

Automotive cybersecurity: CAN and lifecycle testing

To address emerging cybersecurity threats across the automotive industry requires considering the entire vehicle lifecycle, from development and production to operation. The Hydravision security test automation solution from Dissecto, a company from Dspace (both Germany), addresses these challenges by enabling reproducible, automated, and comprehensive security assessments. The tool enables a test engineer to adapt the open and customizable test-case source code to project-specific needs.

(Source: Adobe Stock)

As our daily lives become more digital, data-driven, and connected, cybersecurity has become a critical concern, particularly in the mobility sector, where advancements in autonomous driving, connectivity, and software-defined vehicles are reshaping the landscape. To address emerging threats, regulations such as UN R155 and standards like ISO/SAE 21434 have been introduced to harmonize cybersecurity practices across the automotive industry. These frameworks emphasize a risk-based approach throughout the entire vehicle lifecycle, from development and production to operation, and promote cybersecurity awareness across all organizational levels.

Cybersecurity testing in development

During the development of a vehicle, it is important to perform sufficient testing to verify the specification of the implemented cybersecurity controls and to validate the corresponding cybersecurity goals. This should include functional and conformance testing to ensure correct behavior according to specifications, and offensive testing, such as fuzzing and penetration testing, to minimize unidentified vulnerabilities and weaknesses that could be exploited for malicious purposes. In particular, all external communication interfaces as well as the entire in-vehicle network must undergo a thorough threat analysis and risk assessment due to the large attack surface.

CAN communication requires security

While Ethernet-based traffic is continuously increasing within modern in-vehicle networks, communication via the controller area network (CAN) remains an essential part. Although CAN was not originally designed with security in mind, there are various approaches today to introduce mechanisms, which ensure security properties such as integrity, authenticity, and confidentiality. For CAN CC (classic) and CAN FD (flexible data rate), these mechanisms

have to be introduced via higher-level protocols above the OSI (open systems interconnection) data-link layer, e.g., Autosar Secure Onboard Communication (SecOC), Secure Diagnostics, and proprietary OEM-specific authentication and encryption mechanisms (OEM: original equipment manufacturer). The newer CAN generation CAN XL (extended data-field length) provides an intrinsic security mechanism CANsec on the data-link layer, which is currently under development by member companies from CAN in Automation (CiA).

Attack categories

To understand the threat model of CAN-based communication, we distinguish two broad categories of attacks: bit-level manipulation attacks and data-driven attacks.

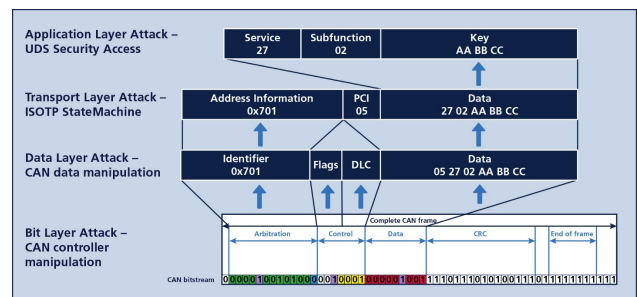


Figure 1: Overview of different attack surfaces across the CAN communication stack, illustrating how bit-level manipulation targets physical-layer controller behavior while data-driven attacks exploit fields and state machines at the data, transport, and application layers. (Source: Dissecto)

Bit-level manipulation attacks target the internal communication mechanisms of CAN controllers, such as retransmission handling, error counters, or collision-resolution behavior (see paper "[CAN bus security](#)"). Executing such attacks typically requires very high privileges on the host ▶

Road vehicles

side of an ECU (electronic control unit), because the attacker needs to control the GPIOs (general purpose input-outputs) connected to the CAN transceiver/controller and actively drive the bus behavior while bypassing the CAN controller's standard framing and error handling. In practice, this implies detailed knowledge of the ECU hardware (exact schematic, MCU variant, and board revision) and a level of execution control that goes beyond "normal" diagnostic access. From a scalability perspective, these attacks are usually confined to the local CAN segment: They can disrupt or manipulate communication on the attacked bus, but they are not naturally propagated by gateways into other CAN networks. A further challenge is detectability: Since these attacks operate below the semantic level of frames and signals, simple IDS (intrusion detection system) approaches that rely on payload plausibility or message timing may fail to identify bit-level manipulation, even though the resulting effects can be severe.

