An Evaluation Framework for Intrusion Prevention Systems on Serial Data Bus Networks

Matthew Rogers
MITRE, University of Oxford
Washington D.C, USA
matthewrogers@mitre.org

Kasper Rasmussen
University of Oxford
Oxford, United Kingdom
kasper.rasmussen@cs.ox.ac.uk

ABSTRACT

Serial data bus networks are a crucial and vulnerable part of modern vehicles and weapons systems. Increasing concern over these networks is resulting in increased demand for intrusion prevention systems (IPSes) to stop attacks, not just detect them with an intrusion detection system (IDS). Considerations must be made to avoid the IPS becoming a de facto attacker. A defender needs to understand what attacks their IPS can safely prevent and how an attacker might circumvent their system. To enable this understanding, we propose a protocol-agnostic evaluation framework which: determines the viability of an IPS for different attack vectors, scores the suitability of an IDS to powering an IPS for certain attacks, and scores the efficacy of the IDS itself against those same attacks. With our framework we analyze IDS and IPS technologies for the CAN and MIL-STD-1553 serial data bus networks. These case studies demonstrate how a defender can use our framework to identify limitations in their IDS, while gearing the aspects of the IDS that work best towards safely powering an IPS. Our framework allows a defender to approach any potential security system fully aware of its limitations and how well it serves their own threat model.

CCS CONCEPTS

• **Security and privacy → Systems security**; **Intrusion detection systems**; • **Computer systems organization → Embedded and cyber-physical systems**.

KEYWORDS

Cyber Physical Systems, Security, CAN Bus, MIL-STD-1553

ACM Reference Format:

Matthew Rogers and Kasper Rasmussen. 2023. An Evaluation Framework for Intrusion Prevention Systems on Serial Data Bus Networks. In *Proceedings of the ACM ASIA Conference on Computer and Communications Security (AsiaCCS '23).* ACM, New York, NY, USA, 13 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Serial data bus networks are an integral piece of decades worth of vehicles and weapon systems from all around the globe. A decade of research demonstrates how vulnerable these networks are to cyber attacks [17], [1], [2]. A critical first step to countering this vulnerability is the intrusion detection system (IDS) research built to reveal these attacks. The natural next step is an intrusion prevention system (IPS), which stops an attack after an IDS flags it. After all, if you know what an attack looks like then why allow the attack to complete? Industry and government maintainers of serial data bus networks want this capability to protect themselves from cyber attacks [28], [32], [19], but safely implementing an IPS is tricky. The IDS behind the IPS must meet certain safety criteria and the IPS needs to be built around an attacker trying to circumvent it. The lack of nuance around calls for IPS technology leads to ambiguity in how exactly IPSes work and what their limitations are. Grouping all intrusion prevention style capabilities into a single bucket oversimplifies what IPSes can do, leads to dangerous gaps in coverage, and results in unspoken assumptions about adversary capabilities. This paper solves these problems by designing a framework that focuses on how to safely integrate IDS and IPS technology.

The first thing to clarify is what exactly an IPS does or at least what an IPS is meant to do. IPS can either imply cancelling out an individual message or outright turning off the attacker's device. The technology and assumptions behind these capabilities are completely different. Cancelling a single message is theoretically simple; mangle an attack message and the network ignores it. This capability places several requirements on the security system. Mangling a signal implies the IPS, and by extension the IDS, can act fast enough to mangle a message before it is accepted by the rest of the network. Speed often comes at the cost of complexity, which may mean an IPS is not suitable for some threat models because the IDS powering it is computationally limited. As for the other type of IPS, turning off an attacker's device has nothing to do with speed, only what the attacker can do. The assumption that an attacker will kindly listen to a shutdown message, accept a firmware upload, or follow some protocol-specific error process is laughable for advanced attackers, but limited attackers do exist for many threat models. A defender can make make assumptions about an attacker but these limitations need to be factored into the defender's security model.

The reality is that no IPS or IDS can do everything; all technical capabilities come at a cost. A defender can only implement so many security systems before the cost becomes prohibitive or the complexity is so high that the signal is not worth the noise. This is the core problem this paper is solving. A defender should be able to examine their current security system and understand the coverage gaps, understand what techniques can be improved, and understand what detection systems can safely be used with a prevention system.

To enable a defender to make more informed security decisions, we propose an evaluation framework that works on three axes: the IPS's viability for different attack vectors, the IDS's suitability to

AsiaCCS '23, July 10–14, 2022, Melbourne, Australia

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

IPS for different attacks, and the IDS's efficacy at detecting those same attacks. We choose these axes as they represent the core of the IPS, the intersection of IDS and IPS, and the core of the IDS. If any of these axes fail, then the entire security system fails. Our axes are broken down for each attack and attack vector. The system model for different serial data bus network may vary dramatically, but the high level attacks and attack vectors are generalizable across all serial data bus networks. For example, an IPS can assume trusted firmware on each device, which may work fine for remote attacks but may not work for supply chain attacks or implants. Nothing is inherently wrong with limitations as long as they are known quantities. To this end, our evaluation framework provides scores for each attack and attack vector across our three axes. To simplify the evaluation process, we built an open-source tool which helps an evaluator test their IDS by randomly adding attacks to a dataset [26].

In this paper we provide case studies for the CAN and MIL-STD-1553 serial data bus protocols. These demonstrate that after using our evaluation framework an evaluator will know: what attack vectors their IPS is secure against, what aspects of their IDS can safely power their IPS, and how well their IDS can detect different attacks. Our evaluation framework allows a defender to make more informed decisions and has the potential to turn IPS for serial data bus networks from a desirable but terrifying concept, to a real tool that can be safely used in the right scenarios. We summarize our contributions as: an evaluation framework which scores IDS/IPS systems based on their efficacy and viability for different attack types and attack vectors, a new categorization strategy for prevention systems, and a protocol-agnostic, open-source attack emulation tool to simulate attacks on serial data bus datasets.

2 RELATED WORK

The most direct comparison to this work is other evaluations of existing IDS technologies. While there are papers that compare different solutions, these largely examine them as standalone solutions rather than integratable components for a larger security goal [35], [3], [**?**]. We are unaware of any comparisons or analyses of IPS technologies. Outside of serial data bus networks, network IPS is a cleaner problem with core routing infrastructure acting as an adversary-in-the-middle device. The distinctions we draw between stopping an attack and stopping attackers from speaking are still relevant, though the safety considerations built into vehicle networks often provide more potential IPS mechanisms [6] [**?**].

For the purposes of our own evaluation framework we assume that a defender has a large list of detection and prevention systems. When deciding on what security system to deploy to a serial data bus network, they plug the prevention system into the detection system and hope it works. Our evaluation framework lets a defender know what scenarios that IPS and IDS combination will work in, if any. The rest of this section presents a list of prevention and detection systems that can be matched up. This paper largely focuses on the CAN and MIL-STD-1553 serial data bus protocols. CAN is common in academic work and used in consumer automobiles. MIL-STD-1553 is less common but is used in military systems. CAN places equal trust in its computers while MIL-STD-1553 relies on a bus controller telling each device when to speak. Discussing two serial data bus protocols with such different architectures allows us to demonstrate that our evaluation framework is applicable to any serial data bus security problem.

2.1 Existing Prevention Systems

IPS technology on serial data bus networks is poorly researched in comparison to IDS technology. Likely because once one IPS solution works for a protocol, there is a limited amount of novel research that can be done without integrating a complicated IDS. Add the complicated collection setup required to get IDS systems to operate before a message is finished transmitting, and you get few integrated IDS/IPS systems.

Let us first discuss prevention systems which are connected to the serial data bus network as if they were an additional device, which we will refer to as a Line Replacable Unit (LRU). LRUs are often referred to as Electronic Control Units (ECUs) in CAN research, and Remote Terminals (RTs) in MIL-STD-1553 research. We use LRU as a generic term. One of the first popular IPS proposals is CANStomper [13], which intentionally flips a bit in the last 11 bits of the CAN bus to trigger a CAN error frame, causing the victim message to be ignored by the rest of the bus. CANStomper triggers this prevention system using an allowlist-based detection system which alerts whenever a new ID appears on the bus. The complexity of the IDS aside, this work demonstrates that intrusion prevention on individual messages is possible with cheap FPGAs.

On the other end of the IPS spectrum is the idea of targeting not individual attack messages, but the attacker's device. Takada [31] does this in a similar method to CANStomper, but instead of monitoring the line for an attack and interrupting it, this IPS copies the attacker message, transmits at the same time as it, then changes one bit to cause an error. The CAN bus then automatically replays both messages, resulting in another error. This loop repeats until the attacker's message changes, or the attacker reaches a 'bus-off' state which disables their transmitter. The benefit of this approach is that it prevents the attacker from sending more attacks, but it does introduce a limitation on the attacker. CAN devices have a mechanism to force themselves out of the bus-off state. Takada assumes the attacker did not think to account for this state, or is somehow restricted such that they cannot exit the bus-off state.

