Education Framework of SOME/IP Security Enhancement through Game-Based Learning and Testing Tool Development[*]

JoHee Park, Sung Bum Park, and Dong Hoon Lee[†]

Korea University, Seoul, Republic of Korea
{zoypark, park785, donghlee}@korea.ac.kr

Abstract

As the automotive industry shifts toward software-centric development, cyber security threats are increasingly diverse. In response, there is growing interest in training experts equipped with practical problem-solving skills for security issues. This study evaluates the effectiveness of Capture The Flag (CTF)-based learning, a pedagogical method specialized for the security field, applied to the automotive cyber security domain with a focus on the SOME/IP protocol. Through surveys, performance analysis, and interviews with practitioners and students, the findings reveal that the CTF approach enhances participants' understanding of automotive protocols and vulnerabilities through hands-on experience, significantly boosting learning engagement and motivation. Furthermore, providing appropriate tools effectively lowers entry barriers, a common challenge in CTF-based learning. This research proposes a practical education method tailored to the demands of automotive cyber security, with promising implications for strengthening vehicle safety and reliability.

Keywords: Automotive Security, SOME/IP, Capture the Flag (CTF), Security Education, Tool Development

1 Introduction

Advances in information and communication technologies are upgrading traditional industries, and the automotive sector is likewise evolving toward software-centric "smart cars". In this process, vehicles combine autonomous driving technologies and diverse software with control units and wireless connectivity; at the same time, the enlarged attack surface increases vulnerability to cyberattacks [4]. Automotive hacking can result not only in privacy breaches and financial loss but also in direct risks to driver safety, underscoring the importance of cybersecurity [4, 2].

Against this backdrop, multiple laws and regulations for automotive cybersecurity have been introduced. Notably, the United Nations Economic Commission for Europe (UNECE) adopted UN Regulation No. 155 (R155) in 2020, which mandates cybersecurity management across the entire vehicle lifecycle—from design to operation—and defines essential requirements to protect vehicles and drivers from cyberattacks [23].

Despite regulatory progress, cyber threats continue to evolve, driving demand for security professionals who can respond with agility [22, 20, 12]. However, current education remains largely theory-centric and struggles to prepare learners for real-world threat environments. In other security domains, hands-on approaches such as Capture the Flag (CTF) have been

[*] Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. 72, December 16-18, 2025, Sapporo , Japan.

[†] Corresponding author

adopted and are regarded as effective for cultivating practical problem-solving skills [8]. Yet there is a shortage of empirical studies that apply and evaluate CTF-based training in the automotive context.

To address this gap, we focus on services using the SOME/IP protocol and evaluate the effectiveness of a game-based (CTF) educational methodology. We developed a set of hands-on challenges and a supporting tool, and conducted a study with 63 participants—practitioners and students—who solved the challenges and completed surveys. Using participant backgrounds, task performance, and survey responses, we present statistical analyses to assess the approach with respect to three research questions.

The remainder of this paper is organized as follows: Section 2 reviews background on SOME/IP and CTF. Section 3 summarizes related work. Section 4 states the research questions and experimental environment. Section 5 details the procedure, challenge design, tools, and survey instrument. Section 6 reports the analysis results. Section 7 discusses implications and concludes.

2 Background

2.1 SOME/IP

SOME/IP is a middleware solution based on a Service-Oriented Architecture (SOA) that was developed to provide high efficiency and scalability over in-vehicle Ethernet. First introduced around 2011 and incorporated into the AUTOSAR standard in 2014 [9, 3], the protocol enables high-throughput data communication over Ethernet/IP and supports application-to-application messaging without direct dependence on specific electronic control units (ECUs). Owing to these properties, SOME/IP is well suited to complex systems compared to the Controller Area Network (CAN) and plays a key role in modern vehicle networks.

The adoption of SOME/IP has advanced in-vehicle networking in multiple dimensions throughput, complexity management, flexibility, and standardization. These improvements enable richer vehicle functions and new services, contributing to convenience and safety for drivers and passengers. As MORE components adopt SOME/IP, it becomes increasingly important to recognize and mitigate security weaknesses in the protocol. Because the scope of SOME/IP-based services continues to expand and the associated vulnerabilities evolve rapidly, there is a growing need for practitioner training that builds hands-on response capabilities.

