Enhancing Automotive Intrusion Detection Systems with Capability Hardware Enhanced RISC Instructions-Based Memory Protection

electronics

MDPI

Article

Enhancing Automotive Intrusion Detection Systems with Capability Hardware Enhanced RISC Instructions-Based Memory Protection

Chathuranga Sampath Kalutharage [1,2,*], Saket Mohan [1], Xiaodong Liu [2] and Christos Chrysoulas [3]

[1] Secure Elements, Coventry CV1 2NT, UK; saket.mohan@secureelements.co.uk
[2] School of Computing, Engineering & the Build Environment, Edinburgh Napier University, Edinburgh EH10 5DT, UK; x.liu@napier.ac.uk
[3] School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh EH14 4AS, UK; c.chrysoulas@hw.ac.uk
* Correspondence: sampath.kalutharage@secureelements.co.uk

Abstract: The rapid integration of connected technologies in modern vehicles has introduced significant cybersecurity challenges, particularly in securing critical systems against advanced threats such as IP spoofing and rule manipulation. This study investigates the application of CHERI (Capability Hardware Enhanced RISC Instructions) to enhance the security of Intrusion Detection Systems (IDSs) in automotive networks. By leveraging CHERI's fine-grained memory protection and capability-based access control, the IDS ensures the robust protection of rule configurations against unauthorized access and manipulation. Experimental results demonstrate a 100% detection rate for spoofed IP packets and unauthorized rule modification attempts. The CHERI-enabled IDS framework achieves latency well within the acceptable limits defined by automotive standards for real-time applications, ensuring it remains suitable for safety-critical operations. The implementation on the ARM Morello board highlights CHERI's practical applicability and low-latency performance in real-world automotive scenarios. This research underscores the potential of hardware-enforced memory safety in mitigating complex cyber threats and provides a scalable solution for securing increasingly connected and autonomous vehicles. Future work will focus on optimizing CHERI for resource-constrained environments and expanding its applications to broader automotive security use cases.

Keywords: automotive cybersecurity; IP spoofing; memory protection

Academic Editors: Tao Huang, Shihao Yan, Guanglin Zhang, Tsz Hon Yuen, YoHan Park, Jusak Jusak and Changhoon Lee

Received: 17 December 2024
Revised: 20 January 2025
Accepted: 21 January 2025
Published: 24 January 2025

Citation: Kalutharage, C.S.; Mohan, S.; Liu, X.; Chrysoulas, C. Enhancing Automotive Intrusion Detection Systems with Capability Hardware Enhanced RISC Instructions-Based Memory Protection. *Electronics* **2025**, *14*, 474. https://doi.org/10.3390/electronics14030474

1. Introduction

The automotive industry is undergoing transformative innovation, driven by rapid technological advancements and the advent of autonomous vehicles. Modern vehicles are no longer purely mechanical systems; they have evolved into complex mechatronic platforms, integrating numerous Electronic Control Units (ECUs) interconnected through sophisticated communication networks [1–3]. With the integration of advanced software systems and electronic hardware components, vehicles now support a wide array of functionalities. Notably, the rise of autonomous vehicles has driven a substantial increase in the deployment of ECUs. These ECUs manage critical vehicle functions, such as engine control, braking, and driver assistance, as well as secondary systems like lighting, infotainment, and connectivity services. Operating within interconnected networks, they utilize various communication protocols to ensure seamless and efficient performance [4]. However, this increased interconnectivity introduces a broad attack surface for cyber

threats, posing significant risks to vehicle security, operational reliability, and passenger safety [5].

One of the critical components in ensuring vehicle cybersecurity is the Intrusion Detection System (IDS) [6]. IDSs play a pivotal role in detecting malicious activities at an early stage, thereby minimizing the potential impact of cyberattacks. They offer both known and unknown threats, and even zero-day vulnerabilities are derived based on pattern analysis within network traffic to deviate from usual behavior [7]. IDSs monitor network traffic to identify and reduce threats, including unauthorized resource access, data manipulation, and violation of protocol rules. However, traditional IDS solutions have failed to ensure the safety and security of automotive systems against complex attack vectors. Key concerns include, but are not limited to, IP spoofing—the attacker forges the source IP address to appear as a legitimate device–and manipulation of IDS rules, enabling them to bypass detection. Furthermore, Hamada et al. [8] present the development of an IDS at the central gateway of automotive networks to detect spoofing messages. This underscores the prevalence of message spoofing attacks and highlights the urgent need for robust detection mechanisms within automotive systems. These studies put together prove that traditional IDSs, while indispensable for automotive cybersecurity, often cannot handle such advanced and evolving attack vectors as IP spoofing and rule manipulation. This underscores the need for innovative approaches to strengthen IDS capabilities in modern automotive environments.

Other notable attack vectors include denial-of-service (DoS) attacks, which flood networks with excessive traffic to overwhelm IDS processing capabilities. These attacks are particularly concerning in resource-constrained environments, such as automotive networks, where even a single malicious source can disrupt operations. While distributed denial-of-service (DDoS) attacks are more complex and involve traffic from multiple sources, DoS attacks are easier to execute and can cause significant disruption with limited resources. This underscores the critical need for IDS solutions designed to withstand such targeted attacks. Additionally, message replay attacks disrupt operations by resending previously captured valid messages. Payload tampering exploits vulnerabilities through data packet injection or modification, while man-in-the-middle (MITM) attacks enable malicious actors to intercept, alter, or inject messages between ECUs. Timing attacks exploit delays in message processing, and protocol-specific exploits target communication standards like CAN, CAN-FD, or SOME/IP. These vulnerabilities highlight the limitations of traditional IDS solutions and the urgent need for robust, hardware-based security measures to safeguard IDS functionality.

As vehicles integrate more advanced connectivity and autonomous features, the attack surface for cyberattacks targeting Electronic Control Units (ECUs) expands significantly. Research highlights that modern vehicles contain an increasing number of ECUs, with high-end models surpassing 100 units, compared to around 25 in typical vehicles from a decade ago [9]. This proliferation of ECUs, coupled with their connectivity to external networks, correlates with a rising number of reported vulnerabilities. For example, studies have demonstrated that cyberattacks can remotely exploit these systems to disable critical functions, such as braking and steering, or even drain a vehicle's battery [10]. The number of connected vehicles globally is projected to reach over 400 million by 2025, further emphasizing the urgent need for robust cybersecurity measures [11]. Attackers can exploit network vulnerabilities to inject malicious packets or spoof legitimate IP addresses, bypassing conventional IDS mechanisms. Furthermore, unauthorized access to IDS configurations can enable attackers to modify detection rules, undermining the system's ability to flag suspicious activity. Without strong protections, these threats compromise both vehicle safety and security.

However, traditional IDS solutions face significant challenges in addressing complex attack vectors such as IP spoofing, rule manipulation, and memory exploitation. These vulnerabilities highlight a critical gap in the current state of IDS development: the lack of robust hardware-enforced mechanisms for ensuring memory safety and rule integrity. Current IDS implementations rely heavily on software protections, which are insufficient against advanced attacks targeting ECUs in resource-constrained automotive environments. This study bridges this gap by leveraging the CHERI (Capability Hardware Enhanced RISC Instructions) architecture to introduce fine-grained memory protection for IDS systems. This novel approach ensures spatial and temporal memory safety, isolating IDS rules from unauthorized access and manipulation while maintaining compatibility with automotive standards.

This study contributes by introducing a novel CHERI-enhanced IDS framework tailored for automotive networks. Key contributions include the following:

- **Integration of CHERI Capabilities:** Demonstrating the application of CHERI to enforce memory protection, preventing unauthorized access to IDS configurations.
- **The 100% Prevention of IDS Rule Manipulation**: Ensuring rule integrity under simulated attack scenarios, including IP spoofing and rule tampering.
- **Real-Time Detection of Spoofing Attacks**: Accurate anomaly detection with minimal performance overhead, ensuring suitability for real-time automotive environments.
- **Evaluation on Realistic Automotive Scenarios**: Simulation of ECUs and attack models to validate the effectiveness of the proposed system, adhering to AUTOSAR standards.

The remainder of this paper is organized as follows: Section 2 provides the Background, outlining key challenges in automotive cybersecurity, IDS vulnerabilities, and an introduction to CHERI technologies. Section 3 presents the Literature Review, discussing related work on IDS frameworks, CHERI applications, and automotive network security. Section 4 explains the Methodology, detailing the experimental setup, CHERI-enhanced IDS framework, and attack simulation scenarios. Section 5 covers the Results and Analysis, highlighting the effectiveness of CHERI in mitigating IP spoofing and rule manipulation attacks. Finally, Section 6 concludes the paper, offering insights into CHERI's potential in real-world automotive applications and recommendations for future research directions.

