived from step 2).
4. Inventory assets and refine the descriptions from step 1 and add additional assets identified in steps 2 and 3.
5. Identify threats, vulnerabilities that can be exploited, and the consequences of the exploitation.
6. Determine the occurrence, likelihood, and impact of the threats.
7. Determine the risks.
8. Identify Cybersecurity Controls to reduce risks.
9. Perform a Cybersecurity Controls cost-benefit analysis to identify which Cybersecurity Controls should be implemented first.
10. Detailed requirements for implementing the Cybersecurity services and capabilities identified in step 9.

TVRA was developed for and is best suited for data / telecommunications networks rather than the combined control and data networks present in cyber physical systems such as vehicles.

A.1.4 OCTAVE

OCTAVE, which stands for Operationally Critical Threat, Asset, and Vulnerability Evaluation, is a process-driven threat assessment / risk assessment methodology which was first developed in 1999 by the Software Engineering Institute (**SEI**) in coordination with the Department of Defense (**DoD**) Telemedicine and Advanced Technology Research Center (**TATRC**). OCTAVE was intended to address the DoD's need to comply with the Health Insurance Portability and Accountability Act (HIPAA) Security Rule. The OCTAVE methodology was adopted by DoD medical treatment facilities and later by a number of commercial medical facilities. Another organization reportedly using OCTAVE is the National Center for Manufacturing Sciences.

OCTAVE is best suited for enterprise information security risk assessments. OCTAVE is especially good at bringing together stakeholders with system experience and subject matter experts with security experience through a progressive series of workshops to develop a thorough organizational and technological view of the problem domain. A series of detailed worksheets are completed in the workshops to identify assets, current practices, Cybersecurity requirements, threats, and vulnerabilities and then to develop a strategy and plan for mitigating risks and protecting assets. The phases of the OCTAVE method are illustrated in Figure 21.

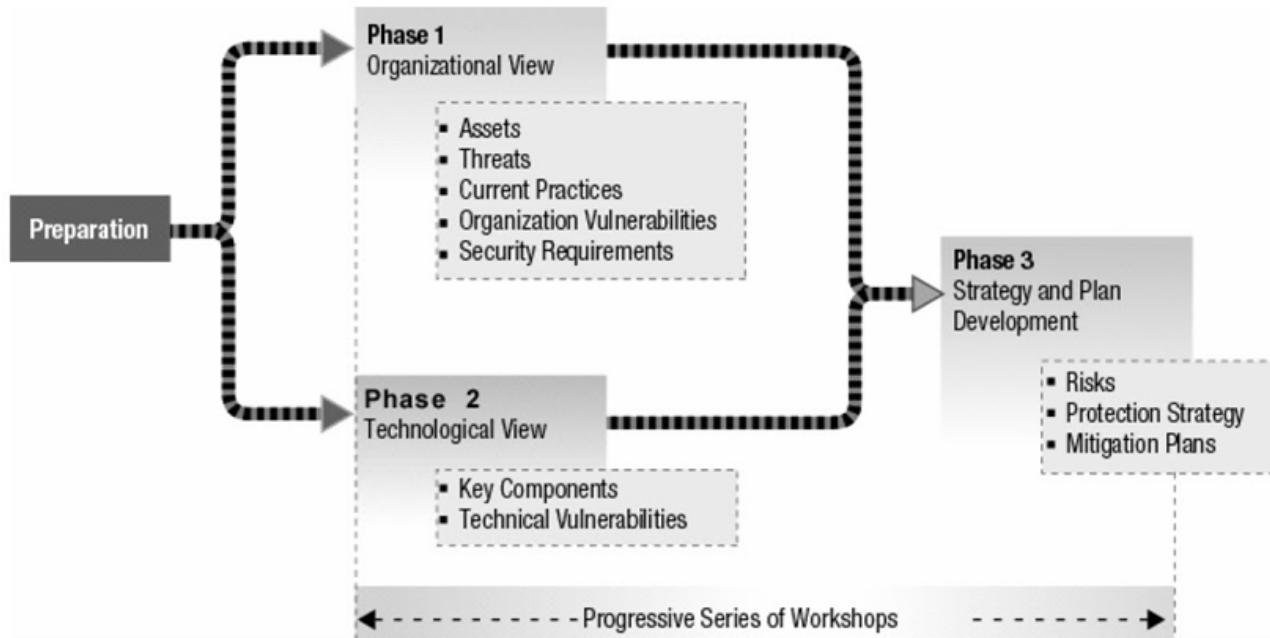


Figure 21 - Phases of the OCTAVE method (21)

OCTAVE workshops include an interdisciplinary team composed of members representing the organization's business units, Information Technology (IT) department and Cybersecurity department. Two more agile variations of OCTAVE have also been developed: one for organizations of fewer than 100 people (OCTAVE-S) and another streamlined approach that is only focused on information assets (OCTAVE Allegro). OCTAVE Allegro includes the eight steps illustrated in Figure 22. These eight steps are completed with the aid of three questionnaires and ten separate worksheets which are completed by participants attending a series of workshops.

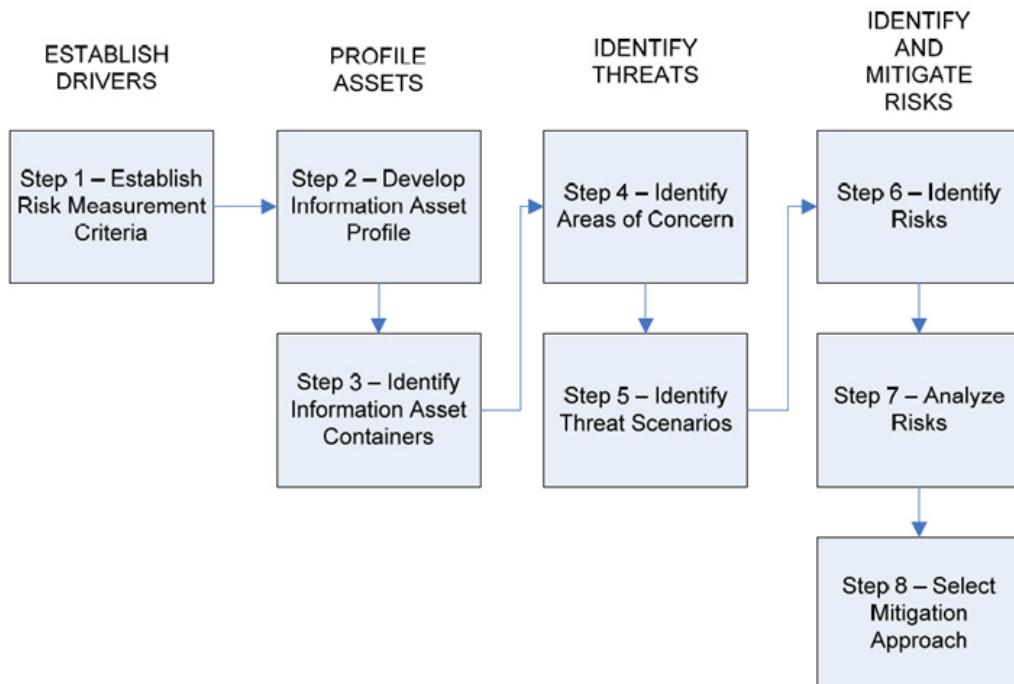


Figure 22 - OCTAVE allegro roadmap (22)

The OCTAVE phases and process steps can be correlated to the steps in the National Institute of Standards and Technology (NIST) Special Publication (SP) 800-30 as illustrated in Table 8. The phases correspond to the phases shown in Figure 21 - Phases of the OCTAVE method and the processes correspond to the steps shown in Figure 22.

Table 8 - Correlation of OCTAVE phases (21) and process steps to NIST SP 800-30 (16)

NIST SP 800-30 Steps	OCTAVE Phase/Process
Step 1: System Characterization	OCTAVE Phase 1/Processes 1 - 3
Step 2: Threat Identification	OCTAVE Phase 1/Process 4
Step 3: Vulnerability Identification	OCTAVE Phase 2/Process 5 - 6
Step 4: Control Analysis	OCTAVE Phase 3/Processes 7 - 8
Step 5: Likelihood Determination	OCTAVE Phase 3/Process 7
Step 6: Impact Analysis	OCTAVE Phases 1/2/3/Processes 1 - 7
Step 7: Risk Determination	OCTAVE Phase 3/Process 7
Step 8: Control Solutions	OCTAVE Phase 3/Process 8
Step 9: Results Documentation	OCTAVE Phases 1/2/3/Processes 1 - 8

Because the OCTAVE approach is thorough and it incorporates input from business, information technology, and security, it is useful in eliciting security-related information that might otherwise be overlooked; however, it may require a sizeable investment in time and resources to complete. The OCTAVE approach seems to have been used most often or even exclusively for assessing risk in existing enterprise information systems. The risk assessment process for a vehicle should cover the entire product lifecycle from the concept phase through production, operation, and maintenance and it should recognize that the vehicle is a mobile cyber physical system and not just an information system.

A.1.5 HEAVENS Security Model

The “HEAVENS Security Model” focuses on methods, processes and tool support for threat analysis and risk assessment with respect to the vehicle Electrical and/or Electronic (E/E) systems (23). The goal is to present a systematic approach of deriving Cybersecurity requirements for the vehicle E/E systems. Also, the results obtained from a proof-of-concept implementation and evaluation of the proposed HEAVENS security model by using an vehicle use case is presented. Please refer to the HEAVENS Deliverable “D2 Security Models” (23) for more information.

We have considered state-of-the-art in threat analysis and risk assessment while developing the HEAVENS security model. The main characteristics of the HEAVENS security model are as follows:

- The proposed model is equally applicable to a wide range of road vehicles, for example, passenger cars and commercial vehicles. The model considers a wide range of stakeholders (e.g., OEM, Fleet owner, Vehicle owner, Driver, Passenger, etc.).
- The model is threat-centric and is realized by applying Microsoft’s STRIDE approach in the context of the vehicle E/E systems.
- The model establishes a direct mapping between security attributes and threats during threat analysis. This facilitates visualizing and making early estimation of the technical impact (confidentiality, integrity, availability) of a particular threat on a particular asset.
- The model maps security objectives (safety, financial, operational, privacy and legislations) with impact level estimation during risk assessment. This assists in understanding the potential business impacts of a particular threat for the relevant stakeholders, for example, OEMs.

- The model provides estimation of impact level parameters (safety, operational, financial, privacy and legislation) based on industry standards. For example, the safety parameter is aligned with the functional safety standard ISO 26262 (24), financial parameter is based on the BSI-Standard (25), operational parameter is based on the Failure Mode and Effect Analysis (FMEA) proposed by the Automotive Industry Action Group (AIAG) (26), and privacy and legislation parameter is connected with “Privacy Impact Assessment Guideline” (27).
- The model is aligned with well-established industry standards and initiatives. For example, Common Criteria for IT Security Evaluation and ISO 26262 for functional safety for road vehicles.

A.1.5.1 Workflow of the HEAVENS Security Model

Figure 23 shows the workflow of the HEAVENS security model. It consists of three components – threat analysis, risk assessment and Cybersecurity requirements.

- Threat Analysis** – Description of the functional use cases (In_01 in the figure) is the input to the threat analysis process. Threat analysis produces two outputs: (a) a mapping between threats and assets (Out_01 in the figure) for each asset in the context of the use case, and (b) a mapping between threats and security attributes (Out_02 in the figure) to establish which security attributes are affected due to a particular threat in the context of an asset.
- Risk Assessment** – Once the threats for the relevant assets are identified, the next step is to rank the threats. This is what is done during risk assessment. The mapping between threats and assets are used as input along with threat level (TL) (In_03 in the figure) and impact level (IL) (In_04 in the figure) parameters. Threat level parameters (**Threat Level (TL)**) and impact level parameters (**Impact Level (IL)**) are presented in A.1.5.1.2). As an end result of risk assessment, the security level (Out_03 in the figure) is identified for each threat associated with each asset of the TOE/use case.
- Security Requirements** – Finally, both the mapping between threat and asset (Out_02 in the figure) as well as security level (Out_03 in the figure) are considered to formulate Cybersecurity requirements for the asset and the TOE. Cybersecurity requirement is a function of asset, threat, security level and security attribute. The derived Cybersecurity requirements are at the level of the functional safety requirements of the ISO 26262 and belong to the concept phase. Later, during product development phase, software Cybersecurity requirements and hardware Cybersecurity requirements [should] be derived based on the high-level Cybersecurity requirements.

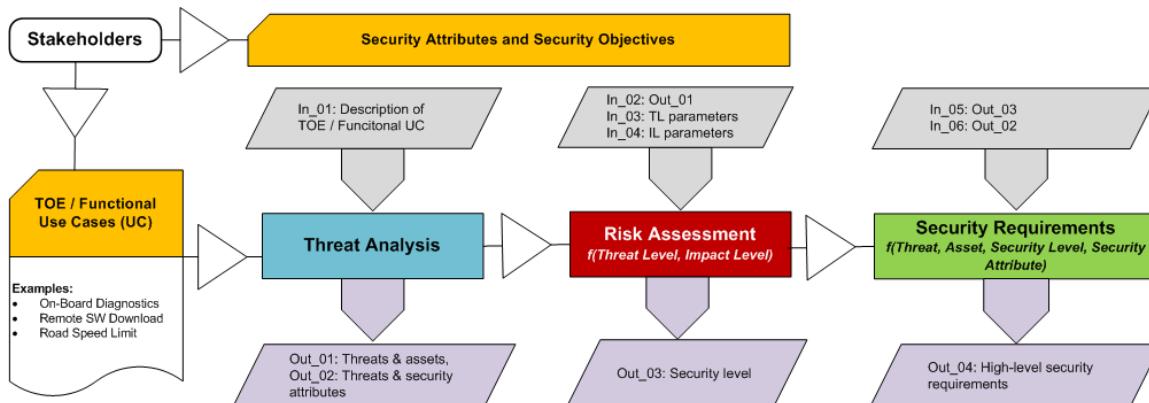


Figure 23 - Workflow of the HEAVENS security model

A.1.5.1.1 Threat analysis

In the HEAVENS security model, threat analysis refers to the identification of the threats associated with the assets of the TOE and mapping of the threats with the security attributes. Microsoft’s STRIDE approach (3) [was adopted] for threat analysis. While STRIDE is a structured and qualitative security approach for discovery and enumeration of threats present in a software system, the applicability of the STRIDE approach [has been extended] to the vehicle E/E systems.

STRIDE provides the opportunity of extending the original CIA model by correlating threats with security attributes (authenticity, integrity, non-repudiation, confidentiality, availability, freshness and authorization). Each category of the STRIDE threats [has been mapped] to a set of security attributes. This mapping is static and is used to formulate Cybersecurity requirements as soon as the security level of a particular threat-asset pair is determined during the risk assessment. The mapping between the STRIDE threats and the security attributes is shown below (Table 9).

Table 9 - Mapping between STRIDE threats and security attributes

STRIDE Threats	Explanation	Security Attribute
Spoofing	attackers pretend to be someone or something else	Authenticity, Freshness
Tampering	attackers change data in transit or in a data store, attackers may change functions as well – implemented in software, firmware or hardware	Integrity
Repudiation	attackers perform actions that cannot be traced back to them	Non-repudiation, Freshness
Information disclosure	attackers get access to data in transit or in a data store	Confidentiality, Privacy
Denial of service	attackers interrupt a system's legitimate operation	Availability
Elevation of privilege	attackers perform actions they are not authorized to perform	Authorization

A.1.5.1.2 Risk Assessment

Risk assessment refers to ranking of the threats. After identifying the threat-asset pairs for a particular use case based on STRIDE approach, the risk assessment to rank the threats proceeds, i.e., to derive security level for each threat-asset pair. Security Level (SL) is a measure of the needed strength of security mechanisms for a security relevant asset to meet a certain level of security. The risks are balanced by usage of security levels for a defined environment including threats and attackers. Risk assessment consists of three steps: (a) determination of threat level (TL): this corresponds to the estimation of the “likelihood” component of risk, (b) determination of impact level (IL): this corresponds to the estimation of the “impact” component of risk, and (c) determination of security level (SL): this corresponds to the final risk rating.

