

ORIGINAL RESEARCH

Open access

Securing Smart Grid Substations: A Trust Model Approach for Cyber Defense

Sara Ben Youssef^{1*}, Amal Trabelsi¹

Abstract

The Smart Grid is an advanced electricity network deeply intertwined with cyber technologies that oversees power generation, long-distance transmission, and final delivery to end users. At its core lies the substation. However, embedding digital systems within substations has dramatically widened the range of potential vulnerabilities. Advanced adversaries, such as those behind the PipeDream APT, deliberately exploit several industrial protocols, including Modbus, DNP3, and IEC61850. Substations face strict operational constraints that complicate the adoption of conventional protective mechanisms, such as data encryption or network intrusion detection systems. To address these difficulties, we present an integrated, trust-centric architecture that significantly strengthens substation defenses. The proposed architecture integrates three main components: a trust evaluation model, a dynamic risk assessment model, and a trust portability model. The trust evaluation model can uncover protocol-specific malicious activities targeting Intelligent Electronic Devices and SCADA human-machine interfaces. The dynamic risk assessment model continuously determines the substation's current security risk exposure. The trust portability model analyzes whether a device, together with its accumulated trust profile, can be effectively migrated and reused in another substation environment. For a realistic evaluation, we constructed a containerized (Docker) substation replica organized around a multi-agent framework and equipped with a live dashboard modeled after modern Security Operations Centers. Effectiveness was measured by simulating a broad spectrum of adversary behaviors aligned with the MITRE ICS ATT&CK matrix. Experimental results indicate robust resistance to various attack vectors, reliable detection of malicious actions, and appropriate incentivization of well-behaved, trustworthy components.

Keywords Smart grid, Trust, Substation, Risk posture, Cybersecurity, Modbus

*Correspondence:

Sara Ben Youssef
sara.benyoussef@utunis.tn

¹ Department of Electrical Energy Analytics, Faculty of Engineering, University of Tunis, Tunis, Tunisia

Introduction

Within the Smart Grid ecosystem, the substation is often described as its central nervous system due to its indispensable role in stepping voltage up or down during electricity production, bulk transport, and local delivery. In parallel with the wider Smart Grid, modern substations have undergone extensive digitalization through pervasive cyber components, evolving into true cyber-physical entities [1]. Because any prolonged interruption—especially one originating at a substation—can disrupt electricity supply to vast numbers of commercial enterprises, factories, and private residences, the Smart Grid, and substations in particular, are classified as vital critical infrastructure. The overall grid relies on an extensive network of interconnected substations to ensure that reliable power reaches every customer.

Supervisory Control and Data Acquisition (SCADA) systems form the backbone that enables this highly interconnected cyber-physical grid, providing real-time visibility, remote command capability, and automated regulation. Typical SCADA

deployments incorporate diverse hardware, including Ethernet switches, purpose-built controllers, and standard servers/workstations.

A specialized subset of SCADA, known as the Substation Automation System (SAS), manages automated functions locally within each substation. As depicted in **Figure 1**, SAS is conventionally structured in three tiers: station level, bay level, and process level. Among the most critical field devices is the circuit breaker (CB), which halts current flow; it is directly commanded by an Intelligent Electronic Device (IED).

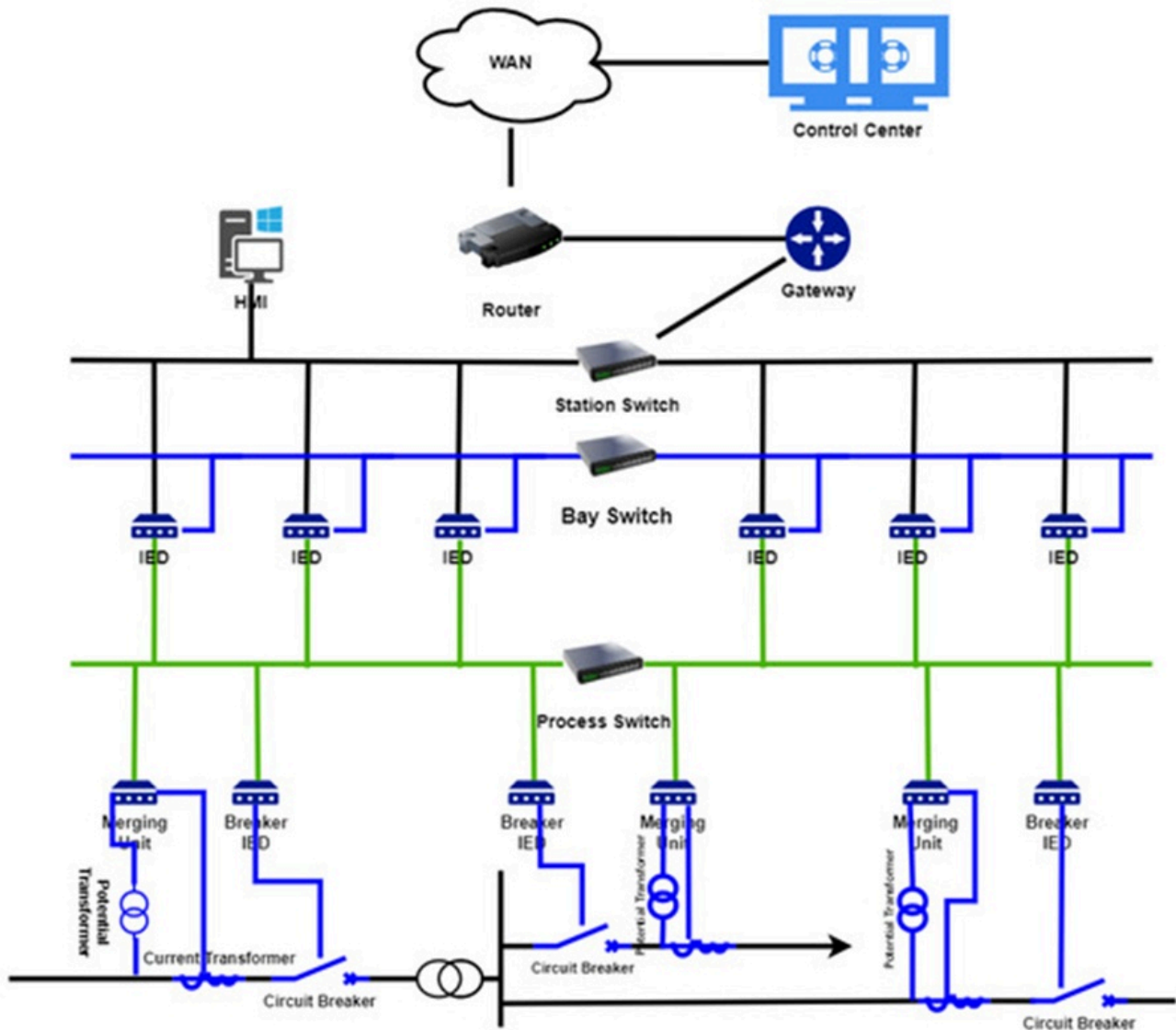


Figure 1. Substation automation system.

Whenever the IED detects an abnormal condition or fault, it trips the associated CB to isolate the affected section and prevent further damage to equipment. Although essential for asset protection, such actions inevitably cause loss of supply to all downstream customers fed by that substation. For this reason, attackers frequently prioritize compromising the IED when attempting to disable or destabilize a substation. The progressive incorporation of networked cyber elements into power infrastructure has coincided with a marked escalation in the frequency and sophistication of targeted attacks—from the well-known Stuxnet campaign [2] to PipeDream (also called Incontroller) [3] and CosmicEnergy [4].

Incidents such as Stuxnet, PipeDream, CosmicEnergy [4], and related operations fall under the category of advanced persistent threats (APTs). An APT describes a carefully planned, resource-intensive intrusion campaign whose primary objectives are to gain and maintain stealthy, long-term access to a target environment, most often for espionage, data exfiltration, and/or eventual sabotage. Compared with opportunistic or commodity malware, APTs exhibit far greater complexity and evasion capabilities: they chain multiple custom techniques, focus narrowly on high-value targets (often specific industries or government entities), and are engineered to bypass or disable standard security controls. These operations are typically sponsored by nation-states or exceptionally capable criminal syndicates possessing substantial funding, expertise, and patience.

In the case of Pipedream, the malware leverages its PLC-focused modules to manipulate targeted devices. It can discover additional devices on the network, attempt password guesses, terminate active connections, and force devices into nonfunctional states. When it encounters a device that lacks exploitable vulnerabilities, the malware's architecture allows it to intercept and repurpose the device's legitimate operations by issuing authentic commands over the protocols the device supports. This is achieved through a variety of protocols, including Modbus, CODESYS, FINS, and OPC-UA. Analysis shows that Pipedream can execute 38% of the known ICS attack techniques and 83% of the ICS attack tactics outlined in the MITRE Industrial Control Systems (ICS) Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) Framework. The MITRE ICS ATT&CK framework serves as an extensive repository of tactics, techniques, and procedures used by adversaries in cyberattacks, helping cybersecurity experts improve threat identification and mitigation.

Substations present a unique security dilemma: once the control center has been compromised, gauging the precise extent of damage to the substation becomes extremely difficult. As depicted in **Figure 1**, the aforementioned attacks typically began by infiltrating the control center before manipulating IEDs to disrupt the power supply. Although unrelated to the Smart Grid, the Colonial Pipeline incident [5] demonstrates how operators may proactively disable an entire network as a safeguard when the control center is breached. A key similarity between the pipeline shutdown and substation scenarios is their reliance on operational technology (OT) environments.

Deploying intrusion detection systems at the station, bay, or process levels within the substation can degrade network performance owing to stringent timing requirements [6, 7]. Certain proposed defenses rely on cutting-edge cryptographic methods that demand modern hardware; however, surveys indicate that, on average, 44% of relays in utility systems have been in service for over 15 years [8, 9]. Solutions that are independent of hardware upgrades remain vulnerable if adversaries gain control of the central station. Consequently, there is a pressing need for protective measures compatible with the existing substation infrastructure. We propose that incorporating the principle of trust offers a viable path forward.

We introduce a trust-oriented framework intended to augment existing security controls and counteract the risks posed by APTs, particularly during their post-breach phases. A core element of this framework empowers IEDs to function as the substation's final defensive barrier, granting operators adequate time to contain and lessen any adverse effects. The key contributions of this work are as follows:

- **Trust-Based Framework:** An all-encompassing architecture crafted to bolster substation protection through the integration of trust mechanisms.
- **Trust Model Component:** A trust evaluation mechanism embedded in the framework, tasked with identifying protocol-specific attacks aimed at IEDs and SCADA human-machine interface (HMI) systems.
- **Risk Posture Model Component:** A framework element that evaluates and updates the substation's overall risk exposure based on identified threats.
- **Trust Transferability Model Component:** A framework module that assesses the viability of migrating a device—along with its trust attributes—to an alternative substation and oversees the seamless incorporation process.

- **Docker-Based Substation Testbed:** A realistic simulation platform built with Docker containers, featuring a multi-agent system that replicates the substation's device interactions. It includes a dashboard inspired by Security Operations Center (SOC) practices to provide live monitoring of the substation and connected devices.
- **Attack Scenario Evaluations:** Rigorous assessment of the framework via emulated attack paths, encompassing external intrusions, internal compromises from breached SCADA HMIs, and internal threats stemming from compromised standard IEDs.
- **Publicly Available Dataset:** An openly accessible collection of network captures from our multi-agent system (MAS) testbed, hosted on the CIC website (<https://www.unb.ca/cic/datasets/modbus-2023.html> (accessed on 31 August 2023)).

The paper is organized to explore multiple dimensions of trust in depth. Provides an in-depth examination of the trust concept, including its contemporary advancements and the underlying research motivations. Subsequent sections establish foundational context, covering multi-agent systems and the Modbus protocol.

Figure 2 offers a graphical overview of the paper's layout.

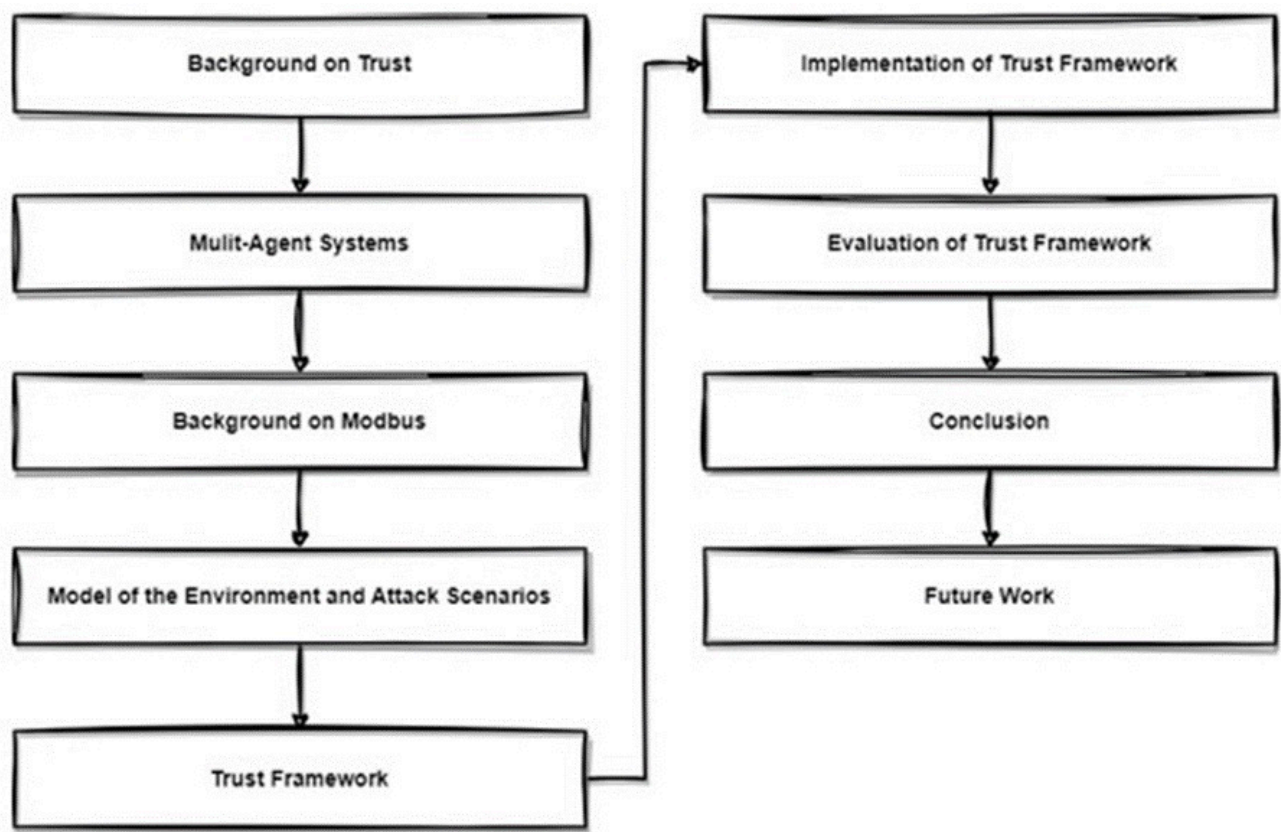


Figure 2. Organization of the paper.

Background on Trust

Trust admits multiple interpretations across disciplines. In social sciences, one view frames it as an individual's conviction regarding the conduct of a particular entity [10]. An alternative perspective treats it as the probability that an agent will perform a designated action [11]. Throughout this work, the trusting party is referred to as the agent or trustor, while the

trusted party is designated the subject or trustee. As articulated by Rousseau *et al.* [12], trust arises when the trustor willingly exposes themselves to risk, grounded in favorable assumptions about the trustee's motives or actions.

This section reviews foundational aspects and conceptual formulations relevant to three trust-related concepts tailored to substations: trust itself, risk posture, trust transitivity, and trust transferability.

Trust

The level of trust (T_{ij}) that one agent (a_i) assigns to another subject (a_j) during a particular time interval (t) can be expressed mathematically as a tuple, as presented in Eq. 1 [13]. Here, r_{ij} inddenotes the level of risk perceived by agent a_i when relying on subject a_j , α_{ij} captures the characteristics of the interaction or exchange between them, k_{ij} reflects the insights gained from their exchanges over the period t , and T'_{ij} denotes the trust value carried over from earlier interactions. This prior trust value corresponds to the agent's confidence score from a previous engagement. In practice, the process often begins by initializing the prior trust to a predefined baseline, followed by iterative refinements through an update mechanism. Existing studies describe diverse strategies for revising trust scores, such as using temporal sliding windows, directly updating the most recent trust value, factoring in attributes specific to the current transaction, or blending several of these approaches. It should be noted that these approaches do not represent an all-inclusive set.

$$T_{ij} = f(a_i, a_j, \alpha_{ij}, r_{ij}, k_{ij}^t, T'_{ij}) \quad (1)$$

Trust values are generally modeled as continuous scalars within a defined interval, most commonly $-1 \leq T \leq 1$ or $0 \leq T \leq 1$. In such scales, a value of 1 signifies full confidence, -1 indicates total distrust, and 0 denotes neutrality or absence of trust.

Trust transitivity and trust transferability

As described in the literature, trust transitivity arises when a trustor agent i extends confidence to an unfamiliar subject j because j is associated with a trusted third party k , whom i already trusts [12]. Eq.2 defines the conditions under which transitivity is permitted, while Eq. 3 specifies scenarios where it is rejected.

$$T_{ik} \neq T_{jk} \text{ where } 1 - T_{ij} \approx 0, 1 - T_{jk} \approx 0 \quad (2)$$

$$T_{ik} \sim T_{jk} \text{ where } 1 - T_{ij} \approx 0, 1 - T_{jk} \approx 0 \quad (3)$$

Trust transferability refers to the ability to relocate both an agent and its established trust attributes into a new, trusted setting. For an agent i possessing trust properties T_i within a given context c , transferability T_f can be formulated as the output of a binary decision function that determines whether to migrate the device's trust-related features, as expressed in Eq. 4.

$$T_f = f(T_i, c) \quad (4)$$

State of the art

Research exploring trust mechanisms specifically applied to substations remains scarce [14]. Earlier efforts in the Smart Grid domain introduced reputation systems to reduce the impact of malfunctioning agents in substation backup protection arrangements. Nonetheless, these approaches struggled to distinguish between intentional malicious actions and benign faults accurately [15, 16]. The studies by Borowski *et al.* [15] and Fadul *et al.* [16] were among the earliest initiatives to

embed trust concepts into multi-agent system architectures for substations—aligning closely with the focus of the present work.

Qureshi *et al.* [17] developed a trust assessment approach relying on metrics such as packet loss rates, timing discrepancies in packet transmission and reception, and packet volumes to flag rogue devices. However, this method falls short at detecting carefully crafted malicious packets that maintain proper timing, embed harmful content, or present fabricated information—characteristics seen in incidents such as Stuxnet.

Wang *et al.* [18] proposed integrating a trusted server into a Purdue-model Smart Grid setup. Their design featured a TPM-enabled trust engine embedded in devices to gather and preserve transaction records. A major limitation lies in the resource demands of such an engine, rendering it impractical for deployment on legacy hardware lacking sufficient capacity or upgrade potential. Additionally, their model overlooked the substation's broader risk profile when detecting untrustworthy node behavior, and the reported minimum response latency was 20 ms.

Boakye-Boateng *et al.* [13] incorporated the idea of interaction familiarity among devices and the resulting implications of honoring a device's requests when calculating trust. Like much of the prior work, however, their study did not leverage trust evaluations to inform the substation's overall risk posture. Moreover, existing publications have not examined the assessment of a trust-enabled IED (one equipped with a trust computation model) through the lens of trust transferability to facilitate its redeployment and integration into a separate substation [19].

Motivation

Due to the stringent operational requirements of substations, it is impractical to deploy conventional security controls—such as encryption or intrusion detection systems—at the station, bay, or process levels. Adding intrusion detection at these tiers would generate redundant network traffic, which, in turn, would degrade the performance of individual devices and the substation as a whole. While encryption remains a feasible option in theory, advanced persistent threats that compromise the SCADA HMI can obtain access to encryption keys or cipher suites, enabling them to issue apparently valid commands to downstream IEDs. Critically, IEDs themselves lack a built-in mechanism to distinguish between authorized and malicious instructions in such scenarios. We argue that building and leveraging trust offers a promising way to overcome these limitations, yet—as highlighted in the previous section—trust-oriented research tailored to substations remains sparse.

Existing studies typically use a per-device trust score solely to determine whether a device is malicious. This approach, however, confines the trust evaluation to a single device and fails to deliver an aggregated view of trustworthiness across the entire substation. Moreover, such isolated trust scores provide no meaningful indication of the substation's broader risk exposure. Our investigation further confirms that no prior trust model has established a direct linkage between device-level trust and substation-wide risk assessment.

A network's risk posture describes its current degree of vulnerability to threats and the potential severity of adverse outcomes resulting from a breach or malfunction. In the context of an electrical substation, this posture is shaped by multiple elements, including the anticipated impact of any successful security compromise.

While device-level trust assessments can mitigate localized risks, a complete understanding of substation security requires considering the combined trust levels of all participating devices. This aggregated trust perspective can then be mapped to an overall evaluation of the substation's risk exposure, thereby delivering actionable insight into its risk posture.

To put this concept into practice, incorporating a multi-agent system (MAS) architecture within the substation becomes essential. Although two earlier publications—Borowski *et al.* [15] and Fadul *et al.* [16]—have examined trust mechanisms inside MAS-based substation designs, neither addresses the particular integration of trust with substation risk posture that this work pursues.

Additionally, there exists a practical scenario in which a trust-enabled IED might need to be relocated to a different substation. In such cases, trust transferability plays a decisive role in evaluating whether the device can safely join and operate within the new environment. Related concerns include verifying that the device remains uncompromised and determining appropriate procedures for identifying and excluding any potentially tainted hardware.

Multi-Agent Systems

A Multi-Agent System (MAS) consists of multiple intelligent agents working together [20]. Each intelligent agent exhibits four fundamental properties: pro-activeness, reactivity, social ability, and autonomy. Pro-activeness requires the agent to take initiative and adapt its actions proactively to achieve its designated goals. Reactivity demands that the agent quickly perceive changes in its surroundings and respond appropriately, consistent with both its objectives and the new conditions. Social ability enables the agent to interact constructively, cooperate, and negotiate effectively with other agents. Autonomy means the agent can operate independently, without continuous direction from external controllers or human operators.

