2024 IEEE 24th International Conference on Software Quality, Reliability and Security (QRS) | 979-8-3503-6563-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/QRS62785.2024.00042

Assets Criticality Assessment of Industrial Control Systems: A Wind Farm Case Study

Shwetha Gowdanakatte[1], Mahmoud Abdelgawad[2], and Indrakshi Ray[2]
[1]Department of Systems Engineering, Colorado State University, Fort Collins, Colorado, USA
[2]Department of Computer Science, Colorado State University, Fort Collins, Colorado, USA
{shwetha.gowdanakatte, m.abdelgawad, indrakshi.ray}@colostate.edu

Abstract—The increasing growth of threats to Industrial Control Systems (ICS) in the energy sector puts this critical infrastructure at high risk. Consequently, holistic approaches are needed to assess the criticality of ICS assets, identify relevant security threats, and develop mitigation techniques to make the energy critical infrastructure cyber-resilient. This paper presents a methodology for analyzing the criticality and resiliency of ICS assets by assessing the impact caused by attacks on such assets. Our approach consists of modeling the ICS architecture in a form that is suitable for analysis. We use Coloured Petri Nets (CPN) for formal representation and analysis – CPN is supported by automated tools for analysis and it has been used for verification of real-world systems. We use Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege (STRIDE) for evaluating the threats in the ICS architecture. We use Microsoft Threat Modeling Tool (MTMT) for ICS threat modeling that classifies the threats into the categories defined in STRIDE. Based on the type of threat on each asset and its impact on the entire ICS, we rank the asset's criticality. The threat modeling framework assesses the criticality and resiliency of tangible and intangible assets, thus addressing the gap in the current research. Threat models are also converted into CPN models. The CPN models of the ICS architecture and the threat model are then composed. The methodology then verifies the resulting CPN. This verification explores the system states where the ICS cannot resist the attacks and identifies the ICS's critical assets that have been compromised. The methodology is applied to a wind farm system comprising many distributed subsystems connected via various networks. The result shows that the methodology is practical for the ICS verification and assets criticality assessment, providing recommended mitigations to construct a robust ICS.

Keywords–Industrial Control Systems (ICS); assets criticality assessment, wind farm system, threat models; formal methods; Coloured Petri Nets (CPN)

1. Introduction

Industrial Control Systems (ICS) maintain critical infrastructure such as energy plants, chemical plants, water treatment plants, railways, and transportation systems. Due to rapid advancements in communication and technology, the operational technology (OT) components are connected to the information technology (IT) components. This integration of OT and IT has increased the attack surface, as evidenced by increasing cyber attacks in recent years, particularly on renewable energy sector such as wind energy [1].

Numerous cyberattack incidents targeting wind energy (i.e., wind farms) have recently been reported. In November 2021, Vestas, a Danish turbine manufacturer, encountered a ransomware attack that allowed unauthorized access to its internal system. The attack shut down IT infrastructure across multiple business units and locations [2]. In February 2022, a cyberattack on a satellite communication caused Enercon, a German wind turbine maker, a Denial of Service (DoS), leading to the loss of the remote monitoring and control systems for 5,800 wind turbines. It took Enercon about two months to recover [1]. In March 2022, Nordex, a German wind turbine manufacturer, disclosed a ransomware attack. The incident forced Nordex to shut down IT infrastructure to prevent the attack's spread. In April 2022, a successful cyberattack drove Deutsche Windtechnik, a German wind turbine maintenance company, to lose control of about 2,000 turbines [3].

ICS devices typically have a long life-span. Many of the ICS equipment are older legacy systems that operated in silos and were designed without security considerations. Moreover, connecting the ICS with the Internet has increased the attack surface: attacks can be launched digitally or physically, and may have more far-reaching consequences. Since it is impossible to make a system completely secure, threat modeling and asset criticality assessment is important to help determine what to secure in ICS. Due to the heterogeneous nature of ICS, assessing the assets' criticality is challenging. ICS incorporates physical (servers, control devices, network components, and electro-mechanical devices) and cyber (software, firmware, and communication protocols) assets.

We use an example wind farm ICS to illustrate our work. Threat identification and analysis of wind farm ICS shows that the connectivity between wind turbines and control center, the wind farm network, and the turbine are the vulnerable points to launch cyberattacks [4]. This interconnection also makes the Supervisory Control and Data Acquisition (SCADA) system, which configures and controls wind turbine parameter values, vulnerable to communication attacks such as Denial of Service (DoS), Man-In-The-Middle (MITM), and replay attacks. Researchers [5] design attack scenarios to demonstrate operations disruption on wind farm systems. Such scenarios help develop mitigation techniques to reduce the cyberattack impacts. Others propose algorithms for assets criticality assessment. Akbarzadeh et al. [6] determine the asset's criticality based on their centrality. Liu et al. [7] use a network topology graph, where nodes represent assets and edges denote logical or physical connections between them,

2693-9177/24/$31.00 ©2024 IEEE
DOI 10.1109/QRS62785.2024.00042

to assess the criticality of the assets. Researchers also use threat modeling frameworks such as Microsoft Corporation's STRIDE that identifies and analyzes threats [8][9].

Although research has been done on various aspects addressing security issues, a design framework that allows one to analyze ICS systems and in particular identify its critical assets is missing. Attack scenarios are often designed in an ad-hoc manner and the impact of an attack on other components is also not well-understood. Towards this end, there is a need for automated analysis of an ICS and evaluation of all potential attacks and their impacts to assess assets criticality. Our research fills this gap.

We design a methodology for assessing asset criticality. We use Unified Modeling Language [10] (UML) sequence diagram to model the ICS system. UML is the de-facto modeling language used in the software industry. We also use the STRIDE threat modeling framework [11] to generate an ICS threat model. Manual analysis of the attacks on an ICS system is tedious and error-prone. Towards this end, we focus on using Coloured Petri Nets (CPN) [12]. CPN is associated with a high-level programming language (CPN-ML [13]) and Integrated Development Environment (CPN Tools [14]) that are practical to simulate processes interactions and do automated analysis. UML sequence diagram describing the process interactions in ICS is converted into CPN using transformation rules [15]. Each threat reported in the ICS threat model is converted to a CPN attack. These CPN attacks are integrated with the CPN of ICS to demonstrate various attack scenarios. We check the state space of the composed CPN for attack impact analysis. Specifically, the state space of CPN model investigation reveals when the ICS fails to function, and determines which asset compromises lead to such failures. We demonstrate that the methodology is practical for ICS verification and asset criticality assessment. It provides recommended mitigations to make the ICS cyber-resilient.

The rest of the paper is organized as follows. Section 2 describes the architecture of a wind farm. Section 3 introduces a methodology for assessing the asset criticality. Section 4 provides wind farm specifications for designing CPN. Section 5 performs STRIDE threat modeling and provides the details of the threat analysis. Section 6 employs formal verification of the threat analysis using the wind farm CPN and provides state space and reachability analysis. Section 7 uses the results from state space and reachability analysis to assess the assets criticality. Section 8 recommends mitigating the threats reported by the threat modeling. Section 9 delivers the findings of our studies on current research. Finally, Section 10 summarizes our work and future implementation plan.

