2023 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm) | 978-1-6654-5556-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/SmartGridComm57358.2023.10333970

On Grid Resiliency: Cyber-Physical Detection Tool Evaluated in a Multi-Stage Attack Scenario

Leen Al Homoud, Namrata Barpanda, Ana Goulart, Katherine Davis, *Texas A&M University;*
Mark Rice, *Pacific Northwest National Laboratory*

***Abstract*—Electric power systems are comprised of cyber and physical components that are crucial to grid resiliency. Data from both components should be collected when modeling power systems: data from communication networks and intrusion detection systems; physical telemetry from sensors and field devices. For accurate and timely detection of malicious activity, should we always account for cyber and physical telemetry data, or *data fusion*? To further investigate the application of data fusion, this paper presents a new threat scenario in which an adversary affects power generation. It is a multi-stage strategy that includes a database intrusion. Multiple industrial communication protocols are applied in a cyber-physical testbed. Packets and alarms are collected using our cyber-physical data fusion engine, and evaluated using an autoencoder algorithm. It predicted malicious packets with high precision at an early stage of the scenario, using cyber-only telemetry.**

I. Introduction

Power systems are critical infrastructures which require reliability and resilience at all locations and events throughout the system. To obtain high resiliency and reliability, power systems need to be modeled, monitored, and controlled as cyber-physical systems. They include the operation of all subsystems including the power grid and the communications network through which field devices are monitored and controlled. In this way, the Cyber-physical Resilient Energy Systems (CYPRES) project has been developing a tool suite of different cyber-physical detection, mitigation, and response algorithms to establish a secure and resilient Energy Management System (EMS) [1]. This will help power systems to achieve cyber-physical intrusion response and situational awareness [2].

The goal of this work is to demonstrate CYPRES detection and mitigation algorithms through the defense against an intricate multi-stage cyber threat. Under advanced persistent threats [3], an unauthorized user may be able to get access to the energy utility's communications network. After this initial compromise, the intruder can escalate privileges to perform reconnaissance on the system's operations. In our scenario, the intruder then moves laterally through the system. To disrupt the physical system, the intruder's goal is to deplete the real power reserves of the system's generating units via the Balancing Authority (BA). As in Figure 1, this attack has four stages, based on the MITRE ATT&CK Framework [4]: reconnaissance and initial access, persistence and privilege escalation, lateral movement, and physical impact. It involves several steps, which are described in detail in Section III.

This work was supported by the US Department of Energy under award DE-OE0000895.

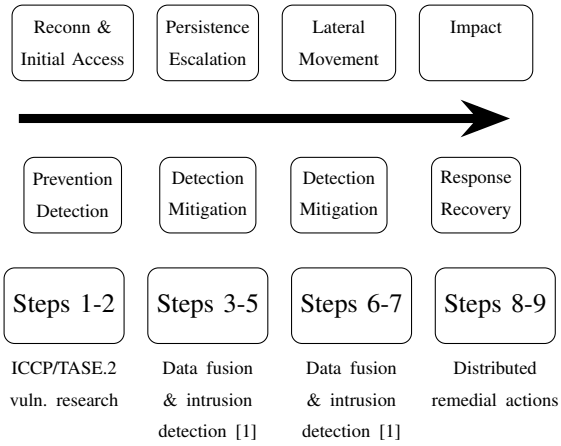

Fig. 1. Multi-stage threat and defense using MITRE's framework.

This multi-stage scenario is modeled and implemented in a cyber-physical testbed called the Resilient Energy System Laboratory (RESLab) [5]. It has the CYPRES EMS which implements several tools that have been evaluated in other detection use cases. In the cases described in [6], a data fusion engine was developed and demonstrated to detect cyber attack, where data fusion is a defense technique that fuses both cyber and physical system data to detect anomalous system behavior. This data fusion engine was evaluated to detect man-in-the-middle (MiTM) attacks on Distributed Network Protocol 3 (DNP3) messages. It concluded that co-training of machine learning (ML) algorithms with the cyber-physical features, instead of just cyber or physical data, improved performance by an additional 15-20% [6]. We propose to extend and evaluate this data fusion detection in a new, multi-stage attack scenario.

The main contributions of this paper are as follows:

- To present a cohesive defense against physical power system impact by considering threat escalation and data indicators throughout the stages of a multi-stage attack.
- To model a scenario in a realistic, multi-organizational synthetic electric utility communications network in a high-fidelity cyber-physical power system testbed.
- To present how to use data fusion and machine learning (ML) to analyze data from multiple communication protocols (DNP3, Inter-Control Center Protocol (ICCP), Structured Query Language (SQL)) to avoid the impacts of an attack on one utility influencing another.

This paper focuses on the vital need to model threats in power system critical infrastructure using a risk-based approach, throughout an event. The threat modeling and mitigations in this scenario follow the resilience life cycle, which includes how a utility will prepare for, withstand, and respond to different threats, while learning from events to better plan the system [7].