Threat Level (TL) Parameters

- The parameter “**Expertise**” refers to the level of generic knowledge of the underlying principles, product type or attack methods that are required to carry out an attack on the TOE. The identified levels are as follows:
 - ✓ “Layman” is unknowledgeable compared to experts or proficient persons, with no particular expertise; Examples may include persons who can only follow simple instructions that come with the available tools to mount simple attacks, but not capable of making progresses himself/herself if the instructions or the tools do not work as expected.
 - ✓ “Proficient” persons have general knowledge about the security field and are involved in the business, for example, workshop professionals. Proficient persons know about simple and popular attacks. They are capable of mounting attacks, for example, odometer tuning and installing counterfeit parts, by using available tools and if required, are capable of improvising to achieve the desired results.
 - ✓ “Experts” are familiar with the underlying algorithms, protocols, hardware, structures, security behavior, principles and concepts of security employed, techniques and tools for the definition of new attacks, cryptography, classical attacks for the product type, attack methods, etc. implemented in the product or system type.
 - ✓ The level “Multiple Experts” is introduced to allow for a situation, where different fields of expertise are required at an Expert level for distinct steps of an attack.

- The parameter “**Knowledge about TOE**” refers to the availability of information about the TOE and the community size that possesses knowledge about the TOE from an attacker perspective. This parameter points to the sources from where attackers can gain knowledge about the TOE and indicates how easy or difficult it can be for an attacker to acquire knowledge about the TOE. Identified levels are as follows:
 - ✓ “Public” information concerning the TOE (e.g., as gained from the Internet, bookstore, information shared without non-disclosure agreements).
 - ✓ “Restricted” information concerning the TOE (e.g., knowledge that is controlled within the developer organization and shared with other organizations, for example, between suppliers and OEMs, under a non-disclosure agreement). Examples include requirements and design specifications, internal documentation.
 - ✓ “Sensitive” information about the TOE (e.g., knowledge that is shared between discrete teams within the developer organization, access to which is constrained only to members of the specified teams). Examples include restricted ECU configuration parameters to enable/disable features in vehicles, vehicle configuration database, and software source code.
 - ✓ “Critical” information about the TOE (e.g., knowledge that is known by only a few individuals, access to which is very tightly controlled on a strict need to know basis and individual undertaking). Examples include secret root signing key.
- The parameter “**Equipment**” refers to the equipment required to identify or exploit vulnerability and/or mount an attack.
 - ✓ “Standard” equipment is readily available to the attacker, either for the identification of vulnerability or for an attack. This equipment may be a part of the TOE itself (e.g., a debugger in an operating system), or can be readily obtained (e.g., Internet downloads, protocol analyzer or simple attack scripts). Examples include simple OBD diagnostics devices, common IT device such as notebook.
 - ✓ “Specialized” equipment is not readily available to the attacker, but could be acquired without undue effort. This could include purchase of moderate amounts of equipment (e.g., power analysis tools, use of hundreds of PCs linked across the Internet would fall into this category), or development of more extensive attack scripts or programs. Examples include in-vehicle communication devices (e.g., CAN cards), costly workshop diagnosis devices. If clearly different test benches consisting of specialized equipment are required for distinct steps of an attack this would be rated as bespoke.
 - ✓ “Bespoke” equipment is not readily available to the public as it may need to be specially produced (e.g., very sophisticated software), or because the equipment is so specialized that its distribution is controlled, possibly even restricted. Alternatively, the equipment may be very expensive.
 - ✓ The level “Multiple Bespoke” is introduced to allow for a situation, where different types of bespoke equipment are required for distinct steps of an attack.
- The parameter “**Window of opportunity**” combines access type (e.g., logical, physical) and access duration (e.g., unlimited, limited) that are required to mount an attack on the TOE by an attacker. The different levels include:
 - ✓ “Low”: Very low availability of the TOE. Physical access required to perform complex disassembly of vehicle parts to access internals to mount an attack on the TOE.
 - ✓ “Medium”: Low availability of the TOE. Limited physical and/or logical access to the TOE. Physical access to vehicle interior or exterior without using any special tool (e.g., opening the hood to access wires).
 - ✓ “High”: High availability and limited time. Logical or remote access without physical presence.
 - ✓ “Critical”: High availability via public/untrusted network without any time limitation (i.e., TOE/asset is always accessible). Logical or remote access without physical presence and time limitation as well as unlimited physical access to the TOE/asset. Examples include wireless or via Internet (e.g., V2X or cellular interfaces).

Table 10 presents the different parameters and the values to be used for each parameter.

Table 10 - Applying the TL parameters to estimate threat level

Parameter	Value	Explanation
Expertise		
Layman	0	Refer to "Overview of the threat level parameters" (23).
Proficient	1	
Expert	2	
Multiple experts	3	
Knowledge about TOE		
Public	0	Refer to "Overview of the threat level parameters" (23).
Restricted	1	
Sensitive	2	
Critical	3	
Window of Opportunity		
Critical	0	Refer to "Overview of the threat level parameters" (23).
High	1	
Medium	2	
Low	3	
Equipment		
Standard	0	Refer to "Overview of the threat level parameters" (23).
Specialized	1	
Bespoke	2	
Multiple bespokes	3	

Finally, for each threat-asset pair, sum the values of each of the parameters and define ranges to determine a threat level corresponding to each identified range. Five different threat levels (None, Low, Medium, High, and Critical) are adopted as shown in Table 11.

Table 11 - Estimating the threat level (TL)

Summation of the Values of the TL Parameters	Threat Level (TL)	TL Value
> 9	None	0
7 – 9	Low	1
4 – 6	Medium	2
2 – 3	High	3
0 – 1	Critical	4

Impact Level (IL) Parameters

It is a first-order requirement to ensure safety of the vehicle occupants, road users and infrastructures. The “**Safety**” parameter to estimate the safety impact is adopted from the ISO 26262 (28):

- No injury
- Light and moderate injuries
- Severe and life-threatening injuries (survival probable)
- Life-threatening injuries (survival uncertain), fatal injuries

The “**Financial**” category considers all financial losses or damages that can be either direct or indirect. Direct financial damages may include product liability issues (e.g., penalties, recalls), legislation issues (e.g., penalties due to nonconformance), product features (e.g., loss in business due to illicit activation of sellable features). On the other hand, indirect financial damages may include damage to OEM reputation, loss of market share, IP infringement, etc. Also, safety issues may contribute to financial damages. For example, recent recalls of certain models of cars by several OEMs due to various safety issues have financial impact on each of the OEMs. To summarize, the financial damage is the sum of direct and indirect costs for the OEM and the root cause may originate from any of the stakeholders.

The “**Operational**” category includes operational damages caused by unwanted and unexpected changes in (or loss of) a vehicle function. Examples of such operational damages include loss of secondary (e.g., cruise control) and comfort/entertainment (e.g., cd-player, air-conditioning) functionalities of the vehicle. However, in certain situations, operational damages may cause safety and financial damages. For example, operational damages in the form of loss of primary and safety-related vehicle functionalities may affect safety of passengers and road users. Consequently, the impact of the operational category on the overall impact is relatively lower with respect to the safety and financial categories.

The “**Privacy and legislation**” category includes damages caused by privacy violation of stakeholders (e.g., fleet owner, vehicle owner, driver) and/or violation of legislations/regulations (e.g., environmental, driving). Privacy and legislation are merged into one parameter because privacy may be enforced through legislation and there exist legislations that are not related to privacy. Usually, such damages do not have direct injury, financial and operational dimensions. However, in certain situations, privacy and legislation violations may cause financial (e.g., fine, loss of access to certain market) and operational damages to the stakeholders. Consequently, the impact of the privacy and legislation category is relatively lower with respect to the safety and financial categories.

In the HEAVENS model, different weights [are assigned] to the different impact parameters. The “Safety” and “Financial” parameters have equal weights while estimating the overall impact level. The impact of safety and financial parameters can lead to the most severe consequences for stakeholders, for example, vehicle occupants may not survive, organizations may bankrupt. On the other hand, the impact of “Operational” as well as “Privacy and legislation” parameters on the overall impact is relatively lower with respect to the safety and financial damages. To reflect this fact during impact level estimation, reduce the corresponding factors by a magnitude of one in case of operational as well privacy and legislation with respect to the safety and financial parameters. The different safety levels and the corresponding values to estimate the impact of safety is shown in Table 12.

Table 12 - Impact level parameter - safety

Safety	Impact	Value	Explanation
No injury	No impact	0	Part 3 of ISO 26262 (28).
Light and moderate injuries	Low	10	
Severe and life-threatening injuries (survival probable)	Medium	100	
Life-threatening injuries (survival uncertain), fatal injuries	High	1,000	

The categorization of financial damages depends on the financial strength of an individual stakeholder. It may therefore be appropriate to express the limits as percentages of total sales, total profit, or on a similar base value as well as to classify the damages qualitatively into damage categories instead of calculating the damages quantitatively (25). Table 13 suggests one possible categorization of financial damages.

Table 13 - Impact level parameter - financial

BSI-Standard (25)	HEAVENS		Explanation based on BSI-Standard 100-4 (25)
	Damage category	Financial	Value
Low	No impact	0	• No discernible effect. No appreciable consequences.
Normal	Low	10	• The financial damage remains tolerable to the organization and other stakeholders (e.g., fleet owners, drivers).
High	Medium	100	• The resulting damage leads to substantial financial losses to the organization and other stakeholders, but does not threaten the existence of the organization.
Very High	High	1,000	• The financial damage threatens the existence of the organization and severely affects other stakeholders.

We adapt the vehicular defect severity categorization such as FMEA (Failure Mode and Effects Analysis) (26) to classify the operational damages. This is shown in Table 14.

Table 14 - Impact level parameter - operational

Severity of Effect on Product (Effect on Customer) (26)	Effect (26)	Severity Rank (26)	HEAVENS Value
No discernible effect	No effect	1	No Impact (0)
Appearance item or audible noise (vehicle still operates, but does not conform, annoys more than 25% of customers)	Minor disruption	2	Low (1)
Appearance item or audible noise (vehicle still operates, but does not conform, annoys more than 50% of customers)		3	
Appearance item or audible noise (vehicle still operates, but does not conform, annoys more than 75% of customers)	Moderate disruption	4	
Degradation of secondary function (vehicle still operable, but comfort or convenience functions work at a reduced level of performance)	Moderate disruption	5	Medium (10)
Loss of secondary function (vehicle still operable, but comfort or convenience functions do not work)		6	
Degradation of primary function (vehicle still operates, but at a reduced level of performance)	Significant disruption	7	
Loss of primary function (vehicle inoperable, but does not affect safe vehicle operation)	Major disruption	8	High (100)
Potential failure mode affects safe vehicle operation with some warning or noncompliance with government regulations		9	
Potential failure mode affects safe vehicle operation without warning or involves noncompliance with government regulations	Fails to meet safety or regulatory requirements	10	

It has already been mentioned that privacy and legislation category includes the damages caused by privacy violation of stakeholders (e.g., fleet owner, vehicle owner, driver) and/or violation of legislations/regulations (e.g., environmental, driving). Table 15 shows one possible way of assigning different values to this parameter. There is a possibility to align the privacy aspect with the "Privacy Impact Assessment Guideline" provided by BSI, Germany (27).

Table 15 - Impact level parameter - privacy and legislation

Privacy & legislation	Value	Explanation
No impact	0	<ul style="list-style-type: none"> • No discernible effects in relation to violations of privacy and legislation
Low	1	<ul style="list-style-type: none"> • Privacy violations of a particular stakeholder (e.g., vehicle owner, driver) which may not lead to abuses (e.g., impersonation of a victim to perform actions with stolen identities) • Violation of legislations without appreciable consequences for business operations and finance (e.g., warning without any significant financial penalty, limited media coverage) for any stakeholder (e.g., OEM, fleet owner, driver)
Medium	10	<ul style="list-style-type: none"> • Privacy violations of a particular stakeholder (e.g., vehicle owner, driver) leading to abuses (e.g., impersonation of a victim to perform actions with stolen identities) and media coverage • Violation of legislations with potential of consequences for business operations and finance (e.g., financial penalties, loss of market share, media coverage)
High	100	<ul style="list-style-type: none"> • Privacy violation of multiple stakeholders (e.g., fleet owners, multiple vehicle owners and multiple drivers) leading to abuses (e.g., impersonation of a victim to perform actions with stolen identities). Such a level of privacy violation may lead to extensive media coverage as well as severe consequences in terms of loss of market share, business operations, trust, reputation and finance for OEMs and fleet owners • Violation of legislations (e.g., environmental, driver) causing significant consequences for business operations and finance (e.g., huge financial penalties, loss of market share) as well as extensive media coverage

Finally, sum the values of all the impact parameters to estimate the impact level (see Table 16).

Table 16 - Estimating impact level (IL)

Summation of the Values of the Impact Parameters	Impact Level (IL)	IL Value
0	No Impact	0
1 – 19	Low	1
20 – 99	Medium	2
100 – 999	High	3
>= 1,000	Critical	4

Security Level (SL)

In the HEAVENS security model, combine Threat Level (TL) and Impact Level (IL) to derive Security Level as shown in Table 17.

Table 17 - Security level based on threat level and impact level

Security Level (SL)	Impact Level (IL)					
	0	1	2	3	4	
0	QM	QM	QM	QM	Low	
1	QM	Low	Low	Low	Medium	
2	QM	Low	Medium	Medium	High	
3	QM	Low	Medium	High	High	
4	Low	Medium	High	High	Critical	

A.1.5.1.3 Security requirements

The final part of the HEAVENS security model deals with deriving security requirements based on asset, threat, security attribute and security level. Consider the example shown in Table 18. As shown in the third row, the asset "CAN Signal X on Bus A" has a security level "QM". Hence, [there] may not [be a] need to formulate any additional Cybersecurity requirement for this asset to deal with spoofing threat. On the other hand, Cybersecurity requirements should be formulated for the other two cases.

Note that there might be several threats for one asset and as a result, [the analysis] may have multiple security levels based on multiple threat levels for all the threats related to an asset. One approach of determining a security level for the asset as a whole is to consider the highest security level out of all the security levels for all the threats associated with the asset. An alternative is to consider the highest threat level together with the impact level to define the security level for the asset.

Table 18 - Examples of deriving Cybersecurity requirements

No.	Asset	Threat	Security Attribute	Security Level
1	Cryptographic Key	Elevation of Privilege	Authorization	Critical
2	ECU Software	Tampering	Integrity	Medium
3	CAN Signal X on Bus A	Spoofing	Authenticity	QM

A.1.6 Attack Trees

Attack trees were initially described by Schneier (29) and later adapted in the Network-on-Wheels (30) and EVITA projects (19) as a means of vulnerability analysis.

In its most basic form as described by Schneier, an attack tree has an attack goal as the top-level node, and various means (sub-goals) of achieving that goal are explored to develop the “leaves” of the tree in a stepwise and hierarchical manner until base level methods of performing an attack are identified. Sub-goals are combined using AND / OR logic where:

- “OR” logic indicates that any of the sub-goals can achieve its parent goal,
- “AND” logic indicates that all of the sub-goals are needed to achieve their parent goal.