In an MAS, all agents pursue a collective goal, with each agent's individual objectives contributing to the shared mission. MAS architectures fall into three main categories: centralized, decentralized, and hybrid. In a centralized setup, agents send information to and receive directives from a single coordinating agent. In contrast, a decentralized structure allows agents to form peer-to-peer clusters where no single entity holds authority. A centralized design carries the risk of total system failure if the central coordinator is lost. In contrast, a fully decentralized approach can be difficult to optimize because interactions remain confined to local groups. A hybrid architecture seeks to combine the strengths of both models.

MAS concepts have been extensively explored across various smart grid applications, including distributed energy generation [21, 22], the incorporation of renewable energy sources [23], and the coordination and control of microgrids [24].

Modbus TCP

The Modbus protocol plays a central role in the automation, monitoring, and control of industrial equipment [25]. Modbus TCP/IP, which runs over TCP/IP networks using port 502 [26], stands out as a practical option thanks to its straightforward implementation, widespread adoption, and well-documented specifications. It was also the protocol featured in earlier related studies [13, 14, 18], making it a logical choice for building upon prior work. In addition, advanced persistent threats like PipeDream [3] include dedicated modules designed to exploit this protocol. Attacks leveraging Modbus typically involve tampering with Modbus packets, extracting sensitive information from Intelligent Electronic Devices (IEDs) via Modbus messages, or abusing IEDs to interfere with substation operations. Modbus adheres to a strict client–server architecture: the client initiates requests, the server provides responses, and the server does not accept or process unsolicited client messages.

Modbus packet structure

As illustrated in Figure 3, a Modbus TCP packet begins with the Modbus Application Protocol (MBAP) Header. This header includes critical fields such as the transaction identifier, protocol identifier, unit identifier, and length indicator. Following the header is the Modbus Protocol Data Unit (PDU), which contains the function code (FC) along with associated data. The data portion typically specifies parameters such as the starting address to access, the number of consecutive addresses involved, the byte count for written data, the number of items being returned, and other values required to complete the transaction. Key characteristics of these Modbus packets will be extracted and used as input features for our proposed trust model.

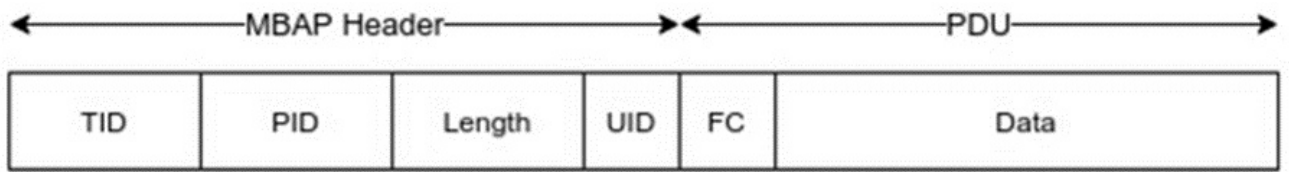


Figure 3. Modbus TCP packet structure.

Modbus address types and function codes

For this study, we have carefully selected a limited set of Modbus function codes (Table 1) based on patterns observed across multiple datasets. In read operations, the client must specify both the initial address and the number of consecutive addresses to retrieve. Similarly, when performing a single-register write, the client provides the target address along with the value to be stored. Every client request sent to the server elicits a corresponding response that returns the relevant address(es) and the current values held at those locations.

Table 1. Modbus address type and its shortlisted function codes.

Address type	Access type	Address size	Address range	Function	Function code
Coil	Read and write	1 bit	1–9999	Read coil	01
				Write a single coil	05
Discrete input	Read only	1 bit	10,001–19,999	Read discrete input	02
Holding register	Read and write	16 bit	40,001–49,999	Read holding register	03
				Write a single register	06
Input register	Read only	16 bit	30,001–39,999	Read input register	04

As indicated in Table 1, both coil and discrete input address types are single-bit entities. In the context of Intelligent Electronic Devices (IEDs), coil addresses are particularly important because they can trigger circuit breaker operations by issuing a Write Single Coil command (Function Code 05) within a Modbus packet. To subsequently verify the breaker’s state, a Read Discrete Input command (Function Code 02) can be sent to obtain the value stored at the corresponding discrete input address. Holding registers, meanwhile, store values that can be modified to affect dynamic parameters—such as the continuously updated voltage readings maintained in input registers.

Models and Scenario

Substation model

The substation, represented as Ξ , is formally defined as $\Xi = (M, N, S)$, where it consists of three primary sets: servers ($S = \{s_1, s_2, \dots\}$), clients ($M = \{m_1, m_2, \dots\}$), and network interconnecting devices ($N = \{n_1, n_2, \dots\}$). Entities in sets M and S can dynamically switch roles between agent (trustor) and subject (trustee). The set N provides the connectivity that links members of S and M . Correspondingly, we define two additional sets: $Q = \{q_1, q_2, \dots, q_i\}$ for queries and $R = \{r_1, r_2, \dots, r_i\}$ for their associated responses. At regular intervals, each client m_i sends queries from Q to a server s_i and then receives the corresponding responses from R . It is important to note that every unique m_i – s_i pairing may produce a distinct Q – R combination. The nature of each query-response transaction is classified according to a memory access type flag (θ),

where $\theta = 0$ indicates a read operation and $\theta = 1$ indicates a write operation. Adversary-generated queries are collectively denoted as $Q' = \{q_1', q_2', \dots, q_i'\}$, while malicious responses issued by the adversary are denoted as $R' = \{r_1', r_2', \dots, r_i'\}$.

Malicious servers and malicious clients are represented by the sets $S' = \{s_1', s_2', \dots\}$ and $M' = \{m_1', m_2', \dots\}$, respectively. These sets include either devices that have been successfully compromised or entirely rogue devices under the adversary's direct control. Any server or client that becomes compromised is reassigned from S or M to S' or M' , respectively.

Attack scenarios

The attacker's main goal when targeting a substation is to seize control of one or more elements in set S , with the ultimate aim of disrupting the broader Smart Grid. In many documented cases, the Intelligent Electronic Device (IED) serves as the primary point of vulnerability. Informed by real-world, publicly reported incidents, this section compiles the specific attack types that serve as the basis for the evaluations conducted in this research. The considered attacks include:

- **Write attack:** Here, the adversary transmits a malicious query q' with $\theta = 1$ directed at server s_i , attempting to overwrite values at all accessible Modbus addresses—either without any prior information gathering or after performing a baseline replay to establish normal patterns. Alternatively, the attack may be more targeted, focusing on a single specific address of s_i using $\theta = 1$, but this variant requires the successful execution of a preceding reconnaissance phase.
- **Query flooding:** In this scenario, a malicious client m' or server s' overwhelms a victim device by sending an abnormally high volume of Q' or R' packets, thereby exhausting the target's processing or memory resources and impairing its normal operation.
- **Malicious packet crafting:** Either m' or s' deliberately constructs and transmits a harmful packet designed to deliver a malicious payload or trigger a buffer overflow condition. Such crafted packets may appear as either q' or r' . Common techniques include payload injection, frame stacking, manipulation of abnormal packet lengths, and false data injection attacks.
- **Baseline replay attack:** After first profiling legitimate substation behavior to avoid raising alarms, the adversary (via m' or s_i') replays previously captured legitimate Q or R messages to a selected target device.
- **Reconnaissance:** When $\theta = 0$, a malicious client m_i' systematically sends q' queries to s_i , scanning across all available Modbus address ranges to map the substation's configuration and gather intelligence about its structure and behavior.

Each of the above scenarios constitutes a separate method by which an adversary can attempt to undermine the substation's operational integrity and reliability.

Mapping to the MITRE ATT&CK ICS Framework

The MITRE ICS ATT&CK framework provides a detailed, structured catalog of the tactics, techniques, and procedures commonly used by adversaries in industrial control system cyberattacks, thereby supporting improved threat detection, analysis, and defensive planning by cybersecurity practitioners. In this work, the attack scenarios described have been systematically mapped to corresponding entries in the MITRE ICS ATT&CK framework [27]. The results of this mapping are summarized in **Table 2**. Out of the 12 tactics defined in the framework, the attacks presented here align with four specific tactics.

Table 2. Attacks mapped to the MITRE ICS ATT&CK framework.

Tactic	Technique	Attack
Collection	Automated Collection	Reconnaissance—Scan addresses
Inhibit Response Function	Denial of Service	Query flooding
		Load malicious payloads
		Delay response
		Modify length parameters
		False injection
		Stack Modbus frames
Impair Process Control	Brute Force I/O	Write to all coils
Evasion	Spoof Reporting Message	Baseline replay

The first tactic, collection, covers actions focused on gathering actionable intelligence about a target system or network, including reconnaissance, network scanning, and systematic data acquisition.

The second tactic, Inhibit Response Function, involves deliberate interference with a system's ability to respond correctly and promptly to legitimate commands or requests. Typical methods include denial-of-service techniques, resource overload, and induced latency.

The third tactic, Impair Process Control, targets the disruption of normal process regulation and control within the system, achieved through actions such as brute-force credential attacks or unauthorized data manipulation and injection.

The fourth tactic, Evasion, focuses on techniques to avoid detection, obscure the origin or intent of malicious activity, or hinder attribution, often through spoofing, masquerading, and other concealment strategies.

Trust Formulation for Substation Devices

In our earlier studies, we emphasized that trust computation rests on two primary elements: familiarity (F_i) and consequence (C_i) [13, 14]. Familiarity is built from three key dimensions: frequency (E_f), intensity (E_i), and similarity (E_s). To assess these dimensions, Modbus-specific attributes derived from either queries (q) or responses (r) serve as inputs. The second component, consequence (C_i), integrates a set of detection flags that reflect different forms of malicious activity: environment status attack flag (τ), replay attack flag (ω), reconnaissance attack flag (ξ), packet manipulation flag (ϕ), and query flooding attack flag (χ). Table 3 provides the definitions of the Modbus features used to calculate F_i and C_i . Comprehensive explanations of these features appear in our prior publications [13, 14].

Table 3. Table of notations .

q or q_i	A query
r or r_i	A response
$x \in \mathbb{Z}^+$	x is a positive integer.
$x \in \mathbb{Q}^+$	x is a positive rational number.
$x \in \{0, 1\}$	x is either 0 or 1.

$x \in [0, 1]$	x is within the range $[0, 1]$.
ϑ	Memory access type (read or write) flag. $\vartheta \in \{0, 1\}$
E_i	Exposure intensity. $E_i \in Q^+$ and $E_i \in [0, 1]$.
E_f	Exposure frequency. $E_f \in Q^+$ and $E_f \in [0, 1]$.
E_s	Similar exposure. $E_s \in Q^+$ and $E_s \in [0, 1]$.
$E_x T$	An exposure's threshold. The notation x is replaced with i , f , or s
κx	An alarm associated with a particular component of trust. The notation x is replaced with E_i , E_f , E_s , C_i , τ , ω , ξ , ϕ , or χ . $\kappa_x \in Z^+$
Γ	A set of Modbus features associated with E_f
Γ_i	A set of Modbus features, extracted from q_i or r_i , associated with E_f
Γ_R	A reference set of Modbus features associated with E_f
y_{frc}	Count for the read coil function code. $y_{frc} \in Z^+$
y_{cq}	Coil quantity. $y_{cq} \in Z^+$
y_{fwsc}	Count for writing the single coil function code. $y_{fwsc} \in Z^+$
y_{cv}	Coil value. $y_{cv} \in \{0, 1\}$
y_{cvs}	Set of coil values. $y_{cvs} = \{x_1, x_2, \dots, x_i\}$ where $x_i \in \{0, 1\}$
y_{fwmc}	Count for the write multiple coils function code. $y_{fwmc} \in Z^+$
y_{cdc}	Coil data byte count. $y_{cdc} \in Z^+$
y_{firdi}	Count for the read discrete input function code. $y_{firdi} \in Z^+$
y_{diq}	Discrete input quantity. $y_{diq} \in Z^+$
y_{didc}	Discrete input data byte count. $y_{didc} \in Z^+$
y_{divs}	Set of discrete input values. $y_{divs} = \{x_1, x_2, \dots, x_i\}$ where $x_i \in \{0, 1\}$
y_{firir}	Count for the read input register function code. $y_{firir} \in Z^+$
y_{irq}	Input register quantity. $y_{irq} \in Z^+$
y_{irdc}	Input register data byte count. $y_{irdc} \in Z^+$
y_{irvs}	Set of input register values. $y_{irvs} = \{x_1, x_2, \dots, x_i\}$ where $x_i \in Q^+$
y_{irv}	Input register value. $y_{irv} \in Q^+$
y_{frhr}	Count for read holding register function code. $y_{frhr} \in Z^+$
y_{hrq}	Holding register quantity. $y_{hrq} \in Z^+$
y_{fwsr}	Count for writing a single register function code. $y_{fwsr} \in Z^+$
y_{hrv}	Holding register value. $y_{hrv} \in Q^+$
y_{fwmr}	Count for Write Multiple Registers function code. $y_{fwmr} \in Z^+$

y_{hrvs}	Set of holding register values. $y_{hrvs} = \{x_1, x_2, \dots, x_i\}$ where $x_i \in Q^+$
y_{hrdc}	Holding register data byte count. $y_{hrdc} \in Z^+$
y_{fs}	Frame size feature. $y_{fs} \in Z^+$
y_{fsR}	Reference frame size feature. $y_{fsR} \in Z^+$
y_{fsi}	Frame size feature for q_i or r_i . $y_{fsi} \in Z^+$
l_{hi}	Length of the MBAP header
y_{fci}	Function code indicator of q_i or r_i . $y_{fci} \in \{0, 1\}$
Z	A set of Modbus features associated with E_i
Z_i	A set of Modbus features, extracted from q_i or r_i , associated with E_i
ZR	A reference set of Modbus features associated with E_i
ζ_{pt}	Pre-time feature. $\zeta_{pt} \in Z^+$
T ζ_{pt}	Pre-time feature threshold. $\zeta_{pt}^T \in Z^+$
ζ_{qq}	Inter-query time feature. $\zeta_{qq} \in Q^+$
ζ_{rr}	Inter-response time feature. $\zeta_{rr} \in Q^+$
ζ_{qr}	Query-response time feature. $\zeta_{qr} \in Q^+$
ζ_{tt}	Transaction time feature. $\zeta_{tt} \in Q^+$
ζ_{to}	Timeout feature. $\zeta_{to} \in Q^+$
y	Replay indicator. $y \in \{0, 1\}$
Ψ	A set of Modbus features associated with E_s
Ψ_i	A set of Modbus features, extracted from q_i or r_i , associated with E_s
Ψ_R	A reference set of Modbus features associated with E_s
ψ_s	State traversed feature. $\psi_s \in \{0, 1\}$
ψ_p or ψ_{pi}	Port mismatch feature. $\psi_p \in \{0, 1\}$
ψ_η or $\psi_{\eta i}$	IP-MAC mismatch feature. $\psi_\eta \in \{0, 1\}$
ψ_{us}	Unknown state feature. $\psi_{us} \in \{0, 1\}$
ψ_{ma}	Address match feature. $\psi_{ma} \in \{0, 1\}$
ψ_{mas}	Address size match feature. $\psi_{mas} \in \{0, 1\}$
ψ_{fc}	Function code match feature. $\psi_{fc} \in \{0, 1\}$
ψ_{mdiq}	Discrete input quantity match feature. $\psi_{mdiq} \in \{0, 1\}$
ψ_{mdir}	Discrete input reference match feature. $\psi_{mdir} \in \{0, 1\}$
ψ_{mcr}	Coil reference match feature. $\psi_{mcr} \in \{0, 1\}$

ψ_{mcq}	Coil quantity match feature. $\psi_{mcq} \in \{0, 1\}$
ψ_{mhrr}	Holding register reference match feature. $\psi_{mhrr} \in \{0, 1\}$
ψ_{mhrq}	Holding register quantity feature. $\psi_{mhrq} \in \{0, 1\}$
ψ_{mirq}	Input register quantity match. $\psi_{mirq} \in \{0, 1\}$
ψ_{mirr}	Input register reference match. $\psi_{mirr} \in \{0, 1\}$
ψ_{ms}	Message sequence flag. $\psi_{ms} \in \{0, 1\}$
F_i	Familiarity. $F_i \in [0, 1]$
τ	Environment status attack flag. $\tau \in \{0, 1\}$
ω	Replay attack flag. $\omega \in \{0, 1\}$
ξ	Reconnaissance attack flag. $\xi \in \{0, 1\}$
χ	Query flooding attack flag. $\chi \in \{0, 1\}$
ϕ	Packet manipulation attack flag. $\phi \in \{0, 1\}$
C_i	Consequence. $C_i \in [0, 1]$
β	Trust score. $\beta \in [0, 1]$
θ_i	Initial state of the device. $\theta_i \in \{0, 1\}$
β_i^o	Previous trust score. $\beta_i^o \in [0, 1]$
β_T	Trust score threshold. $\beta_T \in [0, 1]$
μ	Forgiveness weight. $\mu \in [0, 1]$
θ_μ	Forgiveness state of the device. $\theta_\mu \in \{0, 1\}$

Familiarity-based definitions

Familiarity

Familiarity, expressed as F_i , is mathematically defined in Eq. 5, where E_f stands for frequency, E_s for similarity, and E_i for intensity. Importantly, F_i satisfies the constraint $F_i \geq \min\{E_f, E_s, E_i\}$ and falls within the normalized interval $F_i \in [0, 1]$.

$$F_i = \frac{2}{\sqrt{2}} \begin{pmatrix} \sqrt{\frac{1}{2}}E_f & \sqrt{\frac{1}{2}}E_f & 0 & 1 \\ 0 & \sqrt{\frac{1}{2}}E_s & \sqrt{\frac{1}{2}}E_s & 1 \\ \sqrt{\frac{1}{2}}E_i & 0 & \sqrt{\frac{1}{2}}E_i & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

Exposure frequency

For every received q_i or r_i , a set of characteristics denoted as Γ is derived, as formalized in **Eq. 6**. Γ comprises four specialized feature groups plus one general group: the coil group ($\Gamma_c = \{Y_{frc}, Y_{cq}, Y_{fws}, Y_{cv}, Y_{cvs}, Y_{fwm}, Y_{cdc}\}$), the discrete input group ($\Gamma_{di} = \{Y_{frdi}, Y_{diq}, Y_{didc}, Y_{divs}\}$), the input register group ($\Gamma_{ir} = \{Y_{frir}, Y_{irq}, Y_{irdc}, Y_{irvs}, Y_{irv}\}$), and the holding register group ($\Gamma_{hr} = \{Y_{frhr}, Y_{hrq}, Y_{fwsr}, Y_{hrv}, Y_{fwmr}, Y_{hrvs}, Y_{hrdc}\}$).

The frequency metric E_f is computed according to **Eq.7**, where $E_f \in [0, 1]$ and E_f^T represents the predefined frequency threshold.

$$\begin{aligned} \Gamma &= \{\gamma_{fs}\} \cup \Gamma_c \cup \Gamma_{di} \cup \Gamma_{hr} \cup \Gamma_{ir} \\ 0, \kappa_{E_f} &= 1, \text{ if } l_{h_i} < 7 \\ 0, \kappa_{E_f} &= 2, \text{ if } \gamma_{fs_R} \neq \gamma_{fs_i} \\ 0, \kappa_{E_f} &= 3, \text{ if } \gamma_{fc_i} = 0 \end{aligned} \quad (6)$$

$$\begin{aligned} E_f &= \frac{\Gamma_R \cdot \Gamma_i}{\Gamma_R \cdot \Gamma_i}, \kappa_{E_f} = 0 \text{ if } E_f \geq E_f^T \\ \frac{\Gamma_R \cdot \Gamma_i}{\Gamma_R \cdot \Gamma_i}, \kappa_{E_f} &= 4 \text{ if } E_f < E_f^T \end{aligned} \quad (7)$$

Exposure intensity

During the transmission of q_i or r_i , a collection of timing-related attributes $Z = \{\zeta_{pt}, \zeta_{qq}, \zeta_{rr}, \zeta_{qr}, \zeta_{tt}, \zeta_{to}\}$ is produced and used to determine intensity E_i . These attributes capture: ζ_{pt} (pre-transaction time), ζ_{qq} (inter-query interval), ζ_{rr} (inter-response interval), ζ_{qr} (query-to-response duration), ζ_{tt} (full transaction duration), and ζ_{to} (timeout duration). Intensity E_i is calculated as shown in **Eq. 8**, constrained to $E_i \in [0, 1]$, with E_i^T denoting the intensity threshold.

$$\begin{aligned} 1, \kappa_{E_i} &= 0 \text{ if } \vartheta = 1 \\ 0, \kappa_{E_i} &= 1, \text{ if } \zeta_{pt} > \zeta_{pt}^T \\ E_i &= \frac{\zeta_R \cdot \zeta_i}{\zeta_R \cdot \zeta_i}, \kappa_{E_i} = 0 \text{ if } E_i \geq E_i^T \\ \frac{\zeta_R \cdot \zeta_i}{\zeta_R \cdot \zeta_i}, \kappa_{E_i} &= 2 \text{ if } E_i < E_i^T \end{aligned} \quad (8)$$

Similarity

A feature set denoted as Ψ (**Eq. 9**) is organized into three categories: general packet features $\Psi_{gp} = \{\psi_s, \psi_p, \psi_\eta, \psi_{us}, \psi_{mas}, \psi_{ma}, \psi_{ic}\}$; single-bit register features $\Psi_{sb} = \{\psi_{mdiq}, \psi_{mdir}, \psi_{mcr}, \psi_{mcq}\}$; and byte-oriented register features $\Psi_{br} = \{\psi_{mhr}, \psi_{mhrq}, \psi_{mirq}, \psi_{mirr}\}$. The similarity measure E_s , formulated in **Eq. 10**, is bounded within $E_s \in [0, 1]$ and serves as the basis for generating the corresponding similarity score.

$$\begin{aligned} \Psi &= \Psi_{gp} \cup \Psi_{sb} \cup \Psi_{br} \\ 0, \kappa_{E_s} &= 1 \text{ if } \psi_{\eta_i} \\ 0, \kappa_{E_s} &= 2 \text{ if } \psi_{p_i} \neq 502 \end{aligned} \quad (9)$$

$$E_s = \begin{cases} \frac{\Psi_R \cdot \Psi_i}{\Psi_R \cdot \Psi_i}, \kappa_{E_s} = 0, & \text{if } E_s \geq E_s^T \\ \frac{\Psi_R \cdot \Psi_i}{\Psi_R \cdot \Psi_i}, \kappa_{E_s} = 3, & \text{if } E_s < E_s^T \\ E_s^T, \kappa_{E_s} = 4, & \text{if } \psi_{ms} = 1, \psi_{us} = 1 \end{cases} \quad (10)$$

Consequence

Consequence, denoted as C_i , is computed by combining the following binary detection flags: environment status attack flag (τ), replay attack flag (ω), reconnaissance attack flag (ξ), packet manipulation attack flag (ϕ), and query flooding attack flag (χ). The formal expression for C_i appears in [Eq. 11](#).

$$C_i = \begin{cases} 0, & \text{if } \tau \mid \omega \mid \xi \mid \chi \mid \phi = 0, \kappa_C = 0 \\ \xi, & \text{if } \xi \neq 0, \kappa_C = \kappa_\xi \\ \tau, & \text{if } \tau \neq 0, \kappa_C = \kappa_\tau \\ \omega, & \text{if } \omega \neq 0, \kappa_C = \kappa_\omega \\ \chi, & \text{if } \chi \neq 0, \kappa_C = \kappa_\chi \\ \phi, & \text{if } \phi \neq 0, \kappa_C = \kappa_\phi \end{cases} \quad (11)$$

Trust of a device

The trust assigned to a device, represented as $T_i = \{\beta_i, \kappa_{E_i}, \kappa_{E_s}, \kappa_{E_i}, \kappa_C\}$, is an ordered tuple. Here, β_i is the overall trust score, while the κ terms capture penalizing factors that can reduce trust. The computation of the trust score β_i is given in [Eq. 12](#), where β_i is bounded within the interval $[-1, 1]$. In this formula, θ_i denotes the device's initial trust state before any computation, β_{oi} denotes the trust score from the previous evaluation period, β_{Ti} denotes the predefined trust threshold, μ is a forgiveness weighting parameter restricted to $[0, 1]$, and θ_μ denotes the forgiveness state. Aspects related to forgiveness mechanisms are deferred for future investigation.