This paper is organized as follows. Section II provides the foundational concepts for our scenario, followed by the threat scenario in Section III. Section IV describes the changes in the RESLab testbed, the implemented scenario and results from the data fusion engine. This paper is concluded in Section V.

II. Background

This section reviews this scenario's building blocks: ICCP, SQL injection, and Automatic Generation Control (AGC).

A. ICCP/TASE.2 Protocol

ICCP is a communication protocol that allows utility control centers to exchange data, usually with a BA. It is also known as IEC 60870–6/TASE.2, where TASE means Telecontrol Application Service Element. Figure 2 illustrates the ICCP architecture [8]. At the application layer, ICCP uses the Manufacturing Message Specification (MMS) protocol, which is defined by ISO 9506 and used in IEC 61850 protocols – Communication Networks and Systems in Substations. MMS defines how to name and format the data [9], while ICCP defines methods to request and report data. An implementation of this architecture is presented in [10]. Their ICCP libraries are the same ones used in our testcase scenario.

The vulnerabilities of the ICCP architecture are described in [11]. As ICCP does not have any in-built encryption algorithm, it relies on Transport Layer Security (TLS), or Virtual Private Networks (VPN). The survey [12] explains the security requirements of ICS, such as confidentiality, integrity, and availability. But as TLS relies on Public Key Infrastructure (PKI), the interoperability between utilities and BA's becomes difficult if there are many PKI certificates to manage [13]. Moreover, although IEC 62351 standard proposes security measures for IEC protocols, they are not implemented often.

A Common Vulnerability and Exposure (CVE) has been recently found for ICCP [14]. It explains how an intruder can cause problems to the Supervisory Control and Data Acquisition (SCADA) system and ICCP nodes. In CVE-2022-29490, an intruder is able to login to the utility's Web interface, and then execute the SCADA system's internal scripts. CVE-2022-2227 addresses a validation flaw in ICCP messages when an adversary sends data with timestamps in the future. If all ICCP nodes experience the same time validation flaw, it can lead to a denial-of-service (DoS) threat.

B. SQL Injection

SQL is a client-server application responsible for managing, programming, and querying relational databases, which store data for various physical systems. The SQL server commonly offers a web interface. In cases where this interface is poorly constructed and fails to properly assess user inputs, there exists the potential for exploiting SQL vulnerabilities.

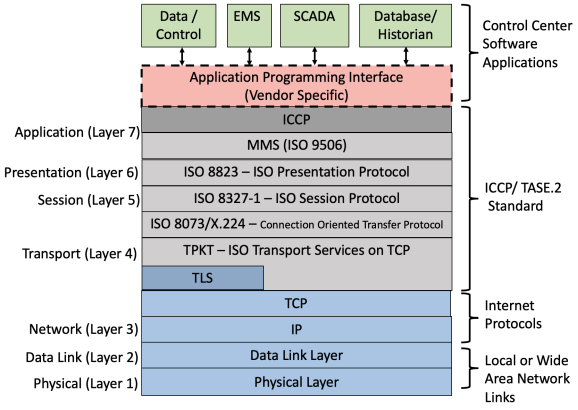

Fig. 2. ICCP protocol stack and its software applications.

Any input provided by users to a susceptible web application, when processed by a database, can become a pathway for unauthorized activities. For example, in [3], a vulnerability is detailed in the context of home area networks. This vulnerability illustrates how data regarding power consumption, stored within an SQL database from smart meters, could be jeopardized. Another research study [15] outlines instances of manipulation targeting power dispatching systems, facilitated by leveraging SQL injection methods. Such incidents occur when an individual with malicious intent infiltrates the utility's network. Expanding on this issue, an IBM report [16] explains that a significant proportion (60%) of incidents targeting energy utilities in 2016 were instances of data injection.

C. Automatic Generation Control (AGC)

An AGC system receives power flow and frequency measurements from sensors at substations, and outputs control commands to keep the frequency stable. It is a real-time control application, sensitive to the measurements it receives. For example, the authors in [17] simulated data integrity threats on AGC, impacting the SCADA system. In [18], the authors studied cyber attacks that caused frequency disturbances in the power system and proposed mitigation techniques.

A threat scenario where false data is injected to an AGC is presented in the paper by Sridhar and Govindarasu [19], where the AGC's integrity is compromised by corrupted measurements. Scaling, random, and pulse attacks were used to change the measurements, which triggered the AGC to modify the generator operating points, or set points. Their detection used ML models that were trained to detect anomalies based on load forecast and comparing it to the commands issued by the AGC. The authors simulated load and generation data, but did not use any cyber data. In our paper, we use data from both the physical and the cyber systems to detect the incident. Details on how we implement AGC and what data is collected will be discussed in Section IV-C.