2.2 Capture the Flag (CTF)

Capture the Flag (CTF) is a distinctive and challenging game-based format in cybersecurity that allows participants to practice and be trained on real skills. CTF events provide scenarios that reflect real-world security problems, giving learners opportunities to discover and exploit vulnerabilities using up-to-date techniques and tools. Since the first DEF CON CTF was held in Las Vegas in 1993, CTF has become a widely recognized format for cybersecurity competitions worldwide [11, 19]. Broadly, CTFs are categorized into two styles: *Jeopardy*, which presents standalone, quiz-like challenges across topics, and *Attack–Defense*, in which teams simultaneously protect their own services while exploiting those of opponents. Modern CTFs encompass a wide range of challenge types and targets, and are designed to fit diverse learning platforms and instructional settings.

Table 1: Different types of problems in CTF

Challenge Type	Description
Crypto	Solving problems based on modern or classical cryptographic techniques.
Pwn	Debugging and reverse engineering of binaries to find vulnerabilities (e.g., buffer overflows) and obtain remote shells via exploitation.
Web	Addressing web-security issues such as injections and XSS, often involving packet sniffing and network exploitation.
Forensic	Analyzing corrupted files, memory artifacts, or network captures to recover hidden information.
Reverse Engineering	Using dynamic debugging and static analysis to recover program secrets and understand code behavior.
Steganography	Detecting hidden information embedded in media such as audio, video, or images.
Miscellaneous	Covering varied security topics including data analysis and traffic analysis across domains.

2.3 Objectives and Problem Types of CTF

The primary purpose of Capture the Flag (CTF) is to impart cybersecurity knowledge to participants. These competitions create controlled yet realistic environments that model real-world security problems, in order to assess and develop participants' security-related skills. Prior research indicates that CTFs have helped raise awareness of cybersecurity across multiple educational levels, from secondary to higher education [13].

CTFs encompass a variety of challenge types and are typically organized into core domains that each assess different security competencies. Table 1 summarizes the principal CTF challenge categories.

2.4 CTF-based Learning vs Instruction-based Learning

CTF-based learning differs from traditional approaches in several important ways, and these differences directly affect learning outcomes and learner motivation. Traditional instruction emphasizes *instruction-based learning*, providing clear, stepwise guidelines and procedures that students can follow in a predictable manner. In contrast, CTF-based learning is *problem-solving-based*: learners actively explore, form hypotheses, and discover solutions on their own. In CTFs, learners directly use a variety of tools and techniques to solve tasks and, upon success, obtain an explicit token (a "flag"), which has been shown to increase their sense of accomplishment as well as engagement and motivation [5]. Experiencing realistic threat scenarios further helps cultivate practice-oriented, real-world problem-solving skills [8].

By comparison, traditional methods—while effective for steadily acquiring knowledge through structured, repeatable practice and for evaluating progress against predefined steps—may provide lower motivation than CTFs and offer fewer opportunities for learners to independently explore and resolve authentic security problems.

3 Related Work

3.1 Security of SOME/IP

SOME/IP's primary security issue is the lack of built-in authentication and encryption, which implicitly assumes trust among communicating nodes. Consequently, adversaries can impersonate legitimate nodes and inject malicious messages (spoofing), while the absence of cryptographic protection renders systems vulnerable to man-in-the-middle (MitM) attacks, enabling data exfiltration and tampering. To mitigate these risks, prior work proposes adopting transport-layer cryptography (e.g., TLS/DTLS) together with authentication and authorization mechanisms [14]. Formal and practical analyses further examine protocol correctness and interoperability under adversarial conditions, with recommendations for tool-supported verification [24]. Ticket-based authentication schemes have also been introduced to protect SOME/IP communications against MitM and to ensure secure inter-node exchanges [15]. Complementary detection-oriented approaches employ intrusion detection systems for automotive Ethernet, including machine-learning-driven methods to identify abnormal traffic [21]. Finally, fuzzing has been leveraged to uncover implementation flaws in SOME/IP stacks; recent work advances greybox and state-sensitive fuzzing tailored to automotive protocols, revealing vulnerabilities that traditional testing may miss [16, 25].