2. Background

2.1. CHERI Project Overview

CHERI (Capability Hardware Enhanced RISC Instructions) is a novel hardware-based architectural framework designed to provide fine-grained memory safety and compartmentalization [12]. It replaces traditional pointers with capabilities—enhanced constructs carrying metadata, including base, bounds, and permissions, enabling explicit control over memory access. This approach addresses vulnerabilities such as buffer overflows, unauthorized memory access, and code injection attacks.

The CHERI project is a research initiative originally developed through a collaboration between the University of Cambridge, SRI International, and ARM Research, with funding from the DARPA-sponsored CRASH and MRC programs [13]. At its core, CHERI introduces a capability-based model that extends traditional architectures, offering hardware-enforced spatial memory safety by preventing out-of-bounds memory access. This design significantly reduces vulnerabilities like buffer overflows and code injection [14].

Compared to traditional software-only approaches, CHERI provides hardware-enforced isolation with minimal performance overhead. While software solutions rely on runtime checks, CHERI integrates security at the hardware level, ensuring strict enforcement of the principles of least privilege and intentional use. Furthermore, its compatibility

with widely used languages like C and C++ allows seamless integration with existing systems without requiring extensive code rewrites. This compatibility is especially critical for automotive systems, where C/C++ dominate due to their low-level control and performance efficiency [15,16].

In our study, we demonstrated that incorporating CHERI into Intrusion Detection Systems (IDSs) significantly enhances their resilience against advanced threats, including IP spoofing and IDS rule manipulation attacks. Unlike software-only IDS frameworks, CHERI protects rule configurations by enforcing memory bounds, preventing attackers from altering detection logic. This hardware-enforced security guarantees rule integrity, even against attackers using legitimate-appearing spoofed IP addresses.

The ARM Morello board serves as a prominent implementation of the CHERI architecture, showcasing its practical applicability in real-world scenarios [17]. By adopting CHERI-based solutions, the automotive industry can address network and software-level vulnerabilities, achieving robust security by design. Furthermore, CHERI aligns with emerging standards like ISO/SAE 21434 [18,19], making it a viable option for protecting critical systems from sophisticated attacks such as rule manipulation and IP spoofing.

Our research highlights the practicality of CHERI by implementing it on the ARM Morello board, showcasing its low-latency performance and alignment with real-time automotive environments. By adopting CHERI-based solutions, the automotive industry can effectively address network- and software-level vulnerabilities, ensuring a robust defense for connected and autonomous vehicles.

2.2. CHERI and ARM Morello Technologies

The CHERI (Capability Hardware Enhanced RISC Instructions) architecture represents a transformative approach to addressing memory safety vulnerabilities, one of the most critical challenges in modern computing systems. By integrating fine-grained memory protection and scalable software compartmentalization directly into hardware, CHERI provides a robust mechanism for securing both traditional and modern computing environments. The novelty of this research lies in its application of CHERI's memory protection capabilities to enhance IDS systems in automotive networks. Unlike traditional IDS solutions, the proposed framework integrates hardware-enforced security mechanisms to protect IDS configurations and detect advanced threats, such as IP spoofing and rule manipulation, in real time. By embedding CHERI's capabilities into automotive-friendly programming environments like C/C++, this approach ensures robust cybersecurity measures without compromising system performance.

2.2.1. CHERI Architecture Overview

CHERI is built upon two foundational principles [20]:

1. **Principle of Least Privilege**: This principle ensures that each component operates with only those permissions required to perform its function, thus reducing the attack surface area and limiting the damage in case of compromise. In automotive cybersecurity, this becomes even more important to ensure that sensitive subsystems, such as IDS rule configurations and ECU communications, are protected. By keeping each process strictly within its predefined boundaries, CHERI hardware prevents unauthorized accesses to critical memory regions. For example, the capability model of CHERI prevents an attempt by a compromised ECU to access the IDS rules or manipulate configurations and thus isolates and contains the threat.
2. **Principle of Intentional Use**: A principle that enforces the explicit declaration of permissions and memory accesses so that actions actually carried out by a program are intentional and well authorized. This rules out ambiguities that result in misusing

or exploiting privileges, such as the confused deputy problem. Scientific studies have shown that intentional use policies can greatly enhance system robustness since it ensures every action follows some predetermined constraints set in security [20,21].

Taken together, these principles form the bedrock of how CHERI is able to offer fine-grained memory safety and software compartmentalization. In the scope of this work, they guarantee IDS rules are isolated from unauthorized processes, and their usage is only possible through explicit and intentional permissions, which in turn mitigate critical vulnerabilities such as the manipulation of rules and IP spoofing. CHERI capabilities are hardware-enforced tokens that combine memory addresses with metadata, such as bounds and permissions, ensuring that memory access is strictly controlled. Each memory region is protected by a unique capability, and these capabilities are immutable in ways that increase their security. For instance, CHERI employs tagged memory to distinguish valid capabilities from arbitrary data, preventing their corruption or unauthorized modification.

2.2.2. CHERI Enhancements and Implementation

CHERI enhances conventional architectures like MIPS, RISC-V, and ARM by introducing a capability-based memory model that coexists with existing software [22]. This hybrid approach facilitates the gradual adoption of CHERI technologies, enabling legacy code to operate alongside CHERI-enabled applications. The architecture extends pointers to include additional metadata, such as base, bounds, and permissions, ensuring that only authorized memory regions are accessed.

Key features of CHERI include the following:

- Memory Protection: By attaching bounds and permission bits to pointers, CHERI ensures spatial memory safety, preventing unauthorized access beyond designated memory regions.
- Software Compartmentalization: CHERI supports the isolation of software components, enabling secure interaction between mutually untrusting programs or processes.
- Tagged Memory: This feature ensures that capabilities cannot be forged or corrupted, as the hardware tracks their validity independently of their location in memory.

2.2.3. ARM Morello: A Prototype for Commercial Application

ARM Morello, a prototype implementation of the CHERI architecture, extends the ARMv8-A architecture with CHERI capabilities to evaluate its feasibility in commercial processors. Developed as part of a collaboration with the UK government, Morello explores the integration of CHERI's fine-grained memory protection mechanisms into mass-market hardware. By leveraging CHERI's capability-based security model, Morello showcases the practical implementation of hardware-enforced memory safety and software compartmentalization, which are critical for mitigating modern cybersecurity threats and addressing evolving challenges [23].

The Morello board incorporates several CHERI advancements:

- Compatibility with Existing Software: Morello provides a hybrid execution environment, enabling unmodified ARMv8-A applications to coexist with CHERI-enhanced programs. This compatibility reduces adoption barriers for developers and organizations.
- Enhanced Security Properties: By integrating CHERI capabilities, Morello strengthens security guarantees for memory safety, compartmentalization, and access control at the hardware level.
- Performance Considerations: ARM Morello demonstrates that CHERI's additional security features can be implemented with minimal performance overhead, making it viable for real-time systems like those in automotive networks.

2.3. IP Spoofing and IDS Rule Manipulation

Spoofing is one of the most general techniques in cyberattacks, where an attacker forges the source IP address in packets to impersonate a legitimate ECU within an automotive network. This tactic can allow unauthorized access to critical systems, disrupt operations, or inject malicious commands within automotive systems. For instance, an attacker may impersonate a trusted ECU in order to manipulate either the braking or steering functions, which will be very dangerous [24,25]. The distributed and resource-constrained nature of automotive networks makes the spoofed packet detection even more challenging. Most of the automotive IDSs depend on IP-based filtering that is generally too weak to identify forged packets when the attacker uses a legitimate-appearing IP address. For example, some works indicate that spoofing attacks of this type could easily evade traditional IDS systems because of a lack of strong memory protection combined with dynamic rule adaptation mechanisms [26,27]. Once attackers establish a foothold within the network, they may attempt to alter IDS rules to avoid detection in future attacks. This manipulation could involve disabling specific detection rules or creating exceptions for their activity [27]. Without robust memory protection for IDS configurations, rule manipulation remains a persistent vulnerability, as rule-based detection alone cannot prevent access to the IDS's internal logic if an attacker gains sufficient privileges.

The following challenges must be addressed with hardware-enforced security mechanisms, such as CHERI capabilities, for advanced IDS solutions, ensuring strict memory isolation for IDS rule configurations. In particular, storing the IDS rules in protected memory, CHERI efficiently prevents attackers from tampering with the detection logics even when spoofed packets are received. Such hardware-enforced protection strengthens IDS resiliency in accurately detecting anomalies without significant performance trade-offs [26,28]. Furthermore, most of the IP spoofing attacks are part of larger campaigns that involve other attack vectors, such as rule manipulation and denial-of-service attacks. These combined threats indicate the need for comprehensive security frameworks, hardware-enhanced for automotive security. This will help in mitigation not only against IP spoofing but also provide long-term scalability and adaptability against upcoming security challenges in connected and autonomous vehicles by integrating CHERI capabilities within IDS systems.