The tree may be analyzed by associating Boolean or continuous values with each sub-goal and propagating these up the tree. In the simplest form Boolean values such as “Possible” or “Impossible” can be assigned to the base attack methods and then propagated up the tree using rules of Boolean algebra to identify all the possible attacks that can achieve the goal. Similar analyses may be performed using continuous numerical values, for example the cost (monetary loss) of an attack, which can then be used to prioritize required actions, for example by identifying all successful attacks that would cost the affected stakeholder more than a certain amount.

Attack trees can be represented in either graphical or textual form; while (29) states a preference for textual representations, in many senses attack trees are entirely analogous to fault trees commonly used in reliability analysis, and FTA tools could be adapted for developing attack trees.

In the EVITA approach to attack trees, a generic structure is proposed consisting of the following levels:

- Level 0: attack goal (analogous to the top event in a fault tree)
- Level 1: attack objectives
- Level 2: attack methods
- Level 3: ($n - 1$): intermediate goals / methods
- Level n : asset attacks (the base level methods of performing an attack; analogous to base events in a fault tree).

It is considered that an asset attack has been identified when a probability of success (or other measure) can be associated with an attack method and this could occur at any level in the tree.

The concepts from the EVITA approach are shown in the following figure (represented using a typical fault tree notation; although dedicated tools are emerging for modeling attack trees):

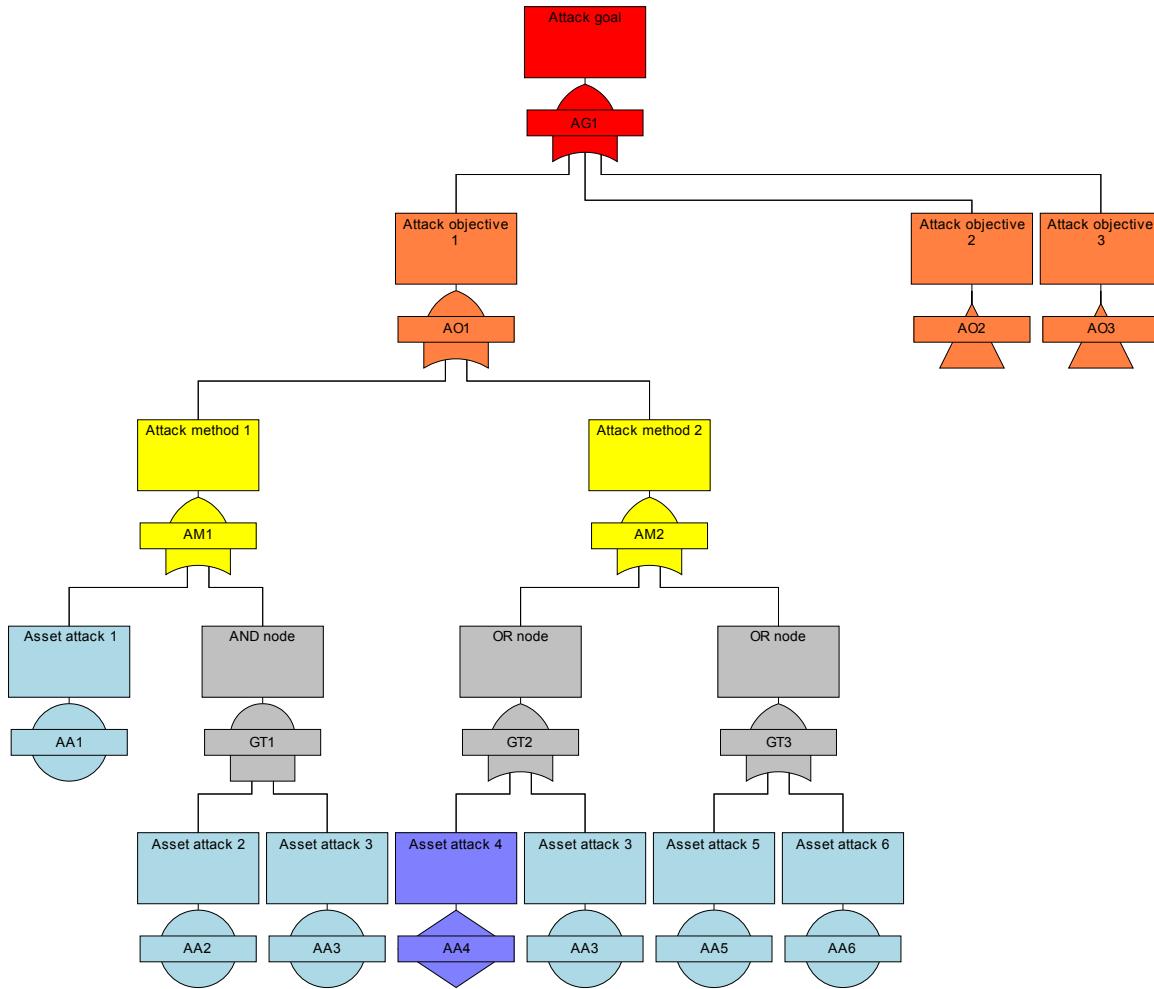


Figure 24 - Generic attack tree

In this example note that:

- The fault tree notation of an “undeveloped event” has been used for AA4 to show part of the tree that needs to be developed further. This could be because additional analysis is needed, or this could be a tree that is further developed elsewhere.
- Asset attack AA3 is common to two of the attack methods and may be an early indication of the need for prioritization, subject to further analysis.
- Asset attacks are not restricted to appearing at a specific level in the attack tree; in the example shown above asset attack AA1 is a direct means of achieving attack method 1, whereas other attack methods may need a combination of asset attacks.

Attack trees may be expanded to differing levels of depth; early on in product development for example, it may only be possible to examine high level concepts; as the design progresses, some of the deeper causes of the attacks could be analyzed.

Specifically during the concept phase, an attack tree can support both severity and probability assessment:

- During evaluation of Severity, the implications of the identified attack objectives for stakeholders can be considered.
- During evaluation of Probability, the analysis can consider attacks (including combinations of attacks) that could contribute to an attack method.

The attack trees can also support:

- Derivation of Cybersecurity requirements through considering use cases and the required Cybersecurity Controls to mitigate the asset attacks.

Prioritization of Cybersecurity Controls – repeated occurrence of particular events or patterns in the attack tree can indicate priorities for action. An example attack tree analysis is provided in Appendix C.

A.1.7 Software Vulnerability Analysis Overview

In software vulnerability analysis there are a number of known software constructs that should be avoided to prevent potential vulnerabilities in the code. Since many of the SW constructs that allow vulnerabilities to exist in the code are known, there have been a number of tools developed to statically analyze the code for use of the undesirable constructs. Many of the tools simply look for the undesirable constructs; however, some tools also add a semantic component to the analysis to increase the likelihood of finding potential vulnerabilities in the code. In addition, there is research being done into analyzing the object code for vulnerabilities.

A.2 OVERVIEW OF CYBERSECURITY TESTING METHODS

In general, the purpose of Penetration Testing is to focus on the highest risk areas identified in the Threat Analysis. The testing should be applied to all external data interfaces (e.g., Bluetooth, USB, cellular, Wi-Fi, ODB2, HMI). This is not intended to be a comprehensive list and this document does not, at this time, recommend specific methods. Therefore, it is up to each organization to determine whether to use one of the methods described below, or whether to use a different method.

A.2.1 Types of Penetration Testing

Generally, there are two primary types of pen tests:

- “**Black Box**” = In Black Box testing there is very little (or no) pre-disclosed information.
- “**White Box**” = In White Box testing the tester has access to all information about the system, such as the results of the vulnerability assessment, access to source code, etc.
- “**Gray Box**” = Gray Box testing is part way between white box and black box testing, with the tester having access to system documentation and algorithms, and knowledge about the internal structure of the system.

For additional information on Penetration Testing, see NIST 800-53, CA-8, Penetration Testing/Independent Penetration Agent or Team (9).

A.2.2 Red Teaming

The Red Team technique was developed by the United States military. With the Red Team approach, a group of internal experts is assembled to be the “**malicious actors**” (or Red Team) who tries to attack the system. The goal is for the Red Team to identify possible attack vectors that could be exploited by malicious or non-malicious actors to gain access to the system and its internal network. Engineering can then work to eliminate these possible attack vectors or mitigate the effect of an attack via an identified attack vector.

A.2.3 Fuzz Testing

Refer to 8.6.10 and Appendix I for information about Fuzz Testing.

APPENDIX B – EXAMPLE TEMPLATES FOR WORK PRODUCTS

Appendix B gives an example of a template for an OCTAVE worksheet as discussed in Appendix A, A.1.4. Appendix B does not yet give examples of the other methods described in Appendix A, and this document does not, at this time, recommend specific methods. Therefore, it is up to each organization to determine whether to use one of the methods described Appendix A, A.1, or whether to use a different method.

B.1 OCTAVE WORKSHEETS

Table 19 - OCTAVE's allegro worksheet 10, information asset risk worksheet

Allegro - Worksheet 10		INFORMATION ASSET RISK WORKSHEET		
Information Asset Risk	Information Asset			
	Area of Concern			
	(1) Actor <i>Who would exploit the area of concern or threat?</i>			
	(2) Means <i>How would the actor do it? What would they do?</i>			
	(3) Motive <i>What is the actor's reason for doing it?</i>			
	(4) Outcome <i>What would be the resulting effect on the information asset?</i>	<input type="checkbox"/> Disclosure	<input type="checkbox"/> Destruction	
		<input type="checkbox"/> Modification	<input type="checkbox"/> Interruption	
	(5) Security Requirements <i>How would the information asset's security requirements be breached?</i>			
	(6) Probability <i>What is the likelihood that this threat scenario could occur?</i>	<input type="checkbox"/> High	<input type="checkbox"/> Medium	<input checked="" type="checkbox"/> Low
Threat	(7) Consequences <i>What are the consequences to the organization or the information asset owner as a result of the outcome and breach of security requirements?</i>	(8) Severity <i>How severe are these consequences to the organization or asset owner by impact area?</i>		
	Impact Area (5 is high)		Value	Score
	Reputation & Customer Confidence – 3			
	Financial – 2			
	Productivity – 1			
	Safety & Health – 5			
	Fines & Legal Penalties – 4			
	User Defined Impact Area			
	Relative Risk Score			

Table 20 - OCTAVE's allegro worksheet 10, risk mitigation section

(9) Risk Mitigation <i>Based on the total score for this risk, what action will you take?</i>			
<input type="checkbox"/> Accept	<input type="checkbox"/> Defer	<input type="checkbox"/> Mitigate	<input type="checkbox"/> Transfer
For the risks that you decide to mitigate, perform the following:			
On what container would you apply controls?	<i>What administrative, technical, and physical controls would you apply on this container? What residual risk would still be accepted by the organization?</i>		

APPENDIX C – EXAMPLES USING IDENTIFIED ANALYSES

Appendix C gives examples of some of the methods that were described in Appendix A, A.1. This is not a comprehensive list and this document does not, at this time, recommend specific methods. Therefore, it is up to each organization to determine whether to use one of the methods described below, or whether to use a different method.

C.1 EXAMPLE OF EVITA APPLICATION AT THE FEATURE LEVEL USING THREAT AND OPERABILITY ANALYSIS (THROP)

Feature: Remote Vehicle Disable

Create Functional Feature Definition: Describe the purpose of the feature, identify the feature's primary functions, and describe the Cybersecurity perimeter.

- **Purpose:** The Remote Vehicle Disable feature is intended to be used by the appropriate authorities to remotely disable a vehicle in the event that a vehicle is stolen, being used in a high-speed chase or other dangerous situation, etc.
- **Example Primary Function:** Remotely disable the vehicle at the request of authorities.

Perform THROP on identified primary functions using guidewords applied to the functions to identify potential threats.

- **Example Potential Threat:** Malicious Intentional Vehicle Disable.

Identify Potential Worst-Case Mishap Scenarios: use brainstorming and expertise to identify multiple potential worst-case mishap scenarios.

- **Example Potential Worst-Case Scenario:** Vehicle is disabled maliciously without the request of authorities leading to unavailability of the vehicle. This is an operational threat, since the vehicle would be unavailable to the driver.

Perform EVITA risk assessment: Once the potential threats and potential worst-case mishap scenarios are identified, the risk of the potential threats would then be analyzed with respect to each potential worst-case mishap scenario, by applying the EVITA risk assessment method. The threat would then be classified according to the highest risk potential mishap scenario. Once all potential threats are classified, the threats can be prioritized based on the determined risk level, such that more detailed analysis can be focused on the highest risk threats.

Table 21 below shows an example spreadsheet that may be used for the example described. The "Threat ID" column in the table is used to provide a unique identifier of each of the high-risk threats that are identified. If a potential threat is determined not to have a risk level that requires further analysis activities to be performed, no "Threat ID" is needed since the potential threat after risk assessment is deemed not a threat. In addition, the table would include a column at the end for "Cybersecurity Goals" (not shown in the table in Figure 20 due to space limitations). If the identified potential threats are deemed true potential threats after the risk assessment, a Cybersecurity goal would be identified for and associated with the potential threat. An example of a Cybersecurity goal for a potential threat of "Malicious Intentional Steering" for a steering assist system may be, "Prevent or mitigate a malicious intentional steering from occurring". This high-level Cybersecurity goal would be transformed into functional and technical Cybersecurity requirements by applying vulnerability analysis to the potential high-level threat to determine potential vulnerabilities that could be exploited and lead to a malicious intentional steering threat being manifested. Identifying the vulnerabilities allows Cybersecurity controls to be identified and implemented to reduce the likelihood of a successful attack leading to the identified potential threat. As previously stated, one cannot guarantee a threat will not be manifest, however, one can reduce the likelihood of the threat being manifested by implementing an appropriate combination of the identified Cybersecurity controls.

Table 21 - Example spreadsheet of EVITA risk assessment at feature level

Item: Remote Vehicle Disable				Potential Worst-Case Threat Scenario	Severity		Attack Potential				AP _{Totl}	Attack Prob.	Cs	Risk				
Threat ID	Function	Potential Item Threats	Potential Vehicle Level Threat		F	O	P	S	Elpsd Time	Expert	Knw	WofO	Eqp	F	O	P	S	
	Remotely disable vehicle at request of authorities	Malicious Intentional Vehicle Disable	Malicious Intentional Loss of Ability to Start Vehicle	Vehicle is disabled maliciously without request of authorities and driver is unable to start the vehicle	1	2	0	1	10	3	3	10	7	33	1	1	R0	RNRA0
		Malicious Loss of Remote Vehicle Disable Capability	Malicious Loss of ability to Stop Vehicle for easy Recovery															

C.1.1 OCTAVE Worksheet Analysis Example

One key worksheet from the OCTAVE Allegro streamlined process is provided here as an example of how a worksheet might appear if completed by an automaker or rental car agency for a hypothetical threat use case. All of the OCTAVE worksheets were completed outside this document for this use case example, but only worksheet 10 is shown here for brevity. Worksheet 10 pulls together information that is produced in the process of filling out OCTAVE worksheets 1 through 9. In the use case a disgruntled airport rental car employee legally rents a vehicle for several weeks and downloads data from the OBD-II port and/or CAN bus in order to learn specifics about the CAN data packets for that make and model of car. The disgruntled employee designs malware that can be installed by flashing the Engine Control Module via a laptop connected into the CAN bus via the OBD II port. The flashed malware contains a trigger that activates it on a specified date. Once activated the malware sends a packet to the ECM that kills the engine and causes the engine to knock on restart. Once the malware is proven to work, the employee installs it on the rental car fleet.