3.2 CTF-based Learning in Cybersecurity

A substantial body of work has investigated the effectiveness of CTF-based learning. [8] evaluates how CTF activities influence students' motivation and learning outcomes compared with traditional instruction. In general, CTF challenges have been found to raise motivation and engagement; features such as scoreboards, progress tracking, and hint systems help learners monitor their progress and stay immersed in the tasks. These characteristics increase learners' confidence with security tooling and foster practically applicable skills that translate into real-world problem solving.

At the same time, prior studies consistently note limitations, especially decreased participation stemming from heterogeneous prior knowledge. To mitigate this, researchers propose introducing prerequisite modules and refining difficulty granularity [7]. When experience levels vary widely, selecting appropriate challenge difficulty becomes nontrivial: tasks that are too hard frustrate beginners, while overly easy tasks disengage experts. Accordingly, tailored challenges and calibrated difficulty ramps are recommended to accommodate diverse skill levels [1, 6]. Because CTFs do not always prescribe a clear learning path, some learners may fall behind; strengthening prerequisite knowledge and providing foundational security instruction help ensure that participants have sufficient background before attempting CTF tasks. Moreover, since many challenges assume baseline security knowledge, beginners can face entry barriers—especially if they are unfamiliar with the relevant tools—which may hinder progress.

Despite many advantages, CTF-based education still has several gaps. First, while there is evidence that CTFs improve practical skills, there is little research on their impact on understanding specialized protocols in automotive security, such as SOME/IP. Second, few studies analyze differences in learning experiences across diverse cohorts—particularly students versus practitioners. Third, although appropriate tool support can lower the entry barrier, systematic analyses of the *need* for and the *effectiveness* of such tools in CTF programs remain limited.

To address these gaps, this paper evaluates a CTF-based education program focused on the SOME/IP protocol in the automotive security domain. We analyze improvements in protocol understanding, differences between students and practitioners, and participants' assessments of both the necessity and helpfulness of the provided tools, with the goal of informing the design of more effective cybersecurity education.

4 Research Questions and Experimental Environment

This section states three research questions designed to evaluate the overall effectiveness of the CTF-based education program and to inform tailored instruction for learners with diverse backgrounds.

4.1 RQ1: Effect of the CTF Program on SOME/IP Understanding

The goal of **RQ1** is to analyze the extent to which participation in the CTF program improves participants' understanding of the SOME/IP protocol. We examine post-program understanding, perceived linkage between theory and practice, and perceived improvement in inspection/assessment skills.

4.2 RQ2: Differences in Learning Across Backgrounds

RQ2 focuses on how learning experiences differ among participants with varied backgrounds (e.g., students versus practitioners). We analyze these differences to derive implications for designing customized training programs that accommodate heterogeneous cohorts.

4.3 RQ3: Impact of Tool Support on Accessibility

The aim of RQ3 is to evaluate the extent to which the tools provided within the CTF program are *necessary* for solving the challenges and *helpful* for task completion. We analyze perceived accessibility gains and the practical contribution of tool support to problem-solving effectiveness.

4.4 Experimental Environment

4.4.1 Participants

To evaluate the effectiveness of the CTF-based automotive security education program, we conducted an experiment with participants from diverse backgrounds, comprising both industry practitioners and students. We compared their understanding of the SOME/IP protocol and their learning experiences. In total, 63 participants took part in the study and were categorized into two groups:

- Practitioners (n=22): Professionals with at least one year of experience in automotive/security-related work. Their prior exposure to SOME/IP varied.
- Students (n=41): Undergraduate and graduate students without prior industry experience. While they possessed basic security knowledge, they were beginners in automotive security.