Within the earlier trust tuple presented in [Eq. 1](#), the risk-related parameter r_{ij} directly corresponds to C_i , and the prior trust value T_{ij}' aligns with β_i^o . The remaining parameters—those tied to the three exposure dimensions (frequency, intensity, and similarity)—are closely interrelated, as these exposure metrics collectively encode substantial information relevant to the trust formulation.

$$\beta_i = \begin{cases} \beta_i^T, & \text{if } \theta_i = 1 \\ F_i - C_i, & \text{if } \theta_i = 0 \\ F_i - C_i + \mu, & \text{if } \theta_i = 0, \theta_\mu = 1, \beta_i^o < \beta_i^T \end{cases} \quad (12)$$

Out of sequence handler

A key limitation identified in the earlier trust model (described in the preceding subsections) was its vulnerability to out-of-sequence packet arrival, which risked generating false positives [\[13, 14\]](#). To manage queries or responses that arrive out of order, the proposed approach implements an iterative tracking mechanism that maintains separate pointers for the query sequence (Q) and the response sequence (R). These pointers, labeled x_q and x_r respectively, mark the expected position of the next (i-th) legitimate query or response that a device should receive ([Eqs.13 and 14](#)).

When an incoming query or response arrives, it is passed, together with the current pointer, to a dedicated handling function. This function decides whether to accept the packet as the current i -th message or to reassign it to an earlier position k within the Q or R sequence (Eq. 15). Based on this decision, two indicators are updated accordingly: the replay indicator (y) and the message sequence flag (ψ_{ms}). Both are activated whenever the processed message corresponds to a position other than i . The value of i is subsequently provided to the computations of E_i , E_s , E_f , and to the various detection flags that contribute to C_i .

$$fx_q, q = q_i \quad (13)$$

$$fx_r, r = r_i$$

$$\begin{aligned} i, \psi_{ms} &= 0, \text{ if } fx_q, q = q_i \\ i-1, y &= 1, \psi_{ms} = 1, \text{ if } fx_q, q = q_{i-1} \\ k, y &= 1, \psi_{ms} = 1, \text{ if } fx_q, q = q_k \end{aligned} \quad (14)$$

$$\begin{aligned} i &= \\ i, \psi_{ms} &= 0, \text{ if } fx_r, r = r_i \\ i-1, y &= 1, \psi_{ms} = 1, \text{ if } fx_r, r = r_{i-1} \\ k, y &= 1, \psi_{ms} = 1, \text{ if } fx_r, r = r_k \end{aligned} \quad (15)$$

Risk Posture

To accurately assess a substation's risk posture, a thorough, systematic risk evaluation process is required. For this purpose, we utilize an established risk assessment methodology previously introduced in our prior work [28]. This methodology enables the evaluation of risk levels for each device within the substation, which are then aggregated to derive the substation's overall risk posture. Here, the set $D = \{d_1, d_2, \dots, d_n\}$ represents the complete collection of n devices present in the substation.

The final reachability matrix captures the functional dependencies and connectivity relationships among the devices in the substation. This matrix, denoted $F = (f_{ij})_{m \times m}$ (where $m < n$), consists of binary entries (0 or 1) and can be graphically represented as a directed graph.

The functional impact associated with a particular device d_i depends on its degree within the reachability structure, as expressed in Eq. 16:

$$Cd_i = \deg d_i. \quad (16)$$

Consequently, the total functional influence contributed by all devices in the substation is defined according to Eq. 17:

$$C_T D = \sum_{i=1}^n Cd_i \quad (17)$$

Each device is assigned to one of several criticality categories, represented as $I = \{d_i, \dots, d_p\}$ where $I \subset D$ and $p < n$. These criticality categories are collectively denoted $L = \{l_1, l_i, \dots, l_m\}$ (with $1 \leq i \leq m$), allowing for up to m distinct levels

while satisfying the constraint $m < n$.

The aggregated functional impact of all devices belonging to a specific criticality level is given by **Eq. 18**:

$$C_T l = \sum_{i=1}^p C d_i \text{ where } d_i \in l \quad (18)$$

Identifying functional influence of affected devices

Consider a set $X = \{d_i, \dots, d_q\}$ containing q devices ($q < n$) that have detected malicious behavior originating from a compromised device. All devices in X have been assigned a trust score $\beta < \beta^T$, and X forms a subset of the full device population D .

Let $Y = \{d_i, \dots, d_g\}$ denote a set of g devices ($g < n$) that were classified as malicious by the devices in X . Y is likewise a subset of D .

Now define $X_k = \{d_i, \dots, d_z\}$ as the subset of z devices within the highest criticality level k , such that $X_k \subset X \cup Y$. The combined functional impact of the devices in X_k is formally expressed in **Eq. 19**:

$$C_T X_k = \sum_{i=1}^z C d_i \quad (19)$$

Calculating risk posture

Let $C_T(X')$ denote the total functional impact contributed by devices belonging to all criticality levels below level k , as expressed in **Eq. 20**:

$$C_T X' = \sum_{i=1}^{k-1} C_T l_i \quad (20)$$

The resulting cascading impact, represented by Λ , is defined according to **Eq. 21**:

$$\Lambda = \begin{cases} 0, & \text{if } \beta \geq \beta^T \quad \forall D \\ C_T D^{-1} C_T X' + C_T X_k, & \\ 1, & \text{if } \beta \leq \beta^T \quad \forall d \in l \end{cases} \quad (21)$$

Finally, the substation's overall risk posture, denoted τ , is formally specified in **Eq. 22**.

$$\tau = 1 - \frac{2(1 - \Lambda)}{m(m+1)} \sum_{j=1}^m \frac{j}{n} \sum_{i=1}^n \beta_i \quad (22)$$

The trust score assigned to the substation as a whole is precisely given by **Eq. 23**:

$$\beta_s = \frac{2}{m(m+1)} \sum_{j=1}^m \frac{j}{n} \sum_{i=1}^n \beta_i \quad (23)$$

Trust Transferability

The topology of the substation network, consisting of n substations, is modeled as a directed graph $S = (\Xi_{ij})_{n \times n}$ with an associated adjacency matrix, where Ξ_{ij} indicates the presence (or absence) of a connection between substations Ξ_i and Ξ_j . The transitive closure of this graph is calculated using the Floyd–Warshall Algorithm [29], as shown in **Eq. 24**:

$$\Xi_{ij} = \Xi_{ij} \vee (\Xi_{ik} \wedge \Xi_{kj}); \forall \Xi_{ij} \quad (24)$$

From the resulting modified matrix, the degree of substation Ξ_i is computed following the expression in **Eq. 25**.

$$C \Xi_i = \deg \Xi_i. \quad (25)$$

This computation produces a vector $C = \{C(\Xi_1), C(\Xi_2), \dots, C(\Xi_n)\}$ containing the degree value for each substation Ξ_i . The vector C , together with the substation identifier Ξ_i , is input into a ranking function that assigns a rank to Ξ_i based on the magnitude of its degree. The complete set of ranks for all substations is then assembled into a vector, as formulated in **Eq. 26**:

$$f \Xi_i, C = (r_i)_{n \times 1} \quad (26)$$

Let $T = \{\beta_1, \beta_2, \dots, \beta_k\}$ represent the sequence of trust scores recorded over a defined time window t . For an Intelligent Electronic Device (IED), trust scores derived from queries sent by the SCADA HMI are denoted T_q , while those derived from responses received by the IED from the SCADA HMI are denoted T_r .

Next, using the trust scores associated with a newly introduced IED j and those of the replaced or removed IED i , the metrics F_q and F_r are calculated according to **Eqs. 27 and 28**, respectively. These metrics are referred to as the query acceptance flag and response acceptance flag:

$$F_q = \begin{cases} 1, & \text{if } \psi \leq \frac{T_{j_q} \cdot T_{i_q}}{T_{i_q} T_{i_q}} \\ 0 & \end{cases} \quad (27)$$

$$F_r = \begin{cases} 1, & \text{if } \psi \leq \frac{T_{j_r} \cdot T_{i_r}}{T_{j_r} T_{i_r}} \\ 0 & \end{cases} \quad (28)$$

Based on the values of F_q and F_r , the overall acceptance flag ϵ is determined as specified in **Eq. 29**:

$$\varepsilon = \begin{cases} 1, & \text{if } f_q \wedge f_r = 1 \\ 0 & \end{cases} \quad (29)$$

A predefined minimum probation interval, denoted t_{pmin} , is established to allow the new IED j to prove its trustworthiness before full integration into the substation network. The actual probation duration t_p assigned to IED j within substation Ξ_i depends on both the substation's rank r_i and the cascading effect Λ introduced by the device, following the expression in **Eq. 30**. Here, t'_p acts as a countdown timer initialized to $t'_p = t_p$ at the start of the probation phase:

$$t_p = t_{pmin} \left(1 + \frac{1}{\log(n + 2 - r_i)} \right) + \Lambda \quad (30)$$

The parameter ϖ represents the probation point, which is used to adjust (increment or decrement) the countdown timer t'_p at designated checkpoints during the probation period:

$$\varpi = \frac{0.5t \left(|T_{j_q}| + |T_{i_r}| \right)}{|T_{j_q}| |T_{i_r}|} \quad (31)$$

The term ϱ indicates the consideration stage, and κ is the consideration weight, where κ is a real number satisfying $1 \leq \kappa \leq 2$:

$$\varrho = \frac{t_p}{\kappa} \quad (32)$$

The countdown timer t'_p tracks the remaining probation time, starting at $t'_p = t_p$. The outcome of the countdown is computed using **Eq. 33**, in which λ denotes the point-weighting factor (bounded by $1 \leq \lambda \leq t_p$) and t_e represents the time that has elapsed since the timer began:

$$t'_p = \begin{cases} t'_p + \lambda \varpi, & \text{if } \beta < \beta^T \\ t'_p - \lambda \varpi, & \text{if } \beta > \beta^T, \varrho \geq t'_p \\ t'_p - t_e & \end{cases} \quad (33)$$

Ultimately, the transferability flag $\aleph$ serves as the definitive criterion for deciding whether IED j should be accepted into the substation network or rejected:

$$\aleph = \begin{cases} 1, & \text{if } t'_p \leq 0 \\ 0, & \text{if } t'_p \geq t_p \end{cases} \quad (34)$$

Implementation

All testing was performed in a virtualized Linux environment (Ubuntu 20.04.3 LTS). The virtual machine ran on an Intel® Xeon® CPU E5-2695 v4 processor configured with 2 cores clocked at 2.10 GHz. The model was developed in Java and

packaged as an executable JAR archive. Individual Docker containers were launched to represent both Intelligent Electronic Devices (IEDs) and SCADA Human-Machine Interfaces (HMI). Separate Python scripts were written to handle the core functional behavior of the IEDs and the SCADA HMIs, respectively.

For IEDs, the implemented logic periodically generated random voltage fluctuations or applied value changes in response to SCADA HMIs' commands. The SCADA HMI logic, on the other hand, performed transformer tap adjustments based on IED-reported voltage readings and issued open or closed commands to circuit breakers (CBs) whenever overvoltage or undervoltage thresholds were crossed.

As depicted in **Figure 4**, each Docker container was configured in one of two ways: either it enclosed both the JAR archive and associated Python scripts, or it contained only the Python scripts. Containers holding merely the scripts represent insecure IEDs or insecure SCADA HMIs. Containers that bundle both the JAR file and the scripts are designated as trust IEDs or trust SCADA HMIs. Within every trusted (secure) device, a dedicated agent periodically forwards trust evaluation scores to a single central agent. That central agent then calculates the substation's current risk level, creates corresponding log entries, and transmits the processed data to the ELK stack [30] to produce a dashboard view styled after a Security Operations Center (SOC).

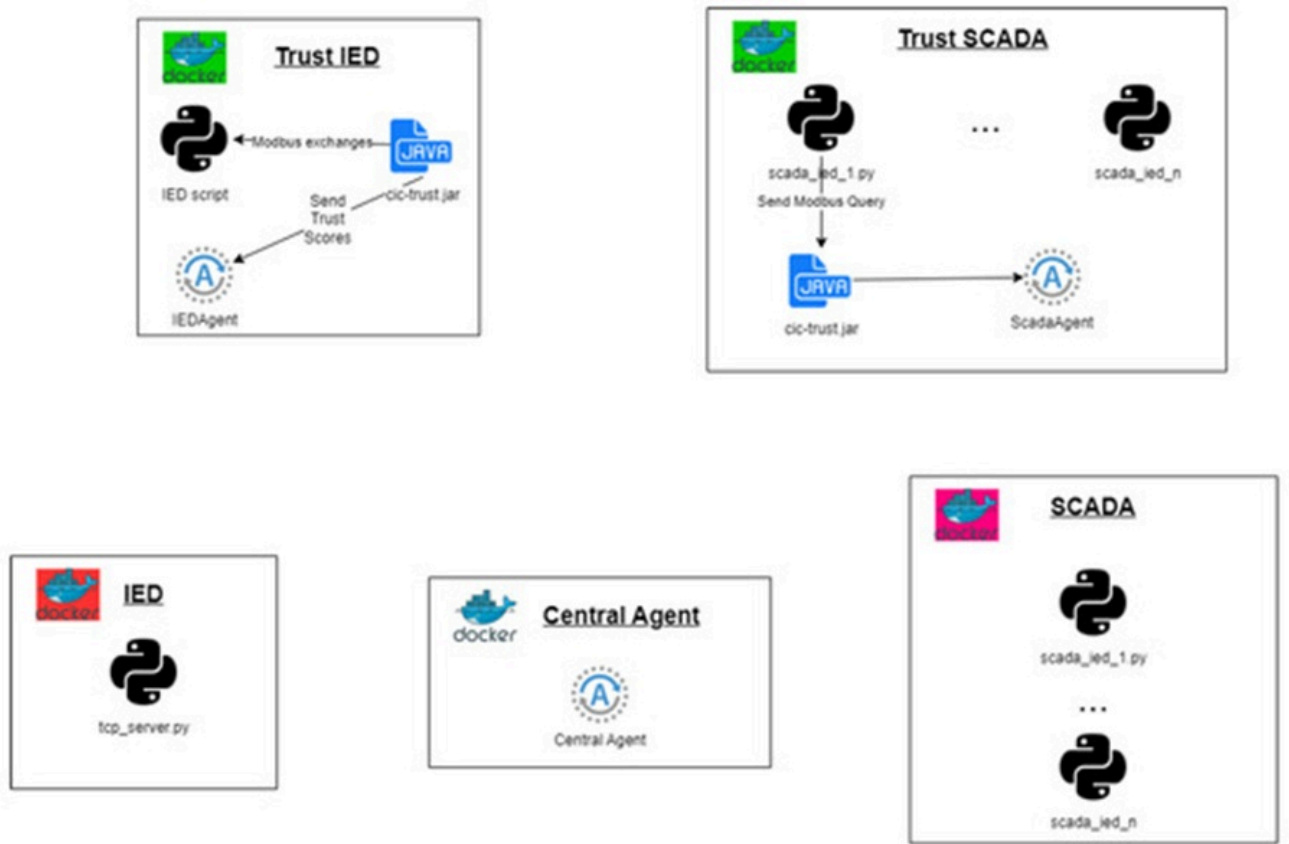


Figure 4. Docker structure of simulated devices.

Log messages emitted by the JAR component are aggregated and rendered graphically via the ELK stack. The overall layout of this container-based test environment appears in **Figure 5**; note that bidirectional communication (double-headed arrows) applies to all links except those directed toward the ELK stack.

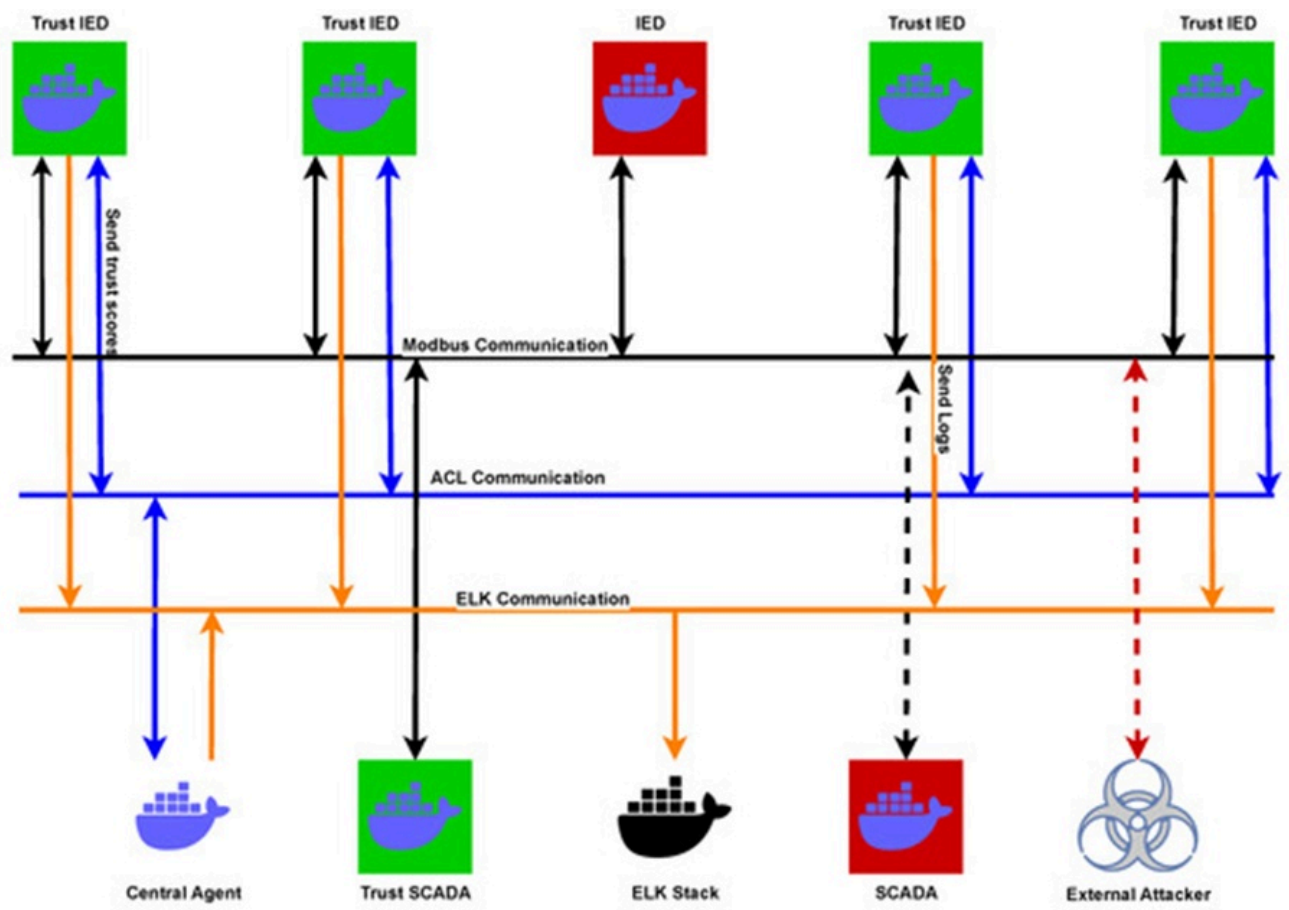


Figure 5. Architecture of containerized testbed.

Implementation of the multi-agent framework relied on JADE. Under this architecture, each trusted device hosts its own agent that interacts with one central coordinating agent. The central agent performs the final evaluation of the substation-wide risk posture.

Trust values produced by our system were aligned with a previously published trust scale [31] and then correlated with alert levels published by the Multi-State Information Sharing and Analysis Center (MS-ISAC) [32] — see Figure 6 for the mapping. Additionally, the numerical output from Eq. 22 was translated onto the standardized NIST Risk Impact Assessment Scale [33], as presented in Figure 7.