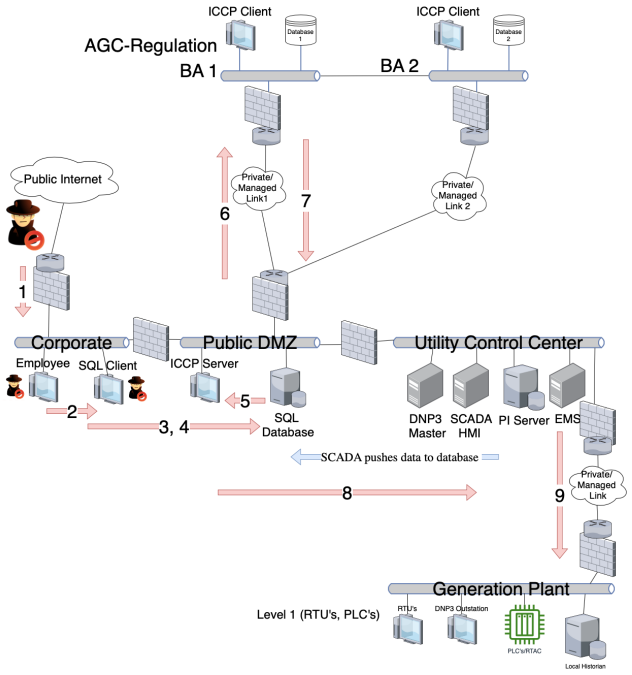

Fig. 3. Network topology for multi-stage threat scenario with all steps.

III. Use Case Scenario

Our scenario assumes a SCADA system for power generation. A fleet of generators are simulated using PowerWorld Dynamic Studio (PWDS), an interactive transient stability environment. PWDS runs a physical model with nine buses and an AGC system. The generators' data are stored in an SQL database. False data injected to the database compromises the generation reports sent to BA. This false data can impact the AGC in one area and cause the BA to issue a control command to alter the generation set points. The communication network for this use case is shown in Figure 3. The arrows show the attack stages:

Stage 1 - Reconnaissance and Initial Access

- **Step 1:** Intruder scans and finds an open network port to the corporate network.
- **Step 2:** Intruder reaches corporate network. Employee machines can access SQL database via Web interface.

Stage 2 - Persistence and Privilege Escalation

- **Step 3:** Intruder performs an SQL injection in database's Web interface and gets login credentials to database.
- **Step 4:** Intruder manipulates generator values in database.

Stage 3 - Lateral Movement

- **Step 5:** Utility's ICCP server reads data from database.
- **Step 6:** ICCP server sends false data to BA.

Stage 4 - Physical Impact

- **Step 7:** BA performs AGC calculations using received false data. BA's ICCP client sends a set point command to the ICCP server in the utility.
- **Step 8:** Utility's ICCP server informs the SCADA master to change generation settings.
- **Step 9:** The command is sent to the generation plant.

These steps illustrate how intruders can compromise the utility's data once they enter its network. Therefore, it is important to test defense mechanisms against them.

IV. Experimental Results

A. Testbed Updates

The RESLab testbed [20] is a cyber-physical testbed with virtual machines (VMs) and physical devices (Figure 4). The VMs are created using VMWare's *vSphere* virtualization platform. One VM runs PWDS, the interactive power simulator that also represents the outstation. A network emulator – Common Open Research Emulator (CORE) – on another VM emulates the communication network. It interconnects the utility control center and the outstation. The SCADA master runs the open source PyDNP3 software libraries. Overseeing all the VMs, the CYPRES EMS [1] collects cyber and physical telemetry and performs data fusion-based defense. The dataset for this multi-stage threat use case can be found in [21].

The power system configured in RESLab's emulated outstations is the 9-bus test case from the Western System Coordinating Council (WSCC) [22] (Figure 5). This system is divided in two areas: Area 1 with generators 1 and 3, and Area 2 with generator 2. Our BA receives data from generators 2 and 3. In addition to this 9-bus system and AGC algorithm, the following updates have been made to RESLab:

- Created an SQL database to save generators' data,
- Added BA network with an ICCP client,
- Added public demilitarized zone (DMZ),
- Implemented real-time ICCP client/server application,
- Implemented and tested new data fusion using telemetry from different parts of the system,
- Updated intrusion detection system (IDS) – SNORT – with local rules to detect new threats.

The ICCP server sends the BA *data set transfer sets*, which report analog values. Based on the 9-bus power system in Figure 5, we configured the ICCP server to report the Generator 2 and 3 values stored in the database.

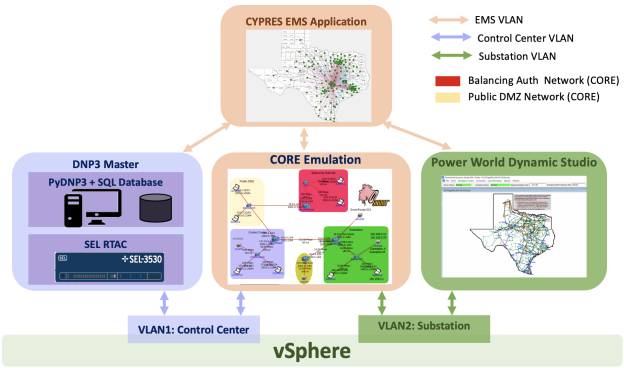

Fig. 4. RESLab configuration, updated from [20].