The two groups reflected heterogeneous demographics (e.g., gender, age, majors), which supports the generalizability of the results. All participants provided informed consent after a clear explanation of the study's purpose and procedures. The study was conducted after consent was obtained, and all collected data were anonymized and used solely for research purposes.

4.4.2 Educational Platform and Setting

We built a CTFd-based web platform and conducted the experiment in both online and offline settings. Participants accessed challenges and submitted answers from their own computers. The back-end environment ran on AWS x64/ARM64 Linux instances with Dockerized Ubuntu 20.04. We deployed SOME/IP service binaries implemented with `vsomeip` and CommonAPI, enabling hands-on interaction with realistic vulnerabilities that can occur in practice. We also provided instructional materials covering basic SOME/IP theory and examples, along with a tool designed to improve accessibility to the protocol-specific tasks (e.g., lowering setup overhead and facilitating message crafting).

5 Methodology

5.1 Procedure

The study proceeded through the following phases:

1. Design of challenges, tools, and survey: Preparatory work for the study; details are described in Sections 4.2, 4.3, and 4.4.
2. Pre-survey: A questionnaire was administered to collect participants' background information and their prior understanding of the SOME/IP protocol.
3. Training session: A concise briefing on SOME/IP fundamentals and CTF problem-solving workflows was provided to ensure smooth participation in the experiment.
4. CTF challenge solving: Participants solved the challenges using the provided platform and tools; facilitators answered questions in real time during task execution.
5. Post-survey: After the CTF, a follow-up questionnaire measured changes in understanding of SOME/IP, learning experiences, and perceived usefulness of the tools.

5.2 CTF Challenge Development

This section describes the key vulnerabilities targeted in our SOME/IP training and how the challenges were designed to evaluate the effectiveness of a CTF-based approach.

5.2.1 Vulnerability Selection Method

The hands-on tasks were designed so that participants could learn about, and practically address, security weaknesses in the SOME/IP protocol while reflecting realistic in-vehicle security scenarios. In this way, learners develop both conceptual understanding and practical skills. We identified the target vulnerabilities using the following three inputs:

- Practitioner interviews: Interviews with security professionals surfaced vulnerability types that frequently arise in practice (e.g., buffer overflows, stack overflows, memory leaks), which were incorporated into the challenge set.
- Report review: We reviewed reports from automotive-security vendors, including Cybellum's *State of Automotive Security* report, which analyzes 100+ automotive software vulnerabilities and highlights issues such as buffer overflows, outdated components, supply-chain weaknesses, and information leakage [10].

Table 2: Primary vulnerabilities identified

Vulnerability	**CWE**	**Description**
Information Leak	CWE-200	Unintended exposure of sensitive information; may occur via log files, verbose debug outputs, or memory dumps.
Path Traversal	CWE-22	Bypassing file-system checks to traverse directories and access restricted files or data.
Command Injection	CWE-77	Execution of arbitrary system commands through unsanitized inputs, allowing attackers to gain control of services.
Stack Buffer Overflow	CWE-121	Memory corruption from oversized input on the stack, potentially leading to crashes or arbitrary code execution.

- CWE & CVE analysis: We consulted the Common Weakness Enumeration (CWE) and Common Vulnerabilities and Exposures (CVE) catalogs to ensure coverage of prevalent classes [18, 17]. Among the recurring issues identified were sensitive information exposure and stack buffer overflows.

Based on this process, we selected four primary vulnerability categories, summarized in Table 2.

5.2.2 Implementation in `vsomeip` and CommonAPI

To instantiate the selected vulnerabilities as hands-on tasks, we implemented challenges on top of both `vsomeip` and CommonAPI. `vsomeip` is an open-source implementation of the SOME/IP protocol for in-vehicle communication; it is widely used for testing and developing data exchange. Because it follows a standardized design, it offers flexibility to control fine-grained parameters such as message headers and service/method IDs, though the initial setup can be time-consuming.

CommonAPI is a standardized API set that enables communication across diverse middleware stacks and is available in multiple languages (e.g., C++, Python). It orchestrates in-vehicle communication at a higher abstraction level and supports multiple protocols, including SOME/IP. The primary vulnerability categories targeted in our challenges are summarized below.

