

Article

Implementation of a Trust-Based Framework for Substation Defense in the Smart Grid

Kwasi Boakye-Boateng ^{1,*}, Ali A. Ghorbani ¹ and Arash Habibi Lashkari ²

¹ Canadian Institute for Cybersecurity (CIC), Faculty of Computer Science, University of New Brunswick (UNB), Fredericton, NB E3B 5A3, Canada; ghorbani@unb.ca

² Behaviour-Centric Cybersecurity Centre, School of Information Technology, York University, Toronto, ON M3J 1P3, Canada; ahabibil@yorku.ca

* Correspondence: kwasi.boakye-boateng@unb.ca

Abstract: The Smart Grid is a cyber-integrated power grid that manages electricity generation, transmission, and distribution to consumers and central to its functioning is the substation. However, integrating cyber-infrastructure into the substation has increased its attack surface. Notably, sophisticated attacks such as the PipeDream APT exploit multiple device protocols, such as Modbus, DNP3, and IEC61850. The substation's constraints pose challenges for implementing security measures such as encryption and intrusion detection systems. To address this, we propose a comprehensive trust-based framework aimed at enhancing substation security. The framework comprises a trust model, a risk posture model, and a trust transferability model. The trust model detects protocol-based attacks on Intelligent Electronic Devices and SCADA HMI systems, while the risk posture model dynamically assesses the substation's risk posture. The trust transferability model evaluates the feasibility of transferring and integrating a device and its trust capabilities into a different substation. The practical substation emulation involves a Docker-based testbed, employing a multi-agent architecture with a real-time Security Operations Center-influenced dashboard. Assessment involves testing against attacks guided by the MITRE ICS ATT&CK framework. Our framework displays resilience against diverse attacks, identifies malicious behavior, and rewards trustworthy devices.

Keywords: substation; trust; risk posture; smart grid; cybersecurity; Modbus; substation security; critical infrastructure; operational technology; trust transferability



Citation: Boakye-Boateng, K.; Ghorbani, A.A.; Lashkari, A.H. Implementation of a Trust-Based Framework for Substation Defense in the Smart Grid. *Smart Cities* **2024**, *7*, 99–140. <https://doi.org/10.3390/smartcities7010005>

Academic Editors: Antonio Moreno-Munoz and Pierluigi Siano

Received: 30 October 2023
Revised: 11 December 2023
Accepted: 22 December 2023
Published: 30 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The substation is regarded as the heart of the Smart Grid because its core role in generating, transmitting, and distributing electricity within the Smart Grid is the adjustment of voltages. Similar to the Smart Grid, the substation is now automated due to the integration of cyber-infrastructure, making it a cyber-physical infrastructure [1]. The Smart Grid is considered a critical infrastructure because any downtime experienced within it, especially by the substation, will affect the lives of many industrial/business and household consumers. There are various substations interconnected within the Smart Grid to ensure electricity reaches its consumers.

The cyber-physical infrastructure of the Smart Grid is enabled by the Supervisory Control and Data Acquisition (SCADA). SCADA enables efficient monitoring, control, and automation within the Smart Grid. SCADA comprises various devices, such as network switches, embedded systems, and computers.

The Substation Automation System (SAS), a part of SCADA, automates the operations within the substation. As shown in Figure 1, it has three levels, namely, the station level, bay level, and process level. A device that disrupts the flow of electricity within the substation when activated is the circuit breaker (CB). It is controlled by the Intelligent Electronic Device (IED).

A substation has a distinctive security challenge: when the control center is breached, the extent of damage within the substation is hard to assess. As Figure 1 illustrates, all of the attacks mentioned involved breaching the control center and then manipulating the IEDs to cut off power. The recent Colonial Pipeline attack [5], although not related to the Smart Grid, is an example of how a network can be shut down as a preventive measure when the control center is breached. The commonality between the pipeline attack and a substation is that they are both OT-based.

The performance of the substation network can be affected by implementing intrusion detection systems at the station, bay, or process levels of the substation due to time constraints [6,7]. Some solutions propose state-of-the-art cryptography that require state-of-the-art hardware, but on average, 44% of relays in the utilities surveyed have been operating for more than 15 years [8,9]. The ones that do not require hardware can be acquired by the APTs when they compromise the control center. Therefore, a solution that can be applied to the current state of the substation is required to secure the substation. We believe that the concept of trust can be that solution.

We present a trust-based framework that can be used to compliment current security measures and mitigate the threats presented by APTs in their post-compromise stages. Part of the framework is to provide the IEDs with the capabilities to be the last line of defence within the substation. This provide operators with time sufficient enough to mitigate any impact within the substation. The contributions of this research include the following:

- Trust-Based Framework: A comprehensive framework designed to enhance the security of substations by incorporating trust-based mechanisms.
- Trust Model Component: A trust model integrated into the framework, responsible for detecting protocol-based attacks targeting IEDs and SCADA HMI systems.
- Risk Posture Model Component: A component within the framework that determines the substation's risk posture in response to detected attacks.
- Trust Transferability Model Component: A component within the framework that determines whether a device and its trust capabilities can be transferred to a different substation environment and monitors its integration.
- Docker-Based Substation Testbed: A practical implementation environment created using Docker containers, establishing a multi-agent-based architecture that mirrors the substation's device ecosystem. An SOC-influenced dashboard provides real-time status updates for the substation and its devices.
- Attack Scenario Evaluations: Testing and evaluation of the framework through simulated attack scenarios, including external attacks, internal attacks from compromised SCADA HMIs, and internal attacks originating from compromised regular IEDs.
- Publicly Available Dataset: A publicly available dataset containing captures of our MAS testbed is provided on the CIC website (<https://www.unb.ca/cic/datasets/modbus-2023.html> (accessed on 31 August 2023)).

The paper has been structured to encompass various facets of trust. Section 2 offers a comprehensive detail of the concept of trust, delving into its current state-of-the-art. The motivation driving the research is also mentioned in this section. Further sections delve into topics such as multi-agent systems (Section 3) and the Modbus protocol (Section 4) to provide the necessary contextual grounding.

The preliminary models are introduced in Sections 5–8, where we present our proposed frameworks for trust assessment, risk posture evaluation, and trust transferability. Section 9 outlines the implementation details. The model is subjected to evaluation in Section 10 and we conclude the paper in Section 11. Figure 2 provides a visual representation of the paper's structure.

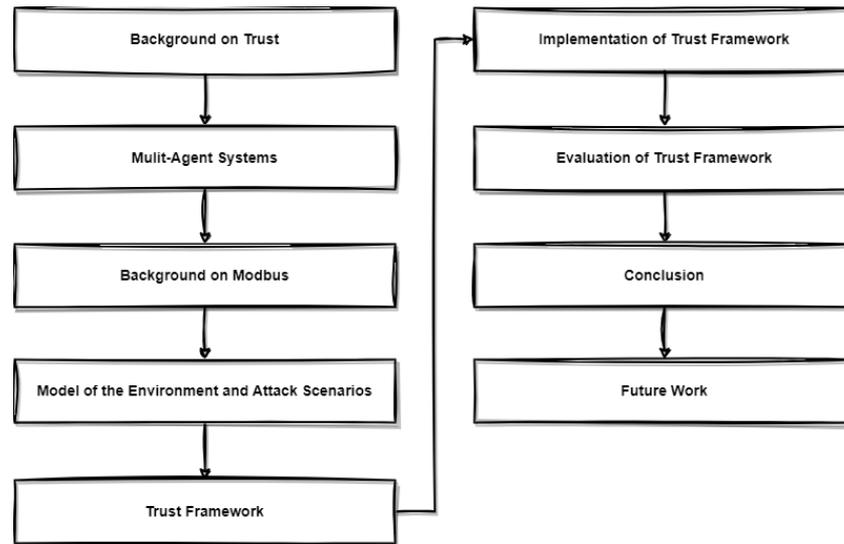


Figure 2. Organisation of the paper.

2. Background on Trust

Trust can be viewed and defined in different ways. One definition from the social sciences describes trust as a personal belief about the actions of a specific entity [10]. Another perspective defines trust as an agency's likelihood of taking a specific action [11]. In this paper, the entity who trusts is called the *agent* or *trustor*, while the entity being trusted is referred to as the *subject* or *trustee*. According to Rousseau et al. [12], trust is when the trustor intends to accept their vulnerability, based on positive expectations about the intentions or behavior of the trustee.

In this section, we present the background and formulation of three substation-centric trust-related topics, namely, trust, risk posture, trust transitivity, and trust transferability.

2.1. Trust

The trust (T_{ij}) that an agent (a_i) places in a subject (a_j) within a given time period (t) can be mathematically formulated as a tuple, as shown in Equation (1) [13]. In this equation, r_{ij} represents the perceived risk associated with agent a_i trusting subject a_j , α_{ij} denotes the nature of the transaction or communication between the two entities, k_{ij}^t signifies the knowledge acquired during the interaction between a_i and a_j within t , and T'_{ij} denotes the previous trust value established between the two entities. The previous trust value refers to the trust score established by the agent in the course of a preceding transaction. Typically, the initial step is to set the previous trust score to a specific threshold, after which an algorithm is applied for continuous adjustment. The literature presents variations in updating trust scores, including the utilization of scores within a time window, dependence on the prior trust score, consideration of transaction-specific parameters, or a combination of these factors. It is essential to emphasize that this list is not exhaustive.

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

Trust is typically represented as a continuous variable over a specified range, usually $-1 \leq T \leq 1$ or $0 \leq T \leq 1$, where 1 represents complete trust, -1 represents complete mistrust, and 0 represents no trust.

2.2. Trust Transitivity and Trust Transferability

According to the literature, trust transitivity occurs when an agent/trustor, i , trusts an unknown subject, j , because the unknown subject is related to a third agent, k , that

the trustor trusts [12]. Equation (2) formulates when trust transitivity is accepted and Equation (3) represents when it is not:

$$T_{ik} \approx T_{jk} \text{ where } 1 - T_{ij} \approx 0, 1 - T_{jk} \not\approx 0 \quad (2)$$

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

Trust transferability is established when the trust capabilities of an agent and the agent can be reintegrated into a different trusted environment. For an agent, i , with trust properties, \mathbb{T}_i , and a context, c , we can define transferability, T_f , as the output of a binary function that evaluates whether a device's trust functionality is transferable or not, as shown in Equation (4):

$$T_f = f(\mathbb{T}_i, c) \quad (4)$$

2.3. State of the Art

The existing research pertaining to trust within substations remains limited [14]. Previous endeavors to address trust within the domain of Smart Grid involved the implementation of a reputation-based system to alleviate the consequences of faulty agents within a substation's backup protection scheme. However, this system encountered challenges in differentiating between malicious and non-malicious causes of the observed effects [15,16]. The work carried out by Borowski et al. [15] and Fadul et al. [16] represented the pioneering attempts to incorporate trust in the context of a multi-agent system-based substation—which is within the scope of this paper.

Qureshi et al. presented a trust model that employs packet drop ratio, packet departure and arrival times, and packet count for detecting malicious devices [17]. Nevertheless, it should be noted that the model exhibits limitations in identifying malicious packets that are precisely timed, carry malicious payloads, or contain falsified data, as exemplified by the Stuxnet cyberattack.

Research conducted by Wang et al. implemented a trusted server within a Purdue-model-based Smart Grid environment [18]. They proposed a TPM-based trust engine implemented within devices to collect data and original stored transactions. The drawback of this research is the limited resources to have a trust engine implemented. Not only that, it will not work on old devices which may not have the expansion or resource capability to implement it. Furthermore, their work did not factor in the substation's risk posture when a node exhibits untrustworthy behavior. Their minimum response time for their model is 20 ms.

The research conducted by Boakye-Boateng et al. on trust incorporated the notion of communication familiarity between devices and the subsequent consequences of accepting a device's request to compute trust [13]. However, it is worth noting that their work, similar to the previously mentioned literature, did not include utilizing trust to determine the risk posture of the substation. Furthermore, the existing literature did not investigate the possibility of a trust IED (an IED that utilizes a trust model) being assessed based on trust transferability to allow it to be integrated into another substation [19].

2.4. Motivation

The substation's operational constraints prevent the implementation of security measures such as encryption and intrusion detection at the station, bay, and process levels. Introducing intrusion detection systems at these levels would lead to duplicated network packets, negatively impacting device and overall substation performance. Although encryption can be applied, APTs compromising SCADA HMI can access cipher suites, allowing them to send seemingly legitimate commands to IEDs. Unfortunately, IEDs lack the capability to discern the malicious nature of such requests. We contend that establishing trust is crucial in addressing this issue, but research on trust in this context is notably limited, as outlined in the preceding section. The extant literature employs a trust score as a metric to determine whether a device is malicious or not. However, the device's trust score is

specific only to that device and does not present a comprehensive overview of the trust score for the entire substation. Additionally, the trust score provides no insight about the substation's risk posture. Furthermore, our study revealed that no trust model exists that implements such a correlation.

A network's risk posture refers to the level of exposure to potential risks and hazards that the network faces. The risk posture of an electrical substation can be influenced by a variety of factors, including the potential consequences of a security breach or failure.

Although individual trust addresses specific risks, it remains crucial to factor in the collective trust established by numerous devices when striving to offer a comprehensive assessment of trust within the substation context. This holistic perspective on trust can subsequently be translated into a comprehension of the substation's overall risk, thereby offering valuable insights into its risk posture.

To operationalize this approach, it becomes imperative to integrate a multi-agent system (MAS) into the substation's architectural framework. Although there exist two notable research papers that have delved into trust within MAS architecture—namely, Borowski et al.'s work (2011) [15] and Fadul et al.'s study (2013) [16]—it is worth noting that these papers did not explore the specific angle that the present paper aims to investigate.

Furthermore, there is a possibility of a trust IED being transferred to another substation. In such a situation, trust transferability becomes important in determining whether a device can be allowed to or is capable of being integrated in the new environment. There is also the issue of ensuring that the device is not compromised and whether it is the required mechanism to detect and remove it.

3. Multi-Agent Systems

A Multi-Agent System (MAS) is a system of intelligent agents [20]. An intelligent agent encompasses four essential characteristics: pro-activeness, reactivity, social ability, and autonomy. Pro-activeness mandates that an agent dynamically adjusts its behavior to fulfill its objectives. Reactivity necessitates prompt responses by an agent to alterations in its environment, coupled with suitable actions aligned with its objectives and environmental shifts. Social ability pertains to an agent's capability to engage in cooperative interactions and proficient negotiations with other agents. Autonomy signifies agents' capacity to function independently of external systems or human intervention.

The MAS operates under a shared objective that necessitates each agent's goals to be contributory. Three distinct types of MAS architectures exist: centralized, decentralized, and hybrid. In a centralized architecture, agents report to a central agent who issues instructions. Conversely, a decentralized architecture involves agents communicating within clusters, each with equal priority. In the event of a centralized architecture, the MAS's failure occurs with the central agent's demise. On the other hand, the optimization of a decentralized architecture presents challenges due to the localized interactions between agents. A hybrid architecture amalgamates the benefits of both these architectural approaches. Extensive research has been undertaken concerning MAS in diverse smart grid domains, encompassing areas such as distributed generation [21,22], renewable energy integration [23], and microgrid management [24].

4. Modbus TCP

The Modbus protocol is integral for automating equipment control and supervision [25]. Modbus TCP/IP, a variant operating over TCP/IP on TCP port 502 [26], is a pragmatic choice due to its user-friendly interface, popularity, and comprehensive documentation. It was also the protocol employed in previous literature [13,14,18], facilitating its selection for further advancements. Furthermore, APTs such as PipeDream [3] have components that target the protocol. Modbus-based attacks necessitate the manipulation of Modbus packets or the utilization of Modbus packets to extract data from IEDs, or exploit IEDs to disrupt the substation. Modbus follows a client-server model where the client sends request packets and the server responds, without authorization for unsolicited packets.

4.1. Modbus Packet Structure

As shown in Figure 3, the Modbus packet comprises a Modbus Application (MBAP) Header, which contains essential details such as the transaction identifier, protocol identifier, unit identifier and length field. Subsequently, the Modbus PDU (Protocol Data Unit) is encapsulated within the packet. The PDU encompasses function code (FC) and data that include the starting address being accessed, count of addresses being accessed, byte count of data, quantity of addresses being returned, and other parameters essential for carrying out Modbus transactions. The properties of the Modbus packet will be extracted as features for our trust model.

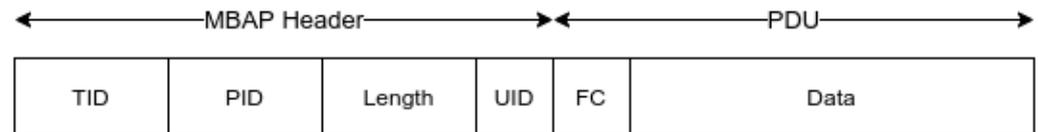


Figure 3. Modbus TCP packet structure.

4.2. Modbus Address Types and Function Codes

We have selectively opted for a subset of Modbus function codes (see Table 1) to form the foundation of our investigation, a decision rooted in the analysis of diverse datasets. When engaging in a read query, it becomes imperative to furnish both the start address and the corresponding quantity of addresses to be read. Analogously, a write query targeted at a singular address mandates the provision of the address itself, in tandem with the associated value to be written. Any query sent by the client to the server generates a response from the server which contains the address and the values stored at that address.