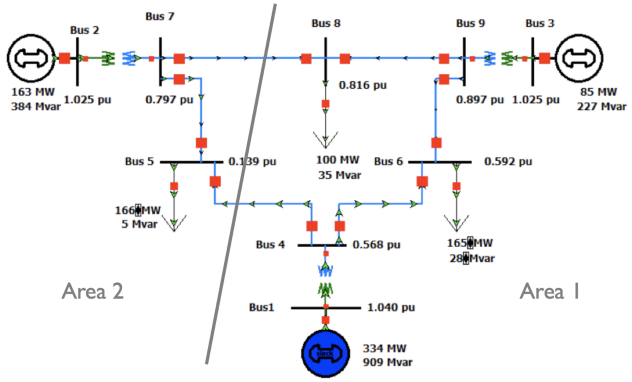

Fig. 5. Power system scenario - WSCC 9-bus system

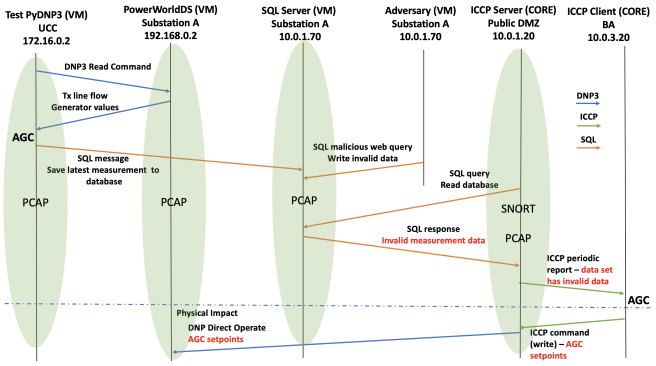

Fig. 6. Scenario timeline with cyber telemetry shown in the oval shapes.

B. Scenario Timeline

The use case's events are illustrated in Figure 6. First, the 9-bus simulation in PWDS starts as a continuous, real-time simulation. A Python script at PyDNP3 master sends DNP3 read commands to the outstation to read analog outputs for the generators, their frequencies, and power flows between the buses. Then, the master calculates Area Control Error (ACE) and set point values and stores them in the database. Next, an unauthorized user performs the SQL injection step, assuming Steps 1 and 2 of the *Reconnaissance and Initial Access* stage have happened. The ICCP node accesses the false data and sends it to the BA.

The BA runs the AGC for an area, and sends the utility new setpoints. We assume that this ICCP command is sent directly to the utility. Then, the DNP3 master sends the DNP3 Direct Operate command to the generation plant. For this paper, we focus on the steps above the dotted line in Figure 6, i.e., Stages 2 and 3 in the scenario from Section III.

Packet captures (PCAP) were collected in [21]:

- CORE VM: contains the scripts that run the ICCP client, server, and SNORT,
- PowerWorld VM: contains the outstation and the WSCC 9-bus system simulation,
- TestPyDNP3 VM: runs the DNP3 master script that reads data from the outstation and sends it to the database,

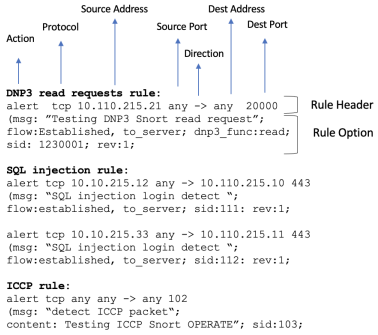

Fig. 7. SNORT rule syntax (above) and our custom/local rules.

- Substation A VM: contains the SQL database along with the software that performs the SQL injection.

Packets are filtered to allow TCP ports 20000 (DNP3), 443 (Web), and 3306 (MySQL protocol).

The second component in the data collection is the SNORT log file, using SNORT version 2.9.14-1. SNORT detection method recognizes suspicious activity from packets, based on the rules defined in Figure 7. Their goal is to detect DNP3 packets between outstation and master, SQL injection using the Web interface (TCP port 443), and ICCP traffic.

C. Physical Data Collection

For power system applications and operation, the purpose of an AGC system is to minimize the Area Control Error (ACE). The ACE is the difference between the actual and scheduled power flow between two different areas, as in (1),

$$ACE = P_{act} - P_{sched} \tag{1}$$

where P_{act} is the actual power flow and P_{sched} is the scheduled power flow between two areas.