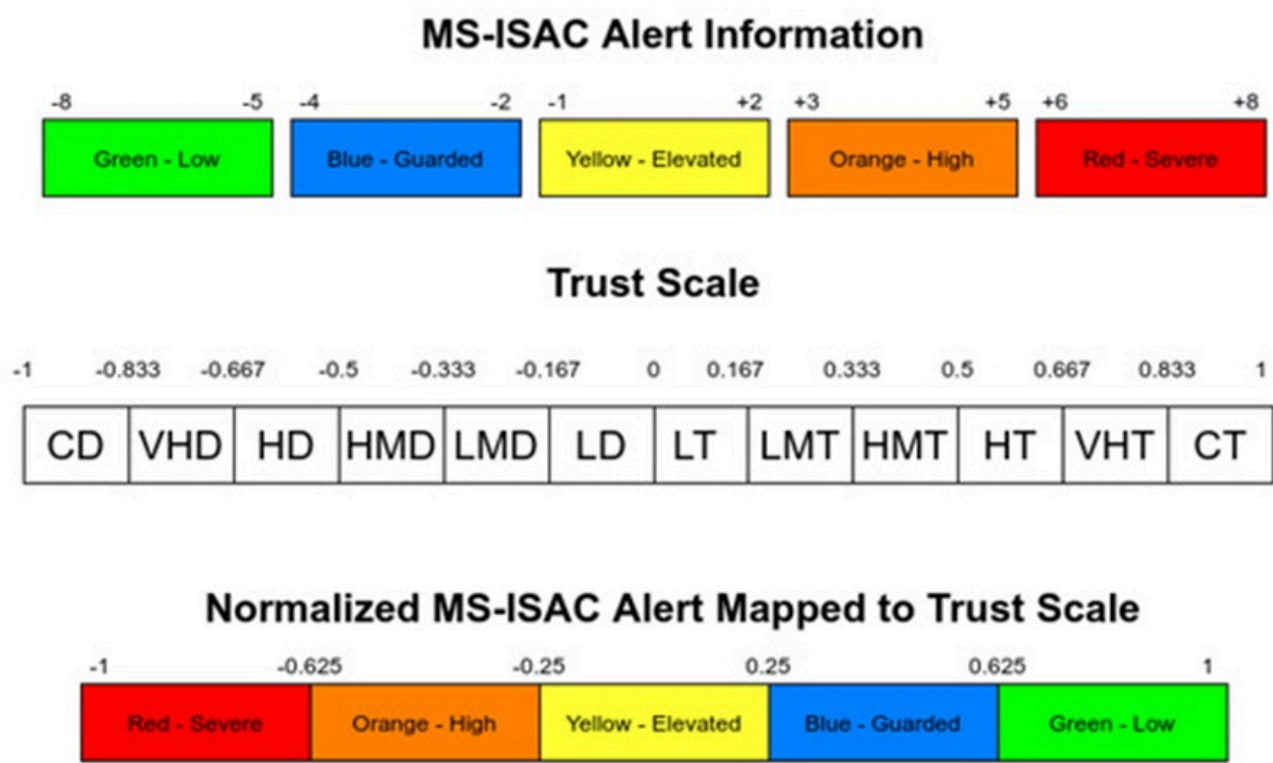


Figure 6. MS-ISAC alert mapped to trust scale.



Figure 7. NIST risk impact assessment scale.

Devices classified as trust IEDs can temporarily suspend all interactions with any identified malicious entity for a set time window. Malicious actions occur at unpredictable moments, each followed by a cooldown interval that exceeds the trust-based blocking duration. In cases where a high volume of incoming messages threatens to saturate the queue, trust IEDs restrict updates sent to the central agent exclusively to occasions when the trust score has actually changed.

The evaluation of the proposed approach rested on several foundational assumptions:

- Network communication inside the substation follows predictable patterns stemming from engineer-defined query schedules.
- Non-Modbus and non-IT-related attack vectors are countered through established CVE and CWE remediation methods and are therefore excluded from the present discussion.
- Adversaries possess the capability to modify only Modbus protocol contents, since the equipment vendor implements strong safeguards against tampering with TCP, IP, and Ethernet layers.
- Network communication by all devices is restricted exclusively to the standard Modbus port.

- The central control facility is presumed already under adversary control, yet no alerting mechanisms have detected this breach.

Based on measurements reported by Boakye-Boateng *et al.* [28], **Table 4** lists the functional impact values associated with the IED population. The recorded CTD figure stands at 598. Space limitations prevented the inclusion of functional influence data for the remaining devices. Similarly, IEDs exhibiting equal dependency counts were consolidated into a single table row. In the context of our experiments, IED1A and IED4C were treated as trusted IEDs, whereas IED1B was categorized as a standard (non-trusted) IED. This differentiation permits analysis of substation risk posture behavior across multiple situations — for instance, attempted compromise of trusted devices or actual takeover of IED1A.

Table 4. List of IEDs’ functional influences.

Devices	Functional influence
IED2A, IED2B	4
IED4A, IED4B, IED4C, IED5A, IED5B, IED5C	9
IED3A, IED3B	11
IED6A	17
IED2C, IED2D	23
IED1C	36
IED1A, IED1B	39

We examined two separate cases to gauge the substation’s risk posture, relying on the Docker-enabled multi-agent system (MAS) architecture described earlier:

- Scenario 1: Trust SCADA HMI Control — Here, the substation operates exclusively under the supervision of a trust SCADA HMI container. Attacks are carried out using two distinct approaches. The first involves an external adversary launching attacks directly from their own device. The second simulates a compromised internal device by using regular (non-trust) IEDs to mimic malicious actions.
- Scenario 2: Regular SCADA Control — In this case, a standard (non-trust) SCADA container (marked with dotted lines) oversees all IEDs. As with the previous scenario, two attack vectors are applied. The first replicates the external attack method used in Scenario 1. The second originates from the regular SCADA container itself, emulating attack patterns commonly reported in real-world documentation.

These two scenarios are designed to evaluate how the substation’s risk level behaves under different operational and threat conditions, thereby revealing potential weak points and demonstrating the practical value of the proposed architecture.

To investigate transferability inside our test environment, we defined four specific scenarios. In every case, any newly introduced or already present trust IED maintains appropriate conduct, reflected by trust scores falling in the Green—Low category. The scenarios are outlined below:

- Scenario 1: Normal Replacement — A freshly deployed trust IED replaces an existing trust IED and behaves normally from the outset.

- Scenario 2: Compromised Replacement (Immediate) — A maliciously altered IED is substituted for an existing trust IED; once accepted, it immediately starts displaying harmful actions.
- Scenario 3: Compromised Replacement (Delayed) — Identical to Scenario 2 in setup, except that the malicious conduct only appears after exceeding the consideration window, defined as half of the full probation duration.
- Scenario 4: Trust IED with Poor Trust Scores — A new trust IED replaces an existing one, yet its associated trust scores remain consistently unfavorable.

These four cases enable a thorough examination of transferability under diverse circumstances, allowing evaluation of the proposed method's resilience and overall performance across different situations.

Evaluation

This section summarizes the key results produced by our trust management framework. We begin by presenting findings on response latency, comparing regular IEDs against trust-enabled IEDs. Next, we analyze how the trust computation model and the risk posture assessment perform under various attack types. Finally, we report the outcomes derived from testing the trust transferability mechanism.

Performance

To verify that trust IEDs introduce only negligible delays and preserve acceptable response times, we conducted a dedicated latency study. Industry vendors commonly accept latencies of 4 ms to 1000 ms for communication between the bay and station levels. **Figure 8** shows the response time measurements for each IED in the setup. Here, IED1B serves as the baseline regular IED, whereas IED1A and IED4C represent trust IEDs. As shown in **Figure 8**, no response time exceeds 10 ms, confirming that the system operates well within tolerable performance boundaries.

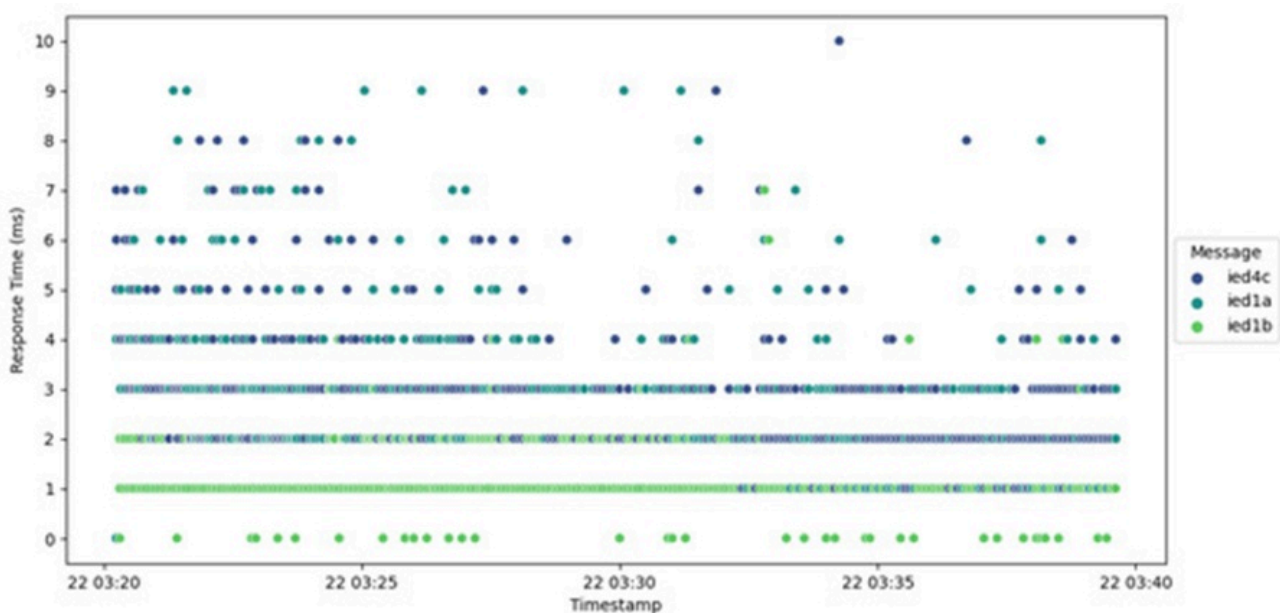


Figure 8. Plot of response time for all IEDs.

Figure 9 focuses specifically on IED1B (the regular IED) and shows response times clustered between roughly 0 ms and 7 ms. Occasional spikes above 2 ms are generally attributable to processing overhead. This plot helps characterize the typical timing behavior of a non-trust IED and highlights areas where computational load may influence performance.

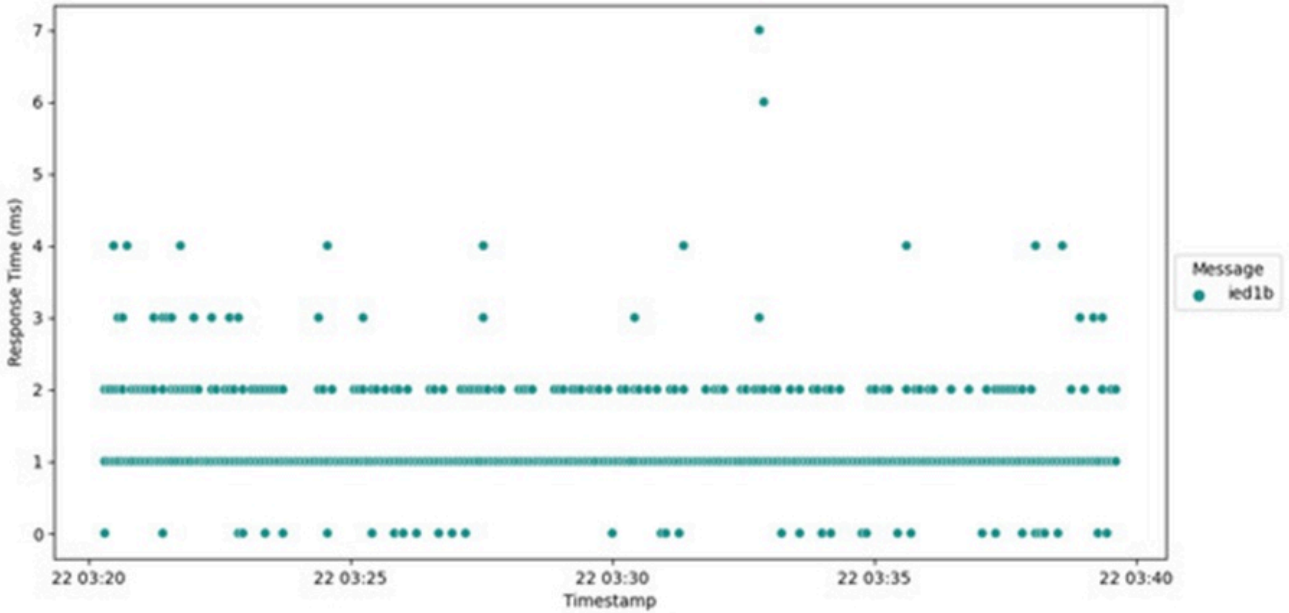


Figure 9. Plot of response time IED1B.

Examination of response times for the trust IEDs—IED1A (Figure 10) and IED4C (Figure 11)—shows that most communication events are approximately 2 ms slower than in IED1B. Yet, all values remain comfortably below the 10 ms threshold. It should be noted that further optimization might be achievable by switching from Java to native code execution, potentially reducing these modest overheads even more.

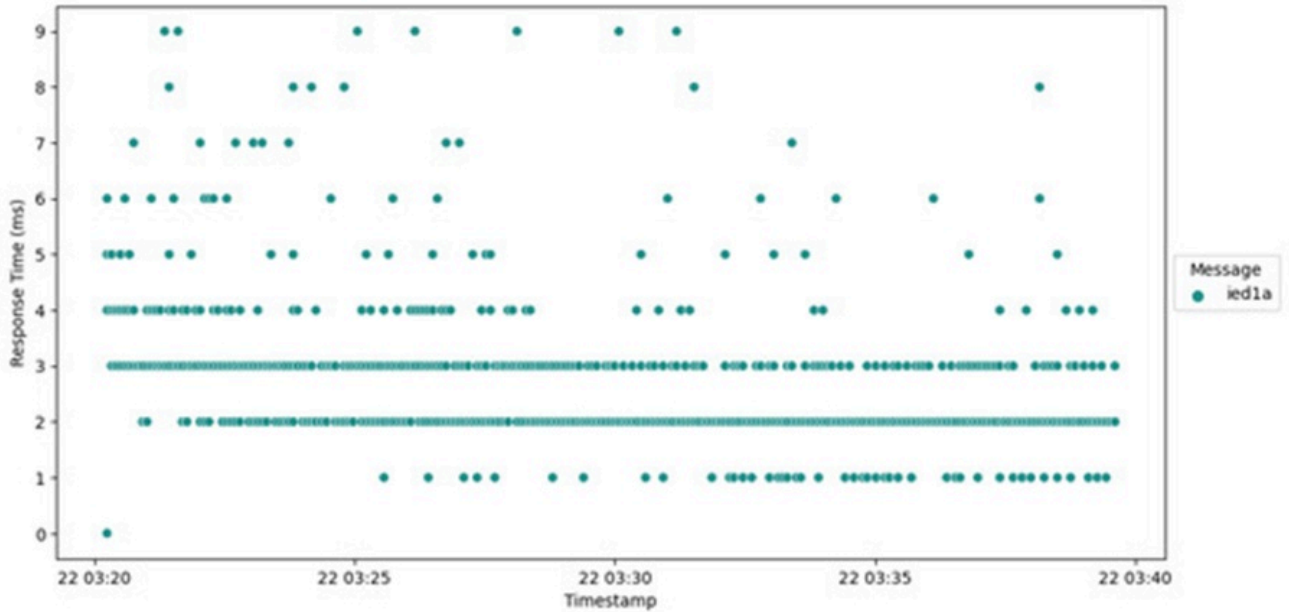


Figure 10. Plot of response time IED1A.

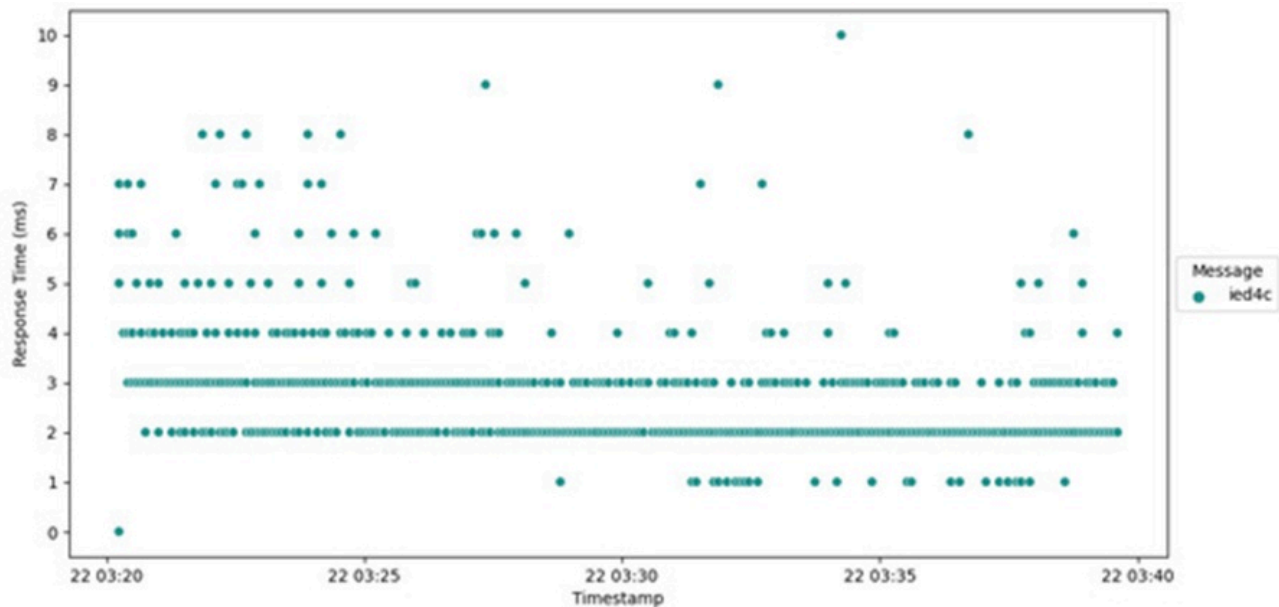


Figure 11. Plot of response time IED4C.

Risk posture assessment

Here, we report how the risk posture computation module behaved when exposed to multiple classes of simulated attacks.

Rogue device attacks

No matter which attack pattern the rogue device applied against the trust-enabled IEDs, the system consistently rejected every request because the originating IP address could not be recognized (**Figure 12**).

When the adversary targeted the high-ranking IED1A, this single incident was sufficient to push the substation-wide risk posture into the “High” category (**Figure 13**). By contrast, applying the same attack style against the lower-ranking IED4C resulted in only a “Low” risk classification for the substation, as shown in **Figure 14**.

The experiments clearly demonstrate that a device's position in the ranking hierarchy plays a decisive role in determining the final risk posture during rogue-device incidents. **Table 5** condenses the key findings.

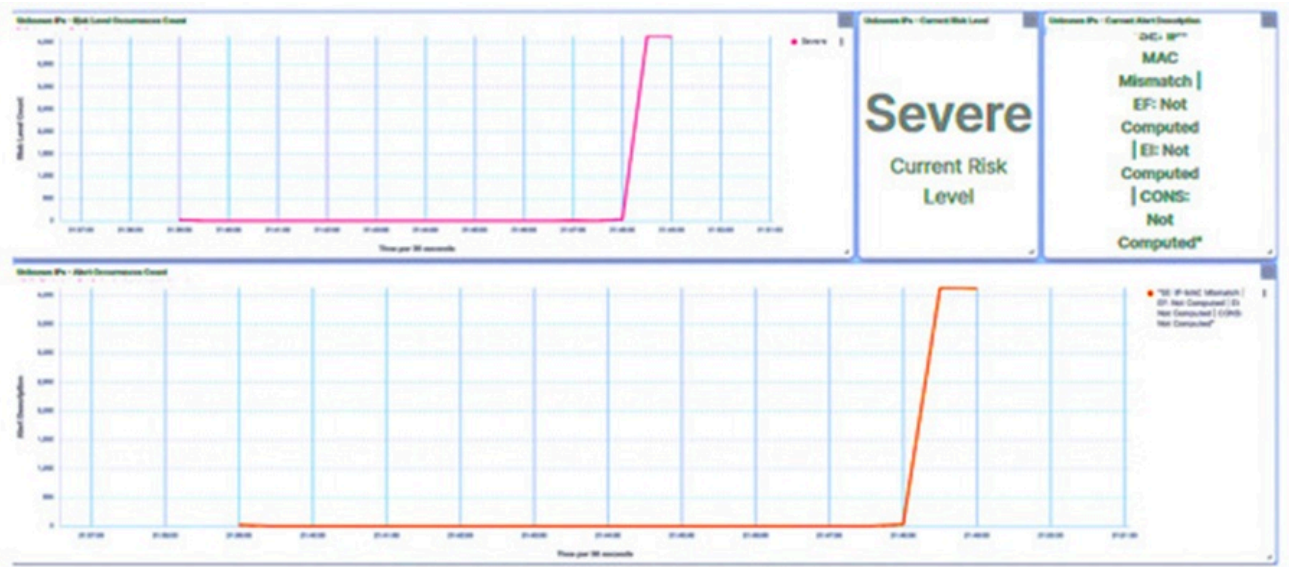


Figure 12. IED1A interface — detection of rogue-device attack

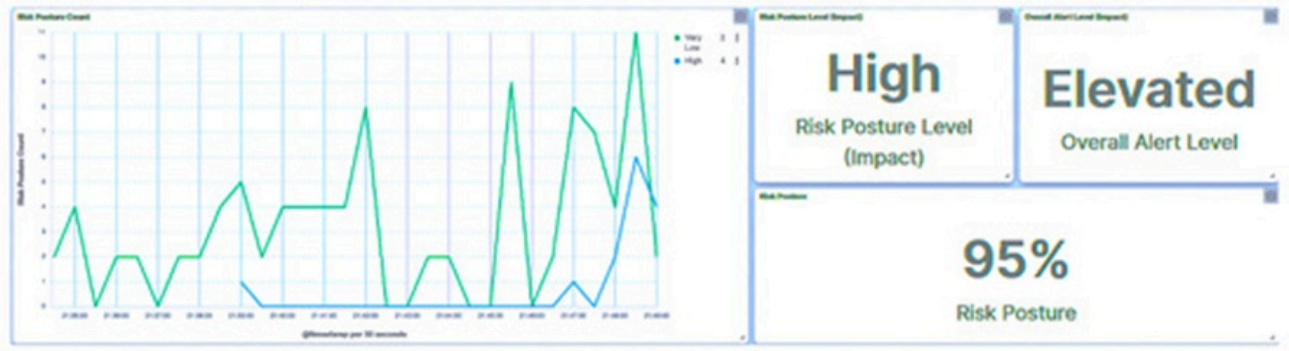


Figure 13. Central agent interface — rogue attack targeting IED1A



Figure 14. Central agent interface — rogue attack targeting IED4C

Table 5. Summary of outcomes—rogue device attacks

Trust device	Attack	Alert	Affected exposure	Device risk level	Risk posture	Outcome

IED1A	Load a malicious payload	IP Mismatch	Similarity	Severe	High	Rogue device blocked
IED4C	Load a malicious payload	IP Mismatch	Similarity	Severe	Low	Rogue device blocked
IED1A	Modify length parameters	IP Mismatch	Similarity	Severe	High	Rogue device blocked
IED4C	Modify length parameters	IP Mismatch	Similarity	Severe	Low	Rogue device blocked
IED1A	Query flooding	IP Mismatch	Similarity	Severe	High	Rogue device blocked
IED4C	Query flooding	IP Mismatch	Similarity	Severe	Low	Rogue device blocked
IED1A	Reconnaissance	IP Mismatch	Similarity	Severe	High	Rogue device blocked
IED4C	Reconnaissance	IP Mismatch	Similarity	Severe	Low	Rogue device blocked
IED1A	Stack Modbus frames	IP Mismatch	Similarity	Severe	High	Rogue device blocked
IED4C	Stack Modbus frames	IP Mismatch	Similarity	Severe	Low	Rogue device blocked
IED1A	Write to all coils	IP Mismatch	Similarity	Severe	High	Rogue device blocked
IED4C	Write to all coils	IP Mismatch	Similarity	Severe	Low	Rogue device blocked

Attacks originating from a compromised SCADA HMI

Reconnaissance/Automated Register Reading:

The attacker attempted a systematic readout of all Modbus registers. **Figure 15** shows that the trust IED almost instantly detected the very first anomalous packet, classified it as an Unknown Read Query, and placed the offending SCADA HMI on its blacklist. Concurrently, the IED reduced its self-assessed trust level to Severe. While the consequence factor was updated, most other risk metrics stayed uncalculated at that stage.

