VicOne

Cybersecurity in Software-Defined Vehicles: Exposing the Gaps, Mapping the Risks

This research paper provides an organized summary of key discussions on software-defined vehicles (SDVs), which have garnered increasing attention in recent years from the perspective of cybersecurity. It offers practical insights and proposals to address specific risks associated with APIs, container technologies, OTA updates, and AI models, as well as possible countermeasures. To support the possibilities that SDV advancement brings, this publication articulates the crucial perspectives of safety and reliability in a clear and accessible manner.

It is a valuable resource for research and development, and reference for future discussions and initiatives. I hope this research paper will be widely read and contribute to the realization of a safer and more secure mobility society.

— Dr. Ryo Kurachi, Designated Professor, Ph.D. (Information Science), Center for Embedded Computing Systems, Graduate School of Informatics, Nagoya University

The Strategic Shift to Software-Defined Vehicles

Software-defined vehicles (SDVs) have become a focal point in the evolution of the automotive industry. But what exactly defines an SDV? According to Maitê Alves Bezerra, chief SDV analyst at Wards Intelligence, the term refers to the industry's digital transformation, reimagining vehicles as dynamic systems that can be continuously upgraded, rather than static, feature-locked machines.

While Tesla and several Chinese automakers have led the charge, others are quickly following suit. At CES 2025, Honda unveiled plans to launch a next-generation electric vehicle (EV), codenamed "0," by 2026. This vehicle will integrate AI and container technology to enable advanced software deployment. Meanwhile, BMW is accelerating its Neue Klasse platform, which will debut a new lineup of EVs beginning in 2025. These developments mark a broader shift toward a smarter, more efficient era of SDVs. Central to this shift is not only the emphasis on the core concept of software-driven design but also the emergence of entirely new architectural frameworks and technological ecosystems.

Compared to traditional vehicles, SDVs are built on fundamentally different design architectures, characterized by the following elements:

- **Electrical/Electronic (EE) architecture:** Domain-centralized or zonal E/E architecture
- **Software architecture:** Service-oriented architecture (SOA) and software isolation techniques such as virtualization, containerization, and memory partitioning
- **Development practices:** Modern software pipelines leveraging CI/CD (continuous integration and continuous deployment)

Thanks to the maturation of virtual electronic control unit (ECU) technology and the efforts of industry leaders like AWS, NXP, Qualcomm, and SOAFEE, SDVs are rapidly becoming a reality. In the SDV ecosystem, cloud services use virtual ECUs and digital twin technology to simulate vehicle functions, enabling software development and testing. Optimized software is then deployed to physical vehicles via over-the-air (OTA) updates, enhancing functionality and performance. This process is completed through close collaboration between automotive manufacturers (OEMs) and suppliers, jointly driving the continuous iteration of the SDV development cycle.

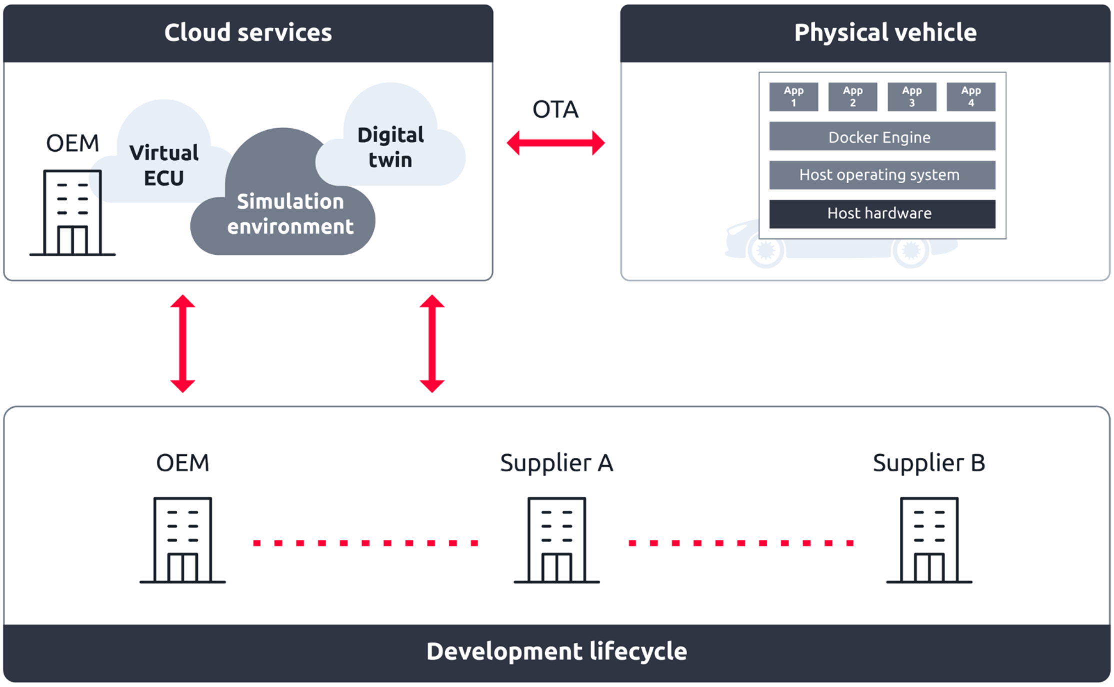


Figure 1. SDV development cycle and flow

By breaking down traditional development barriers, this approach significantly shortens the time needed for development, integration, troubleshooting, bug fixing, and validation. Ultimately, it empowers vehicles with the ability to apply real-time updates. This unlocks new opportunities for OEMs to create more flexible, user-centric mobility experiences.

Cyberthreats to SDVs and Their Business Impact

The highly digitized and connected nature of SDVs opens up new possibilities for innovation, but it also introduces security risks that OEMs cannot afford to ignore. These risks directly impact product safety, user trust, and core business operations:

- **Fatal accidents: compromised vehicle control and safety.** As vehicle functions shift from traditional mechanical power control to electronic signal–based systems, the security of ECUs becomes paramount. A compromised ECU could allow attackers to manipulate essential functions such as braking, steering, and acceleration, potentially leading to severe accidents. Attackers could also exploit software or virtualization vulnerabilities to gain remote control of vehicles or launch denial-of-service (DoS) attacks, disrupting in-vehicle communications and blocking critical updates or navigation data. The consequences extend beyond user safety, potentially triggering reputational damage and regulatory scrutiny.