3. Literature Review

Automotive networks have evolved from isolated, closed systems to highly interconnected networks due to the inclusion of wireless interfaces such as Wi-Fi, Bluetooth, and cellular connectivity [1]. These interfaces, while providing benefits like over-the-air (OTA) updates and telematics services, expose vehicles to numerous cyber risks [24,29]. One of the primary concerns is the complexity and distributed nature of automotive software, with modern vehicles containing over 100 million lines of code distributed across ECUs. These ECUs interconnect through the Controller Area Network (CAN) and Ethernet networks, using protocols such as Scalable service-Oriented Middleware over IP (SOME/IP) to enable efficient communication.

To enhance communication security in automotive electrical and electronic architectures, various approaches have been developed. One notable example is SecOC, standardized by AUTOSAR, which employs symmetric cryptography to secure communication over the Controller Area Network (CAN) [30]. Lee et al. in [31] suggest a hybrid framework leveraging blockchain technology to improve network security within vehicles. Nevertheless, the framework does not include a comprehensive explanation of the protocol's interaction process or demonstrate the practicality of the proposed solution. Islam et al. in [32] present a CAN-based scheme designed to mitigate multiple attack scenarios. They assert that the method ensures security without requiring changes to the underlying CAN protocol.

Additionally, validation experiments are conducted, demonstrating the effectiveness of the proposed approach in preventing and defending against a range of attacks. Hafeez et al. in [25,33] employ a neural network approach to authenticate messages within vehicle networks, including the conventional CAN protocol. They assert that this method significantly enhances the accuracy of Electronic Control Unit (ECU) communication. While the study includes analysis and validation of the approach, it does not provide a detailed assessment of its security. Woo et al. in [34] propose a security architecture leveraging the CAN with Flexible Data Rate (CAN-FD) network to establish a secure transmission environment. They introduce a hierarchical encryption transmission technique for the vehicle network, which has been validated but lacks a formal verification process to confirm its feasibility. Lodge et al. [35] analyze vulnerabilities in CAN and CAN-FD protocols, such as replay, injection, and DoS attacks, emphasizing the need for robust automotive security frameworks. They propose solutions like lightweight cryptography, blockchain-based key provisioning, and multi-layer architectures to enhance communication security. While promising, these methods face challenges in compatibility with legacy systems and increased complexity, highlighting the importance of advanced IDS for Ethernet-based networks.

Kreissl et al. in [36] explored methods for securing SOME/IP-based vehicular communication networks. The authors suggest a central entity to manage key material distribution through (D)TLS during the event group subscription process. However, the "service offer" and "find offer" steps remain unprotected. To support broadcast communication, SOME/IP is enhanced with the TESLA protocol. Despite its advantages, TESLA's delayed authentication for broadcast messages requires modifications to the SOME/IP process flow, such as caching messages until verification data is received. This introduces latency, making it unsuitable for time-sensitive data transmission. Iorio et al. [37] propose a new framework to enhance the security of SOME/IP. While the framework includes security protocol modeling and verification, offering strong assurance of the desired security properties, it imposes significant network overhead and lacks a clear and intuitive formal verification proof. Herold et al. [38] propose an Intrusion Detection System (IDS) for SOME/IP using Complex Event Processing (CEP) to detect attacks. The system identifies threats such as malformed packets, protocol violations, and timing issues. However, like other IDS solutions, it faces challenges with false positives and false negatives. Zorman et al. [28] propose a firewall implementation for embedded automotive systems, utilizing rule-based security and deep packet inspection (DPI) to validate SOME/IP payloads. The solution demonstrates feasibility in resource-constrained environments, highlighting its potential for enhancing security in automotive Ethernet communication. Qi Liu et al. [39] proposed a method that combines deep learning models with residual self-attention to analyze Ethernet traffic, aiming to detect cyberattacks like DDoS and MITM, as well as random hardware failures. This framework demonstrates high accuracy and real-time detection capabilities, highlighting the importance of robust cybersecurity solutions for modern Ethernet-based automotive communication systems.

Recent advances in hardware-based security frameworks have introduced novel approaches to mitigating vulnerabilities in modern automotive systems. Among these, Trusted Execution Environments (TEEs) and Hardware Security Modules (HSMs)have gained prominence. TEEs, such as Intel SGX and ARM TrustZone, provide isolated secure execution environments for sensitive operations, protecting them from unauthorized access by the main operating system. TEEs are widely applied in automotive systems to safeguard critical processes such as firmware updates, cryptographic computations, and secure communications. For example, studies demonstrate that TEEs can effectively isolate critical vehicle control logic, ensuring protection against malware or unauthorized modifications [40,41]. However, TEEs focus primarily on secure execution and do not

directly address memory safety violations, such as buffer overflows or memory corruption, which are significant concerns in automotive ECUs [42].

Hardware Security Modules (HSMs) are dedicated hardware devices designed to securely store cryptographic keys and perform cryptographic operations. In automotive applications, HSMs are integrated into Electronic Control Units (ECUs) to enable secure boot, message authentication, and key management. The AUTOSAR Secure Onboard Communication (SecOC) module relies heavily on HSMs to ensure message integrity and authenticity in Controller Area Networks (CANs) [30,42]. Despite their utility in cryptographic tasks, HSMs do not inherently protect against broader security challenges, such as memory exploitation or unauthorized rule manipulation.

In contrast to TEEs and HSMs, the CHERI (Capability Hardware Enhanced RISC Instructions) architecture provides fine-grained memory safety and compartmentalization by implementing a capability-based hardware model. CHERI prevents memory corruption and unauthorized access to critical IDS configurations, offering hardware-enforced isolation that mitigates advanced attack vectors, such as rule manipulation and spoofing. While TEEs ensure the secure execution of processes and HSMs protect cryptographic assets, CHERI focuses on memory safety at the hardware level, making it uniquely suited for Intrusion Detection Systems in automotive environments [43].

Intrusion Detection Systems (IDS) in automotive environments monitor network traffic to identify suspicious activities that may indicate an attack. Common threats include IP spoofing, where attackers modify IP headers to disguise their identity. While IDS can flag such anomalies, conventional approaches are often limited in countering spoofing when attackers effectively mask their actions. Additionally, if attackers compromise IDS configurations, they could manipulate detection rules to allow undetected future attacks. This research addresses a critical gap in existing solutions, which do not tackle IP spoofing and rule manipulation attacks with a security-by-hardware design approach. To the best of our knowledge, this is the first study to leverage CHERI's hardware-enforced memory safety features to mitigate IDS rule manipulation attacks and detect IP spoofing in automotive networks.

4. Methodology

This section describes the methodology and experimental setup, including the network and components simulated, the configuration of the IDS, and the process of executing spoofing and rule manipulation attacks. The experiment evaluates the CHERI capabilities in restricting unauthorized access and protecting the IDS configuration in an automotive network.

4.1. CHERI in Automotive Security Applications

The integration of CHERI (Capability Hardware Enhanced RISC Instructions) into automotive security offers a transformative approach to mitigating critical vulnerabilities in modern vehicles. CHERI introduces fine-grained memory protection by replacing traditional pointers with capabilities, which include metadata such as base, bounds, and permissions. This architecture prevents unauthorized access and memory safety violations, addressing issues such as buffer overflows and code injection—common attack vectors in automotive systems.

In automotive applications, CHERI can enhance the security of Ethernet-based protocols like SOME/IP by safeguarding IDS configurations and protecting critical ECUs from malicious tampering. By embedding CHERI capabilities, automotive systems can enforce hardware-level isolation for IDS rules, preventing rule manipulation and ensuring secure anomaly detection. Additionally, CHERI mitigates risks associated with IP spoofing by

ensuring that memory access is explicitly authorized, eliminating the possibility of attackers altering network configurations.

This approach aligns with the principles of security by design, enabling robust protections against advanced cyberattacks in the dynamic and interconnected environment of modern vehicles. CHERI's capability-based model ensures that automotive security solutions remain both effective and resilient, providing a foundational framework for safeguarding next-generation vehicle networks.

4.2. CHERI Capability Usage

CHERI (Capability Hardware Enhanced RISC Instructions) enhances security by replacing traditional pointers with capabilities. Capabilities are pointers with extra information, such as memory bounds and access permissions, that strictly control how memory can be accessed. This ensures that a program cannot inadvertently or maliciously access memory outside its designated boundaries. For instance, in standard C programming, a pointer can move freely across memory regions, which might allow it to read or overwrite sensitive data unintentionally. CHERI enforces strict boundaries for each pointer. If a pointer tries to access memory outside its permitted range, the hardware immediately stops the action and raises an alert. Memory safety violations often result from coding errors in unsafe languages like C and C++, making software susceptible to security vulnerabilities and unauthorized access. CHERI introduces capabilities that enforce spatial memory safety, effectively addressing these issues. Figure 1 illustrates a comparison between the behavior of ISO C and CHERI C in handling pointer arithmetic and memory access.