Table 22 - Example of OCTAVE's allegro worksheet 10, information asset risk worksheet - ECU firmware

Allegro - Worksheet 10		INFORMATION ASSET RISK WORKSHEET																						
Information Asset Risk	Information Asset	ECU firmware																						
	Area of Concern	Firmware (integrity and confidentiality) could be compromised.																						
	(1) Actor <i>Who would exploit the area of concern or threat?</i>	Renter, disgruntled employee, external attacker intending to harm the renter, rental agency, and/or automaker.																						
	(2) Means <i>How would the actor do it? What would they do?</i>	Actor would need to obtain firmware from automaker (possibly from an insider) or pull it from ECU and reverse engineer it. Modify firmware and test for effect on one vehicle, and upload new firmware to multiple vehicles.																						
	(3) Motive <i>What is the actor's reason for doing it?</i>	Collect insurance from a staged accident, harm reputation, blackmail, cause physical harm, create panic, get publicity, or actor simply enjoys a challenge.																						
	(4) Outcome <i>What would be the resulting effect on the information asset?</i>	<input checked="" type="checkbox"/> Disclosure <input checked="" type="checkbox"/> Destruction <input checked="" type="checkbox"/> Modification <input type="checkbox"/> Interruption																						
	(5) Security Requirements <i>How would the information asset's security requirements be breached?</i>	Access to firmware could compromise confidentiality and modification would compromise integrity. System malfunction would compromise system availability.																						
	(6) Probability <i>What is the likelihood that this threat scenario could occur?</i>	<input type="checkbox"/> High	<input type="checkbox"/> Medium	<input checked="" type="checkbox"/> Low																				
	(7) Consequences <i>What are the consequences to the organization or the information asset owner as a result of the outcome and breach of security requirements?</i>	(8) Severity <i>How severe are these consequences to the organization or asset owner by impact area?</i> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Impact Area (5 is high)</th> <th style="text-align: center;">Value</th> <th style="text-align: center;">Score</th> </tr> </thead> <tbody> <tr> <td>Reputation & Customer Confidence – 3</td> <td style="text-align: center;">Medium</td> <td style="text-align: center;">6</td> </tr> <tr> <td>Financial – 2</td> <td style="text-align: center;">Medium</td> <td style="text-align: center;">4</td> </tr> <tr> <td>Productivity – 1</td> <td style="text-align: center;">Medium</td> <td style="text-align: center;">2</td> </tr> <tr> <td>Safety & Health – 5</td> <td style="text-align: center;">High</td> <td style="text-align: center;">15</td> </tr> <tr> <td>Fines & Legal Penalties – 4</td> <td style="text-align: center;">Low</td> <td style="text-align: center;">4</td> </tr> <tr> <td>User Defined Impact Area</td> <td></td> <td></td> </tr> </tbody> </table>			Impact Area (5 is high)	Value	Score	Reputation & Customer Confidence – 3	Medium	6	Financial – 2	Medium	4	Productivity – 1	Medium	2	Safety & Health – 5	High	15	Fines & Legal Penalties – 4	Low	4	User Defined Impact Area	
Impact Area (5 is high)	Value	Score																						
Reputation & Customer Confidence – 3	Medium	6																						
Financial – 2	Medium	4																						
Productivity – 1	Medium	2																						
Safety & Health – 5	High	15																						
Fines & Legal Penalties – 4	Low	4																						
User Defined Impact Area																								
The rental agency or automaker's reputation is tarnished because the incident becomes a news story. Rental agency has cars that can't be rented until problem is fixed. Recall may be required.																								
Rental agency employees must work around the shortage of cars. Automakers should investigate attack to determine whether this is a bigger problem that could trigger a recall.																								
Some drivers may be injured. Some drivers may sue rental agency or automaker for injuries or negligence.																								
Relative Risk Score				31																				

Table 23 - Example of OCTAVE's allegro worksheet 10, risk mitigation section – ECU firmware

(9) Risk Mitigation			
<i>Based on the total score for this risk, what action will you take?</i>			
<input type="checkbox"/> Accept	<input type="checkbox"/> Defer	<input checked="" type="checkbox"/> Mitigate	<input checked="" type="checkbox"/> Transfer
For the risks that you decide to mitigate, perform the following:			
On what container would you apply controls?	What administrative, technical, and physical controls would you apply on this container? What residual risk would still be accepted by the organization?		
ECU	It may be that the attack could have been mounted against any make or model. The rental agency is poorly equipped to set up technical controls and monitor the integrity of firmware on individual vehicles may choose to accept ECUs can be attacked. The agency may try to buy cyber-attack insurance. The automaker may set up technical and physical controls that prevent easy access to safety and Cybersecurity critical information and functionality, however, the automaker may determine that it isn't cost effective to prevent all ECU attacks.		
Human Employee	The rental agency and automaker may decide to do more extensive background checks on employees in addition to annual re-investigations. The rental agency and automaker can initiate a personnel or Cybersecurity training program to help identify and cope with disgruntled employees or insider threats and to help identify suspicious employee activity.		
Car	The rental agency accepts that their cars can be attacked and obviously can't watch cars once they are driven off the lot. When cars are on the lot, security cameras can be used to watch for suspicious activity. The automaker can install technical and physical controls that prevent and/or monitor access and unusual activity, but cannot prevent all forms of attack.		

C.2 ATTACK TREE EXAMPLE

As an example, an attack tree has been developed for the potential feature threat “Malicious intentional vehicle disable” (see THROP example Figure 25).

In respect of this example it should be noted that it is necessarily incomplete and to illustrate the principles only.

It is assumed that a definition of the intended function already exists, for example through specification of use cases. In this example it is assumed that a vehicle is equipped with a remote disable facility that the owner can use to remotely prevent the vehicle from being started, and which law enforcement authorities could potentially also use to shut down a vehicle in motion.

It is assumed that the function is implemented through a remote service center which can receive a request from the vehicle owner or law enforcement associated with a unique identifier (e.g., a VIN) and which then sends a command wirelessly to the vehicle. The vehicle is equipped with an interface ECU which receives wireless commands, authenticates them and sends resulting commands or information to internal vehicle systems.

Development of the attack tree starts by considering the potential attacker(s) and their motivations. In this case a generic threat “Malicious intentional vehicle disable” has been specified and this forms the “attack goal” at the head of the tree.

Development of the tree then proceeds by considering the “attack objectives” i.e., the different means by which an attacker may achieve this overall goal. In this example two attack objectives have been identified - malicious remote disable of the vehicle and malicious disable of starting. It should be noted at this point that while many of the attack methods and asset attacks will be found to be common to these attack objectives, the severity of the outcome may be different depending on the context. For example, malicious disable of starting of an individual vehicle is unlikely to have a safety-relevant outcome.

The next stage of the tree considers the “attack methods” i.e., the methods or techniques that an attacker can use to achieve the attack objectives. These may be further refined in an analogous method to a fault tree until “asset attacks” are identified – these represent the base level methods of performing an attack by exploiting vulnerabilities and are analogous to base events in a fault tree. At early stages (e.g., functional Cybersecurity concept development) these are likely to be relatively high level but can be further elaborated in later stages of the design.

For example, an objective of “malicious remote disable” may be achieved by attacking the service center to generate a malicious command, or by attacking the communications channel. In the case of attacking the communications channel, this is developed with further methods e.g., to inject a malicious message or attack the interface in the vehicle.

In the case of a malicious message it is assumed that the attacker will need two successful asset attacks – exploiting keys AND generating a malicious message. This is represented by using an “AND” gate in the attack tree.

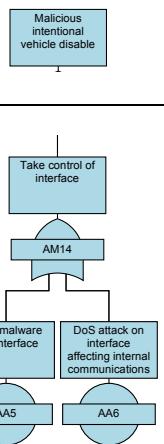
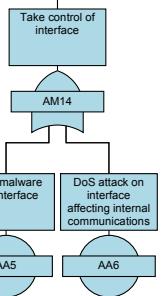
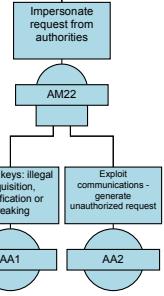
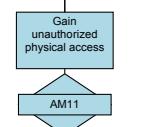
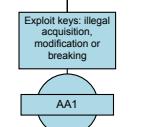
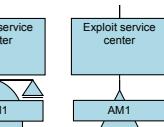
In the case of attacking the interface in the vehicle, the asset attacks shown are intentionally high level and would be developed further as details of the design become known; e.g., to inject malware so that the attacker can execute their own commands.

It should be noted that in this example the attack methods “Exploit service center” and “Exploit remote communications” are common to both attack objectives but this may not be the case in general. A further variation on this is that there are, for example, two asset attacks described as “Exploit keys: illegal acquisition, modification or breaking” but with unique identifiers; this is to reflect that the probability of a successful attack may be different depending on the context (e.g., it may be easier to exploit one type of key compared to another type of key).

Finally it should be noted that attack methods outside the realms of Cybersecurity can be identified, for example, gaining physical access to a service center or social engineering of its operators. These can be included in the attack tree but not developed further. An organization’s Cybersecurity policy or project Cybersecurity plan will typically state the approach to be taken.

An example of an attack tree structure for this attack goal is shown below. Note that this is drawn using a typical fault tree tool and its notation (Isograph Reliability Workbench – incorporating FaultTree+), although dedicated software packages for constructing attack trees are available. Please note that some of the notation used in this example may be different in other tools:

Table 24 - Example of attack tree structure for “malicious intentional vehicle disable”

The attack goal, attack objectives and attack methods are denoted using the fault tree notation for an event (rectangle).	
Events are combined using gates – an OR gate represents that the event can be caused by one or more of the preceding events. In terms of attack trees, then a particular attack method can be achieved by exploiting one or more of the specified asset attacks. In the example shown, an attack method taking control of the vehicle interface may use a malware exploit to permit an attacker to have direct control of the interface, or a means (not specified in this example) of using the interface to mount a denial of service attack on internal communications within the vehicle.	
An AND gate represents that the event can only be caused by all of the preceding events occurring together. In terms of attack trees, then a particular attack method can only be achieved by exploiting all of the specified asset attacks. In the example shown, an attack method that impersonates a request requires both exploiting keys and generating an unauthorized request.	
The fault tree notation for an “undeveloped event” (diamond shape) is used for an attack method that is outside the scope of Cybersecurity activities e.g., AM11 “gain unauthorized physical access” or that is within scope but that requires further development using typical IT security e.g., AM21 “inject malware into control center”.	
The fault tree notation for a “base event” (circle) is used for asset attacks, although these may be developed further in subsequent analyses as part of the functional and technical Cybersecurity concepts.	
Where parts of the tree structure are common to different branches of the tree, then the transfer notation (triangle) is used. This means that the same probabilities of a successful attack are inherited in these branches of the tree. If different probabilities are necessary, then use a separate instance of the attack methods / asset attacks (e.g., AM13 vs AM22 in this example).	

Generally speaking, tools will label the nodes in the tree automatically; in this example the following numbering scheme is in use:

- AG1 refers to an attack goal. Typically there will be multiple attack goals associated with a feature, and separate attack trees will be developed for each attack goal.
- AO1, AO2, etc. refer to attack objectives.
- AM1, AM2, etc. refer to attack methods. The choice of a new numbering sequence for each level in the hierarchy in this example (AM1, ...; AM11 ...; AM21 ...) has been done purely for clarity.
- AA1, AA2, etc., refer to asset attacks.

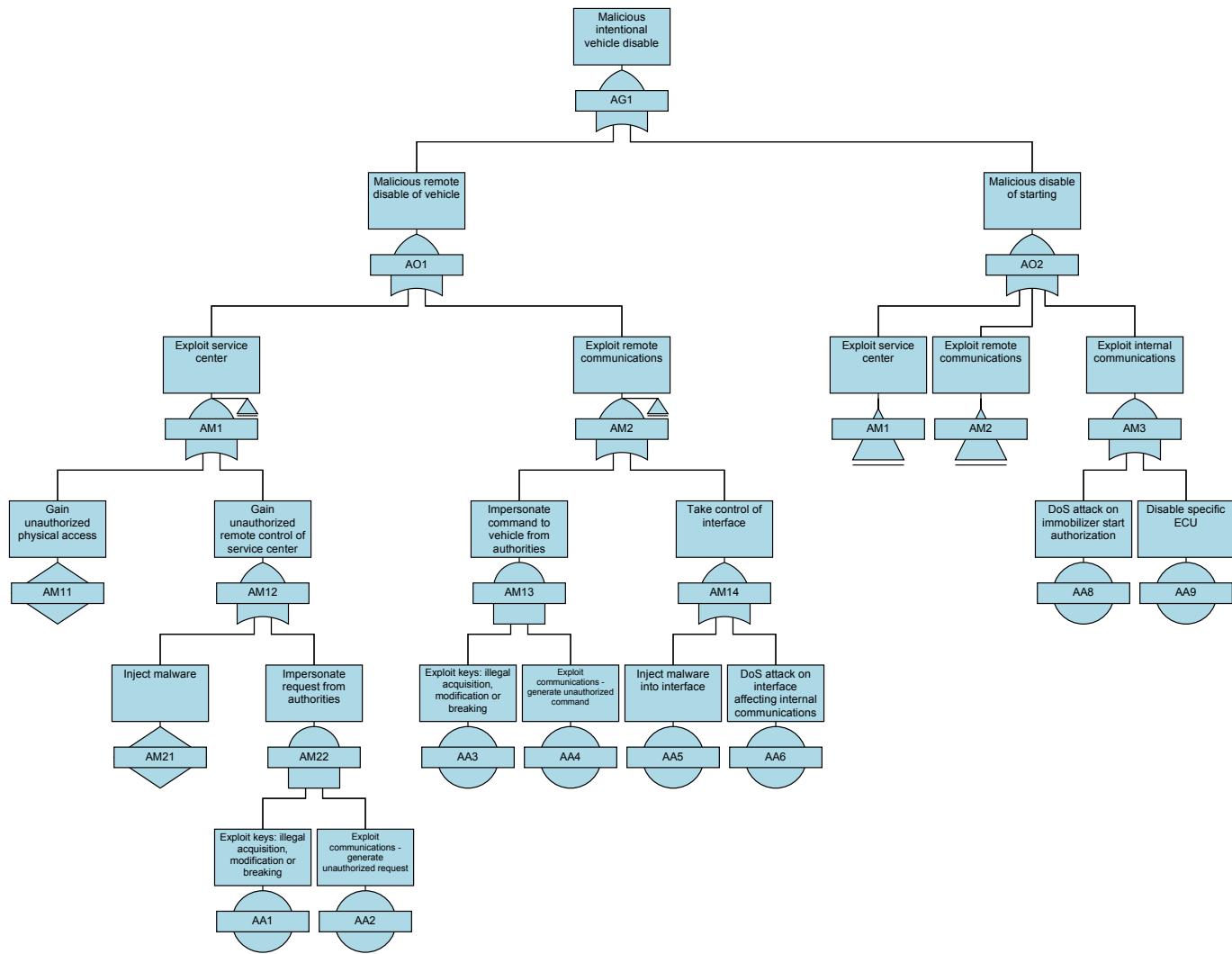


Figure 25 - Attack tree

C.3 EXAMPLE FROM HEAVENS SECURITY MODEL

This section serves as proof-of-concept implementation of the HEAVENS security model. Preliminary results [are presented for] threat analysis and risk assessment based on an vehicle use case. On-board diagnostics (OBD) is a very common use case within vehicles today. A vehicle will perform its own diagnostics and reporting if it detects it is in a faulty state. In order to do this, the vehicular system has what is called on-board diagnostics (OBD). The system basically has the ability to use its instrument cluster to request and present information. This is very useful in various situations such as requesting and presenting diagnostic trouble codes, software identification for the affected ECUs, etc. The main difference between this, wired diagnostics and remote diagnostics is that no diagnostics tool is needed to complete the process. Everything is done within the vehicular system.

