

USENIX

THE ADVANCED COMPUTING
SYSTEMS ASSOCIATION

A Study on Extending the TARA Methodology for Physical AI: A Case Study of ADS

Jihun Woo and Junhyeong Lee, *Korea Testing & Research Institute*

https://www.usenix.org/conference/vehiclesec26/presentation/woo

This paper is included in the Proceedings of the
4th USENIX Symposium on Vehicle Security and Privacy.

August 10–11, 2026 • Baltimore, MD, USA

ISBN 978-1-939133-56-4

Open access to the Proceedings of the
4th USENIX Symposium on Vehicle Security and Privacy
is sponsored by USENIX.

A Study on Extending the TARA Methodology for Physical AI : A Case Study of ADS

Jihun Woo
Korea Testing & Research Institute

Junhyeong Lee*
Korea Testing & Research Institute

Abstract

Automated Driving Systems (ADS) integrate complex Physical AI, rendering traditional software-centric threat modeling inadequate against emerging AI and physical-layer vulnerabilities. We propose an extended Threat Analysis and Risk Assessment (TARA) methodology, compliant with ISO/SAE 21434, that unifies STRIDE, MITRE ATLAS, and OWASP Top 10 for LLMs. Driven by a newly constructed Attack Library, our framework systematically captures multi-layered cyber-physical threats. Applied to an ADS case study on the NVIDIA Jetson platform, the method yields 38 security checklists. These outputs cover all 32 UN R155 Annex 5 Part A threat categories, identify 29% more AI-specific risks than a STRIDE-only baseline, and achieve ML 3 for data-source integration under D2TARA. Our key contributions are: an integrated Physical AI TARA Methodology, a comprehensive Attack Library synthesized from CVEs, CWEs, and Papers, and 38 actionable security checklists validated against UN R155 Annex 5.

1 Introduction

Recently, Physical AI security has been rapidly gaining attention in both academia and industry. Physical AI refers to artificial intelligence systems that directly interact with and influence the physical world through sensor-actuator loops, making autonomous decisions based on real-time perception of their environment. Unlike traditional AI systems that operate purely in digital domains, Physical AI systems are inherently cyber-physical, where algorithmic vulnerabilities can manifest as tangible safety hazards in the real world. This increased attention stems from an increasing number of cases where small manipulations of physical signals or environmental conditions lead to serious safety risks in AI-based perception and decision-making. For example, research has shown that laser-induced acoustic interference can disrupt MEMS (Micro Electro Mechanical Systems) gyroscopes and break camera stabilization. These types of physical-cyber attacks are highly effective; experiments have confirmed that the object detection performance (mAP) of vision-based vehicles and drones can drop to as low as 41%. [1]

Among various Physical AI systems, Automated Driving Systems (ADS) serve as an exemplary testbed for several reasons:

- They integrate multiple sensing modalities (cameras, LiDAR, radar) that are vulnerable to physical-layer attacks
- They operate in safety-critical environments where security failures directly impact human lives
- They are subject to regulation (e.g., UN R155) that mandate systematic security analysis throughout the vehicle lifecycle.

Conventional threat modeling techniques have evolved primarily to focus on software and IT systems. [2–4] However, these traditional approaches do not fully meet the Security-by-Design requirements and safety goals of Physical AI. [2–4] In the context of ADS, this limitation manifests across multiple dimensions. Conventional threat models such as STRIDE primarily address software vulnerabilities but overlook adversarial threats targeting AI perception and decision-making models. Moreover, they do not systematically account for physical-layer attack vectors such as sensor spoofing or environmental manipulation that can compromise the perception-planning-control loop. Additionally, emerging AI components such as LLM introduce new attack surfaces—including prompt injection and context manipulation—that fall outside the scope of traditional IT security frameworks. [5–8]

To address these challenges, this study proposes an extended threat modeling methodology based on the ISO/SAE 21434 standard TARA process. Specifically, we employ a multi-layered approach: STRIDE for traditional software and network vulnerabilities, MITRE ATLAS for adversarial threats to AI/ML models, and OWASP Top 10 for LLM Applications to capture risks in LLM-integrated components. [7–9]

*Corresponding author

By synthesizing these frameworks into a unified attack library, our approach enables systematic analysis of threats spanning IT infrastructure, AI perception models, and emerging LLM-based decision systems, while explicitly considering how physical-layer attacks can propagate through the ADS architecture.

The contributions of this study are listed as follows:

- **Extended TARA Methodology for Physical AI:** We propose a unified framework that integrates STRIDE, MITRE ATLAS, and OWASP Top 10 for LLM Applications into the ISO/SAE 21434 lifecycle to systematically address AI-specific threats such as adversarial examples and model poisoning. Our mapping selectively applies frameworks by asset functionality and is extensible to vehicle systems integrating automation, AI, or LLMs.

- **Construction of Integrated Attack Library:** We synthesize a comprehensive attack library of 89 normalized attack scenarios from CVEs, CWEs, and Papers. This structured knowledge base facilitates the identification of Physical AI threat scenarios and the analysis of cyber-physical attack paths. Unlike proprietary industry libraries, this publicly released artifact covers sensor, perception, and LLM threats and maps them to STRIDE, MITRE ATLAS, and OWASP Top 10 for LLMs.

- **Practical Validation with Security Checklists:** We derive 38 actionable security checklists through an ADS case study based on ISO/TS 5083 and NVIDIA Jetson Platform Services, of which 11 address AI-specific threats beyond STRIDE. These ADS case-study checklists cover all 32 UN R155 Annex 5 Part A threat categories and provide auditable evidence for Vehicle Type Approval and Cyber Security Management System assessments.

2 Background

2.1 UN Regulation No. 155

UN Regulation No. 155 (UN R155) is a vehicle cybersecurity regulation established by the World Forum for Harmonization of Vehicle Regulations (UN/WP.29) [10]. The regulation mandates that Original Equipment Manufacturers (OEMs) establish a Cyber Security Management System (CSMS) covering the entire vehicle lifecycle—from development to post-production phases—to obtain Vehicle Type Approval (VTA) [10, 11].

UN R155 designates ISO/SAE 21434 as the primary technical guideline for these security activities [10]. Specifically, paragraph 5.3.1 (a) stipulates "knowledge of automotive risk assessment" as a competency requirement for approval authorities and technical services, explicitly citing ISO/SAE 21434

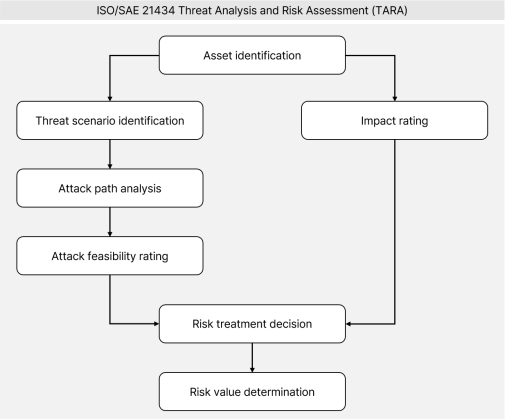


Figure 1: The TARA process on ISO/SAE 21434 [12]

in the accompanying footnote [10]. This implies that conducting Threat Analysis and Risk Assessment (TARA) and implementing security designs in compliance with ISO/SAE 21434 are critical criteria required for passing actual regulatory certification audits.

2.2 ISO/SAE 21434 TARA

ISO/SAE 21434 defines the standard for cybersecurity threat management and requirements in road vehicles [12]. The standard mandates the execution of Threat Analysis and Risk Assessment (TARA) to identify threats and evaluate associated risks. It outlines seven distinct components (or steps) for conducting the TARA process [12].

However, among these seven components, the Attack Feasibility Rating and Risk Value Determination are excluded from the scope of this study, as their evaluation criteria are highly dependent on specific organizational policies and risk appetites. Consequently, this section briefly reviews the remaining five components, which serve as the structural foundation for our extended methodology. [12].

2.3 Related Work: Research on Extending Threat Modeling

Active research is currently underway to integrate or extend existing cybersecurity threat analysis techniques into new contexts. Ghosh et al. [2] proposed a method for autonomous vehicle perception systems that integrates STPA-Sec and STRIDE based on the TARA process. This study derives causal scenarios from a safety perspective using STPA-Sec and identifies potential attack points from a security perspective. Furthermore, it links these findings to the threat analysis phase of ISO/SAE 21434 by mapping the results of STPA-Sec to the STRIDE attack classification scheme [2].

In the domain of AI-integrated systems, Jedrzejewski et al. [7, 8] pointed out that the attack surface of Large Lan-

guage Model Integrated Applications (LIA) is significantly expanded compared to traditional software. They analyzed that existing threat modeling tools are primarily manual-intensive and fail to sufficiently reflect threats specific to LLMs. To overcome this, they proposed utilizing AI-specific security knowledge bases such as OWASP Top 10 for LLM Applications and MITRE ATLAS [7, 8]. Additionally, they introduced a RAG (Retrieval-Augmented Generation) based tool that integrates existing threat models with system architecture information to continuously generate and update threat models. Their research aims to reduce the intervention of security experts in the threat modeling process, thereby saving time and costs [7, 8].