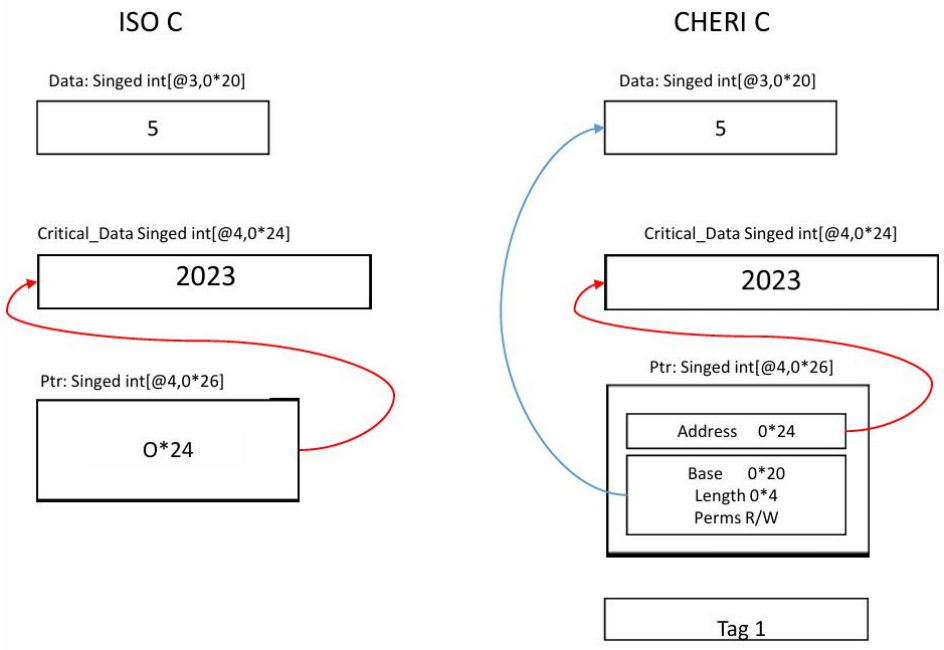

Figure 1. Fine-grained memory protection: comparing ISO C with CHERI C. In ISO C, the flawed program allows undefined behavior, leading to the potential leakage of critical_data due to unrestricted pointer arithmetic. In contrast, CHERI C enforces strict memory bounds through capabilities, ensuring that any out-of-bounds memory access triggers a hardware exception, thereby preventing unauthorized access to critical_data.

In ISO C, two variables, data and critical_data, are declared:

- data is allocated at address 0x20 with a value of 5.
- critical_data is allocated at address 0x24 with a value of 2023.

A pointer, ptr, is initialized to point to data. In ISO C, pointer arithmetic allows ptr to be incremented to point to 0x24, accessing and leaking the value of critical_data unintentionally. This behavior is described mathematically in Equation (1).

$$\begin{aligned} &\text{Variable } data \text{ : signed int, allocated at address } 0x20, \text{ value } 5. \\ &\text{Variable } critical\_data \text{ : signed int, allocated at address } 0x24, \text{ value } 2023. \\ \textbf{ISO C:}\ &\text{Pointer } ptr \text{ : signed int*, initialized to address of } data. \\ &\text{After increment: } ptr = ptr + 1 \implies ptr = 0x24. \\ &\text{Dereferencing: } y = *ptr \implies y = critical\_data = 2023. \end{aligned} \tag{1}$$

In **CHERI C**, pointers are replaced with capabilities that include metadata such as `base`, `length`, and `permissions`. The capability for `ptr` is confined to the memory bounds of `data` (`0x20` to `0x24`). Any attempt to dereference `ptr` outside these bounds (e.g., at `0x24`) triggers a hardware trap, preventing unauthorized access to `critical_data`. This behavior is defined in Equation (2). CHERI enforces the **principle of least privilege** by ensuring `ptr` cannot access memory beyond its defined bounds, maintaining memory safety by design.

$$\begin{aligned} &\text{Variable } data \text{ : signed int, allocated at address } 0x20, \text{ value } 5. \\ &\text{Variable } critical\_data \text{ : signed int, allocated at address } 0x24, \text{ value } 2023. \\ \textbf{CHERI C:}\ &\text{CHERI C Pointer } ptr \text{ : signed int* with metadata as a CHERI capability :} \\ &\quad \text{Address: } 0x20, \text{ Base: } 0x20, \text{ Length: } 0x4, \text{ Permissions: R/W, Tag: Valid.} \\ &\text{Dereferencing: } y = *ptr \implies \text{Bounds check by CHERI: Access Denied if} \\ &\quad ptr > 0x20 + 0x4. \end{aligned} \tag{2}$$

By integrating CHERI capabilities into the program, the hardware ensures that memory safety violations, such as accessing `critical_data` through `ptr`, are impossible. This approach not only protects against common vulnerabilities like buffer overflows and out-of-bounds memory access but also mitigates intentional attacks aimed at exploiting memory corruption.

This demonstrates how CHERI implements **memory safety by design** on our IDS, transforming insecure C code into a secure execution model without significant performance overhead or extensive code changes. As C remains a dominant and automotive-friendly language due to its low-level hardware access, efficiency, and real-time capabilities, CHERI's compatibility with C ensures that security can be integrated into existing automotive systems without compromising these critical advantages.

Implementation: CHERI Usage on IDS

The integration of CHERI (Capability Hardware Enhanced RISC Instructions) within the Intrusion Detection System (IDS) framework was designed to utilize its inherent capability-based memory model to ensure fine-grained memory protection and enhance the security of automotive networks. Specifically, IDS rules are formulated based on SOME/IP packet features as per AUTOSAR standards, ensuring adherence to industry protocols. To maintain robust security, each ECU's rule set is stored in a dedicated and isolated memory region, safeguarded by CHERI's hardware-enforced capabilities. This isolation prevents unauthorized access and ensures that packets are validated strictly against the corresponding ECU's rules, eliminating cross-rule interference.

CHERI's unique memory safety features, such as embedding metadata with bounds and permissions into pointers, enable strict compartmentalization. These capabilities guar-

antee that any access attempt beyond defined memory bounds triggers hardware traps, effectively preventing rule manipulation or unauthorized modification. During operation, incoming SOME/IP packets are analyzed in real time, with their attributes compared against the CHERI-protected rule sets. Packets failing to meet the defined criteria are flagged as anomalous. This implementation ensures that even sophisticated attacks, such as IP spoofing and rule manipulation, are mitigated with 100% accuracy. By leveraging CHERI's hardware-enforced memory protection, the IDS framework enhances resilience while maintaining compliance with real-time performance requirements in automotive environments.

4.3. Memory-Protected IDS

The proposed methodology integrates CHERI (Capability Hardware Enhanced RISC Instructions) to fortify automotive Ethernet-based Intrusion Detection Systems (IDSs) against advanced attack scenarios, such as IP spoofing and rule manipulation. By leveraging CHERI's capability-based memory protection, the framework ensures that IDS rules and configurations are isolated from unauthorized access, even in the event of system compromise. This section explains the approach with examples, attack scenarios, and code-level defenses.

In IP spoofing attacks, an attacker forges the source IP address in network packets to mimic a legitimate ECU, enabling unauthorized access or manipulation of in-vehicle communication. For example, a spoofed packet may claim to originate from 192.168.1.11 (a trusted ECU) but carry malicious payloads targeting safety-critical systems. This can be defended against. Using CHERI, IDS rules are stored with strict memory bounds, ensuring only authenticated packets can access or modify ECU configurations. Spoofed packets are detected by comparing their source IP and payload against predefined rules confined to CHERI-protected memory as in Listing 1.

Listing 1. Function to check ECU rules using CHERI capabilities.

```
int check_ecu_rule(const char *src_ip, const struct someip_packet *
    someip_pkt) {
    for (int i = 0; i < 5; i++) {
        const char *ecu_ip = (const char *)ecu_memory[i]->ecu_ip; //
             Access protected by~CHERI

        // Match source IP and packet attributes
        if (strcmp(src_ip, ecu_ip) == 0 &&
            ntohs(someip_pkt->method_id) == ecu_memory[i]->method_id
                 &&
            ntohs(someip_pkt->client_id) == ecu_memory[i]->client_id
                 &&
            ntohs(someip_pkt->session_id) == ecu_memory[i]->
                session_id) {
            return i;  // Rule matched
        }
    }
    return -1;  // No match found (potential spoofed packet)
}
```

Rule manipulation attacks occur when an attacker gains unauthorized access to the IDS configuration to modify detection thresholds or disable specific rules, enabling malicious packets to bypass detection. **CHERI** isolates IDS rules within protected memory regions, ensuring that only authorized processes can modify these configurations. For example,

the `initialize_ecu_memory` function demonstrates how CHERI enforces tight memory bounds to prevent unauthorized access or modification, with any violations triggering hardware traps as in Listing 2.