C.3.1 HEAVENS Threat Analysis Example

We have started threat analysis activities based on the high-level operational description of the on-board diagnostics (OBD) use case. A DFD [is created] as shown in Figure 26.

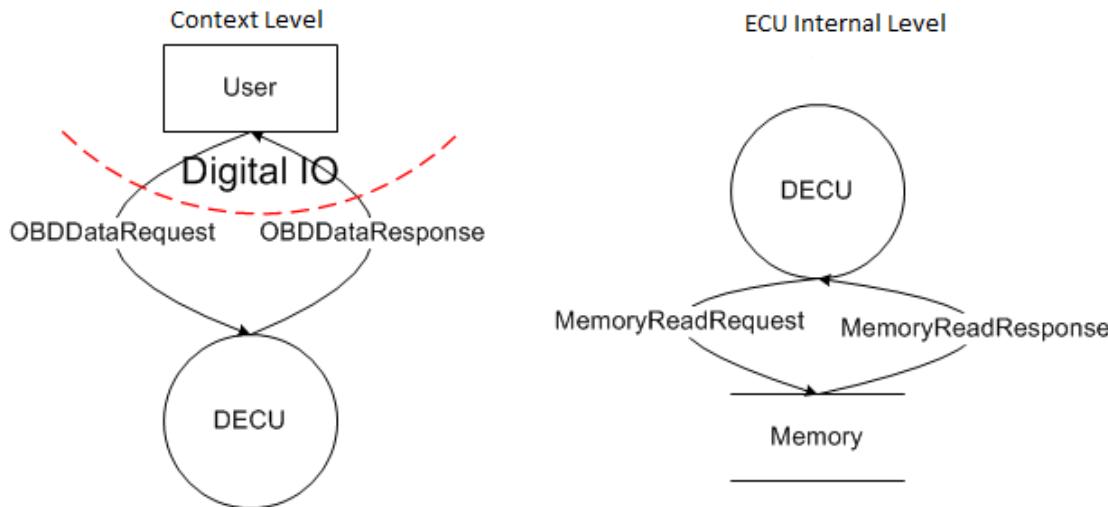


Figure 26 - Data flow diagram of on-board diagnostics (OBD) use case

The DFD consists of two different abstraction levels that are both shown in the same figure. Once the DFD is completed and no validation error is found, use the tool to analyze the DFD and to automatically generate the threats associated with the assets of the OBD use case. An extract from the identified threats are shown in Table 25.

Table 25 - Threats associated with the OBD use case

Element Name	Threat Type
MemoryReadRequest (DECU to Memory)	Tampering
MemoryReadRequest (DECU to Memory)	InformationDisclosure
MemoryReadRequest (DECU to Memory)	DenialOfService
MemoryReadResponse (Memory to DECU)	Tampering
MemoryReadResponse (Memory to DECU)	InformationDisclosure
MemoryReadResponse (Memory to DECU)	DenialOfService
OBDDataRequest (User to DECU)	Tampering
OBDDataRequest (User to DECU)	InformationDisclosure
OBDDataRequest (User to DECU)	DenialOfService
OBDDataResponse (DECU to User)	Tampering
OBDDataResponse (DECU to User)	InformationDisclosure
OBDDataResponse (DECU to User)	DenialOfService
Memory	Tampering
Memory	Repudiation
Memory	InformationDisclosure
Memory	DenialOfService
User	Spoofing
User	Repudiation
DECU	Spoofing
DECU	Tampering
DECU	Repudiation
DECU	InformationDisclosure
DECU	DenialOfService
DECU	ElevationOfPrivilege

C.3.2 Risk Assessment Example from HEAVENS Method

Table 26 shows an extract from the results of HEAVENS Risk Assessment Methodology for the OBD use case. During [the] analysis, [the result is a] "Low" for threat level and "Medium" for impact level for each asset-threat pair (see Table 26). This leads to a security level "Low" as per the analysis.

Table 26 - Risk rating of the OBD use case based on the HEAVENS methodology

Name	Asset	Threat	Risk
Name	Asset	Threat	HEAVENS
Tamper DECU to provide wrong data	DECU	Tampering	Low
Spoof the OBD Response	OBDDataResponse	Spoofing	Low
Block the OBD request to the DECU	OBDDataRequest	Denial of service	Low

C.3.3 Cybersecurity Requirements Example from HEAVENS Method

A mapping [is established] across asset, threat, security attribute, and security level for each of the asset-threat pair of the OBD use case as shown in Table . Derive a Cybersecurity requirement for each row of the table. Currently, the security level [is not considered] while deriving high-level Cybersecurity requirements. However, security level is expected to be considered to estimate the required level of strength and protection while developing Cybersecurity Controls to fulfill the derived Cybersecurity requirements to estimate the required

Table 27 - Asset, threat, security attribute and security level for the OBD use case

No.	Asset	Threat	Security Attribute	Security Level
1	DECU	Tampering	Integrity	Low
2	OBDDataResponse	Spoofing	Authenticity	Low
3	OBDDataRequest	Denial of service	Availability	Low

Security Requirement 1

The DECU shall ensure integrity of the stored data.

Security Requirement 2

The authenticity of the OBDDataResponse signal shall be ensured.

Security Requirement 3

The authorized users shall be able to use the OBDDataRequest signal to extract information from the DECU whenever needed.

APPENDIX D – SECURITY & PRIVACY CONTROLS DESCRIPTION AND APPLICATION

This appendix lists a sample set of 14 security control families and 5 privacy control families and a few controls within each family that might be applicable for vehicle system security. The environmental scope of coverage includes design, manufacturing, customer operation, maintenance, and disposal. The family names and control names/labels were derived from NIST SP 800-53, *Security and Privacy Controls for Federal Information Systems and Organizations*⁷ which lists 17 control families and 240 security and privacy controls for protecting information assets from security threats (9). Though NIST SP 800-53 was developed to produce a unified information security framework for the US federal government, commercial companies with information security needs are also a target audience.

NIST SP 800-53 describes two kinds of control customization processes (tailoring and overlays) which may be used to modify an existing baseline control list to make it more applicable, when needed. Tailoring is the process of customizing a baseline set of controls to achieve a more focused and relevant security capability for an organization. Baseline sets are recommended collections of controls for low, medium, and high impact information systems, as defined by FIPS 200. An overlay is a specialized list of controls that addresses the specialized requirements, technologies, or unique environments of a community of interest such as the transportation industry. This appendix could be seen as a small sample of what an overlay developed for the vehicle industry could provide. An overlay is a fully specified set of security controls, enhancements, and supplemental guidance. An overlay may be derived from an existing security control baseline, if an appropriate baseline exists. Existing baselines, which exist primarily for information systems rather than cyber-physical vehicle systems, probably overlook key assumptions or may be based on false assumptions, and a specific overlay for vehicle systems would remedy this.

An overlay may be either abstract in order to be applicable to a large class of systems in different environments or it may be specific with respect to the system hardware, firmware, and software and the environment in which the system operates. In addition, a general vehicle industry overlay could provide tailoring guidance to address specialized requirements, business functions, technologies, or operational environments for individual automakers. An overlay for the transportation industry, which does not exist at the time of this writing, would probably be too broad to meet the specific needs of the vehicle industry. The creation of a full formal overlay template for the vehicle industry could be a useful parallel effort or follow-on effort to this SAE Recommended Practice. In addition to describing specific vehicle security controls, it would be useful to provide guidance on the application of controls to specific technologies and in different operating environments. NIST SP 800-53 describes a sample overlay template with the following sections: 1.) Identification, 2.) Overlay Characteristics, 3.) Applicability, 4.) Overlay Summary, 5.) Detailed Overlay Control Specifications, 6.) Tailoring Considerations, 7.) Definitions and Additional Information or Instructions.

The sample set of 14 security control families (that may be applicable to the vehicle industry) and their associated controls are outlined in Table 28. Short control descriptions have been drafted for a tiny subset of the controls as examples for the reader. The task of creating a full formal overlay template for the vehicle industry with all associated documentation is beyond the scope of this appendix.

Table 28 - Sample list of potential security control families & controls for vehicle industry

NIST SP 800-53 ID	Control Family and Control Name	Control Description
	Control Family: Access Control	
AC-1	Access Control Policy and Procedures	Specifies policies and procedures for all access controls. Should define who should have access to sensitive information, hardware, and / or software. This may include design files, manufacturing systems, privileged vehicle operations, and safety and security critical internal components.
AC-14	Permitted Actions without Identification or Authentication	Identifies actions that should be permitted without requiring a user to identify and authenticate. This may include operating the hood, trunk, or gas cap release or performing emergency procedures when safety is an issue.
AC-17	Remote Access	Specifies what should be remotely accessible, who is permitted access and why. The control should address remote access methods, encryption of remote access links, remote access monitoring, and remote access entry points.
AC-18	Wireless Access	Specifies what should be accessible via wireless and who or what is permitted access and why. Remote access and wireless access will be similar or the same for the case of a vehicle while remote access for design, manufacturing, and maintenance systems may be wired or wireless. Wireless access control should address access methods, encryption of links, entry points, and applicable wireless technologies / protocols. There may be some overlap here with controls specified under other families.
AC-19	Access Control for Mobile Devices	Specifies what should be accessible from a mobile device and who or what is permitted access and why. Once connected, mobile devices could send malware or exfiltrate private or proprietary data from design, manufacturing, vehicle and maintenance systems. A smart phone mobile device locally linked may be controllable from nearly any location on Earth via the cellular phone network. There may be some overlap here with controls specified under other families.
	Control Family: Awareness and Training	
AT-1	Security Awareness Training and Education Policy and Procedures	Specifies policies and procedures for conducting security awareness training and education. Should specify periodicity of training; training goals by role or job category; security training due to legal or regulatory issues; privacy training if applicable; security as part of lifecycle process including design, coding, hardware and component specification, manufacturing, testing, maintenance, and disposal; threat analysis and risk assessment.

Table 28 - Sample list of potential security control families & controls for vehicle industry (cont.)

	Control Family: Audit and Accountability	
AU-1	Audit and Accountability Policy and Procedures	Control description to be created as part of a follow-on task.
AU-2	Audit Events	Control description to be created as part of a follow-on task.
AU-6	Audit Review, Analysis, and Reporting	Control description to be created as part of a follow-on task.
	Control Family: Security Assessment and Authorization	
CA-1	Security Assessment and Authorization Policies and Procedures	Control description to be created as part of a follow-on task.
CA-2	Security Assessments	Control description to be created as part of a follow-on task.
CA-7	Continuous Monitoring	Control description to be created as part of a follow-on task.
CA-8	Penetration Testing	Control description to be created as part of a follow-on task.
	Control Family: Configuration Management	
CM-1	Configuration Management Policy and Procedures	Control description to be created as part of a follow-on task.
CM-3	Configuration Change Control	Control description to be created as part of a follow-on task.
CM-4	Security Impact Analysis	Control description to be created as part of a follow-on task.
	Control Family: Identification and Authentication	
IA-1	Identification and Authentication Policy and Procedures	Control description to be created as part of a follow-on task.
IA-3	Device Identification and Authentication	Control description to be created as part of a follow-on task.
IA-4	Identifier Management	Control description to be created as part of a follow-on task.
IA-5	Authenticator Management	Control description to be created as part of a follow-on task.
IA-9	Service Identification and Authentication	Control description to be created as part of a follow-on task.

Table 28 - Sample list of potential security control families & controls for vehicle industry (cont.)

	Control Family: Incident Response	
IR-1	Incident Response Policy and Procedures	Control description to be created as part of a follow-on task.
IR-4	Incident Handling	Control description to be created as part of a follow-on task.
IR-5	Incident Monitoring	Control description to be created as part of a follow-on task.
IR-6	Incident Reporting	Control description to be created as part of a follow-on task.
	Control Family: Media Protection	
MP-1	Media Protection Policy and Procedures	Control description to be created as part of a follow-on task.
MP-6	Media Sanitization	Control description to be created as part of a follow-on task.
	Control Family: Physical and Environmental Protection	
PE-1	Physical and Environmental Protection Policy and Procedures	Control description to be created as part of a follow-on task.
PE-4	Access Control for Transmission Medium	Control description to be created as part of a follow-on task.
PE-6	Monitoring Physical Access	Control description to be created as part of a follow-on task.
	Control Family: Planning	
PL-1	Security Planning Policy and Procedures	Control description to be created as part of a follow-on task.
	Control Family: Risk Assessment	
RA-1	Risk Assessment Policy and Procedures	Control description to be created as part of a follow-on task.
RA-3	Risk Assessment	Control description to be created as part of a follow-on task.
RA-5	Vulnerability Scanning	Control description to be created as part of a follow-on task.
	Control Family: System and Services Acquisition	
SA-1	System and Services Acquisition Policy and Procedures	Control description to be created as part of a follow-on task.
SA-11	Developer Security Testing and Evaluation	Control description to be created as part of a follow-on task.
SA-12	Supply Chain Protection	Control description to be created as part of a follow-on task.
SA-18	Tamper Resistance and Detection	Control description to be created as part of a follow-on task.
SA-19	Component Authenticity	Control description to be created as part of a follow-on task.

Table 28 - Sample list of potential security control families & controls for vehicle industry (cont.)

	Control Family: System and Communications Protection	
SC-1	System and Communications Protection Policy and Procedures	Control description to be created as part of a follow-on task.
SC-5	Denial of Service Protection	Control description to be created as part of a follow-on task.
SC-7	Boundary Protection	Control description to be created as part of a follow-on task.
SC-8	Transmission Confidentiality and Integrity	Control description to be created as part of a follow-on task.
SC-13	Cryptographic Protection	Control description to be created as part of a follow-on task.
SC-24	Fail in Known State	Control description to be created as part of a follow-on task.
SC-28	Protection of Information at Rest	Control description to be created as part of a follow-on task.
SC-40	Wireless Link Protection	Control description to be created as part of a follow-on task.
	Control Family: System and Information Integrity	
SI-1	System and Information Integrity Policy and Procedures	Control description to be created as part of a follow-on task.
SI-3	Malicious Code Protection	Control description to be created as part of a follow-on task.
SI-4	Information System Monitoring	Control description to be created as part of a follow-on task.
SI-5	Security Alerts, Advisories, and Directives	Control description to be created as part of a follow-on task.
SI-7	Software, Firmware, and Information Integrity	Control description to be created as part of a follow-on task.
SI-16	Memory Protection	Control description to be created as part of a follow-on task.

The privacy controls in the table below are specifically targeted at vehicle systems and components, including operation, maintenance, and disposal and are in addition to an already established company privacy program.