The previous solutions are designed around the idea of a single device connected to the bus, as most intrusion detection research follows this model. The other approach is in-line detection systems which intercept packets as they are communicated and drop them if they are anomalous [20] [15]. This approach can be extremely effective, but it comes with higher integration costs as one of these computers needs to be put in-line for each LRU. Given the increasing numbers of LRUs in vehicles [10], this approach does not scale well.

2.2 Existing Detection Systems

We observe four distinguishable types of detection systems for serial data bus protocols. These are timing-based, voltage-based, data-based, and protocol-based. We believe cryptography based solutions [14], [9], [24], [22] are outside the scope of IDS/IPS, in part because IDSes still provide a useful layer of security even with a fully cryptographically authenticated bus.

Timing-based systems generally rely on the timing between messages [29] [7], acting on the assumption that an attacker spoofing

a legitimate device will lead to an anomaly in the time interval between messages. We have found that straight interval approaches like those used by Song [29] are often too limited to messages with tight, consistent transmission patterns. Meaning an attacker can bypass timing intervals by selecting messages with varied timing intervals. This research can be expanded through greater contextual awareness for why timing intervals change. Higher level timing research like Cho's clock skew analysis [7] and Genereux's response time analysis [12] provide the detection system with fingerprinting capabilities. However, the greater complexity of the analysis makes it spoofable by an attacker, as shown by Sagong emulating clock skew [27]. We include state machine based solutions in the timing category as they try to predict the time slots a message could appear in [36], [30], [5], [23]. Orly [30] demonstrates this by predicting the order of MIL-STD-1553 messages and alerting on any deviations.

Data-based systems assume some level of correlation between the engineering values transmitted by different LRUs. In aggregate these correlations create a model of the system. If an attacker transmits a message that somehow breaks the invariance of the overall system, then an anomaly is detected [34], [25]. This approach is good at detecting unexpected deviations in engineering values, but can be defeated by an attacker who corrupts an important LRU, or an attacker slowly creating small deviations.

Voltage-based systems fingerprint each LRU and search for unseen fingerprints that indicate a new device is transmitting on the system [8], [16], [11], [4], [30]. These systems are good at detecting when one device pretends to be another but have weak security properties when an existing device is taken over. The key problem with voltage based approaches is that they require consistent retraining to account for environmental impacts on vehicle circuity. Retraining provides an easy attack window for persistent attackers. Corrupted ECUs can bypass voltage fingerprinting as long as they only transmit messages their corrupted device would transmit.

Protocol-based detection systems are usually limited in scope, being focused entirely on built in error processes and attacks specific to a protocol, i.e. CAN, MIL-STD-1553. These are often proposed in the countermeasures sections of papers proposing some new attack [6], [21], [18]. The nature of these attacks makes them difficult to detect with any of the aforementioned generalist techniques. As a result protocol-specific detection systems are an inevitable requirement of any security system.

In terms of detecting attacks - timing and voltage are generally good at spoofing attacks but bad at detecting an attacker who has taken over an existing ECU and is transmitting messages the victim ECU would normally send at the right time. An advanced attackers can transmit in the appropriate time intervals, making voltage and timing based detection systems have weak security guarantees against corrupted LRUs. Data-based detection systems have strong security guarantees against attackers sending malicious data across the bus, but the threshold for what qualifies as malicious is variable and may need tuning from vehicle to vehicle to avoid false positives. The security guarantee of data-based systems against corrupted LRUs is significantly weaker if the attacker is in control from the moment the vehicle is started, as then there are no significant deviations and the corrupted LRU is the source of truth for some data. This allows an attacker to transmit whatever they wish as long as they only transmit malicious messages that are not a function of other bus traffic. More research is necessary to address this persistent corrupted attacker. Protocol based detection systems offer no rigid security guarantee.

3 ADVERSARY MODEL

In this paper we define two adversaries, both with the goal of transmitting, modifying, or cancelling messages on a serial data bus network. The first is a strong attacker. strong attackers are in full control of their attack vector, and everything on it. Including any firmware, protocol-specific controller chips, and the messages they choose to process. We assume they have persistence on their attack vector, and cannot be reliably erased. In contrast, weak attackers have limited control over their attack vector. Weak attackers are restricted to simply transmitting and receiving messages on standard hardware, with no control beyond which messages are sent when.

Through strong and weak attackers we can properly evaluate how effective and resilient an IPS is. For example, if the IPS requires transmitting a complete message on the bus to disable an attacker, then the IPS would work on a weak attacker, but not a strong one. This informs how we evaluate prevention systems, ensuring that the chosen IPS is effective against whichever attacker the defender expects to face.

4 CATEGORIZING IPSES AND ATTACK TYPES

Before defining our evaluation framework we must address two issues: the ambiguity of how a given intrusion prevention system works, and the lack of standardized attack types for IDS evaluation. IPS ambiguity leads to a defender not understanding when their prevention capability can be safely used or when it can be circumvented. To address this we identify two distinct types of prevention systems. After defining these types of prevention systems a defender can better evaluate their IPS. This leaves their IDS. A plethora of attacks exist to test an IDS against but not every IDS addresses every type of attack. Not every IDS needs to detect every type of attack, but they do need to be tested against every type of attack to ensure the defender knows the limitations of their security system. By defining high-level standardized attack types a defender can intentionally cover all possible ways of interacting with the serial data bus network.

4.1 Types of Prevention Systems

We put intrusion prevention systems into two categories: cancelling individual attack messages and preventing an attacker from sending further messages. We will refer to these as Message Intrusion Prevention System (MIPS) and LRU Intrusion Prevention System (LIPS) respectively. Put another way, MIPS is prevention targeting the attack and LIPS is prevention targeting the attacker. As an attack message and the attacker are the highest abstractions of executing an attack, these IPS definitions are the highest level abstraction past simplifying everything into a single name. In this section we evaluate the pros, cons, configuration considerations, and feasibility of MIPS and LIPS.

4.1.1 MIPS. Message intrusion prevention works by cancelling out a message as it transmits across the bus. It does this by exploiting a common error process on networks where a message with invalid encoding is dropped by the controller before it is processed as a

'message' by the victim LRU. MIPS intentionally mangles the signal of an attack by briefly driving the bus voltage in the opposite direction. The main benefit of this approach is that MIPs technologies are not concerned with where the attack is from. By avoiding the source attestation problem, MIPS works even if the attack vector is spoofing another device. MIPS has two primary limitations.

The first is based on timing. In order for MIPS to work it has to mangle a message before it finishes transmitting. If the attack is only detectable in the last few bits of a message, then MIPS has limited time to execute. MIPS inherently favors detection techniques which make it clear that an attack is occurring near the beginning of the message, this way time-to-detect is less of an issue.

The second problem is that MIPS only stops a single attack, without addressing the root cause of how that attack was transmitted. The fact that an individual attack was prevented is undeniably beneficial, but at the end of the day MIPS's technique is akin to a denial of service (DoS) attack. If an attacker rapidly transmits attacks on the bus, then MIPS is simply performing a DoS attack on the bus. This is better than whatever the attacker's payload would have done, but only if the defender knows what the system does when it stops receiving valid messages. Most protocols do not specify this behavior and instead leave it up to the manufacturers, making it vary from system to system. The knowledge of what happens when the system stops receiving messages makes the effect of the MIPS DoS attack predictable, which is always better than an unpredictable attack. If the DoS attack is not a flood of messages but instead an attacker mangling messages on the bus, then MIPS is ineffective, as it cannot correct mangled signals.

Based on these limitations we make two conclusions. MIPS is best against attackers with limited attack windows - otherwise MIPS is akin to a denial of service attac, albeit an intentional one. And MIPS requires early detection metrics to be reliable at stopping attacks. This usually implies that data based detection techniques are sub-optimal for prevention, as the data payload is often near the end of a serial data bus packet.

4.1.2 LIPS. LRU intrusion prevention is stopping the attacker from executing any future attacks after the first attack, making LIPS the ideal silver bullet solution for protecting serial data bus networks. Unsurprisingly, LRU prevention systems only work in a limited set of scenarios, and can be avoided entirely by some attackers.

LIPS relies on the idea that there is some way of disabling the transmitter of another system. Such as a message or sequence of events that cause the victim to turn off, or be disconnected from the system. In our experience a built in off-switch exists in many serial data bus protocols. How exactly it works varies from serial data bus protocol to serial data bus protocol, and will be elaborated on in Section 7. We can generally split LIPS techniques into four categories: firmware uploads, protocol-specific 'turn off' messages, device disconnection, and error manipulation.

Firmware uploads are based on the following attack scenario: somehow an attacker managed to upload new firmware to a victim LRU. By uploading known good firmware to that victim LRU, the attack will stop happening. The techology to perform this firmware upload on the fly, literally, was demonstrated by the U2-Dragonfly [33]. Its unlikely most systems will be able to commence a firmware upload without stopping the vehicle first, making it a slow prevention system. But it is worth noting as a viable option.

Protocol-specific turn off messages are exactly that - a message in the serial data bus protocol that tells an LRU to disable itself. These are more likely to occur in bus controller based protocols, where the bus controller has complete control over the flow of traffic on the bus. In Section 7 we will discuss examples of this technique in the MIL-STD-1553 protocol.