Table 1. Modbus Address Type and their 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 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 Single Register	06
Input Register	Read Only	16 bit	30,001–39,999	Read Input Register	04

Referring to Table 1, coil and discrete input address types are characterized by 1-bit sizes. In the context of Intelligent Electronic Devices (IEDs), a coil address assumes significance as it can actuate the circuit breaker through the transmission of a Write Single Coil command (Function Code 05) contained within a Modbus packet, and subsequently, programmatically ascertaining the status of the circuit breaker involves issuing a Read Discrete Input command (Function Code 02) to retrieve the value stored at the discrete input address. Additionally, a Holding Register address serves to retain a value that, when manipulated, can influence the constantly updated voltage value stored in the input register.

5. Models and Scenario

5.1. Substation Model

The substation, denoted as Ξ , where $\Xi = (M, N, S)$, encompassing sets of servers, clients, and network devices, denoted by $S = \{s_1, s_2, \dots\}$, $M = \{m_1, m_2, \dots\}$, and $N = \{n_1, n_2, \dots\}$, respectively. The members of M and S can assume the interchangeable roles of agent and subject. Set N establishes an interconnection between S and M . In concordance, we introduce sets $Q = \{q_1, q_2, \dots, q_i\}$ and $R = \{r_1, r_2, \dots, r_i\}$, signifying queries and correspondings responses. At periodic intervals, m_i transmits queries (Q) to s_i and

subsequently receives responses (R) from s_i . It is notable that each pairing of m_i and s_i may engender a unique pair of Q and R . The inherent operations related to a query and its ensuing response are categorized using the memory access type flag (θ) into either read ($\theta = 0$) or write ($\theta = 1$). The queries launched by the adversary, $Q' = \{q'_1, q'_2, \dots, q'_i\}$, and malicious responses provided by the adversary, $R' = \{r'_1, r'_2, \dots, r'_i\}$, are systematically defined.

$S' = \{s'_1, s'_2, \dots\}$ and $M = \{m'_1, m'_2, \dots\}$ are defined as malicious servers and clients, respectively. They are either compromised devices or rogue devices that belong to the adversary. Any compromised m or s becomes a part of M' or S' .

5.2. Attack Scenarios

The primary objective of the attacker in relation to the substation is to gain control over one or more components within S , ultimately leading to a disruption in the Smart Grid. Often, the IED serves as the vulnerable element in this context. Drawing from publicly documented attacks, this section outlines a compilation of attacks to be implemented in this research. The list of attacks is as follows:

- **Write Attack:** In this attack, q' bearing $\theta = 1$ is directed towards s_i , targeting all existing Modbus addresses, either without preceding reconnaissance or subsequent to a baseline replay attack. Alternatively, this attack could be tailored to concentrate on a specific address of s_i with $\theta = 1$, necessitating the successful completion of a reconnaissance attack.
- **Query Flooding:** In this attack, m' or s' inundates a device with an excessive volume of Q' or R' , subsequently causing the targeted device to deplete its available resources.
- **Malicious Packet Crafting:** This involves the transmission of a malevolent packet by either m' or s' . The crafted packet is designed to execute a payload or initiate a buffer overflow. The packet itself can take the form of q' or r' . Examples encompass payload injection, frame stacking, manipulation of packet length, and false data injection.
- **Baseline Replay Attack:** Following a thorough profiling of the substation, aimed at evading detection, m'_i or s'_i can initiate the replay of Q or R to a designated device.
- **Reconnaissance:** When $\theta = 0$, m'_i can dispatch q' to s_i , systematically covering all existing Modbus addresses. This endeavor is undertaken to accumulate intelligence about the substation.

Each of these attack scenarios represents a distinct vector through which the attacker seeks to compromise the integrity and functionality of the substation.

Mapping to the MITRE ATT&CK ICS Framework

The MITRE ICS ATT&CK framework is a comprehensive knowledge base that outlines the tactics, techniques, and procedures employed by adversaries in cyber attacks, aiding cybersecurity professionals in enhancing threat detection and response strategies. We have correlated the discussed attack scenarios from Section 5.2 with the MITRE ATT&CK ICS Framework [27]. The outcomes of this mapping effort are presented in Table 2. Among the 12 tactics encompassed within the framework, the identified attack scenarios align with four distinct tactics.

The first tactic, denoted as Collection, encompasses activities geared towards accumulating pertinent information about a network or system. This involves pivotal actions such as reconnaissance, scanning, and comprehensive data gathering.

The subsequent tactic, Inhibit Response Function, revolves around the disruption of normal operations within a network or system by impeding its capacity to duly respond to requests or commands. This tactic encompasses activities such as denial of service attacks, the injection of excessive load, and the introduction of delays.

Impair Process Control constitutes the third tactic, centered on the interference with the smooth operation of a network or system by impairing its ability to effectively govern processes. This encompasses strategic actions such as brute force attacks and the illicit injection of data.

Finally, the fourth tactic, Evasion, pertains to efforts aimed at sidestepping detection or thwarting the attribution of an attack. This is executed by concealing the source or nature of the attack through activities such as spoofing and masking.

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

Tactic	Technique	Attack
Collection	Automated Collection	Reconnaissance—Scan addresses
		Query flooding
		Load malicious payloads
Inhibit Response Function	Denial of Service	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

6. Trust Formulation for Substation Devices

In our preceding research, we underscored that computing trust involves two core constituents: familiarity (F_i) and consequence (C_i) [13,14]. Familiarity encapsulates three fundamental factors: frequency (E_f), intensity (E_i), and similarity (E_s). Evaluating these factors demanded the utilization of Modbus characteristics extracted from q or r as input to these factors. The second facet, consequence (C_i), encompasses a synthesis of flags, encompassing environment status attack flag (τ), replay attack flag (ω), reconnaissance attack flag (ζ), packet manipulation flag (ϕ), and query flooding attack flag (χ). Table 3 includes the definition of Modbus features that were extracted to compute F_i and C_i . Detailed descriptions of the features can be found in our previous work [13,14].

Table 3. Table of notations for Section 6.

Symbol	Description
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 of 0 and 1.
ϑ	Memory access type (read or write) flag. $\vartheta \in \{0, 1\}$
E_i	Exposure intensity. $E_i \in \mathbb{Q}^+$ and $E_i \in [0, 1]$.
E_f	Exposure frequency. $E_f \in \mathbb{Q}^+$ and $E_f \in [0, 1]$.
E_s	Similar exposure. $E_s \in \mathbb{Q}^+$ and $E_s \in [0, 1]$.
E_x^T	An exposure's threshold. The notation x is replaced with i, f or s
κ_x	An alarm associated with a particular component of trust. The notation x is replaced with $E_i, E_f, E_s, C_i, \tau, \omega, \zeta, \phi$, or χ . $\kappa_x \in \mathbb{Z}^+$
Γ	A set of Modbus features associated E_f
Γ_i	A set of Modbus features, extracted from q_i or r_i , associated E_f
Γ_R	A reference set of Modbus features associated E_f
γ_{frc}	Count for read coil function code. $\gamma_{frc} \in \mathbb{Z}^+$

Table 3. Cont.

Symbol	Description
γ_{cq}	Coil quantity. $\gamma_{cq} \in \mathbb{Z}^+$
γ_{fwsc}	Count for write single coil function code. $\gamma_{fwsc} \in \mathbb{Z}^+$
γ_{cv}	Coil value. $\gamma_{cv} \in \{0, 1\}$
γ_{cvs}	Set of coil values. $\gamma_{cvs} = \{x_1, x_2, \dots, x_i\}$ where $x_i \in \{0, 1\}$
γ_{fwmc}	Count for write multiple coils function code. $\gamma_{fwmc} \in \mathbb{Z}^+$
γ_{cdc}	Coil data byte count. $\gamma_{cdc} \in \mathbb{Z}^+$
γ_{frdi}	Count for read discrete input function code. $\gamma_{frdi} \in \mathbb{Z}^+$
γ_{diq}	Discrete input quantity. $\gamma_{diq} \in \mathbb{Z}^+$
γ_{didc}	Discrete input data byte count. $\gamma_{didc} \in \mathbb{Z}^+$
γ_{divs}	Set of discrete input values. $\gamma_{divs} = \{x_1, x_2, \dots, x_i\}$ where $x_i \in \{0, 1\}$
γ_{frir}	Count for read input register function code. $\gamma_{frir} \in \mathbb{Z}^+$
γ_{irq}	Input register quantity. $\gamma_{irq} \in \mathbb{Z}^+$
γ_{irdc}	Input register data byte count. $\gamma_{irdc} \in \mathbb{Z}^+$
γ_{irvs}	Set of input register values. $\gamma_{irvs} = \{x_1, x_2, \dots, x_i\}$ where $x_i \in \mathbb{Q}^+$
γ_{irv}	Input register value. $\gamma_{irv} \in \mathbb{Q}^+$
γ_{frhr}	Count for read holding register function code. $\gamma_{frhr} \in \mathbb{Z}^+$
γ_{hrq}	Holding register quantity. $\gamma_{hrq} \in \mathbb{Z}^+$
γ_{fwsr}	Count for write single register function code. $\gamma_{fwsr} \in \mathbb{Z}^+$
γ_{hrv}	Holding register value. $\gamma_{hrv} \in \mathbb{Q}^+$
γ_{fwmr}	Count for Write Multiple Registers function code. $\gamma_{fwmr} \in \mathbb{Z}^+$
γ_{hrvs}	Set of holding register values. $\gamma_{hrvs} = \{x_1, x_2, \dots, x_i\}$ where $x_i \in \mathbb{Q}^+$
γ_{hrdc}	Holding register data byte count. $\gamma_{hrdc} \in \mathbb{Z}^+$
γ_{fs}	Frame size feature. $\gamma_{fs} \in \mathbb{Z}^+$
γ_{fs_R}	Reference frame size feature. $\gamma_{fs_R} \in \mathbb{Z}^+$
γ_{fs_i}	Frame size feature for q_i or r_i . $\gamma_{fs_i} \in \mathbb{Z}^+$
l_{hi}	Length of the MBAP header
γ_{fc_i}	Function code indicator of q_i or r_i . $\gamma_{fc_i} \in \{0, 1\}$
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
Z_R	A reference set of Modbus features associated with E_i
ζ_{pt}	Pre-time feature. $\zeta_{pt} \in \mathbb{Z}^+$
ζ_{pt}^T	Pre-time feature threshold. $\zeta_{pt}^T \in \mathbb{Z}^+$
ζ_{qq}	Inter-query time feature. $\zeta_{qq} \in \mathbb{Q}^+$
ζ_{rr}	Inter-response time feature. $\zeta_{rr} \in \mathbb{Q}^+$
ζ_{qr}	Query-response time feature. $\zeta_{qr} \in \mathbb{Q}^+$
ζ_{tt}	Transaction time feature. $\zeta_{tt} \in \mathbb{Q}^+$
ζ_{to}	Timeout feature. $\zeta_{to} \in \mathbb{Q}^+$
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_{p_i}	Port mismatch feature. $\psi_p \in \{0, 1\}$

Table 3. Cont.

Symbol	Description
ψ_η or ψ_{η_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 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 device. $\theta_\mu \in \{0, 1\}$

6.1. Familiarity-Based Definitions

6.1.1. Familiarity

Familiarity, denoted as F_i , is formally articulated in Equation (5), wherein E_f represents frequency, E_s signifies similarity, and E_i denotes intensity. Moreover, it is important to note that F_i adheres to the condition $F_i \not\geq \min\{E_i, E_s, E_f\}$ and $F_i \in [0, 1]$:

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

6.1.2. Exposure Frequency

For each q_i or r_i received, Γ , which is extracted from it, is defined in Equation (6). Γ consists of four feature sets and a general feature: coil set ($\Gamma_c = \{\gamma_{frc}, \gamma_{cq}, \gamma_{fwsc}, \gamma_{cv}, \gamma_{cvs}, \gamma_{fwmc}, \gamma_{cdc}\}$), discrete input set ($\Gamma_{di} = \{\gamma_{frdi}, \gamma_{diq}, \gamma_{didc}, \gamma_{divs}\}$), input register set ($\Gamma_{ir} = \{\gamma_{frir}, \gamma_{irq}, \gamma_{irdc}, \gamma_{irov}, \gamma_{iro}\}$), and holding register set ($\Gamma_{hr} = \{\gamma_{frhr}, \gamma_{hrq}, \gamma_{fwsr}, \gamma_{hrv}, \gamma_{fwmr}, \gamma_{hrvs}, \gamma_{hrdc}\}$).

The frequency, E_f , is governed by Equation (7), with $E_f \in [0, 1]$ and E_f^T , signifying the frequency threshold:

$$\Gamma = \{\gamma_{fs}\} \cup \Gamma_c \cup \Gamma_{di} \cup \Gamma_{hr} \cup \Gamma_{ir} \quad (6)$$

$$E_f = \begin{cases} 0, \kappa_{E_f} = 1, \text{ if } l_{h_i} < 7 \\ 0, \kappa_{E_f} = 2, \text{ if } \gamma_{fs_R} \neq \gamma_{fs_i} \\ 0, \kappa_{E_f} = 3, \text{ if } \gamma_{fc_i} = 0 \\ \frac{\Gamma_R \cdot \Gamma_i}{\|\Gamma_R\| \|\Gamma_i\|}, \kappa_{E_f} = 0 \text{ if } E_f \geq E_f^T \\ \frac{\Gamma_R \cdot \Gamma_i}{\|\Gamma_R\| \|\Gamma_i\|}, \kappa_{E_f} = 4 \text{ if } E_f < E_f^T \end{cases} \quad (7)$$

6.1.3. Exposure Intensity

When transmitting q_i or r_i , a set of attributes denoted as $Z = \{\zeta_{pt}, \zeta_{qq}, \zeta_{rr}, \zeta_{qr}, \zeta_{tt}, \zeta_{to}\}$ is generated and utilized for the computation of intensity, represented as E_i . The features include ζ_{pt} for pre-time, ζ_{qq} for inter-query time, ζ_{rr} for inter-response time, ζ_{qr} for query-response time, ζ_{tt} for transaction time, and ζ_{to} for timeout. The intensity is governed by Equation (8), with $E_i \in [0, 1]$ and E_i^T signifying the intensity threshold:

$$E_i = \begin{cases} 1, \kappa_{E_i} = 0 \text{ if } \vartheta = 1 \\ 0, \kappa_{E_i} = 1, \text{ if } \zeta_{pt} > \zeta_{pt}^T \\ \frac{Z_R \cdot Z_i}{\|Z_R\| \|Z_i\|}, \kappa_{E_i} = 0 \text{ if } E_i \geq E_i^T \\ \frac{Z_R \cdot Z_i}{\|Z_R\| \|Z_i\|}, \kappa_{E_i} = 2 \text{ if } E_i < E_i^T \end{cases} \quad (8)$$

6.1.4. Similarity

A collection of attributes denoted as Ψ (refer to Equation (9)) encompasses three distinct groups of features: the general packet features, $\Psi_{gp} = \{\psi_s, \psi_p, \psi_\eta, \psi_{us}, \psi_{mas}, \psi_{ma}, \psi_{fc}\}$; single-bit register features, $\Psi_{sb} = \{\psi_{mdi}, \psi_{mdir}, \psi_{mcr}, \psi_{mcq}\}$; and byte register features $\Psi_{br} = \{\psi_{mhr}, \psi_{mhrq}, \psi_{mirq}, \psi_{mirr}\}$. The variable E_s , as defined in Equation (10), is constrained within the range $[0, 1]$ and determines the subsequent generation of the associated κ_{E_s} :

$$\Psi = \Psi_{gp} \cup \Psi_{sb} \cup \Psi_{br} \quad (9)$$

$$E_s = \begin{cases} 0, \kappa_{E_s} = 1 \text{ if } \psi_{\eta_i} \\ 0, \kappa_{E_s} = 2 \text{ if } \psi_{p_i} \neq 502 \\ \frac{\Psi_R \cdot \Psi_i}{\|\Psi_R\| \|\Psi_i\|}, \kappa_{E_s} = 0, \text{ if } E_s \geq E_s^T \\ \frac{\Psi_R \cdot \Psi_i}{\|\Psi_R\| \|\Psi_i\|}, \kappa_{E_s} = 3, \text{ if } E_s < E_s^T \\ E_s^T, \kappa_{E_s} = 4, \text{ if } \psi_{ms} = 1, \psi_{us} = 1 \end{cases} \quad (10)$$

6.2. Consequence

Using the environment status attack flag (τ), replay attack flag (ω), reconnaissance attack flag (ξ), packet manipulation attack flag (ϕ), and query flooding attack flag (χ), consequence, C_i , is calculated in Equation (11):

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

6.3. Trust of a Device

The trust of a device, denoted as $T_i = \{\beta_i, \kappa_{E_f}, \kappa_{E_s}, \kappa_{E_i}, \kappa_C\}$, is represented as an ordered set of values, specifically a tuple, where β_i signifies the trust score. The κ values encapsulate factors that can lead to a deterioration in trust. The interpretation of β_i is outlined in Equation (12), where it assumes values within the range $[-1, 1]$. In the equation, θ_I denotes the initial state prior to trust calculation, β_i^o corresponds to the previous trust score, β_i^T serves as the trust score threshold, μ represents the weight assigned to forgiveness with μ constrained within the interval $[0, 1]$, and θ_μ pertains to the state of forgiveness. The attributes pertaining to forgiveness, however, are reserved for subsequent research efforts.