The ACE value [23] considers the produced electricity's nominal and measured frequencies, the frequency bias factor, and the sum and initial values of the power flow, as in (2),

$$\begin{aligned} ACE = (f_{meas} - f_{nom}) * 10 * B + \\ (tieflows_s - tieflows_i) \end{aligned} \tag{2}$$

where f_{meas} is the measured bus frequency, f_{nom} is the nominal frequency of 60 Hz, B is the frequency bias factor in MW/0.1Hz, $tieflows_s$ is the sum of tie flows between two areas at a specific time, and $tieflows_i$ is the initial sum of tie flows between two areas.

AGC's purpose is to minimize the ACE. The ideal ACE value is zero (the actual power flow should match the scheduled power flow). Another important variable to consider is the participation factor P of the generators, which is defined as the amount of real power that a generator contributes relative to the amount of change of load consumption in the system. With the use of the ACE and the participation factor P, the AGC setpoint can then be calculated as in (3),

$$Gen_{setpoint} = Gen - 2 * ACE * P_f \tag{3}$$

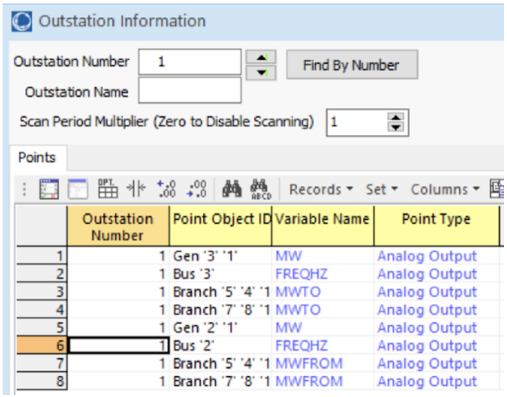

Fig. 8. DNP3 data points configured in our PWDS simulation.

where Gen is the generator's power output, $Gen_{setpoint}$ is generator's set point, ACE is the Area Control Error, and P_f is the participation factor. Hence, if the data used in the calculations comes from rogue sensors or is altered, the AGC's integrity will be compromised.

To calculate the ACE in (2), we collect eight DNP3 data points, as shown in Figure 8 and listed below [21]:

- Point0 (Gen): generator 3 value (90 MW)
- Point1 (f_{meas}): frequency of generator 3 (59.998 Hz)
- Point2 ($tieflow$): tie flow from bus 5 to 4 (64.135 MW)
- Point3 ($tieflow$): tie flow from bus 7 to 8 (-63.4748 MW)
- Point4 (Gen): generator 2 value (126.335 MW)
- Point5 (f_{meas}): frequency of generator 2 (59.998 Hz)
- Point6 ($tieflow$): tie flow from bus 4 to 5 (-63.6673 MW)
- Point7 ($tieflow$): tie flow from bus 8 to 7 (63.8081 MW)

The measured frequency f_{meas} should be close to 60 Hz with no disturbance to the system. To calculate the tie flows, four collected values are used: power flow in both directions for each of the two branches in the power system (Branches 4-5 and 7-8). As such, the tie flow for Area 1 is the summation of the tie flows going from bus 4 to 5 and bus 8 to 7 (Point6 and Point7, respectively). For Area 2, the tie flow is the summation of the tie flows going from bus 5 to 4 and bus 7 to 8 (Point2 and Point3, respectively). The $tieflow_i$ is set to the initial tie flow measurement from the simulation. Then, the tie flows are collected continually as the simulation runs. Once the ACE value is calculated, the generation values and the participation factors (P_f) are used to calculate the $Gen_{setpoint}$. For Areas 1 and 2, *Gen3* and *Gen2* values are used, respectively. The P_f value is assumed to be 1, meaning both generators contribute equally to changes in generation.

D. Performing the SQL Injection

In the RESLab testbed, we performed Stage 2 of the scenario on the SQL's database Web interface using HTTP GET and POST requests. A query is sent to the authentication page, as shown in Algorithm 1. The "OR 'a'='a'" expression causes the authentication check to be ignored [24]. The intruder can now retrieve all the data in the database including account details. The malicious actor can also modify the generator values from the tables or drop the tables completely. Figure 9 shows the utility's database after the SQL injection, where all values of *gen2* column were altered to $250MW$.

E. Defense - Data Fusion

Data fusion is the process of combining and integrating information from multiple sources to produce a more complete and accurate representation of the information that underlies it. In the CYPRES EMS [1], the data fusion engine [6] combines data from cyber and physical sensors, performs multiple steps to process and integrate this data, and then applies machine learning techniques to identify anomalies, and improve the understanding of the monitored system's behavior. Here are the steps it performs:

- Collects packet captures using Wireshark (*cyber_table*),
- Collects logs from the IDS SNORT (*snort_table*),
- Extracts physical data from DNP3 (*physical_table*),
- Extracts DNP3 payload data points (*dnp3_table*),
- Merges the *snort_table* with the *cyber_table*,
- Merges *cyber_table*, *physical_table* and *dnp3_table*,
- Encodes and normalizes the data in the tables,
- Analyzes the data using machine learning techniques.