- **Mass recall crises: exploited software and updates:** OTA updates may be regarded as a double-edged sword: They can prevent the need for mass recalls, but they can also instigate them. While OTA updates offer the convenience of remote maintenance, they also introduce a potential attack surface. For example, vulnerabilities in third-party hardware or software could allow attackers to inject malicious code in OTA updates, affecting the operation of multiple vehicle models. If not properly secured, OTA updates could result in systemwide malfunctions, forcing large-scale recalls that incur substantial costs and erode market trust.

- **Data leak fines: data breaches and privacy violations.** SDVs generate and store a wealth of sensitive data — from user locations and driving behavior to biometrics and personal preferences — in both local storage and the cloud. If attackers gain access to this data — whether through onboard systems or compromised cloud platforms — the fallout can be extensive. For instance, if a cloud platform is breached, attackers could simultaneously access data from multiple vehicles or even extort money from fleet users. This could not only undermine user trust but also result in regulatory fines and legal action.

- **Market losses: operational and financial disruptions.** SDVs are tightly integrated with third-party services and infrastructure, raising the risk of operational interference. For example, attackers could exploit vulnerabilities in APIs or OTA mechanisms to launch fleetwide attacks, remotely disabling vehicles or tampering with logistics routes. The resulting operational downtime and financial losses could ripple across the supply chain, damaging business partnerships and weakening market competitiveness.

According to VicOne's 2025 automotive cybersecurity report, the top three cybersecurity threats to SDVs from 2014 to 2024 were:

- **Supply chain threats,** potentially leading to exploited software and updates

- **Third-party integration threats,** potentially leading to data breaches and privacy violations

- **Vehicle hijacking threats,** potentially leading to compromised vehicle control and safety

Threat type	Count
Supply chain threats	1,564
Third-party integration threats	308
Vehicle hijacking threats	295
Fleet-specific threats	44
Cloud and back-end threats	30
Network threats	27
Virtualization threats	3

Table 1. Top SDV cybersecurity threats based on the number of published vulnerabilities associated with them from 2014 to 2024

From a system architecture perspective, vulnerabilities can be mapped across four primary SDV domains:

- **Onboard domain:** in-vehicle software and hardware components
- **Offboard domain:** build-time tools and processes
- **Cloud domain:** cloud-based services for connected vehicles
- **Development domain:** development environments and toolchains

The **onboard domain** accounts for most published vulnerabilities from the past decade (83%), driven by the increasing complexity of in-vehicle systems like ECUs, communication networks, and operating system platforms. This highlights the urgent need to implement security measures for critical vehicle functions, such as OTA updates and internal communication protocols.

Meanwhile, the **cloud domain** has seen a significant rise in vulnerabilities in recent years, reflecting the growing dependence on cloud-based services for real-time data processing, feature deployment, and EV charging networks. This trend underscores the importance of securing cloud infrastructure and reliable vehicle-to-cloud (V2C) communication.

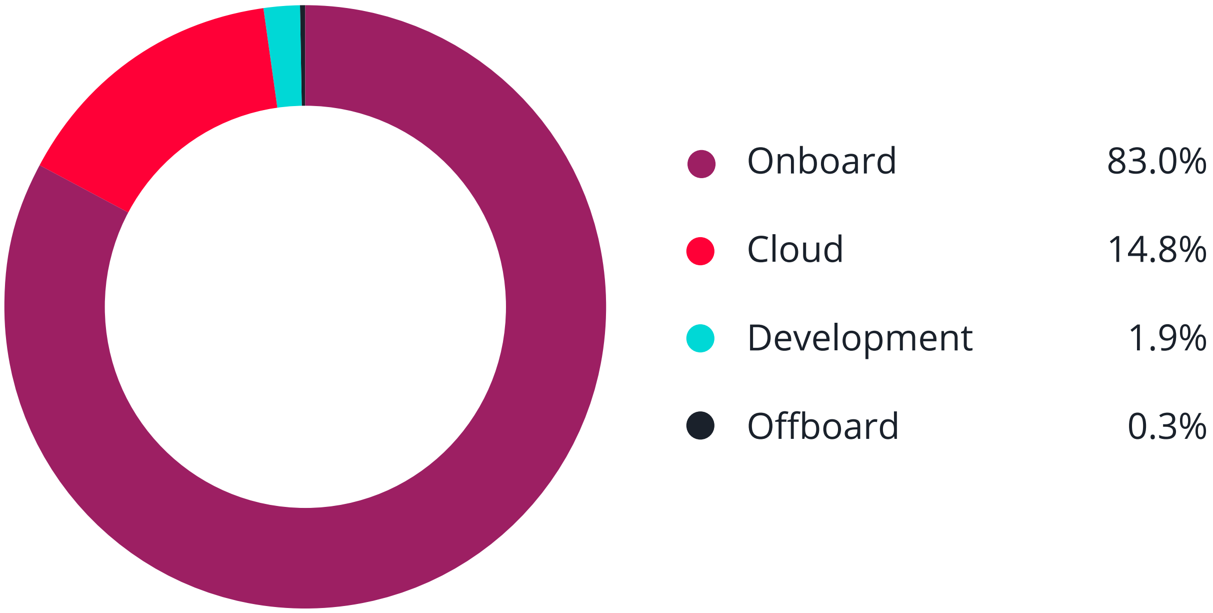


Figure 2. Distribution of automotive vulnerabilities published from 2014 to 2024 by SDV domain

Four Risk Scenarios: Pinpointing SDV Vulnerabilities

To better illustrate how SDV cyberthreats play out in practice, we use the following scenarios to explore common yet high-impact service functions. Each case highlights a specific area — ranging from APIs and virtualization to OTA updates and in-vehicle AI — where vulnerabilities can emerge, propagate, and potentially compromise vehicle safety, user trust, and business operations.