Within Equation (1), the parameter r_{ij} is associated with risk and maps to C_i , while T'_{ij} corresponds to β_i^o . Notably, the remaining parameters linked to the three exposures are intertwined due to the significant information these exposures contain about said parameters:

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

6.4. Out of Sequence Handler

The weakness of the previous model (presented in the previous subsections) was addressing out of sequence packets, which could lead to false positives [13,14]. In addressing queries or responses that arrive in a non-sequential manner, we employ an iterative process involving a pointer for both the Q and R sequences. The pointers, denoted as x_q and x_r , respectively, indicate the anticipated position of the i -th query or response a device is set to receive (refer to Equations (13) and (14)). By passing the incoming query or response along with the pointer to a designated function, a determination is made regarding the selection of the current i -th message or another message at position k -th within Q or R (outlined in Equation (15)). Subsequently, the replay indicator (y) and the message sequence flag (ψ_{ms}) are adjusted as required and then conveyed to both E_s and C_i . The indicator and the flag are set when the value set is other than i . The value of i is passed on the E_i, E_s, E_f and to the flags of C_i :

$$f(x_q, q) = q_i \quad (13)$$

$$f(x_r, r) = r_i \quad (14)$$

$$i = \begin{cases} i, \psi_{ms} = 0, & \text{if } f(x_q, q) = q_i \\ i - 1, y = 1, \psi_{ms} = 1, & \text{if } f(x_q, q) = q_{i-1} \\ k, y = 1, \psi_{ms} = 1, & \text{if } f(x_q, q) = q_k \\ i, \psi_{ms} = 0, & \text{if } f(x_r, r) = r_i \\ i - 1, y = 1, \psi_{ms} = 1, & \text{if } f(x_r, r) = r_{i-1} \\ k, y = 1, \psi_{ms} = 1, & \text{if } f(x_r, r) = r_k \end{cases} \quad (15)$$

7. Risk Posture

In order to ascertain the risk posture of the substations, a comprehensive risk assessment is imperative. For this purpose, we employ a previously developed risk assessment tool, as documented in our earlier research [28]. This tool facilitates the acquisition of risk assessments for individual devices within the substation, which subsequently culminate in the computation of the substation's overall risk posture. Here, $D = \{d_1, d_2, \dots, d_n\}$ denotes a collection of n devices situated within the substation.

The final reachability matrix delineates the functional interconnections existing among devices housed within the substation. This final reachability matrix, denoted as $F = (f_{ij})_{m \times m}$ (where $m < n$), possesses binary elements and finds its visual representation in the form of a graph.

The functional impact exerted by a specific device (d_i) is contingent upon the degree of the said device, as illustrated in Equation (16):

$$C(d_i) = \text{deg}(d_i). \quad (16)$$

Consequently, the cumulative functional influence stemming from all devices is formally characterized within Equation (17):

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

Every individual device is attributed a criticality level denoted as $l = \{d_i, \dots, d_p\}$, where $l \subset D$, with the condition that $p < n$. The criticality levels, $L = \{l_1, l_i, \dots, l_m\}$ (where $1 \leq i \leq m$), are structured in such a way that there can exist m distinct levels, while adhering to the constraint $m < n$.

The combined functional impact of devices within a given level is precisely outlined in Equation (18):

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

7.1. Identifying Functional Influence of Affected Devices

Consider a set denoted as $X = \{d_i, \dots, d_q\}$, comprising q devices (with the constraint $q < n$) that have identified a malicious activity originating from a compromised device. It holds true that $\beta < \beta^T$ was scored by all q devices, and these devices are a subset of the overall device set D .

Let $Y = \{d_i, \dots, d_g\}$ represent a set comprising g devices (with the condition $g < n$) that were identified as malicious by the set X . It is understood that Y is a subset of the device set D .

Let $X_k = \{d_i, \dots, d_z\}$ denote a set encompassing z devices characterized by the utmost criticality level k , where it holds that $X_k \subset X \cup Y$. The collective functionality of all devices within the set X_k is formally established through the expression detailed in Equation (19):

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

7.2. Calculating Risk Posture

Let $C_T(X')$ represent the cumulative functionality of devices across all critical levels that are lower than k , as articulated in Equation (20):

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

The cascading effect, denoted as Λ , is formally characterized within Equation (21):

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

Lastly, the risk posture of the substation, denoted as τ , is conclusively delineated in Equation (22).

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

The trust score attributed to the substation is precisely outlined in Equation (23):

$$\beta_S = \frac{2}{m(m+1)} \left[\sum_{j=1}^m \left(\frac{j}{n} \sum_{i=1}^n \beta_i \right) \right] \quad (23)$$

8. Trust Transferability

The configuration of the substation grid, comprising n substations, is represented as a directed graph denoted by $\mathbb{S} = (\Xi_{ij})_{n \times n}$, having an adjacency matrix, where Ξ_{ij} signifies the connection between two distinct substations, namely, Ξ_i and Ξ_j . The transitive closure of graph \mathbb{S} is computed through the utilization of the Floyd–Warshall Algorithm [29], as depicted in Equation (24):

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

Based on the modified matrix, the degree of substation Ξ_i is determined following the formulation outlined in Equation (25).

$$C(\Xi_i) = \text{deg}(\Xi_i). \quad (25)$$

Consequently, this process yields a vector denoted as $C = \{C(\Xi_1), C(\Xi_2), \dots, C(\Xi_n)\}$, encompassing the degree values for each individual Ξ_i . The vector C along with Ξ_i are fed into a function that assigns a rank to Ξ_i based on the highest degree count. Subsequently, the computed ranks for each Ξ_i are compiled into a vector, as depicted in Equation (26):

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

Consider $\mathbb{T} = \{\beta_1, \beta_2, \dots, \beta_k\}$ as a collection of trust scores documented over a specific period denoted as t . In the context of an IED, the trust scores originating from queries received from SCADA HMI are denoted as \mathbb{T}_q , while the trust scores arising from responses received by the IED from SCADA HMI are indicated as \mathbb{T}_r .

Subsequently, by employing the trust scores attributed to a new IED denoted as j , as well as the replaced or disconnected IED labeled as i , the values F_q and F_r are derived, following the formulations outlined in Equations (27) and (28). These values, F_q and F_r , are referred to as the query acceptance and response acceptance flags, respectively:

$$F_q = \begin{cases} 1, & \text{if } \psi \leq \frac{\mathbb{T}_{jq} \cdot \mathbb{T}_{iq}}{\|\mathbb{T}_{iq}\| \|\mathbb{T}_{iq}\|} \\ 0 & \end{cases} \quad (27)$$

$$F_r = \begin{cases} 1, & \text{if } \psi \leq \frac{\mathbb{T}_{jr} \cdot \mathbb{T}_{ir}}{\|\mathbb{T}_{jr}\| \|\mathbb{T}_{ir}\|} \\ 0 & \end{cases} \quad (28)$$

Employing the derived values F_q and F_r , the acceptance flag denoted as ε is established in accordance with the formulation presented in Equation (29):

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

Let there exist a designated minimum probation period denoted as $t_{p_{min}}$, during which the IED j is afforded the opportunity to demonstrate its suitability for integration into the substation's network. The computation of the probation period t_p for an IED within a given substation Ξ_i is contingent upon both the rank of the substation r_i and the cascading effect Λ introduced by the IED within Ξ_i , following the structure elucidated in Equation (30). Here, t'_p serves as a countdown timer employed to monitor the elapsed time, and it equates to t_p at the onset of the timer:

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

The symbol ω represents the probation point, a parameter utilized to increment or decrement the countdown timer t'_p during specific stages of the probation period:

$$\omega = \frac{0.5t(|\mathbb{T}_{j_q}| + |\mathbb{T}_{i_r}|)}{|\mathbb{T}_{j_q}| |\mathbb{T}_{i_r}|} \quad (31)$$

The term ϱ denotes the consideration stage, while \varkappa represents the consideration weight; it is established that \varkappa is a real number adhering to the condition $1 \leq \varkappa \leq 2$:

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

The countdown timer t'_p functions to monitor the passage of time, initially set as $t'_p = t_p$ when the timer is initiated. The ultimate countdown outcome is ascertained through the formulation presented in Equation (33); in this equation, \beth signifies the weight attributed to points, constrained within the range $1 \leq \beth \leq t_p$, and t_e denotes the duration that has elapsed since the timer's initiation:

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

Ultimately, the transferability flag, denoted as \aleph , serves as the decisive factor in determining whether the IED j is deemed suitable for inclusion or rejected from becoming a component of the substation's network:

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

9. Implementation

The experimentation was conducted within a virtual environment utilizing a Linux operating system (Ubuntu 20.04.3 LTS). The virtual machine employed an Intel® Xeon® CPU E5-2695 v4 with 2 cores operating at 2.10 GHz. The model was implemented using Java and compiled into a JAR file. Docker containers were instantiated to emulate IEDs and SCADA HMIs. Python scripts were generated to execute the operational logic of both IEDs and SCADA HMIs. The IED logic involved periodic random voltage value alterations or adjustments triggered by requests received from SCADA HMIs. In contrast, the SCADA HMI logic encompassed tap-changing based on IED-received values and the initiation of closure or opening actions towards CBs in response to overvoltage or undervoltage conditions.

The Docker containers were configured to encapsulate either the JAR files and scripts or solely the scripts themselves, as illustrated in Figure 4. Devices categorized as IEDs or SCADA HMIs, containing only the scripts, are considered insecure. Conversely, those labeled as trust IEDs or trust SCADA HMIs include both the JAR files and scripts. Each secure device incorporates an agent responsible for transmitting trust scores to a central agent. This central agent undertakes the computation of the risk posture, generates logs, and subsequently, channels this information to the ELK stack [30] for the purpose of generating a Security Operations Center (SOC)-influenced visualization.

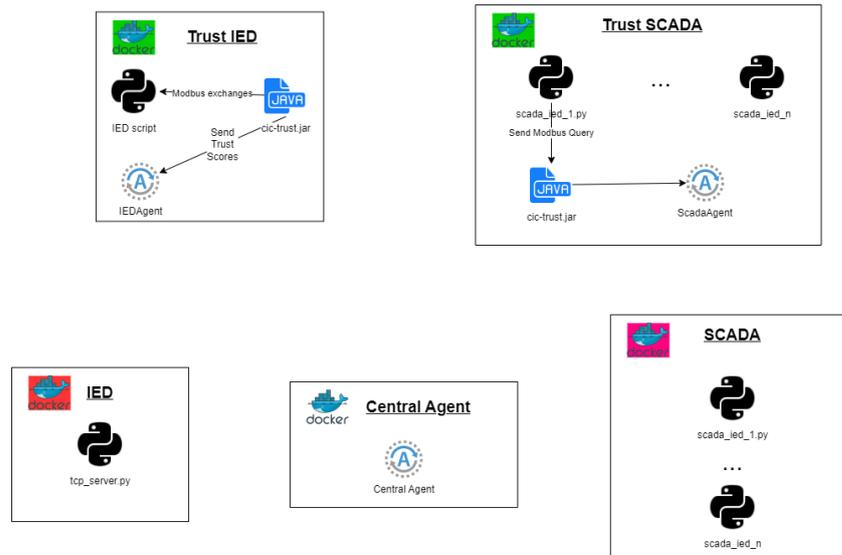


Figure 4. Docker structure of simulated devices.

The logs produced by the JAR files are gathered and presented visually through the utilization of the ELK stack. Figure 5 illustrates this architecture where all lines are bidirectional (with double arrows) with the exception of the ELK communication.

For the realization of a multi-agent system, we employed JADE. Within this framework, every secure device is equipped with an agent that engages in communication with a central agent. This central agent assumes the responsibility of computing the substation’s risk posture.

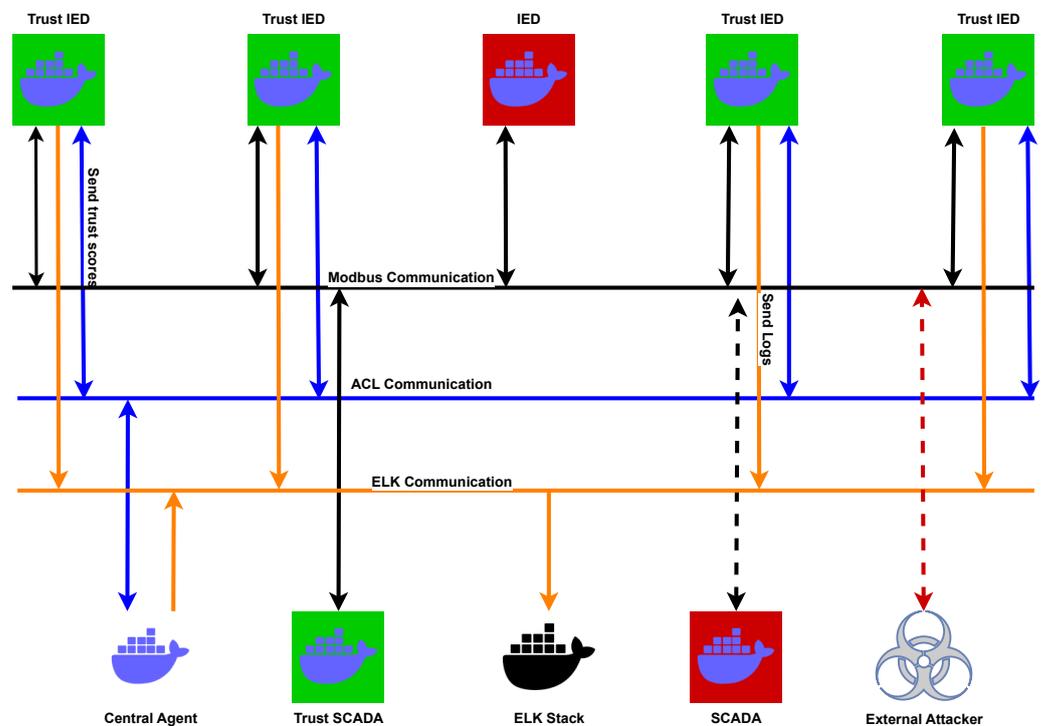


Figure 5. Architecture of containerized testbed.

We referenced a trust scale outlined in the existing literature [31] and aligned our generated trust scores with the Multi-State Information Sharing and Analysis Center (MS-ISAC) Alert Information [32], as depicted in Figure 6. The outcome obtained from

Equation (22) is further mapped to the NIST Risk Impact Assessment Scale[33] as depicted in Figure 7.

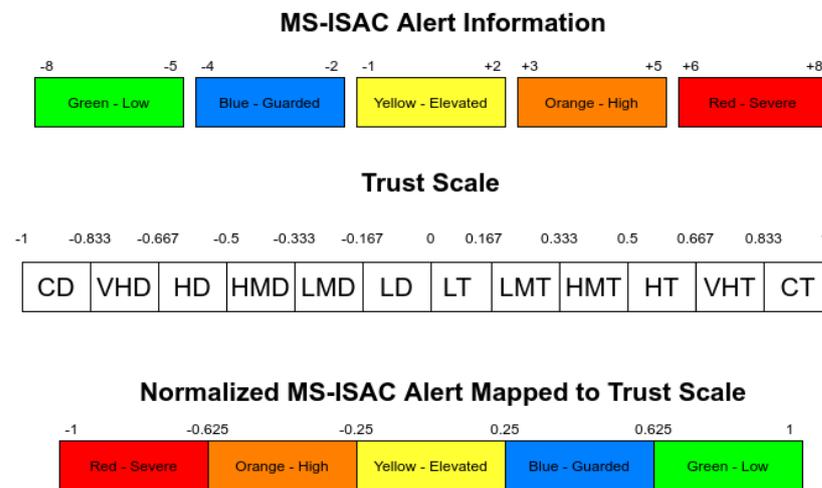


Figure 6. MS–ISAC alert mapped to trust scale

The trust IEDs are capable of inhibiting communication with a malicious device for a predetermined interval. Attacks are scheduled randomly, accompanied by a backoff time that surpasses the blocked period subsequent to each attack. In situations where the inflow of incoming messages results in a potentially flooded message queue, the trust IEDs exclusively transmit updates to the central agent when alterations in trust scores occur.

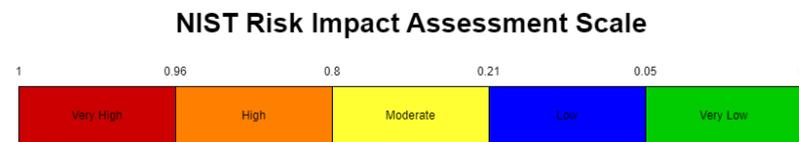


Figure 7. NIST risk impact assessment scale.

Prior to conducting tests on our model, the following assumptions were established:

- The network traffic within the substation is predictable due to predefined queries issued by engineers.
- Attacks unrelated to Modbus or IT are addressed using various Common Vulnerabilities and Exposures (CVE) and Common Weakness Enumerations (CWE) mitigation techniques, and as such, they fall beyond the scope of this paper.
- Attackers are limited to manipulating Modbus packets due to the vendor’s robustness against TCP, IP, and Ethernet frame manipulations.
- Devices are confined to utilizing the Modbus port number for network communication.
- We assume that the control center has been compromised without any corresponding detections being made.

Utilizing data sourced from Boakye-Boateng et al. [28], we present Table 4, which encapsulates the functional influence of the IEDs. The value of $C_T(D)$ is recorded as 598. However, due to constraints imposed by page limitations, the functional influence of other devices has not been included in the table. Additionally, IEDs sharing identical dependency counts have been grouped within the same row, again owing to page restrictions. For purposes of our analysis, we have classified IED1A and IED4C as trust IEDs, while designating IED1B as a regular IED. This classification enables us to observe the substation’s risk posture under different scenarios, such as an attempt to compromise the trust IEDs or the compromise 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 have considered two distinct scenarios to evaluate the risk posture of the substation, based on the Docker-based MAS architecture presented:

- Scenario 1: Trust SCADA HMI Control—In this test, the substation is controlled solely by a trust SCADA HMI container. The attacks will be executed through two different methods. The first attack involves an adversary utilizing their own device to launch an attack on the system. The second type of attack will be simulated by employing regular IEDs to replicate a compromised IED scenario.
- Scenario 2: Regular SCADA Control—For the second test, a regular SCADA container (as indicated by the dotted lines) is employed to manage all the IEDs. As in the first scenario, attacks will be conducted in two ways. The first attack mirrors the approach taken in the initial test. The second type of attack will be orchestrated from the regular SCADA to mimic scenarios that are publicly documented.