Data-driven attacks operate on the CAN CC/CAN FD frame content itself and on higher-layer protocols carried over CAN, such as ISO-TP (transport protocol), XCP (universal measurement and calibration protocol), or UDS (unified diagnostic services). At the lowest layer, common threats include message spoofing, replay, and bus-flooding DoS (denial-of-service). In secure deployments, these attacks often aim to undermine authentication and integrity mechanisms (e.g., by exploiting misconfiguration, key management weaknesses, or replay windows), with SecOC primarily threatened by loss of authenticity and integrity if counters, freshness values, or MAC verification can be bypassed or desynchronized. Moving up the stack, transport protocols such as ISO-TP implement internal state machines to segment and reassemble payloads across multiple CAN frames; These parsers and reassembly buffers introduce an additional attack surface for denial-of-service conditions (e.g., state exhaustion and reassembly blocking) and implementation vulnerabilities such as memory corruption in constrained ECUs. Finally, at the application layer, where diagnostic and measurement protocols such as UDS and XCP are located, attackers face a broad and attractive functionality set, ranging from session control and memory access to routine control and flashing. This richness often translates directly into a wider vulnerability space (see paper "[Automated threat evaluation of automotive diagnostic protocols](#)"): Insecure service exposure, weak access control, inconsistent state handling, and implementation bugs can all enable impactful attacks. In general, the higher the functionality and the broader the service surface, the greater the opportunity for abuse, especially if security mechanisms are added late or deployed inconsistently across ECUs.

Challenges in modern cybersecurity testing

Verifying security mechanisms and validating cybersecurity goals within a mature, process-integrated cybersecurity test strategy remains a significant challenge in the automotive domain. Key obstacles include:

- ◆ Delayed and infrequent testing: Cybersecurity testing is often conducted late in the development cycle, increasing the risk of discovering critical issues too late;

- ◆ Limited automation and tool support: Many current testing approaches rely heavily on manual methods and custom scripts, leading to high effort and low reusability;
- ◆ Inadequate test environments: Setting up suitable environments for today's complex ECUs is time-consuming, especially when specific operational modes are required for testing;
- ◆ Lack of expertise and guidance: The absence of clear guidelines and sufficient know-how on what and how to test hinders the establishment of effective cybersecurity testing strategies.

Requirements for security test automation solutions

Addressing these challenges is essential to ensure robust and scalable cybersecurity validation throughout the vehicle development lifecycle. To address these challenges, the industry is increasingly adopting a shift-left approach by integrating cybersecurity testing earlier into the development process, while leveraging existing test infrastructures. Additionally, the degree of automation of cybersecurity testing is being enhanced to support continuous testing (CT) and increasing process maturity.

In this context, state-of-the-art security test automation solutions (such as Hydravision from Dissecto) must ensure reproducibility, robust configuration management, and full customizability, by providing open and customizable test-case source code that a cybersecurity test engineer or penetration tester can adapt to project-specific needs. In vehicle networks, overall protection is only as strong as the weakest ECU on a bus, so a scalable testing strategy must aim to minimize the attack surface of each ECU and systematically validate all exposed interfaces. This naturally leads to a comprehensive unit-testing approach to security, complemented by automated exploratory testing that continuously assesses each relevant attack surface. Especially for higher-layer protocols and ECUs with multiple interfaces, comprehensive assessments can become very time-consuming due to the complexity and footprint of the exposed services. For example, if a gateway ECU provides multiple ISO-TP endpoints, each endpoint constitutes a separate attack surface that must be tested against known protocol vulnerabilities and supplemented by targeted protocol-level fuzzing.

Cybersecurity as a continuous lifecycle obligation

In contrast to functional safety, cybersecurity does not remain "valid" indefinitely: Security controls age as threat landscapes evolve. A function that was considered secure yesterday may no longer meet regulatory or industry expectations tomorrow, and newly disclosed vulnerabilities can compromise the security of an entire vehicle platform. As a result, cybersecurity testing cannot stop at SOP (start of production) or the end of development. Instead, comprehensive security testing must be performed throughout the development lifecycle from the early phases, when defects are cheapest to fix, through integration ►

and validation, and into in-field operation with continuous monitoring and risk analysis to detect when previously secure features become vulnerable.