5.2.3 `vsomeip` Challenges

We implemented four challenges in `vsomeip` corresponding to the selected vulnerability classes:

1. Path Traversal: Insufficient validation of user-supplied file paths in a `vsomeip` service allowed directory traversal and access to sensitive files.

2. Information Leak: Verbose error and debug outputs exposed internal system details and sensitive information.

3. Command Injection: Lack of proper input sanitization enabled execution of arbitrary system commands.

4. Stack Buffer Overflow: Missing length/bounds checks on input led to stack memory corruption.

Table 3: Five modules added to vsomeip_ctrl

Module	Description
V1. Service/Method ID Input	Supports sending specific requests by receiving service and method IDs as parameters.
V2. Message Length Auto-Calculation	Automatically calculates the message length based on the input data to reduce input errors.
V3. Custom Header Value Input	Allows manual input of specific header values to facilitate testing in various situations.
V4. Random Payload Generation	Provides the functionality to generate random payload values to enhance the diversity of tests.
V5. String-Based Payload Generation	Offers the capability to generate string-based payloads, allowing configuration for different scenarios.

5.2.4 CommonAPI Challenges

Within CommonAPI, we implemented two challenges drawn from the selected vulnerability set:

1. Command Injection: Input-handling flaws permitted execution of arbitrary system commands.

2. Stack Buffer Overflow: Insufficient bounds checking on input caused stack memory corruption.

5.3 Tool Development

To lower the setup burden of SOME/IP communication—which typically requires building source code and configuring a complex environment—and to reduce the entry barrier for learners with limited coding background, we developed and provided a dedicated tool. In this study, the tool `vsomeip_ctrl` is tailored specifically for `vsomeip`, implemented in C++ on Linux to offer an environment optimized for the SOME/IP protocol.

Building on the upstream project [9], we added functionality and distributed the tool to participants so that we could also analyze its impact with respect to RQ3 (the accessibility and usefulness of tool support). The `vsomeip_ctrl` utility consists of five helper modules that facilitate access to, and experimentation with, the targeted vulnerabilities, as summarized in Table 3.

5.4 Survey Instrument

After providing the developed challenges to the participants and observing their problem-solving processes, we designed a survey to assess vulnerability awareness, problem-solving ability, and the overall effectiveness of the education program. The instrument was structured with reference to the ARCS motivational model, and eight quantitative items were adapted from the Instructional Materials Motivation Survey (IMMS).

Following completion of the CTF, participants answered a questionnaire consisting of eight 5-point Likert items and three open-ended questions. Table 4 lists the items: Q2–Q9 are Likert-scale questions, and Q10–Q12 are open-ended.

Table 4: Survey Questions: Responses to Quantitative Questions (Q2-Q9) Were Given on a Likert Scale from 1 (Very Low) to 5 (Very High), and Qualitative Questions (Q10-Q13) Were Answered in Descriptive Form