Khalil et al. [4] noted that threat modeling, which is widely used in software development projects, has not yet been systematized in the Cyber-Physical Systems (CPS) domain. This is because identifying threats in a CPS environment is complex due to the continuous interaction between cyber and physical spaces. Accordingly, their study proposed detailed procedures and step-by-step guides for applying the STRIDE technique to CPS [4]. Through a CPS case study, they demonstrated that systematic threat analysis is feasible in this field. However, this approach is limited in that it does not account for the specific threat factors of Physical AI systems that incorporate complex AI algorithms [4].

While preceding studies have laid the groundwork for extending threat modeling to specific CPS or AI applications, our research distinguishes itself by proposing a holistic framework that simultaneously addresses the triadic nature of Physical AI: IT infrastructure, AI algorithmic logic, and physical interaction. The threat scenarios are systematically derived from a unified Attack Library that synthesizes STRIDE, MITRE ATLAS, and the OWASP Top 10 for LLM Applications. It further extends prior efforts by rigorously validating the derived security requirements against UN Regulation No. 155, thereby bridging the gap between academic methodology and industrial compliance to support the actual vehicle type approval (VTA) process.

2.4 Focus Group Interview

ISO/SAE 21434 references several threat analysis frameworks applicable to threat scenario identification, including STRIDE, EVITA, TVRA, and PASTA [12]. STRIDE [13] classifies threats via data flow diagram analysis, while EVITA [14], TVRA [15], and PASTA [16] employ attack-tree- and risk-centric workflows for security-requirement derivation.

To ground our methodological design in industry practice, we conducted a focus group interview (FGI) with four automotive cybersecurity experts, each having 5–15 years of experience in automotive cybersecurity. The panel comprised two practitioners from a vehicle OEM and two engineers from a Technical Service organization accredited for UN R155 Vehicle Type Approval and CSMS conformity assessments. Two consistent themes emerged: (i) STRIDE — particularly the STRIDE-per-element variant — is the dominant framework used during threat scenario identification, and (ii) STRIDE alone is insufficient for capturing AI-specific threats, with the panel favoring incremental extension of the existing STRIDE-based workflow over replacement by an alternative end-to-end methodology.

These findings directly inform our design: STRIDE is retained as the universal baseline Section 3.3.2, and Physical AI–specific frameworks — MITRE ATLAS and OWASP Top 10 for LLM Applications — are selectively layered on top by asset archetype. This design principle also motivates the choice of a STRIDE-only reference baseline in the comparative analysis presented in Section 5.2.

3 Extended TARA Methodology

3.1 Overview of the Extended Approach

The extended TARA methodology proposed in this study is designed to reflect the multidimensional security characteristics of Physical AI systems while strictly adhering to the ISO/SAE 21434 standard process [12]. Figure 2 illustrates the unified workflow, highlighting the Attack Library Construction and multi-framework integration phases that constitute the core extensions of our methodology.

Traditional TARA approaches primarily rely on a single framework, such as STRIDE [13], to identify threats during the early lifecycle stages [17]. However, Physical AI systems—characterized by the tight coupling of IT infrastructure, AI algorithmic logic, and continuous physical interaction—present a complex attack surface that conventional software-centric models fail to capture exhaustively [2–4]. To overcome these limitations, this study introduces an integration of three distinct security frameworks.

Initially, the process begins with the construction of a comprehensive Attack Library, which systematically collects and normalizes diverse attack cases from CVEs [18], CWEs, Papers, and Technical Reports [19]. During this phase, each attack is abstracted into a structured attribute vector—comprising Attack Vector, Target, and Consequence—forming a reusable knowledge base that ensures the analytical rigor of subsequent steps.

The core of our integration strategy lies in the selective and parallel application of frameworks based on asset functionality. Unlike conventional approaches that limit framework scope, our methodology applies the STRIDE model across all identified assets to establish a consistent security baseline [17]. This ensures that fundamental security property violations—such as spoofing, tampering, and denial of service—are exhaustively addressed regardless of the asset type.

Building upon this baseline, specialized frameworks are layered according to the asset's archetype. For assets involving sensing and perception logic, the MITRE ATLAS

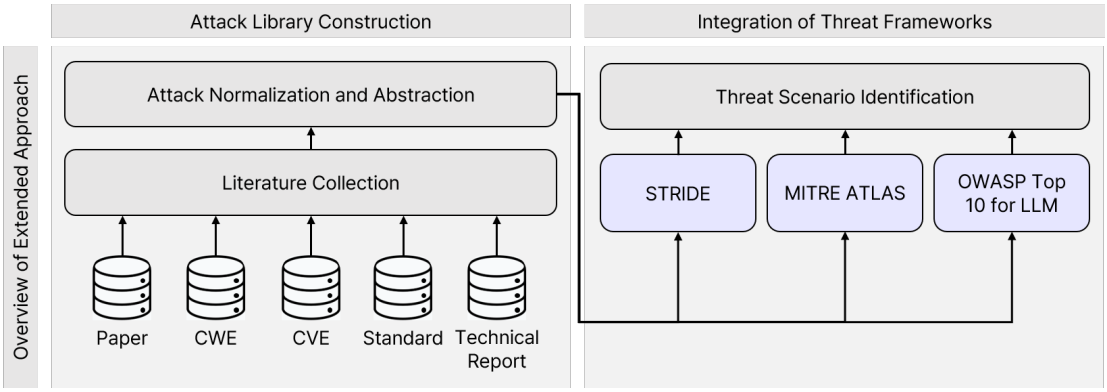


Figure 2: Extended Approach

framework [20] is integrated to identify adversarial threats, including sensor spoofing [21] and model evasion [22]. For assets characterized by reasoning or interaction via Large Language Models (LLMs), the OWASP Top 10 for LLM Applications [23] is additionally mapped to detect interface-specific risks like prompt injection [24, 25]. This modular strategy allows for a granular threat analysis that spans the entire perception-planning-control loop. Ultimately, these derived threat scenarios lead to structured Attack Path Analysis and the formulation of Security Requirements, with practical applicability validated against global regulatory standards such as UN R155 [10, 11].

3.2 Attack Library Construction

To ensure a comprehensive threat analysis for Physical AI systems, we construct a structured Attack Library by synthesizing multi-domain security knowledge. This process transforms fragmented vulnerability data into systematic attack scenarios through two phases: systematic collection and formal normalization [19].

3.2.1 Literature Collection

The collection process targets authoritative sources to capture the unique threat landscape of Physical AI, encompassing both traditional software vulnerabilities and AI-specific adversarial maneuvers. Initially, vulnerability databases such as CVE [18] and CWE are utilized to identify real-world exploitable weaknesses. To ensure practical applicability to the target architecture, the search strategy employs an approach that combines functional keywords (e.g., "Sensor Fusion", "Localization") with specific platform identifiers (e.g., "NVIDIA Jetson", "Linux Kernel") derived from the asset identification phase.

To address specialized AI threat vectors, taxonomies such as MITRE ATLAS [20] and OWASP for LLM [23] are integrated to capture techniques like data poisoning and model evasion. Furthermore, papers are analyzed to include state-of-the-art attacks validated in academic settings, such as sensor-spoofing [1, 21, 26] and perception-layer adversarial attacks [22, 27]. This systematic collection ensures that the library is grounded in both the theoretical rigor of AI security research and the practical reality of the deployed hardware and software stack.

3.2.2 Attack Normalization and Abstraction

The raw literature collected from diverse sources often contains redundant or overlapping descriptions. To transform this into a reusable and structured Attack Library, we perform normalization based on the core attributes of each threat [19]. We define an attribute vector for each record r as $Attr(r) = \langle Vector, Target, Consequence \rangle$. Within this framework, *Vector* represents the attack medium or technique (e.g., physical perturbation, signal injection), *Target* denotes the functional archetype of the victim asset (e.g., sensing interface, reasoning logic), and *Consequence* signifies the resulting system state change (e.g., integrity loss, excessive agency).

We define a normalization function $A_k = \text{Norm}(r_i, r_j)$, where two records are consolidated into a single representative attack scenario A_k if they share equivalent targets and consequences while employing technically similar vectors:

$$\begin{aligned} r_i \equiv r_j \iff & (Vector_i \approx Vector_j) \\ & \wedge (Target_i = Target_j) \\ & \wedge (Consequence_i = Consequence_j) \end{aligned} \tag{1}$$

Here, $\approx$ denotes technical similarity: the same physical or logical channel, the same DFD-level target archetype, and the same consequence category. Records are consolidated only when all three criteria are satisfied. For example, different physical artifacts that induce perception misclassification are consolidated because they share the same target and consequence.

3.3 Integration of Threat Frameworks

To systematically categorize and identify threats within the Physical AI environment, we adopt a multi-layered mapping strategy that integrates traditional and AI-specific security frameworks. This approach ensures that the analysis encompasses not only conventional software vulnerabilities but also the unique adversarial attack vectors inherent in autonomous systems [2, 28].

3.3.1 Mapping Attack Library to Frameworks

The entries in our constructed Attack Library are systematically mapped to three complementary frameworks: STRIDE [13], MITRE ATLAS [20], and the OWASP Top 10 for LLM Applications [23]. In this hierarchical mapping, every identified attack is first classified under the STRIDE model to establish its fundamental security property violation, such as Spoofing, Tampering, or Denial of Service [17].