Based on the newly received trust score, the central agent upgraded the substation risk posture to High after reconnaissance focused on IED1A (**Figure 16**). The pattern repeated for IED4C—its trust score also dropped to Severe (**Figure 17**)—yet, because of its lower hierarchical rank, the substation risk only rose from Very Low to Low (**Figure 18**).

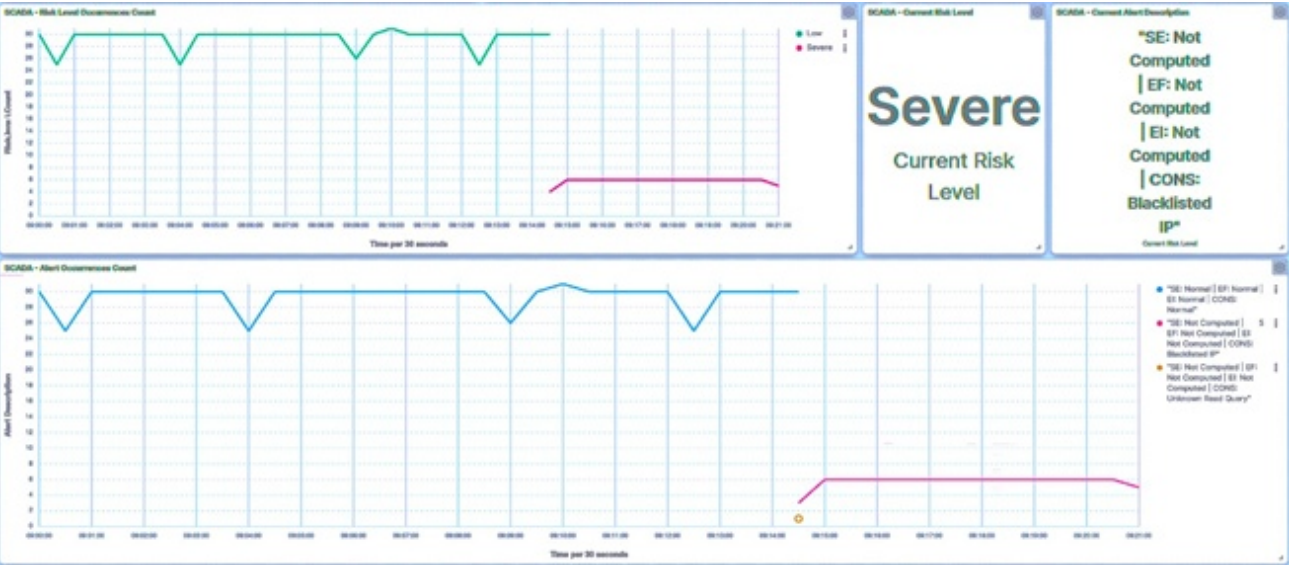


Figure 15. IED1A interface—SCADA reconnaissance identified



Figure 16. Central agent interface—reconnaissance attack against IED1A



Figure 17. IED4C interface—SCADA reconnaissance identified

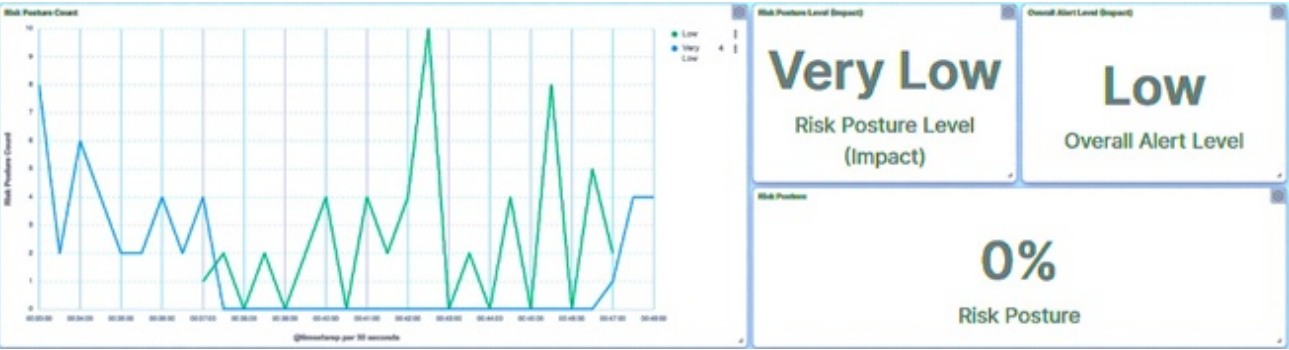


Figure 18. Central agent interface—reconnaissance attack against IED4C

Denial-of-Service Attacks:

We evaluated three DoS variants: query flooding, malicious payload insertion, and stacked Modbus frames.

- Query Flooding

A valid query was repetitively transmitted to overload the trust IEDs. The detection engine correctly identified the flood, as displayed in **Figure 19**. Certain packets arrived out of order (tagged with a sequence mismatch), while others exceeded the query rate threshold (tagged as flooding). Both types of anomaly prompted the trust IED to downgrade its trust score to Severe and blacklist the SCADA source.

For attacks against IED1A, the central agent—acting on the updated trust value—immediately raised the substation risk posture to High (**Figure 20**). The outcome was analogous for IED4C: the trust score fell to Severe (**Figure 21**), but the substation risk posture advanced only from Very Low to Low owing to the device's lower rank (**Figure 22**).

Importantly, the counts of generated alerts and risk-level transitions were identical in both the IED and SCADA dashboards, verifying model consistency.



Figure 19. IED1A interface—SCADA flooding detected

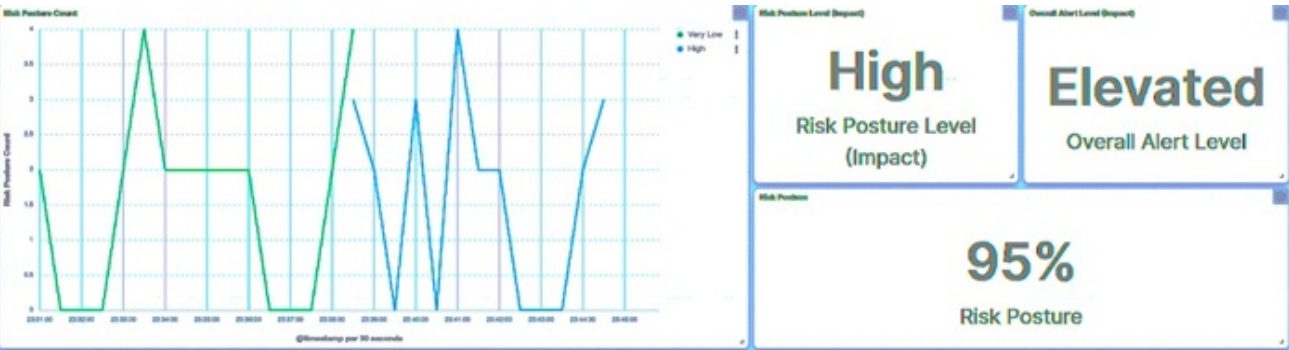


Figure 20. Central agent interface—flooding attack on IED1A

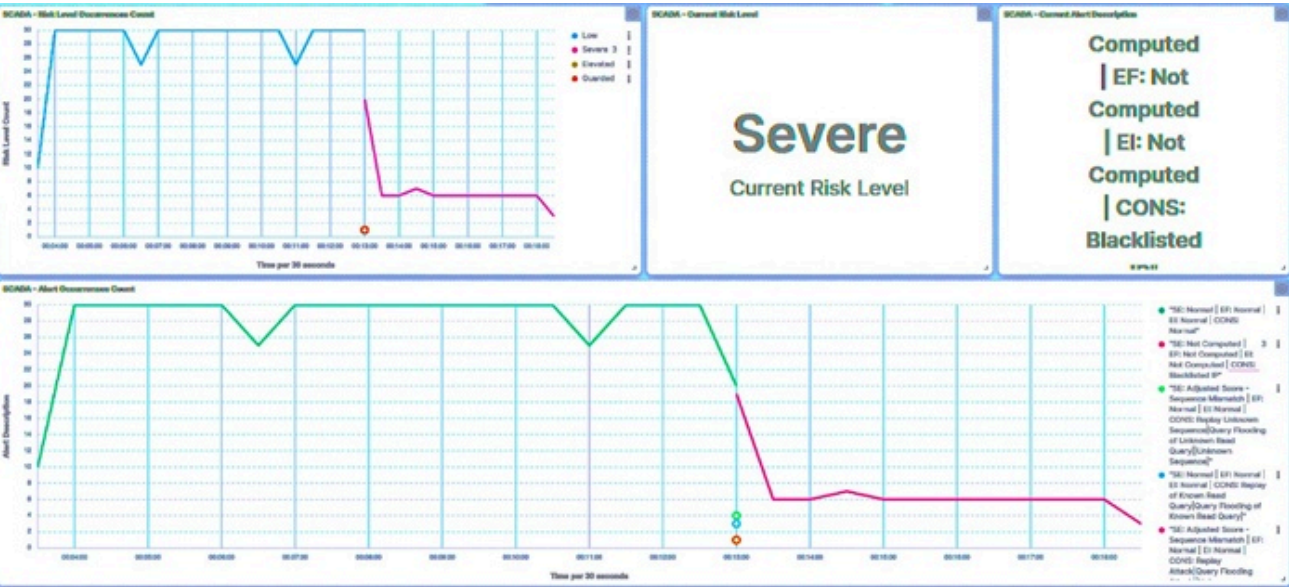


Figure 21. IED4C interface—SCADA flooding detected

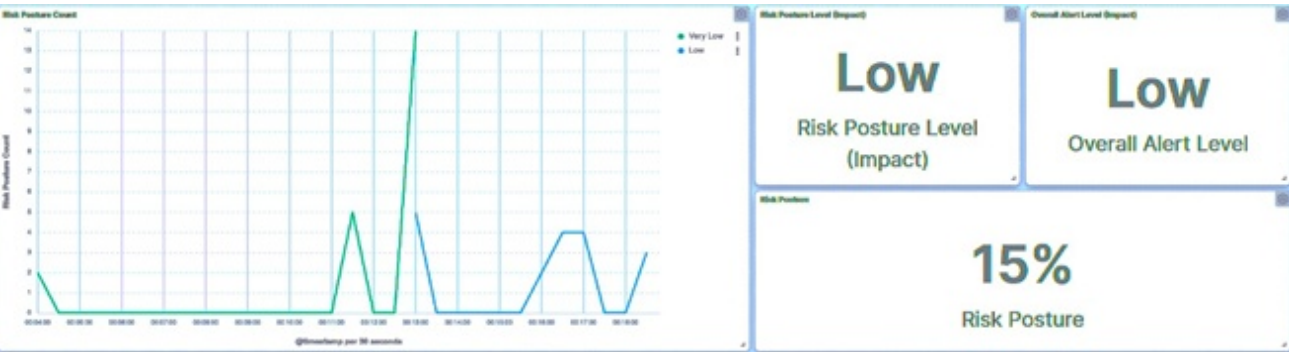


Figure 22. Central agent interface—flooding attack on IED4C

- Payload Injection

The adversary embedded a tampered payload inside a Modbus frame and forwarded it to the trust IEDs. The receiving IED flagged the packet due to a Length Mismatch and immediately blocklisted the compromised HMI (Figure 23). The trust score was then set to Severe.

When this attack hit IED1A, the substation risk posture was escalated to High (Figure 24). An equivalent attack on IED4C likewise caused its trust score to become Severe (Figure 25). Still, the substation risk posture only moved from Very Low to Low because of IED4C’s inferior rank in the system (Figure 26).

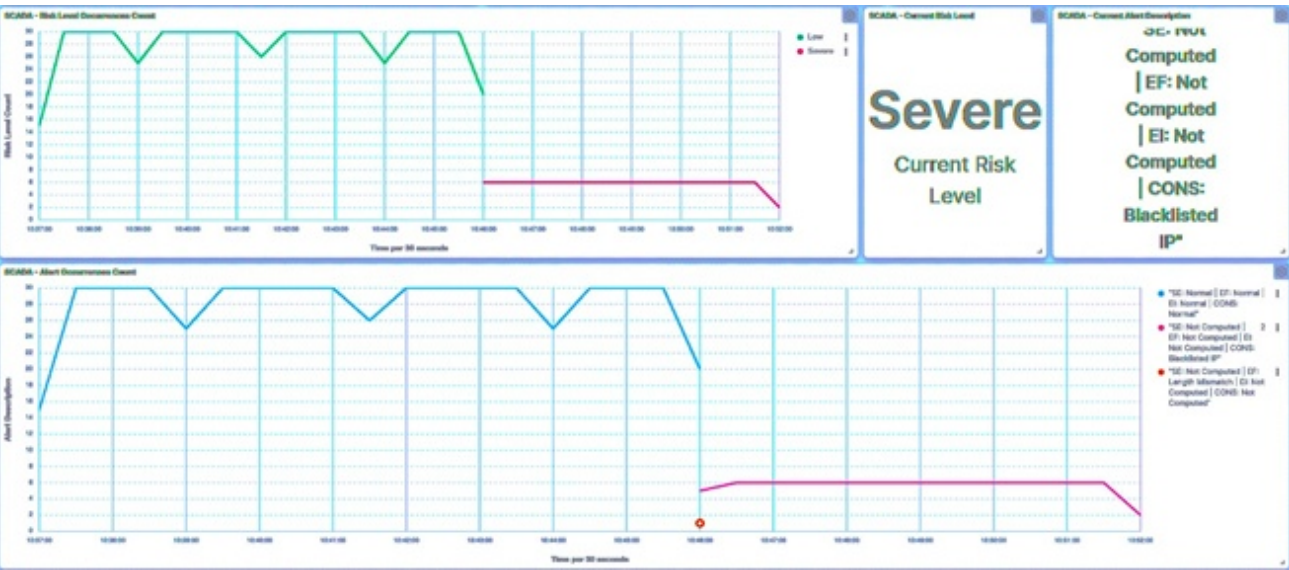


Figure 23. IED1A interface—SCADA payload injection detected

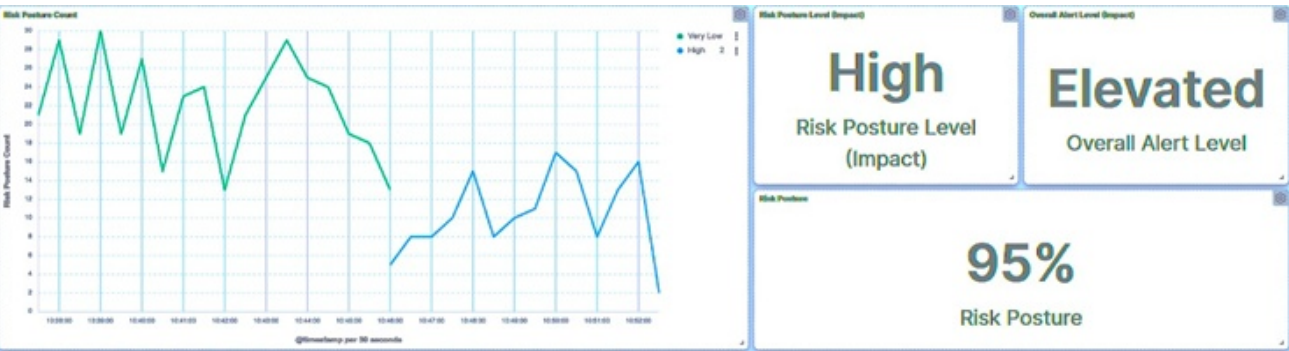


Figure 24. Central agent interface—payload injection against IED1A

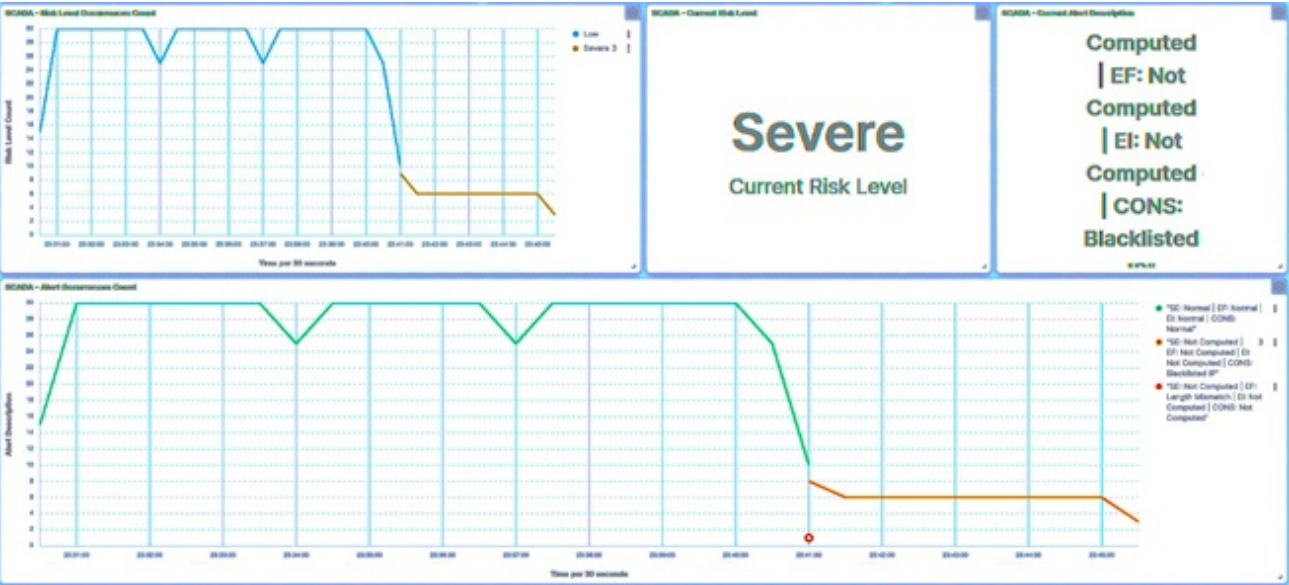


Figure 25. IED4C interface—SCADA payload injection detected

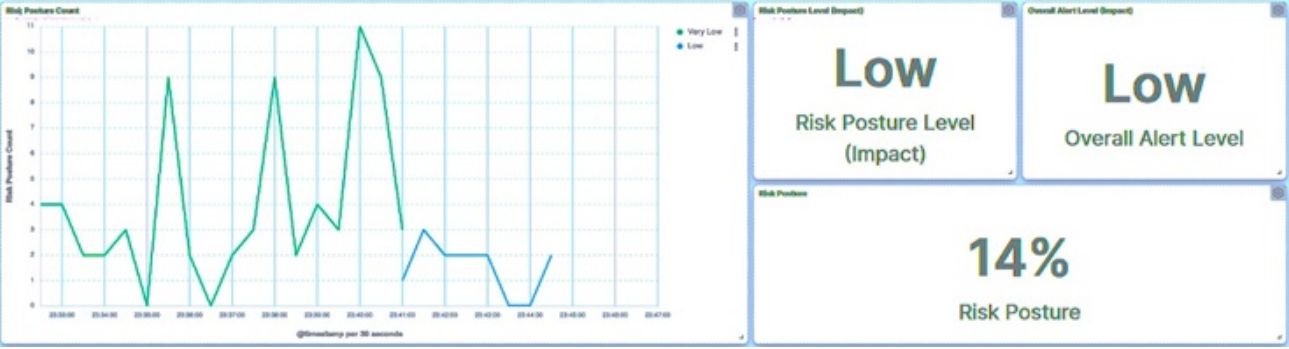


Figure 26. Central agent interface—payload injection against IED4C

Modbus Frame Stacking:

This attack variant consisted of bundling multiple Modbus frames together before sending them to the trusted IEDs. The detection results closely mirrored those seen in the payload injection case, as shown in Figure 27. Upon identifying the malicious sequence, the trust IED immediately set its trust level to Severe.

As a direct outcome of this attack on IED1A, the substation risk posture was raised to High (Figure 28). The identical attack pattern applied to IED4C likewise reduced its trust level to Severe. However, owing to IED4C's lower ranking, the substation's overall risk posture only increased from Very Low to Low (refer to the corresponding central agent view).