2. Wind Farm Architecture

Figure 1 shows a typical wind farm network topology [16]. It comprises different subsystems denoted by the rectangular blocks outlined in red color:

- **Wind Turbine** consists of electro-mechanical and control devices to generate and regulate the power. The control devices include a Programmable Logic Controller (PLC), a slave Human Machine Interface (HMI), and a master HMI. The PLC communicates with HMIs and distributes inputs/outputs through an Ethernet switch. Distributed inputs/outputs communicate with physical sensors to send or receive operation-specific information through a physical connection.
- **Wind Turbine Control Center** consists of a Supervisory Control and Data Acquisition (SCADA) from the Original Equipment Manufacturer (OEM). It communicates with the wind turbine through a Process Control Network (PCN) using industrial protocols such as MODBUS over TCP/IP, PROFINET, and DNP3.
- **Wind Farm Control Room** has servers communicating with wind turbines using SCADA protocols on the Demilitarized Zone (DMZ).
- **Customer Control Room** allows the remote customer access to the wind turbine through a local control center via a secured Virtual Private Network (VPN) connection.
- **Vendor Control Room** allows the remote vendor access to the wind turbines for remote troubleshooting and product updates through a secure VPN. The reference architecture uses 1) Siemens S71500 PLC and Siemens HMI for control devices. 2) Windows Server 2012 for the OEM SCADA Server, the wind farm control room server, and the customer grid control server. 3) PROFINET and DNP3 as industrial communication protocols.

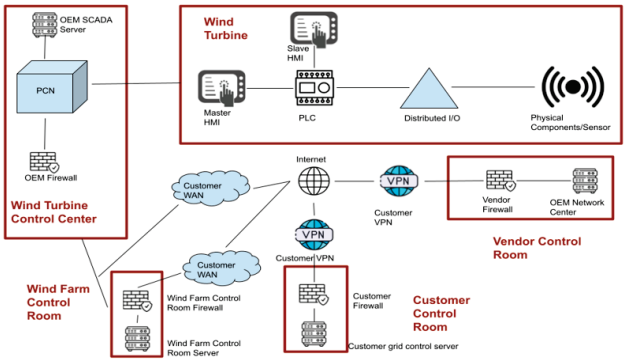

Figure 1: Wind Farm Architecture [16]

The wind farm control room, customer control room, and vendor control room independently communicate securely with the wind turbine control center. The wind turbine control center communicates securely with the wind turbine at the field site to regulate the turbine's stability. The wind farm, customer, and vendor control rooms request access to the OEM SCADA, and the OEM firewall verifies the access request. If the access request is granted, the subsystem's operation request will be communicated to the HMI. The HMI communicates with the PLC to perform the requested operation, and the results are sent to the corresponding subsystem. For instance, when the wind farm control requests access to the OEM SCADA to perform reconfiguration operations on the PLC, and the request is granted, the result of the PLC reconfiguration is sent back to

the wind farm control room. Similarly, if the request is denied, the denial response is sent to the corresponding subsystem. This wind farm architecture needs assets criticality assessment to pinpoint the highly impactful security threats that require proper mitigations in order to construct robust wind farm ICS.

3. Assets Criticality Assessment Methodology

The current research effort by [6], [7], and [16] focuses on assessing the criticality of tangible assets based on their interconnections. However, assessing the criticality based on the potential threats on both tangible and non-tangible assets is imperative to achieving security by design. Our method demonstrates a methodology of assessing the criticality of both physical and cyber assets.

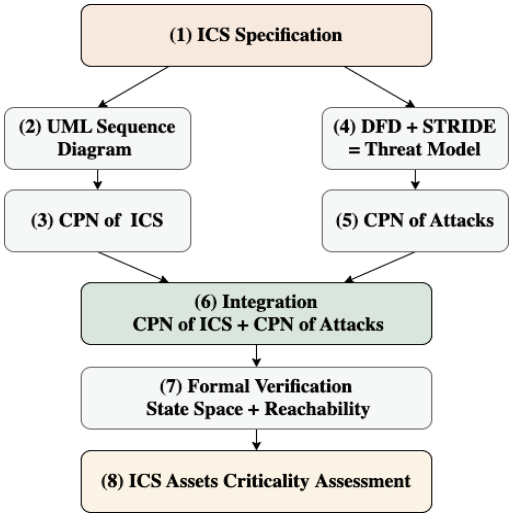

Figure 2: Assets Criticality Assessment Methodology

As illustrated in Figure 2, the methodology consists of 8 steps, starting from ICS specification as input and ending with ICS assets criticality assessment. Step 1, ICS specification, is essential because the next steps rely on its outcome; if a portion of ICS specification is overlooked, it will not be covered in the following steps. Steps 2 and 3 (UML Sequence Diagram and CPN of ICS, respectively) are processed sequentially; the output of one step is the input for the next step. Similarly, Step 4 (Data Flow Diagram processed by STRIDE threat model framework) gets the output from Step 1, processes it, and passes the outcome to Step 5 (CPN attacks). Notice that the outcomes from Steps 3 and 5 are required in order to perform Step 6 (Integrating the CPN of ICS with the CPN of attacks). In Step 7 (formal verification using state space and reachability), the integration of the CPN of ICS with the CPN of attacks is formally verified. Based on this formal verification, Step 8 analyzes the ICS security impact of attacks and assesses the ICS assets' criticality. The 8 steps are described as follows.

Step 1: Define the ICS Specification It specifies the ICS entities that provide executable processes and services. The ICS entities are defined as $ENTITIES = \{entity_1, entity_2, \ldots, entity_n\}$. Each entity comprises a set of variables that describe a portion of the ICS status. Step 1 also specifies use cases that show the control flow and data flow between these entities. These use cases are expressed using text, diagrams, or both.

Step 2: Derive UML Sequence Diagram UML sequence diagram is derived directly from the ICS specification. It describes interactions between the system's entities through exchanging messages [17]. Step 2 defines a set of messages as $MESSAGES = \{message_1, message_2, \ldots, message_n\}$. These messages are sent and received from one entity to another sequentially. The messages include *request*, *response*, and *command* messages. A message comprises a set of parameter-value pairs as $message_i(parameter_1 = value_1, parameter_2 = value_2, \ldots, parameter_i = value_i)$. When an entity sends or receives a message from another entity, it defines an interaction. The interaction is defined as $interaction_k = (entity_i, message_n, entity_j)$. The entities, messages, and interactions help construct the UML sequence diagram.

Step 3: Convert UML Sequence Diagram into CPN-ICS The literature presents research efforts to transform the UML sequence diagram to CPN [18][19][15]. We use the transformation rules introduced in [15] because they are adoptable in our context.

Step 4: Derive DFD and Apply STRIDE The data flow diagram (DFD) is also derived directly from the ICS specification. DFD is a diagramming technique to describe activities and processes of a system [20]. It includes four elements: process, data flow, data store, and external entity. The entities defined in the ICS specification are mapped to DFD as processes. The use cases that express the control flow of data transferring between entities are used to draw the DFD data flows. The communication between entities also represents DFD data flow. The data saver described in the ICS specification is a DFD data store. An entity interacting with the ICS, but not part of the ICS, is referred to as an DFD external entity. We use an ICS template [21], which has been developed to construct ICS DFD diagrams using Microsoft Threat Modeling Tool (MTMT). The MTMT automatically generates a threat model representing six threat types (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege) corresponding to the DFD elements. It also assigns priorities (High, Medium, Low) to the generated threats based on the threat consequences.