However, as traditional models often fail to capture the nuances of AI-specific threats like data poisoning or model evasion [28,29], we further layer relevant attacks with MITRE ATLAS techniques to characterize adversarial behaviors. For assets characterized by Large Language Model (LLM) integration, we additionally apply the OWASP LLM Top 10 categories to address specialized vulnerabilities such as prompt injection and excessive agency [24,25]. This multi-framework alignment ensures that each threat is analyzed through multiple lenses, providing a comprehensive view of the physical AI risk landscape.

3.3.2 Threat Scenario Identification Strategy

Based on the integrated mapping, we derive detailed threat scenarios by applying specific frameworks to their corresponding asset archetypes defined in the DFD. To maintain analytical rigor while ensuring generality, we utilize the "STRIDE-per-element" approach across all system components [19]. This establishes a universal security baseline where even AI-specific assets are first evaluated for foundational vulnerabilities.

The identification of complex Physical AI scenarios is then refined by layering specialized frameworks onto specific functional archetypes. For *Perception Assets* responsible for environmental interpretation, we focus on Physical Environment Access techniques from ATLAS to identify scenarios such as signal injection or sensor blinding [1, 21, 26]. Simultaneously, for *Reasoning Assets* employing MLLM-based logic, we identify scenarios involving context contamination and system prompt leakage using the OWASP framework [23, 25]. This strategy enables the systematic derivation of multi-stage threat scenarios—such as a physical adversarial perturbation propagating through the perception layer to induce erroneous decision-making in the planner [22, 27]—which would be overlooked by conventional, single-framework TARA methods.

4 Case Study

To demonstrate the practical applicability and comprehensive coverage of our extended TARA methodology, we conduct a detailed case study on an Autonomous Driving System (ADS). The target system is modeled based on the logical architecture defined in ISO/TS 5083 [30] and implemented using the NVIDIA Jetson Platform Services software stack [31]. The case study proceeds through a systematic workflow aligned with the ISO/SAE 21434 lifecycle.

4.1 Asset Identification

We initiated the asset identification phase by constructing Data Flow Diagrams (DFD) using the Microsoft Threat Modeling Tool. To ensure the threat model possesses both architectural validity and implementation realism, we synthesized a reference architecture from two primary sources. First, the high-level functional logic was grounded in the ISO/TS 5083 standard. We specifically adopted the "Example architecture of the intended functionality" (Figure A.1) to define the core *Perception-Planning-Control* loop, which dictates how the ADS interprets sensor data and executes maneuvers.

However, a purely functional model lacks the granularity required to identify vulnerabilities in specific software components or underlying system layers. To bridge this gap, we instantiated the ISO model using the NVIDIA Jetson Platform Services as the concrete implementation baseline; the detailed software stack for this implementation is provided in Appendix A.2. The significance of this instantiation lies in the determinism of the operating system environment—NVIDIA Jetson Linux—which dictates the specific implementation of device drivers and hardware interfaces. By fixing the OS environment, we could define assets not as abstract logic, but as concrete entities whose security boundaries are governed by Jetson-specific kernel modules and proprietary drivers. This allows the TARA process to account for real-world constraints, such as the memory management behavior of NVIDIA proprietary GPU drivers and the specific interrupt handling of Camera Serial Interface drivers. Consequently, we identified and modeled specific Device Drivers and Shared Memory/IPC mechanisms as explicit assets. This mapping enables a systematic analysis of implementation-level vulnerabilities, such as GPU memory isolation failures (e.g., CVE-2023-4969) or UEFI Management mode side-channels (e.g., AL-V-74), which are inherent to the Jetson Linux stack. This approach ensures that the resulting threat model is a high-fidelity representation of the actual deployment stack, capturing risks that arise from the hardware-software interface.

The culmination of this modeling process is the Level 2 DFD presented in Figure A.3. This diagram provides a granular view of the internal data pathways, explicitly exposing the interactions between the AI algorithms and the system bus.

Through this decomposition, we successfully identified a total of 17 critical assets—comprising 10 processes and 7 external entities—along with 5 data stores and 30 distinct data flows. These identified elements serve as the definitive targets for the subsequent threat scenario identification phase.

4.2 Threat Scenario Identification

Following the methodology proposed in Section 3, threat scenarios were systematically identified for each asset in the ADS architecture by applying the constructed Attack Library and integrated threat frameworks.

4.2.1 Attack Library Construction Results

The literature collection process utilized keywords derived from the identified assets in the DFD. Representative keywords included "Sensor", "Sensor Fusion", "Localization", "LLM", "NVIDIA Jetson", "Jetson Driver", and "Linux Kernel", which were combined to query the five source categories described in Section 3.2.1. This targeted search resulted in the collection of 42 CWEs, 76 CVEs, 47 academic papers, 3 technical reports, and 2 standards.

Following the normalization and abstraction process described in Section 3.2.2, these 170 literature sources were analyzed to extract their attack vectors, targets, and consequences. Literature sources describing attacks with similar vectors and identical targets and consequences were consolidated into single attack scenarios. Through this process, the 170 literature sources were consolidated into 89 normalized attack scenarios, forming the unified Attack Library that serves as the foundation for threat scenario identification.

For example, AL-33 and AL-55 remain separate because their target archetypes differ, whereas sticker- and paint-based sign attacks are consolidated into AL-36 because they satisfy all three criteria.

4.2.2 Framework Mapping Results

Each attack in the constructed Attack Library was mapped to STRIDE, MITRE ATLAS, and OWASP Top 10 for LLM Applications according to the mapping strategy described in Section 3.3.1. All attacks were classified under at least one STRIDE category to capture traditional security property violations. Attacks targeting AI/ML assets were additionally mapped to MITRE ATLAS tactics and techniques, while attacks specific to LLM-integrated components were further classified under OWASP Top 10 categories.

Table 1 presents representative examples from the constructed Attack Library, illustrating how attacks are characterized by their reference sources and classified across multiple threat frameworks. The complete Attack Library is available in the public repository. [37]

4.2.3 Threat Scenario Derivation

Threat scenarios were derived by systematically applying the hierarchical strategy defined in Section 3.3.2. The DFD for the ADS architecture contains 10 processes, 7 external entities, 5 data stores, and 30 data flows, each of which was analyzed for applicable threats based on their component type and the mapped Attack Library.

STRIDE per Element Application We applied the STRIDE per Element framework to all identified assets in the DFD to systematically derive potential threat scenarios. As summarized in Table 2, each DFD component type is susceptible to a specific subset of the STRIDE threat categories based on its functional role within the system.

For each asset, attacks from the Attack Library matching the applicable STRIDE categories were selected to derive threat scenarios. For example, for the External Entity E1.1 (GPS/IMU), attacks classified as Spoofing (S) or Repudiation (R) were mapped to generate threat scenarios such as "GNSS spoofing causing contamination of position, velocity, and time data" and "Failure in post-incident actor identification due to lack of logging for GNSS/IMU anomalies."

Table 3 presents examples of threat scenarios derived through STRIDE for representative assets including external entities and OS-level processes. Through STRIDE per Element application, baseline threat scenarios covering traditional software and network vulnerabilities were systematically identified for all assets in the ADS architecture.

MITRE ATLAS Application for AI Assets For AI-related assets that perform perception, prediction, and planning functions, MITRE ATLAS was additionally applied to capture adversarial threats specific to AI/ML systems. These assets—including P2.2.3 (Interpretation & Prediction) and P2.3 (Planning)—rely on DNN-based inference and are therefore exposed to attacks such as adversarial examples, data poisoning, and model extraction that fall outside the scope of traditional STRIDE categories.

Attack Library entries classified under ATLAS tactics were mapped to these AI assets according to their relevance. For instance, attacks categorized as "Physical Environment Access" were applied to perception assets vulnerable to sensor-level manipulation, while "Poison Training Data" attacks were mapped to assets where model integrity could be compromised during the training phase.

Table 4 illustrates AI-specific threat scenarios identified for the P2.2.3 asset using MITRE ATLAS. Examples include "LiDAR point cloud injection inducing phantom object detection," "Traffic sign misclassification via physical adversarial stickers," and "DNN misclassification via FGSM/PGD adversarial examples." Through ATLAS application, comprehensive threat scenarios covering the adversarial threat landscape

Table 1: Examples : Attack Library

ID	Attack	Ref.	STRIDE	ATLAS	OWASP
AL-33	Injecting laser points to manipulate LiDAR point clouds, generating non-existent obstacles and misleading DNN-based object detectors.	[21]	S	Physical Environment Access	N/A
AL-36	Applying graffiti or stickers to a STOP sign to perform a physical adversarial attack that causes computer vision systems to misclassify the traffic sign.	[22] [32]	T	Physical Environment Access	N/A
AL-43	Executing indirect prompt injection by embedding hidden instructions in external content to manipulate the behavior of an LLM.	[24]	T	LLM Prompt Injection	LLM01 Prompt Injection
AL-51	CAN bus flooding attack — causing denial of service by monopolizing the bus using high-priority message IDs.	[33]	D	N/A	N/A
AL-52	Forged CAN message injection — remotely controlling vehicle functions via compromised telematics or gateway ECUs by injecting spoofed CAN frames.	[34] [35]	S	N/A	N/A
AL-55	Inaudible ultrasonic command injection — sending ultrasonic signals to issue hidden commands and manipulate voice assistants.	[36]	S	Physical Environment Access	N/A

Table 2: STRIDE per Elements

Threat	Data Flows	Data Stores	Processes	External Entities
Spoofing			✓	✓
Tampering	✓	✓	✓	
Repudiation			✓	✓
Information Disclosure	✓	✓	✓	
Denial of Service	✓	✓	✓	
Elevation of Privilege			✓	

of AI perception and planning modules were systematically derived.