Disconnection is the idea of physically disconnecting LRUs from the network. We are unaware of cases where this is built into the protocol, though in-line prevention systems [15] have integrated it into existing systems. Adding the capability to disconnect LRUs to a system is dangerous, as it means an attacker can now also disconnect LRUs from the bus.

The final technique is error manipulation. Many serial data bus systems have a built in error process to ensure the safety of the vehicle should a single LRU malfunction. By intentionally producing errors in a target LRU, the LIPS can make it believe it is in an erroneous state, such that it disables itself. This often is the result of many MIPS triggers in a row but the idea of error states and targeted messages raise an important question for LIPS. Will an attacker actually listen when it is told to turn off?

LIPS fundamentally requires the attacker to have limited control over their attack vector or to be ignorant of LIPS. This does not mean LIPS is impractical, it raises the minimum technical competency of the attacker. Much like how a traditional malware author must now bypass stack canaries and ASLR, the serial data bus attacker must now gain further control over their victim. Specifically they must gain control over the 'controller', the chip which receives messages from the bus, and enforces the protocol. If they can selectively disable their receiver, or ensure the controller ignores certain messages, such as firmware messages, then they can bypass LIPS. But this often requires rewriting the firmware of the controller. An attacker gaining enough control over a victim to transmit a message, does not necessarily have the ability to reflash the controller.

4.2 Types of Attacks

To create an effective IDS and IPS evaluation framework we need attacks to evaluate the security system against. But selecting a subset of attacks will inevitably lead to gaps in our evaluation. Instead, we need to base our evaluation on higher level ways of interacting with the bus. Let us examine what actions are possible on a serial data bus network and which of those we would consider an attack. By examining attacks through the lens of how anybody can interact with the bus we can avoid gaps in coverage.

The first choice an LRU makes is whether to speak when the bus is free, or the bus is busy. If the bus is busy then they are speaking while another message is being transmitted on the bus. At this point two outcomes can occur: either our LRU's transmission collides with the active message, mangling it such that the message is considered invalid and cancelled, or our LRU's transmission collides with the active message such that the message is still valid and the message is manipulated by our LRU speaking. This means if the bus is busy an LRU can either cancel out an existing message, or manipulate it to be different. A legitimate LRU would never

intentionally collide with another message, meaning that both of these options are potential attack paths.

Now let us assume the bus is free, meaning our LRU can transmit a normal message. When transmitting on the bus the first thing an LRU decides is what type of message it is going to send. It can either send a message as itself or it can send a message as someone else, i.e., the engine can send a message as the engine, or as the transmission. Again this is not a choice for most LRUs, but imagine an implant or a corrupted LRU. An implant is a new device connected to the network, it cannot help but pretend to be something else. A corrupted LRU is the aforementioned scenario where our engine is pretending to be the transmission. We refer to this scenario where a device is pretending to be something it is not as spoofing. But let us say the LRU is claiming to be the device it actually is. Now they are transmitting a message on the bus, and all that is left to determine is if the message itself is malicious. Or more specifically, if the data contained within the message is malicious. If it is malicious, then we refer to it as corrupt, otherwise our LRU is transmitting a normal message on the bus.

Figure 1 presents these choices as a tree of actions an actor can take. From it we can see a clear normal path, where an LRU transmits when the bus is free, transmits as themselves, and transmits normal data. All other paths are attacks which we refer to as corruption, spoofing, manipulation, and cancelling. It is the role of an IDS to differentiate the normal messages from the attacks.

Our four attacks are representative of how an attacker must interact with the serial data bus to execute their attack. This is true for both the strong and weak adversaries depicted in our adversary model. Strong adversaries have greater control over attack execution but the attacks are the same. Evaluating detection systems against high level attacker actions allows us to robustly test the security guarantees of IDSes. If an IDS is alerting on lower level attacks then an attacker can still change their behavior to execute an attack on the system. If an IDS is basing their detection on how an attacker must interact with the system then even a strong attacker cannot adapt to avoid detection.

Notably our attacks are all based on what an attacker must do on a message by message basis. There is likely a way to consider attacks in a much broader context based on their consequences rather than their prerequisites to action, but the message by message context fits better for evaluating IPSes. For the purposes of an IDS/IPS evaluation framework we define our four attacks as follows.

Spoofing is when one device/LRU pretends to be a different device/LRU. To spoof another LRU, an attacker must transmit a new message onto the bus. High level detection for spoofing usually involves the artifacts from an attacker transmitting deviating from the artifacts of a legitimate device transmitting.

Corruption is when an attacker takes over another LRU and sends malicious data. Corrupting an ECU implies transmitting as it, meaning messages become malicious without any deviation from normal transmission patterns. The typical assumption for corrupted attackers is that their transmission artifacts are indistinguishable from the victim LRU. Thus detection is based on the behavior of the overall system.

Manipulation is when an attacker modifies bits as they are transmitted on the line. To manipulate bits an attacker must override the physical layer signal. It is not a corruption attack, as they do not actually control the LRU. But they are capable of changing what the LRU says. Because of this detection is usually based around the protocol level side effects of flipping bits [6], or voltage based detection systems [8].

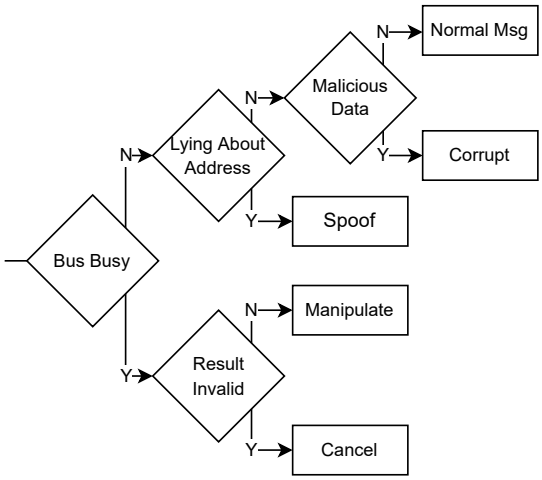

Figure 1: A tree of the options one can take when wanting to interact with a serial data bus network. Spoof, Manipulate, Cancel, and Corrupt are attack paths.

Cancelling is when an attacker interrupts some number of messages on the bus with the goal of stopping those messages from appearing on the system. Cancel attacks often make good prevention systems when used by a defender, as the principle of a cancel attack is the same principal used by MIPS. This attack is easy for a defender with standard receiver hardware to miss, as a standard receiver might drop any victim messages. Detection for this attack is normally protocol-specific, timing, or voltage based [18], [29], [6]. Protocol by the consequences of the protocol for cancelling a message, timing by the absence of a message, and voltage-based by the physical layer artifacts of mangling a signal.

5 EVALUATION FRAMEWORK

Our attacks give defenders have something to test existing IDSes and IPSes against. However, defenders still require a way to evaluate their security systems and a mechanism for comparing and contrasting different potential security systems. To this end, we propose an evaluation framework which measures the viability and efficacy of prevention and detection systems as they are faced with different threat models. Our framework highlights what scenarios a security system is designed to function in while shining a light on any limitations or assumptions designed into the system. The result is a necessary clarity, which allows defenders to understand the limitations of their own system and what new systems they need to detect the attacks and attack vectors facing them.

Our evaluation framework asks three questions:

- How viable is an IPS for my threat model?
- How suitable is my IDS to powering an IPS?
- How effective is my detection system against Spoofing, Corruption, Manipulation, and Cancelling attacks?

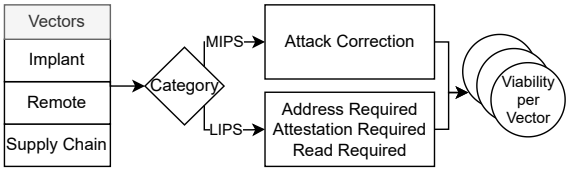

Figure 2: Depiction of our framework for IPS validity

These questions cover the three pillars of a security system, which are the prevention system, the detection system, and the integration of the two. The prevention system is graded against attack vectors to allow the defender to see what scenarios the prevention system functions in. Figure 2 depicts the attack vectors, and the criteria used to determine the viability of the IPS. Both the detection system, and the IDS/IPS integration, are graded against the four attacks described in Section 4.2 to identify any gaps in the detection system. Particularly any gaps that result in prevention not being viable for a given detection scheme, or a lack of detection coverage for specific attacks. Figure 3 depicts our grading process and the criteria used to evaluate IPS suitability and IDS efficacy. In it each attack is added to a trace from a serial data bus network, which then feeds the evaluation framework and IDS. The evaluation framework outputs our criteria which are fed into the appropriate suitability and efficacy buckets to produce a score per attack. Throughout this section we describe our criteria and how to calculate them, with a formal subsection with algorithms after the criteria descriptions.

An important note for our evaluation framework is that we assume an accurate IPS is better than allowing the attacker's message to be delivered. The rationale behind this is that the ramification of an attacker's payload, and more importantly the length of the attack, are difficult to rapidly predict compared to the known effects of an IPS triggering, assuming the vehicle maintains some level of functionality. A defender with the appropriate cyber threat intelligence may be able to make this determination that but it is outside the scope of this paper. The result is that our framework draws no distinction between attacks regardless of length or scale.