Step 5: Convert Threats into CPN-Attacks An attack is an action taken by an adversary that changes the state of a entity in a way that renders it unable to execute a process or provide a service. Each attack described in the threat model is modeled as an CPN attack. An attack is specified using preconditions and postconditions; preconditions are conditions that must be true for the attack to take place and postconditions describe the effects of the attack. Step 5 defines the preconditions and postconditions of the attacks, where

the postconditions include the changes on the ICS entity caused by the attack.

Step 6: Integrate CPN-ICS with CPN-Attacks This step integrates a set of CPN attacks with the CPN of ICS to present attack scenarios. We consider various types of attacks in the context of our ICS: Spoofing attack, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. We represent them in CPN and check whether these attacks can be integrated with the CPN model of ICS to form attack scenarios. We check the CPN nodes of ICS to identify where these attack preconditions are satisfied – these are the exact points where the attack can be integrated with the ICS. The attack modifies the postcondition of the corresponding CPN node, demonstrating an alteration. This is how the various attack scenarios are constructed.

Step 7: Verify CPN-ICS Attached with CPN-Attacks For every attack scenario, the state space of the CPN-ICS is generated. Step 7 verifies the state transitions of the ICS with the attack attached, where the state ICS terminates the dispute of attached attacks, and why it terminates. It also verifies the paths (sequence of states) from the ICS initial state to the states where the attacks occur and checks how these attacks affect the ICS entities.

Step 8: Analyze and Assess the ICS Assets Criticality It uses the information provided in Step 7 to analyze and assess the ICS assets criticality. The ICS assets criticality assessment is determined based on the impact size (how many and what ICS entities are affected when an attack disrupts one of them). Step 8 also highlights recommended mitigations that reduce the damage when attacks occur.

We apply these 8 steps to the wind farm architecture presented in Section 2. Section 4 covers Steps 2 and 3, Section 5 applies Steps 4 and 5, and Section 6 employs Steps 6 and 7. Section 6 utilizes Step 8, showing the analysis and criticality assessment of the wind farm assets.

4. Wind Farm Specification to CPN

The wind farm architecture presented in Section 2 illustrates 7 entities: vendor control room, customer control room, wind farm control room, wind turbine control center, and wind turbine, which composes three entities, HMIs, PLC, and physical components and sensors. VPNs, firewalls, PCN, Wide Area Networks (WAN), and the Internet are not considered entities. Instead, they provide secure communication between wind farm entities. Each control room and control center employs a specific server system; for instance, the wind turbine control center employs the OEM SCADA server, and the customer control room utilizes the customer grid server. These servers are not entities; instead, they are entity attributes.

The set of wind farm entities is defined as $ENTITIES = \{VendorControlRoom,\ CustomerControlRoom,\ WindFarm\ ControlRoom,\ WindTurbineControlCenter,\ MasterHMI,\ PLC,\ PhysicalComponents/Sernsors\}$. It also describes three interaction scenarios: (i) vendor control room requests access to SCADA, (ii) customer control room requests access to SCADA, and (iii) wind farm control room requests access to SCADA. These access requests are sent to the wind turbine control center, which employs the SCADA, to allow performing operations (e.g., reconfiguration) on the wind turbine entities (HMI, PLC, physical components, and sensors).

The UML sequence diagram in Figure 3 represents the interaction between the wind farm entities. The vendor control room, customer control room, and wind farm control room independently communicate securely with the wind turbine control center and request access to the OEM SCADA. The OEM firewall verifies the access request. The operation request is sent to the HMI if the access request is granted. The HMI communicates with the PLC to perform the requested operation, and the result is returned to the corresponding entity. If the request is denied, a request denied response is sent to the corresponding entity. The set of messages are $MESSAGES = \{AccessRequest,\ Request,\ ProcessRequest,\ DenialResponse,\ SucceesfulResponse\}$. The access request message parameters defined as $AccessRequest(UserID : String,\ Password : String,\ SourceIPAddress : String,\ DestinationIPAddress : String,\ SourcePortNo : Integer,\ DestinationPortNo : Integer,\ Operation : List)$. The operation list includes $[Start,\ Stop,\ ReadStatus,\ ReadDatalog,\ ReadPLCTag,\ WritePLCTag]$. The process request parameters include only the required operation since it passes between the wind turbine subsystems (HMI, PLC, and physical components) through hardwire communication. The parameters of denial response is defined as $DenialResponse(SourceIPAddress : String,\ DestinationIPAddress : String,\ SourcePortNo : Integer,\ DestinationPortNo : Integer,\ Message : String)$. The $SuccessfulResponse$ message parameters are the same as the denial response. They differ in the message content. There are 9 interactions that might happen between the vendor control room, wind turbine center, and wind turbine, defined as $interaction_1 = (VendorControlRoom,\ AccessRequest,\ WindTurbineCenter)$, $interaction_2 = (WindTurbineCenter,\ DenialResponse,\ VendorControlRoom)$, $interaction_3 = (WindTurbineCenter,\ Request,\ HMI)$, $interaction_4 = (HMI,\ ProcessRequest,\ PLC)$, $interaction_5 = (PLC,\ ProcessRequest,\ PhysicalComponents)$, $interaction_6 = (PhysicalComponents,\ Response,\ PLC)$, $interaction_7 = (PLC,\ Response,\ HMI)$, $interaction_8 = (HMI,\ Response,\ WindTurbineCenter)$, $interaction_9 = (WindTurbineCenter,\ SuccessfulResponse,\ VendorControlRoom)$, Similarly, the interchanges between the customer control room, wind turbine center, and wind turbine entities have 9 interactions, and another 9 interactions between the wind farm room, wind turbine center, and wind turbine entities. The total of interchanges that might occur in the wind farm system is 27 interactions.

The messages and interactions of wind farm subsystems are converted into CPN. We applied the transformation rules mentioned in Step 3 of the methodology to convert the UML