These scenarios aim to assess the substation's risk posture under varying conditions, shedding light on potential vulnerabilities and providing insights into the efficacy of the proposed architecture.

To evaluate transferability within our environment, we have devised three distinct scenarios. In each of these scenarios, the replacement or existing trust IED demonstrates well-behaved behavior, indicated by trust scores categorized as *Green—Low*. The scenarios are delineated as follows:

- Scenario 1: Normal Replacement—A new trust IED is introduced, replacing an existing trust IED, and it operates as expected, exhibiting normal behavior.
- Scenario 2: Compromised Replacement (Immediate)—A new compromised IED replaces an existing trust IED, but after acceptance, it begins to exhibit malicious behavior.
- Scenario 3: Compromised Replacement (Delayed)—Similar to Scenario 2, a new compromised IED takes the place of an existing trust IED. However, the malicious behavior emerges only after surpassing the consideration period, which constitutes half of the probation period.
- Scenario 4: Trust IED with Poor Trust Scores—A new trust IED is introduced, replacing an existing IED, but the trust scores associated with it are not deemed favorable.

These scenarios facilitate an exploration of transferability across different contexts, allowing us to assess the robustness and effectiveness of our approach under varying conditions.

10. Evaluation

In this section, we provide an overview of the outcomes yielded by our trust framework. Initially, we delve into the results concerning response times, drawing a comparison between regular IEDs and trust IEDs. Subsequently, we scrutinize the performance of both the trust model and the risk posture model under diverse attack scenarios. Following this analysis, we proceed to showcase the findings obtained from the trust transferability model.

10.1. Performance

In order to assess the response time of the trust IEDs and ensure minimal overhead, an examination was conducted. Several vendors permit response latencies ranging from 1000 milliseconds (ms) to as low as 4 ms between the station level and the bay level. Figure 8 provides a visual representation of the response times for all IEDs. In this context, IED1B functions as the regular IED, while IED1A and IED4C are classified as trust IEDs. Notably, Figure 8 illustrates that response times do not exceed 10 ms, thereby affirming the system's ability to maintain an acceptable level of performance.

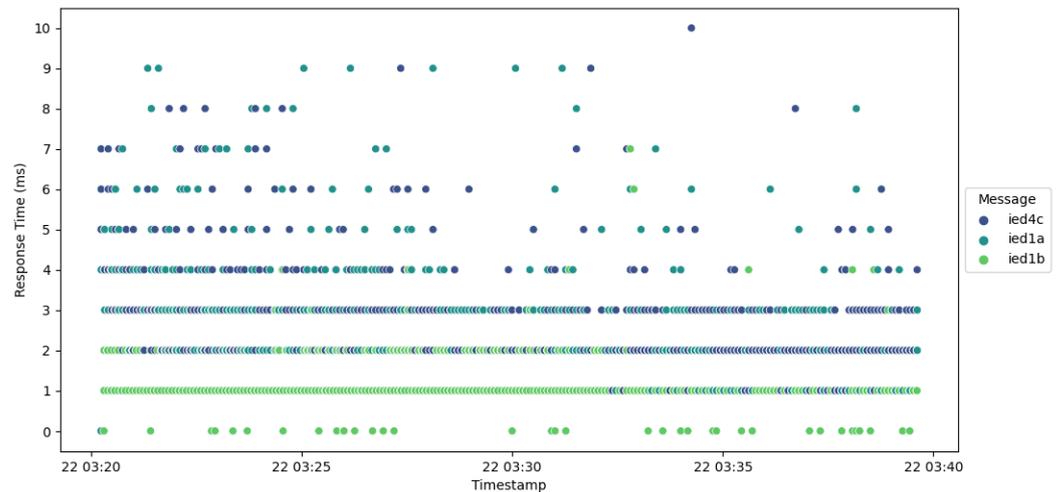


Figure 8. Plot of response time for all IEDs.

Figure 9 portrays the response time profile of IED1B, exhibiting a range spanning from around 0 ms to 7 ms. Instances where values surpass 2 ms typically stem from computational factors. This representation provides insight into the response behavior of IED1B and aids in the identification of potential anomalies or computational variations.

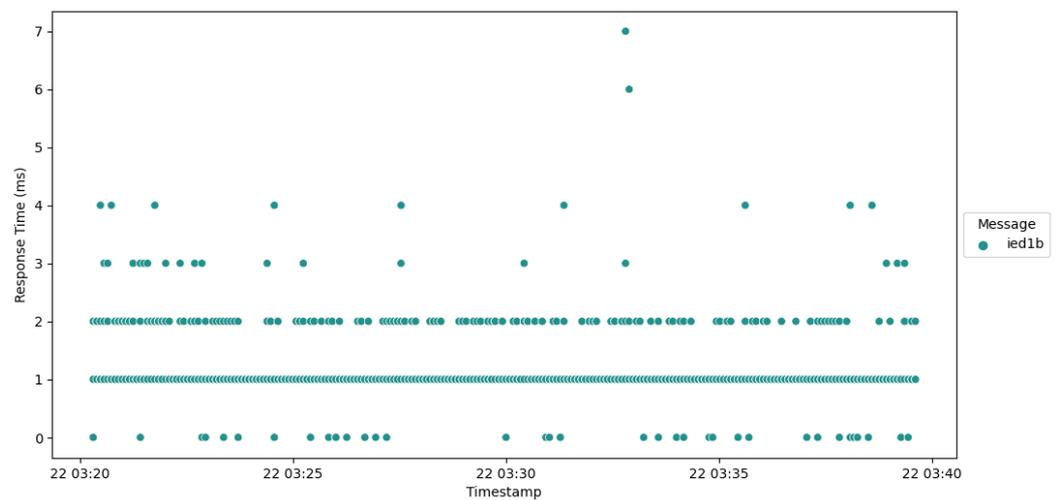


Figure 9. Plot of response time IED1B.

Analysis of the response times exhibited by IED1A (Figure 10) and IED4A (Figure 11) reveals that a majority of their communication instances are approximately 2 ms higher than those of IED1B, although still remaining within a limit of 10 ms. It is worth considering that the performance of these devices could potentially be enhanced by implementing native code execution rather than utilizing Java, which might result in improved response times.

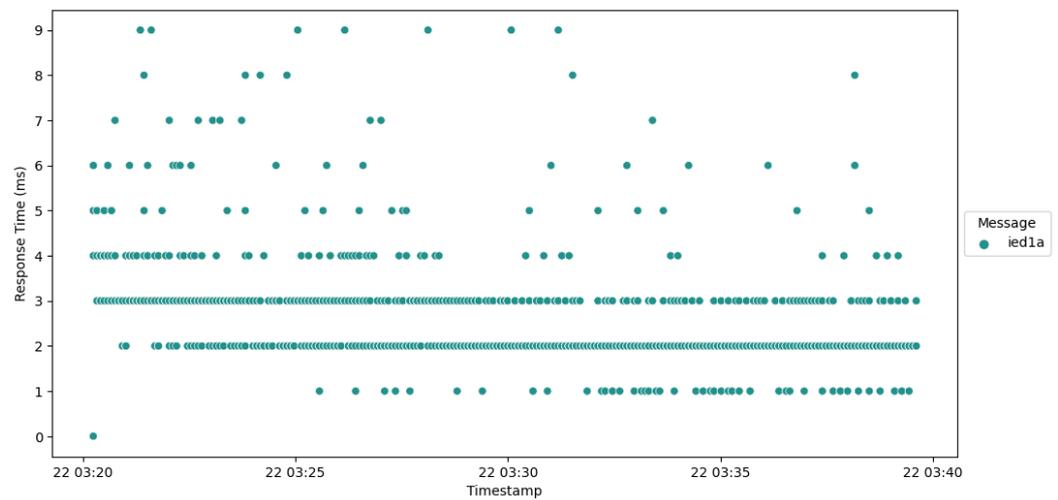


Figure 10. Plot of response Time IED1A.

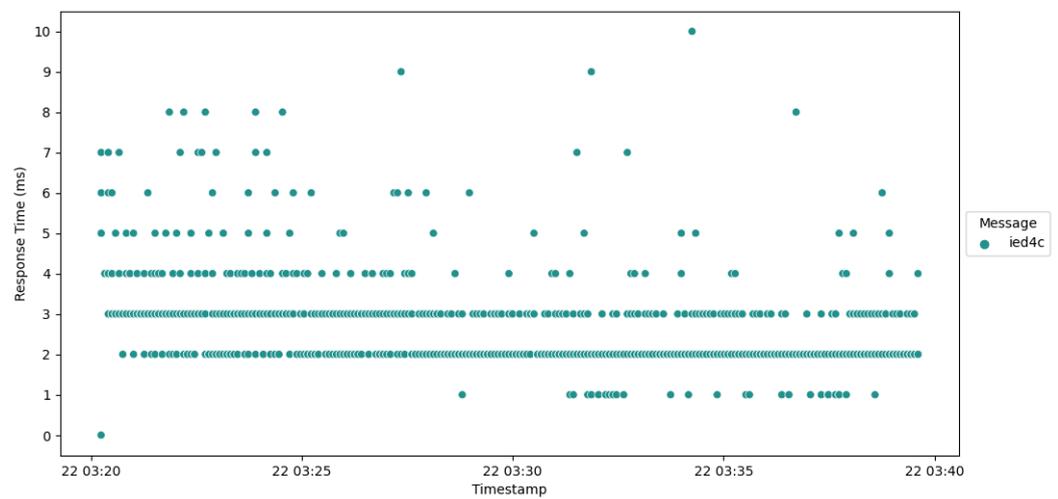


Figure 11. Plot of response time IED4C.

10.2. Risk Posture

In this section we present the results of the risk posture model tested under various attack scenarios.

10.2.1. Attack from Rogue Device

Irrespective of the nature of the attack targeted at the trust IEDs by the rogue device, all requests were flagged with unrecognizable IP addresses, as evidenced in Figure 12. Given the elevated rank of IED1A, an attempt involving IED1A led to the substation’s risk posture being categorized as “High”, as depicted in Figure 13. Conversely, an attempt targeting IED4C resulted in a shift to a “Low” risk posture for the substation due to IED4C’s lower rank, showcased in Figure 14. These observations underscore the influence of device rank on the resultant risk posture under varying attack scenarios. A summary of the results is presented in Table 5.

10.2.2. Attack from Compromised SCADA HMI Automated Collection

Within this approach, endeavors were undertaken to read all registers. As illustrated in Figure 15, the trust IED promptly identifies the initial packet and designates it as an *Unknown Read Query*, subsequently moving to blacklist the compromised HMI. Simultaneously, the trust IED alters its trust level to *Severe*. Notably, it is observed that the consequence

factor is influenced, with other metrics remaining uncalculated. Drawing from the received trust score, the central agent modifies the substation’s risk posture to *High*, as showcased in Figure 16, in response to the attack directed at IED1A. Similar repercussions are evident in the case of IED4C, with its trust level being adjusted to *Severe*, as depicted in Figure 17. Nonetheless, the substation’s risk posture undergoes a transition from *Very Low* to *Low*, as portrayed in Figure 18, reflecting the impact of IED4C’s relatively lower rank.

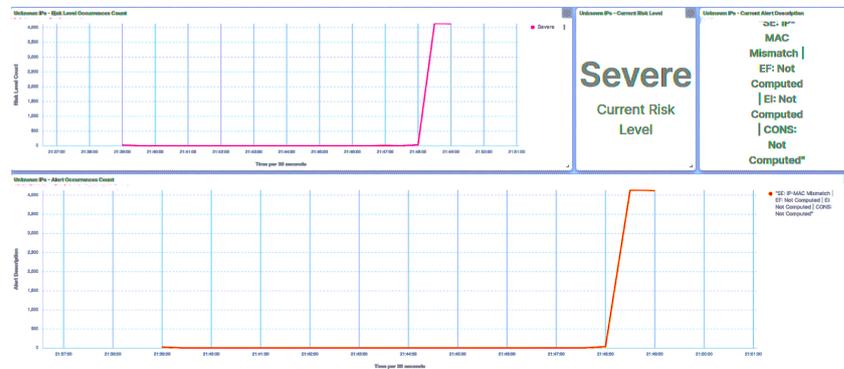


Figure 12. IED1A UI—rogue attack.

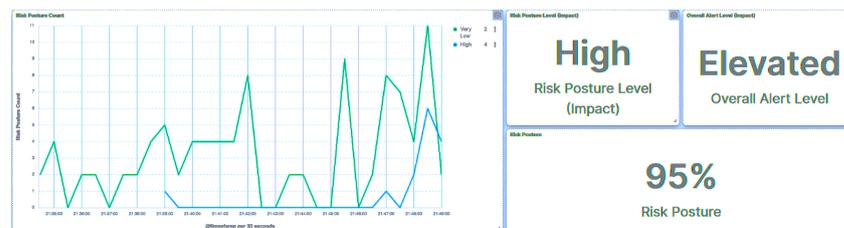


Figure 13. Central agent UI—rogue attack on IED1A.

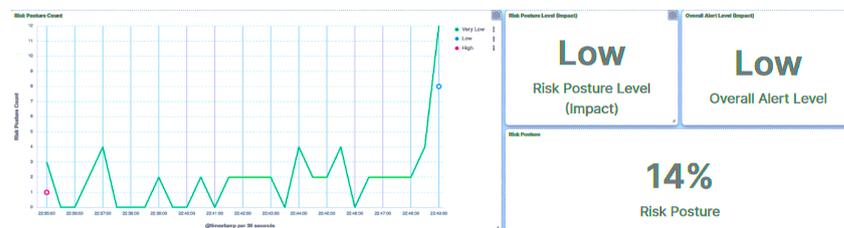


Figure 14. Central agent UI—rogue attack on IED4C.

Table 5. Results for attack from rogue device.

Trust Device	Attack	Alert	Affected Exposure	Device Risk Level	Risk Posture	Outcome
IED1A	Load malicious payload	IP Mismatch	Similarity	Severe	High	Rogue Device Blocked
IED4C	Load 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

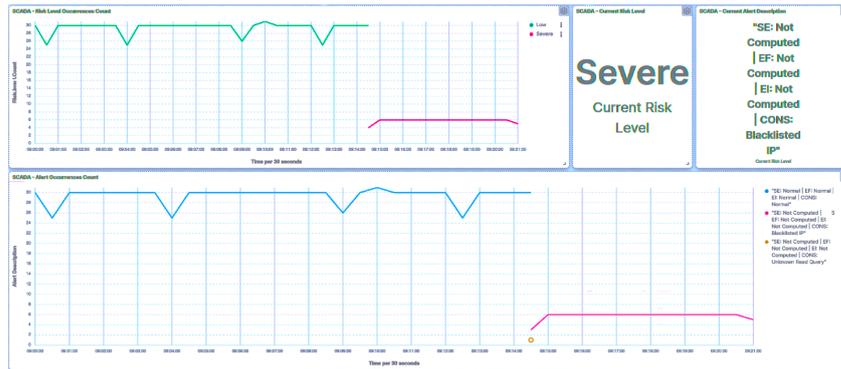


Figure 15. IED1A UI—SCADA: reconnaissance detection.

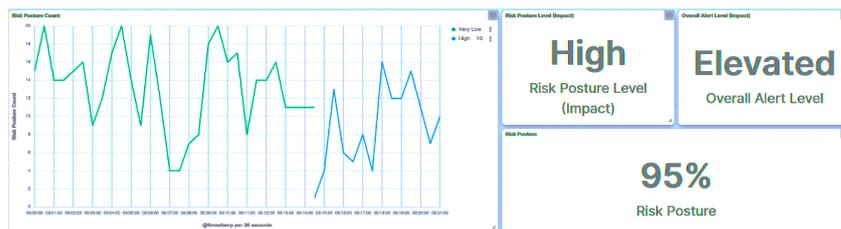


Figure 16. Central agent UI—SCADA: reconnaissance on IED1A.

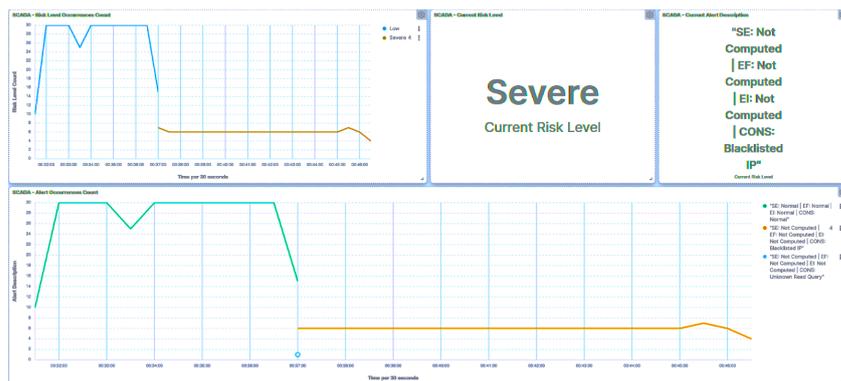


Figure 17. IED4C UI—SCADA: reconnaissance detection.

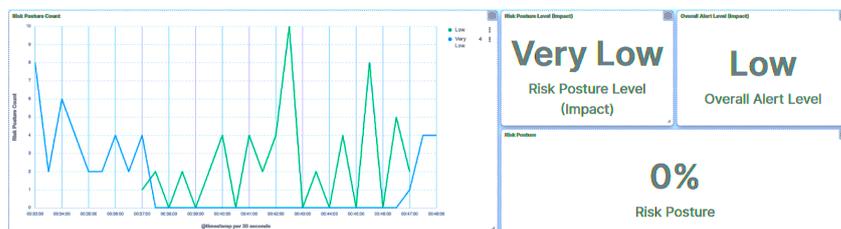


Figure 18. Central agent UI—SCADA: reconnaissance on IED4C.