5.1 IPS Evaluation - Viability

For evaluating prevention systems we are less concerned with scoring the efficiency of the system and more concerned with the viability of the system. That is, what scenarios the IPS works in. The reason for this is that in the overall IDS/IPS system the IDS is the brains while the IPS is a triggered effect. An IPS could always be faster or more efficient, but we assume that if an IPS is being evaluated by this framework then it functions as a prevention system. The question is then what assumptions were made when designing the prevention system, and if there are any situations in the defenders threat model where the IPS either fails to work or can be circumvented. To that end, we do not produce a score for IPS evaluation. We report either true or false for if the IPS is viable for each of our attack vectors. These attacks vectors (implants, remote attackers, and supply chain attackers) represent a new device with custom hardware, an unchanged existing device, and existing device with modified hardware respectively. If any of the criteria are unviable, then the entire IPS is unviable against

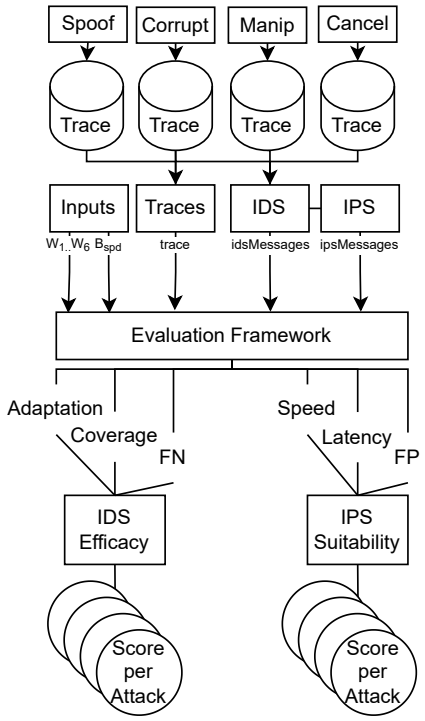

Figure 3: Depiction of how our evaluation framework is tested for IDS efficacy and IDS suitability to IPS (IPS suitability), where the evaluator provides weights, and bus speed. Attacks are added to each trace, which are then sent into the evaluation framework and the IDS, producing a score per attack. Algorithm 1 and Algorithm 2 describe how the six criteria are calculated.

that attack vector. This makes explicit how the assumptions of an IPS match up against the different threat models a defender may have. Notably our criteria are split between MIPS and LIPS as these prevention systems have different requirements for success.

Attack Correction (MIPS) is whether or not the attacker can unmangle the prevention signal. The pretense behind this is that an attacker could learn how a MIPS mangles a message, then counteract it. This requires additional hardware, and advanced capabilities, but would invalidate some implementations of MIPS if attack correction is in the defender's threat model. A MIPS is viable if it can counteract attack correction, or attack correction is not in the defender's threat model.

Address Required (LIPS) is the question of whether or not the LIPS needs to be able to talk to a specific address. In other words, if the LIPS functions by telling source address 2 to turn off, does it work against an attacker without an address? This criterion is most likely to affect implant attackers, as they do not need to claim an address on the bus. A LIPS is viable if the IPS does not rely on an address to function.

Attestation Required (LIPS) is the determination of whether the LIPS needs a way to verify which LRU sent the attack. This is not relevant to MIPS as MIPS is unconcerned with targeting

the attacker. A LIPS is viable if it does not require attestation to function, has attestation capabilities, or the attacker is unable to spoof traffic.

Read Required (LIPS) is the determination of whether the attacker needs to be able to receive data for the LIPS to function. This is not relevant to MIPS as MIPS invalidates attacker messages without interacting with the attacker. A LIPS is viable if it functions despite an attacker selectively reading the bus, or the threat model assumes the attacker must read the bus.

5.2 IDS Evaluation - Prevention Suitability

The criteria below focus on how suitable a chosen IDS is to powering a chosen IPS - be it MIPS or LIPS. After all, an IDS may be very good at detecting attacks, but be too slow or too prone to false positives to be viable as an IPS. This is similar to our previous concept of IPS viability. But while IPS viability is capabilities based and centered around threat models, IPS suitability is based around attacks, and the measurable differences in the way an IDS handles each of our four attacks. To this end, IPS suitability produces 4 scores which indicate how safe and viable an IDS/IPS combo is for spoofing, corruption, manipulation, and cancelling attacks. One way to use this metric is to selectively enable IPS for different attack types, ensuring you only risk running the IPS if you are confident in the speed and accuracy of the IDS.

Speed (MIPS) is representative of how much of a message is transmitted before the IDS makes a determination and the IPS fires. This is used to determine whether or not there is time to mangle a message for the purposes of intrusion prevention. We calculate speed as the percentage of the message taken by the IPS, plus the percent of the message taken by the IDS, divided by the total time taken to transmit the message. If this number is greater than one, then the message has completed and MIPS is impossible, so it receives a score of 0. Anything less than or equal to 1 receives a score of 100, as it does not matter how fast it is as long as message prevention is possible.

Latency (LIPS) is the LIPS side of the coin for speed. If the IDS is too slow then the entire security system could start to build a backlog of messages. The consequence is that our IDS will start to register attacks well after they are transmitted. A delayed LIPS could disable an attacker, but there is a significant distinction between a LIPS which stops an attacker after one attack, and a LIPS which disables an attacker several minutes or hours after the first attack.

Our scoring for latency reflects this difference by calculating how long the detection system takes relative to the time it takes to transmit a complete message. The score is the time taken by the detection system, divided by the transmission time multiplied by one plus the percentage of dead bus time. For example, if we assume a message takes 500 microseconds to transmit, with an average bus utilization of 70%, and our detection system takes 750 microseconds, then we get 1.15. This indicates that after we turn the vehicle off we would expect the IDS to keep processing messages for 15% of the time it collected data. Any number greater than 1 receives a score of 0. Any score less than 1 is a 100 as the total latency is irrelevant as long as the bus does not fall behind.

False Positives are whenever the IDS alerts on a benign message. A prevention system essentially acts as a DoS attack when presented with a false positive. It follows that installing an IPS with high false positives makes the system worse off than one without a security system. At least a system with no IPS has a chance that it will never be targeted by an attacker. Because of this logic, our evaluation framework aggressively weights false positives. The score is one minus the percentage of false positives multiplied by ten million. For example, a 0.00001% false positive rate receives a score of 0. We came to ten million based on our existing serial data bus datasets. In those datasets we would expect at least one million messages per hour of traffic. If a greater than single digit number of false positives occurs in an hour, then it is likely to result in alert fatigue at best and safety risks at worst. This problem is more pronounced for LIPS than for MIPS, as MIPS firing on accident is recoverable while LIPS firing on accident may require a system reboot.

5.3 IDS Evaluation - Detection Efficacy

The following criteria are used to test the efficacy of a chosen IDS. We do this against our four attacks of spoofing, corruption, manipulation, and cancelling as an IDS may specialize in one kind of attack and not cover others, or incidentally detect types of attacks it was not necessarily designed for. This is separate from IPS suitability in that IPS suitability focuses on speed and the risks of alerting on benign messages, while IDS efficacy is centered on how effective the IDS is at detecting attacks, without concern for timescales or any negative consequences. Our criteria are focused on how an IDS handles changing attacks, any gaps in coverage, and any attacks completely missed by the IDS. We choose these criteria as our evaluation framework is focused on how a defender can improve their detection system. An evaluator may use this metric to understand what attacks they cannot detect, while gaining an understanding of how an attacker might circumvent their IDS.

Adaptability is a metric indicating the flexibility of the detection system as an attacker modifies their attack. If the attacker can change either how they deliver their attack, or the payload of it, and remain undetected, then the detection system is too rigid for competent attackers. Increased adaptability often comes with the risk of decreased accuracy. We rank adaptability on a 4 point scale.

- **25** Static Rule, detects a specific payload
- **50** Dynamic Rule, based on bus behavior but relies on particular attack targets
- **75** Independent of the attacker behavior with exceptions
- **100** Independent of attacker behavior

Coverage is calculated as the percentage of traffic the intrusion detection system is capable of evaluating. Take an IDS which detects spoofing attacks by looking for anomalies in timing intervals between periodic messages. In this case coverage would be the percentage of traffic in which periodic messages occur. Coverage is differentiated from adaptation by it being concerned with the types of messages an attack uses. Adaptation is concerned with how the attack is delivered. For example, aperiodic messages are types of messages and so affect coverage, while an attacker changing their transmission scheme, or the payload, would be considered adaptation. For categorical evaluations of an IDS, listing all message IDs not covered by that IDS is a useful exercise, as it indicates where static rules or future detection systems may be useful.