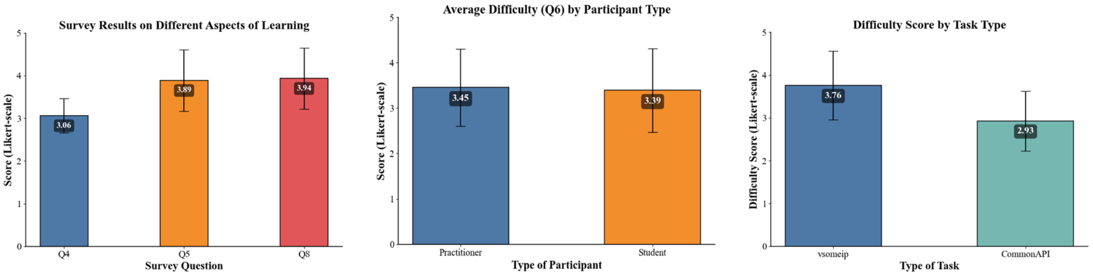
ID	Category	Question
Prior knowledge		
Q1	Prior understanding	What was your prior understanding of the SOME/IP protocol? (Low/Medium/High)
Likert-scale questions		
Q2	Satisfaction	How would you rate your overall satisfaction with this CTF training program?
Q3	Motivation/Engagement	To what extent did the CTF training help enhance your motivation and engagement in learning?
Q4	Skill improvement (inspection)	How much do you feel your skills in SOME/IP inspection have improved through the CTF training?
Q5	Theory–practice linkage	How well do you think the theory and practice were connected?
Q6	Difficulty appropriateness	Was the difficulty level of the CTF challenges appropriate?
Q7	Tool helpfulness	How helpful were the provided learning tools?
Q8	Understanding improvement	How has your understanding of the SOME/IP protocol improved?
Q9	Need for tools	To what extent did you feel the need for tools while performing the CTF?
Qualitative questions		
Q10	Job role	What is your job role?
Q11	Advantages of CTF-based learning	What do you think are the advantages of CTF-based learning? (Multiple choices allowed)
Q12	Issues encountered	If you encountered any issues during the CTF, what were they?
Q13	Desired features (future tool)	What features would you like in a future SOME/IP inspection tool?

5.5 Data Analysis

We collected data via surveys and used the responses to quantitatively evaluate the effectiveness of CTF-based training in the automotive security domain. Because some measures did not satisfy normality assumptions, we employed nonparametric statistics: the Kruskal–Wallis test for comparisons across three or more groups (e.g., prior-knowledge tiers) and the Mann–Whitney U test for two-group comparisons (e.g., students vs. practitioners).

6 Results

This section reports the analysis of survey responses from 63 participants to address the research questions.



(a) RQ1: Post-program improvement (Q4, Q5, Q8).

(b) RQ2: Students vs. practitioners (understanding, Q8).

(c) RQ3: Difficulty with/without tools (`vsomeip` vs. CommonAPI).

Figure 1: Survey highlights across research questions: (a) improvements after the CTF, (b) cohort comparison, and (c) tool impact on perceived difficulty.

6.1 RQ1: Effect of the CTF Program on SOME/IP Understanding

Items most closely tied to RQ1 are Q1, Q4, Q5, and Q8. The means (and standard deviations) are summarized in Fig. **??**. Prior understanding of the SOME/IP protocol (Q1) was very low (mean = 1.23), indicating that most participants began the CTF with limited familiarity. After the program, participants reported positive gains: perceived improvement in SOME/IP inspection skills/knowledge (Q4; mean = 3.06), perceived linkage between theory and practice (Q5; mean = 3.89), and improvement in protocol understanding (Q8; mean = 3.94). These findings suggest that the CTF-based approach can enhance practical problem-solving by connecting hands-on tasks with core concepts.

We further tested whether prior understanding affected post-program understanding using a Kruskal–Wallis test across three prior-knowledge tiers (low/medium/high). The result ($p = 0.9481$) exceeded the significance threshold $\alpha = 0.05$, indicating no statistically significant differences among tiers. In other words, participants' understanding of SOME/IP improved regardless of their initial familiarity.

6.2 RQ2: Differences in Learning Across Backgrounds

For between-group analyses, we adopted the cohort composition reported in Section 4 (22 practitioners, 41 students); We then analyzed how their learning experiences differed during the CTF program. Post-program understanding of the SOME/IP protocol (Q8)—summarized by means and standard deviations in Fig. **??**—was comparable between groups; baseline (pre-program) understanding did not show a meaningful difference.

Overall satisfaction with the CTF was high (grand mean = 4.17). A Mann–Whitney U test comparing practitioners and students yielded $p = 0.0602 > \alpha = 0.05$, indicating no statistically significant difference in satisfaction(Q2); both cohorts rated the program favorably. Likewise, motivation and engagement showed a high grand mean (= 4.03). Another Mann–Whitney U test indicated no group difference ($p = 0.1920 > \alpha = 0.05$), suggesting that both practitioners and students reported similarly strong motivation and engagement.