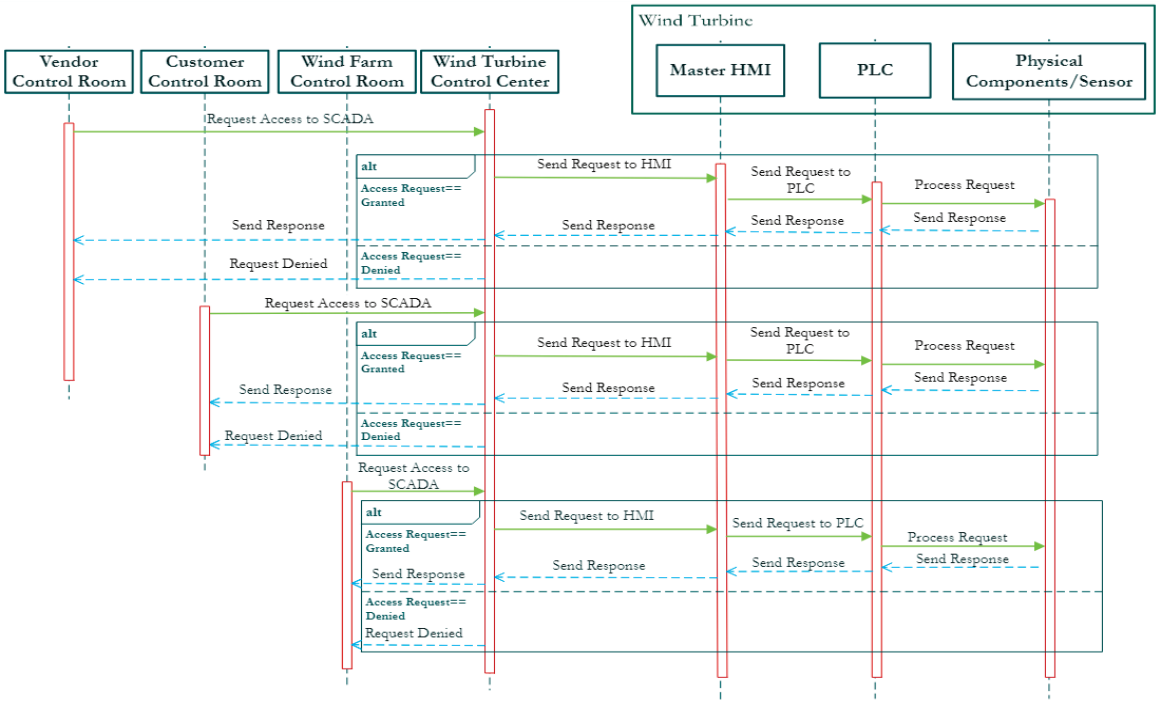

Figure 3: UML Sequence Diagram for Wind Farm System

sequence diagram into CPN. Figure 4 illustrates the CPN of the wind farm system. The wind farm CPN consists of 7 nets, three representing the vendor, customer, and wind farm control rooms. One net represents the wind turbine center, and three nets represent the wind turbine subsystems (HMI, PLC, and physical components). The three CPN nets of the control rooms (vendor, customer, and wind farm) communicate with the CPN of the wind turbine center simultaneously. The CPN of the wind turbine center communicates sequentially with the CPN of the HMI, CPN of the PLC, and CPN of the physical components, respectively. As shown in Figure 4, we define tokens that represent access requests from vendor, customer, and wind farm control rooms. We distinguish between these tokens by the User ID, password, IP Address, and Post number from which the access request is sent. For instance, the vendor control room has a token that includes UserID="VCR123", Password="PW123", scrIPAddr="10.10.10.35", dstIPAddr="14.30.2.11", srcPort="1414", dstPort="5353", operation="ReadStatus". This token shows that the access request is sent from a user "VRC123", who uses a machine with IP address "10.10.10.35" through post number 1414, to the wind turbine center machine (running SCADA) that has IP address "14.30.2.11", and the request is received through port 5353. We added a place (i.e., "Submit") that holds a "YES" or "NO" token to demonstrate which control room submits an access request. For instance, three access requests are sent simultaneously if all control rooms have "YES" tokens assigned in the "Submit" places. The wind turbine center verifies the request's source IP address and port number and checks for user authentication. If the request is granted, the CPN of the wind turbine center passes this token to the CPN of HMI. If the request fails, the CPN of the wind turbine center returns a response token with a denial message.

We use a CPN Tool [14] that provides communication between CPN nets (i.e., Fusion). It also provides module hierarchy tools (i.e., In, Out, and I/O ports) that are used to connect a sub-CPN into a main-CPN for subroutine processes. In section 6, we use these hierarchy tools to connect attacks into various places of the wind farm CPN to manipulate these tokens and demonstrate attack scenarios.

5. Wind Farm Threat Modeling

This section applies the STRIDE framework to the wind farm system, analyzing threats using the Microsoft Threat Modeling Tool (MTMT). The Data Flow Diagram (DFD) for the reference wind farm is derived from the specification analysis shown in Section 4 and the architecture description presented in Section 2.

Figure 5 presents the data flowing between the wind farm subsystems. The OEM network from the vendor control room, the customer control room server, and the wind farm room server communicate with PCN through secure VPN connections protected by firewalls. The OEM SCADA server communicates with the PCN through the DNP3 SCADA protocol. The PCN communicates with the wind turbine components through a vendor-specific protocol encapsulated in TCP/IP. The PLC of the wind turbine communicates with the master