Listing 2. Function to initialize ECU memory with CHERI capabilities.

```
void initialize_ecu_memory() {
    for (int i = 0; i < 5; i++) {
        // Allocate memory with CHERI capabilities
        ecu_memory[i] = (struct ecu_rule *__capability)malloc(sizeof
            (struct ecu_rule));
        if (!ecu_memory[i]) {
            perror("Failed to allocate memory for ECU rule");
            exit(EXIT_FAILURE);
        }
    }

    // Assign rules and enforce bounds
    ecu_memory[0]->ecu_ip = (char *__capability)cheri_setbounds("
        192.168.1.11", sizeof("192.168.1.11"));
    ecu_memory[0]->method_id = 0x1234;
    ecu_memory[0]->client_id = 0x5678;
    ecu_memory[0]->session_id = 0x9abc;
}
```

The **cheri_setbounds** function ensures that IDS rules are stored in tightly bounded memory regions. This prevents unauthorized processes from accessing or altering these rules. Any attempt to modify the rules outside authorized contexts results in a deterministic hardware trap.

Furthermore, the IDS must monitor and analyze network traffic for potential anomalies in real time, including spoofed packets and unauthorized rule manipulations. During live packet capture, the IDS compares incoming packets against the CHERI-protected rule set. CHERI ensures that only validated rules are accessed during this process, flagging spoofed or tampered packets. Packets are captured and validated against the CHERI-protected rule set, as shown in Listing 3. Any anomaly, such as a spoofed IP or a rule mismatch, triggers a warning, enabling the IDS to respond in real time. The **libpcap** library is used to capture network traffic in real time. Additionally, the libraries **arpa/inet.h, netinet/ip.h, and netinet/udp.h** are utilized to parse and analyze specific fields in the captured packets, such as IP and UDP headers.

Listing 3. Function to process packets and validate against ECU rules.

```
void process_packet(u_char *args, const struct pcap_pkthdr *header,
    const u_char *packet) {
    struct ip *ip_hdr = (struct ip *)(packet + 14);  // Parse IP
        header
    char src_ip[INET_ADDRSTRLEN];

    inet_ntop(AF_INET, &(ip_hdr->ip_src), src_ip, INET_ADDRSTRLEN);

    if (ip_hdr->ip_p == IPPROTO_UDP) {  // Check for UDP packets
        const struct someip_packet *someip_pkt = (struct
            someip_packet *)(packet + 14 + ip_hdr->ip_hl * 4);
        int rule_index = check_ecu_rule(src_ip, someip_pkt);

```

```
        if (rule_index >= 0) {
            printf("Valid packet matched with ECU%d rules\n",
                rule_index + 1);
        } else {
            printf("Unmatched packet from %s! Possible attack
                detected.\n", src_ip);
        }
    }
}
```

4.4. Experimental Setup

The experimental setup includes several simulated components representing elements of an automotive network, such as ECUs, an IDS, a packet sniffer, and an attacker model, as shown in Figure 2. The network layout mimics a typical vehicular communication structure, where the IDS monitors network traffic between ECUs and detects potential anomalies.

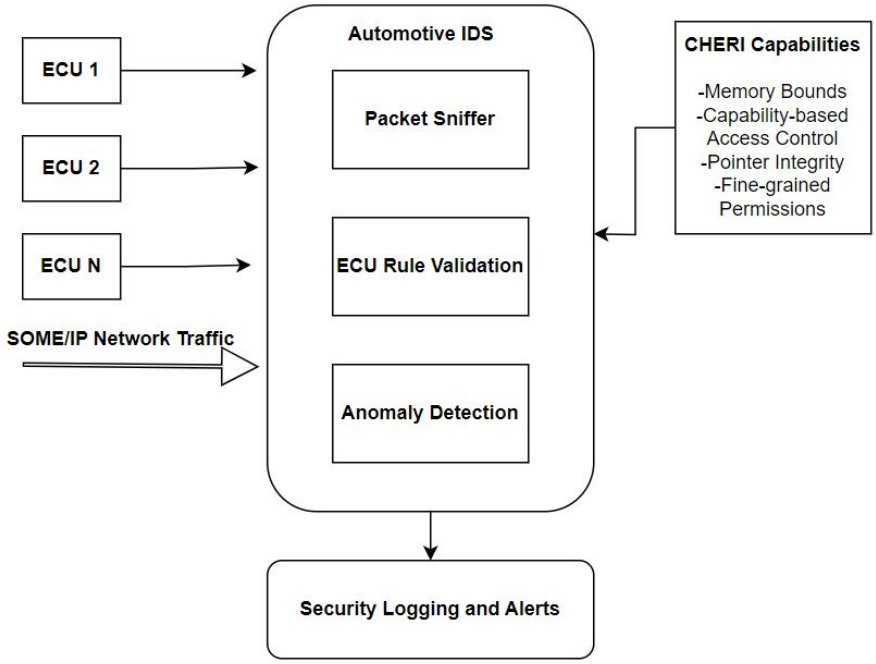

Figure 2. Automotive Intrusion Detection System (IDS) with CHERI capabilities experimental setup: The diagram illustrates the experimental setup of an IDS architecture for automotive networks. SOME/IP traffic from multiple ECUs, along with potential attacker scripts masquerading as random ECUs, is processed through a pipeline comprising packet sniffing, ECU rule validation, and anomaly detection. CHERI capabilities enhance the IDS by enforcing memory bounds, capability-based access control, pointer integrity, and fine-grained permissions. These features ensure robust protection against threats such as spoofed packets, rule manipulation, and unauthorized ECU impersonation. Detected anomalies are logged and trigger alerts, maintaining the integrity and security of the network.

- **ECU Components (ECU1, ECU2, ECU3, ECU4, and ECU5)**: These scripts simulate legitimate ECUs within the vehicle's network, transmitting standard, authorized packets. `ECU1` and `ECU2` generate and send routine communication packets across the network to emulate regular vehicular functions.
- **Intrusion Detection System (IDS)**: This IDS script monitors the network for suspicious packets or anomalies. With CHERI capabilities enabled, the IDS memory is

compartmentalized, ensuring that only authorized processes can access or modify its rule set. The IDS is configured to detect IP spoofing attempts and log unauthorized access to the IDS memory or rule configurations. Separate memory locations are defined on the IDS for each ECU's rules, meaning that each ECU has its own set of rules stored in an isolated memory location.

- **Attacker Model (Attacker)**: The attacker model is designed to send spoofed IP packets, using legitimate IP addresses to disguise malicious activities. The attacker's objective is to bypass the IDS by impersonating a legitimate ECU. If successful, the attacker aims to modify IDS rules to prevent future detections and alter the rules to allow malicious data to be sent to critical ECUs.
- **Packet Sniffer (Sniff)**: This component captures network traffic for analysis and logs all packets, enabling verification of whether spoofed packets are correctly identified by the IDS.

4.5. Execution of Attack

The Attacker: The attacker injects packets with spoofed IP addresses, mimicking the IP addresses of `ECU1` and `ECU2` to evade detection. The IDS monitors network traffic, comparing expected and actual packet source data to detect anomalies. CHERI capabilities within the IDS memory ensure that unauthorized packets cannot influence IDS functionality beyond their initial reception. Subsequently, the attacker attempts to access and modify the IDS rule set. By spoofing legitimate ECU IP addresses, the attacker seeks to gain access to IDS rule configurations and alter them to disable detection for future spoofing activities. However, the CHERI-enhanced IDS enforces strict memory protection, preventing unauthorized access or modification of its rules. Any access violation attempt is flagged by CHERI's memory bounds checks, and the IDS logs the incident while ensuring the attacker's script is terminated.

4.6. CHERI-Enhanced Memory Protections

CHERI capabilities introduce fine-grained memory protections to the IDS. In this experiment, the CHERI memory model is applied to compartmentalize IDS configurations, ensuring the following:

- **Access Permissions**: Only authorized memory regions can be accessed by the IDS. Unauthorized attempts to modify IDS rules or configurations trigger CHERI alerts.
- **Memory Bounds**: The memory boundaries around IDS rules are strictly enforced, preventing any modification or access from processes outside of predefined bounds.
- **Logging Violations**: Each unauthorized memory access attempt is logged, providing detailed records of any access violations. This functionality allows for verification of CHERI's effectiveness in blocking unauthorized access to IDS configurations.