6.3 RQ3: Impact of Tool Support on Accessibility

Across all participants, the perceived *necessity* of tool support (Q9) showed a high mean of 4.37, indicating that learners regarded tools as important for completing CTF tasks. As illustrated in Fig. **??**, participants who rated the challenges as more difficult tended to focus on tasks for which tools were available (e.g., `vsomeip`), whereas those who perceived lower difficulty also attempted tasks without dedicated tools (CommonAPI). This pattern suggests that, for participants reporting higher perceived difficulty (Q6), the presence of tools substantially increased task accessibility.

The perceived *helpfulness* of the tools had a mean of 3.98. To examine whether prior knowledge moderated these perceptions, we conducted a Kruskal–Wallis test across three prior-understanding tiers (low/medium/high). The result ($p = 0.5097 > \alpha = 0.05$) indicates no statistically significant differences among tiers; participants benefited from tool support regardless of initial familiarity. We also compared practitioners and students using the Mann–Whitney U test, which yielded $p = 0.6037 > \alpha = 0.05$, showing no significant group differences; both cohorts rated the tools as helpful.

7 Implications and Conclusion

7.1 Implications for Education and Practice

Our findings offer several actionable takeaways for automotive cybersecurity training focused on SOME/IP:

- CTF as a protocol-specific pedagogy. Even with low prior familiarity, participants reported meaningful gains in understanding and inspection skills, indicating that CTFs can effectively teach protocol behaviors and failure modes.
- Tool-enabled accessibility. Usability-oriented helpers (e.g., automatic message-length calculation, configurable payload generation) reduced setup burden and supported engagement on higher-difficulty tasks, suggesting that carefully designed tooling is integral to learning in middleware-heavy environments.
- Bridging theory and practice. Participants perceived strong theory–practice linkage, reinforcing the value of combining concise conceptual briefings with hands-on, diagnostically rich challenges.
- Cohort-independent benefits. Comparable outcomes across students and practitioners suggest that a single, well-scaffolded track—with optional difficulty ramps—can serve heterogeneous audiences.

7.2 Limitations

This study has several limitations. First, it centers on a single protocol family (SOME/IP), which may limit generalizability to other in-vehicle technologies (e.g., UDS/DoIP, CAN(-FD), AUTOSAR Adaptive services). Second, most outcomes are *perceived* measures; richer objective metrics (e.g., exploit time-to-solution, error rates, and post-test diagnostics) would strengthen causal claims. Third, the intervention duration was limited, so retention and longitudinal transfer remain open questions. Finally, the toolchain targets `vsomeip` in particular; results may vary with alternative stacks and deployment constraints.

7.3 Future Work

Future directions include (i) expanding coverage to additional automotive protocols and cross-protocol interactions, (ii) incorporating longitudinal assessments to measure retention and skill transfer, (iii) adding adaptive difficulty and prerequisite micro-modules to personalize scaffolding, (iv) integrating automated evaluators and trace-based analytics for objective performance measurement, and (v) generalizing the tooling to multiple stacks and operating conditions.

7.4 Conclusion

We presented and evaluated a CTF-based education framework for SOME/IP security that pairs curated vulnerability-centric challenges with usability-focused tools. Across a mixed cohort, participants reported improved protocol understanding, strong theory–practice linkage, and high motivation, with tool support perceived as both necessary and helpful. These results support CTFs as a practical, scalable approach to protocol-specific upskilling in automotive cybersecurity and motivate broader, longitudinal, and multi-protocol studies to consolidate evidence and maximize impact.

References

[1] S. Austen and S. Addison. Improving the educational efficacy of beginner-friendly cybersecurity competitions. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education (SIGCSE)*, volume 2, pages 1233–1233, May 2023.

[2] AutoCrypt. Trends in vehicle vulnerabilities: A 2023 report. https://autocrypt.io/trends-in-vehicle-vulnerabilities-2023-report/, 2023. Accessed: 2024-10-09.

[3] AUTOSAR. Specification of some/ip protocol. Technical report, November 2015.