Table 29 - Sample list of potential privacy control families & controls for vehicle industry

NIST SP 800-53 ID	Control Family and Control Name	Control Description
	Control Family: Authority and Purpose	
AP-2	Purpose Specification	Describes any privacy information that will be maintained within the vehicle and why.
	Control Family: Accountability, Audit, and Risk Management	
AR-2	Privacy Impact and Risk Assessment	Performs a privacy impact and risk assessment from the collection, storage, and disposal of any privacy information used in the vehicle.
AR-3	Privacy Requirements for Contractors and Service Providers	Establishes privacy responsibilities and access requirements for maintainers and service providers.
AR-5	Privacy Awareness and Training	Conducts regular privacy training for all employees who may have access to privacy information stored in a vehicle component or tracked using backend systems which support vehicle operation, maintenance, and disposal.
	Control Family: Data Minimization and Retention	
DM-2	Data Retention and Disposal	Destroys or anonymizes privacy information stored in a vehicle component to prevent loss, theft, misuse, or unauthorized access.
	Control Family: Individual Participation and Redress	
IP-1	Consent	Provides means for customers to understand the consequences of data collection and the means for customers to accept or decline the collection and use of their private information within the vehicle.
IP-2	Individual Access	Provides means for customers to access or request access to their private information which is stored within the vehicle.
IP-4	Control Family: Security	
SE-1	Inventory of Personally Identifiable Information	Maintains an inventory of programs and vehicle systems that collect, use, and/or maintain privacy information.

APPENDIX E – VULNERABILITY DATABASES AND VULNERABILITY CLASSIFICATION SCHEMES

Vulnerabilities are generally understood to be ‘weaknesses or inherent properties of systems that expose the systems or applications to the influence of a threat’. Some authors use the term ‘vulnerability’ in the same sense as the term ‘attack’. Often, the terms weakness, vulnerability and attack are used interchangeably.

Before vulnerability data bases and classification schemes can be discussed, it is necessary to discuss the generally accepted meanings of some related terms – which are frequently used interchangeably.

- **Error/Fault**

A deviation from correct behavior is an *error*. The adjudged or hypothesized cause of an error is called a *fault* (31). An example of a fault could be a statement to copy input data of unchecked length to a buffer of fixed length in C/C++.

- **Weakness**

A weakness is a “reason” for a vulnerability. Generally, weaknesses are abstract formulations of security problems. The formulation of weaknesses varies from very generic ones, e.g., “*Missing Encryption of Sensitive Data*”, to quite concrete ones, e.g., “*Uncontrolled Format String*” which could be a generic weakness with ‘C’ language programs. There could be several sources leading to a weakness. The estimation of severity of a weakness is complicated, due to the missing information on the environment.

- **Vulnerability**

All faults do not lead to vulnerabilities. Vulnerabilities are some internal faults that enable an external threat to harm the system leading to security consequences (31). The term vulnerability is used to indicate those faults and particular conditions generated by them that could be exploited in the implementation of an attack. For e.g., the existence of a missing input validation fault leads to a buffer overflow at an appropriate position in the stack for inputs of certain lengths, which could be effectively utilized to inject malicious code of required characteristics. The system/application is then known to have a ‘Buffer overflow’ vulnerability. Sometimes, the term vulnerability is also used to indicate a security problem in an application or in a set of applications, for e.g., usage of an affected shared library. All vulnerabilities need not be exploited. Vulnerabilities are typically abstracted from concrete exploits, hence sometimes; there is no clear distinction that could be drawn between vulnerabilities and weaknesses. The likelihood of exploitation of a vulnerability can be estimated, since it is known how complicated it is to come up with a working exploit. The severity could also be assessed.

- **Exploit**

An exploit is an implementation and or method for the deployment of an attack. An exploit could be packaged as a software program, or it could simply be data of some length. An exploit is typically related to a concrete application or a set of applications, i.e., if they share a common vulnerability. It is easy to assess the severity of an exploit, because the consequences of a successful attack are known and it is also known how many systems are affected.

Not all faults lead to weaknesses. Not all weaknesses lead to vulnerabilities and not all vulnerabilities lead to a useable exploit.

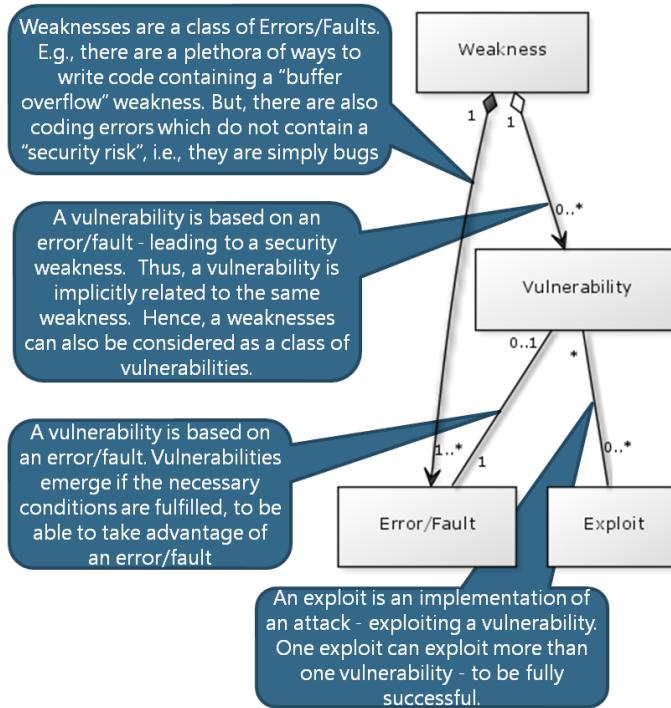


Figure 27 - Relationships between faults, weaknesses, vulnerabilities and exploits

The following example explains the abstraction levels with respect to the “W32.Blast.Worm” security incident:

Table 30 - Example on abstraction levels concerning software security issues

Abstraction Level	Example
Exploit(s)	Blaster/Msblast/LovSAN and Nachi/Welchia worms etc.
Vulnerability	CVE-2003-0352: Buffer overflow in a certain DCOM interface for RPC in Microsoft Windows NT 4.0, 2000, XP, and Server 2003 allows remote attackers to execute arbitrary code via a malformed message...
Weakness	CWE-119 “Failure to constrain operations within the bounds of a memory buffer” or CWE-121 “Stack-based buffer overflow.”
Error/Fault:	Windows Distributed Component Object Model (DCOM) Remote Procedure Call (RPC) interface, while loop in the GetMachineName() function: while (*pwszTemp != L'\\') *pwszServerName++ = *pwszTemp++; /* ... */

Before enlisting vulnerability databases, it is necessary to mention two major and current efforts in creating a dictionary and terminology base which are in turn used in the vulnerability databases.

Table 31 - Examples of dictionary and terminology sources for vulnerability databases

Common Weakness Enumeration(CWE™)	<p>The CWE is a “formal list of software weakness types” hosted by MITRE Corporation</p> <p>Targeted at both the development community and the community of security practitioners, Common Weakness Enumeration (CWE™) is a formal list or dictionary of common software weaknesses that can occur in software’s architecture, design, code or implementation that can lead to exploitable security vulnerabilities. CWE was created to serve as a common language for describing software security weaknesses; serve as a standard measuring stick for software security tools targeting these weaknesses; and to provide a common baseline standard for weakness identification, mitigation/security controls, and prevention efforts (32).</p> <p>Thus, the CWE – provides a good abstraction level and common language to compare the results from different sources on software weaknesses/vulnerabilities.</p>
Common Vulnerability Enumeration (CVE™)	<p>CVE is sponsored by the office of Cybersecurity and communications at the U. S. Department of Homeland Security. MITRE Corporation maintains the CVE. CVE is not vulnerability database. CVE is designed to allow vulnerability databases and other capabilities to be linked together, and to facilitate the comparison of security tools and services. As such, CVE does not contain information such as risk, impact, fix information or detailed technical information. CVE only contains standard identifier number with status indicator, a brief description and a reported vulnerability reports and advisories (33).</p>

At the time of the initial publication of SAE J3061, the following are the two prominent current and running vulnerability databases:

Table 32 - Examples of Vulnerability Databases

BugTraq	<p>Securityfocus – a subsidiary of Symantec Cooperation – hosts a database of known security vulnerabilities at securityfocus.com (34). The database contains around 57000 entries when this document is written, among others from the Bugtraq mailing list, which was founded in 1993. The main purpose of the site is to inform the security community on vulnerabilities, before they are exploited. Thus, the most relevant information is the list of vulnerable applications per entry. The moderators – of the mailing list/database – try to derive for each vulnerability a related “<i>Vulnerability Class</i>”, i.e., source of vulnerability, But, this isn’t possible in a lot of cases, e.g., because the affected software packages are proprietary and the source is not available.</p>
National Vulnerability Database by NIST (NVD)	<p>The National Vulnerability Database (NVD) (35) operated by the National Institute of Standards and Technology (NIST), contains more than 57000 entries when this document is written. All entries are identified by an ID, borrowed from the Common Vulnerabilities and Exposures (CVE®) dictionary (33). The benefit of working with the NVD, instead of the CVE, is that – where possible – more details are presented per vulnerability. E.g., as a technical detail the vulnerability type/reason, according to the Common Weakness Enumeration (CWE), is provided.</p>

There are several ways in which vulnerabilities could be classified. For example, the point at which they are injected such as the phase of the lifecycle, or where they are found such as, in hardware or in software etc.. From the product lifecycle point of view, weaknesses could be specific flaws in implementation, oversights in design, missing requirements and inadequate operational procedures that could be exploited by a threat source. Vulnerabilities may also result from non-application of Cybersecurity Controls, or which, while applied, retain some weakness. There are quite a few vulnerability classification schemes proposed in the academic literature. But, two major current and useful efforts are the following:

Table 33 - Examples of Vulnerability Classification Schemes

Common Weakness Scoring System (CWSS™)	<p>The CWSS™ in its current version 0.8, is a scoring system that should help several stakeholders – concerned with software security – to assess and – where applicable – prioritize reported software weakness. A <i>weakness in the meaning of the CWSS™ is “a software coding error – that is, CWEs – found” in a concrete application</i>². Thus, <i>CWSS cannot be easily applied to the weaknesses in the CWE database</i>, i.e., because they are too abstract. A downgraded/adapted³ version of the CWSS was used for the scoring in the “CWE/SANS Top 25 Most Dangerous Software Errors” list, c.f. 3.3.1.</p> <p>To assess a current weakness according to CWSS, one has to score the weakness according to 18 factors. For each factor, a table of available values and related weights is provided, c.f. (36) for a full overview. The 18 factors are grouped into three metric groups. For each metric group, a specific formula to combine the weights of the factors into the related sub score is given. Finally, the sub scores are multiplied to achieve the weakness score value, which is in a range between 0 and 100.</p>
Common Vulnerability Scoring System (CVSS)	<p><i>CVSS is a scoring system for vulnerabilities and is widely used</i>, e.g., in the national vulnerability database. Similar to CWSS, the scoring is based on different factors – called metrics – in CVSS 12 metrics are used. For each metric, a list of selectable choices and related weights is given. The weights of the metrics are combined by fixed formulas per metric group: “Base, Temporal and Environmental”. It is assumed, that “<i>the threat posed by a vulnerability may change over time</i>” and that “<i>different environments can have an immense bearing on the risk that a vulnerability poses</i>”. Thus, the metrics in the temporal and environmental group are “time/user” dependent and often not ascertained at all, e.g., in the NVD only the score for the base metric group is given. The base metric group contains the following metrics:</p> <ul style="list-style-type: none"> • Access Vector (AV) Assess how the vulnerability can be exploited, e.g., only local or also remote • Access Complexity (AC) Asses how complicated it is or which conditions are necessary to exploit the vulnerability • Authentication (Au) Assess the amount of authentication steps and attacker must possess, before the vulnerability can be exploited. • Confidentiality Impact I Assess the impact for the confidentiality by a successful exploit • Integrity Impact (I) Assess the impact for the integrity by a successful exploit • Availability Impact (A) Assess the impact for the availability by a successful exploit <p>The list of the selectable choices per metric, the related weights and the formula to combine the weights to the base score can be found in (37).</p> <p><i>Since CVSS is an open system and the effort to assess at least the basic score is manageable, it could be a valuable tool.</i></p>

The above data bases and classification schemes concern themselves with software vulnerabilities. These databases do not have vulnerabilities reported in Cyber-Physical Vehicle Systems. There are efforts underway to develop a means of reporting and sharing vulnerability information within the vehicle industry. One such effort is an Auto ISAC etc.

In any case, the NCD, and CVE, CVSS presented in this section could be used also by the vehicle sector. The information could be directly used or could be used after adaption to the context.

2. Note that this definition is quite similar to the presented definition of vulnerabilities. However, with the CVSS MITRE has a different scoring system for vulnerabilities.

3. Actually only the values for “prevalence, importance, and likelihood of exploit” have been assessed by the voters, while “the remaining 15 factors were all assigned “Not Applicable” values.

APPENDIX F – VEHICLE LEVEL CONSIDERATIONS

Appendix F discusses aspects of Cybersecurity that should be addressed at the vehicle-level. The material in this Appendix is not comprehensive, but is provided as a reference to further research into the vehicle-level implications of Cybersecurity.

Vehicle-Level:

Some key aspects of Cybersecurity are defined at the vehicle-level, but will significantly affect the system-level technical requirements. The following are examples of vehicle-level decisions that impact system-level Technical Cybersecurity Requirements:

- What electrical architecture will the vehicle have?
- Definition of the number and types of internal vehicle communication networks in the vehicle. Examples of vehicle network types include: CAN, LIN, Ethernet, MOST, Flexray, etc.
- Identification of the number of electronic modules/ ECUs in the vehicle.
- Which vehicle network each ECU (and sensors with hardware I/O) will be located on.
- How each ECU will be identified?
- Which ECU's should be authenticated in order to perform their functions? If so how will this be done?
- Which messages will be created by which ECU's,
- Required hardware features for specific ECU's,
- Required software for the various ECU's, perhaps including requirements based on the functionality of an ECU (e.g., an ECU affecting a safety-critical system may have specific requirements associated with Cybersecurity).

At the Vehicle-Level it is also important to prevent users from making unauthorized changes that could reduce Cybersecurity for the vehicle after it has been sold. Potential sources of unauthorized changes include:

- “Tuners” who may change calibration settings or software to get different powertrain performance features. The OEM needs to determine what changes by “Tuners” are permissible, and block those that are not, or that could reduce Cybersecurity.
- Software that may come from a DVD, Bluetooth-paired phone, etc. and may attempt to become installed on the vehicle without informing the user or telling the users about possible risks.

OEM's should analyze if there are any modules that, for Cybersecurity reasons, should be returned to default factory conditions or even disabled/destroyed to prevent the module from getting onto the used part market. For a vehicle's End of Life, OEM's should provide instructions to Salvage Yard operators on what items require special treatment (e.g., the quickest way to erase data from infotainment modules, specific modules that need to be destroyed and how to destroy them) relative to Cybersecurity.

OEM's should ensure that their Tier 1 suppliers (and possibly Tier 2) know the process for reporting any possible Cybersecurity incidents, and that process should be as clear and easy to use as possible.

Example Architecture Approach:

In February 2014, NIST released a “Framework for Improving Critical Infrastructure Cybersecurity” (8). This framework provides broad guidelines to apply best practices of risk management to improving Cybersecurity and resilience of critical infrastructure not specific to any domain. It aims to help organizations align its business drivers and requirements with risk tolerances and resources to provide the strongest cyber secure responses. It has identified characteristics of a strong vehicle architecture as follows:

Identify → Protect → Detect → Respond → Recover

Architectures that incorporate elements from each tenet that can co-exist provide strong defenses against a variety of Cybersecurity risks.

Examples for each of the tenets above that tend to make an architecture more secure are, (but not limited to):

1. Identify:

- a. Capture existing regulations and guidelines
- b. Capture business risks
- c. Capture best known state of Cybersecurity threat landscape (evolving)
- d. Create Threat Assessment and Response Policy

2. Protect:

Every component in a subsystem that impacts (i.e., initiates/terminates/provides inputs for) key Cybersecurity functions and stores/transports Cybersecurity critical data should be secured by a security mechanism that is appropriate for the level of risk.