Algorithm 1 IPS Suitability

$T_{\text{transmitting}}, FP, i \leftarrow 0$
$fastEnough \leftarrow$ True
// Iterate through Trace/IDS Lists
while $i <$ trace.len **do**
  // Measure Speed and FPs for IDS
  if idsMessages$[i]$.$result$ = True **then**
    // Calculate IDS (T_d) and IPS (T_p) Time
    $T_d \leftarrow$ idsMessages$[i]$.$time$ − trace$[i]$.$time$
    $T_p \leftarrow$ ipsTimes$[i] - T_d$
    // Calculate total msg transmit time (T_t)
    $T_t \leftarrow B_{spd} *$ trace$[i]$.$numBits$
    // Get % of msg taken by IDS/IPS
    Speed $\leftarrow (T_d + T_p)/T_t$
    if Speed > 1 **then** $fastEnough \leftarrow False$
    // Check for False Positives
    if trace$[i]$.attack = $False$ **then**
      $FP \leftarrow FP + 1$
  // Track total transmitting time
  $T_{\text{transmitting}} \leftarrow T_{\text{transmitting}} + T_t$
  $i \leftarrow i + 1$
// Fail if IDS/IPS longer than message
if $fastEnough$ = True **then**
  $S_S \leftarrow 100$
else
  $S_S \leftarrow 0$
// Calculate Bus Utilization (B_{util})
$B_{util} \leftarrow T_{\text{transmitting}}$/totalTraceTime
// Calculate how far IDS falls behind
$Latency \leftarrow T_d * (1 + (1 - B_{util}))/avgTransmitTime$
// Fail if Latency is > one message
if $Latency \leq 1$ **then**
  $S_L \leftarrow 100$
else
  $S_L \leftarrow 0$
// Calculate False Positives Score
$S_{FP} \leftarrow 1 - (FP/\text{trace}.len) * 10000000$
if Category = MIPS **then**
  $S_{\text{Suitability}} \leftarrow (w_1 * S_S) + (w_3 * S_{FP})/2$
else
  $S_{\text{Suitability}} \leftarrow (w_2 * S_L) + (w_3 * S_{FP})/2$

False Negatives occur whenever an IDS fails to alert on an attacker message. This is a measure of how effective the IDS is. We expect that many detection systems will have high false negatives for some attack types, as different detection techniques are better for different kinds of attacks. The false negatives score is calculated as one minus the percentage of false negatives.

5.4 Calculating a Score

After the evaluation process the evaluator is left with eight scores, along with three pass/fails. The three pass/fails are based on IPS viability for each attack vector. If any of the IPS viability metrics fail, then the IPS is deemed unviable for that attack vector. A summary of the previously mentioned criteria calculation methods can be

Algorithm 2 IDS Efficacy

$S_A \leftarrow$ See Step Function
IdSet, alertedSet $\leftarrow$ []
$FN, i \leftarrow 0$
// Get unique IDs of Trace and Attacks
while $i <$ trace.len **do**
  IdSet.add(trace$[i]$.id)
  if $idsMessages[i].result = True$ **then**
    alertedSet.add(idsMessages$[i]$.id)
    // Check for False Negatives
  else if trace$[i]$.$attack$ = True **then**
    $FN \leftarrow FN + 1$
  $i \leftarrow i + 1$
// Calculate Coverage of alerted IDs (S_C)
$S_C \leftarrow$ alertedSet.len/idSet.len
// Calculate Percent False Negatives (S_{FN})
$S_{FN} \leftarrow 1 - (FN/\text{trace.len})$
// Calculate Final Efficacy Score
$S_{\text{Efficacy}} \leftarrow (w_4 * S_A) + (w_5 * S_C) + (w_6 * S_{FN})/3$

seen in Algorithms 1 and 2. These algorithms require the input of the bus speed, weights, the original trace, the trigger time for the IPS and IDS, and the IDS's verdict on the message. IPS suitability creates 4 scores, one each for spoofing, corruption, manipulation, and cancelling attacks. This score is computed by using Algorithm 1, and calculating IPS suitability. If the desired outcome is a MIPS, then speed and false positives are averaged together to compute a score. Otherwise latency and false positives are averaged. The components of each of our scores are equally weighted. IDS efficacy works the same way as IPS suitability with 4 scores across our four attacks. The process to calculate the components of the scores is described in Algorithm 2. Once again the score is equal to the average of all three components, weighted equally.

5.5 Weights

Evaluator defined weights are our mechanism for accounting for different threat models. As we stated in our introduction, all security tools inherently come at some cost. And the evaluator is best positioned to adjust the weights of the evaluation framework to reflect how they see that cost. By default the evaluation framework assumes that all of the criteria are equally important, and thus equally weighted. But there are scenarios where some criteria are more important than others. Let us imagine an example where the most important thing to a defender is stopping known attacks. The defender has excellent threat intelligence and knows roughly what an attack is going to look like. In this scenario false negatives is the most important criterion for IDS efficacy, and adaptation is inconsequential if false negatives are low. Meanwhile coverage is only important for the attacks they expect to see (which are presumably the ones in their testing dataset). For IPS suitability, stopping attacks requires MIPS. Between speed and false positives, speed is most important for ensuring known attacks are prevented. Especially if preventing a few benign messages does not impact the safety of the vehicle. After applying weights the evaluator has a framework that works for their specific situation.

6 ATTACK TESTING

Our evaluation framework makes the assumption that the evaluator has the capacity to test at minimum spoofing, corruption, manipulation and cancelling attacks against an IDS. To enable this we provide an open source tool which randomly adds our attacks to datasets [26]. Our tool is generic and can support any protocol for our 4 attacks. Emulating these attacks in software ensures we can easily test all of our attack types without complicated hardware setups. While spoofing attacks are simple enough to support in a hardware in the loop setup, corruption attacks require loading 'malicious' software onto an LRU, and manipulation and cancelling attacks require custom hardware to drive the bus. Attempting to emulate these attacks, particularly corruption attacks, outside of software often leads to the attacker using an implant and then performing a spoofing attack by the nature of using an implant. This results in extremely limited attack testing. By testing the same attack payload with corruption, spoofing, and manipulation attacks the defender can see gaps in coverage for each attack.

The entire tool works by processing each message in a dataset and randomly triggering on messages. Spoofing attacks copy the triggered on message and changes to timestamp to a random time in the future. Corruption attacks randomize the data of the triggered message in-place. Manipulation attacks flip random bits in the triggered message. Cancelling attacks are done by randomly removing messages from the dataset. Notably our injected attacks do not modify the rest of the dataset to add the ramifications of the attack, as this is irrelevant to testing a prevention system. An IDS/IPS needs to be able to detect attacks based on a single message to effectively prevent attacks.

7 CASE STUDIES

This section provides case studies for two different protocols: CAN and MIL-STD-1553. These protocols are both serial data bus protocols but with vastly different architectures. Testing multiple architectures gives us confidence in the broad applicability of our framework to multiple serial data bus network protocols. For each protocol we will choose an IPS technology and an IDS technology then evaluate them against our framework, filling in protocol-specific information and limitations as we go along. We will discuss how to test each step in detail rather than re-implement four technologies ourselves. Our framework assumes the defender has a threat model in mind. For these case studies we assume all attack vectors and attacks are viable. We assume remote attackers are weak attackers and implants and supply chains default to being strong attackers. Table 1 provides a list of scores for other IDSes and IPSes. Some assumptions were made to produce scores for detection systems with different datasets and missing data. See the appendix for an explanation of how these scores were produced.

7.1 CAN

The CAN bus is perhaps the best studied of all serial data bus networks, as it is used in consumer automobiles and 'car hacking' has been a popular topic in academia for the last decade. CAN works using a differential pair signal between two wires. The LRUs connected to the bus regularly transmit messages. The easiest mechanism for doing MIPS on CAN is actually the one that enables LIPS. In brief, the CAN bus has an error mechanism such that when an LRU observes a different bit on the bus than what it sends, it transmits an error frame. This error frame informs the rest of the bus to drop the message, and increments an internal counter by 8. When it transmits an error free message, that LRU decreases the same counter by 1. If that internal counter ever reaches 255, then the LRU enters the 'bus-off' state, where its transceiver is turned off [6]. Meaning that if the MIPS triggers enough times on a CAN bus, then it effectively becomes a LIPS. A side effect of this error process is that manipulation attacks are exceedingly difficult on CAN. Manipulating any individual bit results in the message being cancelled unless the attacker commits to driving the bus and overpowering any error frames so the rest of the bus thinks the victim is a legitimate message.

For our IPS we will be using CANstomper [13] which flips one of the final 11 bits of a message to get the rest of the bus to drop it. Let us go through IPS validity. First we identify that CANstomper is a MIPS, as it cancels messages as it goes across the bus, with the option to be a LIPS across multiple attacks. We will evaluate the validity for both. For MIPS we assume attack correction is not possible for implants, remote attackers, or supply chain attackers, as a 1 to 0 transition is simple on the CAN bus and so CANstomper can reasonably randomize the strength of their attack. LIPS is more complicated, but CANstomper is well suited to it. Because it cancels a message as it goes on the bus, there is no need to know an address and no need to determine which device is transmitting (attestation required). This is true for all of our attack vectors. Read required is more complicated as the LIPS relies on the the attacker reading the bus, realizing it made an error, and incrementing an error counter such that it eventually turns off. Notably CAN devices have a way to reset their own error counters, meaning the attacker would also need to choose to not do that. From this we can conclude that LIPS is effective against weak attackers but not strong attackers. In our threat model this means our IPS is invalid against implants and supply chain attacks but valid against weak remote attacks.