By structuring the IDS with CHERI-enabled protections, the experiment tests how effectively CHERI prevents spoofed packets from bypassing IDS detection and protects against unauthorized rule manipulation.

5. Results and Analysis

The results section presents the findings from the attack simulation, demonstrating the impact of CHERI capabilities on the effectiveness of the IDS in detecting spoofed IP packets and blocking unauthorized access attempts. Table 1 outlines the simulation setup, which adheres to the AUTOSAR standard and is implemented on the Arm Morello board. Powered by a Cortex-A72 processor with up to eight cores, the board operates at typical frequencies around 1.5 GHz and integrates CHERI capabilities for fine-grained memory safety, including capability-based addressing, pointer integrity, and bound checking. These

features provide robust protection against vulnerabilities such as buffer overflows and use-after-free errors. The Morello board is equipped with 1 GB of DDR4 RAM, enabling efficient memory usage through CHERI's tagged memory system, which adds metadata for access control and memory isolation. Storage is supported via onboard eMMC and expansion options, while networking capabilities include integrated Gigabit Ethernet for high-speed data transfer. Additional I/O interfaces include USB 3.0 and USB 2.0 ports, HDMI for graphical output, and UART and JTAG for debugging. The platform also offers PCIe slots for peripheral expansions and GPIO pins for hardware interfacing, making it highly versatile for development and experimentation.

Table 1. Packet sending rates for ECU attack scenarios over 30 min.

ECU Type	Packet Rate (Packets/s)	Number of ECUs	Duration (s)	Total Packets Sent
Telemetry ECU	12.5	1	1800	22,500
Control Signal ECU	30	2	1800	108,000
Sensor Data ECU	75	2	1800	270,000

The IDS successfully flagged 100% of anomalous packets originating from undefined ECUs as shown in Figure 3, demonstrating a high detection rate for known attacks. In our tests, 15 undefined IPs were used to repeatedly send multiple packets over a 30-min timeframe, all of which were accurately detected. Furthermore, packets from known IPs were sent with altered features that deviated from the predefined rules assigned to each ECU and IP. In each scenario, the IDS accurately flagged the anomalies. The CHERI capabilities enabled the IDS to maintain rule integrity, preventing unauthorized packets from influencing detection mechanisms. This demonstrates that CHERI-enhanced memory isolation significantly enhances detection robustness, even against attackers using legitimate IPs as a disguise. The latency of the IDS response to anomalous packets was also measured to evaluate the performance impact of CHERI protections. Observed latency increases were minimal, indicating that the CHERI enhancements do not significantly affect IDS performance. This minimal overhead makes CHERI a viable solution for real-time applications in automotive IDS contexts. Furthermore, no existing literature could be found that use CHERI-based IDS for automotive applications, limiting direct comparisons.

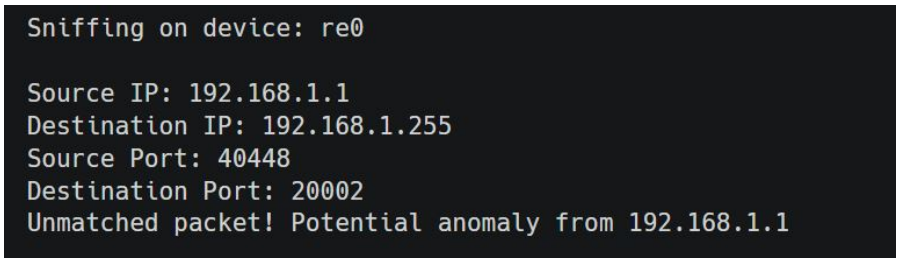

Figure 3. Detection of a potential anomaly in network traffic by the CHERI-enhanced IDS, flagging an unmatched packet from source IP 192.168.1.1 as a potential spoofing attempt.

5.1. IDS Rule Integrity and Unauthorized Access Blocking

During the IDS rule manipulation attempt, the attacker sought to alter the IDS configuration to disable specific detection rules.

- **Memory Access Violation Logs**: The CHERI-enabled IDS logged multiple unauthorized access attempts by the attacker as they tried to modify the IDS rules. Each access violation was flagged by CHERI's capability checks, which block any process outside the authorized scope from modifying memory. This illustrates CHERI's effectiveness

in maintaining the integrity of the IDS rule set, as all modification attempts were successfully prevented as shown in Figure 4.

- **Rule Integrity Preservation**: The IDS maintained its rule configurations intact throughout the attack scenario, as evidenced by the lack of any successful modifications to the IDS rules. CHERI's memory protection mechanisms effectively isolated the IDS configuration, preventing the attacker from altering detection logic. This result demonstrates the value of CHERI capabilities in protecting critical security configurations, ensuring consistent IDS performance even under attack.

```
root@cheribsd:/home/sampath/IDS # ./Attacker
In-address space security exception (core dumped)
root@cheribsd:/home/sampath/IDS # ./Attacker
In-address space security exception (core dumped)
root@cheribsd:/home/sampath/IDS #
```

Figure 4. Execution of the Attacker program resulting in in-address space security exceptions, demonstrating the enforcement of memory safety features by CheriBSD (core dump generated).

5.2. Analysis of CHERI's Impact on IDS Security

The experimental results indicate that CHERI's fine-grained memory protections significantly enhance IDS security in the following ways:

- **Increased Resilience to Spoofing Attacks**: CHERI's hardware-enforced memory compartmentalization makes the IDS resilient to IP-based spoofing attacks. Although the attacker could mimic legitimate IPs, CHERI's memory protection mechanisms prevented unauthorized influence on the IDS logic.
- **Enhanced Rule Protection Against Manipulation Attempts**: The compartmentalized memory model prevented the attacker from modifying IDS rules, effectively securing the IDS from rule manipulation attacks. The unauthorized access logs demonstrate that CHERI successfully restricted all access attempts to the IDS configuration, ensuring rule integrity.
- **Implications for Real-Time Security Applications**: The minimal performance overhead observed in this experiment suggests that CHERI is suitable for real-time security applications in automotive contexts, where high-speed detection and low latency are critical.

The analysis of these results supports the conclusion that CHERI-based memory protection is an effective method for bolstering IDS security against IP spoofing and rule manipulation in automotive networks. To address sophisticated manipulation methods, such as real-time modifications of network traffic, the proposed system leverages several mechanisms:

1. **Hardware-Enforced Memory Isolation**: CHERI's capability-based architecture ensures that IDS rules and configurations are stored in strictly isolated memory regions. This isolation prevents unauthorized access, even if attackers employ advanced methods like spoofing or real-time packet alterations. CHERI's memory bounds enforce deterministic hardware traps when unauthorized attempts are made to access or modify IDS rule sets.
2. **Dynamic Anomaly Detection**: The IDS processes incoming SOME/IP packets in real time, validating attributes like source IP, payload structure, and session identifiers against CHERI-protected rule sets. CHERI ensures that, even if attackers manipulate live network traffic to mimic legitimate patterns, any deviation from the CHERI-enforced rules triggers alerts, ensuring the integrity of the detection mechanism.

3. **Real-Time Logging and Response**: CHERI capabilities enable the real-time logging of unauthorized memory access attempts. These logs allow the IDS to adapt dynamically by flagging patterns of sophisticated manipulation, such as timing inconsistencies or malformed packets, enhancing its ability to counter evolving attack vectors.

To evaluate this, we simulated three types of ECU packet data based on the AUTOSAR standard: telemetry ECU, control signal ECU, and sensor data ECU. These simulations involved injecting spoofed IP packets and attempting to access and modify IDS rules. Each IDS rule was defined within distinct CHERI-protected memory regions corresponding to the specific ECU type. Despite multiple attempts by the attacker to exploit IP spoofing and alter IDS configurations, the CHERI-enforced memory boundaries successfully isolated these memory regions. As a result, all unauthorized access attempts triggered deterministic traps, and the attacker's script terminated with a core dump, indicating that the attack was completely mitigated. By leveraging CHERI's hardware-enforced protections in tandem with dynamic detection capabilities, the proposed IDS effectively counters real-time manipulation techniques, ensuring robust defense for automotive networks.

By enforcing strict memory access boundaries, CHERI not only prevents attackers from bypassing IDS detection mechanisms but also ensures the integrity of IDS rules. These results demonstrate CHERI's robust capability to safeguard against advanced threats, highlighting its significant potential for broader applications in automotive cybersecurity.

5.3. Simulated Scenarios and Scalability Analysis

In this study, we evaluated the performance and scalability of the CHERI-enhanced IDS by simulating three types of ECU packet data based on the AUTOSAR standard: **telemetry ECU**, **control signal ECU**, and **sensor data ECU**. Each ECU type transmitted packets at rates conforming to standard automotive communication protocols. Initially, the simulation included 15 ECUs, with both benign and malicious packets. The attackers employed IP spoofing to impersonate legitimate ECUs and attempted to manipulate IDS rule configurations stored in protected memory.