Security Mechanism - System functionality that preserves the desired Cybersecurity properties of data and routines. Security mechanisms include 1.) authentication mechanisms (is an agent allowed access to a certain resource) 2.) integrity mechanism (prevent unauthorized modification/writing of messages) 3.) confidentiality mechanism (preventing unauthorized reads of data).

Security Mechanisms:

- a. Isolation/partitioning of systems that have external access (e.g., Wi Fi, Bluetooth, OBD) from safety-critical systems and systems that can have important impacts on the operation of the vehicle.
- b. Access control mechanisms to restrict access to critical ECUs, critical modes of every ECU (diagnostic mode), and their data.
- c. Access control mechanisms that tend to restrict access to information about the operational state of the vehicle, privacy sensitive information, financially sensitive information, detailed design information, etc.
- d. Central gateway or domain controller based architectures.
- e. Communication mechanisms between devices that tend to verify the transmitter of a message and that it hasn't been modified in transit.
- f. Mechanisms that effectively secure maintenance operations, including both diagnostic procedures and update procedures.
- g. Steps taken to make reverse engineering of an effective attack methodology more difficult (for example, code obfuscation, minimize Return-Oriented-Programming (ROP) usage, etc.).

Benefits of the below items are still under investigation. Implementation of any of the following are still being researched and significant concerns with in-field feasibility and customer acceptance exist.

1. Detect:

- a. In-vehicle network intrusion detection capability to detect incorrect or unexpected behavior of the system.
- b. Continuous monitoring of the vehicle's network during operation.
- c. Active firewalls that report when a positive hit is encountered.

2. Response Mechanisms:

- a. For positive intrusion or unexpected behavior detection, the possibility of reacting to the detection either locally on vehicle or off board. For example, cut off safety features from connected features (e.g., limp-home mode).
- b. Incident reporting to cloud and/or on-board logging functions (Event Data Recorder).
- c. Creating forensic logs of events on the vehicle allowing interpretation of vehicle operation.

3. Recover:

- a. Systems designed to rapidly and securely deploy updates and patches to fielded systems, perhaps via a remote mechanism (Secure over-the-air updates to vulnerable software components).
- b. Return to dealer or manufacturer for hardware vulnerability.

The examples stated above, if incorporated into vehicular architectures, will likely make them more robust against failures due to cyber-attacks. However concerns about privacy and response time of key vehicle functions (e.g., brakes, steering, airbags, etc.) should also be considered when attempting to optimize in-vehicle architecture against Cybersecurity threats. This does not imply that this is a comprehensive list, or that architectures cannot be secure without these practices.

APPENDIX G – CURRENT CYBERSECURITY STANDARDS AND GUIDELINES THAT MAY BE USEFUL TO THE VEHICLE INDUSTRY

Appendix G lists Standards and Guidelines from a variety of sources that may be useful for members of the Vehicle Industry in understanding the overall Cybersecurity realm, and in determining the details of how Cybersecurity will be implemented in their organizations. The following Table is not comprehensive.

Table 34 - Cybersecurity standards and guidelines that may be useful to the vehicle industry

Number	Date	Title	Link/Notes
Federal Information Processing Standards (FIPS)			
FIPS 140-2	May-01	FIPS 140-2 (EPUB) FAQ: Security Requirements for Cryptographic Modules; Annex A: Approved Security Functions (EPUB) FAQ; Annex B: Approved Protection Profiles (EPUB) FAQ; Annex C: Approved Random Number Generators (EPUB) FAQ; Annex D: Approved Key Establishment Techniques (EPUB) FAQ	FIPS 140-2 (including change notices as of 12-03-2002); Annex A: Approved Security Functions; Annex B: Approved Protection Profiles; Annex C: Approved Random Number Generators; Annex D: Approved Key Establishment Techniques
FIPS 140-3	25-Apr-14	Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Validation Program	http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402IG.pdf
FIPS 200	Mar-06	Minimum Security Requirements for Federal Information and Information Systems	FIPS 200
Nuclear Regulatory Commission Publication			
RG 5.71		“Cyber Security Programs For Nuclear Facilities,” U.S. Nuclear Regulatory Commission, Rockville, MD, January 2010	http://pbstdupws.nrc.gov/docs/ML0903/ML090340159.pdf

Table 34 - Cybersecurity standards and guidelines that may be useful to the vehicle industry (cont.)

NIST Technical Reports (NISTIR)			
NISTIR 7946	Apr-14	CVSS Implementation Guidance	http://nvlpubs.nist.gov/nistpubs/ir/2014/NISTIR.7946.pdf
NIDTR 7981	7-May-14	Mobile, PIV, and Authentication	http://csrc.nist.gov/publications/drafts/nistir-7981/nistir7981_draft.pdf
NISTIR 8006	23-Jun-14	NIST Cloud Computing Forensic Science Challenges	http://csrc.nist.gov/publications/drafts/nistir-8006/draft_nistir_8006.pdf
NIST Special Publication 800 Series			
SP 800-16 Rev. 1 (3rd draft)	Mar. 14, 2014	DRAFT A Role-Based Model for Federal Information Technology / Cyber Security Training (3rd public draft)	Announcement and Draft Publication
SP 800-18 Rev.1	Feb-06	Guide for Developing Security Plans for Federal Information Systems	SP 800-18 Rev. 1
SP 800-30 Rev. 1	Sep-12	SP 800-30 Rev. 1 (EPUB) FAQ: Guide for Conducting Risk Assessments	SP 800-30 Rev. 1
SP 800-37 Rev. 1	Feb-10	SP 800-37 Rev. 1 (including updates as of June 05, 2014) FAQ: Guide for Applying the Risk Management Framework to Federal Information Systems: A Security LifeCycle Approach	doi:10.6028/NIST.SP.800-37r1 [Direct Link] ; Supplemental Guidance on Ongoing Authorization, (June 2014)
SP 800-39	Mar-11	Managing Information Security Risk: Organization, Mission, and Information System View	SP 800-39
SP 800-40 Rev. 3	Jul-13	SP 800-40 Rev. 3 FAQ: Guide to Enterprise Patch Management Technologies	doi:10.6028/NIST.SP.800-40r3 [Direct Link]

Table 34 - Cybersecurity standards and guidelines that may be useful to the vehicle industry (cont.)

SP 800-46 Rev. 1	Jun-09	Guide to Enterprise Telework and Remote Access Security	SP 800-46 Rev. 1
SP 800-50	Oct-03	Building an Information Technology Security Awareness and Training Program	SP 800-50
SP 800-51 Rev. 1	Feb-11	Guide to Using Vulnerability Naming Schemes	SP 800-51 Rev. 1
SP 800-52 Rev. 1	Apr. 2014	SP 800-52 Rev. 1 FAQ: Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations	doi:10.6028/NIST.SP.800-52r1 [Direct Link]
SP 800-53 A Rev. 1	Jun-10	Guide for Assessing the Security Controls in Federal Information Systems and Organizations, Building Effective Security Assessment Plans	SP 800-53A Rev. 1; Security Assessment Overview and Cases
SP 800-53A Rev. 4	Apr-13	SP 800-53 Rev. 4 (including updates as of 01-15-2014) FAQ; Security and Privacy Controls for Federal Information Systems and Organizations	doi:10.6028/NIST.SP.800-53r4 [Direct Link]; Word version of SP 800-53 Rev. 4 (01-15-2014); XML file for SP 800-53 Rev. 4 (01-15-2014); Summary of NIST SP 800-53 Revision 4
SP 800-53 Appendix H	28-Aug-14	DRAFT Appendix H: Security and Privacy Controls for Federal Information Systems and Organizations	Announcement and Draft Publication
SP 800-60 Rev. 1	Aug-08	Guide for Mapping Types of Information and Information Systems to Security Categories	Vol. 1: Guide for Mapping Types of Information and Information Systems to Security Categories; Vol. 2: Appendices

Table 34 - Cybersecurity standards and guidelines that may be useful to the vehicle industry (cont.)

SP 800-70 Rev. 2	Feb-11	National Checklist Program for IT Products: Guidelines for Checklist Users and Developers	SP 800-70 Rev. 2
SP 800-82 Rev.2	14-May-14	DRAFT Guide to Industrial Control Systems (ICS) Security	Announcement and Draft Publication
SP 800-94	Feb-07	Guide to Intrusion Detection and Prevention Systems (IDPS)	SP 800-94
SP 800-101 Rev.1	May-14	SP 800-101 Rev. 1 FAQ: Guidelines on Mobile Device Forensics	doi:10.6028/NIST.SP.800-101r1 [Direct Link]
SP 800-115	Sep-08	Technical Guide to Information Security Testing and Assessment	SP 800-115
SP 800-117 Rev. 1	Jan. 6, 2012	DRAFT Guide to Adopting and Using the Security Content Automation Protocol (SCAP) Version 1.2	Announcement and Draft Publication
SP 800-121 Rev. 1	Jun-12	Guide to Bluetooth Security	SP 800-121 Rev. 1
SP 800-124		Guide to Data-Centric System Threat Modeling	http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-124r1.pdf
SP 800-124 Rev. 1	Jun-13	SP 800-124 Rev. 1 FAQ: Guidelines for Managing the Security of Mobile Devices in the Enterprise	doi:10.6028/NIST.SP.800-124r1 [Direct Link]
SP 800-126 Rev. 2	Sep-11	The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.2	SP 800-126 Rev. 2: Errata for SP 800-126 Rev. 2
SP 800-127	Sep-10	Guide to Securing WiMAX Wireless Communications	SP 800-127

Table 34 - Cybersecurity standards and guidelines that may be useful to the vehicle industry (cont.)

SP 800-128	Aug-11	Guide for Security-Focused Configuration Management of Information Systems	<u>SP 800-128</u>
SP 800-137	Sep-11	Information Security Continuous Monitoring for Federal Information Systems and Organizations	<u>SP 800-137</u>
SP 800-144	Dec-11	Guidelines on Security and Privacy in Public Cloud Computing	<u>SP 800-144</u>
SP 800-150	Oct. 28, 2014	DRAFT Guide to Cyber Threat Information Sharing	<u>Announcement and Draft Publication</u>
SP 800-160	12-May-14	DRAFT Systems Security Engineering: An Integrated Approach to Building Trustworthy Resilient Systems	<u>Announcement and Draft Publication</u>
SP 800-161	Jun. 3, 2014	DRAFT Supply Chain Risk Management Practices for Federal Information Systems and Organizations (Second Draft)	<u>Announcement and Draft Publication</u>
SP 800-163	19-Aug-14	DRAFT Technical Considerations for Vetting 3rd Party Mobile Applications	<u>Announcement and Draft Publication</u>
SP 800-164	Oct. 31, 2012	DRAFT Guidelines on Hardware-Rooted Security in Mobile Devices	<u>Announcement and Draft Publication</u>
SP 800-167	Aug. 22, 2014	DRAFT Guide to Application Whitelisting	<u>Announcement and Draft Publication</u>

Table 34 - Cybersecurity standards and guidelines that may be useful to the vehicle industry (cont.)

SP 800-173	In Development	Cloud-adapted Risk Management Framework: Guide for Applying the Risk Management Framework to Cloud-base Federal Information Systems	Document not available, for a presentation of the publication: http://gsi.nist.gov/global/docs/sit/2014/AKarmelWed.pdf
SP 800-174	In Development	Security and Privacy Controls for Cloud-based Federal Information Systems	Document not available, for a presentation of the publication: http://gsi.nist.gov/global/docs/sit/2014/AKarmelWed.pdf
SP 800-299	May-14	Cloud Computing Security Reference Architecture	http://collaborate.nist.gov/twiki-cloud-computing/pub/CloudComputing/CloudSecurity/NIST_Security_Refere http://collaborate.nist.gov/twiki-cloud-computing/pub/CloudComputing/CloudSecurity/NIST_Security_Refere
Other Related Documents			
Department of Homeland Security (DHS)	Jan-08	“Catalog of Control Systems Security Recommendations for Standards Developers,” Department of Homeland Security Control Systems Security Program, September 2009	http://trygstad.rice.edu:8000/Policies%20&%20Tools/IndustrySpecifcTools/Catalog%20of%20Control%20Systems%20Security-%20Recommendations%20for%20Standards%20Developers%20-%20US%20CERT.pdf
DHS	Feb 2008	“Cybersecurity procurement language for Control Systems”, Revision 3 Version 1.8,	http://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/SCAD_A_Procurement
Contains sample Cybersecurity requirements procurement language guide for SCADA system owners/operators. The goal of this DHS project was to define procurement language that will help ensure that security is integrated into control systems.			

Table 34 - Cybersecurity standards and guidelines that may be useful to the vehicle industry (cont.)

NIST Paper	Oct-10	“Security Assurance Levels: A Vector Approach to Describing Security Requirements,” National Institute of Standards and Technology, Gaithersburg, MD,	http://www.nist.gov/customcf/get_pdf.cfm?pub_id=906330
APTA Recommended Practice APTA SS-CCS-RP-001-10	Jul-10	“Securing Control and Communications Systems in Transit Environments, Part 1: Elements, Organization and Risk Assessment/Management,” American Public Transportation Association Control and Communications Working Group, Washington DC, .	http://www.apta.com/resources/standards/Documents/APTA-SS-CCS-RP-001-10.pdf
NIST National Vulnerability Database		Guide to Securing Apple OS X 10.8 Systems for IT Professionals	https://web.nvd.nist.gov/view/nvcp/repository/checklistDetail?id=511 , Project Contacts: Mr. Lee Badger, (301) 975-3176, lee.badger@nist.gov; Mr. Murugiah Souppaya, (301) 975-4102, murugiah.souppaya@nist.gov
Dept of Energy	Apr-14	Cybersecurity Procurement Language for Energy Delivery (April 2014)	http://energy.gov/sites/prod/files/2014/04/f15/CybersecProcurementLanguage-EnergyDeliverySystems_040714_fin.pdf
DARPA HACMS	Ongoing	Defense Advanced Research Projects Agency (DARPA): Open Catalog for High-Assurance Cyber Military Systems (HACMS) project	http://www.darpa.mil/opencatalog/HACMS.html

APPENDIX H – VEHICLE PROJECT AWARENESS

Appendix H summarizes the key research projects on Vehicle Cybersecurity beginning with 2004 and up through the present (i.e., initial release of SAE J3061). This listing may not be comprehensive since this is a rapidly developing and evolving area for research.

Table 35 - Vehicle Cybersecurity-relative research projects (2004 to present)

Project, Timeline and Origin	Project Objectives and Website
EVITA, 2008 – 2011, EU	<p>The objective of the EVITA project is to design, verify, and prototype an architecture for vehicle on-board networks where Cybersecurity-relevant components are protected against tampering and sensitive data are protected against compromise when transferred inside a vehicle. By focusing on the protection of the intra-vehicle communication EVITA complements other e-safety related projects that focus on the protection of the vehicle-to-X communication.</p> <p>Website: http://www.evita-project.org/</p>
PRECIOSA, 2008 – 2010, EU	<p>The major objectives of the PRECIOSA project are to:</p> <ul style="list-style-type: none"> • Define an approach for the privacy evaluation of co-operative systems in terms of communication privacy and data storage privacy • Define a privacy aware architecture for co-operative systems which involves suitable trust models and ontologies, a V2V privacy verifiable architecture, and a V2I privacy verifiable architecture, and which includes the architecture components for protection, infringement detection, and auditing • Define and validate guidelines for privacy aware co-operative systems • Investigate specific challenges for privacy. <p>Website: http://www.preciosa-project.org/</p>
SeVeCOM, 2006 – 2010, EU	<p>SEVECOM will focus on communications specific to road traffic. This includes messages related to traffic information, anonymous safety-related messages, and liability-related messages. The following work is foreseen:</p> <ol style="list-style-type: none"> 1) Identification of the variety of threats 2) Specification of an architecture and of Cybersecurity mechanisms. The following topics will be fully addressed: <ul style="list-style-type: none"> - Key and identity management - Secure communication protocols, including secure routing - Tamper proof device and decision on crypto-system - Privacy. The following topics will be investigated in preparation of further work: <ul style="list-style-type: none"> - Intrusion Detection - Data consistency - Secure positioning - Secure user interface 3) The definition of cryptographic primitives which take into account the specific operational environment. <p>Website: http://cordis.europa.eu/project/rcn/80592_en.html</p>

Table 35 - Vehicle Cybersecurity-relative research projects (2004 to present) (cont.)