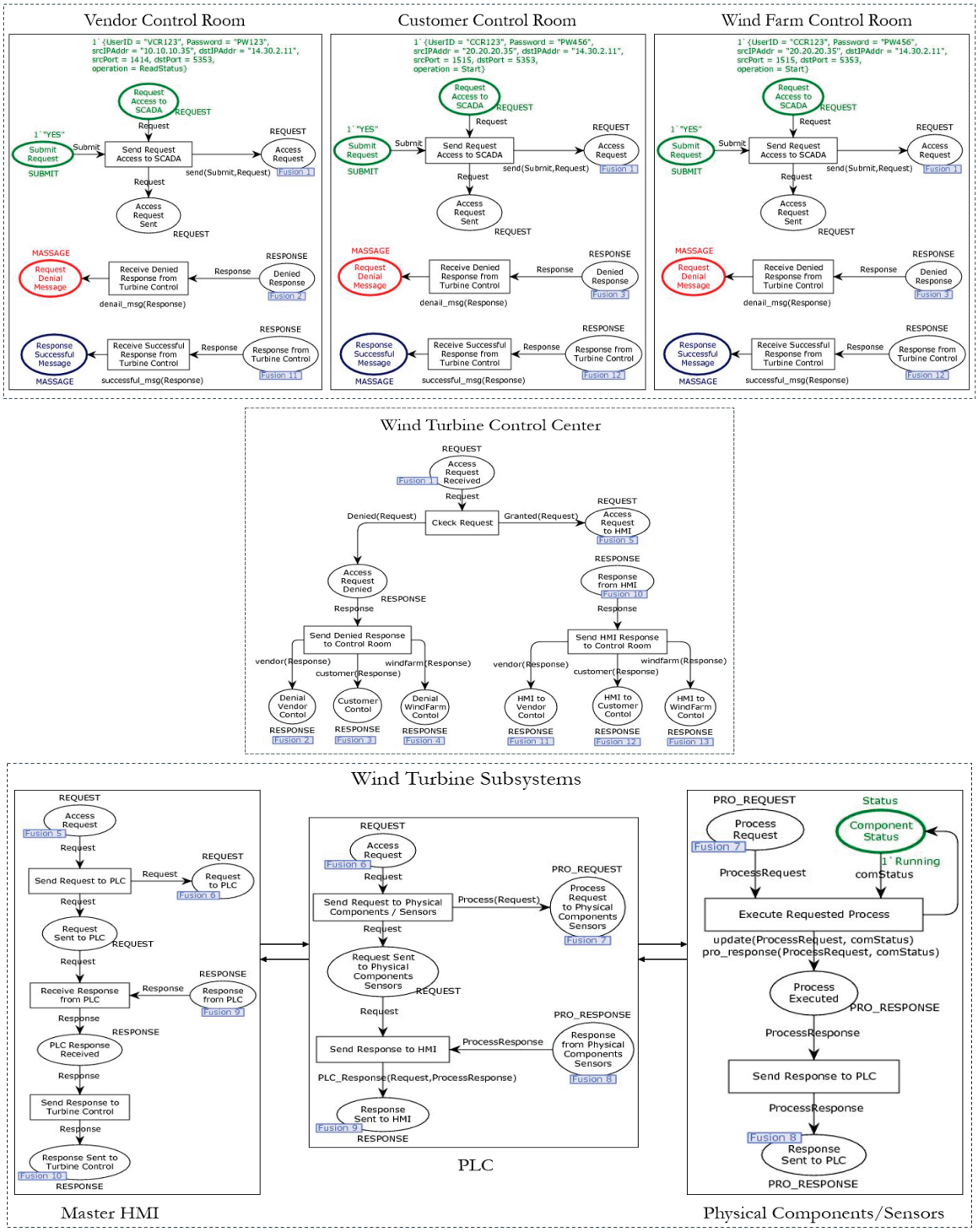

Figure 4: Coloured Petri Nets of the Wind Farm System

and slave HMI through the PROFINET protocol. It connects to the distributed IO through Ethernet, and the distributed IO connects directly to the physical devices and sensors through hardwire. We consider the assets of the wind turbine control center and wind turbine subsystems for threat analysis. The MTTM STRIDE threat modeling analyzes and reports the

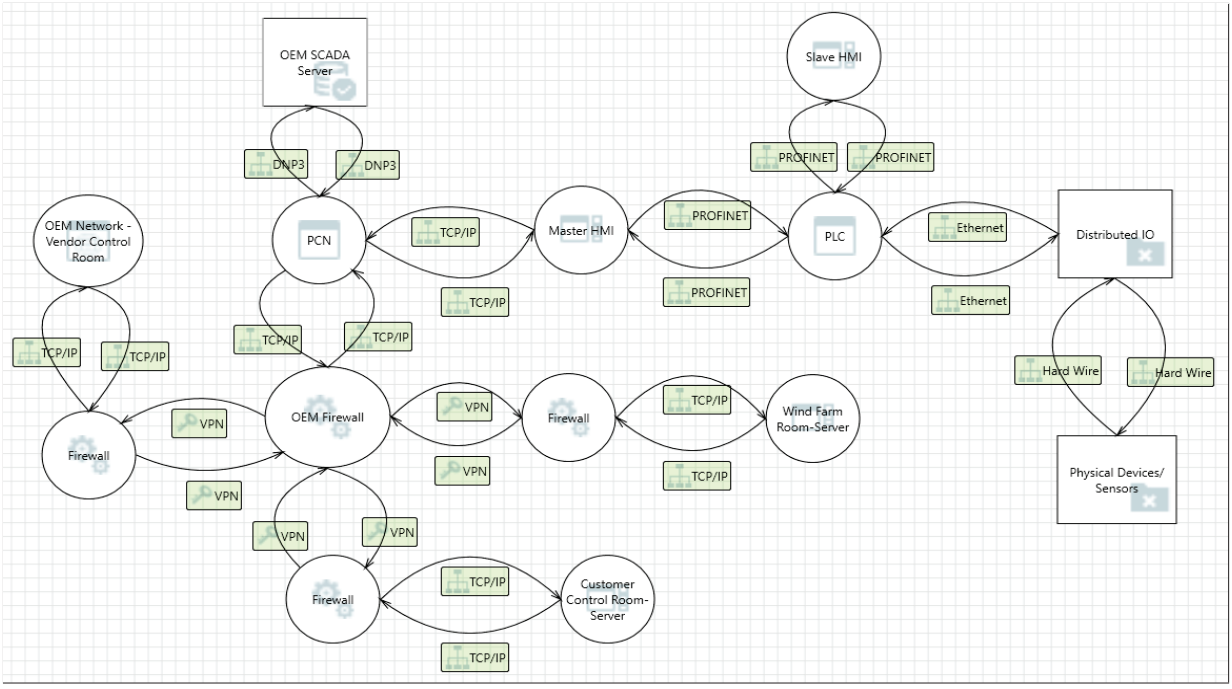

Figure 5: Data Flow Diagram of Wind Farm System

potential threats and consequences for physical and cyber assets. The threats analysis report is summarised as follows.

The OEM SCADA Server

- Spoofing: Incorrect data delivered to the Process Control Network (PCN) can lead to incorrect information to the PLC, resulting in turbine malfunctions and compromising safety.
- Information Disclosure: Improper access control and authentication can lead to unauthorized access to wind farm's critical information.
- Tampering by SQL injection attacks: Data tampering can deliver unintended operational commands to the PLC, leading to wind turbine malfunction.
- Denial of Service: Compromised PCN can lead to resource consumption of SCADA by network flooding, causing loss of availability of SCADA to control and monitor the turbine operation.

DNP3 Protocol

- Spoofing: DNP3 protocol does not facilitate data security. Data transmitted over DNP3 between SCADA and PCN can be spoofed, delivering incorrect wind turbine operational data.
- Tampering: Data communicated over DNP3 can be monitored and modified to cause DoS attacks, compromising the turbine safety.
- Information Disclosure: As DNP3 does not encrypt the data, an unauthorized user can access the information communicated over it, causing a confidentiality attack.

Master HMI

- Tampering: Unauthenticated access to the master HMI can lead to the tampering of wind turbine control parameters in the PLC, compromising safety and availability.
- Elevation of Privilege: Master HMI can impersonate the context of PLC to gain the additional privilege to perform unintended turbine operations, compromising safety.
- Repudiation: Unauthorized access to the master HMI can lead to deleting audit logs and turbine operational data.

TCP/IP Protocol

- Denial of Service: TCP/IP SYN flood can lead to the communication breakdown between the wind turbine components and the PCN, causing DoS and compromising the availability of the PLC to monitor and control the turbine operations.
- Tampering: Data communication can be tampered with reply attacks, causing incorrect operational data.
- Information Disclosure: Lack of encryption can lead to unauthorized access to critical information.

PLC

- Spoofing: Communication between the HMI, the PLC, and PCN can be spoofed due to improper encryption, compromising critical information about wind turbine operation.
- Tampering: Unintended modification of PLC control parameters, software, and firmware can compromise operational safety.

- Repudiation: Deleting PLC logs by unauthorized HMI access can lead to false information about the operation.
- Information Disclosure: Unauthorized access to critical information can cause business loss and operational safety.
- Denial of Service: Unavailability of the PLC can lead to the uncontrolled malfunction of the entire turbine, compromising operational safety.
- Elevation of Privilege: Master HMI or slave HMI can impersonate the context of PLC to gain the additional privilege to perform unintended turbine operations, compromising safety.

These potential threats and their consequences are used to design CPN attacks. As shown in Figure 6, we design 6 CPN attacks: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege.