API Risks in Automated Valet Parking

Automated valet parking (AVP) enables a driver to exit their vehicle at a designated drop-off point, leaving the vehicle to park itself autonomously. It offers convenience and efficiency, especially in dense urban environments with limited parking spaces. OEMs like BMW and Mercedes-Benz have already integrated AVP into select premium models, positioning it as a key differentiator in the evolving mobility landscape.

However, developing AVP functionality requires SDVs to orchestrate complex software interactions. Applications from multiple suppliers are integrated into a single ECU, communicating with vehicle systems through APIs. While APIs enable seamless functionality, weak API governance could allow attackers to exploit them. Unauthorized access to critical APIs, for example, could lead to loss of vehicle control or even safety incidents, posing reputational and legal risks to OEMs.

The integration of applications from multiple suppliers introduces challenges such as:

- **Diverse supplier security standards:** Varying levels of cybersecurity maturity across vendors might result in inconsistent protections and unpatched vulnerabilities.
- **High-risk applications:** Applications associated with critical functions like braking and steering require strict isolation and access control to prevent misuse.
- **Expanded attack surfaces:** Applications that serve as entry points for external communication can be exploited by attackers to gain unauthorized access to other APIs.
- **Continuous functional evolution:** OTA updates that add, remove, or modify applications and APIs require adaptive access control to avoid introducing new vulnerabilities.

Attackers could take advantage of these challenges to launch attacks such as:

- **Service API attacks:** Attackers could exploit service API vulnerabilities to tamper with sensor data (e.g., LIDAR, radar), causing AVP systems to misjudge parking space locations and instigate collisions, or to steal map data, compromising user privacy.

- **Hardware abstraction layer (HAL) API attacks:** Attackers could use HAL API access to gain hardware control and remotely manipulate steering or braking to cause AVP systems to malfunction or even injure pedestrians — threatening user safety and potentially triggering a recall crisis.
- **Middleware/Operating system API attacks:** Attackers could exploit middleware/operating system API vulnerabilities to implant backdoors or conduct cross-layer attacks across multiple operating systems. These could disable AVP functionality, disrupt fleet operations, or expose supplier data, damaging business partnerships.
- **Chained attacks:** Escalating from service APIs to HAL and middleware/operating system APIs, attackers could achieve systemic control to cause simultaneous steering and braking failures during parking, which in turn could lead to serious safety and reputational consequences.

To protect AVP functionality and ensure user safety, OEMs must immediately strengthen API access controls and enhance security isolation across layers while establishing unified security standards with suppliers to build a dynamic cybersecurity defense mechanism.

Virtualization Risks in Isolating Safety and Non-Safety Systems

Security Recommendations

- **Enhance isolation with zero trust.** Implement a zero trust approach: Assume that no entity — inside or outside the system — is inherently trustworthy. Verify every access request and isolate applications and APIs at each layer (e.g., service, HAL, middleware/OS) to limit the blast radius of potential breaches.
- **Strengthen API access control with identity and access management (IAM).** Adopt an IAM framework to ensure that only authenticated and authorized entities can access in-vehicle APIs. Enforce consistent security policies across all components to prevent unauthorized access and maintain the integrity of the vehicle’s network.

Virtualization Risks in isolating Safety and Non-Safety Systems

Container technology has become a cornerstone of SDV architecture, providing essential support for isolating vehicle functions and enabling dynamic software updates. By separating safety-critical functions (e.g., brake control) from non-safety functions (e.g., entertainment apps), containers help ensure compliance with ISO/SAE 26262 functional safety standards. At the same time, they facilitate OTA updates, enabling OEMs to quickly deploy new features or vulnerability patches, which ultimately enhance the user experience and extend vehicle longevity.

The lightweight nature of containers also improves ECU resource utilization and supports flexible business models that create new revenue streams for OEMs, such as multi-supplier integration and feature-based subscription services. However, as with any digital innovation, the adoption of container technology introduces new cybersecurity challenges that could impact both vehicle safety and business operations.

> *"In SDVs, container technology plays a key role in ensuring the independence between safety-related and non-safety-related applications that have different ASIL levels. Platforms like Docker Engine are required to support functional safety by enabling security features such as signed image verification and least-privilege execution, as well as facilitating the exchange of safety-related signals between applications. Furthermore, tools like Docker Desktop also need to consider functional safety compliance."*
>
> ***— Yuho Aoki, Functional Safety Manager, SGS Japan Inc.***

Key security challenges include:

- **Unsecure container images:** Container images from multiple suppliers might contain unpatched vulnerabilities or malicious code.
- **Incomplete isolation:** Misconfigured containers could allow unauthorized access between applications, jeopardizing safety-critical functions.
- **Poorly configured hosts:** Improperly or unsecurely set-up host systems could expose vulnerabilities that could be exploited by attackers to compromise the containers running on them or the hosts themselves.
- **OTA update risks:** Unverified container OTA updates could be intercepted and replaced with malicious versions by attackers.
- **Dynamic access control complexity:** Feature-based subscription services and OTA updates require frequent container activations or removals, increasing the difficulty of maintaining consistent, secure permissions.

Attackers could take advantage of these challenges to launch attacks such as:

- **Container image attacks:** Attackers could embed malicious code into container images during the supply chain phase or forge OTA updates to push malicious containers to vehicles. They could then exfiltrate sensitive data (e.g., navigation history) or install persistent backdoors, undermining user privacy and brand integrity.

- **Container escape attacks:** Attackers could exploit shared system resources or runtime flaws to achieve container escape — accessing safety-critical containers (e.g., brake control) from non-safety containers (e.g., entertainment apps) — to manipulate vital vehicle functions. The consequences could trigger safety incidents or recalls.
- **Container vulnerability exploits:** Attackers could exploit security weaknesses within the container runtime or configuration to gain unauthorized access, escalate privileges, or compromise broader system integrity.
- **OTA update interception attacks:** Attackers could intercept the OTA update process to replace legitimate containers with malicious versions that disable subscription features (e.g., advanced driver assistance), disrupt services, or steal proprietary data — impacting user satisfaction and revenue.