Project, Timeline and Origin	Project Objectives and Website
DRIVE-C2X, 2011 – 2014, EU	<p>DRIVE C2X focuses on communication among vehicles (C2C) and between vehicles, a roadside and backend infrastructure system (C2I). Previous projects such as PReVENT, CVIS, SAFESPOT, COOPERS, and PRE-DRIVE C2X have proven the feasibility of safety and traffic efficiency applications based on C2X communication. DRIVE C2X goes beyond the proof of concept and addresses large-scale field trials under real-world conditions at multiple national test sites across Europe.</p> <p>The systems to be tested are built according to the common European architecture for cooperative driving systems defined by COMeSafety, thus guaranteeing the compliance with the upcoming European ITS standards. This approach also ensures that the results of DRIVE C2X have long-term validity on European level, giving system developers as well as decision makers on the industry and authority side the necessary decision confidence.</p> <p>For the first time in Europe, DRIVE C2X is also implementing and testing a concept for the integration of a data backend, therefore enabling commercial services based on C2X communication data to private and commercial customers. Such services are expected to become a major revenue source for cooperative driving systems and can be the key for successful implementation of this technology on European roads.</p> <p>Website: http://www.preciosa-project.org/</p>
PRESERVE, 2011 – 2014, EU	<p>The mission of PRESERVE is, to design, implement, and test a secure and scalable V2X Cybersecurity Subsystem for realistic deployment scenarios.</p> <p>Preserve Objectives</p> <ol style="list-style-type: none"> 1. Create an integrated V2X Cybersecurity Architecture (VSA) and design, implement, and test a close-to-market implementation termed V2X Cybersecurity Subsystem (VSS). 2. Prove that the performance and cost requirements for the VSS arising in current FOTs and future product deployments can be met by the VSS, especially by building a Cybersecurity ASIC for V2X. 3. Provide a ready-to-use VSS implementation and support to FOTs and interested parties so that a close-to-market Cybersecurity solution can be deployed as part of such activities. 4. Solve open deployment and technical issues hindering standardization and product pre-development. <p>Website: http://www.preserve-project.eu/</p>
OVERSEE, 2010 – 2012, EU	<p>The OVERSEE project develops an open, standardized in-vehicle software and communication platform, which will empower (almost) everyone to develop vehicular applications and offering these applications to all vehicles that are equipped - very similar to Apple's App Store.</p> <p>The OVERSEE platform will in particular ensure that such applications cannot harm each other or any other vehicle-internal IT system. This protection mechanism applies regardless whether the interference is caused by an application or user failure or a malicious manipulation caused by a hacker or computer virus.</p> <p>Another task of the OVERSEE platform is providing internal and external vehicular communication interfaces to vehicle applications.</p> <p>Website: https://www.oversee-project.com/</p>
EURO-MILS, 2012 – 2015, EU	<p>The project will provide trustworthiness by design and high assurance based on the Multiple Independent Levels of Security (MILS) approach. MILS is a high-assurance security architecture based on the concepts of separation and controlled information flow.</p> <p>This project modifies the ICT security ecosystem by introducing a verified and validated design at the point where incentives and prospects of success are the highest. The project introduces into the European trustworthy ICT landscape the heart of the MILS architecture, the MILS platform: a small virtualisation platform that offers the secure decomposition of complex embedded systems into independent components.</p> <p>Website: http://www.euromils.eu/</p>

Table 35 - Vehicle Cybersecurity-relative research projects (2004 to present) (cont.)

Project, Timeline and Origin	Project Objectives and Website
SESAMO, 2012 – 2015, EU	<p>The SESAMO project addresses the root causes of problems arising with convergence of safety and security in embedded systems at architectural level, where subtle and poorly understood interactions between functional safety and Cybersecurity mechanisms impede system definition, development, certification, and accreditation procedures and standards.</p> <p>Key elements of the SESAMO approach are:</p> <ul style="list-style-type: none"> ▪ a methodology to reduce interdependencies between safety and security mechanisms and to jointly ensure their properties ▪ constructive elements for the implementation of safe and secure systems ▪ procedures for integrated analysis of safety and Cybersecurity ▪ an overall design methodology and tool-chain utilizing the constructive elements and integrated analysis procedures to ensure that safety and Cybersecurity are intrinsic characteristics of the system. <p>The goal of SESAMO is to enable cost-efficient and systematic design, analysis, development, and assessment of distributed safety and Cybersecurity critical embedded systems.</p> <p>Website: http://sesamo-project.eu/</p>
SHIELDS, 2008 – 2010, EU	<p>Software systems continue to be crippled by Cybersecurity vulnerabilities. The main objective of SHIELDS is to increase software security by bridging the gap between security experts and software practitioners and by providing the software developers with the means to effectively prevent occurrences of known vulnerabilities when building software.</p> <p>We will achieve this objective by developing novel formalisms for representing security information, such as known vulnerabilities, in a form directly usable by development tools, and accessible to software developers. This information will be stored in an internet-based Security Vulnerabilities Repository Service (SVRS) that facilitates fast dissemination of vulnerability information from security experts to software developers. We will also present a new breed of security methods and tools (some open source, some commercial) that are constantly kept up-to-date by using the information stored in the SVRS.</p> <p>In addition to the SVRS, and new security tools, the project will create a SHIELDS Compliant certification for tools and a SHIELDS Verified logo program for software developers that will offer an affordable and yet technically effective evaluation and certification method in the fight against common security vulnerabilities.</p> <p>Website: http://cordis.europa.eu/project/rcn/85431_en.html</p>
CVIS, 2006 – 2010, EU	<p>The CVIS objectives are:</p> <ul style="list-style-type: none"> ▪ to create a unified technical solution allowing all vehicles and infrastructure elements to communicate with each other in a continuous and transparent way using a variety of media and with enhanced localization ▪ to enable a wide range of potential cooperative services to run on an open application framework in the vehicle and roadside equipment ▪ to define and validate an open architecture and system concept for a number of cooperative system applications, and develop common core components to support cooperation models in real-life applications and services for drivers, operators, industry and other key stakeholders ▪ to address issues such as user acceptance, data privacy and security, system openness and interoperability, risk and liability, public policy needs, cost/benefit and business models, and roll-out plans for implementation. <p>Website: http://www.cvisproject.org/</p>

Table 35 - Vehicle Cybersecurity-relative research projects (2004 to present) (cont.)

Project, Timeline and Origin	Project Objectives and Website
TECOM, 2008 – 2011, EU	<p>The project will make a systematic approach for the development of trusted embedded systems, consisting of hardware platforms with integrated trust components and also work on the necessary trusted operating systems.</p> <ul style="list-style-type: none"> ▪ Trusted hardware, Trusted Platform Module as VHDL design, which could adapted to different host systems together with processors supporting a trust architecture. ▪ Trusted operating systems mainly based on the new virtualization/hypervisor architecture which are already in use at the PC world. Adapting it to the specific requirements of small platforms and trusted modules. ▪ Security layers for implementing easy accessible security mechanisms. ▪ Trusted protocols: Elementary TC protocols like TSS (Host interface API) and TNC (trusted network connect, an advanced secure communication protocol) will be adapted for embedded platforms. <p>Application examples will give us feedback about available trust functionality, application friendliness and user requirements. The project will use the results and experience from the project to influence the TC standardization and prepare for application scenarios like mobile phones, communication, e-commerce, industrial and vehicle.</p> <p>Website: http://cordis.europa.eu/project/rcn/85362_en.html</p>
SEPIA, 2010 – 2013, EU	<p>SEPIA will focus on three topics: Cybersecurity enhancements of mobile platforms, cryptography and privacy protecting technologies, as well as delta-evaluation and certification methodologies. A major objective of SEPIA is to define a next-generation security-architecture for mobile and embedded systems, addressing topics such as isolated execution space, virtualization and secure protection of confidential data.</p> <p>Moreover, privacy protecting mechanisms, based on strong cryptography and time- and cost efficient certification processes, reducing the time from design to market, will be researched in the project. In SEPIA, establishing trustworthiness is seen as an asset that is considered right from the design phase rather than being addressed as add-on feature. SEPIA will include theoretical and practical research as well as the development of proof-of-concept prototypes.</p> <p>Website: http://sepiaproject.eu</p>
NoW, 2004 – 2008, Germany	<p>The NOW project aims to solve technical key questions on the communication protocols and data security for car-to-car communications and to submit the results to the standardization activities of the Car2Car Communication Consortium.</p> <p>Furthermore, a test bed for functional tests and demonstrations is implemented which will be developed further on toward a reference system for the Car2Car Communication Consortium specifications.</p> <p>The NOW project addresses all issues from design and standardization of a car-to-car and car-to-infrastructure communication system until its market introduction. The main objectives of the project were to:</p> <ul style="list-style-type: none"> • Support active safety applications and infotainment applications with an infrastructure between vehicles. • Enhance radio systems based on IEEE 802.11 technology. • Be active in standardization at European level with the Car2Car CC. • Implementation of a reference system. • Planning of introduction strategies and business models. <p>Website: www.network-on-wheels.de, https://dsn.tm.kit.edu/english/projects_now-project.php</p>
ARAMiS, 2011 – 2014, Germany	<p>Available only in German language.</p> <p>Website: http://www.projekt-aramis.de/index.php</p>

Table 35 - Vehicle Cybersecurity-relative research projects (2004 to present) (cont.)

Project, Timeline and Origin	Project Objectives and Website
HEAVENS, 2013 – 2016, Sweden	<p>The project will focus on Cybersecurity aspects of electrical and/or electronic (E/E) systems within vehicle systems. Project deliverables are available on request.</p> <p>The project goals are as follows:</p> <ul style="list-style-type: none"> ▪ Study and identify state-of-the-art in Cybersecurity in the vehicle industry. ▪ Identify needs and requirements of Cybersecurity in the vehicle industry. ▪ Identify potential threats, threat agents and vulnerabilities to construct Cybersecurity models. ▪ Define methodologies and identify tool support for evaluating software Cybersecurity. ▪ Investigate the interplay of safety and Cybersecurity in the E/E architecture, considering ISO 26262, AUTOSAR and other relevant standards. ▪ Demonstrate proof of concepts. <p>Website: http://www.vinnova.se/sv/Resultat/Projekt/Effekta/HEAVENS-HEAling-Vulnerabilities-to-ENhance-Software-Security-and-Safety/ (In Swedish)</p>

APPENDIX I – SECURITY TEST TOOLS OF POTENTIAL USE TO THE VEHICLE INDUSTRY

This appendix lists some security test tool categories, and descriptions, for testing tools that may be of potential use to the vehicle industry for Cybersecurity. Some examples of tools are also given for purposes of illustration only. SAE is not endorsing or recommending any of these example tools or their manufacturers. A complete list of potential testing tools is beyond the scope of this document.

In addition, the choice of a tool can be a personal or an organizational decision based on past experience with the tool or the tool provider, tool features or capabilities, technical support, or ease of use, just to name a few differentiators. Also, it isn't uncommon for two or more tools to overlap in terms of their general or specific capabilities. Some tools may commonly be used by attackers, but these same tools may also be used by system developers to test for Cybersecurity vulnerabilities in their products before they are released and even after they are released.

Traditional security testing tools from the information technology domain may be applied to the vehicle domain if the network environments are a good match. For example, a network sniffing tool designed for use on a typical information technology network may also be usable on a vehicle Ethernet network or vehicular Wi-Fi interface. Likewise, code analyzers may be equally useful for uncovering potential problems in the traditional information technology environment and the vehicle environment. In contrast, an Ethernet/TCP/IP sniffer would be of no use in monitoring vehicle CAN bus traffic; a CAN bus sniffer would be needed.

Table 36 - Sample categories of security test tools

Tool Category	Description	Note: Example Tools are Non-Comprehensive
Static Code Analyzer	Tools that perform analysis of source code or object code, without executing or running the target program, which can highlight poor coding practices or coding errors. Examples: Polyspace, C/C++test, Insure++, IDA Pro, STACK.	
Dynamic Code Analyzer	Tools that perform analysis by executing or running the target program to highlight possible runtime errors, performance bottlenecks, and memory leaks or errors. Examples: Veracode, Insure++, IDA Pro.	
Network Traffic Analyzer	Tools that monitor or capture data from a live network for real-time or offline analysis. The network data can be filtered according to different network protocols and analyzed to determine whether the network is performing as it should or whether there is some evidence of existing malware or an intrusion or anomaly. Some network traffic analyzers can also re-inject or replay captured data. The network traffic may appear on either wired or wireless media. Examples: CANalyzer, Vehicle Spy, Wireshark, Kismet.	
Vulnerability Scanner	Tools that scan networks and software for common security misconfigurations, missing patches, or known code vulnerabilities. Vulnerability scanning on vehicles can be problematic due to the fact that many of the devices, code, and messages on a vehicle are unique to a make and model and the messages may also be proprietary. A vulnerability scan against a standard Wi-Fi interface on a vehicle could indicate a known vulnerable configuration, but vulnerability information simply may not exist for other vehicle subsystems. (Examples: Codenomicon Appcheck, Blackduck, NESSUS)	
Fuzzer	Tools that generate invalid, unexpected, or random data for a system or application interface to detect faulty exception handling or filtering of input, which could then result in a program crash or an unexpected state which could be exploited by an attacker. Examples CANbuster, Codenomicon Defensics.	
Encryption Cracker	Tools that can be used to crack encrypted data. These tools typically search the key space for the key that will convert ciphertext to plaintext. The ciphertext may include encrypted transmissions or stored information. Examples: Aircrack, Cain and Abel, John the Ripper.	
Hardware Debugger	Tools that can be used to extract on-chip program code or data and modify memory contents. Examples: JTAGulator, OpenOCD.	
Known Answer Tester	Tools that can be used to determine the correctness of the implementation of an algorithm such as an encryption algorithm. KAT tools are generally available in connection with NIST-approved algorithms.	

Table 36 - Sample categories of security test tools (cont.)

Tool Category	Description	Note: Example Tools are Non-Comprehensive
Application Tester	Tools designed to test applications such as web applications or mobile applications that may exist in the vehicle domain. Examples: Burp Suite, Nikto.	
Interface Scanner	Tools designed to scan network interfaces for open / listening connections to services running on the device. Examples: nmap to scan a vehicle's Wi-Fi interface.	
Network Stress Tester	Tools designed to generate a high network traffic load to determine how the network and connected devices will respond to a load which might represent a Denial of Service attack on the system or a high user demand for services. Examples: IxChariot, LoadTest	
Exploit Tester	Tools designed to be use in developing, testing, and using exploit code against a system. Examples: Metasploit.	