Denial of Service

In the context of this scenario, three distinct denial of service (DoS) attacks were evaluated: query flooding, payload injection, and Modbus frame stacking. A known query was employed to inundate the trust IEDs for the query flooding attack. In this scenario, packets were successfully identified as malicious, as illustrated in Figure 19. One of these packets was flagged as out of sequence due to out-of-order arrival (marked as *Sequence Mismatch*).

The another packet was flagged as a query flooding as it exceeded the threshold indicator for query flooding. In response to these detected anomalies, the trust IED adjusted its trust level to *Severe*, leading to the subsequent blacklisting of the associated SCADA HMI.

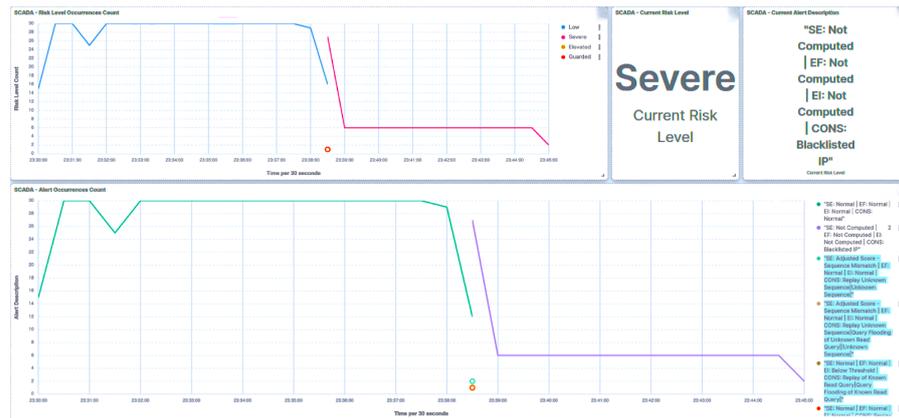


Figure 19. IED1A UI—SCADA: flooding detection.

Utilizing the received trust score, the central agent promptly updates the substation’s risk posture to *High*, as depicted in Figure 20, in response to the attack directed at IED1A. A similar outcome unfolds in the case of IED4C, whereby the attack triggers a shift in its trust level to *Severe*, showcased in Figure 21. Nevertheless, the substation’s risk posture undergoes a transition from *Very Low* to *Low*, as illustrated in Figure 22, due to IED4C’s relatively lower rank. It must be noted that in the IED’s dashboard (and SCADA HMI’s as well), the alert occurrence count and the risk level occurrence count for attacks match to demonstrate the consistency of the model.

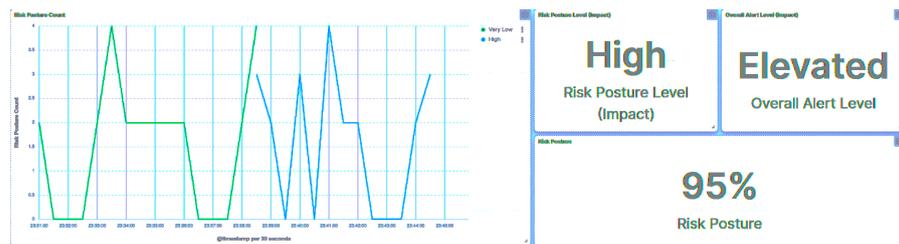


Figure 20. Central agent UI—SCADA: flooding on IED1A.

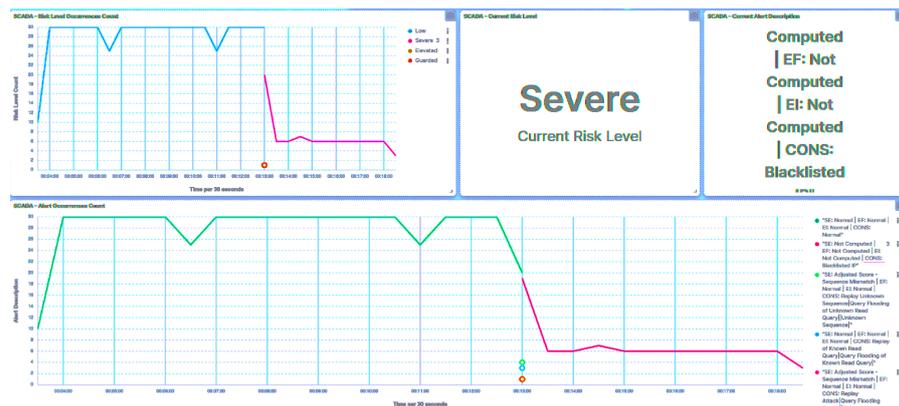


Figure 21. IED4C UI—SCADA: flooding detection.

During the payload injection attack, a payload was introduced into a Modbus packet and transmitted to the trust IEDs. Subsequently, the packet was detected and marked with a *Length Mismatch* indicator. The implicated HMI was consequently blacklisted, a depiction of which is presented in Figure 23. Following the detection of this malicious event, the trust IED promptly adjusts its trust level to *Severe*.

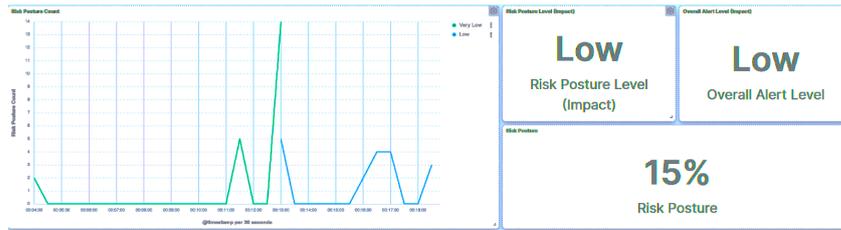


Figure 22. Central agent UI—SCADA: flooding on IED4C.

Reflecting the implications of this attack, the substation’s risk posture is elevated to *High*, as demonstrated in Figure 24, as a direct result of the incident involving IED1A. Analogously, IED4C’s trust level experiences a shift to *Severe*, a manifestation captured in Figure 25. Despite this, the substation’s overall risk posture undergoes a transition from *Very Low* to *Low*, as illustrated in Figure 26, owing to IED4C’s relatively lower rank within the hierarchy.

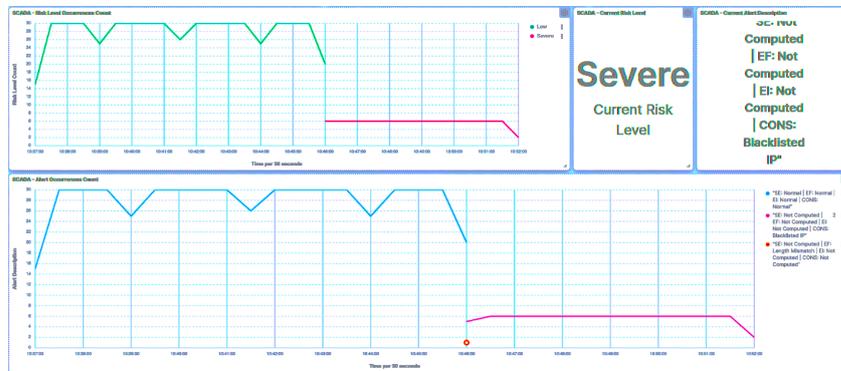


Figure 23. IED1A UI—SCADA: payload detection.

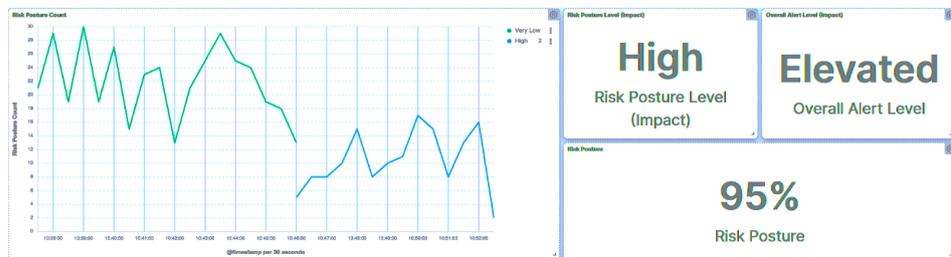


Figure 24. Central Agent UI—SCADA: payload on IED1A.

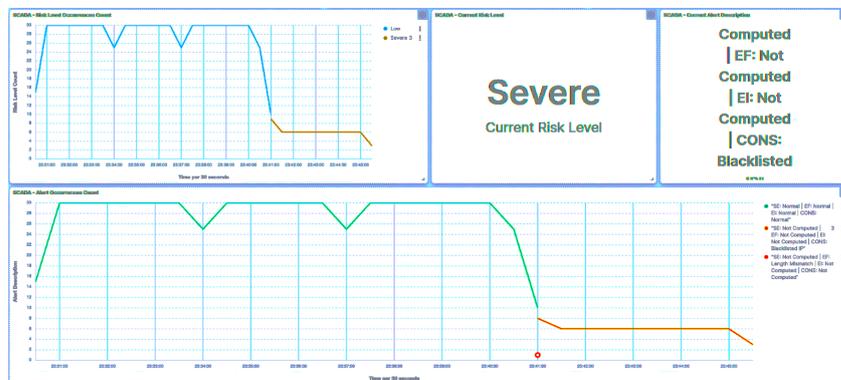


Figure 25. IED4C UI—SCADA: payload detection.



Figure 26. Central agent UI—SCADA: payload on IED4C.

The Modbus frame stacking attack involved stacking Modbus frames and transmitting them to the trust IEDs. As delineated in Figure 27, the outcomes parallel those observed during the payload injection attack. Following the detection of the malicious event, the trust IED promptly designates its trust level as *Severe*.

In response to this incident, the substation’s risk posture is elevated to *High*, as depicted in Figure 28, attributable to the attack targeted at IED1A. Analogously, the same attack leads to IED4C’s trust level transitioning to *Severe*. Furthermore, the substation’s overall risk posture undergoes a shift from *Very Low* to *Low* due to IED4C’s lower hierarchical ranking.

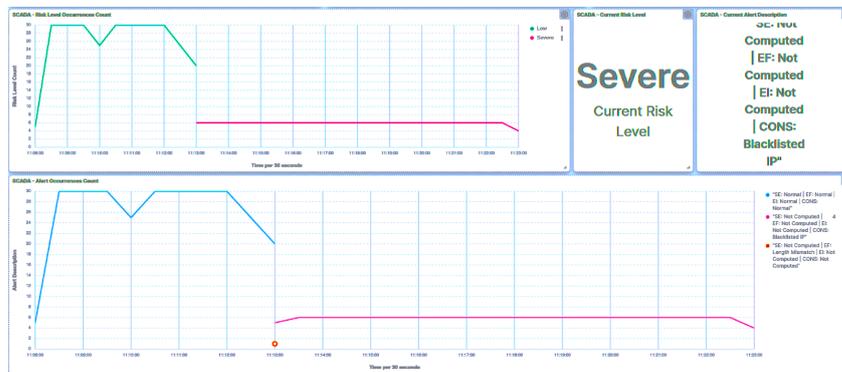


Figure 27. IED1A UI—SCADA: frame stacking detection.

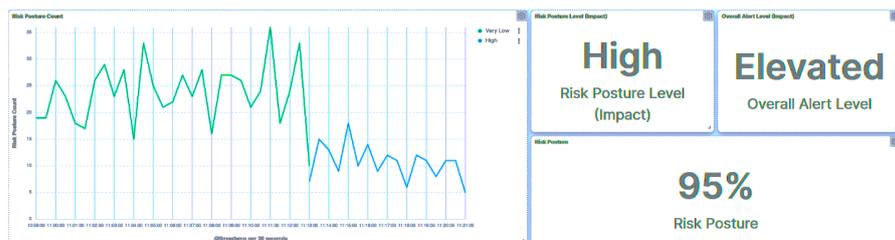


Figure 28. Central agent UI—SCADA: frame stacking on IED1A.

In the context of the replay attack, authentic queries were directed at the trust IEDs. Notably, the initial query, categorized as a write query, was flagged with an *Environment Attack. APT Threat*, as depicted in Figure 29. Swiftly responding to this detection, the trust IED designates its trust level as *Severe*.

In light of this incident, the substation’s risk posture is elevated to *High*, as demonstrated in Figure 30, owing to the attack on IED1A. A parallel outcome will also be evident in the case of IED4C, where the attack prompts a shift in trust level to *Severe* (Figure 31). However, the substation’s broader risk posture will transition from *Very Low* to *Low* (Figure 32), as this is attributed to IED4C’s lower hierarchical ranking.

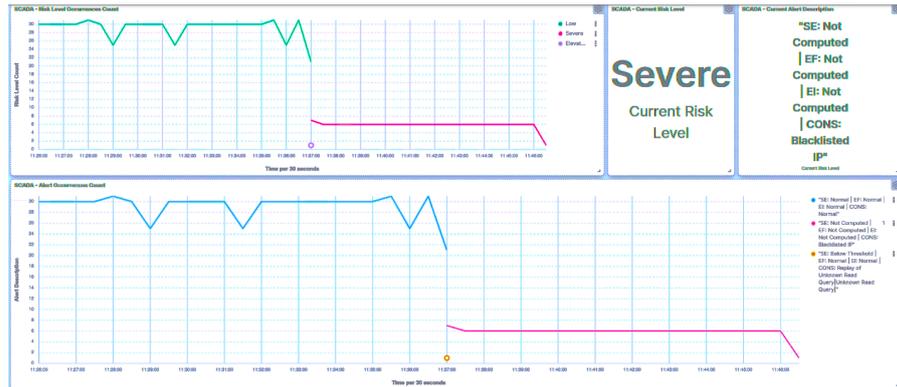


Figure 29. IED1A UI—SCADA: replay packets detection.

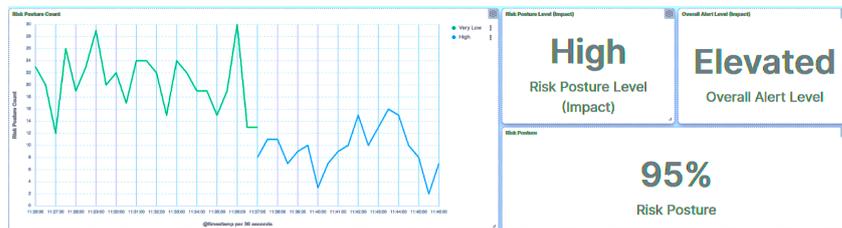


Figure 30. Central agent UI—SCADA: replay packets detection on IED1A.

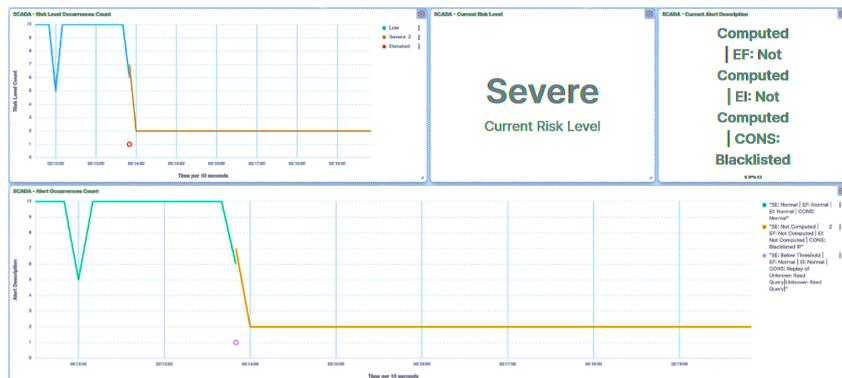


Figure 31. IED4C UI—SCADA: replay packets detection.

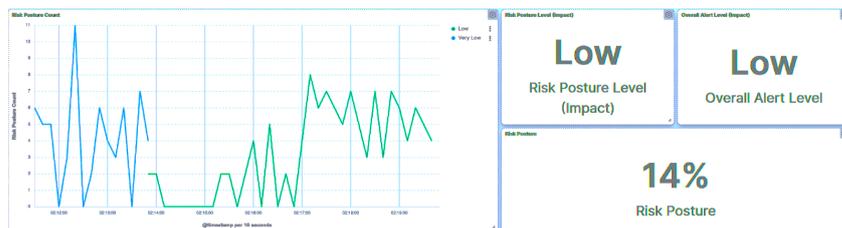


Figure 32. Central agent UI—SCADA: replay packets detection on IED4C.

Brute Force I/O

Queries were systematically directed to all addresses within the trust IEDs, specifically targeting write operations. In this context, the initial packet of this sequence was promptly identified and categorized as an *Unknown Write Query Attack*, as depicted in Figure 33. This detection triggered a swift response from the trust IED, which promptly elevated its trust level to *Severe* in acknowledgment of the detected malicious activity.

Simultaneously, the substation’s risk posture underwent a significant escalation, reaching a classification of *High*, as illustrated in Figure 34. This heightened risk assessment directly corresponds to the attack directed at IED1A. It is noteworthy that employing the

same attack strategy against IED4C would result in a similar outcome, leading to a *Severe* trust level designation for IED4C (Figure 35) and prompting a transition in the substation’s risk posture from *Very Low* to *Low* (Figure 36). A summary of the results is presented in Table 6.