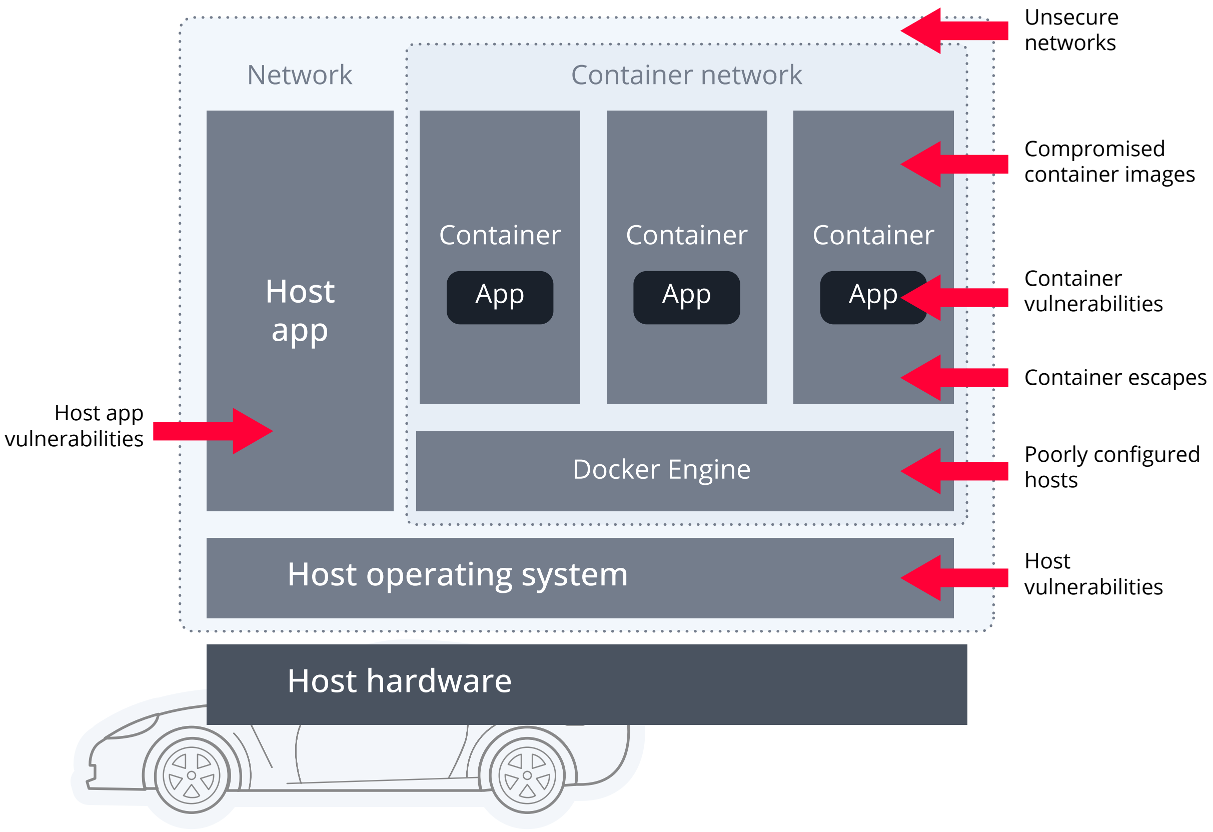


Figure 3. Container attack vectors

Given the potential consequences of compromised containers, many developers have recognized the importance of properly addressing the security issues of containers themselves.

Security Recommendations

- **Implement adaptive container escape detection.** Continuously monitor for abnormal behavior to detect container escape attempts. Use an adaptive solution (such as VicOne xCarbon) that learns from attack patterns, extracts signatures, and applies expert rules to prevent similar threats effectively.
- **Lock down overprivileged configurations.** Secure container configuration files to prevent overprivileged settings or unsecured networks. Implement "config lockdown" features to block unauthorized access and reduce risks of privilege escalation and data theft.
- **Validate OTA updates with cryptographic integrity checks.** Enforce strict verification for OTA-deployed containers using cryptographic signatures to ensure that updates are legitimate and prevent attackers from injecting malicious versions.
- **Leverage threat intelligence for proactive defense.** Track threat intelligence sources, including dark web activity, and correlate findings with supplier vulnerabilities to support timely and targeted security responses.

In some designs, additional layers of protection, such as using virtual machines (VMs), are implemented to further enhance isolation between different domains.

Vulnerability Propagation via Frequent OTA Updates

One of the core advantages of SDVs lies in their open, standardized software architecture, which accelerates innovation and development cycles, enabling faster and more cost-effective product iterations. Through software reuse and widespread adoption of open-source components, OEMs can reduce development costs and enhance market competitiveness.

Frequent OTA updates are a key enabler of this agility. Compared to traditional vehicles, which historically receive software updates only four or five times per year, SDVs can now receive updates on a monthly or even weekly basis. Tesla vehicles, for instance, have made regular OTA updates a hallmark of their user experience. However, this increased flexibility comes with trade-offs. The same open and connected infrastructure that supports OTA updates also creates new opportunities for attackers, who generally find exploiting open-source software easier than targeting proprietary software.

According to a white paper published by the digital.auto open community, titled "Continuous Homologation for Software-Defined Vehicles," OTA capabilities help OEMs shift toward "continuous value streams." Yet these same mechanisms can become attack vectors. If attackers breach cloud services and inject flawed or malicious software, OTA updates could deploy it before vulnerabilities are detected, threatening vehicle safety and business stability.

Key challenges include:

- **Exposure to cloud attacks:** Cloud-based SDV development environments, as opposed to traditional closed-network setups, introduce new, externally accessible attack surfaces.
- **Inconsistent integration of open standards:** While standardized frameworks (e.g., AUTOSAR) improve interoperability, uneven implementation across OEMs and suppliers can result in exploitable security gaps.
- **Risks of software reuse:** While software reuse speeds up development, unvetted shared libraries or components can introduce and spread vulnerabilities across multiple models or platforms.