OWASP Top 10 Application for LLM Assets For LLM-integrated assets, specifically P2.6.2 (MLLM), OWASP Top 10 for LLM Applications was employed to address vulnerabilities unique to large language models. The MLLM asset involves natural language interaction, multimodal context processing, and external knowledge retrieval via Retrieval-Augmented Generation (RAG), introducing attack surfaces not adequately covered by STRIDE or ATLAS alone.

Attack Library entries classified under OWASP Top 10 categories were mapped to the MLLM asset. Representative threat categories include Prompt Injection, which encompasses both direct and indirect manipulation of model behavior through crafted inputs; System Prompt Leakage, involving extraction of internal instructions that govern model behavior; Excessive Agency, where insufficient privilege constraints enable unauthorized actions; and Vector and Embedding Weaknesses, which cover attacks exploiting the semantic representation space of LLMs.

Table 5 demonstrates LLM-specific threat scenarios derived from OWASP Top 10 for the P2.6.2 asset. Examples include "Indirect prompt injection via ingestion of untrusted external content," "Malicious fine-tuning removing RLHF-based safety guardrails," and "Sensitive data reconstruction via embedding inversion attacks." Through OWASP Top 10 application, the unique threat landscape of LLM-integrated decision-making components was comprehensively captured.

4.2.4 Threat Scenario Identification Result

Through the hierarchical application of STRIDE, MITRE ATLAS, and OWASP Top 10 for LLM, a comprehensive set of threat scenarios was systematically identified across the ADS architecture. The methodology successfully captured threats spanning traditional IT security concerns, AI-specific adversarial attacks, and emerging LLM vulnerabilities, providing a holistic view of the Physical AI threat landscape.

The complete set of identified threat scenarios, including detailed descriptions and framework mappings, is available in the public repository. [37] These threat scenarios serve as the foundation for subsequent Impact Rating (Section 4.3) and Attack Path Analysis (Section 4.4), ultimately leading to the derivation of security requirements and actionable checklists for ADS development and deployment.

Table 3: Example : Threat Scenarios for the OS Asset based on STRIDE

Asset	No.	Threat Scenario	STRIDE	AL Ref.
P2.1. OS	T94	Corruption of kernel data and memory structures by exploiting vulnerabilities in the Linux CAN Broadcast Manager.	T	AL-62
	T105	Modification of configuration, log, or module files through Path Traversal vulnerabilities.	T	AL-29
	T106	Corruption of kernel data and memory structures by exploiting vulnerabilities in the Linux CAN Broadcast Manager.	T	AL-16

Table 4: Example : Threat Scenarios for the Planning Asset based on MITRE ATLAS

Asset	No.	Threat Scenario	ATLAS	AL Ref.
P2.3. Planning	T558	Induction of maladaptive behaviors (e.g., abrupt braking, lane deviation) in path planning algorithms via training data poisoning.	Poison Training Data	AL-40
	T559	Replication of the internal planning model and reverse engineering of vulnerabilities via exploitation of the model inference API.	Exfiltration via AI Inference API	AL-60
	T560	Induction of erroneous decision-making in the AI planning system via manipulation of external visual semantics.	Physical Environment Access	AL-36

4.3 Impact Rating

Following the identification of threat scenarios, we evaluated the potential adverse consequences on road users and ADS stakeholders. As stated in our scope, we do not perform a numerical scoring of the impact levels (e.g., Negligible to Severe), as such ratings are highly subjective and contingent upon specific organizational risk appetites and deployment contexts. Instead, we focus on the systematic derivation of damage scenarios and their qualitative categorization (Safety, Financial, Operational, and Privacy) to establish a clear causal link between AI-specific threats and vehicle-level hazards.

4.3.1 Damage Scenario Derivation

The damage scenarios were derived by analyzing how each identified threat (from STRIDE, ATLAS, and OWASP) propagates through the ADS data flow to compromise the system's primary goals. As documented in our artifacts, we identified a total of 165 damage scenarios (DS-1 to DS-165). Each scenario represents a physical or operational failure resulting from one or more threat scenarios.

For instance, the damage scenario "Incorrect path planning and lane departure" (DS-1) is not merely a software bug but a consequence of a specific chain of AI threats: GNSS spoofing (T1, T2) or authentication bypass of location data (T8). Similarly, the "Unintended braking" (DS-4) scenario is mapped to perception-layer threats such as ghost object detection via sensor spoofing (T11-T19). This mapping ensures that every identified damage has a traceable root cause in our expanded Attack Library.

4.3.2 Impact Categories

In accordance with ISO/SAE 21434, each damage scenario was categorized based on its impact domain:

- **Safety (S):** Scenarios leading to physical hazards, such as collisions due to manipulated object tracks (DS-157).
- **Financial (F):** Impacts related to liability and economic loss, often linked to the failure of non-repudiation and logging mechanisms (DS-3).
- **Operational (O):** Disruption of ADS availability, such as positioning service outages (DS-2).
- **Privacy (P):** Leakage of sensitive data, particularly relevant in MLLM-based systems where user intent or biometric data can be exposed (DS-165).

4.3.3 Impact Rating Results

Table 6 presents the representative mapping between assets, damage scenarios, and their associated threat scenarios. The analysis reveals that Physical AI threats (e.g., LiDAR/Radar spoofing) predominantly converge on Safety impacts, while LLM-specific threats (e.g., Prompt Injection) introduce a complex mix of Privacy and Operational risks, highlighting the necessity of the integrated TARA approach proposed in this paper.

The analysis revealed that while AI-specific threats (ATLAS, OWASP) often originate in the digital domain, their

Table 5: Example : Threat Scenarios for the MLLM Asset based on OWASP Top 10 for LLMs

Asset	No.	Threat Scenario	OWASP	AL Ref.
P2.6.2. MLLM	T536	Exposure of the System Prompt (internal constraints and instructions), enabling adversaries to reverse-engineer and circumvent safety policies.	System Prompt Leakage	AL-46
	T537	Ablation of RLHF-based safety guardrails via malicious Fine-tuning, aimed at eliciting restricted or harmful outputs.	Data and Model Poisoning	AL-49
	T538	Autonomous execution of high-risk commands resulting from the exploitation of Excessive Agency and insufficient scope limitation.	Excessive Agency	AL-45

propagation to the physical domain (Safety Impact) is significant in ADS, justifying the need for high-integrity protection measures in the subsequent Risk Treatment phase.

4.4 Attack Path Analysis

An attack path is defined as a series of deliberate actions undertaken to realize a mapped threat scenario [12]. In this study, attack paths were derived by establishing the previously identified threat scenarios and damage scenarios as the respective endpoints, and detailing the specific attack actions connecting them based on the Attack Library. Specifically, the construction of these attack paths comprehensively accounted for asset characteristics, adversary intent, and the technical means required for execution.

First, we identified the preconditions and entry points an adversary must secure to realize the threat scenario mapped to each asset. Next, we ensured the technical realization of the attacks by integrating specific techniques from the Attack Library—which originally grounded the derivation of these threat scenarios—as the core execution steps along the path. Finally, we completed the causal chain linking these attack actions to the ultimate damage scenarios by analyzing their functional impact on the system. Table 7 presents examples of the attack paths derived for each threat scenario through this analytical process. The comprehensive set of attack paths can be accessed via our public repository. [37]

4.5 Derivation of Security Requirements

In the standard ISO/SAE 21434 TARA process, the risk treatment phase determines the appropriate mitigation strategy for each identified threat scenario based on an organization's specific evaluation criteria and risk appetite. When the "Reducing the Risk" option is selected, technical and organizational controls must be established, which necessitates the derivation of precise security requirements. These requirements subsequently serve as the foundational elements for defining the overarching Cybersecurity Goals in the later stages of the engineering lifecycle.

However, consistent with our methodological scope—which abstracts away organization-specific risk evaluation criteria—we bypass the subjective risk treatment decision step. Instead, we proactively derive security requirements for *all* identified threat scenarios to ensure comprehensive coverage against the expanded threat landscape.

The security requirements are systematically synthesized based on the causal chain established in the previous phases: the threat scenarios, damage scenarios, attack paths, and the underlying Attack Library mapped to each asset. To optimize the mitigation strategy and prevent redundancy, threat scenarios that necessitate identical protective measures are clustered and mapped to a unified security requirement. This consolidation process not only streamlines the defense architecture but also provides the foundational artifacts necessary for constructing a comprehensive security checklist for Physical AI systems.