To assess scalability, the number of simulated ECUs was increased tenfold, from 15 to 150. Each ECU transmitted packets at rates aligned with AUTOSAR specifications:

- **Telemetry ECU**: Transmitted data packets at a rate of 12.5 packets per second.
- **Control Signal ECU**: Maintained a rate of 10 packets per second for command updates.
- **Sensor Data ECU**: Generated sensor readings at 25 packets per second.

Despite the substantial increase in packet volume, the CHERI-enhanced IDS maintained its performance metrics:

- **Detection Accuracy**: Consistently achieved 100% detection of spoofed packets across all scenarios.
- **Latency**: Average response time remained at 12 ms, only a marginal increase compared to the baseline IDS without CHERI, which exhibited a latency of 10 ms.

The implementation of CHERI in IDS systems does introduce certain performance trade-offs, but they are minimal and manageable, as demonstrated by our experimental results. These trade-offs are critical to balancing security with real-time operational requirements in automotive environments. Key findings include the following:

- **Latency Considerations**: Our study observed an average latency increase of 2 ms (from 10 ms to 12 ms) compared to a baseline IDS without CHERI. This marginal overhead, caused by the additional memory bound checks enforced by CHERI, remains well within the acceptable limits for real-time automotive systems. For instance, in automatic braking systems, where the latency threshold is typically below 50 ms, the additional 2 ms does not compromise responsiveness. This ensures that the

enhanced security provided by CHERI does not hinder the performance of critical safety mechanisms. Future work will aim to optimize CHERI's integration into more resource-constrained environments, further minimizing latency to meet the stringent requirements of ultra-fast vehicular control systems.

- **Memory Overheads**: The fine-grained memory protection mechanisms of CHERI require additional memory for storing metadata (capabilities). While this increases memory consumption, the impact was minimized by the efficient memory management features of the ARM Morello board. This trade-off is justified by the significant security benefits gained, including robust protection against memory corruption and unauthorized access.
- **Scalability in High-Throughput Scenarios**: During simulations with increased ECU counts and packet volumes, the CHERI-enhanced IDS maintained consistent performance. Even under high packet rates, the system demonstrated 100% detection accuracy with negligible degradation in processing times, proving its scalability for large-scale automotive networks.
- **Real-Time Packet Processing**: CHERI's capability-based protections allow the IDS to process SOME/IP packets dynamically, ensuring strict adherence to rule configurations. This ensures that security is maintained without significant delays, which is vital for time-sensitive operations like braking and collision avoidance.
- **Hardware Costs and Automotive Integration**: While adopting CHERI requires CHERI-enabled hardware, such as the ARM Morello board, these costs are offset by the long-term benefits of hardware-enforced security. However, it is important to note that current experiments are conducted on research-grade hardware, and further work is needed to deploy CHERI capabilities on automotive-grade devices, such as the SONATA board or equivalent platforms. This step will ensure real-world feasibility, compliance with industry standards, and suitability for production environments in the automotive sector.

In summary, the trade-offs associated with CHERI adoption are outweighed by its advantages, such as enhanced rule integrity, increased resilience against spoofing and manipulation, and real-time security assurance. By aligning with automotive standards like ISO/SAE 21434, CHERI-based solutions present a viable path toward secure, scalable, and high-performance IDS systems for the connected vehicle ecosystem.

6. Conclusions

The results highlight the effectiveness of CHERI capabilities in securing automotive networks against advanced threats such as IP spoofing and rule manipulation. CHERI's fine-grained memory protection mechanisms, particularly suited for systems developed in C/C++, enhance resilience while preserving performance, a critical requirement in automotive environments. C/C++ remain the dominant programming languages in the automotive domain due to their efficiency and close-to-hardware capabilities, making CHERI's compatibility with these languages a significant advantage for adoption in vehicle systems.

Although the results are promising, implementing CHERI in automotive systems requires careful consideration of performance trade-offs, hardware costs, and integration complexity. Future research should focus on optimizing CHERI configurations for resource-constrained environments and extending its applications to other critical automotive components beyond IDS. This study provides a practical framework for implementing CHERI-based IDS in real-world automotive environments. The results confirm the feasibility of deploying CHERI on industry-standard platforms, such as the ARM Morello board, ensuring compatibility with existing automotive communication protocols like SOME/IP.

Testing under diverse cyber–physical attack scenarios and aligning CHERI-based solutions with automotive standards, such as ISO/SAE 21434, would further validate its role in secure automotive design. The proposed IDS demonstrates scalability by efficiently handling increased network traffic volumes and ECU configurations without significant latency or resource overhead, making it viable for large-scale automotive systems. Additionally, deploying CHERI on lightweight, automotive-friendly platforms like the SONATA board could demonstrate its feasibility in real-world systems. Future work should also explore integrating machine learning-based anomaly detection into CHERI-enabled IDS to address advanced threats, including massive DoS attacks and compromised ECUs. Combining machine learning models with CHERI's memory safety can enhance detection accuracy, isolate compromised components, and improve resilience against evolving cyber threats in automotive networks.

This study underscores that CHERI-based memory protection significantly enhances IDS resilience against IP spoofing and rule manipulation attacks. By leveraging CHERI's fine-grained memory control, this work showcases a hardware-based security solution that maintains IDS rule integrity under attack. By addressing critical vulnerabilities, such as spoofed packets and unauthorized configuration changes, CHERI enhances the real-time security of connected and autonomous vehicles. The findings further highlight CHERI's alignment with industry standards, providing a robust foundation for secure automotive design. The findings suggest that adopting CHERI could strengthen the cybersecurity of critical vehicle control systems while seamlessly integrating with the C/C++-centric automotive software ecosystem.

Author Contributions: Conceptualization, C.S.K. and S.M.; methodology, C.S.K.; software, C.S.K.; validation, C.S.K., S.M., X.L., and C.C.; formal analysis, C.S.K.; investigation, C.S.K.; resources, C.S.K. and S.M.; data curation, C.S.K.; writing—original draft preparation, C.S.K writing—review and editing, S.M., X.L., and C.C.; visualization, C.S.K.; supervision, S.M., X.L., and C.C.; project administration, S.M.; funding acquisition, C.S.K. and S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by DSDB cohort 5.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in this study are openly available on the first PoC's GitHub repository (https://github.com/sampathkcs/DSDB, accessed on 1 October 2024). The extended version of the experimental data is available upon request from the corresponding author, as the extended project is being developed in collaboration with an industry partner to increase TR levels.

Conflicts of Interest: Authors Chathuranga Sampath Kalutharage and Saket Mohan were employed by the company Secure Elements. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AUTOSAR	Automotive Open System Architecture
CAN	Controller Area Network
CAN-FD	Controller Area Network with Flexible Data Rate
CEP	Complex Event Processing
CHERI	Capability Hardware Enhanced RISC Instructions

DDoS	Distributed Denial of Service
DPI	Deep Packet Inspection
ECU	Electronic Control Unit
IDS	Intrusion Detection System
ISO	International Organization for Standardization
MITM	Man-in-the-Middle
OTA	Over-the-Air
PCAP	Packet Capture
SOME/IP	Scalable service-Oriented Middleware over IP
TESLA	Timed Efficient Stream Loss-tolerant Authentication