Attackers could take advantage of these challenges to launch attacks such as:

- **Open-source software attacks:** Attackers could exploit known or zero-day vulnerabilities in open-source software to inject malicious code into the SDV software stack and steal sensitive user data, execute remote commands, or destabilize vehicle systems.

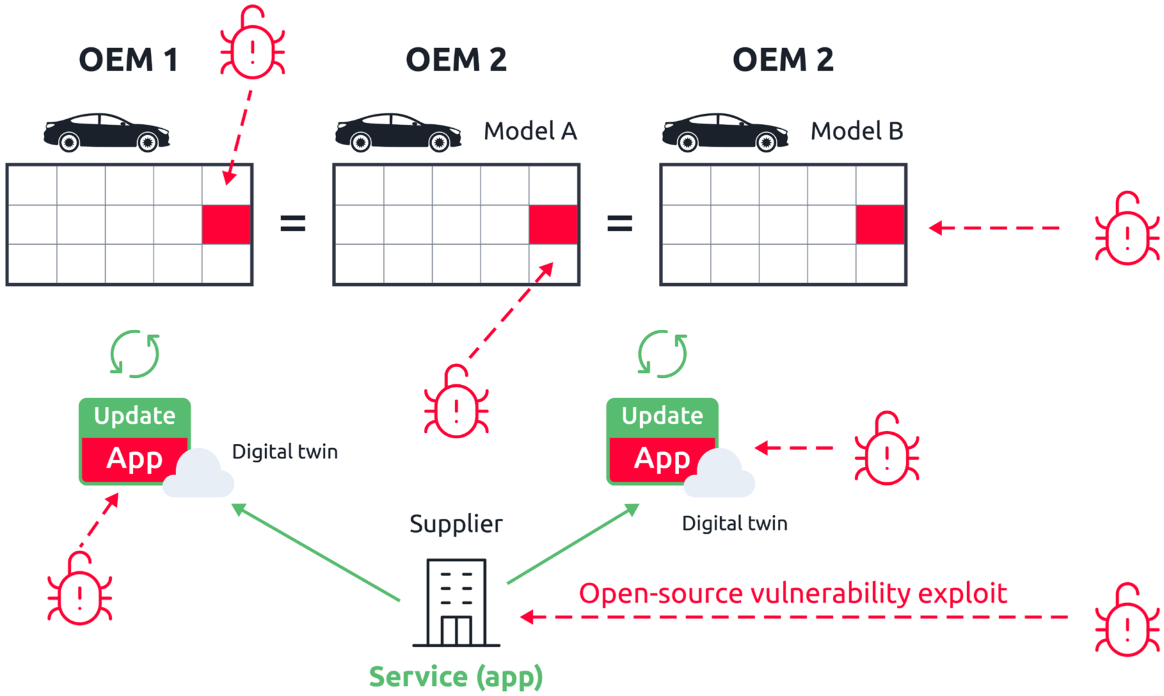


Figure 4. Unseen vulnerabilities in standardized and open-source software could lead to supply chain attacks.

Security Recommendations

- **Validate supply chain software.** Implement strict code and firmware scanning processes for third-party and open-source software to catch vulnerabilities early and prevent supply chain attacks.
- **Deploy zero-day threat detection.** Use an advanced vulnerability management system (such as VicOne xZETA) that detects zero-day and undisclosed vulnerabilities to spot threats missed by known-vulnerability scanning.
- **Continuously monitor emerging threats.** Keep track of novel cyberattack incidents across the industry to assess potential exposure to the same threats or vulnerabilities.
- **Secure cloud infrastructure.** Regularly audit cloud environments for misconfigurations and enforce strict encryption, access control, and least-privilege policies using automated compliance tools to reduce exploitable gaps.

AI Risks in Intelligent Vehicle Functions

AI technologies are becoming essential to SDVs, enabling personalized, adaptive features that enhance both user experience and brand value. By analyzing driver states — such as fatigue levels, behavior patterns, and preferences — AI systems can adjust settings like seating, music, and navigation in real time. These systems also support context-aware features such as “home-to-office” modes and multi-driver profiles.

To deliver these capabilities, AI models require large volumes of data, including sensitive in-vehicle inputs (e.g., voice commands, biometric data) and personal information (e.g., location history). While this data helps enhance the user experience and strengthen brand competitiveness, it also makes AI systems a potential target for attackers.

> *“With the advancement of AI, intelligent vehicle functions such as seat adjustment and music selection are rapidly improving in terms of convenience and personalized optimization. However, cybersecurity threats targeting AI—such as prompt injection and data poisoning attacks—are also emerging. To counter these risks, highly advanced security measures and flexible strategic frameworks, including the use of NIST AI risk management frameworks, will be key in the future.”*
>
> ***— Makoto Kayashima, Ph.D. (Engineering), Chief Engineer, Technology Development Functional Div., Astemo, Ltd.***

Key challenges include:

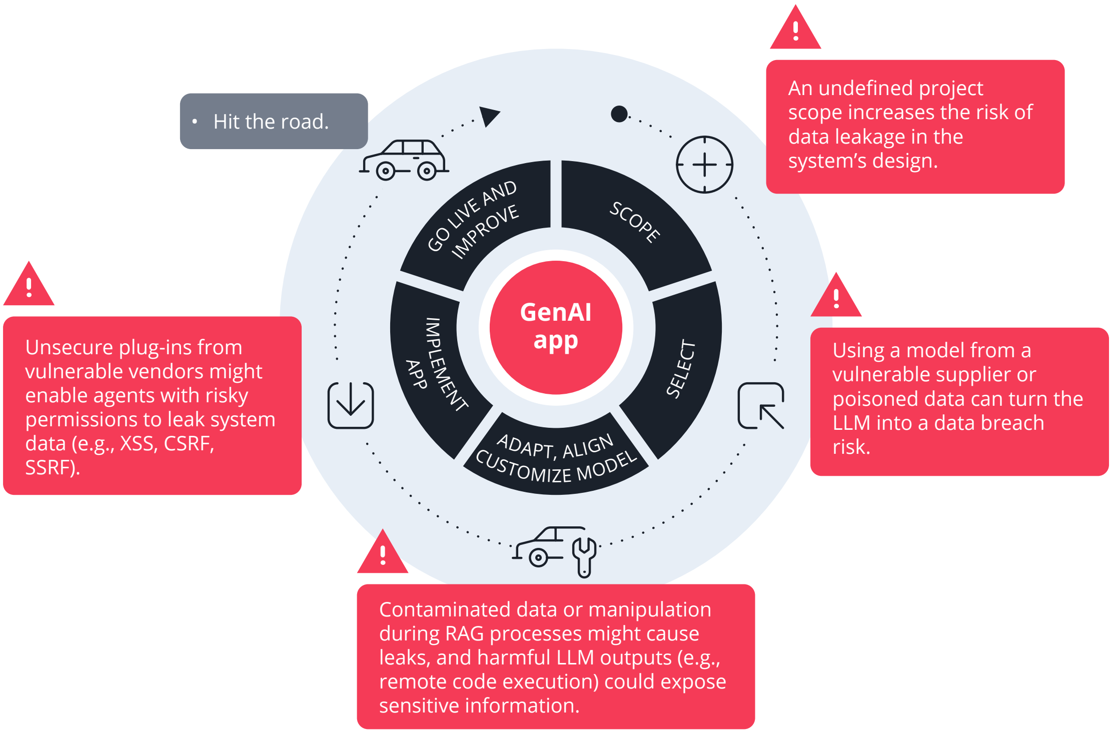
- **Data overexposure:** If sensitive data is not properly encrypted during collection, transmission, or storage, it could be easily accessed by attackers.
- **Prompt injection vulnerabilities:** AI models, especially small language models (SLMs) deployed in vehicles, could be tricked by attackers into misinterpreting malicious prompts to leak data or execute malicious commands. The limited computational resources available to SLMs could introduce gaps in their ability to understand complex contexts, which attackers could easily exploit.
- **Unsecure model training:** Improperly handled training data, such as datasets containing unscrubbed private information, could be reverse-engineered or misused by attackers.
- **Multi-driver confusion:** AI systems operating in shared-vehicle scenarios could inadvertently expose personal data between users, complicating privacy enforcement.

Attackers could take advantage of these challenges to launch attacks such as:

- **Prompt injection attacks:** Attackers could use simple text prompts to dupe AI systems into leaking sensitive data, increasing the risk of identity theft, user extortion, or brand damage.
- **Data poisoning attacks:** Attackers could corrupt training data to bias AI behavior, potentially leading to unsafe decisions, system malfunctions, or reputational and legal consequences.
- **AI model tampering:** Security weaknesses like the Trojan Source vulnerability (CVE-2021-42574) could allow attackers to manipulate code using visually misleading characters.

Attackers could weaponize invisible characters to manipulate code and data in ways that make attacks difficult to detect through normal review processes. For example, in AI chatbots, researchers have repeatedly forced models to reveal or ignore safety guidelines by inserting carefully placed Unicode patterns. As more OEMs adopt AI-assisted infotainment or semiautonomous features, invisible injection techniques could compromise the user experience or even system integrity.

- **Development lifecycle security risks:** Vulnerabilities might arise at every phase of the generative AI (GenAI) application development lifecycle, from scoping and selection to



An undefined project scope increases the risk of data leakage in the system's design.

Using a model from a vulnerable supplier or poisoned data can turn the LLM into a data breach risk.

Contaminated data or manipulation during RAG processes might cause leaks, and harmful LLM outputs (e.g., remote code execution) could expose sensitive information.

Unsecure plug-ins from vulnerable vendors might enable agents with risky permissions to leak system data (e.g., XSS, CSRF, SSRF).

Figure 5. Security risks in the GenAI application development lifecycle

"NIST-AI-600-1: Artificial Intelligence Risk Management Framework – Generative Artificial Intelligence Profile" helps organizations identify and manage GenAI risks. In the automotive industry, these risks could result in safety incidents, user attrition, regulatory fines, and significant financial losses. To ensure the safe application of AI technologies, OEMs must strengthen data encryption, secure AI model training processes, and establish defenses against prompt injection and data poisoning attacks.

Security Recommendations

- **Implement secure AI design principles:** Limit AI access to essential data only — using encryption, access control, and real-time monitoring (e.g., VicOne's Smart Cockpit Protection solution) — to avoid sensitive data leaks and address AI security risks.
- **Validate suppliers:** Enforce transparency with vendors via software and machine learning bills of materials (SBOMs and MLBOMs), paired with strong vulnerability checks to secure third-party components.
- **Safeguard data:** Encrypt sensitive training data, track usage with audit trails, and comply with privacy laws to prevent unauthorized access and build user trust.
- **Strengthen governance:** Set up dedicated AI teams and a clear governance framework to manage risks, ensure compliance, and foster a data-driven culture.
- **Leverage the NIST AI Risk Management Framework:** Refer to the NIST AI Risk Management Framework to systematically assess and manage AI-related risks.