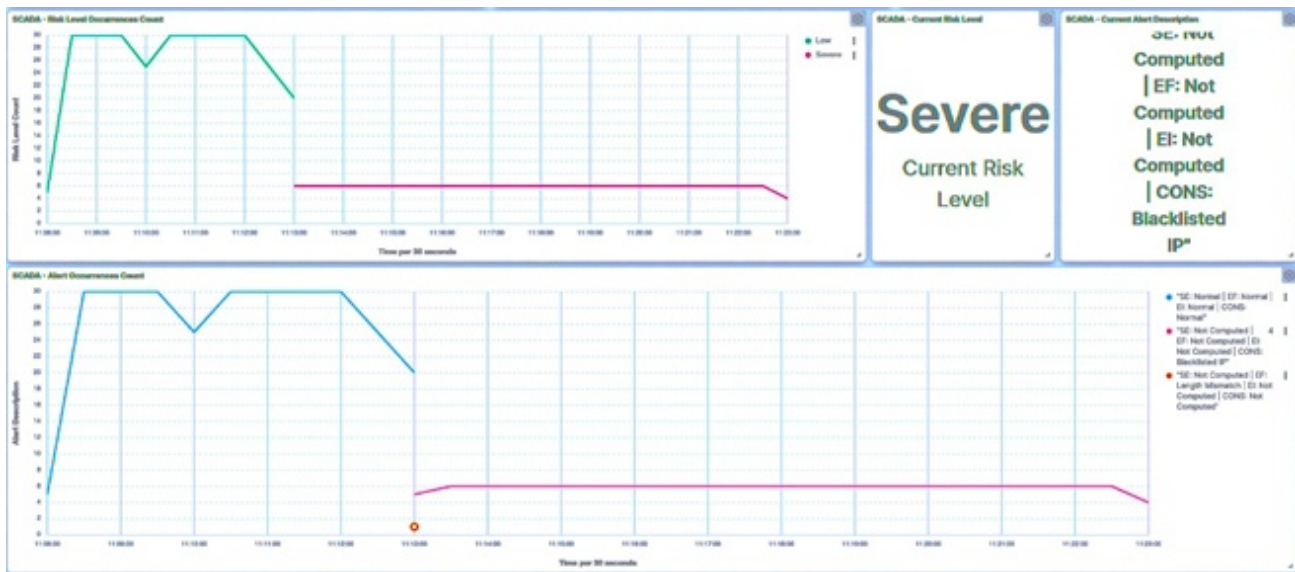


Figure 27. IED1A interface—SCADA frame stacking detected

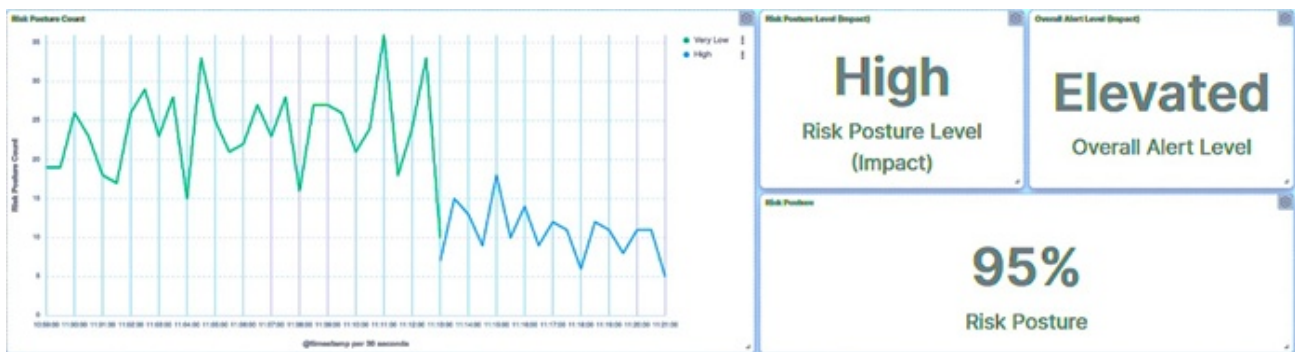


Figure 28. Central agent interface—SCADA frame stacking on IED1A

Replay Attack:

In this scenario, legitimate queries previously captured were resent toward the trust IEDs. The very first replayed packet — a write operation — was immediately recognized and tagged as an Environment Attack – APT Threat (Figure 29). The trust IED responded without delay by downgrading its trust score to Severe.

This event caused the substation risk posture to be upgraded to High when targeting IED1A (Figure 30). The same behavior occurred for IED4C: its trust level shifted to Severe (Figure 31). Yet, because of IED4C's inferior rank, the substation risk posture advanced only from Very Low to Low (Figure 32).

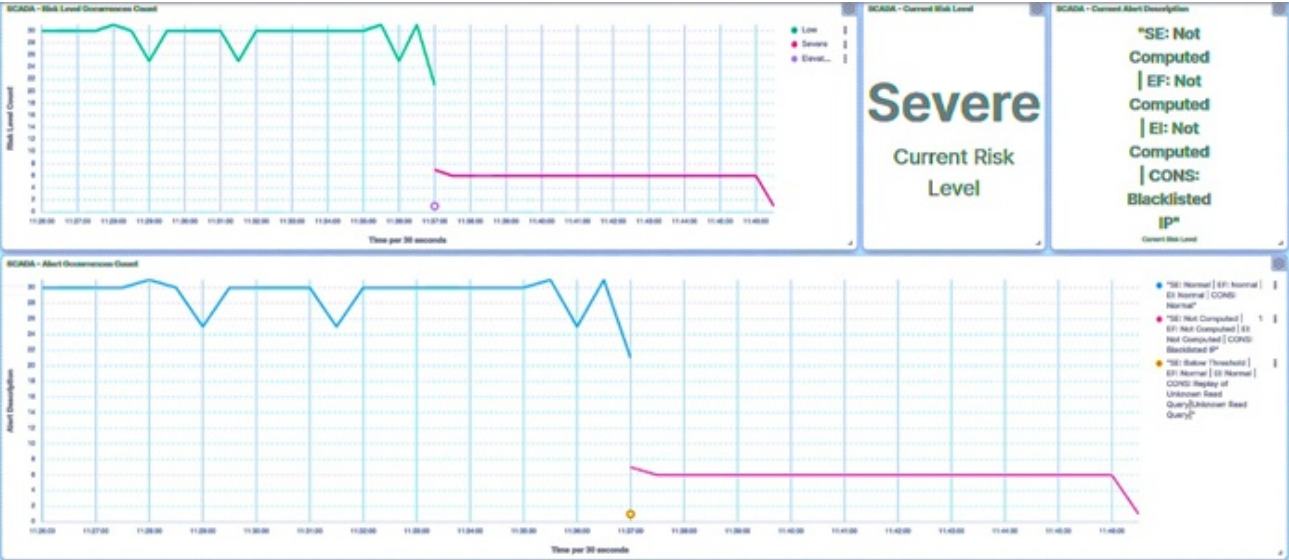


Figure 29. IED1A interface—SCADA replay packet detection



Figure 30. Central agent interface—SCADA replay on IED1A



Figure 31. IED4C interface—SCADA replay packet detection

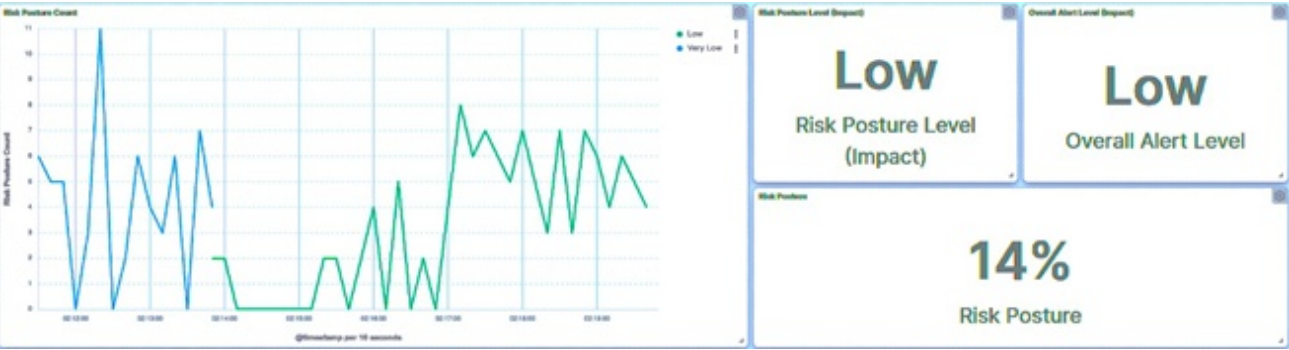


Figure 32. Central agent interface—SCADA replay on IED4C

Brute Force I/O:

The adversary sent sequential write queries to every possible address within the trust IEDs. The opening packet in this barrage was quickly detected and labeled as an Unknown Write Query Attack (Figure 33). In reaction, the trust IED raised its trust level to Severe.

Simultaneously, the substation risk posture was elevated to High, reflecting the attack on IED1A (Figure 34). Applying the same brute-force technique against IED4C produced a parallel effect: trust level set to Severe (Figure 35), while the substation risk posture moved from Very Low to Low due to the device's lower rank (Figure 36). Table 6 provides a condensed summary of outcomes for all compromised SCADA HMI attack types.

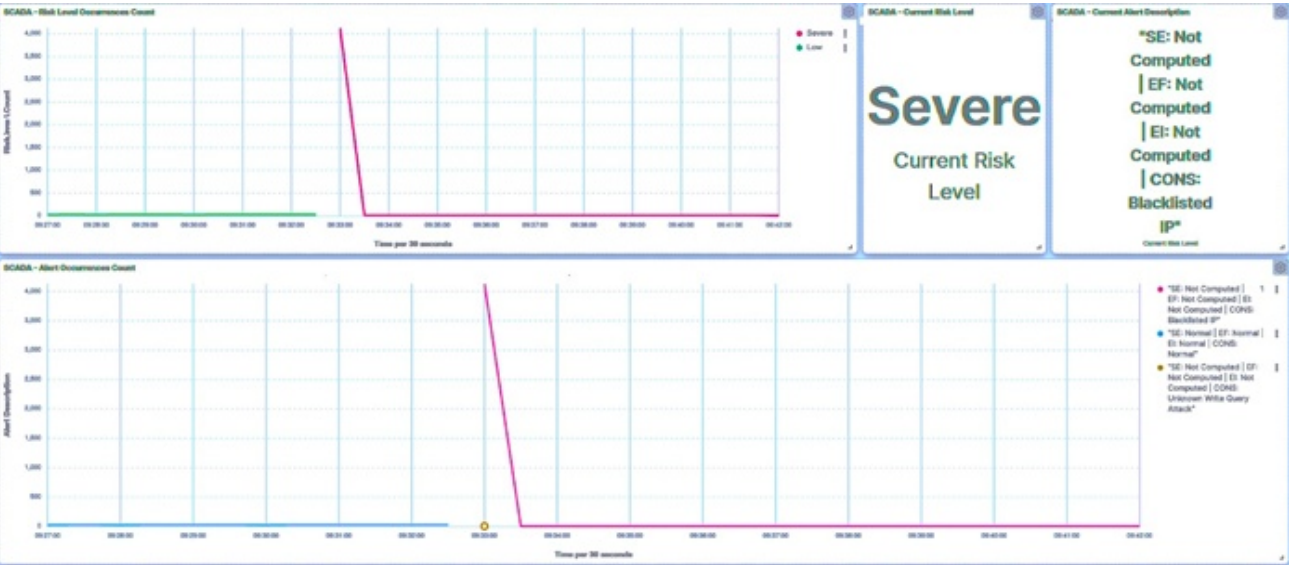


Figure 33. IED1A interface—SCADA brute force I/O detected



Figure 34. Central agent interface—SCADA brute force I/O on IED1A



Figure 35. IED4C interface—SCADA brute force I/O detected



Figure 36. Central agent interface—SCADA brute force I/O on IED4C

Table 6. Summary of results—attacks from compromised SCADA HMI

Trust device	Attack	Alert	Affected component	Device risk level	Risk posture	Outcome

IED1A	Load a malicious payload	Length mismatch	Frequency	Severe	High	Communication with the SCADA HMI is blocked
IED1B	Load a malicious payload	Length mismatch	Frequency	Severe	Low	Communication with the SCADA HMI is blocked
IED1A	Query flooding	Query flooding of known read query	Intensity	Severe	High	Communication with the SCADA HMI is blocked
IED1B	Query flooding	Query flooding of known read query	Intensity	Severe	Low	Communication with the SCADA HMI is blocked
IED1A	Reconnaissance	Unknown read query	Consequence	Severe	High	Communication with the SCADA HMI is blocked
IED1B	Reconnaissance	Unknown read query	Consequence	Severe	Low	Communication with the SCADA HMI is blocked
IED1A	Replay packets	Replay of an unknown read query	Consequence	Severe	High	Communication with the SCADA HMI is blocked
IED1B	Replay packets	Replay of an unknown read query	Consequence	Severe	Low	Communication with the SCADA HMI is blocked
IED1A	Stack Modbus frames	Length mismatch	Frequency	Severe	High	Communication with the SCADA HMI is blocked
IED1B	Stack Modbus frames	Length mismatch	Frequency	Severe	Low	Communication with the SCADA HMI is blocked
IED1A	Write to all coils	Unknown write query attack	Consequence	Severe	High	Communication with the SCADA HMI is blocked
IED1B	Write to all coils	Unknown write query attack	Consequence	Severe	Low	Communication with the SCADA HMI is blocked

Attack from Compromised IED

Denial of Service:

Similar to the DoS patterns observed from a compromised SCADA source, the malicious IED1B responded by concatenating multiple Modbus frames into one oversized reply. Upon receiving this malformed response, the trust SCADA HMI analyzed it, detected the anomaly, and raised a Length Mismatch alert (**Figure 37**). This triggered immediate blocklisting of IED1B and assignment of a Severe risk classification to the device.

The central agent, acting on this information, consistently classified the substation risk posture as High (**Figure 38**).



Figure 37. SCADA interface—IED1B stacking detected



Figure 38. Central agent interface—IED1B stacking

Length Manipulation:

Here, the attacker deliberately tampered with the Modbus length field, resulting in a substantial mismatch between the declared length and the actual payload size. The trust SCADA HMI promptly identified this inconsistency and flagged it as Length Mismatch (Figure 39). IED1B was then marked as Severe risk, prompting the central agent to upgrade the substation risk posture to High (Figure 40).



Figure 39. SCADA interface—IED1B length manipulation detected



Figure 40. Central agent interface—IED1B length manipulation

Payload Injection:

The compromised IED1B inserted extraneous bytes into the Modbus payload before transmission. This caused a clear discrepancy between the stated length and actual payload size, which the trust SCADA HMI detected and labeled Length Mismatch (Figure 41). IED1B's trust status was set to Severe immediately, prompting the central agent to raise the substation risk posture to High (Figure 42).



Figure 41. Trust SCADA HMI interface—IED1B payload injection detected

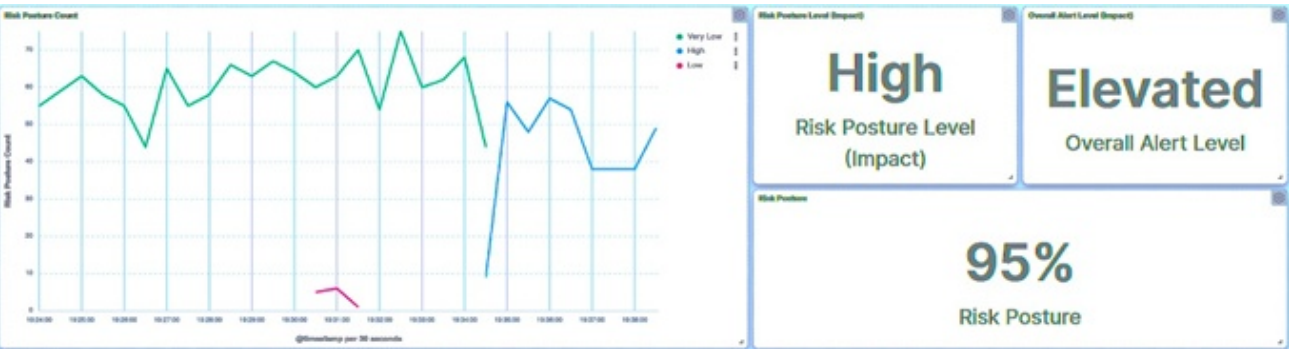


Figure 42. Central agent interface—IED1B payload injection

False Data Injection:

By altering the data content within a Modbus packet, the attacker attempted to feed incorrect values into the system. The trust SCADA HMI recognized the suspicious command and generated an Unknown Read Command alert (Figure 43). IED1B was classified as Severe risk, and the central agent—synchronized with this trust update—adjusted the substation risk posture to High (Figure 44).



In this attack, IED1B deliberately delayed its responses, inserting intentional lags of 30 seconds to 1 minute. Such timing deviations disrupted normal communication cadence and could impair system responsiveness.

Spoofed Reporting:

As shown in **Figure 45**, the trust SCADA HMI accepts the replayed message and detects no irregularities or deviations in the content. Consequently, it does not mark the packet as malicious; instead, it treats it as routine, expected traffic.

Because the trust SCADA HMI does not raise an alert, the central agent receives no indication of a compromise. This leaves the substation risk posture unchanged at a low level, as illustrated in **Figure 46**. The attack succeeds in blending

seamlessly with legitimate communication flows, preventing any escalation of the perceived risk.

Examination of the timestamped logs reveals a distinct event window between 10:20 p.m. and 10:22 p.m. During this interval, the SCADA HMI assigned an Elevated status to IED1B. This change in IED1B's status directly led to the substation's overall risk posture being upgraded to High during the same period.

Within that time frame, IED1B transmitted a voltage reading of 0, indicating that a circuit breaker (CB) had opened and severed power from one of the substation's main sources.

The trust's SCADA HMI recognized the resulting voltage drop in a critical circuit as an irregular condition. This prompted classification of IED1B as Elevated, which, in turn, drove the substation risk posture to High, reflecting a credible threat to normal substation operation. A consolidated overview of outcomes for compromised IED attacks appears in Table 7.



Figure 45. SCADA interface—IED1B baseline replay detected

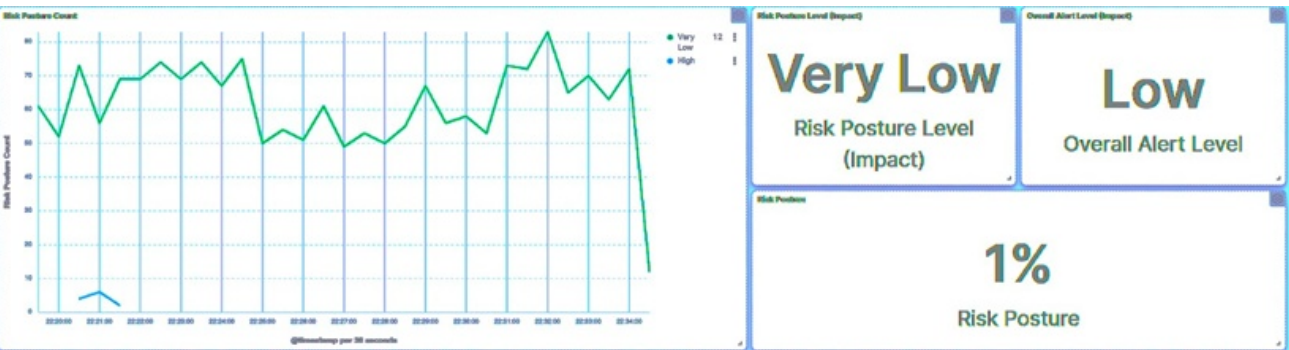


Figure 46. Central agent interface—IED1B baseline replay

Table 7. Summary of results—attacks from compromised IED

Trust device	Attack	Alert	Affected component	Device risk level	Risk posture	Outcome
IED1A	Baseline replay	None	None	Low	Very Low	None

IED1B	Baseline replay	None	None	Low	Very Low	None
IED1A	Delay response	None	None	Low	Very Low	None
IED1B	Delay response	None	None	Low	Very Low	None
IED1A	False data injection	Unknown read query	Consequence	Severe	High	Communication with IED is blocked
IED1B	False data injection	Unknown read query	Consequence	Severe	Low	Communication with IED is blocked
IED1A	Length manipulation	Length mismatch	Intensity	Severe	High	Communication with IED is blocked
IED1B	Length manipulation	Length mismatch	Intensity	Severe	Low	Communication with IED is blocked
IED1A	Load a malicious payload	Length mismatch	Frequency	Severe	High	Communication with IED is blocked
IED1B	Load a malicious payload	Length mismatch	Frequency	Severe	Low	Communication with the SCADA HMI is blocked
IED1A	Stack Modbus frames	Length mismatch	Frequency	Severe	High	Communication with IED is blocked
IED1B	Stack Modbus frames	Length mismatch	Frequency	Severe	Low	Communication with IED is blocked

Transferability

Scenario 1—Normal/Good Behavior

Figure 47 presents a chronological trace—captured from the central agent’s perspective—of the transferability sequence for IED4C. Each entry in the diagram represents a logged milestone in the handover procedure.

The process begins when IED4C sends a disconnection request to the central agent. This initial step is recorded as Start Disconnect, signaling the formal launch of device replacement.

Next, IED4C transmits its current trust scores to the central agent. At the same time, the trust SCADA HMI forwards its own trust assessments of IED4C. Both sets of values are archived by the central agent under the event Scores Shared.

The central agent then instructs IED4C to carry out the disconnection. Once disconnection is confirmed successful, the central agent logs the milestone as Disconnect Complete.

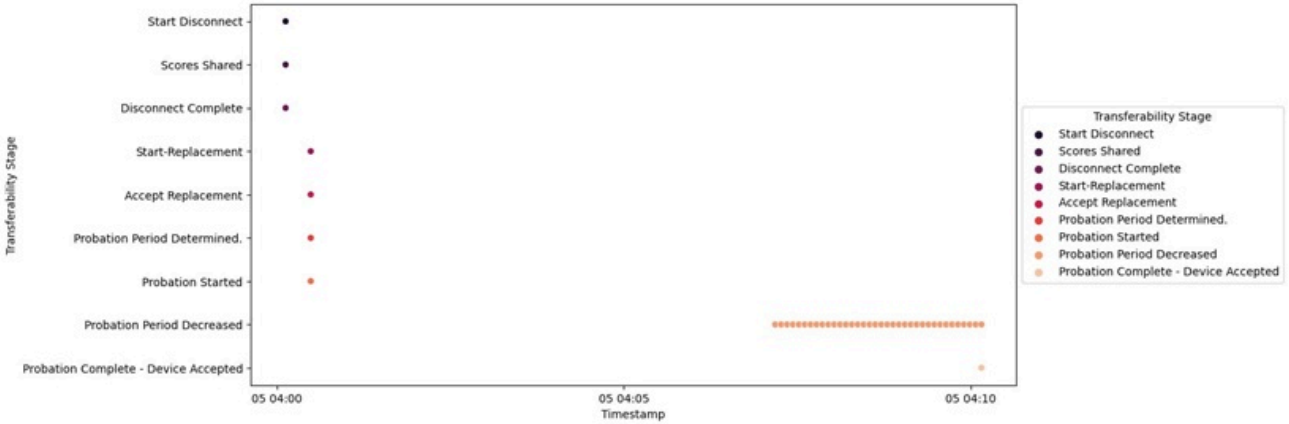


Figure 47. IED4C Scenario 1—transferability sequence

Removal of IED4C from the active network temporarily suspends continuous risk scoring by the trust SCADA HMI, creating a brief gap in the trust score timeline (Figure 48).