For our detection system we select Cho's clock skew fingerprinting solution [7]. Let us start with the suitability of clock skew fingerprinting to becoming a prevention system. We evaluate this for spoofing, corruption, manipulation, and cancelling attacks. Cho's solution was built to detect what we refer to as spoofing, corruption, and cancelling attacks. Manipulation attacks are not detectable, so we will assume a score of zero.

The first criterion for IPS suitability is speed. Based on their implementation we have no reason to assume speed varies based on the kind of attack. Speed is not covered in the paper, but we know from the algorithm presented in the paper that the IDS has enough data to analyze the CAN packet after 29 bits (116 microseconds). We know CANstomper can trigger in the last 11 bits of a CAN packet, meaning if Cho's IDS takes less than 340 microseconds then it passes this test. Assuming competent hardware, we can reasonably conclude a score of 100 for speed. False positives are at 0% for spoofing attacks, 0.055% for corruption attacks, and 0% for cancelling attacks. This leads to a score of 100, 0, and 100 for these attacks respectively. The IPS suitability scores for our 4 attacks are 100, 50, 0, 100. From this a defender can conclude that IPS is unsafe for corruption and cancelling attacks in their threat model.

Table 1: Evaluation framework applied to IDS/IPS combinations. Assumes a strong implant and supply attacker, and weak remote attacker. '?' indicates the system was not designed for those attacks but would likely detect them.

	IPS Viability			IPS Suitability				IDS Efficacy			
IDS / IPS (MIPS/LIPS)	Implant	Remote	Supply	Spoof	Corrupt	Manip	Cancel	Spoof	Corrupt	Manip	Cancel
[29] / [31] (LIPS)	-	✓	-	100	0	0	100	98	0	0	98
[7] / [13] (MIPS)	✓	✓	✓	100	50	0	100	89.7	88.7	0	89.7
[25] / [13] (MIPS)	✓	✓	✓	0	0	0	0	0	91.3	0	0
[16] / [13] (LIPS)	-	✓	-	100	0	100	100	99.99	0	?	?
[30] (Sequence) /1553 Mode Code (LIPS)	-	✓	-	100	0	0	100	99.8	0	0	100
[30] (Voltage) /1553 Mode Code (LIPS)	-	✓	-	50	0	0	0	99.7	0	0	0
[12] / [15] (MIPS)	-	✓	-	50	0	0	0	91.7	0	0	0

Now we will evaluate adaptation, coverage, and false negatives to calculate the efficacy of Cho's IDS. Adaptation we give a 75 for all attacks. Interval based solutions are not concerned with the payload of the attacker, making them resilient to specific attacks. However, clock-skew can be spoofed, making it weak to certain attackers [27]. Coverage is also the same for each of our attacks, as the IDS's algorithm detects each attack in the same way. For coverage we see that Cho's work does not cover aperiodic messages. Based on our datasets roughly 6% of CAN IDs are not periodic, meaning coverage is 94%. False negatives vary from attack to attack. Spoofing and cancelling attacks have 0 false negatives (100), while corruption attacks have 3% (97). The total score is the average of the three criteria, resulting in: 89.7, 88.7, 0, 89.7. The cumulative score is primarily useful for comparing IDSes to each other, while the component scores highlight the weaknesses of the given IDS. These scores demonstrate that their testing dataset needs expanding to improve coverage and the detection mechanism needs improvement for corruption attacks.

7.2 MIL-STD-1553 and Combining IDSes

MIL-STD-1553 is a protocol primarily used by military aircraft and weapon systems. It makes an interesting case study because it uses a bus controller to dictate all communication and includes built in LIPS techniques. A message called a 'mode code' can inform any LRU to disable itself; the premise being that if an LRU is malfunctioning it can be told to stop transmitting until either told to resume or the system restarts. This makes LIPS easy for the MIL-STD-1553 bus, assuming we have some mechanism for source attestation. The shutdown mode code uses a 5 bit address field to target the victim LRU, but an attacker capable of spoofing this address field can trivially cause a LIPS to disable any LRU they choose. Additionally, LIPS only works if the attacker is taking over a legitimate device (e.g., a supply chain attack, or remote attack). Otherwise the attacker has no address that can be targeted by a mode code. In terms of counter-acting LIPS, there are mode codes which re-enable LRUs, but once the attacker is affected by the shutdown mode code, their only option is resetting themselves, or waiting for the vehicle to reset. From this description we can conclude the following for LIPS validity: addressing is required, attestation is required, and an attacker reading the bus is required. This implies our LIPS only works against weak attackers of remote and supply chain attacks, and does not work at all against implants due to the addressing issue. Crucially our choice of IDS hinges on the need for source attestation, otherwise a spoofing attack could weaponize our LIPS.

Based on the need for source attestation we choose Orly's IDS synthesis of voltage fingerprinting and message sequence detection [30]. It works by first determining if an LRU matches a known fingerprint, then determining the likelihood that the observed message ID would appear in that time slot. It is designed to function against spoofing and cancelling attacks. With corruption attacks not considered, as the detection system does not look at the data words in MIL-STD-1553. Manipulation attacks are possibly detected by the voltage fingerprinting, but the paper does not run this experiment. Spoofing and cancelling detection is handled by the message sequence IDS, and voltage fingerprinting detects only spoofing.

In terms of IPS suitability Orly does not specify the latency of either detection system, or the system as a whole. Given that neither technique relies on data words and only requires the first 16 bits of a message to function, we can likely assume both techniques would be fast enough to not extend into the next message. For false positives the score changes for each attack. Spoofing and cancelling attacks receive a 100, as no false positives occur for the sequence based detection system. The same is not true for the voltage fingerprinting IDS. At best the false positive rate is 0.4%. Even if only 200,000 messages appeared in an hour of traffic, we would expect 800 false positives an hour. An unacceptable number for the prevention system of a fighter aircraft. Meaning we have a score of 100, 0, 0, 100 for sequence based detection, and a score of 50, 0, 0, 0 for voltage based detection. From this we can conclude that sequence based detection is suitable to IPS, while the voltage fingerprinting solution is not.

IDS efficacy for our two detection systems perfectly demonstrates why one might want to have an IDS still exist, but not be a prevention system. The sequence-based IDS scores a 100 on adaption for spoofing attacks as the attacker cannot change how they transmit a message to avoid detection. Coverage is reduced as aperiodic messages are not included in the sequence based detection. While we do not know how many aperiodic messages occur in a MIL-STD-1553 dataset, we do know the message powering our LIPS is considered aperiodic. If an attacker can circumvent the entire IDS while executing a highly impactful (and simple) attack, then the IDS has serious issues. The final metric for spoofing is

false negatives, which scores a 100 as no attacks in their dataset are missed. Cancelling attacks score a 100 for adaptation as transmitting an aperiodic message is irrelevant to cancelling messages. Coverage is 100% for cancelling attacks as the next message is delayed. False negatives are also reported at 0, resulting in a score of 100, (100 - the percent of aperiodic messages), 100 for spoofing attacks, and 100,100,100 for cancelling attacks. Assuming 0.5% of aperiodic messages we get scores of 99.8,0,0,100 for spoofing, corruption, manipulation, and cancelling attacks respectively.

But now let us consider the voltage fingerprinting solution, their 'RT Authentication'. The detection is based on the electrical fingerprint of one device being different from another. Assuming decent hardware, an attacker cannot change their attack to avoid detection. The fingerprinting receives a 100 for spoofing attack adaptation. Coverage is also 100 as it works on any spoofing attack, not just periodic messages. False negative rates vary depending on the spoofing device, but at lowest are .9%, giving a score of 99.1. Resulting in a score of 99.7,0,0,0 for each of our attacks.

Comparing the scores of these two detection systems can reveal issues in either the testing dataset, or the IDS efficacy. We have zero false negatives for sequence based detection, but not full coverage for one of the detection systems. Indicating that our dataset has the appropriate attacks, but the sequence-based IDS does not process them. For the sake of becoming a prevention system this is good, as Orly indicates that aperiodic messages would result in false positives, which would then make the IPS suitability score of the sequence-based IDS low for spoofing attacks. For the sake of actually detecting attacks this would be a disastrous hole in coverage. Orly smartly pairs the fingerprinting technique with the periodic detection technique to cover all spoofing attacks. And because they are paired the overall system does not lose anything by having only the fingerprinting solution consider aperiodic messages. This flicking on and off of functionality both within and between attack categories maximizes what each IDS is good at without increasing the overall noise of the system. Scores for an individual IDS highlight what functionality is missing. Scores for multiple IDSes demonstrate how to configure each IDS to maximize the accuracy of the overall system and safely prevent attacks when one can.

The last example that needs covering is weights. Based on how our LIPS works, it is clear that some rare messages are particularly dangerous. It follows that coverage is the most important criterion for understanding how good an IDS is for MIL-STD-1553. The exact numbers are a bit vague as we lack Orly's implementation and datasets, but we can apply a higher weight to coverage such that missing aperiodic messages result in an extremely low overall score. It is important to note that evaluators do not necessarily need to stick completely to our criteria. If not detecting LRU disabling messages or not detecting weapon system messages immediately invalidates an IDS, then giving a score of 0 unless those conditions are met is perfectly fine. Better to do simple conditional checks for system specific issues than risk the 0.01% in a 99.99% coverage score being a protocol-specific attack that every attacker would try.