After the *cyber_table* and *snort_table* are collected, they are merged resulting in 17 cyber features: *['Time', 'frame.len', 'frame.protocols', 'eth.src', 'eth.dst', 'ip.src', 'ip.dst', 'ip.len', 'ip.flags', 'tcp.srcport', 'tcp.dstport', 'tcp.len', 'tcp.flags', 'tcp.nxtseq', 'tcp.ack', 'snort.alert', 'snort.alert.type']*

The *snort.alert* indicates whether an alert was triggered for the observed network traffic, and the *snort.alert.type* specifies the type of alert triggered. In Figure 10, an operator can see SNORT alerts for SQL injection followed by DNP3 read request messages.

Those tables are merged with *physical_table* and *dnp3_table*, with 18 features: *['LL.dnp3.src', 'LL.dnp3.dst', 'LL.dnp3.len', 'LL.dnp3.ctl', 'TL.dnp3.tr.ctl', 'AL.dnp3.al.func', 'AL.dnp3.al.ctl', 'AL.dnp3.obj', 'DNP3 Object Count', 'DNP3 Objects', 'Point0', 'Point1', 'Point2', 'Point3', 'Point4', 'Point5', 'Point6', 'Point7']*

Algorithm 1 SQL Injection on Generator SQL Server

1. *Client sends query to log in*
2. $SELECT * FROM\ users\ WHERE\ username$ = 'X' $AND\ password$ = '$example$' OR 'a'='a'
3. *Server returns TRUE. Bypass authentication.*
4. *Client accesses data and modifies one column*
5. *Use Postman API to change gen2 columnn*

id	time	gen2	gen3	ACEgen2	ACEgen3	spGen2	spGen3
1	12:20:59	250	90	0	0	126.335	90
2	12:20:59	250	90	0	0	126.335	90
3	12:21:04	250	90	0.0001	0.0001	126.335	89.9998
4	12:21:09	250	90	0.0001	-0.00000000000000710543	126.335	90
5	12:21:14	250	90	0	0.0001	126.335	89.9998
6	12:21:19	250	90	0.0001	0.0001	126.335	89.9998
7	12:21:24	250	90	0.0001	-0.00000000000000710543	126.335	90
8	12:21:29	250	90	0.0001	-0.00000000000000710543	126.335	90
9	12:21:34	250	90	0.0001	-0.00000000000000710543	126.335	90
10	12:21:39	250	90	0.0001	-0.00000000000000710543	126.335	90
11	12:21:44	250	90	-0.0000...	-0.00000000000000710543	126.335	90

Fig. 9. SQL database with false data for Generator 2.

```
04/11-01:07:43.028651  [**] [1:111:1] SQL injection detected  [**] [Priority: 0] {TCP} 10.110.215.12:51910 -> 10.110.215.10:443
04/11-01:07:43.028722  [**] [1:111:1] SQL injection detected  [**] [Priority: 0] {TCP} 10.110.215.12:51910 -> 10.110.215.10:443
04/11-01:07:43.031027  [**] [1:111:1] SQL injection detected  [**] [Priority: 0] {TCP} 10.110.215.12:51910 -> 10.110.215.10:443
04/11-01:07:43.031047  [**] [1:111:1] SQL injection detected  [**] [Priority: 0] {TCP} 10.110.215.12:51910 -> 10.110.215.10:443
04/11-01:07:46.795365  [**] [1:1230001:1] Testing DNP3 Snort read request [**] [Priority: 0] {TCP} 10.110.215.21:48265 -> 10.110.215.
25:20000
04/11-01:07:51.797351  [**] [1:1230001:1] Testing DNP3 Snort read request [**] [Priority: 0] {TCP} 10.110.215.21:48265 -> 10.110.215.
25:20000
04/11-01:07:56.798371  [**] [1:1230001:1] Testing DNP3 Snort read request [**] [Priority: 0] {TCP} 10.110.215.21:48265 -> 10.110.215.
25:20000
```

Fig. 10. SNORT alerts seen during the experiment.

Fig. 11. Sample of cyber features: frame length, frame protocols, TCP port, and snort alert flag, before and after the attack.

Fig. 12. Sample of physical features: DNP3 function code, Gen2 value and frequency, and SNORT alert type, before and after the attack. For the SNORT alert type, 1 means a DNP3 request, 2 means the SQL injection.

The *LL.dnp3.len* is the size of the DNP3 packet, such as 11 bytes for read requests or 55 bytes for responses. The *AL.dnp3.al.func* is the DNP3 function code, with two values: $0x01$ for read request and $0x81$ for solicited response. The *DNP3 Object Count* is the number of objects in each message.

After merging all tables by time, the *merged_table* has 35 features. Most samples do not have values for the physical features. Therefore, after normalizing the data, we fill the missing data with average values.