[4] Beaumont and Samantha Isabelle. Commonalities in vehicle vulnerabilities, 2024. Accessed: 2024-10-09.

[5] T. J. Burns, S. C. Rios, T. K. Jordan, Q. Gu, and T. Underwood. Analysis and exercises for engaging beginners in online CTF competitions for security education. In *USENIX Workshop on Advances in Security Education (ASE 17)*, 2017.

[6] A. Chothia and N. Novakovic. Pwn the learning curve: Education-first ctf challenges. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education (SIGCSE)*, volume 2, pages 1233–1233, March 2024.

[7] K. Chung and J. Cohen. Learning obstacles in the capture the flag model. In *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*, 2014.

[8] S. V. Cole. Impact of capture the flag (ctf)-style vs. traditional exercises in an introductory computer security class. In *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, volume 1, pages 470–476, July 2022.

[9] COVESA. vsomeip. https://github.com/COVESA/vsomeip, 2024. Accessed: 2024-10-09.

[10] Cybellum. The state of automotive security 2023. https://security.cybellum.com/the-state-of-automotive-security-2023, 2023. Accessed: 2024-10-14.

[11] DEF CON. Def con official website. https://defcon.org/, 2024. Accessed: 2024-10-13.

[12] (ISC)^2. Cybersecurity workforce study 2023. Technical report, 2023. Accessed: 2023-10-01.

[13] S. Karagiannis and E. Magkos. Adapting ctf challenges into virtual cybersecurity learning environments. *Information & Computer Security*, 29(1):105–132, 2020.

[14] J. Kreissl. Securing some/ip for in-vehicle service protection. Master's thesis, University of Stuttgart, Stuttgart, Germany, 2017.

[15] Seulhui Lee, Wonsuk Choi, and Dong Hoon Lee. Protecting some/ip communication via authentication ticket. *Sensors*, 23(14):6293, July 2023.

[16] Y. Li, H. Chen, C. Zhang, S. Xiong, C. Liu, and Y. Wang. Ori: A greybox fuzzer for some/ip protocols in automotive ethernet. In *2020 27th Asia-Pacific Software Engineering Conference (APSEC)*, pages 495–499. IEEE, December 2020.

[17] MITRE Corporation. Common vulnerabilities and exposures (cve). https://cve.mitre.org/, 2024. Accessed: 2024-10-14.

[18] MITRE Corporation. Common weakness enumeration (cwe). https://cwe.mitre.org/, 2024. Accessed: 2024-10-14.

[19] Nautilus Institute. Def con ctf 2025 qualifier. https://ctftime.org/event/2604/, 2025. Finals held August 2025 at DEF CON 33, Las Vegas; accessed 2025-08-17.

[20] NIST. Nice factsheet: Workforce demand. Technical report, June 2023.

[21] N. Quadar, A. Chehri, B. Debaque, I. Ahmed, and G. Jeon. Intrusion detection systems in automotive ethernet networks: Challenges, opportunities and future research trends. *IEEE Internet of Things Magazine*, 7(2):62–68, March 2024.

[22] Justin Rende. Track these 7 trends for proactive cybersecurity in 2024. https://www.isaca.org/resources/news-and-trends/industry-news/2023/track-these-7-trends-for-proactive-cybersecurity-in-2024, 2023. Accessed: 2024-10-09.

[23] UNECE. Un regulation no. 155 – cyber security and cyber security management system. https://unece.org/transport/documents/2021/03/standards/un-regulation-no-155-cyber-security-and-cyber-security, 2021. Accessed: 2024-10-09.

[24] D. Zelle, T. Lauser, D. Kern, and C. Krauß. Analyzing and securing some/ip automotive services with formal and practical methods. In *Proceedings of the 16th International Conference on Availability, Reliability and Security (ARES)*, pages 1–20, August 2021.

[25] F. Zuo, Z. Luo, J. Yu, Z. Liu, and Y. Jiang. Pavfuzz: State-sensitive fuzz testing of protocols in autonomous vehicles. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pages 823–828. IEEE, December 2021.