8 DISCUSSION

Comparing Scores - In our case studies we use individual scores to identify potential areas of improvement. Each criterion indicates where resources need to be focused. The other use case is comparing scores generated with the same dataset but from multiple security systems. Scores should only be compared within the same attack category. The goal is to maximize scores across attack categories by taking the highest scoring component from each IDS. Weights ensure that the defender's priorities are incorporated into the score, meaning we can trust aggregate scores to indicate which system is better for the defender. The only complication is how the criteria are calculated. For example, a 1% difference in coverage does not mean that IDSes have the same coverage except for that 1%. If some subset of messages are more important or more likely to be attacked then the system with lesser coverage may be better. The magnitude of the difference is helpful for maximizing detection with available resources, but more analysis into how each criterion is calculated is needed to draw meaningful conclusions from a scoring difference. These conclusions could be used to synthesize IDSes or create focused analytics for known gaps.

Use By System Owners and Red Teams - It is our opinion that the evaluation framework is better used as a relative score by maintainers and evaluators trying to select detection and prevention systems. Particularly in the increasingly frequent scenario of a system owner trying to determine if a security product actually does what they need. Of course, a maintainer is not interested in the exact language academics use. But it is important for them to understand what attacks and attack vectors they defend against. Take the example of an interval based detection approach [29]. This approach is designed to detect spoofing attacks and cancelling attacks. Spoofing detection is great for implants, but what about remote and supply chain attackers? If the attacker controls a victim device and transmits the messages it would normally transmit in the correct time slot, but with malicious data, then they detect nothing. After using our evaluation framework the maintainer knows the IDS scores highly against spoofing attacks, but is largely useless against corruption, and manipulation attacks. Now they know to procure another IDS solution which covers these attacks.

Extending the Evaluation Framework - While we believe our evaluation framework is high level enough to accommodate new attacks and new IDS/IPS techniques, an evaluator may wish to extend the framework to provide more details for their threat model. Beyond the weights covered in this paper they can also add a criterion. Adding a criterion only requires adding a weight parameter, as long as some basic rules are followed. For IDS efficacy and IPS suitability, the new criterion must produce a score from -100 to 100 and be testable based on an attack dataset. The criterion does not necessarily need to be quantitative, as shown by adapation, but qualitative scores are worse for setups which expect multiple evaluators who could interpret data differently. Qualitative criteria are particularly effective when they encourage the evaluator to further examine a critical aspect of the IDS/IPS. The more distinguished the qualitative options are, the better. Adding criteria to IPS validity is as simple as adding yes/no questions for MIPS/LIPS.

9 CONCLUSION

In this paper we propose an evaluation framework for serial data bus networks to help researchers and defenders understand the capabilities and limitations of their detection and prevention systems.

Our framework can be used to securely and efficiently develop new systems, or to evaluate existing technologies against specific threat models. We encourage the safe and intentional adoption of IPS technology by defining a more nuanced view of IPS. We examine IPS technology through the lens of attack vectors, and create new categories of IPS based on whether the defender wants to prevent individual messages or disable LRUs. Additionally, we evaluate IDS technology based on the high level attack behavior it can detect, and its suitability to powering an IPS. These insights allow a defender to know when an attacker can circumvent their IPS, what attacks their IDS does not cover, and what aspects of their IDS can be safely be connected to an IPS. The security and safety of serial data bus networks is crucial, and with the knowledge of our framework defenders can more confidently achieve it.

REFERENCES

[1] Omar Y. Al-Jarrah, Carsten Maple, Mehrdad Dianati, David Oxtoby, and Alex Mouzakitis. 2019. Intrusion Detection Systems for Intra-Vehicle Networks: A Review. *IEEE Access* 7 (2019), 21266–21289. https://doi.org/10.1109/ACCESS.2019.2894183

[2] Emad Aliwa, Omer Rana, Charith Perera, and Peter Burnap. 2021. Cyberattacks and Countermeasures for In-Vehicle Networks. *ACM Comput. Surv.* 54, 1, Article 21 (mar 2021), 37 pages. https://doi.org/10.1145/3431233

[3] Omid Avatefipour and Hafiz Malik. 2018. State-of-the-Art Survey on In-Vehicle Network Communication (CAN-Bus) Security and Vulnerabilities. *arXiv e-prints*, Article arXiv:1802.01725 (Feb 2018), arXiv:1802.01725 pages. arXiv:1802.01725 [cs.CR]

[4] Rohit Bhatia, Vireshwar Kumar, Khaled Serag, Z Berkay Celik, Mathias Payer, and Dongyan Xu. 2021. Evading Voltage-Based Intrusion Detection on Automotive CAN. *Ndss 2021* February (2021).

[5] Matthew Butler. 2017. An Intrusion Detection System for Heavy Duty Vehicle Networks. In *ICCWS*, Vol. 12. 399–405.

[6] K. Cho and K Shin. 2016. Error Handling of In-Vehicle Networks Makes Them Vulnerable. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. Association for Computing Machinery, New York, NY, USA, 1044–1055. https://doi.org/10.1145/2976749.2978302

[7] K. Cho and K Shin. 2016. Fingerprinting Electronic Control Units for Vehicle Intrusion Detection. In *Proceedings of the 25th USENIX Conference on Security Symposium* (Austin, TX, USA) *(SEC'16)*. USENIX Association, Berkeley, CA, USA, 911–927. http://dl.acm.org/citation.cfm?id=3241094.3241165

[8] W. Choi, K. Joo, H. J. Jo, M. C. Park, and D. H. Lee. 2018. VoltageIDS: Low-Level Communication Characteristics for Automotive Intrusion Detection System. *IEEE Transactions on Information Forensics and Security* 13, 8 (2018), 2114–2129.

[9] Jeremy Daily, David Nnaji, and Ben Ettlinger. 2021. Securing CAN Traffic on J1939 Networks. *NDSS Automotive and Autonomous Vehicle Security (AutoSec) Workshop* February (2021).

[10] embitel 2017. Automotive ECU: Journey from Mechanical to Electronics Based Control Units. https://www.embitel.com/blog/embedded-blog/automotive-control-units-development-innovations-mechanical-to-electronics

[11] Mahsa Foruhandeh, Yanmao Man, Ryan Gerdes, Ming Li, and Thidapat Chantem. 2019. Simple: Single-frame based physical layer identification for intrusion detection and prevention on in-vehicle networks. *PervasiveHealth: Pervasive Computing Technologies for Healthcare* (2019), 229–244. https://doi.org/10.1145/3359789.3359834

[12] Sebastien J.J. Genereux, Alvin K.H. Lai, Craig O. Fowles, Vincent R. Roberge, Guillaume P.M. Vigeant, and Jeremy R. Paquet. 2020. MAIDENS: MIL-STD-1553 Anomaly-Based Intrusion Detection System Using Time-Based Histogram Comparison. *IEEE Trans. Aerospace Electron. Systems* 56, 1 (2020), 276–284. https://doi.org/10.1109/TAES.2019.2914519

[13] Hristos Giannopoulos, Alexander M Wyglinski, and Joseph Chapman. 2017. Securing Vehicular Controller Area Networks: An Approach to Active Bus-Level Countermeasures. *IEEE Vehicular Technology Magazine* 12, 4 (2017), 60–68.

[14] Bogdan Groza, Stefan Murvay, Anthony van Herrewege, and Ingrid Verbauwhede. 2017. LiBrA-CAN: Lightweight broadcast authentication for controller area networks. *ACM Transactions on Embedded Computing Systems* 16, 3 (2017).

[15] Idaho Scientific. 2021. BusCop milSTD1553B cybersecurity.

[16] Marcel Kneib and Christopher Huth. 2018. Scission: Signal Characteristic-Based Sender Identification and Intrusion Detection in Automotive Networks. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (Toronto, Canada) *(CCS '18)*. Association for Computing Machinery, New York, NY, USA, 787–800. https://doi.org/10.1145/3243734.3243751

[17] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage. 2010. Experimental Security Analysis of a Modern Automobile. In *2010 IEEE Symposium on Security and Privacy*. 447–462. https://doi.org/10.1109/SP.2010.34

[18] Sekar Kulandaivel, Shalabh Jain, Jorge Guajardo, and Vyas Sekar. 2021. Cannon: Reliable and stealthy remote shutdown attacks via unaltered automotive microcontrollers. *2021 IEEE Symposium on Security and Privacy (SP)* (2021). https://doi.org/10.1109/sp40001.2021.00122

[19] Kyle Mizokami. 2021. NSA to Pentagon: Lock down your weapons before hackers get to them. https://www.popularmechanics.com/military/weapons/a37896509/nsa-pentagon-weapons-systems-cyberattack-risk/

[20] Thomas Morris, Rayford Vaughn, and Yoginder Dandass. 2012. A Retrofit Network Intrusion Detection System for MODBUS RTU and ASCII Industrial Control Systems. In *2012 45th Hawaii International Conference on System Sciences*. 2338–2345. https://doi.org/10.1109/HICSS.2012.78

[21] Subhojeet Mukherjee, Hossein Shirazi, Indrakshi Ray, Jeremy Daily, and Rose Gamble. 2016. Practical DoS Attacks on Embedded Networks in Commercial Vehicles. In *International Conference on Information Systems Security*, Vol. 10063. 23–42. https://doi.org/10.1007/978-3-319-49806-5_2

[22] P. Murvay and B. Groza. 2018. Security Shortcomings and Countermeasures for the SAE J1939 Commercial Vehicle Bus Protocol. *IEEE Transactions on Vehicular Technology* 67, 5 (May 2018), 4325–4339. https://doi.org/10.1109/TVT.2018.2795384

[23] Krzysztof Pawelec, Robert A Bridges, and Frank L Combs. 2017. Towards a CAN IDS based on a neural-network data field. arXiv. , 2–5 pages.