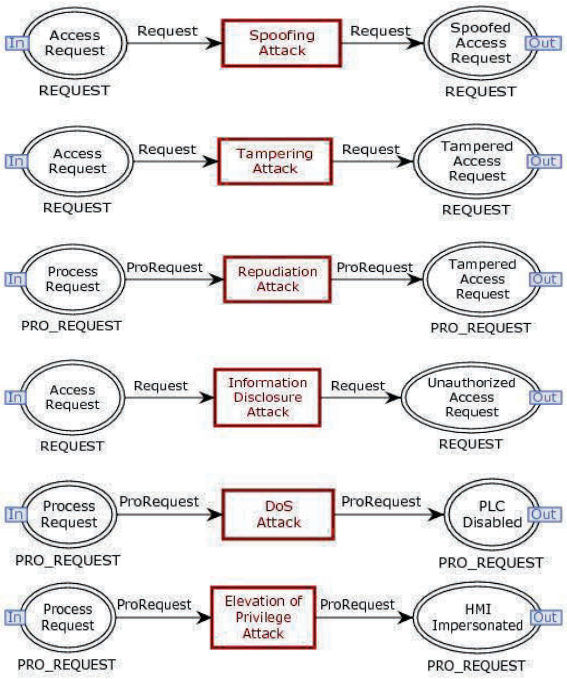

Figure 6: Coloured Petri Nets of Attacks

Each CPN attack requires pre-and postconditions to be attached to the wind farm system's CPN. For instance, the Tampering CPN attack requires an access request token that includes an operation that needs to be executed in one of the wind turbine subsystems (e.g., Read Status). The Tampering CPN attack intercepts this token, changes the operation (e.g., from Read Status to Stop), and passes it back to the wind turbine center.

6. Formal Verification

The CPN attacks are integrated into the wind farm CPN to demonstrate various attack scenarios. Table I examines five attack scenarios focusing on the wind turbine control center (OEM SCADA Server) and wind turbine subsystems, specifically Master HMI and PLC.

Subsystem	Attack	Expected Impact
Wind Turbine Control Center	Spoofing and Tampering	Turbine malfunctions; it may render the PLC unavailable
Wind Turbine Control Center	Spoofing and Information Disclosure	Turbine malfunctions
HMI and PLC	Repudiation and Denial of Service	Unauthorized access to the HMI and rendering the PLC unavailable
Wind Turbine Control Center and HMI	Tampering and Elevation of Privilege	Perform unintended turbine operations and render the PLC unavailable
Wind Turbine Control Center, HMI, and PLC	Tampering, Elevation of Privilege, and Denial of Service	Turbine Control Center disconnected with Wind Turbine, HMI disconnected with PLC, and render the PLC unavailable

TABLE I: Wind Farm Attack Scenarios

The first and second attack scenarios examine attacking only the wind turbine control center with various attacks. The first scenario executes Spoofing and Tampering attacks, and the second executes Spoofing and Information Disclosure attacks. These two scenarios reveal the impact of the wind farm control center being attacked on the entire wind farm system. The third attack scenario focuses on the wind turbine, targeting the HMI and PLC. This scenario executes a Repudiation attack on the HMI and a Denial of Service attack on the PLC. It discloses whether the HMI or the PLC has a critical impact on the wind farm system. The fourth and fifth attack scenarios investigate attacking the wind turbine control center and wind turbine simultaneously. The fourth executes a Tampering attack on the wind turbine control center and an Elevation of Privilege attack on the HMI. In the fifth scenario, the wind turbine control center is attacked by a Tampering attack, an Elevation of Privilege attack hits the HMI, and a Denial of Service attack strikes the PLC. These intensive scenarios show that if one subsystem resists attack, the others cannot, rendering the entire farm system inaccessible.

6.1 State Space Analysis

The analysis first describes the state space of the wind farm system's CPN without attacks. All access requests are legitimate, and no denial response is received. Table II reports that the CPN of the wind farm system is strongly connected components (SCC) and has no live transitions. This indicates that the state model generated by the CPN is acyclic; it does not have infinite loops. The report shows that state space includes 646 nodes connected via 1563 arcs. It also reports one dead marking (state number 646) and 4 dead transitions. Dead markings are states where the CPN is terminated. The report indicates that the wind farm CPN typically terminates at the final state, 646, without disruption. The dead transitions are those transitions that can possibly never occur (a token may not visit them). These dead transitions are:

```
Customer_Control_Room'Receive_Denied_Response_from_Turbine_Control 1
Vendor_Control_Room'Receive_Denied_Response_from_Turbine_Control 1
```

State Space		SCC Graph		Status
#Node	#Arcs	#Node	#Arcs	Full
646	1563	646	1563	
Dead Markings		#Dead Transitions		Live Transition
[646]		4		None

TABLE II: State Space Analysis of The Wind Farm

```
Wind_Farm_Control_Room'Receive_Denied_Response_from_Turbine_Control 1
Wind_Turbin_Control_Center'Send_Denied_Response_to_Control_Room 1
```

These dead transitions represent the denial responses received by the three control rooms: vendor, customer, and wind farm control room. In addition, the turbine control center sends a denial response to these control rooms. The transitions are dead because the access requests are granted, and the denial response transition was not executable.

Now, the analysis examines the state space of the CPN attached to each attack scenario. Table III reports that all CPNs of attack scenarios are strongly connected components (SCC).

Subsystem & Attack	State Space		SCC Graph		Status
Wind Turbine Control Center. Spoofing, and Tampering.	#Node	#Arcs	#Node	#Arcs	Full
	56	108	56	108	
	Dead Markings		#Dead Transition		Live Transition
	[23,24,33,35,56]		11		None
Wind Turbine Control Center. Spoofing and Information Disclosure	#Node	#Arcs	#Node	#Arcs	Full
	54	103	54	103	
	Dead Markings		#Dead Transition		Live Transition
	[20,21,25,54]		11		None
HMI and PLC. Repudiation and Denial of Service	#Node	#Arcs	#Node	#Arcs	Full
	63	122	63	122	
	Dead Markings		#Dead Transition		Live Transition
	[36,37,38,58,59]		5		None
Turbine Control Center, and HMI. Tampering, Elevation of Privilege	#Node	#Arcs	#Node	#Arcs	Full
	74	133	74	133	
	Dead Markings		#Dead Transition		Live Transition
	[42,43,57,74]		3		None
Turbine Control Center, HMI, PLC. Tampering, Elev. of Privilege, DoS	#Node	#Arcs	#Node	#Arcs	Full
	117	253	117	253	
	Dead Markings		#Dead Transition		Live Transition
	[61,62,63,97,98, 104,105,117]		26		None

TABLE III: State Space Report with Attack Scenarios

However, it reports dead markings and dead transitions for each attack scenario, which are considered for verification. Verifying the dead markings signifies the values of the wind farm system's parameters where the system is terminated, whereas verifying the dead transitions signifies why the wind farm system did not perform specific functionality. The state space of the first attack scenario reports 5 dead markings ([23,24,33,35,56]) and 11 dead transitions. The second attack scenario shows 4 dead markings ([20,21,25,54]) and 11 dead transitions. The third attack scenario reports 4 dead markings ([37,58,59,63]) and 5 dead transitions. The fourth attack scenario reports 5 dead markings ([42,43,56,57,58]) and 3 dead transitions. The fifth attack scenario reports 8 dead markings ([61,62,63,97,98,104,105,117]) and 26 dead transitions.

Next, for each attack scenario, the reachability analysis, described below, thoroughly analyzes the dead markings and dead transitions to identify the impacts of attacks on one subsystem to another.