Table 8 presents representative examples of the derived security requirements. The complete set of requirements is accessible via our public repository. [37]

4.6 Derivation of Checklist

In this section, we consolidated the security requirements derived across all assets by performing systematic merging, normalization, and deduplication. This process ensures technical consistency and culminated in the final derivation of a comprehensive security checklist comprising 38 distinct items. Representative items from the checklist are presented in Appendix A.4; the complete 38-item checklist is available in our artifact [37].

The primary objective of this transformation is to operationalize abstract security requirements into concrete, quantifiable verification criteria, thereby eliminating ambiguity during the system implementation phase. Furthermore, this checklist is designed to serve as objective, verifiable evidence demonstrating the practical application of security controls during future CyberSecurity Management System (CSMS) audits and Vehicle Type Approval (VTA) processes.

Table 6: Representative Damage Scenarios and Impact Categories

Asset	No.	Damage Scenario	CIA	SFOP	Threats ID
P2.1. OS	DS-27	Collapse of security boundaries via privilege escalation or isolation bypass, resulting in persistent manipulation, data exfiltration, and the neutralization of safety functions, ultimately causing hazardous driving or collisions.	I/A	S	T90, T91, T92, T93, T94, T105, T106, T107, T108
	DS-30	Exposure of sensitive data resulting in privacy violations, security breaches, and regulatory non-compliance.	C	P	T97, T98, T99, T100
P2.3. Planning	DS-47	Imminent risk of collision or pedestrian injury caused by erratic maneuvers (e.g., inappropriate braking, abrupt steering, lane departure).	I/A	S	T558, T560, T561, T565, T567, T568, T571
	DS-50	Compromise of user privacy facilitating unauthorized surveillance, tracking, and targeted social engineering attacks.	C	P	T559, T563, T566, T572
P2.6.2. MLLM	DS-68	Induction of malformed vehicle control messages or invalid authorization codes via the generation of hazardous commands by the MLLM.	I	S	T576, T577, T578, T581, T582, T583, T592, T534, T535, T538, T543
	DS-70	Execution of restricted operations resulting from the circumvention of safety guardrails and misuse of privileges.	I	O	T573, T574, T584, T593, T537, T541

5 Validation

In this section, we validate the effectiveness and applicability of the proposed extended TARA methodology through four distinct dimensions: (1) Regulatory Compliance, demonstrating alignment with UN Regulation No. 155; (2) Comparative Analysis, quantifying improvements over traditional threat modeling; (3) Maturity Assessment, evaluating the data-driven robustness of the process; and (4) Real-World Validation, establishing institutional accreditation as a basis for accredited testing activities.

5.1 Legal Coverage

The derived security checklists must transcend academic theoretical proposals and serve as practical guidelines in the industrial field. To verify this applicability, we analyzed the coherence between our 38 derived checklists and the technical requirements of UN Regulation No. 155 (UN R155), the mandatory international standard for vehicle cybersecurity.

Validation Criteria: Annex 5. We selected UN R155 Annex 5 as the baseline for validation. The regulation mandates that manufacturers consider the threats listed in Annex 5 Part A and the mitigations in Part B and C during risk assessment. Therefore, coverage of Annex 5 Part A by the ADS case-study checklists indicates that the methodology can generate artifacts useful for demonstrating regulatory alignment.

Mapping and Conformity Analysis. We performed a semantic mapping between the derived checklists and the sub-level threats defined in Annex 5 Part A. The results in Table B.1 confirm that the ADS case-study checklists cover both general embedded-system threats and AI- and sensor-specific threats in Annex 5 Part A.

The 38 ADS case-study checklists collectively cover all 32 Annex 5 Part A threat categories, including emerging AI-related vulnerabilities. As illustrated in the mapping results, the derived checklists correspond to the detailed threat categories of Annex 5. For instance, checklists addressing "Sensor Spoofing" and "Adversarial Examples" directly map to the regulation's requirements for verifying sensor data integrity. This mapping indicates that the ADS case-study checklists can support compliance evidence for Vehicle Type Approval (VTA) audits and Cyber Security Management System (CSMS) assessments.

5.2 Comparative Analysis

To evaluate the improvement of the proposed methodology, we conducted a comparative analysis against the traditional STRIDE-based TARA approach using the same ADS architecture. Per Section 2.4, STRIDE-per-element is the dominant baseline in industrial ISO/SAE 21434 TARA practice, making it the most realistic comparison target for evaluating practical improvements.

Qualitative Analysis: Granularity of Threats. The primary distinction lies in the granularity and scope of threat identification. While the traditional STRIDE model is effective for identifying software and network vulnerabilities based on data flow, it exhibits structural limitations in characterizing threats related to AI inference and physical sensor data. For

Table 7: Representative Attack Paths Mapping Threat Scenarios to Damage Scenarios

Asset	Damage Scenario	Threat Scenario	Attack Path
P2.1. OS	DS-27	T105	i. Reconnaissance: The attacker analyzes the system's privilege management policies and default configurations. ii. Vulnerability Discovery: Excessive default privileges or incorrect permission assignments are identified. iii. Exploitation (Escalation): The attacker utilizes privilege escalation techniques from a low-integrity process to acquire administrative (root) privileges [AL-29]. iv. Action on Objectives: With elevated privileges, the attacker bypasses all security boundaries, neutralizes safety systems, and establishes remote control over the vehicle.
P2.3. Planning	DS-47	T560	i. The attacker modifies the physical environment with adversarial stickers or paint. ii. These visual changes alter the semantic meaning perceived by the AI [AL-36]. iii. The planner generates a trajectory that follows the manipulated semantics (e.g., driving into a wall).
P2.6.2. MLLM	DS-68	T537	i. The attacker gains access to the model's fine-tuning pipeline. ii. A malicious dataset designed to reverse safety alignment is used for fine-tuning [AL-49]. iii. The deployed model unlearns its safety constraints and executes previously restricted commands.

Table 8: Representative Security Requirements mapped to Threat Scenarios

Asset	ID	Security Requirement	Threats ID
P2.1. OS	SR-37	A hardware-rooted Secure Boot chain shall be implemented to verify the integrity of the bootloader and kernel.	T92, T103, T105, T108
P2.3. Planning	SR-62	Adversarial defense mechanisms and input sanitization shall be implemented to detect semantic perturbations and prompt injections.	T169, T172, T560, T561, T565, T567, T569
P2.6.2. MLLM	SR-89	Supply chain security controls including model signing and backdoor scanning shall be applied to all AI artifacts.	T248, T259, T260, T537, T541, T573, T574, T584, T593

example, under STRIDE, a "Poisoned Model" or an "Adversarial Example" is often broadly categorized as generic "Tampering" or "Information Disclosure," failing to distinguish the specific attack vector. In contrast, our extended methodology, by integrating MITRE ATLAS and OWASP Top 10 for LLM, successfully decomposed these vague categories into specific Physical AI threat scenarios such as *Evasion*, *Model Inversion*, and *Prompt Injection*.

Quantitative Analysis: Expanded Coverage. The quantitative results further highlight the efficacy of the proposed approach. Both methods identified standard cybersecurity threats such as network eavesdropping and unauthorized access. However, the proposed methodology exclusively identified specialized threats related to **AI & Sensor Data**, involving sensor fusion manipulation and model bias (e.g., CL-19, CL-20, CL-25 to CL-27), and **LLM Integration**, involving prompt injection and context poisoning (e.g., CL-24, CL-28, CL-29, CL-31 to CL-33). The proposed methodology exclusively identified 11 AI-specific checklist items that could not be derived through conventional STRIDE-based TARA alone—accounting for approximately 29% (11 out of 38) of the total derived checklists. These items address threat categories that are structurally absent from traditional software-centric frameworks, including sensor fusion manipulation, model bias exploitation, adversarial perturbation attacks, prompt injection, and LLM context poisoning. The fact that nearly one-third of the security controls emerged only through the integration of MITRE ATLAS and OWASP Top 10 for LLM Applications underscores a critical blind spot in conventional TARA methods when applied to Physical AI environments: a system analyzed solely with STRIDE would have left these attack surfaces unaddressed, potentially resulting in incomplete security coverage and non-compliance with emerging regulatory expectations for AI-integrated vehicles.

5.3 Maturity Assessment

This section evaluates the data integration capabilities and methodological maturity of the proposed approach by applying the Data-Driven TARA Maturity Model (D2TARA) proposed by Vielberth et al. [38]. The D2TARA model assesses the Capability Level (CL) of a TARA process based on the extent to which external data sources are standardized and integrated into the workflow. Accordingly, this assessment is limited to data-source standardization and integration, not organizational maturity.

Standardization (CL 2). We established a comprehensive Standard Pool of External Data Sources by systematically constructing an Attack Library that incorporates CVE, CWE, Papers, Technical Reports, and Standards. This collection strictly satisfies the requirements for *Standardization (CL 2)* across all data-driven TARA sections.

Integration (CL 3). The D2TARA model evaluates maturity based on data integration across five core TARA sections: *DAMAGE*, *IMPACT*, *ATTACK*, *FEASIB*, and *RISK TREAT*. [38] According to expert feedback documented in the D2TARA validation study (Section 6.2), there exists justified disagreement on whether *IMPACT* and *FEASIB* should be considered data-driven sections due to the discrepancy between ISO/SAE 21434 rating schemes and externally available metrics such as CVSS. [38] Following this expert perspective, we focus our maturity assessment on the three universally recognized data-driven sections: *DAMAGE*, *ATTACK*, and *RISK TREAT*.