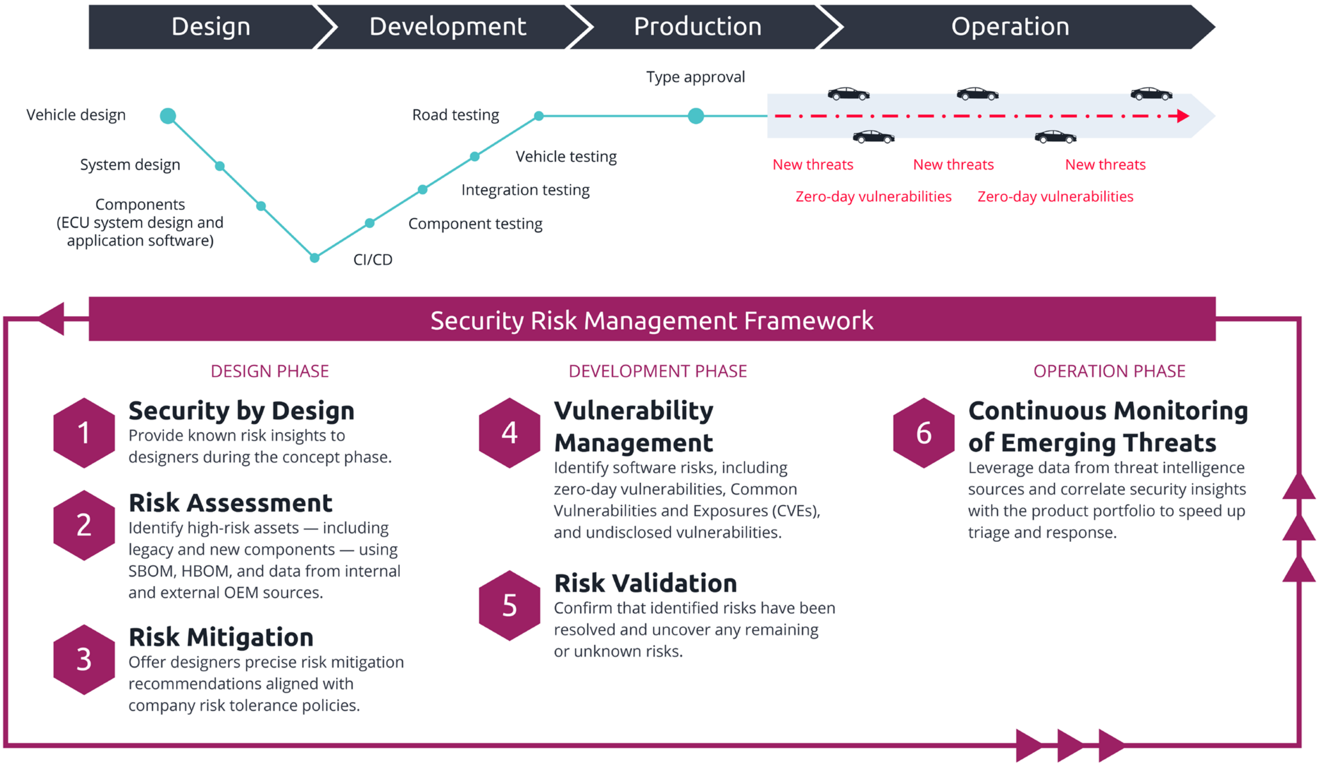
Boosting SDV Security Today: A Strategic Roadmap

Given that the lifespan of a vehicle typically spans 12 to 15 years, cybersecurity is no longer a short-term consideration — it is a long-term commitment tied to brand reputation and operational resilience. VicOne's automotive threat intelligence reveals a 600% increase in vehicle-related cyberattacks over the past four years, with attack methods growing more sophisticated and scalable. As software becomes the backbone of the modern vehicle, the question emerges: **How can OEMs ensure the security of their SDVs — and safeguard their reputation — for over a decade on the road?**

To address the escalating cybersecurity challenges and protect the SDV ecosystem, user safety, and brand integrity, we recommend the following forward-thinking strategies across organizational, cultural, personnel, and technical dimensions.

Organizational Level: Establishing a Robust Security Risk Management Framework for the Entire Vehicle Lifecycle

Form a cross-functional cybersecurity governance board within your organization, uniting R&D, supply chain, product security incident response, vehicle security operations center (VSOC), and operations teams to develop an SDV security strategy aligned with UN R155 and ISO/SAE 21434 requirements. This ensures end-to-end security across the vehicle lifecycle — from design to operation. In addition, enforce unified cybersecurity contracts with suppliers and cloud service providers, clearly defining responsibilities to mitigate supply chain risks and safeguard your production pipeline.



1 **Security by Design**
Provide known risk insights to designers during the concept phase.

2 **Risk Assessment**
Identify high-risk assets — including legacy and new components — using SBOM, HBOM, and data from internal and external OEM sources.

3 **Risk Mitigation**
Offer designers precise risk mitigation recommendations aligned with company risk tolerance policies.

4 **Vulnerability Management**
Identify software risks, including zero-day vulnerabilities, Common Vulnerabilities and Exposures (CVEs), and undisclosed vulnerabilities.

5 **Risk Validation**
Confirm that identified risks have been resolved and uncover any remaining or unknown risks.

6 **Continuous Monitoring of Emerging Threats**
Leverage data from threat intelligence sources and correlate security insights with the product portfolio to speed up triage and response.

Figure 6. Establishing a robust security risk management framework for the entire vehicle lifecycle

Cultural Level: Fostering a Security-First Development Culture

Embed a “security by design” philosophy into your corporate culture, integrating cybersecurity checkpoints at every stage of SDV development, from initial design to OTA updates. This cultural shift will help mitigate vulnerabilities stemming from the transition from hardware-centric to cloud-based mindsets, protecting your vehicles from emerging threats.