Depending on the lifecycle phase, the test execution environment can differ significantly: from desktop-based setups during development, to HIL (hardware-in-the-loop) systems in integration, and finally to mobile or vehicle-side test environments for full-vehicle assessments under real-world conditions, e.g., cellular connectivity.

Conclusion

As vehicles become increasingly connected and software-driven, cybersecurity must be treated as an ongoing responsibility rather than a one-time milestone. Because CAN will continue to play a central role in in-vehicle communication, it is essential to understand its attack

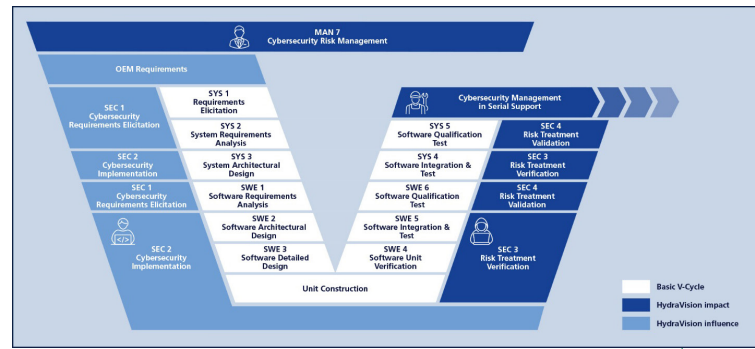


Figure 2: Cybersecurity as a continuous lifecycle obligation, illustrating how security activities extend across the V-model: From early requirements elicitation through software development, verification, validation, and integration to ongoing risk management in serial support. The highlighted areas indicate where HydraVision enables automated, reproducible testing and continuous cybersecurity management to maintain long-term resilience as threats evolve. (Source: Dissecto)

Filter builder and OBD tools

The editor tools from CSS Electronics (Denmark) enable the user to create filters and OBD configuration files for the company's CANedge data loggers tenfold faster than before, states the provider. By default, the data logger records all raw CAN frames at their original frequencies, which can result in a ton of data. Configuration of CAN-ID filters/prescalers to optimize the file size can be useful, but is complex and time-consuming to set up. The Filter builder tool should help to solve these tasks. It enables to sort CAN-IDs by size contribution, to view summary statistics such as MiB/min, frames/second, etc. Additionally, it allows to browse CAN-IDs via DBC (data base CAN) info like message names, signal names, etc. For J1939/Isobus/NMEA data, it is possible to group CAN-IDs by their 18-bit PGNs (parameter group numbers) and to set PGN filters. The batch filtering function is used to select multiple entries and add them with shared prescalers. The search box lets one search entries based on channel, CAN-ID, name, signal names, and more. Using the "unmatched" function allows selection of all CAN-IDs not matched by DBCs to e.g., add them as rejection filters. Added filters can be seen in a summary window. Further, the user can enable compression to reduce file sizes by 50 % to 70 %. If data should be recorded only under certain circumstances, the start/stop logging based on CAN signal thresholds is possible.

Many of the company's end users deploy the CANedge to log OBD data from cars, trucks, and buses for telematics, diagnostics, or development purposes. Previously, configuring the data logger with custom OBD PID (parameter ID) requests was time-consuming and complex, in particular, if many PIDs have had to be logged. With the OBD tool it is now possible to add PIDs from a list thus creating the configuration file in seconds, informs the provider. The tool also enables to auto-add OBD filters (if one only wishes to log OBD responses) and a control signal (to toggle transmission and thus avoid battery drainage). Finally, the tool lets the user to optionally test which PIDs are supported and to limit the PIDs to only those supported by the user's car. of

vectors and consistently validate security mechanisms. By shifting testing earlier, increasing automation, and reusing scalable test assets across ECUs and vehicle programs, organizations can detect vulnerabilities sooner and maintain long-term robustness. Testing platforms already available on the market, for example, the Dissecto's HydraVision, address these challenges by enabling reproducible, automated, and comprehensive security assessments. The lifecycle approach is vital for ensuring trust, safety, and resilience in modern mobility systems. ◀

Authors



Dr. Nils Weiß
Dissecto
nils.weiss@dissecto.com
www.dissec.to

Dr. Matthias Pukrop
Dspace
mpukrop@dspace.de
www.dspace.com