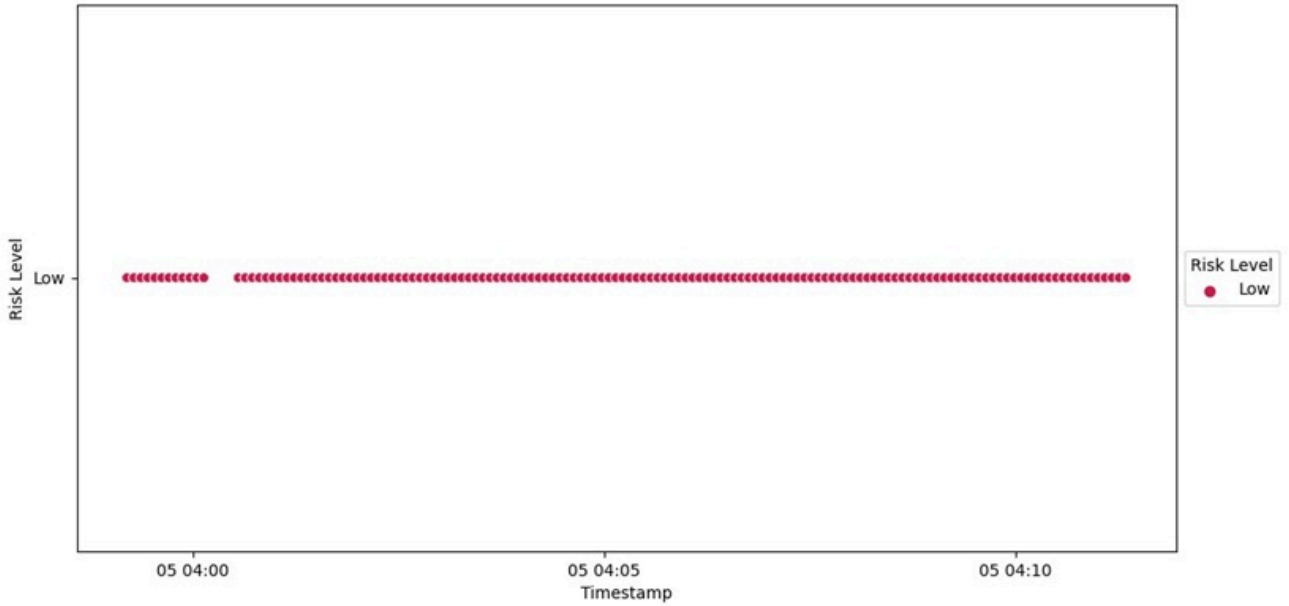


Figure 48. IED4C Scenario 1—trust SCADA HMI trust scores

Returning to the event log in Figure 47, the arrival of a replacement device is marked by the Start Replacement event. At this point, the incoming trust IED submits its initial trust scores to the central agent. The central agent compares these fresh scores against the historical trust profile of the departing IED4C to determine suitability for onboarding.

The subsequent Probation Period Determined event reflects the central agent's calculation of both the total probation duration and the critical probation threshold point. Probation started, then indicates the beginning of active monitoring of the replacement device.

Throughout the probation interval — and especially during the later consideration sub-phase — every instance of compliant, expected behavior by the new IED earns a reduction in remaining probation time. These incremental improvements are logged as Probation Period Decreased.

The process concludes successfully with the Probation Complete — Device Accept event, confirming full integration of the replacement IED into the substation network. Throughout the entire sequence, the substation risk posture stays stable at Very Low (Figure 49), demonstrating the robustness of the transferability mechanism. Identical results were observed for both IED4C and IED1A, confirming that the procedure performs reliably irrespective of the original device's rank.

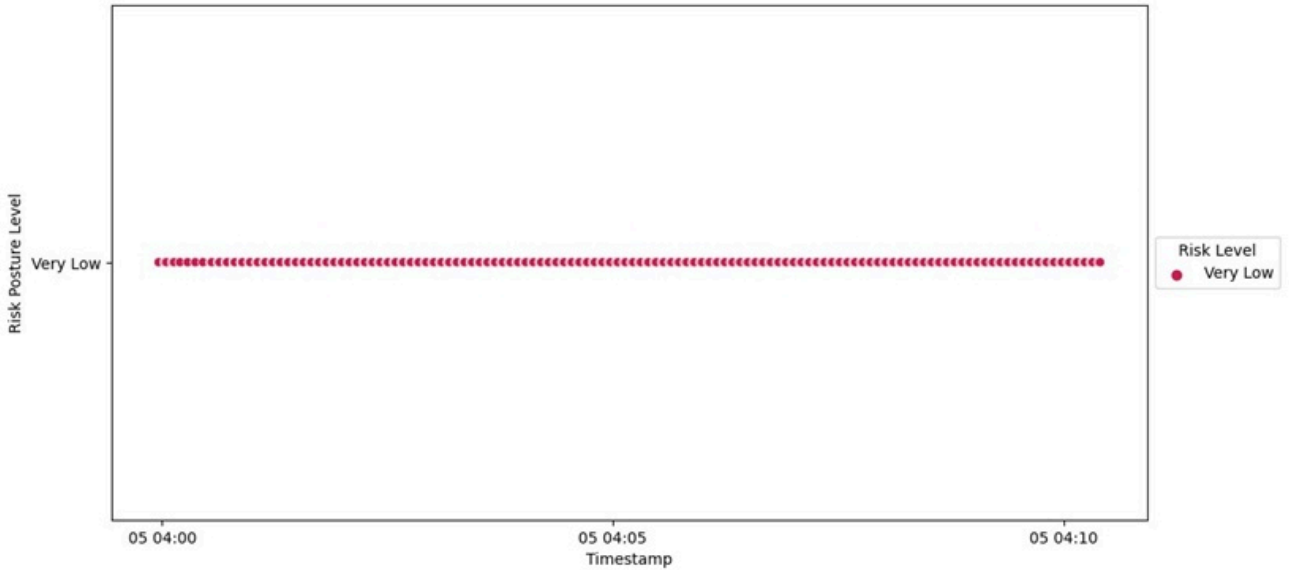


Figure 49. IED4C Scenario 1—Substation Risk Posture

Scenario 2—Malicious behavior following probation acceptance

As shown in Figure 50, the sequence of steps from Start Disconnect through to Probation Started follows the same pattern observed in Scenario 1. All phases—including disconnection of the outgoing device and initiation of probation for the incoming one—remain unchanged up to and including the Probation Started milestone. During the disconnection phase, a temporary break in communication with the trust SCADA HMI occurs, identical to the interruption visible in Figure 51 for Scenario 2.

Once Probation Started is logged, the replacement trust IED immediately begins exhibiting harmful behavior by launching a Modbus frame-stacking attack against the trust's SCADA HMI.

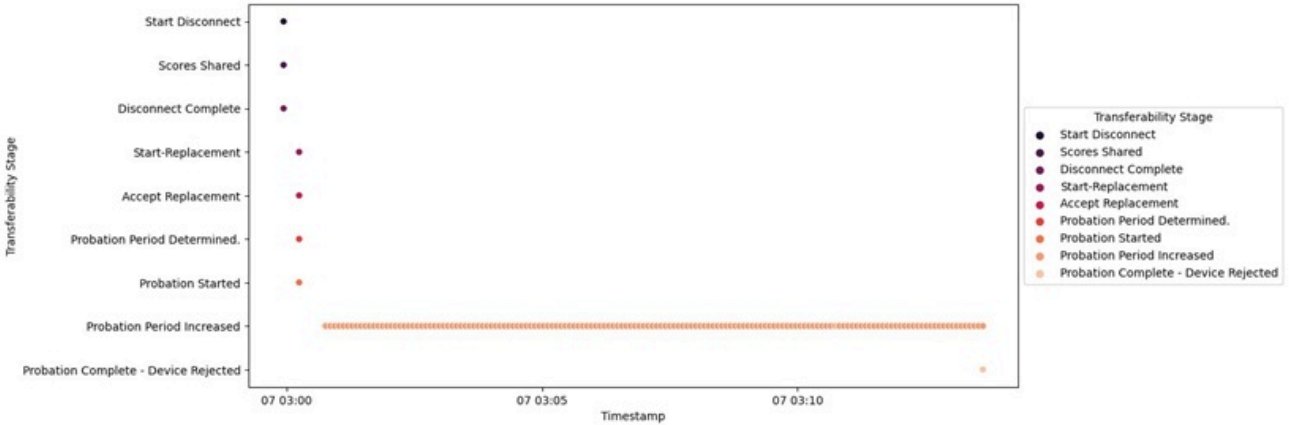


Figure 50. IED4C Scenario 2—transferability sequence

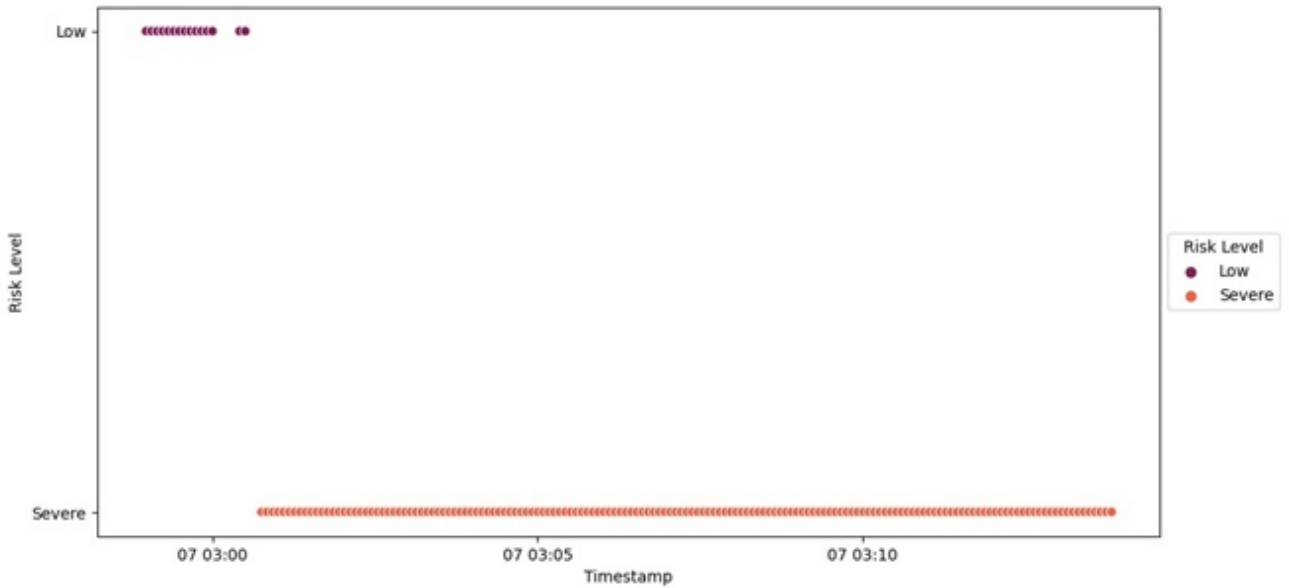


Figure 51. IED4C Scenario 2—trust SCADA HMI trust scores

The trust SCADA HMI quickly identified the frame stacking attack, blocklisted the offending IED4C (Figure 52), and raised its risk classification from Low to Severe. In response, the central agent added penalty probation points to the active probation timer (Figure 50), which shifted the substation's overall risk posture from Very Low to Low (Figure 53). This relatively modest increase reflects IED4C's lower position in the device hierarchy.



Figure 52. IED4C Scenario 2—SCADA dashboard view

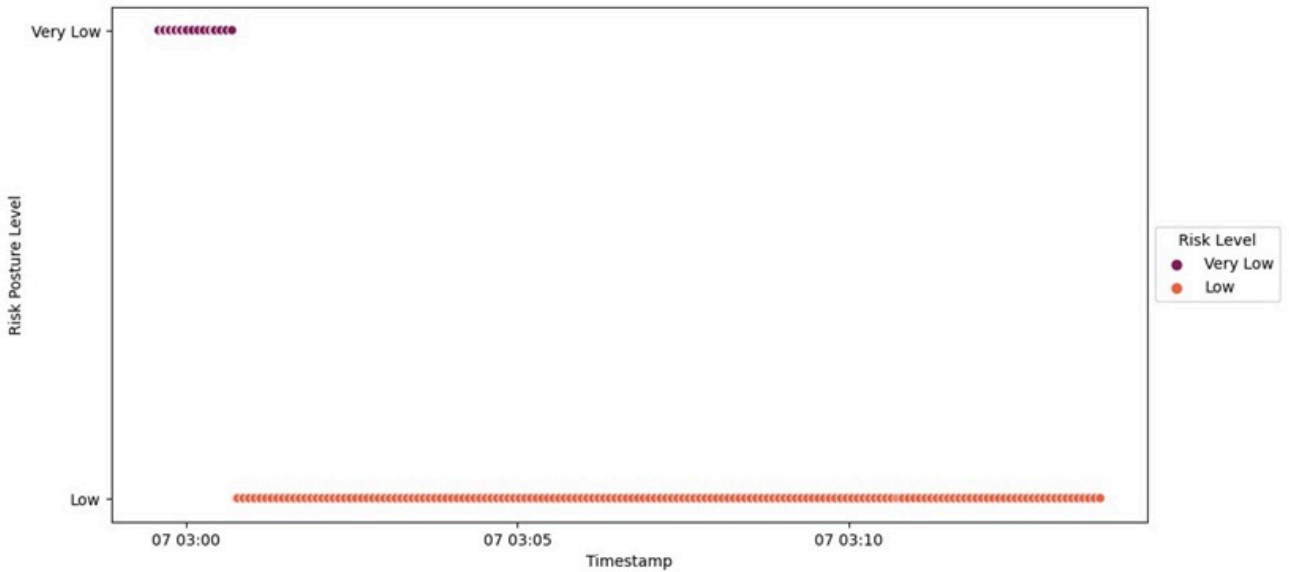


Figure 53. IED4C Scenario 2—substation risk posture

Moreover, because the replacement IED continued to refrain from sending any benign (non-malicious) messages, the central agent repeatedly extended the probation duration, as recorded in Figure 50. As a result, the consideration sub-phase was effectively bypassed. Eventually, the original probation window expired before the extended (penalized) period could be completed, triggering the Probation Complete—Device Rejected event and permanently rejecting the incoming device.

The identical sequence unfolded when the same attack strategy was applied during the replacement of IED1A (Figure 54). The trust SCADA HMI raised IED1A's risk level from Low to Severe (Figure 55). However, owing to IED1A's higher hierarchical rank, the substation risk posture escalated sharply from Very Low to High (Figure 56).

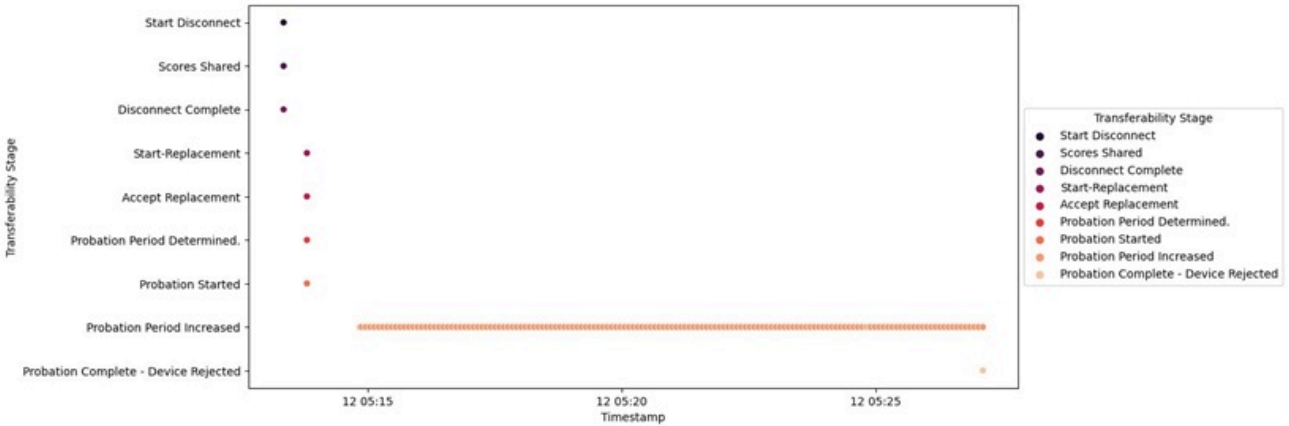


Figure 54. IED1A Scenario 2—transferability sequence

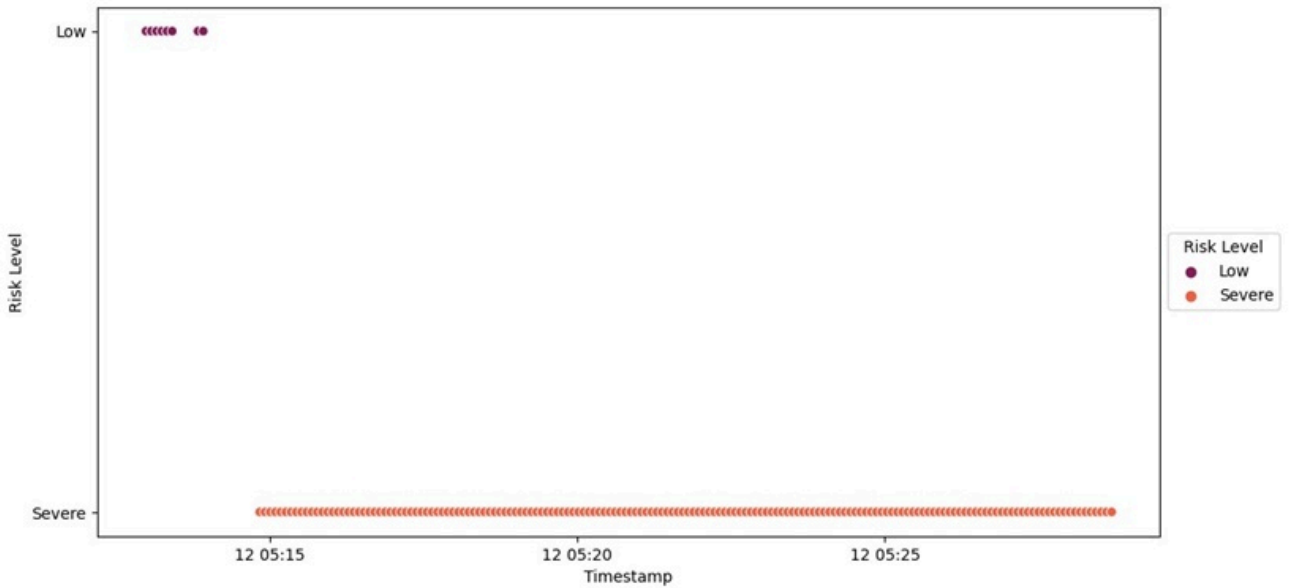


Figure 55. IED1A Scenario 2—trust SCADA HMI trust scores

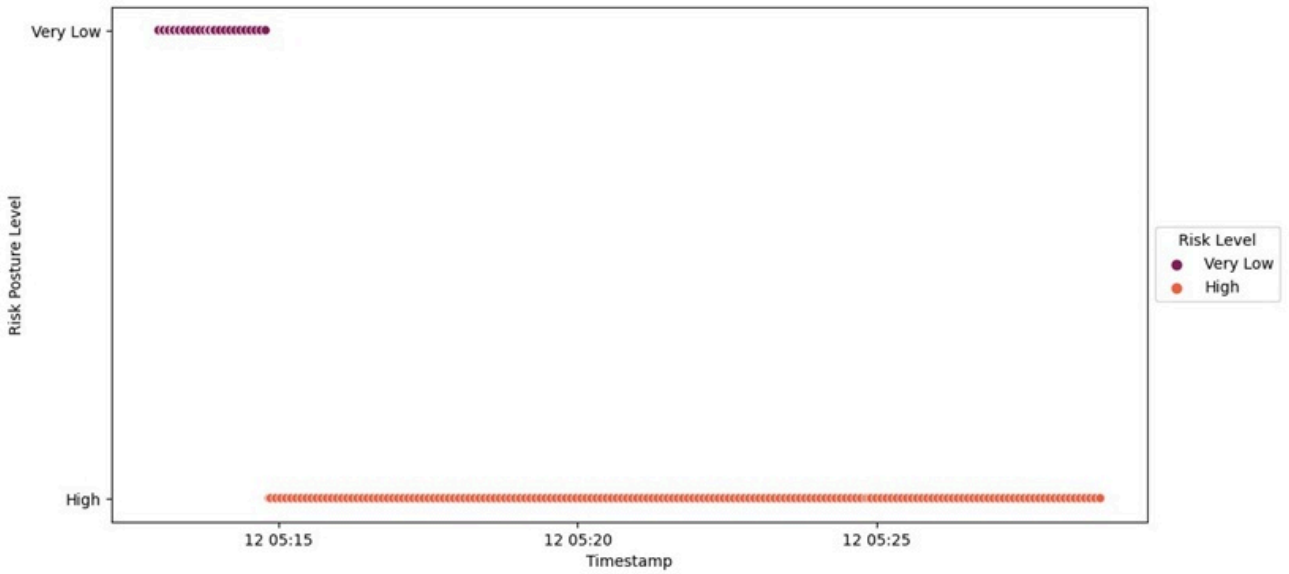


Figure 56. IED1A Scenario 2—Substation Risk Posture

Scenario 3—Malicious behavior emerging after consideration period

Figure 57 shows that the initial chain of events in Scenario 3—from Start Disconnect through Probation Started—replicates the flow seen in Scenario 1. A comparable interruption in trust SCADA-HMI communication occurs during device disconnection, as shown in Figure 58.

Throughout the consideration phase, the newly introduced trust IED exhibited fully compliant behavior, which is reflected in multiple Probation Period Decreased entries. Only after the consideration period concluded did the replacement IED switch to malicious activity, performing a Modbus frame-stacking attack against the trust SCADA HMI.