Figures 11 and 12 show some cyber and physical features from the *merged_table* that a plant operator would see. When the attack happens, changes in the cyber features, such as frame lengths and protocols, are very clear. Hence, early stage attack detection and mitigation can be done. Thus, there is no impact on the physical system, and the Generator 2 and frequency values remain constant as $126MW$ and $60Hz$. However, if the attack is not detected, the ICCP data to the BA will contain the false Generator 2 value of $250MW$.

In addition to SNORT intrusion detection, as a corroborating defense, what would happen if we ran a machine learning model to predict which packets were malicious? Which dataset, the *cyber_table* or the *merged_table*, would give us better accuracy in predicting which packets triggered the SNORT alert?

To do this, we used an unsupervised ML method called an autoencoder. It compresses the data to a reduced state space, then it reconstructs it. Autoencoders have recently been demonstrated for robust AI-enabled detection in power systems cybersecurity, with one chosen here based on the rationale of [25], to be tested as part of our data fusion engine.

The autoencoder model used in our experiments has as input layer the same number of neurons as the features of our data. We first tested it with 14 cyber features, after eliminating *time* and *snort_alert_type*, resulting in 14 neurons. We evaluated models with 6, 8 and 10 layers of encoding. The 8-layer model had the best results. After each layer, sigmoid function was used as the activation function. We then tested the model using all 32 cyber and physical features, also using an 8-layer model. For instance, with 32 features, the encoder had layers with 32, 32, 16, 8, 4, 2, 1, 0 units then back again.

In both cases, the model was trained using benign traffic from the outstation. The *merged_table* had no malicious packets. Then, the model was tested with all packets, including packets that triggered the alerts. SNORT decodes the intercepted packets to retrieve data from the headers and payloads, and the packet itself that is being transported. Thus, we added the decoded SNORT packets to the *merged_table*, which now contains benign and malicious packets.

The performance of the autoencoder model are presented in Table I, which shows the performance of the autoencoder algorithm had a 96% precision using the 14 cyber features where all SNORT alert packets were detected correctly. The confusion matrix (Figure 13) shows 8 benign packets were detected as malicious for cyber-only data, as opposed to 36 benign packets for cyber-physical data.

The results validate two different but mutually supportive layers of defense: a dashboard where operators can see cyber-

Strategy	Accuracy	Precision	Recall	F1-Score
Cyber-only	0.98	0.96	1	0.98
Cyber-physical	0.80	0.81	0.77	0.79

TABLE I
PERFORMANCE RESULTS OF ALARM PREDICTIONS.

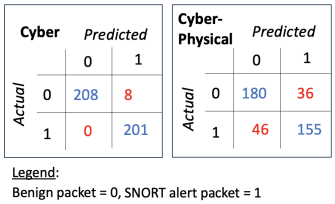

Fig. 13. Confusion matrix results.

physical features and intrusion detection alarms, and an automation of the intrusion analysis using unsupervised ML. The dashboard shows the SQL injection's SNORT alarms. However, there are many features and alarms that may be overlooked. Hence, we use an autoencoder to compress and compare the reconstructed data between benign traffic and all traffic including alarms. The cyber telemetry used in this way is sufficient to predict the alarms at Stage 2 – persistence and privilege escalation.

V. Conclusions

This paper presented a use case with generators in a two-area power system with a balancing authority. The 9-bus test case is the exemplar physical system [22]. Without loss of generality, it can be expanded to larger power systems. Future work involves expanding the use case to larger systems.

One takeaway from this scenario is that its implementation is non-trivial, due to the complexities of real-life cyber-physical systems. Setting it up for high-fidelity emulation took longer than a simple simulation (that would abstract important details and analysis into models that are not realistic).

During Stage 2 - privilege escalation - the data fusion engine detected the attack with high accuracy using cyber-only data. This shows the threat can be detected and corrected early, before it impacts the physical system. Future work includes the implementation of Stage 4. If early-stage mitigation fails, in Stage 4 there is physical impact, and we consider the response of generator redispatch, of automatic actions in remedial or special protection systems, and their impact on physical operational reliability and transient stability. Stage 4 also involves coordinated response from a power system's operators and the utility's security team. A detailed analysis of the response of the cyber-physical controllers in the models and verified in the emulation is important as a next step alongside the design of the data fusion engine and next-generation cyber-physical energy management systems.

References

[1] A. Sahu, K. Davis, H. Huang, A. Umunnakwe, S. Zonouz, and A. Goulart, "Design of next-generation cyber-physical energy management systems: Monitoring to mitigation," *IEEE Open Access Journal of Power and Energy*, vol. 10, pp. 151–163, 2023.

[2] K. Davis, "An energy management system approach for power system cyber-physical resilience," in *invited position paper for Virtual Workshop on Cyber Experimentation and Science of Security (CESoS)*, 2021.