Our methodology achieves CL 3 (Integration) by establishing explicit processes for utilizing external data sources in each section. The foundation of this integration is the Attack Library construction process, which systematically analyzes CVE, CWE, Papers, technical reports, and standards to extract three core attributes: Vector (attack medium and techniques), Target (affected assets), and Consequence (system state changes). Attacks sharing similar Vectors, identical Targets, and equivalent Consequences are classified into unified attack categories, which are then mapped to the taxonomies of OWASP Top 10 for LLM, STRIDE, and MITRE ATLAS.

For the **DAMAGE** section, damage scenarios are derived through a structured integration process. Threat scenarios are generated using STRIDE per element, OWASP categories for MLLM assets, and MITRE ATLAS for AI assets. Since each threat scenario is mapped to specific Attack Library entries through the framework alignment process, the resulting damage scenarios inherently incorporate the Consequence attributes from CVE impact descriptions and technical report findings. This ensures that damage identification is not based on expert judgment alone but is systematically grounded in documented external data sources.

The **ATTACK** section demonstrates clear integration through attack path construction. For each threat scenario, attack paths are analyzed by leveraging the Attack Library's Vector and Target attributes. Because threat scenarios are explicitly mapped to Attack Library entries, the attack path analysis directly incorporates attack techniques documented in CVE/CWE databases and MITRE ATLAS tactics. This mapping mechanism ensures that attack feasibility assessment is traceable to specific external data sources rather than being purely expert-driven.

For the **RISK TREAT** section, mitigation strategies are systematically derived from the attack paths through framework-based guidance. The integrated OWASP Top 10 for LLM and MITRE ATLAS frameworks provide structured mitigation recommendations that are mapped to security requirements. Since attack paths are connected to specific framework categories through the Attack Library mapping process, the risk treatment decisions are inherently informed by the mitigation strategies documented in these external sources. This integration ensures consistent and comprehensive risk treatment based on industry-recognized countermeasures.

Extended Maturity Level (ML 3). According to the D2TARA maturity level definitions, ML 3 (Extended) is achieved when CL 3 (Integration) is achieved in all data-driven TARA sections. Our methodology demonstrably achieves this through the systematic integration described above. The Attack Library serves as the integration hub, transforming raw external data sources into structured attributes (Vector, Target, Consequence) that flow through the entire TARA process. Threat scenarios are generated through framework mapping, damage scenarios inherit Consequence attributes, attack paths are constructed from Vector/Target information, and risk treatment leverages framework-embedded mitigation guidance. This end-to-end integration of external data sources into each TARA section fulfills the CL 3 requirement that "the pool of data sources is integrated into the TARA process" with explicit specification of how data is utilized. Therefore, the proposed methodology achieves **Extended Maturity Level (ML 3)** for data-source integration under the D2TARA model.

5.4 Real-World Validation: KOLAS Accreditation

Beyond the analytical validation dimensions presented above, the proposed methodology has been adopted as the technical procedure underlying accreditation by the Korea Laboratory Accreditation Scheme (KOLAS) as an internationally recognized testing laboratory for the ISO/SAE 21434 TARA process. This means that test reports produced through the framework can serve as formally recognized deliverables in real-world automotive cybersecurity testing practice. As KOLAS is a full signatory to the ILAC MRA and APAC MRA, this places reports produced through the methodology within an internationally acknowledged conformity assessment framework.

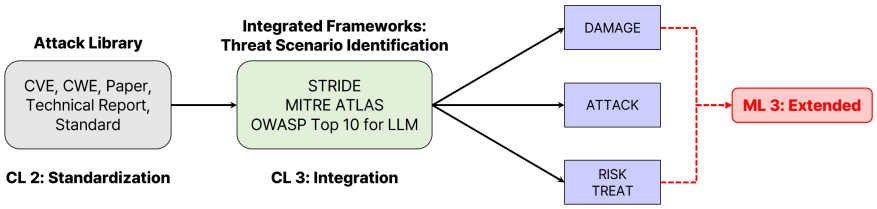


Figure 3: Proposed TARA Maturity Validation Workflow

6 Discussion

This study demonstrates that integrating STRIDE, MITRE ATLAS, and OWASP Top 10 for LLM into the ISO/SAE 21434 TARA process enables systematic and comprehensive threat identification for Physical AI systems—an area where traditional software-centric frameworks have proven structurally insufficient.

6.1 Industrial and Regulatory Significance

The proposed methodology carries direct implications for the automotive industry. As UN Regulation No. 155 mandates CSMS establishment and Vehicle Type Approval (VTA) across the entire vehicle lifecycle, manufacturers face increasing pressure to demonstrate systematic cybersecurity coverage. Our framework provides a structured, auditable process. In the ADS case study, the resulting 38 security checklists cover all 32 UN R155 Annex 5 Part A threat categories and can support regulatory audits.

6.2 Scope of Risk Assessment

A deliberate limitation of this study is the exclusion of the Attack Feasibility Rating and Risk Value Determination phases from the ISO/SAE 21434 process. These phases are highly dependent on an organization's specific risk appetite, available resources, and deployment context, making universal prescriptions impractical. While our framework comprehensively identifies and catalogs threats, practitioners adopting this methodology must define their own feasibility criteria—considering factors such as the attacker's required window of opportunity and equipment cost—to derive final risk values and prioritize mitigation efforts accordingly.

6.3 Extensibility to Broader Physical AI

Although validated on ADS, the proposed methodology exhibits high transferability to broader Physical AI domains. However, effective application to other domains requires two domain-specific adaptations: the Asset Identification phase must be tailored to the target system's physical architecture, and the Attack Library must be extended with domain-specific attack vectors. For instance, applying this framework to industrial robots would necessitate incorporating sensor spoofing techniques specific to robotic arms and manufacturing floor environments. Future work will aim to validate this transferability by applying the methodology to diverse intelligent mobility platforms beyond ADS.

7 Conclusion

This paper proposed an extended TARA methodology aligned with ISO/SAE 21434 that applies STRIDE, MITRE ATLAS, and OWASP Top 10 for LLM in a multi-layered manner to address the cyber-physical threat surface of Physical AI systems. Through an ADS case study based on the ISO/TS 5083 logical architecture and implemented on the NVIDIA Jetson platform, the framework produced 38 actionable security checklists validated against UN Regulation No. 155 Annex 5.

The contributions of this work are threefold: (i) an integrated Physical AI TARA methodology unifying three complementary security frameworks; (ii) an Attack Library synthesized from CVEs, CWEs, and academic literature, providing a structured knowledge base for cyber-physical attack path analysis; and (iii) 38 actionable security checklists demonstrating direct regulatory compliance utility for VTA and CSMS certification processes.

This work has two main limitations: the exclusion of Attack Feasibility Rating and Risk Value Determination leaves threat prioritization to adopting organizations, and the current validation is scoped to ADS on the NVIDIA Jetson platform, requiring domain-specific adaptation for broader Physical AI generalization.

Future work will pursue two complementary directions: incorporating Attack Feasibility Rating and Risk Value Determination to complete the full ISO/SAE 21434 TARA lifecycle, and applying the methodology to diverse Physical AI platforms such as autonomous robotics. AI-agent–based automation of the proposed methodology also remains a promising direction.

8 Acknowledgments

This work was supported by the Ministry of Trade, Industry and Resources (MOTIR) (No. RS-2025-13732975)

References

[1] N. Shamsi, K. Chandrasekar, Y. Long, C. Limbach, K. Rebello, and K. Fu. Wip: threat modeling laser-induced acoustic interference in computer vision-assisted vehicles. In *Symposium on Vehicles Security and Privacy (VehicleSec)*, 2024.

[2] S. Ghosh, A. Zaboli, H. Junho, and J. Kwon. An integrated approach of threat analysis for autonomous vehicles perception system. *IEEE Access*, 11, 2023.

[3] A. Chattopadhyay, K.-Y. Lam, and Y. Tavva. Autonomous vehicle: Security by design. *IEEE Transactions on Intelligent Transportation Systems*, 22(11):7015–7029, 2021.

[4] S. M. Khalil, H. Bahsi, H. O. Dola, T. Korõtko, K. McLaughlin, and V. Kotkas. Threat modeling of cyber-physical systems: A case study of a microgrid system. *Computers & Security*, 124:102950, 2023.

[5] A. Chattopadhyay and K.-Y. Lam. Security of autonomous vehicle as a cyber-physical system. In *Proceedings of the 7th International Symposium on Embedded Computing and System Design (ISED)*, pages 1–6, 2017.

[6] A. A. Ali, S. Krayem, B. Chramcov, and M. F. Kadi. Self-stabilizing fault tolerance distributed cyber physical systems. In *Proceedings of the 29th DAAAM International Symposium*, pages 1173–1180, 2018.

[7] F. V. Jedrzejewski, D. Fucci, and O. Adamov. Thremolia: Threat modeling of large language model-integrated applications. *arXiv preprint arXiv:2504.18369*, 2025.

