講演番号 161

Penetration Tests for Guiding Development

- Shift left Approach for Penetration Tests during Development -

Dr. Claude-Pascal Stöber-Schmidt [1)] Florian Look [1)] Patrick Loster [1)] Dr. Dennis Kengo Oka[2)] Takuya Nigoro [2)]

1) IAV GmbH Ingenieurgesellschaft Auto und Verkehr, Carnotstraße 1, 10587 Berlin, Germany
2) IAV Japan Co., Ltd., Uchikanda Chuo Bldg. 3F, 1-18-13 Uchikanda, Chiyodaku, Tokyo, 101-0047, Japan

KEY WORDS: **Software and its underlying technologies, cyber attack methods,** cybersecurity (E3),

This paper analyzes how penetration testing can be integrated earlier and more continuously into the vehicle development lifecycle to improve automotive cybersecurity. Traditionally, penetration testing occurs late in development, when systems are nearly finalized and design changes are costly. This delay increases risk because vulnerabilities rooted in architecture, hardware, or software integration are often discovered only when remediation becomes expensive and disruptive.

The paper contrasts the idealized, standards-based development process with real industrial practice, which is shaped by legacy components, incomplete early documentation, late-maturing features, and frequent architectural changes. As a result, activities required by ISO/SAE 21434 such as early TARA, architectural reviews, and systematic verification often occur iteratively or are postponed, raising the likelihood that weaknesses appear during late testing.

To address this, the paper proposes a shift-left approach: embedding penetration testing throughout all development stages. In the concept phase, security concepts and TARA-derived requirements are reviewed. With A-sample prototypes, testers examine hardware protections, debug interfaces, boot modes, and memory exposure. As hardware matures (B- and C-samples), tests better reflect realistic attacker capabilities.

The paper highlights hardware-focused penetration testing, including schematic and PCB reviews, identification of documentation gaps, and early validation of secure boot, fuse settings, memory protections, and fault-injection resistance while design changes are still feasible.

On the software side, the paper examines privilege separation, controlled debug paths, memory protection, driver robustness, and multicore isolation. It argues that secure coding practices, security-focused code reviews, SAST, fuzzing, and CI-integrated checks should complement penetration testing to enable faster, cheaper vulnerability detection.

At the system level, the paper emphasizes integrated testing of ECUs and vehicles under realistic load, timing, and fault conditions. System-level tests validate secure boot chains, update mechanisms, domain separation, and communication security across automotive networks and backends.

Automation and AI further enhance the process by generating test inputs, improving fuzzing, analyzing logs, and supporting continuous testing in CI pipelines expanding coverage and accelerating feedback loops.

While shift-left penetration testing increases coordination demands and complicates planning, the paper concludes that its advantages earlier vulnerability discovery, fewer late redesigns, and stronger confidence in system resilience far outweigh the challenges. Continuous, evidence-driven validation ultimately leads to more secure and robust vehicle systems throughout their lifecycle.