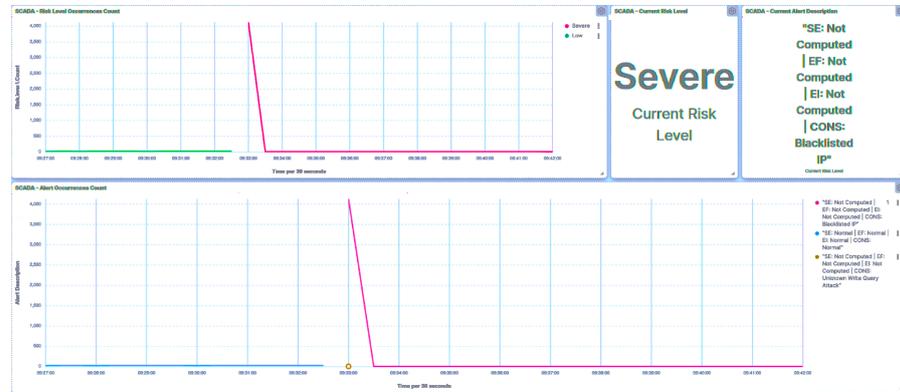


Figure 33. IED1A UI—SCADA: brute force I/O detection.



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

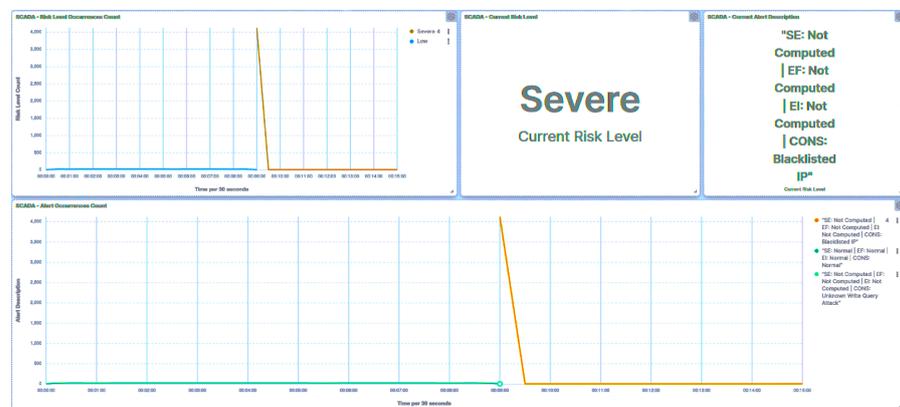


Figure 35. IED4C UI—SCADA: Brute Force I/O detection.

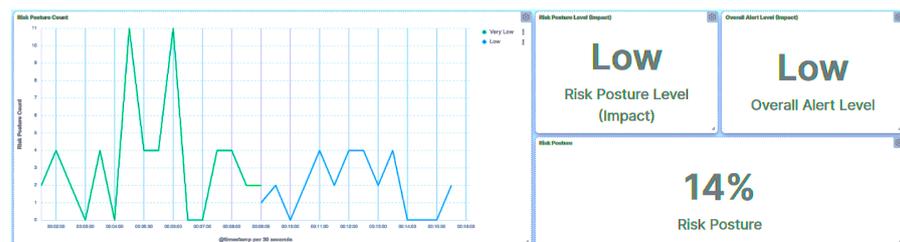


Figure 36. Central agent UI—SCADA: brute force I/O on IED4C.

Table 6. Results for attack from compromised SCADA HMI.

Trust Device	Attack	Alert	Affected Component	Device Risk Level	Risk Posture	Outcome
IED1A	Load malicious payload	Length mismatch	Frequency	Severe	High	Communication with SCADA HMI blocked
IED1B	Load malicious payload	Length mismatch	Frequency	Severe	Low	Communication with SCADA HMI blocked
IED1A	Query flooding	Query flooding of known read query	Intensity	Severe	High	Communication with SCADA HMI blocked
IED1B	Query flooding	Query flooding of known read query	Intensity	Severe	Low	Communication with SCADA HMI blocked
IED1A	Reconnaissance	Unknown read query	Consequence	Severe	High	Communication with SCADA HMI blocked
IED1B	Reconnaissance	Unknown read query	Consequence	Severe	Low	Communication with SCADA HMI blocked
IED1A	Replay packets	Replay of unknown read query	Consequence	Severe	High	Communication with SCADA HMI blocked
IED1B	Replay packets	Replay of unknown read query	Consequence	Severe	Low	Communication with SCADA HMI blocked
IED1A	Stack modbus frames	Length mismatch	Frequency	Severe	High	Communication with SCADA HMI blocked
IED1B	Stack modbus frames	Length mismatch	Frequency	Severe	Low	Communication with SCADA HMI blocked
IED1A	Write to all coils	Unknown write query attack	Consequence	Severe	High	Communication with SCADA HMI blocked
IED1B	Write to all coils	Unknown write query attack	Consequence	Severe	Low	Communication with SCADA HMI blocked

10.2.3. Attack from Compromised IED

Denial of Service

In a manner akin to the attacks originating from the compromised SCADA HMI, the response stemming from the compromised IED1B exhibits a similar behavior of Modbus frames being stacked together within a single response. Upon receipt of this response, the trust SCADA HMI undertakes an evaluation and identifies that it is a malicious packet, thereby generating a *Length Mismatch* alert, as visualized in Figure 37. This response initiates the blacklisting of IED1B and the classification of IED1B as a *Severe* risk entity.

Concurrently, the central agent responsible for risk assessment and management maintains a consistent response pattern by assigning the substation’s risk posture to a classification of *High*, as demonstrated in Figure 38.



Figure 37. SCADA UI—IED1B: stacking detection.



Figure 38. Central agent UI—IED1B: stacking.

In the context of the length manipulation attack, a deliberate alteration is made to the Modbus length field. This manipulation leads to a significant divergence between the value specified within the length field and the actual length of the Modbus payload. As a direct consequence of this mismatch, an anomaly is promptly detected and flagged as a *Length Mismatch* (as evidenced in Figure 39). The trust SCADA HMI promptly designates IED1B as a *Severe* risk. Due to elevated risk associated with IED1B’s compromised state, the central agent adjusts the substation’s risk posture to *High* (as depicted in Figure 40).



Figure 39. SCADA UI—IED1B: length manipulation detection.

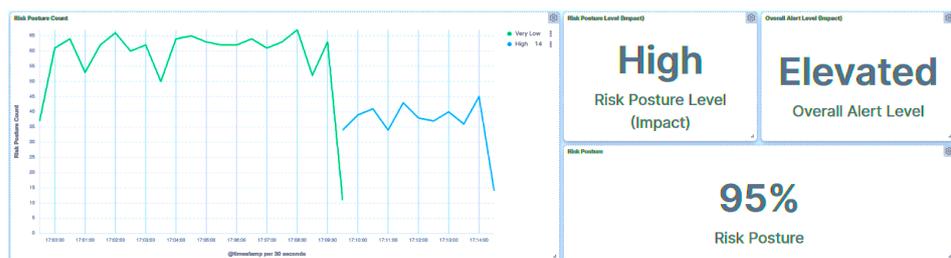


Figure 40. Central agent UI—IED1B: length manipulation.

The payload injection attack involves a strategic insertion of payload bytes into the Modbus packet, which is subsequently transmitted within the system. This manipulation leads to a notable incongruence between the designated length within the Modbus packet and the actual length of the payload. The resulting discrepancy is identified and labeled as a *Length Mismatch*, as visually represented in Figure 41.

The trust SCADA HMI promptly designates IED1B as a *Severe* risk, signifying a compromised state that warrants immediate attention and mitigation measures. Recognizing the escalated risk associated with IED1B’s compromised state, the central agent updates the risk posture of the entire substation, classifying it as *High* (see Figure 42).



Figure 41. Trust SCADA HMI UI—IED1B: payload injection detection.



Figure 42. Central agent UI—IED1B: payload injection.

The false data injection attack requires modifying the data field of the Modbus packet. This manipulated packet is then transmitted within the network, with the intention of introducing false data into the system. This action triggers an alert mechanism, specifically an *Unknown Read Command* alert, as illustrated in Figure 43.

The trust SCADA HMI classifies IED1B as a *Severe* risk. The synchronization of trust scores with the central agent causes the central agent to dynamically update the substation's risk posture, designating it as *High* (see Figure 44).

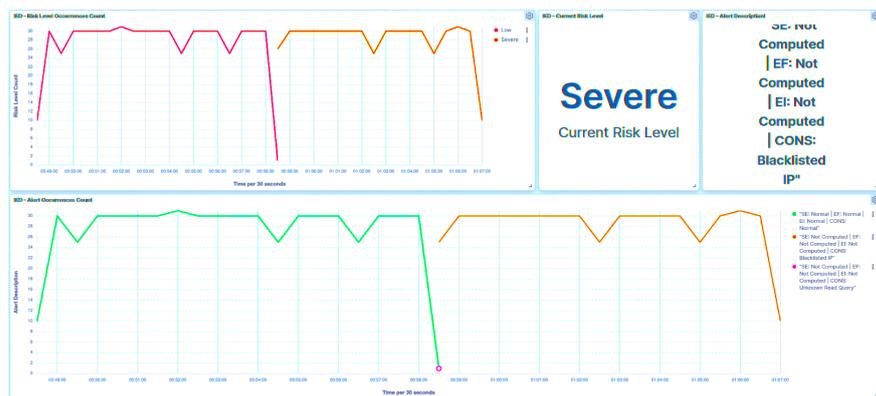


Figure 43. SCADA UI—IED1B: false data injection detection.

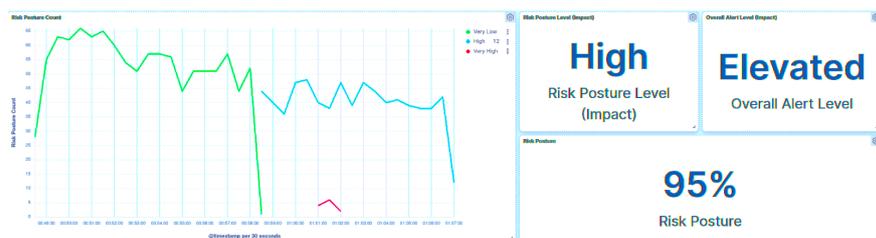


Figure 44. Central agent UI—IED1B: false data injection.

In the context of the delay response attack, IED1B intentionally introduces a deliberate delay in transmitting its response, encompassing a time range of 30 s to 1 min. This alteration disrupts the expected timing of communication, potentially impacting the overall operational efficiency and responsiveness of the system.

Notably, the trust SCADA HMI was unable to effectively flag the delayed response as a malicious event. The rationale behind this lies in the relative moderation of the delay introduced.

Spoof Reporting Message

In the context of a baseline replay attack, IED1B orchestrates the transmission of a response to the trust SCADA HMI without introducing any modifications to the original content. This replication of the required response aims to mimic genuine communication between the IED and the HMI, with the intention of evading detection by the trust SCADA HMI.

As illustrated in Figure 45, the trust SCADA HMI receives the replayed response and, crucially, does not identify any alterations or anomalous attributes within the transmitted data. As a result, the trust SCADA HMI fails to flag the response as malicious, as it perceives the received communication as consistent with normal operational behavior.

The lack of detection by the trust SCADA HMI subsequently influences the risk assessment conducted by the central agent, as depicted in Figure 46. Given the absence of any indication of suspicious activity or malicious intent, the substation's risk posture remains categorized as low by the central agent, as the attack successfully mimics legitimate communication patterns and avoids raising any alarms.

The time analysis of the provided data reveals a notable incident occurring between 10:20 p.m. and 10:22 p.m., during which the SCADA HMI tagged, as IED1B underwent an *Elevated* state. This transition in the state of IED1B had a corresponding impact on the overall risk posture of the substation, which was promptly updated to *High* during the same time frame.

During the aforementioned time window, IED1B reported a voltage measurement of 0. This reading signified that a circuit breaker (CB) had tripped, resulting in the interruption of voltage flow through one of the primary sources within the substation.

As a consequence of this CB trip and the resultant absence of voltage in one of the critical circuits, the trust SCADA HMI identified an abnormal condition, subsequently classifying IED1B as *Elevated*. This elevated state, in turn, triggered an adjustment in the substation's risk posture to *High*, as the anomalous situation signaled a potential disruption to the normal operational state of the substation. A summary of all results is presented in Table 7.



Figure 45. SCADA UI—IED1B: baseline replay detection.



Figure 46. Central agent UI—IED1B: baseline replay detection.

Table 7. Results for attack 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 blocked
IED1B	False data injection	Unknown read query	Consequence	Severe	Low	Communication with IED blocked
IED1A	Length manipulation	Length mismatch	Intensity	Severe	High	Communication with IED blocked
IED1B	Length manipulation	Length mismatch	Intensity	Severe	Low	Communication with IED blocked
IED1A	Load malicious payload	Length mismatch	Frequency	Severe	High	Communication with IED blocked
IED1B	Load malicious payload	Length mismatch	Frequency	Severe	Low	Communication with SCADA HMI blocked
IED1A	Stack modbus frames	Length mismatch	Frequency	Severe	High	Communication with IED blocked
IED1B	Stack modbus frames	Length mismatch	Frequency	Severe	Low	Communication with IED blocked

10.3. Transferability

10.3.1. Scenario 1—Good Behavior

Figure 47 provides a visualization of the transferability process involving IED4C, as recorded by the central agent. Each message depicted in the figure corresponds to a distinct event that was logged by the central agent, offering a sequential depiction of the entire process.

At the outset of the transferability process, IED4C initiates a disconnection request directed towards the central agent. This event is logged and labeled as *Start Disconnect*, marking the start of the transferability procedure.

IED4C proceeds to share its trust scores with the central agent. These trust scores, alongside the scores generated by the trust HMI for IED4C, are stored by the central agent. This action of sharing and storing trust scores is captured in the event log as *Scores Shared*.

Subsequently, the central agent directs IED4C to initiate the disconnection process, thereby prompting it to execute the disconnection procedure. Upon successful execution of the disconnection, the central agent logs this event as *Disconnect Complete*.

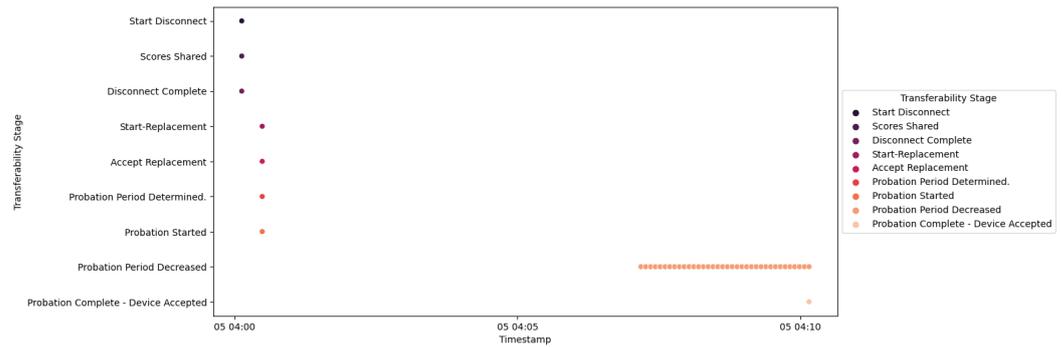


Figure 47. IED4C Scenario 1—transferability process.

The disconnection of IED4C from the network introduces a temporary interruption or gap in the continuous computation of the *Risk Level* by the trust SCADA HMI, as evidenced in Figure 48.

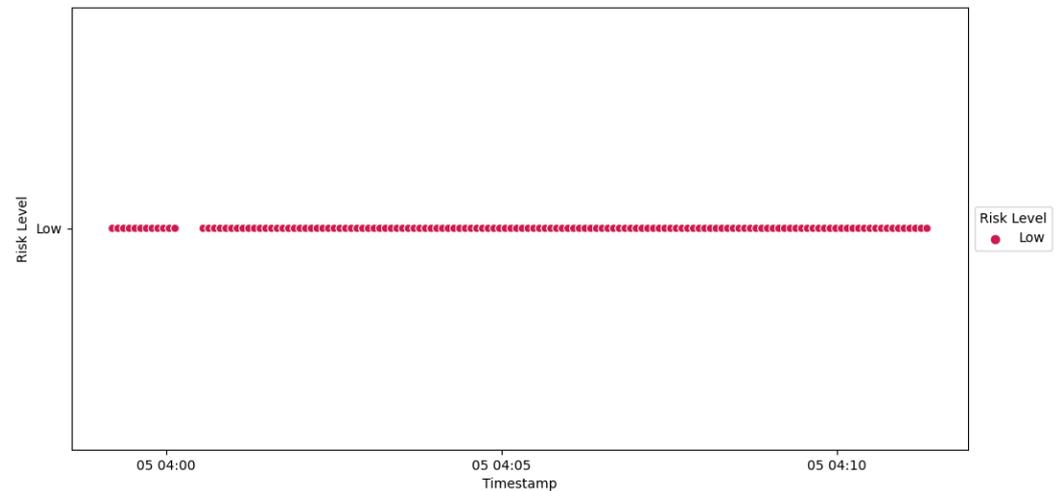


Figure 48. IED4C Scenario 1—trust SCADA HMI scores.

Revisiting the sequence of events illustrated in Figure 47, the initiation of the transferability process is marked by the *Start Replacement* event. This event signifies the commencement of the process wherein a new trust IED requests to replace IED4C. The new trust IED transmits its own trust scores to the central agent. The central agent utilizes both the trust scores provided by the new trust IED and the historical trust scores associated with IED4C to evaluate whether the new trust IED is suited for integration into the substation’s network.

The *Probation Period Determined* event indicates that the central agent has computed the probation period and the probation point. The *Probation Started* event signifies the commencement of the probation period, during which the new trust IED’s behavior is monitored.

As the probation period evolves into the consideration period, every good behavior exhibited by the replacement IED is rewarded with a probation point reduction in the remaining probation time. This is marked by the *Probation Period Decreased* event.