[8] F. V. Jedrzejewski. Threat modeling of ml-intensive systems: Research proposal. In *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering – Software Engineering for AI*, pages 264–266, 2024.

[9] S. Dellersnyder and Toreon. Mind the Gap: STRIDE-AI - Your Clear Path to Understanding AI Vulnerabilities. https://www.toreon.com/stride-ai-your-clear-path-to-understanding-ai-vulnerabilities, 2025.

[10] UNECE WP.29. UN Regulation No. 155 – Cyber security and cyber security management system, 2021. United Nations Economic Commission for Europe.

[11] S. Lee and S. Cho. Attack scenarios based on unece automotive cybersecurity regulation (un r.155). *Transactions of KSAE*, 29(8):717–732, 2021.

[12] ISO/SAE International. Road vehicles – Cybersecurity engineering (ISO/SAE 21434:2021), 2021. Standard.

[13] S. Hernan, S. Lambert, T. Ostwald, and A. Shostack. Uncover security design flaws using the stride approach. *MSDN Magazine*, 2006.

[14] Henniger, O., Ruddle, A., Seudié, H., Weyl, B., Wolf, M., Wollinger, T. Securing vehicular on-board it systems: the evita project. 2009.

[15] Judith E. Y. Rossebo, Scott Cadzow, and Paul Sijben. etvra, a threat, vulnerability and risk assessment method and tool for eeurope. In *The Second International Conference on Availability, Reliability and Security (ARES'07)*, pages 925–933, 2007.

[16] Moranana. M.M. UcedaVelez. T. *Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis*. 1st ed. John Wiley & Sons, 2015.

[17] Z. Abuabed, A. Alsadeh, and A. Taweel. Stride threat model-based framework for assessing the vulnerabilities of modern vehicles. *Computers & Security*, 133:103391, 2023.

[18] MITRE Corporation. CVE. https://cve.mitre.org/cgi-bin/cvename.cgi, 2025.

[19] A. Shostack. *Threat Modeling: Designing for Security*. Wiley, 2014.

[20] MITRE Corporation. MITRE ATLAS (Adversarial Threat Landscape for Artificial-Intelligence Systems). https://atlas.mitre.org/, 2021.

[21] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao. Adversarial sensor attack on lidar-based perception in autonomous driving. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19)*, pages 2267–2281, 2019.

[22] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634, 2018.

[23] OWASP Gen AI Security Project. OWASP Top 10 for LLM Applications 2025. https://genai.owasp.org/llm-top-10/, 2025.

[24] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec '23)*, pages 79–90, 2023.

[25] P. Zambare, V. N. Thanikella, and Y. Liu. Securing agentic ai: Threat modeling and risk analysis for a network monitoring agentic ai system. *arXiv preprint arXiv:2508.10043*, 2025.

[26] B. Nassi, Y. Mirsky, D. Nassi, R. Ben-Netanel, O. Drokin, and Y. Elovici. Phantom of the adas: Securing advanced driver-assistance systems from split-second phantom attacks. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20)*, pages 293–308, 2020.

[27] H. Zhou, W. Li, Z. Kong, J. Guo, Y. Zhang, B. Yu, L. Zhang, and C. Liu. Deepbillboard: Systematic physical-world testing of autonomous driving systems. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering (ICSE '20)*, pages 347–358, 2020.

[28] A. Vassilev, A. Oprea, A. Fordyce, H. Anderson, X. Davies, and M. Hamin. Adversarial machine learning: A taxonomy and terminology of attacks and mitigations (nist ai 100-2e2025). Technical report, National Institute of Standards and Technology (NIST), 2025.

[29] M. Ozdag. Adversarial attacks and defenses against deep neural networks: A survey. *Procedia Computer Science*, 140:152–161, 2018.

[30] ISO International. Road vehicles – Safety for automated driving systems – Design, verification and validation (ISO/TS 5083:2025), 2025. Technical Specification.

[31] NVIDIA Corporation. Jetson Platform Services documentation. https://docs.nvidia.com/jetson/jps/moj-overview.html, 2025.

[32] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramèr, A. Prakash, T. Kohno, and D. Song. Physical adversarial examples for object detectors. In *Proceedings of the 12th USENIX Conference on Offensive Technologies (WOOT '18)*, 2018.

[33] S.-H. Chen and C.-H. R. Lin. Evaluation of dos attacks on vehicle can bus system. In *Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2018)*, volume 110, pages 308–314, 2018.

[34] MITRE Corporation. CVE-2015-5611. https://cve.mitre.org/cgi-bin/cvename.cgi, 2025.

[35] MITRE Corporation. CVE-2016-9337. https://cve.mitre.org/cgi-bin/cvename.cgi, 2025.

[36] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*, pages 103–117, 2017.

[37] J. Woo. ADS-Threat-Modeling GitHub Repository. https://github.com/ddiyoung/ADS-Threat-Modeling, 2026. GitHub repository.

[38] Manfred Vielberth, Kristina Raab, Magdalena Glas, Patrick Grümer, and Günther Pernul. Elevating tara: A maturity model for automotive threat analysis and risk assessment. In *Proceedings of the 19th International Conference on Availability, Reliability and Security*, ARES '24, New York, NY, USA, 2024. Association for Computing Machinery.

A Appendix : Case Study

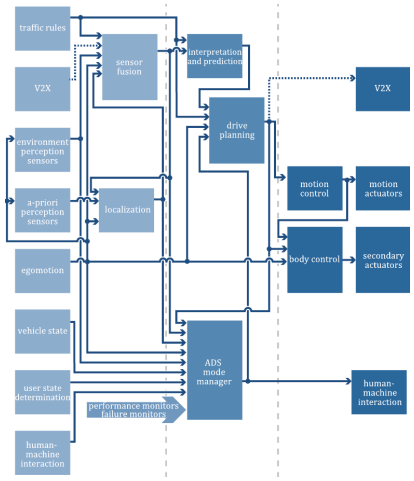


Figure A.1: Functional Architecture based on ISO/TS 5083

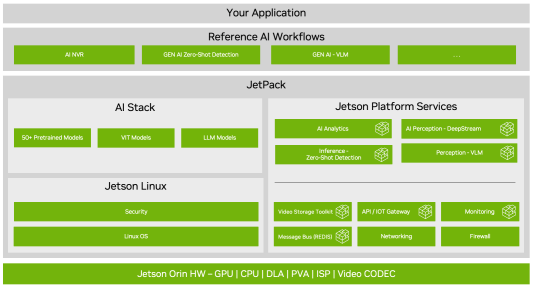


Figure A.2: NVIDIA Jetson Platform Services Software Stack

B Appendix : Validation

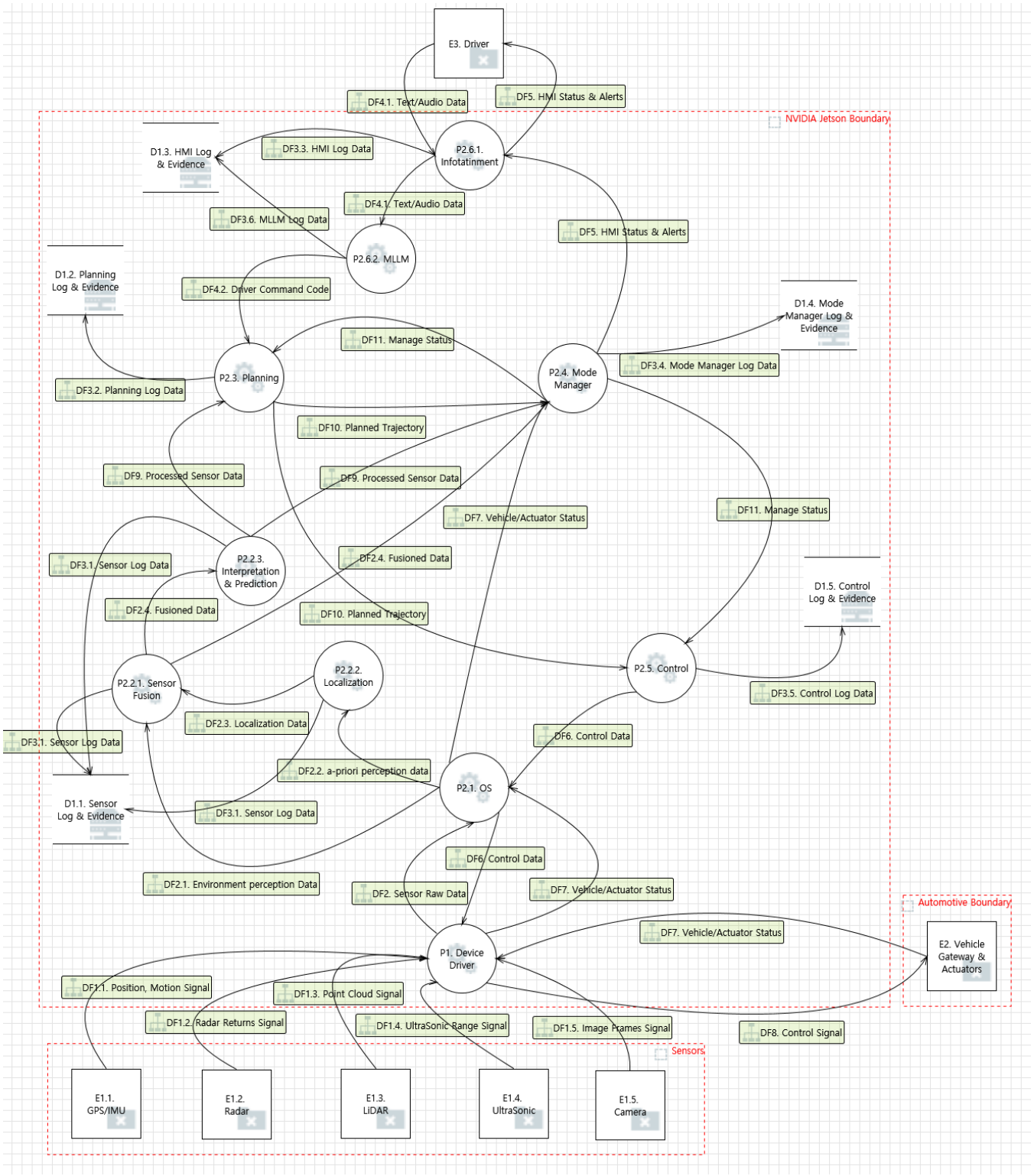


Figure A.3: Level 2 Data Flow Diagram (DFD)