[3] M. Z. Gunduz and R. Das, "Cyber-security on smart grid: Threats and potential solutions," *Computer Networks*, vol. 169, p. 107094, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128619311235

[4] MITRE, "MITRE ATT&CK: ICS Matrix," May 2022. [Online]. Available: https://attack.mitre.org/matrices/ics/

[5] A. Sahu, P. Wlazlo, Z. Mao, H. Huang, A. Goulart, K. Davis, and S. Zonouz, "Design and evaluation of a cyber-physical testbed for improving attack resilience of power systems," *IET Cyber-Physical Systems Theory & Applications*, https://doi.org/10.1049/cps2.12018, June 2021.

[6] A. Sahu, Z. Mao, P. Wlazlo, H. Huang, K. Davis, A. Goulart, and S. Zonouz, "Multi-source multi-domain data fusion for cyberattack detection in power systems," *IEEE Access*, vol. 9, pp. 119 118–119 138, 2021.

[7] S. S. H. Toroghi and V. M. Thomas, "A framework for the resilience analysis of electric infrastructure systems including temporary generation systems," *Reliability Engineering & System Safety*, p. 107013, 2020.

[8] D. Becker, "ICCP User Guide: TR-107176," Electric Power Research Institute (EPRI), Tech. Rep., December 1996.

[9] EPRI, "Inter-Control Center Communications Protocol (ICCP, TASE.2): Threats to Data Security and Potential Solutions," Electric Power Research Institute (EPRI), Tech. Rep., 2001.

[10] P. Ilgner, P. Cika, and M. Stusek, "SCADA-based message generator for multi-vendor smart grids: Distributed integration and verification of tase. 2," *Sensors*, vol. 21, no. 20, p. 6793, 2021.

[11] M. Franz, "ICCP exposed: Assessing the attack surface of the utility stack," April 2020. [Online]. Available: https://s4xevents.com/wp-content/uploads/2020/04/3franz.pdf

[12] A. Volkova, M. Niedermeier, R. Basmadjian, and H. deMeer, "Security challenges in control network protocols: A survey," *IEEE Communication Surveys & Tutorials*, vol. 21, no. 1, pp. 619–639, 2019.

[13] M. J. Rice, C. A. Bonebrake, G. K. Dayley, and L. J. Becker, "Secure ICCP final report," Pacific Northwest National Lab (PNNL), Richland, WA (United States), Tech. Rep., 2017.

[14] National Institute of Standards and Technology (NIST) - Information Technology Laboratory, "CVE-2022-2277 detail," 2022. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2022-2277

[15] J. Sheng, "Research on SQL injection attack and defense technology of power dispatching data network: Based on data mining," *Mobile Information Systems*, vol. 2022, no. 6207275, 2022.

[16] E. Kovacs. (2017) Injection attacks common in energy and utilities sector: IBM. [Online]. Available: https://www.securityweek.com/injection-attacks-common-energy-and-utilities-sector-ibm/

[17] S. Sridhar and G. Manimaran, "Data integrity attacks and their impacts on scada control system," in *IEEE Power and Energy Society General Meeting*, 2010, pp. 1–6.

[18] M. Hassan, N. Roy, and M. Sahabuddin, "Mitigation of frequency disturbance in power systems during cyber-attack," in *2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*. IEEE, 2016, pp. 1–4.

[19] S. Sridhar and M. Govindarasu, "Model-based attack detection and mitigation for automatic generation control," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 580–591, 2014.

[20] A. Sahu, P. Wlazlo, Z. Mao, H. Huang, A. Goulart, K. Davis, and S. Zonouz, "Design and evaluation of a cyber-physical testbed for improving attack resilience of power systems," *IET Cyber-Physical Systems: Theory & Applications*, 2021.

[21] L. Al Homoud, N. Barpanda, A. Goulart, K. Davis, and M. Rice, "Dataset of a cyber-physical detection tool evaluated in a multi-stage attack scenario," IEEE Dataport, 2023. [Online]. Available: https://dx.doi.org/10.21227/21zx-9a54

[22] WSCC 9-bus system. Illinois Center for a Smarter Electric Grid (ICSEG). [Online]. Available: https://icseg.iti.illinois.edu/wscc-9-bus-system/

[23] "Available Generation Control Modeling," 2022. [Online]. Available: https://www.powerworld.com/WebHelp/Content/MainDocumentation_HTML/Available_Generation_Control_Modeling.htm

[24] C. Dougherty, "Practical identification of SQL injection vulnerabilities," 2013. [Online]. Available: https://www.cisa.gov/uscert/sites/default/files/publications/Practical-SQLi-Identification.pdf

[25] A. Takiddin, M. Ismail, R. Atat, K. R. Davis, and E. Serpedin, "Robust graph autoencoder-based detection of false data injection attacks against data poisoning in smart grids," *IEEE Transactions on Artificial Intelligence*, pp. 1–15, 2023.