The *Probation Complete—Device Accept* event signifies the successful acceptance of the replacement IED into the substation’s network, marking the culmination of the transferability process. The substation’s risk posture remains consistently categorized as *Very Low* (Figure 49) throughout the process, indicating the stability and effectiveness of the transferability framework. This outcome is applicable to both IED4C and IED1A, reaffirming the process’s reliability regardless of device rankings.

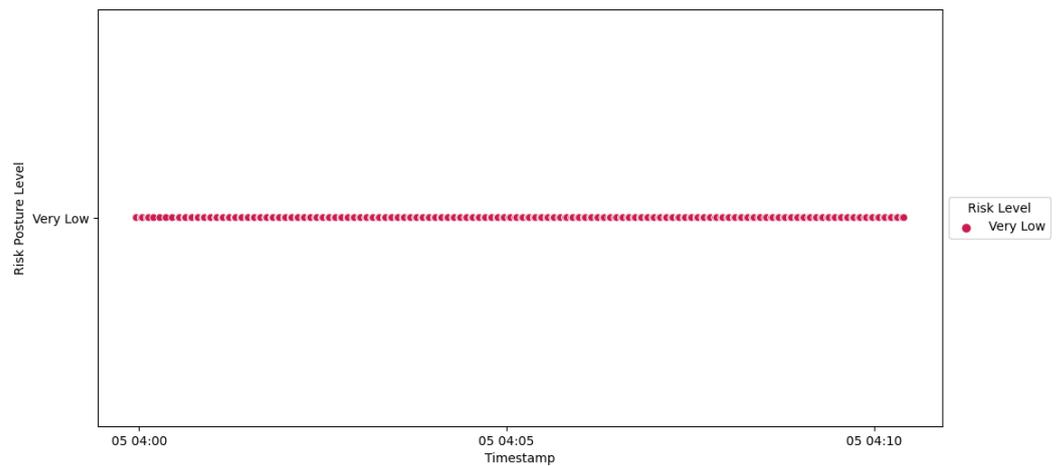


Figure 49. IED4C scenario 1—substation’s risk posture.

10.3.2. Scenario 2—Misbehavior after Probation Acceptance

In Figure 50, the succession of events unfolding from *Start Disconnect* to *Probation Started* mirrors the course of actions witnessed in Scenario 1. The processes remain consistent across both scenarios up to the point of *Probation Started*, encompassing stages such as disconnection and probation initiation. A communication gap with the trust SCADA HMI akin to that depicted in Figure 51 is similarly observed in Scenario 2 during the disconnection phase. After the *Probation Started* event, the new trust IED engages in malicious conduct by launching a Modbus frame stacking attack during its interaction with the trust SCADA HMI.

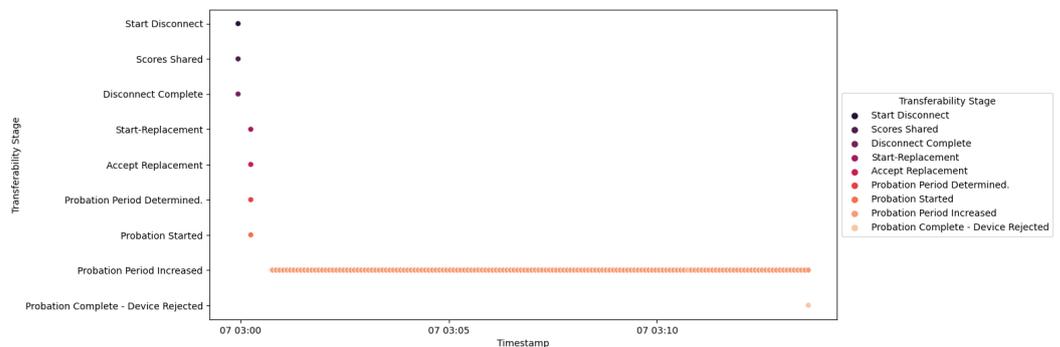


Figure 50. IED4C Scenario 2—transferability process.

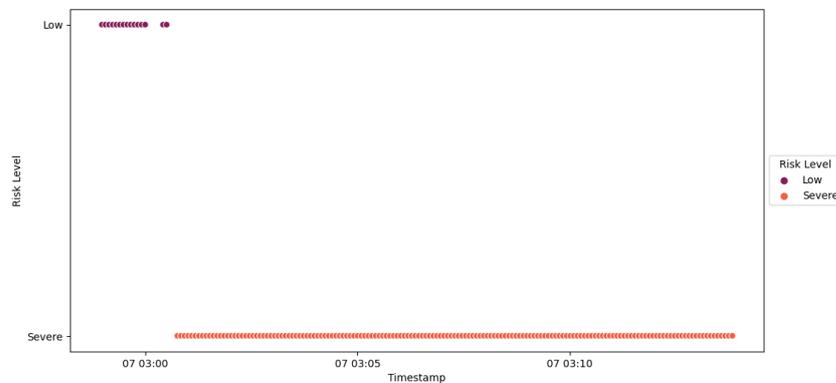


Figure 51. IED4C Scenario 2—trust SCADA HMI scores.

After the Modbus frame stacking attack was detected by the HMI, IED4C was black-listed (Figure 52), and its risk level was increased from *Low* to *Severe*. The central agent

responded by adding probation points to the ongoing probation period (Figure 50), leading to a change in the substation’s risk posture from *Very Low* to *Low* (Figure 53). This adjustment was influenced by IED4C’s lower rank.

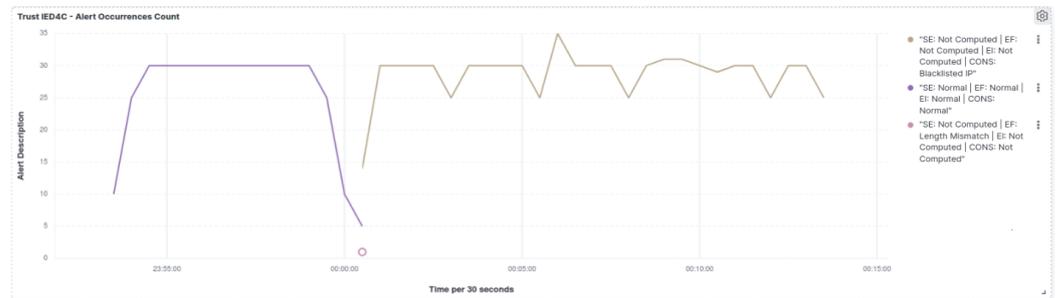


Figure 52. IED4C Scenario 2—SCADA dashboard.

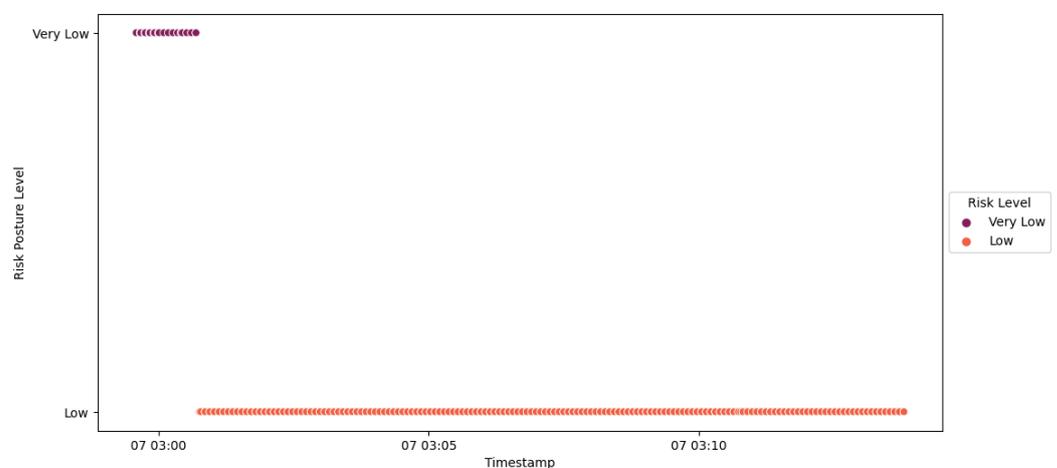


Figure 53. IED4C Scenario 2—substation’s risk posture.

Additionally, the probation period was consistently extended in the absence of non-malicious communication from the new trust IED, as depicted in Figure 50. Furthermore, the consideration period was disregarded. The occurrence of the *Probation Complete—Device Rejected* event indicated that the initial probation period had expired before the revised probation period, resulting in the device being rejected.

A similar process was repeated for IED1A, mirroring the behavior observed with IED4C, as depicted in Figure 54. The trust SCADA HMI escalated the risk level of IED1A from *Low* to *Severe*, as illustrated in Figure 55. However, due to the elevated rank of IED1A, the substation’s risk posture was elevated from *Very Low* to *High*, as indicated in Figure 56.

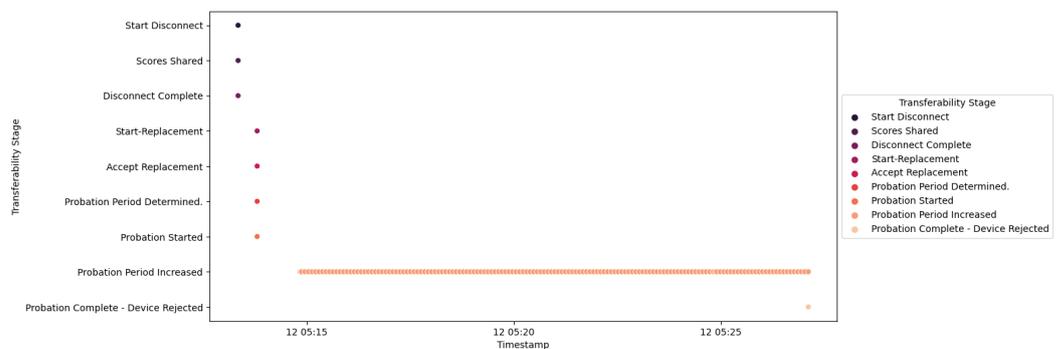


Figure 54. IED1A Scenario 2—transferability Process.

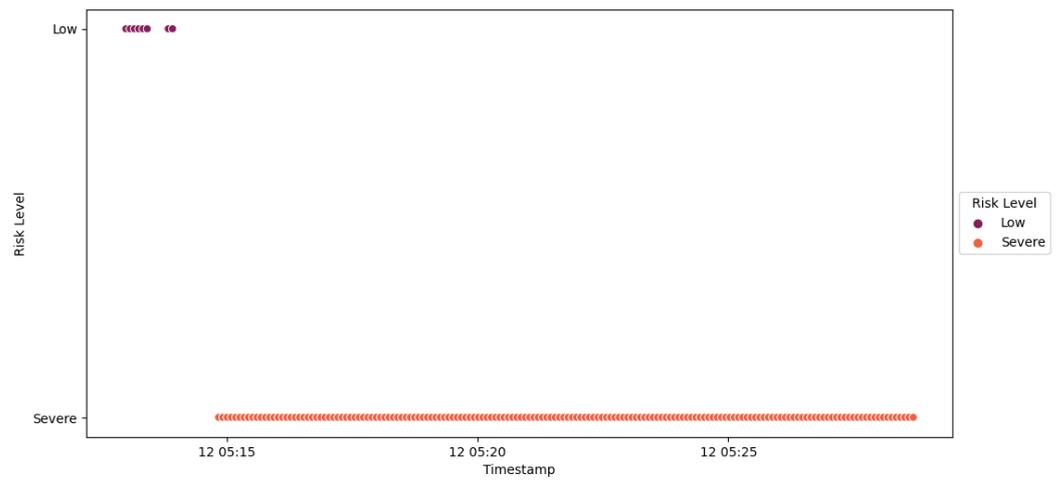


Figure 55. IED1A Scenario 2—trust SCADA HMI scores.

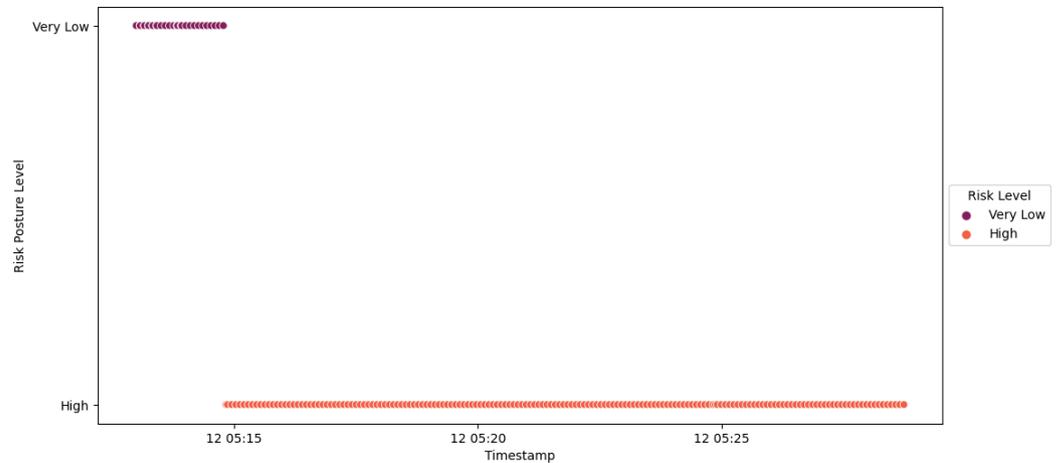


Figure 56. IED1A Scenario 2—substation’s risk posture.

10.3.3. Scenario 3—Misbehavior after Consideration Period

As illustrated in Figure 57, the sequence of events from *Start Disconnect* to *Probation Started* in Scenario 3 closely mirrors that of Scenario 1. A communication gap with the trust SCADA HMI is also evident during the disconnection process, depicted in Figure 58. The newly introduced trust IED demonstrated proper behavior during the consideration period, with *Probation Period Decreased* events being documented. However, following this period, the new trust IED engaged in malicious behavior by executing a Modbus frame stacking attack in response to the trust SCADA HMI.

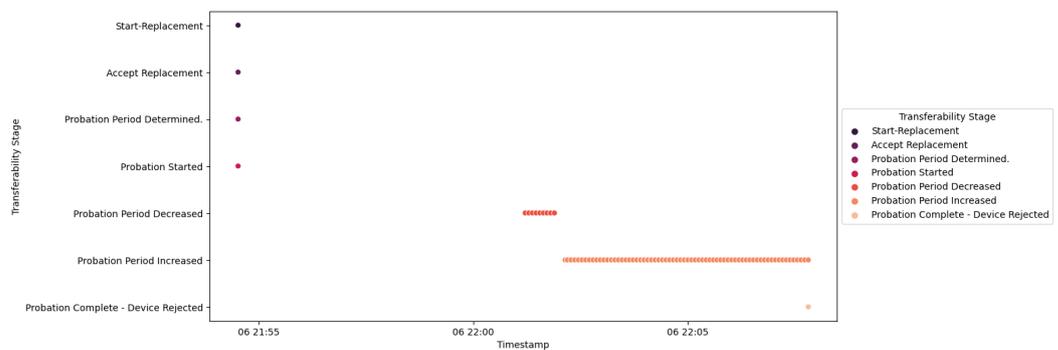


Figure 57. IED4C Scenario 3—transferability process.

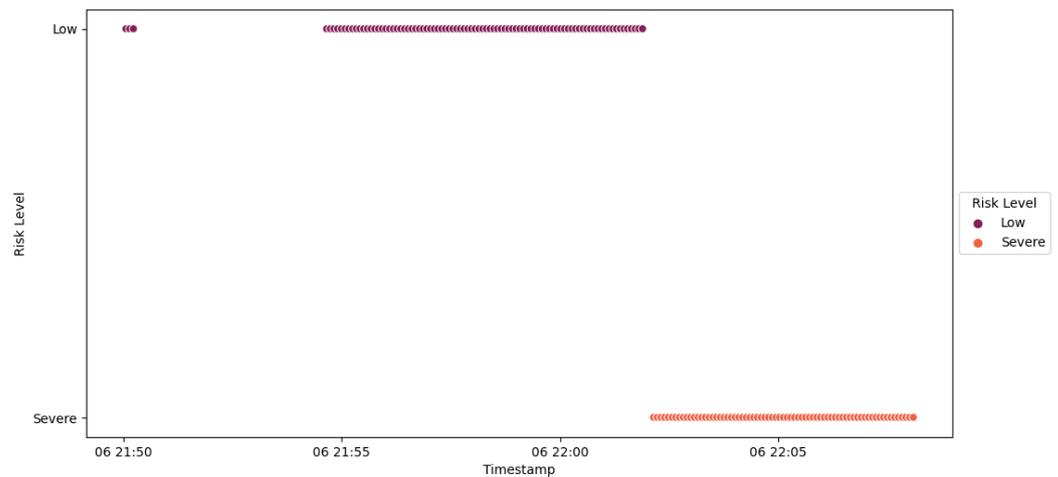


Figure 58. IED4C Scenario 3—trust SCADA HMI scores.

The trust SCADA HMI promptly detected this malicious behavior, resulting in the blacklisting of the IED, as illustrated in Figure 59. Consequently, the HMI escalated the risk level of IED4C from *Low* to *Severe*. It is worth noting that the distinction between Scenario 2 and Scenario 3 lies in the duration for which the risk level scores of the new trust IEDs were tracked. In Scenario 3, these scores were monitored for an extended period.