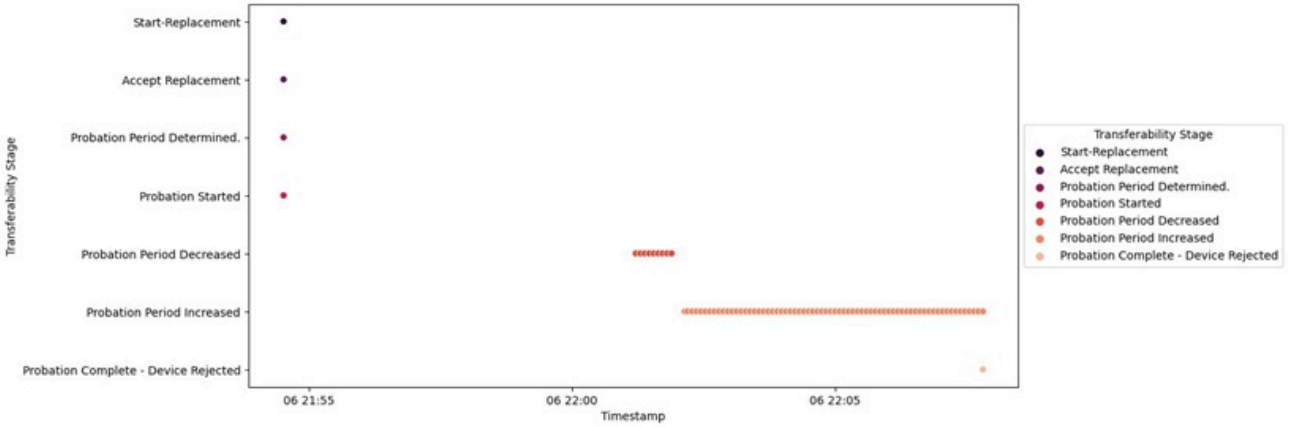


Figure 57. IED4C Scenario 3—transferability sequence

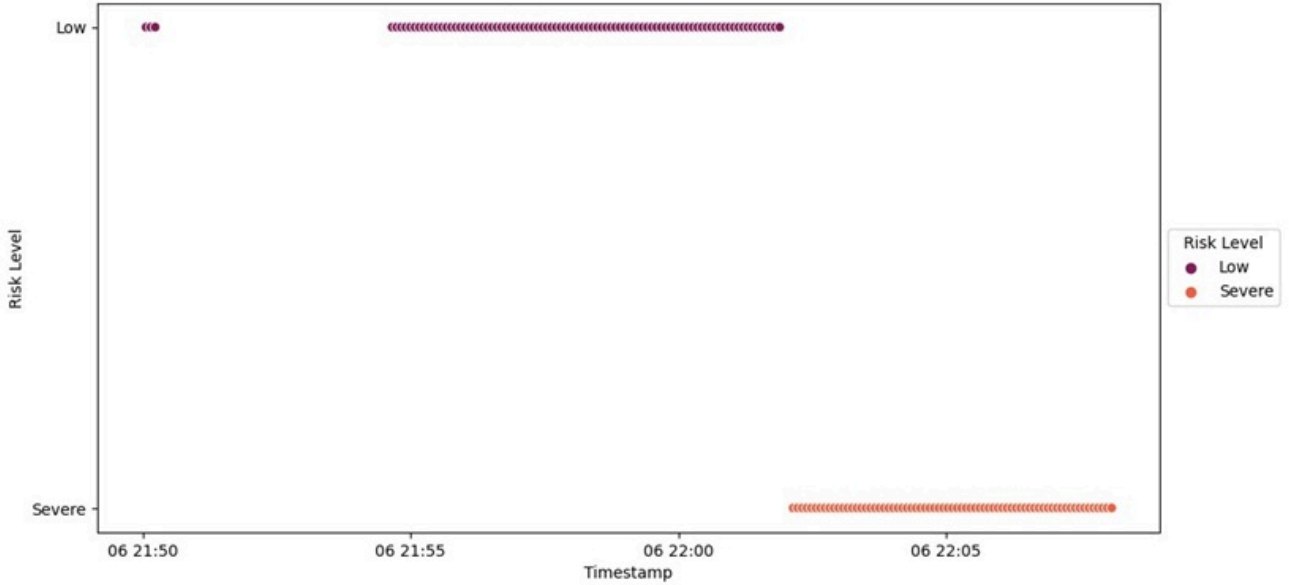


Figure 58. IED4C Scenario 3—trust SCADA HMI trust scores

The trust SCADA HMI detected this late-onset attack without delay, blocklisted the device (Figure 59), and upgraded IED4C's risk level from Low to Severe. The key difference between Scenario 2 and Scenario 3 is the length of time during which the replacement IED's risk scores were observed: in Scenario 3, monitoring continued over a noticeably longer window.



Figure 59. IED4C Scenario 3—SCADA dashboard view

In response to the detected misbehavior, the central agent incremented probation points, thereby extending the probation interval (Figure 57). This extension produced a substation risk posture adjustment from Very Low to Low (Figure 60). In contrast to Scenario 2, the central agent's log in Scenario 3 records a sustained sequence of low-risk posture states over an extended duration.

Consistent with the ongoing absence of legitimate (non-malicious) traffic from the replacement IED, the probation timer continued to lengthen (Figure 57). This prolonged penalty phase ultimately caused the original probation deadline to pass before the adjusted period was completed, resulting in the Probation Complete—Device Rejected event and the rejection of the incoming trust IED.

When the same delayed-misbehavior pattern was tested with IED1A, the outcome followed a parallel course—except that the substation risk posture advanced from Very Low to High due to the higher rank of IED1A (Figure 61).

Scenario 4—Unsatisfactory trust scores

The sequence of events from Start Disconnect to Disconnection in Scenario 4, as illustrated in Figure 62, follows the same pattern as the previous scenarios. During the Start Replacement event, however, the new trust IED sends its poor trust scores to the central agent. The central agent then uses these incoming scores, together with the trust scores of IED4C, to decide whether the new trust IED should be allowed to connect to the network. After completing the calculation, the trust scores of the new trust IED are found to be below the acceptable threshold. As a result, this assessment triggers a Device Rejected event.

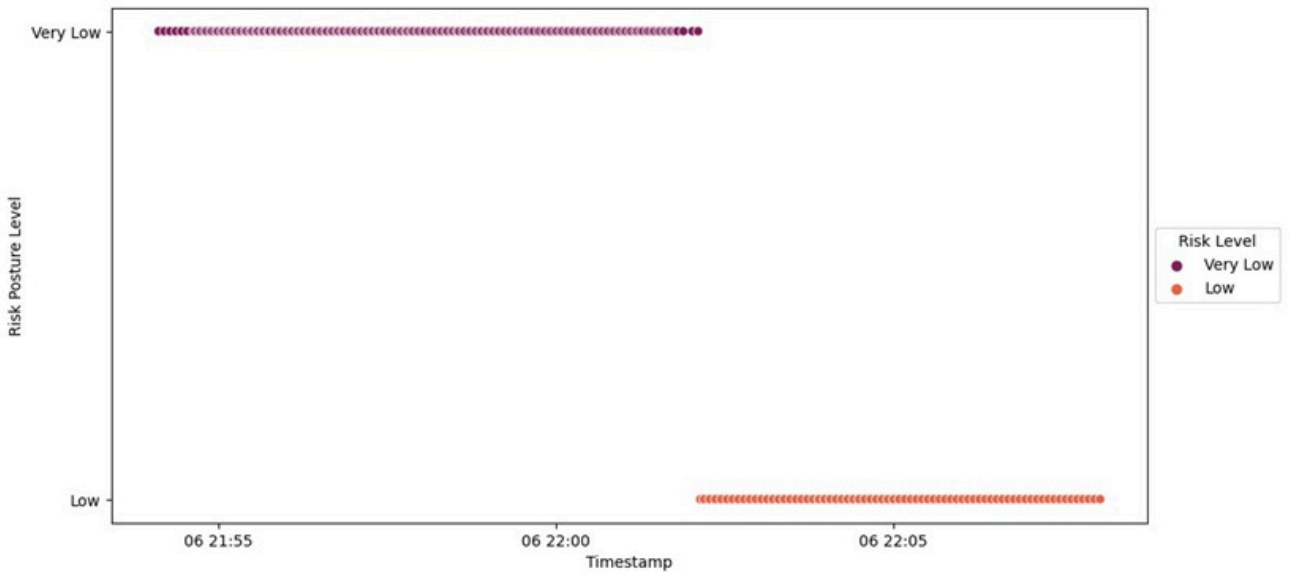


Figure 60. IED4C Scenario 3—substation's risk posture.

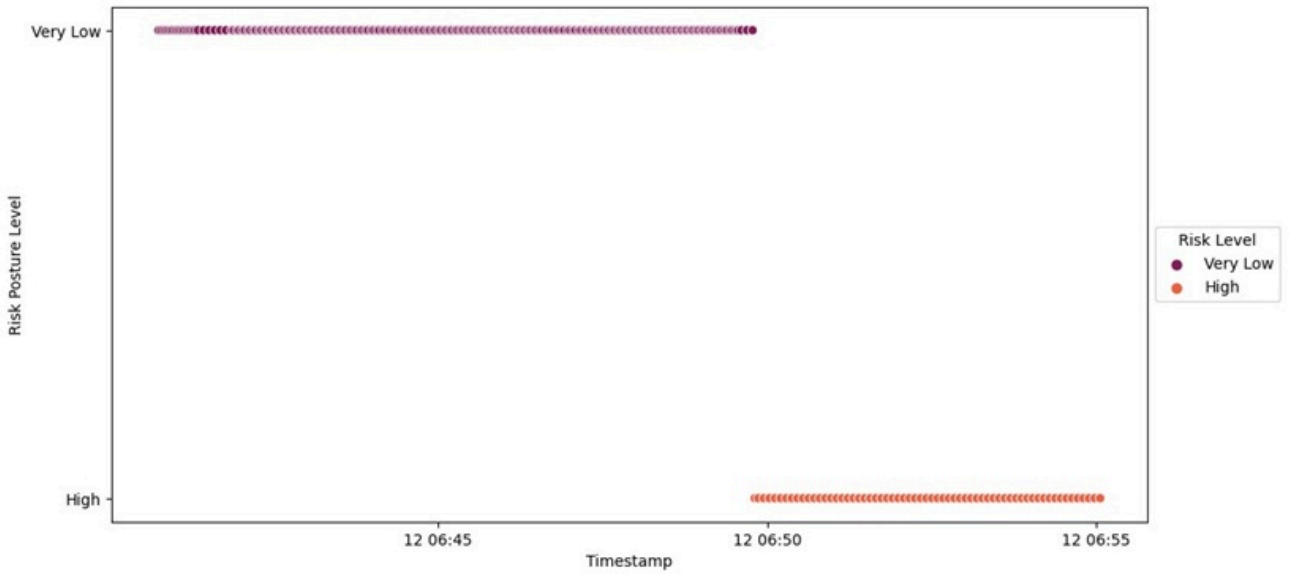


Figure 61. IED1A Scenario 3—substation's risk posture.

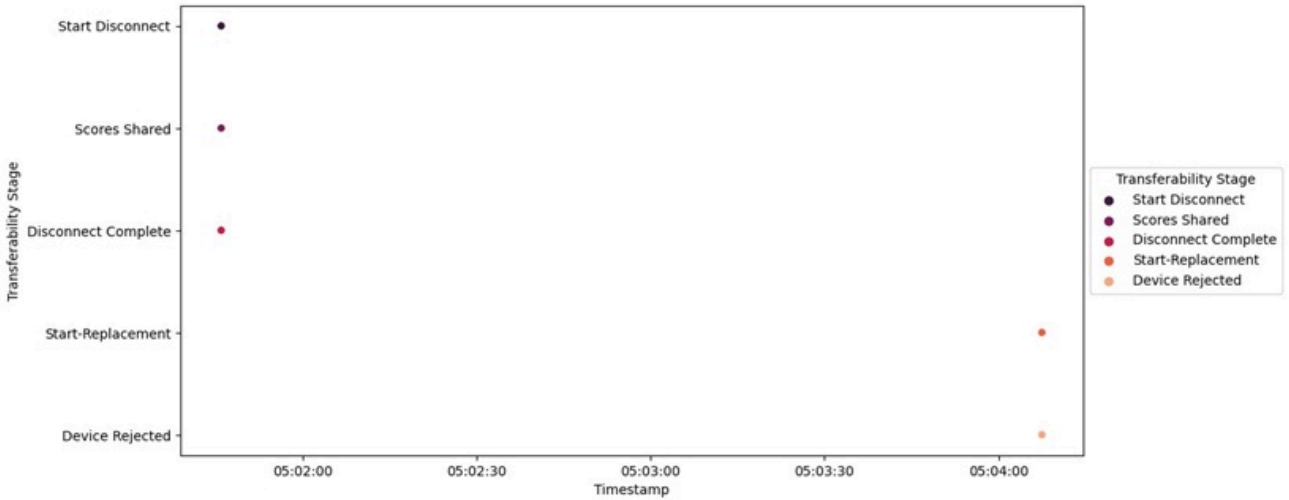


Figure 62. IED4C Scenario 4—transferability process.

Challenges

Although [Table 8](#) demonstrates that our approach offers significant advantages over other models, several challenges remain. Specifically, the model has pinpointed two critical attack types that threaten its overall performance: the delayed response attack and the baseline attack. If successful, the delayed response attack could severely impair substation operations due to the strict timing requirements of these systems. While this attack type has rarely been documented in real-world public cases, its potentially severe consequences demand attention.

Table 8. Comparison with other trust models.

Research	Trust model	Detect malicious payloads	Response latency	Risk posture model	Transferability
Trust framework	Yes	Yes	< 10 ms	Yes	Yes

Fadul <i>et al.</i> [16]	Yes	No	> 20 ms	No	No
Wang <i>et al.</i> [18]	Yes	Yes	> 20 ms	No	No
Qureshi <i>et al.</i> [17]	Yes	No	< 10 ms	No	No

Mitigating the delayed-response attack would require adding a strict time-constraint parameter to the model. Implementing this is difficult, however, because of existing infrastructure limitations—such as those imposed by virtual machine (VM) environments—which can cause processing delays and jitter, potentially resulting in false positives. Resolving this issue remains a significant challenge and is identified as an important direction for future research and development.

The baseline attack, in which an attacker replays legitimate requests without modifying their content, poses another serious risk. This technique can be used to manipulate particular scenarios, for example, by blocking a circuit breaker from receiving a legitimate close command. Defending against this attack requires strengthening the trust model's ability to observe the IED's surrounding context more thoroughly, including adding logic to identify unusual patterns and counteract malicious actions.

Throughout the probation period, the SCADA system applies blocklisting when a malicious query is detected for more than half of the allocated time. Once the blacklist period ends, normal functioning resumes. Nevertheless, the current model leaves some situations unresolved. Examples include cases where the replaced device exhibits low trust scores or when a trusted device delivers correct responses while simultaneously disseminating false information about other devices. Additionally, the distinctive characteristics of the substation setting create further difficulties. Conventional testing methods are often impractical due to the risk of network flooding. Moreover, constraints in the current testbed limit the ability to evaluate certain sophisticated scenarios fully. For example, identifying baseline replay attacks remains problematic in the existing environment.

It must be recognized that fully and accurately capturing every possible scenario in an actual substation environment is highly complex. Therefore, continued research and improvement are necessary to overcome these shortcomings and strengthen the model's robustness and real-world applicability.

Conclusion

We presented a comprehensive trust framework tailored for substations, comprising three main elements: a trust model for detecting protocol-based attacks, a risk posture model to evaluate the substation's response to attacks, and a trust transferability model to oversee the integration of new devices. Evaluation of the framework in a Docker-based multi-agent environment confirmed its effectiveness against a range of attacks, though it revealed susceptibility to baseline replay and delayed-response attacks.

We also examined various trust transferability scenarios, including both standard and compromised device replacements, and confirmed the framework's ability to identify malicious conduct by trust devices successfully.

Nevertheless, some issues—such as handling unsatisfactory trust scores or preventing trusted devices from propagating false information—still require additional attention in future studies.

Acknowledgements

None

Conflict of interest

None

Financial support

The authors acknowledge the funding from the Atlantic Canada Opportunities Agency

(ACOA) through the Atlantic Innovation Fund (AIF) project #212420 and a grant from the Natural Sciences and Engineering Research Council of Canada—NSERC (Grant# RGPIN 231074) to Dr.

Ali Ghorbani.

Ethics statement

None

Received: 22 Sep 2023

Revised: 29 Oct 2023

Accepted: 07 Dec 2023

Published online: 20 March 2024

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

References

- Xu Z, Salehi Shahraki A, Rudolph C. Blockchain-Based Malicious Behaviour Management Scheme for Smart Grids. *Smart Cities*. 2023;6:3005-31.
- The Real Story of Stuxnet [Internet]. *IEEE Spectrum*; 2022 [cited 2022 Dec 11]. Available from: <https://spectrum.ieee.org/the-real-story-of-stuxnet>
- Pipedream: Chernovite's Emerging Malware Targeting Industrial Control Systems [Internet]. *Dragos*; 2023 [cited 2023 Feb 15]. Available from: https://hub.dragos.com/hubfs/116-Whitepapers/Dragos_ChernoviteWP_v2b.pdf?hsLang=en
- COSMICENERGY: New OT Malware Possibly Related to Russian Emergency Response Exercises [Internet]. *Mandiant*; 2022. Available from: <https://www.mandiant.com/resources/blog/cosmicenergy-ot-malware-russian-response>
- Recommendations Following the Colonial Pipeline Cyber Attack [Internet]. *Dragos*; 2023. Available from: <https://www.dragos.com/blog/industry-news/recommendations-following-the-colonial-pipeline-cyber-attack/>
- Lei H, Singh C, Sprintson A. Reliability modeling and analysis of IEC 61850 based substation protection systems. *IEEE Trans Smart Grid*. 2014;5(5):2194-202.

Requirements for IDS in Substations [Internet]. Electrical Engineering Portal; 2022 [cited 2022 Dec 22]. Available from: <https://electrical-engineering-portal.com/improving-cybersecurity-substations-intrusion-detection#requirements-ids-substations>

A Researcher's Perspective on the North American Protective Relay Marketplace [Internet]. T&D World; 2023. Available from: <https://www.tdworld.com/test-and-measurement/article/20972654/a-researchers-perspective-on-the-north-american-protective-relay-marketplace/>

The Worldwide Study of the Protective Relay Marketplace in Electric Utilities: 2019–2022 . Newton-Evans; 2023. Available from: <https://www.newton-evans.com/product/the-worldwide-study-of-the-protective-relay-marketplace-in-electric-utilities-2019-2022/>

Cook K. Trust in Society. Vol. 2, Russell Sage Foundation Series on Trust. New York: Russell Sage Foundation; 2003. 432 p.

Gambetta D. Can we trust trust? In: Gambetta D, editor. Trust: Making and Breaking Cooperative Relations. Oxford: Blackwell; 2000. p. 213-37.

Rousseau DM, Sitkin SB, Burt RS, Camerer C. Not so different after all: A cross-discipline view of trust. *Acad Manag Rev*. 1998;23(3):393-404.

Boakye-Boateng K, Ghorbani AA, Lashkari AH. A novel trust model in detecting final-phase attacks in substations. In: 2021 18th International Conference on Privacy, Security and Trust (PST); 2021 Dec 13-15; Auckland, New Zealand. IEEE; 2021. p. 1-11.

Boakye-Boateng K, Ghorbani AA, Lashkari AH. A Trust-Influenced Smart Grid: A Survey and a Proposal. *J Sens Actuator Netw*. 2022;11(3):34.

Borowski JF, Hopkinson KM, Humphries JW, Borghetti BJ. Reputation-based trust for a cooperative agent-based backup protection scheme. *IEEE Trans Smart Grid*. 2011;2(2):287-301.

Fadul JE, Hopkinson KM, Andel TR, Sheffield CA. A trust-management toolkit for smart-grid protection systems. *IEEE Trans Power Deliv*. 2013;29(4):1768-79.

Qureshi KN, ul Islam MN, Jeon G. A trust evaluation model for secure data aggregation in smart grids infrastructures for smart cities. *J Ambient Intell Smart Environ*. 2021;13(3):235-52.

Wang J, Zhang Z, Wang M. A Trust Management Method against Abnormal Behavior of Industrial Control Networks under Active Defense Architecture. *IEEE Trans Netw Serv Manag*. 2022;19(3):2549-72.

Boakye-Boateng K, Ghorbani AA, Lashkari A. Securing Substations with Trust, Risk Posture, and Multi-Agent Systems: A Comprehensive Approach. In: 2023 20th Annual International Conference on Privacy, Security and Trust (PST); 2023 Aug 21-23; Copenhagen, Denmark. IEEE; 2023. p. 1-12.

Bellifemine FL, Caire G, Greenwood D. Developing Multi-Agent Systems with JADE. Vol. 7. Hoboken: John Wiley & Sons; 2007.

Wang P, Govindarasu M. Multi-Agent Based Attack-Resilient System Integrity Protection for Smart Grid. *IEEE Trans Smart Grid*. 2020;11(4):3447-56.

Mohamed AAR, Omran WA, Sharkawy R. Centralized/Decentralized Power Management Strategy for the Distribution Networks based on OPF and Multi-Agent Systems. In: 2021 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe); 2021 Oct 18-21; Espoo, Finland. IEEE; 2021. p. 1-5.

Elena DO, Florin D, Valentin G, Marius P, Octavian D, Catalin D. Multi-agent System for Smart Grids with Produced Energy from Photovoltaic Energy Sources. In: 2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI); 2022 Jun 30-Jul 1; Ploiesti, Romania. IEEE; 2022. p. 1-6.

Priyadarshana H, Hemapala KUM, Wijayapala WS, Saravanan V, Boralessa MKS. Developing multi-agent based micro-grid management system in JADE. In: 2019 2nd International Conference on Power and Embedded Drive Control (ICPEDC); 2019 Aug

21-23; Chennai, India. IEEE; 2019. p. 552-6.

Modbus Organization. Modbus Application Protocol Specification V1.1b. Andover: Modbus Organization; 2006.

Modbus Organization. MODBUS Messaging on TCP/IP Implementation Guide: V1.0b. Andover: Modbus Organization; 2006.

Techniques - ICS | MITRE ATT&CK® [Internet]. MITRE; 2022 [cited 2022 Dec 27]. Available from: <https://attack.mitre.org/techniques/ics>

Boakye-Boateng K, Ghorbani AA, Lashkari AH. RiskISM: A Risk Assessment Tool for Substations. In: 2021 IEEE 9th International Conference on Smart City and Informatization (iSCI); 2021 Oct 20-22; Shenyang, China. IEEE; 2021. p. 23-30.

Papadimitriou C, Sideri M. On the Floyd–Warshall algorithm for logic programs. *J Log Program*. 1999;41(2-3):129-37.

The ELK Stack: From the Creators of Elasticsearch | Elastic [Internet]. Elastic; 2023. Available from: <https://www.elastic.co/what-is/elk-stack>

Cho JH, Chan K, Adali S. A survey on trust modeling. *ACM Comput Surv*. 2015;48(2):1-40.

Critical Infrastructure Threat Information Sharing Framework. A Reference Guide for the Critical Infrastructure Community. Washington, DC: USA Homeland Security; 2016. 5 p.

Greer C, Wollman DA, Prochaska DE, Boynton PA, Mazer JA, Nguyen CT, et al. NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 3.0. Gaithersburg: National Institute of Standards and Technology; 2014. Technical Report.