Personnel Level: Bridging IT and Automotive Expertise to Counter Dual Threats

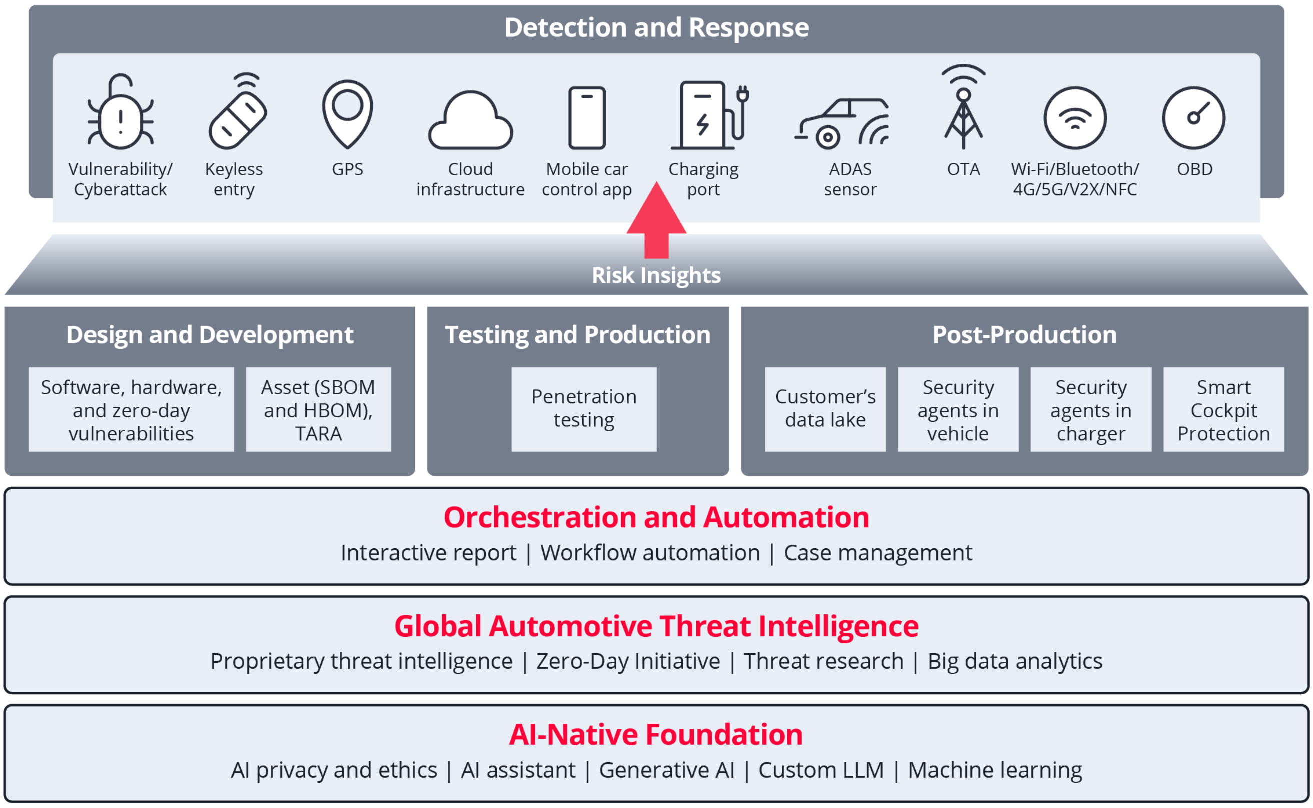
SDVs blur the lines between IT and automotive domains, making cybersecurity attacks a two-way threat — where IT vulnerabilities can compromise vehicle safety and vehicle exploits can expose IT systems. Equip your engineering and IT teams with specialized training in cloud and AI cybersecurity. Educate employees on ways to mitigate AI risks, such as choosing reliable AI tools, keeping confidential data away from non–enterprise level AI, carefully verifying results from AI systems, and avoiding overreliance on AI. Partner with external cybersecurity experts or hire talent with expertise in zero trust architectures and supply chain security to bolster your workforce. By cultivating teams with cutting-edge, cross-disciplinary skills, your organization can proactively address SDV-specific cybersecurity challenges, ensuring resilience and maintaining a competitive edge in this evolving landscape.

Technical Level: Building a Comprehensive and Closed-Loop Risk Management Solution to Fully Address SDV Risks

Protect your SDVs with a holistic defense strategy powered by advanced technologies. As vehicles become increasingly dynamic, with more frequent software updates and feature releases, each vehicle's risk profile becomes unique and ever-changing. To address this, you need a comprehensive and tightly integrated risk management solution that adapts to the evolving SDV landscape while enhancing risk visibility and bridging the gap between the VSOC team, the product security incident response team (PSIRT), and the product engineering team. When new threats or vulnerabilities emerge, the solution should instantly enable you to assess their impact and severity, swiftly determine necessary actions, and assign the right personnel. Additionally, it should empower your design teams to access production data for design improvements, feeding into a continuous learning model within the vehicle's digital twin for further testing.

This solution should also ensure rapid and reliable validation of software functions' cybersecurity, enabling OTA updates to vehicles on the road. It should have a modular design, built on open standards, that facilitates secure and efficient integration of diverse data sources, allowing automotive industry users to tailor deployment models to their cybersecurity roadmaps. It should adopt a zero trust approach to enforce dynamic verification for API, container, and cloud access, preventing unauthorized access and container escape vulnerabilities. Furthermore, it should leverage automotive threat intelligence to preempt supply chain risks — safeguarding vehicles throughout their 12-to-15-year lifespan, ensuring user safety, and reinforcing your market leadership.

Effective Risk Management Driven by AI



Detection and Response

Vulnerability/ Cyberattack | Keyless entry | GPS | Cloud infrastructure | Mobile car control app | Charging port | ADAS sensor | OTA | Wi-Fi/Bluetooth/ 4G/5G/V2X/NFC | OBD

Risk Insights

Design and Development: Software, hardware, and zero-day vulnerabilities | Asset (SBOM and HBOM), TARA

Testing and Production: Penetration testing

Post-Production: Customer’s data lake | Security agents in vehicle | Security agents in charger | Smart Cockpit Protection

Orchestration and Automation

Interactive report | Workflow automation | Case management

Global Automotive Threat Intelligence

Proprietary threat intelligence | Zero-Day Initiative | Threat research | Big data analytics

AI-Native Foundation

AI privacy and ethics | AI assistant | Generative AI | Custom LLM | Machine learning

Figure 7. Building a comprehensive risk management solution to fully address SDV risks

As SDV adoption accelerates, cybersecurity risks escalate, threatening production, fleet operations, and customer trust. For OEMs, ignoring these risks could lead to costly recalls, regulatory fines, and brand damage. Prioritizing cybersecurity is no longer optional; it is essential to future-proofing vehicles, protecting users, and sustaining leadership in a software-defined automotive era.

Threats are inevitable, but risks are manageable. The time to act is now.

Cybersecurity in Software-Defined
Vehicles: Exposing the Gaps,
Mapping the Risks