Table A.4: Representative Items from the Derived Checklist

ID	Checklist Item (Security Control)	Mapped SRs
CL-01	Is data ingested from external assets subject to rigorous validation of integrity and freshness (anti-replay)? *(e.g., Implementing HMAC-SHA256 or CMAC for integrity, and verifying Timestamps or Monotonic Counters to prevent replay)*	SR-1, SR-3, SR-6, SR-7, SR-61, SR-62, ...
CL-02	Is input data accepted as trusted only when originating from registered and authenticated external assets? *(e.g., Enforcing mutual authentication via X.509 Certificates (mTLS) or validating Digital Signatures (Ed25519/ECDSA))*	SR-38, SR-43, SR-56, SR-61, SR-81, ...
CL-03	Is secure channel communication (e.g., end-to-end authentication and encryption) enforced between external assets?	SR-41, SR-53, SR-59, SR-66, SR-71, ...
CL-04	Are all events related to external assets logged at a granularity sufficient to ensure non-repudiation and enable post-incident forensic analysis?	SR-5, SR-9, SR-13, SR-17, SR-21, SR-26, ...
	...	
CL-19	Are defense functions in place to detect and mitigate spoofing or adversarial patterns at the sensor level (LiDAR, Radar, Camera, Ultrasonic, GNSS)?	SR-1, SR-6, SR-10, SR-14, SR-18, SR-27, ...
	...	
CL-24	Are Command Codes (or other AI inputs) generated by MLLMs verified for authenticity and checked for Prompt Injection and Context Contamination?	SR-62, SR-80, SR-86, ...
CL-25	Does the system possess mechanisms to detect and mitigate physical and digital adversarial examples (stickers, light sources, lasers, phantom objects, etc.)?	SR-18, SR-55, SR-62, SR-86, SR-126, ...
CL-26	Does the system possess mechanisms to detect and mitigate Transfer Attacks?	SR-44, SR-90, SR-159, SR-200
CL-27	Does the system possess mechanisms to detect and mitigate Poisoned Dataset injection?	SR-50, SR-55, SR-57, SR-62, SR-63, SR-88, SR-89
CL-28	Are inputs to the MLLM monitored to detect, verify, and block Prompt Injection, Context Contamination, and Adversarial Inputs?	SR-55, SR-62, SR-80, SR-86, SR-88, SR-194
CL-29	Does the MLLM accept input (voice, text) only from legitimate, authenticated users and processes?	SR-27, SR-79, SR-81, SR-198, SR-222
	...	
CL-32	Are sensitive data elements (user voice, text, intent, command codes, key slots) protected via encryption and strict access control mechanisms?	SR-24, SR-29, SR-38, SR-53, SR-71, SR-75, ...
CL-33	Does the system maintain operational stability without computational or memory exhaustion under conditions of excessive token input, long-context ingestion, or alert flooding?	SR-52, SR-58, SR-65, SR-76, SR-84, SR-91, ...
CL-34	Does the system transition to a defined "Safe Mode" (fail-safe state) in the event of computational or memory resource exhaustion?	SR-2, SR-70, SR-76, SR-95, SR-100, ...
CL-35	Is access to log repositories and collectors restricted exclusively to authenticated and authorized subjects? *(e.g., Implementing RBAC (Role-Based Access Control) policies and ensuring OS-level isolation via SELinux or AppArmor)*	SR-25, SR-94, SR-98, SR-103, ...
CL-36	Are attempts at unauthorized access, modification, or deletion of log repositories blocked, with such attempts being explicitly audited (logged)?	SR-93, SR-94, SR-98, SR-99, SR-103, SR-104, ...
CL-37	Is the system hardened against insecure deserialization and arbitrary code execution vulnerabilities? *(e.g., Avoiding unsafe serialization formats like Pickle, and enforcing strict schema validation for JSON/Protobuf inputs)*	SR-31, SR-33, SR-36, SR-37, SR-40, SR-45, ...
CL-38	Is the logging infrastructure resilient against manipulation or service interruption caused by the injection of malicious event records (e.g., Log Injection or Log Smashing)?	SR-80, SR-95, SR-100, SR-104, SR-105, SR-109, ...

Table B.1: Mapping to UN R 155 Annex 5 Threats

UN R155 Annex 5 Threat Category (Part A)	Mapped Checklists (CL)
1. Back-end servers used as a means to attack a vehicle or extract data	CL-01, CL-02, CL-03, CL-05
2. Services from back-end server being disrupted, affecting the operation of a vehicle	CL-08, CL-15, CL-22, CL-33, CL-34
3. Vehicle related data held on back-end servers being lost or compromised ("data breach")	CL-03, CL-06, CL-07, CL-14, CL-32, CL-35, CL-36
4. Spoofing of messages or data received by the vehicle	CL-01, CL-02, CL-05, CL-10, ...
5. Communication channels used to conduct unauthorized manipulation, deletion or other amendments to vehicle held code/data	CL-01, CL-02, CL-03, CL-09, CL-10, CL-20, ...
6. Communication channels permit untrusted/unreliable messages to be accepted or are vulnerable to session hijacking/replay attacks	CL-01, CL-02, CL-03, CL-05, CL-07, CL-10, CL-14, ...
7. Information can be readily disclosed. For example, through eavesdropping on communications or through allowing unauthorized access to sensitive files or folders	CL-03, CL-04, CL-07, CL-13, CL-14, CL-16, ...
8. Denial of service attacks via communication channels to disrupt vehicle functions	CL-08, CL-15, CL-22, CL-33, CL-34
9. An unprivileged user is able to gain privileged access to vehicle systems	CL-12, CL-17, CL-23
10. Viruses embedded in communication media are able to infect vehicle system	CL-09, CL-11, CL-31, CL-37
11. Messages received by the vehicle (for example X2V or diagnostic messages), or transmitted within it, contain malicious content	CL-01, CL-10, CL-19, CL-21, CL-24, CL-26, CL-28, CL-30
12. Misuse or compromise of update procedures	CL-03, CL-06, CL-07, CL-09, ...
13. It is possible to deny legitimate updates	CL-08, CL-09, CL-15
15. Legitimate actors are able to take actions that would unwittingly facilitate a cyber-attack	CL-05, CL-08, CL-17, CL-23
16. Manipulation of the connectivity of vehicle functions enables a cyber-attack, this can include telematics; systems that permit remote operations; and systems using short range wireless communications	CL-02, CL-03, CL-05, CL-08, CL-09
17. Hosted 3rd party software, e.g. entertainment applications, used as a means to attack vehicle systems	CL-08, CL-09, CL-12, CL-18, CL-23, CL-28, CL-31, CL-37
18. Devices connected to external interfaces e.g. USB ports, OBD port, used as a means to attack vehicle systems	CL-05, CL-08, CL-09, CL-12, CL-17, CL-23
19. Extraction of vehicle data/code	CL-06, CL-07, CL-14, CL-16, ...
20. Manipulation of vehicle data/code	CL-04, CL-13, CL-17, ...
21. Erasure of data/code	CL-04, CL-13, CL-35, CL-36, CL-38
22. Introduction of malware	CL-09, CL-11, CL-18, CL-31, CL-37
23. Introduction of new software or overwrite existing software	CL-11, CL-18, CL-20, CL-31, CL-37
24. Disruption of systems or operations	CL-08, CL-09, ...
25. Manipulation of vehicle parameters	CL-20, CL-21
26. Cryptographic technologies can be compromised or are insufficiently applied	CL-06, CL-07, CL-14, CL-16, ...
27. Parts or supplies could be compromised to permit vehicles to be attacked	CL-18, CL-27, CL-31
28. Software or hardware development permits vulnerabilities	CL-06, CL-11, CL-12, CL-15, ...
29. Network design introduces vulnerabilities	CL-08, CL-15, CL-22, CL-23, CL-33
31. Unintended transfer of data can occur	CL-05, CL-17, CL-25
32. Physical manipulation of systems can enable an attack	CL-25, CL-26, CL-27, CL-28