References

1. Luo, F.; Zhang, X.; Hou, S. Research on cybersecurity testing for in-vehicle network. In Proceedings of the 2021 International Conference on Intelligent Technology and Embedded Systems (ICITES), Chengdu, China, 31 October–2 November 2021; pp. 144–150.
2. Abdelkader, G.; Elgazzar, K.; Khamis, A. Connected vehicles: Technology review, state of the art, challenges and opportunities. *Sensors* **2021**, *21*, 7712.
3. Kabilan, N.; Ravi, V.; Sowmya, V. Unsupervised intrusion detection system for in-vehicle communication networks. *J. Saf. Sci. Resil.* **2024**, *5*, 119–129.
4. Almehdhar, M.; Albaseer, A.; Khan, M.A.; Abdallah, M.; Menouar, H.; Al-Kuwari, S.; Al-Fuqaha, A. Deep learning in the fast lane: A survey on advanced intrusion detection systems for intelligent vehicle networks. *IEEE Open J. Veh. Technol.* **2024**, *5*, 869–906.
5. Meneghello, F.; Calore, M.; Zucchetto, D.; Polese, M.; Zanella, A. IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices. *IEEE Internet Things J.* **2019**, *6*, 8182–8201.
6. Al-Jarrah, O.Y.; Maple, C.; Dianati, M.; Oxtoby, D.; Mouzakitis, A. Intrusion detection systems for intra-vehicle networks: A review. *IEEE Access* **2019**, *7*, 21266–21289.
7. Zuech, R.; Khoshgoftaar, T.M.; Wald, R. Intrusion detection and big heterogeneous data: A survey. *J. Big Data* **2015**, *2*, 3.
8. Hamada, Y.; Inoue, M.; Adachi, N.; Ueda, H.; Miyashita, Y.; Hata, Y. Intrusion detection system for in-vehicle networks. *SEI Tech. Rev.* **2019**, *88*, 76–81.
9. Samaila, M.G.; Neto, M.; Fernandes, D.A.B.; Freire, M.M.; Inácio, P.R.M. Challenges of Securing Internet of Things Devices: A Survey. *Secur. Priv.* **2018**, *1*, e20. https://doi.org/10.1002/spy2.20.
10. Greenberg, A. *Millions of Vehicles Could Be Hacked and Tracked Thanks to a Simple Website Bug*; Wired: London, UK, 2021.
11. Smith, J.; et al. Connected Cars: Cybersecurity Challenges and Future Directions. *IEEE Commun. Surv. Tutorials* **2020**, *22*, 1686–1721. https://doi.org/10.1109/COMST.2020.2973312.
12. Woodruff, J.D. *CHERI: A RISC Capability Machine for Practical Memory Safety*; Technical Report UCAM-CL-TR-858; University of Cambridge, Computer Laboratory: Cambridge, UK, 2014.
13. University of Cambridge. The CHERI Project. Available online: https://www.cl.cam.ac.uk/research/security/ctsrd/cheri/ (accessed on 2 December 2024).
14. Davis, B.; Watson, R.N.; Richardson, A.; Neumann, P.G.; Moore, S.W.; Baldwin, J.; Chisnall, D.; Clarke, J.; Filardo, N.W.; Gudka, K.; et al. CheriABI: Enforcing valid pointer provenance and minimizing pointer privilege in the POSIX C run-time environment. In Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, Providence, RI, USA, 13–17 April 2019; pp. 379–393.
15. Grisenthwaite, R. A Safer Digital Future, by Design. 2019. Available online: https://www.arm.com/blogs/blueprint/digital-security-by-design (accessed on accessed on 20 November 2024).
16. Embedded.com. Why C/C++ Dominate in Automotive Systems. Available online: https://www.embedded.com/ (accessed on 2 December 2024).
17. ARM. Introducing the Morello Prototype Board. Available online: https://www.arm.com/morello (accessed on 2 December 2024).
18. *ISO/SAE 21434*; Road Vehicles—Cybersecurity Engineering. International Standards Organization: Geneva, Switzerland, 2020. https://www.iso.org/standard/70918.html (accessed on 2 December 2024).
19. Ibrahim, M.E.; Abbas, Q.; Daadaa, Y.; Ahmed, A.E. A Novel PPG-Based Biometric Authentication System Using a Hybrid CVT-ConvMixer Architecture with Dense and Self-Attention Layers. *Sensors* **2024**, *24*, 15. https://doi.org/10.3390/s24010015.
20. Saltzer, J.H. Protection and the control of information sharing in Multics. *Commun. ACM* **1974**, *17*, 388–402.
21. Woodruff, J.; Watson, R.N.M.; Chisnall, D.; Moore, S.W.; Anderson, J.; Davis, B.; Laurie, B.; Neumann, P.G.; Norton, R.; Roe, M. The CHERI Capability Model: Revisiting RISC in an Age of Risk. In Proceedings of the 41st Annual International Symposium on Computer Architecture (ISCA), Minneapolis, MN, USA, 14–18 June 2014; pp. 457–468. https://doi.org/10.1145/2665671.2665740.

22. Kho, N.M.D.; Uy, R.L. MIPSers: MIPS extension release 6 simulator. In Proceedings of the 2017IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Manila, Philippines, 1–3 December 2017; pp. 1–6.
23. Grisenthwaite, R.; et al. The Arm Morello Evaluation Platform—Validating CHERI-based Security in a High-performance System. *IEEE Micro* **2023**. Available online: https://www.cl.cam.ac.uk/research/security/ctsrd/pdfs/202305ieeemicro-morello-platform.pdf (accessed on 2 December 2024).
24. Aliwa, E.; Rana, O.; Perera, C.; Burnap, P. Cyberattacks and countermeasures for in-vehicle networks. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 21.
25. Hafeez, A.; Topolovec, K.; Awad, S. ECU fingerprinting through parametric signal modeling and artificial neural networks for in-vehicle security against spoofing attacks. In Proceedings of the 2019 15th International Computer Engineering Conference (ICENCO), Cairo, Egypt, 29–30 December 2019; pp. 29–38.
26. Song, S.; Manikopoulos, C.N. IP Spoofing Detection Approach (ISDA) for Network Intrusion Detection System. In Proceedings of the 2006 IEEE Sarnoff Symposium, Princeton, NJ, USA, 27–28 March 2006; pp. 1–4. https://doi.org/10.1109/SARNOF.2006.4534792.
27. Parameshwarappa, P.; Chen, Z.; Gangopadhyay, A. Analyzing attack strategies against rule-based intrusion detection systems. In Proceedings of the Workshop Program of the 19th International Conference on Distributed Computing and Networking, Varanasi, India, 4–7 January 2018; pp. 1–4.
28. Zorman, E.H.; Isoaho, J.; Mohammad, T. Implementation of a SOME/IP Firewall with Deep Packet Inspection for Automotive Use-Cases. Ph.D. Thesis, University of Turku, Turku, Finland, 2024.
29. Ghadi, M.; Sali, Á.; Szalay, Z.; Török, Á. A new methodology for analyzing vehicle network topologies for critical hacking. *J. Ambient Intell. Humaniz. Comput.* **2021**, *12*, 7923–7934.
30. AUTOSAR. *Specification of Module Secure Onboard Communication—CP*; Technical Report, Release 4.2.2; AUTOSAR: Hörgertshausen, Germany, 2017.
31. Lee, T.Y.; Lin, I.A.; Liao, R.H. Design of a FlexRay/Ethernet gateway and security mechanism for in-vehicle networks. *Sensors* **2020**, *20*, 641.
32. Islam, R.; Refat, R.U.D. Improving CAN bus security by assigning dynamic arbitration IDs. *J. Transp. Secur.* **2020**, *13*, 19–31.
33. Hafeez, A.; Malik, H.; Irtaza, A.; Uddin, M.Z.; Noori, F.M. Enhancing ECU identification security in CAN networks using distortion modeling and neural networks. *Front. Comput. Sci.* **2024**, *6*, 1392119.
34. Woo, S.; Jo, H.J.; Kim, I.S.; Lee, D.H. A practical security architecture for in-vehicle CAN-FD. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2248–2261.
35. Lodge, N.; Tambe, N.; Saqib, F. Addressing Vulnerabilities in CAN-FD: An Exploration and Security Enhancement Approach. *IoT* **2024**, *5*, 290–310.
36. Kreissl, J. Absicherung der Some/IP Kommunikation bei Adaptive Autosar. Master's Thesis, Universität Stuttgart, Stuttgart, Germany, 2017.
37. Iorio, M.; Reineri, M.; Risso, F.; Sisto, R.; Valenza, F. Securing SOME/IP for in-vehicle service protection. *IEEE Trans. Veh. Technol.* **2020**, *69*, 13450–13466.
38. Herold, N.; Posselt, S.A.; Hanka, O.; Carle, G. Anomaly detection for SOME/IP using complex event processing. In Proceedings of the NOMS 2016—2016 IEEE/IFIP Network Operations and Management Symposium, Istanbul, Turkey, 25–29 April 2016; pp. 1221–1226.
39. Liu, Q.; Li, X.; Sun, K.; Li, Y.; Liu, Y. SISSA: Real-time Monitoring of Hardware Functional Safety and Cybersecurity with In-vehicle SOME/IP Ethernet Traffic. *IEEE Internet Things J.* **2024**, *11*, 27322–27339.
40. McKeen, F.; Alexandrovich, I.; Berenzon, A.; Rozas, C.; Shafi, H.; Shanbhogue, V.; Savagaonkar, U. Innovative instructions and software model for isolated execution. In Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy, Tel-Aviv, Israel, 23–24 June 2013; pp. 1–8.
41. ARM. Building a Secure System Using TrustZone Technology. 2019. Available online: https://developer.arm.com/documentation (accessed on 2 December 2024).
42. Kumar, A.; Gholve, A.; Kotalwar, K. *Automotive Security Solution Using Hardware Security Module (HSM)*; Technical Report, SAE Technical Paper; SAE International: Warrendale PA, USA, 2024.
43. Saltzer, J.H.; Schroeder, M.D. The protection of information in computer systems. *Proc. IEEE* **1975**, *63*, 1278–1308.