6.2 Reachability Analysis

We use CPN-ML programming language [13] to verify dead marking and dead transition for each attack scenario. We also use built-in functions provided by the CPNTools [14], such as *Reachable(x,y)*, *AllReachable()*, *NodesInPath(x,y)*, *DeadMarking(x)*, *ListDeadMarkings()*, and *ListDeadTransitions()*, for the reachability analysis. These functions return paths (execution sequences of system states), starting from the initial state (submitting an access request) to final states (dead markings where CPN terminates). The paths represent various possibilities: i) an access request is executed, and the response is received with a successful message; ii) an access request is denied, and iii) the access request is executed, but the response is not received. For each attack scenario, we backtrack through these paths and locate the states where the attacks occur. While backtracking, we verify the change of states in the sequence to inspect the reasons that lead to dead transitions and what state values the sequence terminates with at dead markings.

The first attack scenario targeted the wind turbine control center; the wind turbine subsystems (HMI, PLC, physical components) were secured from the attacks. Investigating this scenario's dead markings and dead transitions shows that the impact reaches the control rooms (vendor, customer, and wind farm control room). In the first attack scenario, backtracking the path starting from the dead marking number 56, where the CPN terminates to the initial state where the vendor control room submits an access request to the wind turbine control center, we found that in the state number 28 the first attack, Spoofing attack, occurs. In state number 31, the second attack, the Tampering attack, occurs, and the following states receive a spurious request. The second attack scenario also targeted the wind turbine control center. The impact of the second attack scenario is similar to the first attack scenario; the vendor control room receives a spurious request. In other words, the wind turbine's response impacts the vendor control room. It expected to read the PLC status but received a response that the PLC was stopped.

Investigating the dead markings and dead transitions generated from the third attack scenario indicates that the wind turbine's response impacted the three control rooms and the wind. The wind turbine's response did not impact the wind turbine control center because the attacks occurred on the HMI and PLC (insider attack). The wind turbine control center passes the turbine's response to the control rooms without verification. The fourth and fifth attack scenarios show the entire wind farm was affected. The HMI was compromised by Elevation of Privilege, a DoS attack disabled the PLC, and the turbine control center was bypassed by the Tampering attack. These attacks also affected the three control rooms, which did not receive responses.

7. Wind Farm Assets Criticality Assessment

The interdependency between subsystems causes cascading attack impact. Although the first and second attack scenarios include fewer attacks than the others, they affect the three control rooms, causing them to receive unexpected responses from the wind turbine control center. This is because the first and the second attack scenarios target the wind turbine control center, which runs the SCADA server connecting the three control rooms to the wind turbine subsystems. Thus, the assets of the wind turbine control center are rated high critical assets. Protection of such central control assets must be hardened. A proper authorization mechanism is needed to prevent unauthorized access to the wind turbine subsystems. Cybersecurity strategies, such as asset redundancy, are also required for the SCADA server to reduce the risk of cyberattacks impacting the control rooms.

The formal verification of the third attack scenario indicates that the wind turbine subsystems, HMI and PLC, are more critical assets than the wind turbine control center's assets. This is because these tangible assets are susceptible to cyberattacks, and their impact affects the entire wind farm system, the wind turbine control center, and the control rooms. Thus, the wind turbine assets such as Master and Slave HMIs, PLCs, and physical components are rated as more critical than the wind turbine control center's assets and require high priority for cybersecurity protection. Cyber assets such as DNP3 and TCP/IP communication protocols are vulnerable to confidentiality, integrity, and availability attacks due to insufficient encryption techniques and improper authentication. This can lead to the unavailability or malfunction of the wind turbine if the communication is compromised. Therefore, the communication protocols are considered medium-priority critical assets, requiring proper encryption and message authentication. Cyberattacks on the wind turbine are typically insider attacks; therefore, specific authorization mechanisms can be applied to the communication between the HMI and the PLC. Further, the PLCs require safeguard techniques to prevent intrusion and improper use.

In summary, the wind control rooms are independent, and the impact of cyberattacks does not affect one another. Thus, the assets criticality of these subsystems are rated from medium to low. However, these subsystems depend on the assets of the wind turbine control center, which are dependent on the assets of the wind turbine. Thus, the assets on which other assets are dependent are the more critical ones, and they require high-priority protection.

8. Recommendations for Improvements

Threat modeling and formal verification from Sections 5 and 6 reveal that wind-farm assets are vulnerable to Spoofing, Tampering, Information Disclosure, Denial of Service, and Elevation of Privilege attacks, with potential impact on the availability, confidentiality, reliability, and safety of the turbine operations. Threat analysis shows that improper authentication, improper authorization, and lack of data encryption are the root cause of reported threats. The firewalls protect against these threats with Access Control Lists (ACL), essential access control, and authentication techniques. However, they cannot provide complete protection against specific attacks that can bypass the firewall protection mechanisms and from insider threats. Control devices, such as PLC and HMI, do not incorporate proper authentication and access control mechanisms. Industrial protocols such as DNP3 and MODBUS TCP/IP are not designed with security features.

In addition to generic security measures such as data encryption, authentication and boundary protection, we recommend Attribute Based Access Control to protect the devices from the attacks exploiting access control vulnerabilities. [22] implements the ABAC as a plug-and-play gateway module to address the attacks exploiting improper access control. Wind farm architecture can adapt this technique to strengthen access control features and ensure confidentiality and integrity. Furthermore, the Zero Trust Architecture proposed by [16] can be adopted to mitigate the threats discussed in Section 5

9. Related Work

The ICS assets criticality assessment literature presents various quantitative and qualitative analysis techniques, including simulation-based, graph-based, and algorithm-based techniques.

The algorithm proposed by Akbarzadeh *et al.* [6] determines the criticality of the assets based on Closeness Centrality [CC], Tacit Input Centrality [TIC], and Tacit Output Centrality [TOC] to identify the critical assets. This is an effective algorithm to automate the critical asset identification process. However, considering the complexity of the ICS, it is not sufficient. We can identify certain assets based on closeness centrality and the number of interconnections. However, certain assets may not satisfy this criteria yet, be performing critical functions. For instance, a servo drive in a wind turbine does not have many links and interconnections, yet its failure can lead to a wind turbine's malfunction.

The approach of Liu *et al.* [7] constructs a network topology graph; nodes represent the assets, and edges denote logical or physical connections between them. The number of edges associated with a node indicates the criticality of the corresponding asset. A high number of edges incident on a node makes it critical. An asset's criticality rank depends on the number of edges associated with the node representing the asset. This algorithm considers interconnection information to determine the criticality. The current research on identifying critical assets does not consider intangible assets such as software, firmware, and communication protocols. Additionally, it lacks formal verification of critical assets based on vulnerability exploitation and potential threats.

Shi *et al.* [23] discuss the interaction between the cyber-physical components of power systems. The dynamic and volatile nature of cyber-physical interactions in power systems pose a challenge to exploring the impact of cyber-physical exploits. Therefore, the extensive modeling of the system is essential for quantitative and qualitative analysis of the cyber-physical interactions in the power systems. This paper reviews

various modeling techniques. Graphical modeling techniques include Graph Theory (GT) and Complex Network (CN) theory; the dynamic behavior and cyberattack process are described by Finite State Machine (FSM), Petri Net (PN) models, attack tree models, attack graph models, and state transition diagrams. The system dynamics are modeled with differential-algebraic equations.