[24] O. Pfeiffer. 2017. *Implementing Scalable CAN Security with CANcrypt: Authentication and Encryption for CANopen, J1939 and Other Controller Area Network Or CAN FD Protocols*. Embedded Systems Academy Incorporated. https://books.google.co.uk/books?id=96qAAQAACAAJ

[25] Raul Quinonez, Jairo Giraldo, Luis Salazar, Erick Bauman, Alvaro Cardenas, and Zhiqiang Lin. 2020. SAVIOR: Securing Autonomous Vehicles with Robust Physical Invariants. In *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, 895–912. https://www.usenix.org/conference/usenixsecurity20/presentation/quinonez

[26] Redacted. 2022. Attack Testing Tool. https://github.com/matthewRekos/serialNightshade.

[27] Sang Uk Sagong, Xuhang Ying, Andrew Clark, Linda Bushnell, and Radha Poovendran. 2018. Cloaking the Clock: Emulating Clock Skew in Controller Area Networks. *Proceedings - 9th ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS 2018* (2018), 32–42. https://doi.org/10.1109/ICCPS.2018.00012 arXiv:1710.02692

[28] U S Senate. 2018. WEAPON SYSTEMS CYBERSECURITY DOD Just Beginning to Grapple with Scale of Vulnerabilities United States Government Accountability Office. October (2018).

[29] H. M. Song, H. R. Kim, and H. K. Kim. 2016. Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network. In *2016 International Conference on Information Networking (ICOIN)*. 63–68. https://doi.org/10.1109/ICOIN.2016.7427089

[30] Orly Stan, Adi Cohen, Yuval Elovici, and Asaf Shabtai. 2019. Intrusion Detection System for the MIL-STD-1553 Communication Bus. *IEEE Trans. Aerospace Electron. Systems* 9251, c (2019), 1–17. https://doi.org/10.1109/TAES.2019.2961824

[31] Masaru Takada, Yuki Osada, and Masakatu Morii. 2019. Counter attack against the bus-off attack on CAN. *Proceedings - 2019 14th Asia Joint Conference on Information Security, AsiaJCIS 2019* (2019), 96–102. https://doi.org/10.1109/AsiaJCIS.2019.00004

[32] Joseph Trevithick. 2020. Army hires company to develop cyber defenses for its Strykers after the vehicles got hacked. https://www.thedrive.com/the-warzone/37684/army-hires-company-to-develop-cyber-defenses-for-its-strykers-after-they-were-hacked

[33] Joseph Trevithick. 2020. U-2 Spy Plane Got New Target Recognition Capabilities In First Ever In Flight Software Updates. https://www.thedrive.com/the-warzone/37131/u-2-spy-plane-got-new-target-recognition-capabilities-in-first-ever-in-flight-software-update

[34] A Wasicek, Mert D Pesé, André Weimerskirch, Yelizaveta Burakova, and Karan Singh. 2017. Context-aware intrusion detection in automotive control system. *Escar* (2017).

[35] Clinton Young, Joseph Zambreno, Habeeb Olufowobi, and Gedare Bloom. 2019. Survey of automotive controller area network intrusion detection systems. *IEEE Design and Test* 36, 6 (2019), 48–55. https://doi.org/10.1109/MDAT.2019.2899062

[36] Aiguo Zhou, Zhenyu Li, and Yong Shen. 2019. Anomaly detection of CAN bus messages using a deep neural network for autonomous vehicles. *Applied Sciences (Switzerland)* 9, 15 (2019), 1–12. https://doi.org/10.3390/app9153174

A APPENDIX

In this appendix we expand the scoring for Table 1. This can be seen in Table 2. All of these scores should be taken with a healthy

Table 2: Evaluation framework applied to IDS/IPS combinations, where this table breaks down the criteria for IPS suitability and IDS efficacy for different attacks.

		IPS Suitability			IDS Efficacy		
IDS / IPS (MIPS/LIPS)	Attack	Speed	Latency	FP	Adapt	Coverage	FN
Song et. al [29] / [31] (LIPS)	Spoof	-	100	100	100	94	100
Song et. al [29] / [31] (LIPS)	Cancel	-	100	100	100	94	100
Cho and Shin [7] / [13] (MIPS)	Spoof	100	-	100	75	94	100
Cho and Shin [7] / [13] (MIPS)	Corrupt	100	-	0	75	94	97
Cho and Shin [7] / [13] (MIPS)	Cancel	n.a	-	n.a	75	94	100
Quinonez et. al [25] / [13] (MIPS)	Corrupt	0	-	0	75	100	99
Kneib and Huth [16] / [13] (LIPS)	Spoof	0	100	100	100	100	99.99
Kneib and Huth [16] / [13] (LIPS)	Manip	0	100	?	100	100	?
Kneib and Huth [16] / [13] (LIPS)	Cancel	0	100	?	100	100	?
Orly (Sequence) [30] /1553 Mode Code (LIPS)	Spoof	-	100	100	100	99.5	100
Orly (Sequence) [30] /1553 Mode Code (LIPS)	Cancel	-	100	100	100	100	100
Orly (Fingerprint) [30] /1553 Mode Code (LIPS)	Spoof	-	100	0	100	100	99.1
Orly (Fingerprint) [30] /1553 Mode Code (LIPS)	Manip	-	100	?	100	100	?
Generuex [12] / [15] (MIPS)	Spoof	0	-	100	75	100	100

degree of skepticism, as we cannot perform dynamic testing, each paper uses unpublished attack datasets, the accuracy is based on their own testing, and we extrapolate attack performance, speed, and coverage based on details in the paper. We generally favor more generous interpretations of existing research when we lacked concrete details. We would expect fewer 100s on an independently created attack dataset applied to each IDS.

Song [29] is evaluated with a dataset with no aperiodic messages, and only tests transmitting fast messages. We assume cancelling attacks are incidentally detected by the interval deviation caused by a message not appearing. Song does not specify the algorithm watches for messages which appear too slowly, but its a small enough leap for us to consider cancelling attacks as having the same security guarantees as spoofing attacks. In terms of scoring, the basis of relying on intervals makes it similar to Cho [7], but without the additional fingerprinting and corruption attacks. And adaptation is 100 instead of 75 as pure intervals cannot be faked like Cho's clock skews. Our definition of corruption attacks is different from Cho's masquerade attacks, but the principle is similar enough that we put it in the corruption category. We define corruption as a more advanced version of masquerade attacks, where only one attacker is necessary. We would expect Cho to out perform Song in terms of accuracy with a different dataset, as the fingerprinting provided by Cho is a boon for detecting spoofing attacks.

Quinonez [25] only covers corruption attacks, as it is specifically watching for deviations in the data field, without any concern for where the data is coming from. We know this by it detecting the flood of sensor data by the deviations in the data, rather than the sudden influx of inputs. In terms of scores, adaptability being 75 is in reference to it not detecting data fields with small fluctuations. Coverage is technically limited to the 13 sensors they base their invariance model on, though this is seemingly the only sensors on their small vehicle. We would expect less coverage on an actual automobile with more computers.

Kneib and Huth [16] detect spoofing attacks via electronic fingerprinting, and have explicit limitations that show they cannot detect corruption attacks. A latency score of 100 may be somewhat misleading as the system uses a suspicion counter to detect devices which were not present during training. Meaning it may take tens of messages to actually generate an alert. It is unclear how high the false positives would be without this suspicion system. We believe this research approach would also detect manipulation and cancelling attacks. Manipulation attacks generally cause spikes in voltages as the bus is overridden, this should cause a fingerprint mismatch. Cancelling attacks can be done via a bit flip in CAN, and result in a bus-off attack that is referenced in the paper. That said, neither of these attacks are explicitly tested, so we are unaware of how well their IDS performs.

Generux [12] has an adaptability of 75 because the response time of messages is a high level detection metric, but it is also fake-able by any attacker capable of pre-processing their attack. In terms of speed we found detection time was greater than the length of a message.

Our table does not include Wasicek [34] and Choi [8], among others from Section 2, as they did not provide enough data on the performance of their IDS to make meaningful score estimations. This is not to say these detection systems do not work, merely that our detection system expects certain inputs and their papers do not happen to provide them.