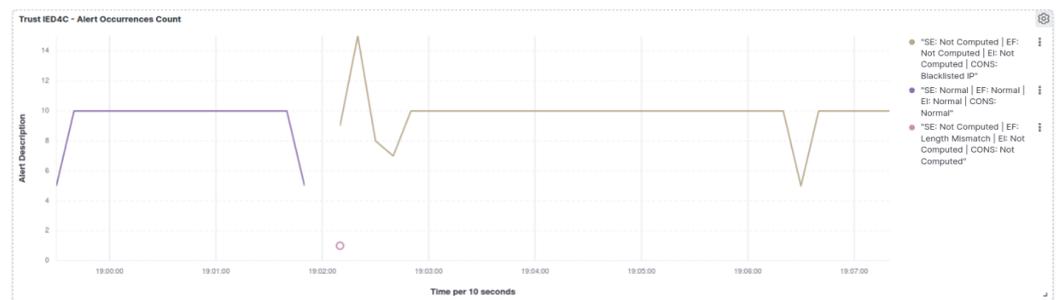


Figure 59. IED4C Scenario 3—SCADA dashboard

The central agent responded by incrementing probation points to the probation period, as indicated in Figure 57. This action aimed to extend the probation period due to the observed malicious behavior. Consequently, the substation’s risk posture was adjusted by the central agent, transitioning from *Very Low* to *Low*, as depicted in Figure 60. Notably, the central agent’s records show a prolonged period of *Low* risk posture events in Scenario 3, compared to those in Scenario 2.

In line with this trend, the probation period was consistently extended in response to the absence of non-malicious communication from the new trust IED, as highlighted in Figure 57. Ultimately, this led to the triggering of the *Probation Complete—Device Rejected* event, culminating in the non-acceptance of the new trust IED. A similar outcome was observed for IED1A, with the distinction that the substation’s risk posture transitioned from *Very Low* to *High*, as displayed in Figure 61.

10.3.4. Scenario 4—Unsatisfactory Trust Scores

The sequence of events from *Start Disconnect* to *Disconnection* in Scenario 4, as depicted in Figure 62, aligns with the preceding scenarios. Notably, during the *Start Replacement* event, the new trust IED transmits its subpar trust scores to the central agent. The central agent employs these received scores, in conjunction with the trust scores of IED4C, to make a determination regarding the connection of the new trust IED to the network. Upon performing the computation, it becomes evident that the trust scores of the new trust IED fall below the satisfactory threshold. Consequently, this evaluation triggers a *Device Rejected* event.

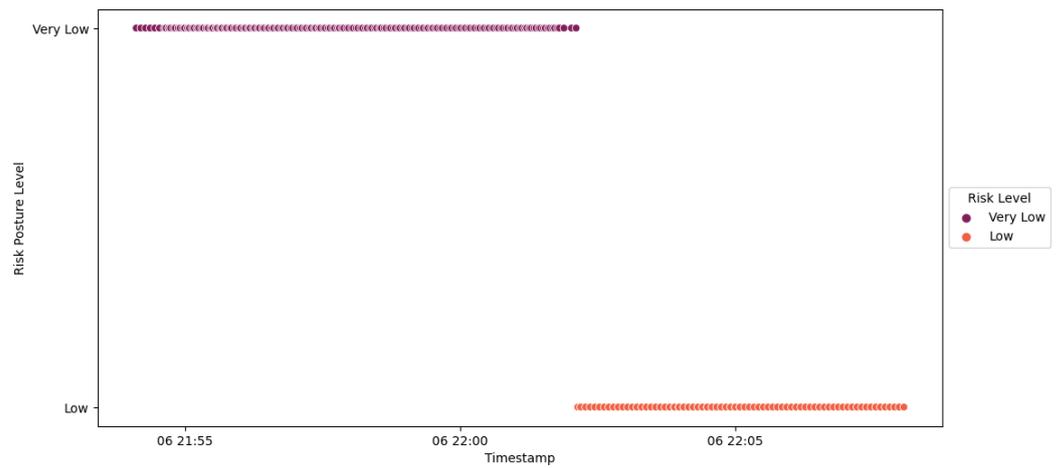


Figure 60. IED4C Scenario 3—substation's risk posture.

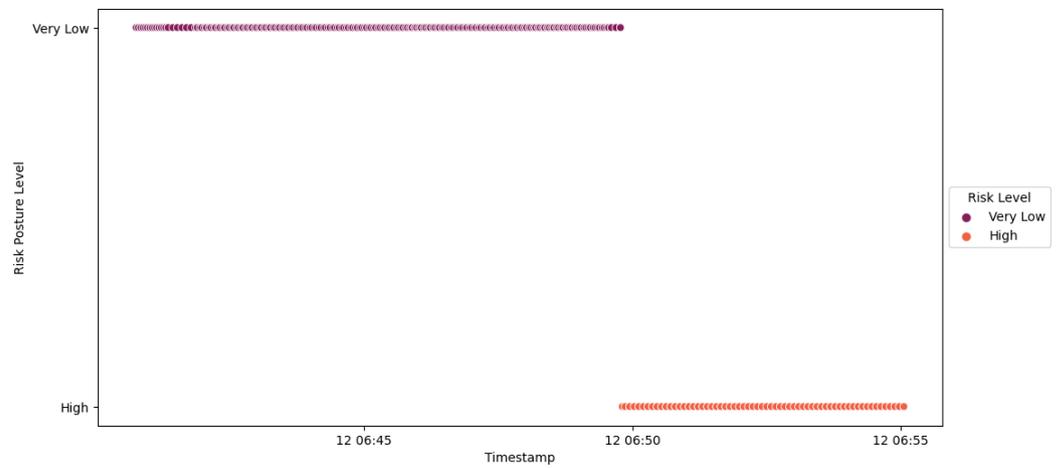


Figure 61. IED1A Scenario 3—substation's risk posture.

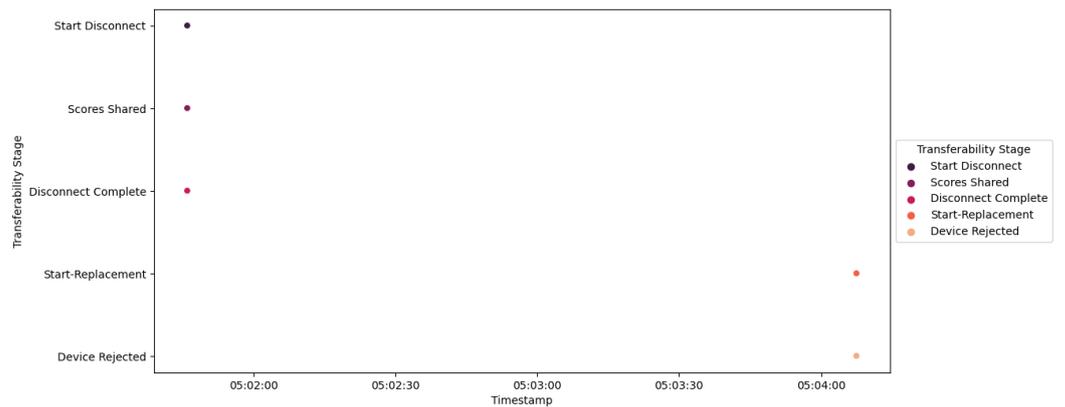


Figure 62. IED4C Scenario 4—transferability process.

10.4. Challenges

Even though Table 8 shows our work has considerable advantages of other models, there are some challenges. Indeed, the model has identified two critical types of attacks that pose challenges to its effectiveness: the delayed response attack and the baseline attack. The delayed response attack, if successfully executed, has the potential to significantly disrupt the substation's operations due to the stringent time constraints within which these

systems operate. Even though this attack vector has not been widely observed in public instances, its potential impact necessitates consideration.

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

Addressing the delayed response attack requires the incorporation of a strict time constraint parameter into the model. However, this implementation is complex due to the inherent limitations of the current infrastructure, such as virtual machine (VM) constraints, leading to potential false alarms stemming from processing limitations and jitters. Finding a solution to this issue remains a challenge and is identified as a future avenue of research and development.

The baseline attack, which allows an attacker to replay requests without altering their content, represents another potential threat. This attack method enables manipulation of specific situations, such as preventing a circuit breaker from receiving a command to close. Countering such an attack requires enhancing the trust model's capabilities to monitor the IED's environment more comprehensively. This would involve the integration of additional logic to detect anomalous behavior and mitigate potential malicious activities.

During the probation period, the SCADA system enforces blacklisting in response to the detection of a malicious query for over half of the designated period. Following the expiration of the blacklist duration, normal operations are resumed. However, certain aspects have not been fully addressed in the current model. For instance, scenarios where the trust scores of the replaced device are subpar or when a trust device provides accurate responses while spreading false information about other devices have not been explicitly considered.

Furthermore, the uniqueness of the substation environment introduces challenges. Traditional testing approaches are not always viable due to potential network flooding issues. Moreover, the limitations of the existing testbed hinder the ability to thoroughly assess certain complex scenarios. For instance, detecting baseline replay attacks remains a challenge within the current setup.

It is essential to acknowledge that achieving a comprehensive and accurate representation of all potential scenarios in a real-world substation environment can be intricate. As such, further research and refinement are required to address these limitations and enhance the model's robustness and applicability.

11. Conclusions and Future Work

We introduced a comprehensive trust framework for substations, featuring three components: a trust model detecting protocol-based attacks, a risk posture model assessing the substation's response to attacks, and a trust transferability model monitoring device integration. Testing the framework in a Docker-based environment with multi-agent architecture demonstrated its resilience against various attacks, though vulnerabilities to baseline replay and delayed response attacks were identified.

We also explored trust transferability scenarios, including normal and compromised replacements, and observed successful detection of malicious behavior from trust devices. However, certain aspects, such as addressing subpar trust scores or trust devices spreading false information, require further consideration in future research.

Author Contributions: Conceptualization and methodology, K.B.-B., A.A.G. and A.H.L.; software, K.B.-B.; validation, formal analysis and investigation, K.B.-B., A.A.G. and A.H.L.; writing—original draft preparation, K.B.-B.; writing—review, editing and visualization K.B.-B., A.A.G. and A.H.L.; supervision, A.A.G. and A.H.L.; funding acquisition, A.A.G. All authors have read and agreed to the published version of the manuscript.

Funding: 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.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in a publicly accessible repository that does not issue DOIs Publicly available datasets were analyzed in this study. This data can be found here: <https://www.unb.ca/cic/datasets/modbus-2023.html> (accessed on 31 August 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Xu, Z.; Salehi Shahraki, A.; Rudolph, C. Blockchain-Based Malicious Behaviour Management Scheme for Smart Grids. *Smart Cities* **2023**, *6*, 3005–3031. [CrossRef]
- The Real Story of Stuxnet. Available online: <https://spectrum.ieee.org/the-real-story-of-stuxnet> (accessed on 11 December 2022).
- Pipedream: Chernovite’s Emerging Malware Targeting Industrial Control Systems. Available online: https://hub.dragos.com/hubfs/116-Whitepapers/Dragos_ChernoviteWP_v2b.pdf?hsLang=en (accessed on 15 February 2023).
- COSMICENERGY: New OT Malware Possibly Related to Russian Emergency Response Exercises. Available online: <https://www.mandiant.com/resources/blog/cosmicenergy-ot-malware-russian-response> (accessed on 11 December 2022).
- Recommendations Following the Colonial Pipeline Cyber Attack. Available online: <https://www.dragos.com/blog/industry-news/recommendations-following-the-colonial-pipeline-cyber-attack/> (accessed on 13 February 2023).
- Lei, H.; Singh, C.; Sprintson, A. Reliability modeling and analysis of IEC 61850 based substation protection systems. *IEEE Trans. Smart Grid* **2014**, *5*, 2194–2202. [CrossRef]
- Requirements for IDS in Substations. Available online: <https://electrical-engineering-portal.com/improving-cybersecurity-substations-intrusion-detection#requirements-ids-substations> (accessed on 22 December 2022).
- A Researcher’s Perspective on the North American Protective Relay Marketplace. Available online: <https://www.tdworld.com/test-and-measurement/article/20972654/a-researchers-perspective-on-the-north-american-protective-relay-marketplace/> (accessed on 1 February 2023).
- The Worldwide Study of the Protective Relay Marketplace in Electric Utilities: 2019–2022. Available online: <https://www.newton-evans.com/product/the-worldwide-study-of-the-protective-relay-marketplace-in-electric-utilities-2019-2022/> (accessed on 13 February 2023).
- Cook, K. *Trust in Society*; Russell Sage Foundation Series on Trust; Russell Sage Foundation: New York, NY, USA, 2003; Volume 2, p. 432.
- Gambetta, D. Can we trust trust. In *Trust: Making and Breaking Cooperative Relations*; Blackwell: Oxford, UK, 2000; Volume 13, pp. 213–237.
- Rousseau, D.M.; Sitkin, S.B.; Burt, R.S.; Camerer, C. Not so different after all: A cross-discipline view of trust. *Acad. Manag. Rev.* **1998**, *23*, 393–404. [CrossRef]
- Boakye-Boateng, K.; Ghorbani, A.A.; Lashkari, A.H. A novel trust model in detecting final-phase attacks in substations. In Proceedings of the 2021 18th International Conference on Privacy, Security and Trust (PST), Auckland, New Zealand, 13–15 December 2021; pp. 1–11.
- Boakye-Boateng, K.; Ghorbani, A.A.; Lashkari, A.H. A Trust-Influenced Smart Grid: A Survey and a Proposal. *J. Sens. Actuator Netw.* **2022**, *11*, 34. [CrossRef]
- Borowski, J.F.; Hopkinson, K.M.; Humphries, J.W.; Borghetti, B.J. Reputation-based trust for a cooperative agent-based backup protection scheme. *IEEE Trans. Smart Grid* **2011**, *2*, 287–301. [CrossRef]
- Fadul, J.E.; Hopkinson, K.M.; Andel, T.R.; Sheffield, C.A. A trust-management toolkit for smart-grid protection systems. *IEEE Trans. Power Deliv.* **2013**, *29*, 1768–1779. [CrossRef]
- Qureshi, K.N.; ul Islam, M.N.; Jeon, G. A trust evaluation model for secure data aggregation in smart grids infrastructures for smart cities. *J. Ambient Intell. Smart Environ.* **2021**, *13*, 235–252. [CrossRef]
- 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*, 2549–2572. [CrossRef]

19. Boakye-Boateng, K.; Ghorbani, A.A.; Lashkari, A. Securing Substations with Trust, Risk Posture, and Multi-Agent Systems: A Comprehensive Approach. In Proceedings of the 2023 20th Annual International Conference on Privacy, Security and Trust (PST), Copenhagen, Denmark, 21–23 August 2023; pp. 1–12. [CrossRef]
20. Bellifemine, F.L.; Caire, G.; Greenwood, D. *Developing Multi-Agent Systems with JADE*; John Wiley & Sons: Hoboken, NJ, USA, 2007; Volume 7
21. Wang, P.; Govindarasu, M. Multi-Agent Based Attack-Resilient System Integrity Protection for Smart Grid. *IEEE Trans. Smart Grid* **2020**, *11*, 3447–3456. [CrossRef]
22. Mohamed, A.A.R.; Omran, W.A.; Sharkawy, R. Centralized/Decentralized Power Management Strategy for the Distribution Networks based on OPF and Multi-Agent Systems. In Proceedings of the 2021 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe), Espoo, Finland, 18–21 October 2021; pp. 1–5.
23. Elena, D.O.; Florin, D.; Valentin, G.; Marius, P.; Octavian, D.; Catalin, D. Multi-agent System for Smart Grids with Produced Energy from Photovoltaic Energy Sources. In Proceedings of the 2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Ploiesti, Romania, 30 June–1 July 2022; pp. 1–6.
24. Priyadarshana, H.; Hemapala, K.U.; Wijayapala, W.S.; Saravanan, V.; Boralessa, M.K.S. Developing multi-agent based micro-grid management system in jade. In Proceedings of the 2019 2nd International Conference on Power and Embedded Drive Control (ICPEDC), Chennai, India, 21–23 August 2019; pp. 552–556.
25. Modbus Organization. *Modbus Application Protocol Specification V1.1b*; Modbus Organization: Andover, MA, USA, 2006.
26. Modbus Organization. *MODBUS Messaging on TCP/IP Implementation Guide: V1.0b*; Modbus Organization: Andover, MA, USA, 2006.
27. Techniques—ICS | MITRE ATT&CK[®]. Available online: <https://attack.mitre.org/techniques/ics> (accessed on 27 December 2022).
28. Boakye-Boateng, K.; Ghorbani, A.A.; Lashkari, A.H. RiskISM: A Risk Assessment Tool for Substations. In Proceedings of the 2021 IEEE 9th International Conference on Smart City and Informatization (iSCI), Shenyang, China, 20–22 October 2021; pp. 23–30.
29. Papadimitriou, C.; Sideri, M. On the Floyd–Warshall algorithm for logic programs. *J. Log. Program.* **1999**, *41*, 129–137. [CrossRef]
30. The ELK Stack: From the Creators of Elasticsearch | Elastic. Available online: <https://www.elastic.co/what-is/elk-stack> (accessed on 13 May 2023).
31. Cho, J.H.; Chan, K.; Adali, S. A survey on trust modeling. *ACM Comput. Surv. CSUR* **2015**, *48*, 1–40. [CrossRef]
32. Critical Infrastructure Threat Information Sharing Framework. *A Reference Guide for the Critical Infrastructure Community*; USA Homeland Security: Washington, DC, USA, 2016; p. 5.
33. Greer, C.; Wollman, D.A.; Prochaska, D.E.; Boynton, P.A.; Mazer, J.A.; Nguyen, C.T.; FitzPatrick, G.J.; Nelson, T.L.; Koepke, G.H.; Hefner, A.R., Jr.; et al. *Nist Framework and Roadmap for Smart Grid Interoperability Standards, Release 3.0*; Technical Report; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2014.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.