The work of Asal Zabetian-Hosseini *et al.* [24] integrates the wind farm SCADA model into the IEEE 9-bus SCADA system to simulate penetration testing. It models the cyberattack scenarios on wind farm SCADA to investigate the overall impact of each attack. It proposes an anomaly cyberattack detection algorithm to stop the attacks that attempt to trip the wind turbines.

10. Conclusion

This paper addresses the assets criticality assessment for ICS. It introduces a systematic methodology that uses a threat modeling framework, namely, STRIDE, to identify relevant security threats for ICS. It also utilizes formal methods, specifically Coloured Petri Nets (CPN), to verify the state transitions of ICS and demonstrate various attack scenarios and their impact. Investigating the attack's impact on wind farm assets directs us on how to assess their criticality. The result shows that the methodology is practical for the ICS verification and assets criticality assessment, providing recommended mitigations to develop robust ICS.

Future work will apply the methodology to other ICS applications to investigate model generalizability. It will also investigate the computational complexity, focusing on the model state space explosion when the system size and the number of attacks increase. We will use various threat modeling frameworks such as PASTA (Process for Attack Simulation and Threat Analysis). We will also use additional formal models, such as UPPAAL and SPIN, for verification. A comparison of the use of these threat modeling frameworks and formal models will be provided.

Acknowledgment

This work was supported in part by funding from NIST under Award Number 60NANB23D152 and from NSF under Award Numbers CNS 2335687, DMS 2123761, CNS 1822118, NIST, ARL, Statnett, AMI, NewPush, and Cyber Risk Research.

References

[1] S. Freeman, M. Kress-Weitenhagen, J. Gentle, M. J. Culler, M. M. Egan, and R. V. Stolworthy, "Attack surface of wind energy technologies in the united states," Idaho National Laboratory (INL), Idaho Falls, ID (United States), Tech. Rep., 2024.

[2] E. Ambarita, I. Kuncara, A. Widyotriatmo, A. Karlsen, F. Scibilia, and A. Hasan, "On cyber-attacks against wind farms," in *Annual Conference of the IEEE Industrial Electronics Society (IECON)*. IEEE, 2023, pp. 1–6.

[3] N. Farrar and M. H. Ali, "Cyber-resilient converter control system for doubly fed induction generator-based wind turbine generators," *Electronics*, vol. 13, no. 3, p. 492, 2024.

[4] N. Tatipatri and S. Arun, "A comprehensive review on cyber-attacks in power systems: Impact analysis, detection and cyber security," *IEEE Access*, 2024.

[5] J. Staggs, D. Ferlemann, and S. Shenoi, "Wind farm security: attack surface, targets, scenarios and mitigation," *International Journal of Critical Infrastructure Protection*, vol. 17, pp. 3–14, 2017.

[6] A. Akbarzadeh and S. K. Katsikas, "Identifying critical components in large scale cyber-physical systems," in *International Conference on Software Engineering ICSE*. Seoul, Republic of Korea: ACM, 2020, pp. 230–236.

[7] C. Liu, Y. Alrowaili, N. Saxena, and C. Konstantinou, "Cyber risks to critical smart grid assets of industrial control systems," *Energies*, 2021.

[8] L. Kohnfelder and G. Praerit, "The threats to our products, microsoft interface," *Microsoft Interface, Redmond, WA, USA: Microsoft Corporation*, 1999.

[9] B. Yang and Y. Zhang, "Cybersecurity analysis of wind farm industrial control system based on hierarchical threat analysis model framework," in *2022 International Conference on Computing, Communication, Perception and Quantum Technology (CCPQT)*. IEEE, 2022, pp. 6–13.

[10] L. Ordinez, G. Eggly, M. Micheletto, and R. Santos, "Using uml for learning how to design and model cyber-physical systems," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, pp. 50–60, 2020.

[11] A. Shostack, *Threat modeling: Designing for security*. John Wiley & Sons, 2014.

[12] K. Jensen, L. M. Kristensen, and L. Wells, "Coloured petri nets and CPN tools for modelling and validation of concurrent systems," *International Journal on Software Tools for Technology Transfer*, vol. 9, no. 3-4, pp. 213–254, 2007.

[13] K. Jensen and L. M. Kristensen, *CPN ML Programming*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 43–77.

[14] A. V. Ratzer, L. Wells, H. M. Lassen, M. Laursen, J. F. Qvortrup, M. S. Stissing, M. Westergaard, S. Christensen, and K. Jensen, "Cpn tools for editing, simulating, and analysing coloured Petri Nets," in *Proceedings of the International conference on application and theory of Petri Nets (ICATPN)*. Eindhoven, Netherlands: Springer, 2003, pp. 450–462.

[15] J. a. A. Custódio Soares, B. Lima, and J. a. Pascoal Faria, "Automatic model transformation from UML sequence diagrams to Coloured Petri Nets," in *Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*. Funchal, Madeira, Portugal: SciTePress, 2018, p. 668–679.

[16] S. Gowdanakatte, I. Ray, and M. Abdelgawad, "Model based risk assessment and risk mitigation framework for

cyber-physical systems," in *2023 5th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. IEEE Computer Society, 2023, pp. 203–212.

[17] X. Li, Z. Liu, and J. He, "A formal semantics of UML sequence diagram," in *15th Australian Software Engineering Conference (ASWEC 2004), 13-16 April 2004, Melbourne, Australia*. IEEE Computer Society, 2004, pp. 168–177.

[18] A. Alhroob, K. Dahal, and A. Hossain, "Transforming UML sequence diagram to high level Petri Net," in *2010 2nd International Conference on Software Technology and Engineering*, vol. 1, 2010, pp. V1–260–V1–264.

[19] N. Yang, H. Yu, H. Sun, and Z. Qian, "Modeling UML sequence diagrams using extended Petri Nets," *Telecommunication Systems*, vol. 51, no. 2-3, pp. 147–158, 2012.

[20] Q. Li and Y.-L. Chen, "Data flow diagram," in *Modeling and Analysis of Enterprise and Information Systems*. Springer, 2009, pp. 85–97.

[21] M. Da Silva, M. Puys, P.-H. Thevenon, S. Mocanu, and N. Nkawa, "Automated ics template for stride microsoft threat modeling tool," in *Proceedings of the 18th International Conference on Availability, Reliability and Security*, ser. ARES '23. New York, NY, USA: Association for Computing Machinery, 2023.

[22] S. Gowdanakatte, M. Abdelgawad, and I. Ray, "Security hardening of industrial control systems using attribute based access control," in *Proceedings of the 9th Annual Industrial Control System Security Workshop (ICSS ACSAC)*. Austin, TX, USA: ACSAC, 2023.

[23] L. Shi, Q. Dai, and Y. Ni, "Cyber–physical interactions in power systems: A review of models, methods, and applications," *Electric Power Systems Research*, vol. 163, pp. 396–412, 2018.

[24] A. Zabetian-Hosseini, A. Mehrizi-Sani, and C.-C. Liu, "Cyberattack to cyber-physical model of wind farm scada," in *Annual Conference of the IEEE Industrial Electronics Society (IECON)*. IEEE, 2018, pp. 4929–4934.