RICSel21 Data Collection: Attacks in a Virtual Power Network

Chih-Yuan Lin[†], August Fundin[†], Erik Westring[*], Tommy Gustafsson[*], and Simin Nadjm-Tehrani[†],
[*]FOI, Swedish Defense Research Agency, Sweden
[†]Linköping University, Sweden

***Abstract*—Attacks against Supervisory Control and Data Acquisition (SCADA) systems operating critical infrastructures have increased since the appearance of Stuxnet. To defend critical infrastructures, security researchers need realistic datasets to evaluate and benchmark their defense mechanisms such as Anomaly Detection Systems (ADS). However, real-world data collected from critical infrastructures are too sensitive to share openly. Therefore, testbed datasets have become a viable option to balance the requirement of openness and realism. This study provides a data generation framework based on a virtual testbed with a commercial SCADA system and presents an openly available dataset called RICSel21, with packets in IEC-60870-5-104 protocol streams. The dataset is the result of performing 12 attacks, identifying the impact of attacks on a power management system and recording the logs of the seven successful attacks.**

I. Introduction

Supervisory Control and Data Acquisition (SCADA) systems, which control and monitor critical infrastructure, have become attractive targets for hackers [1], [2]. To protect these systems, and due to the special characteristics of SCADA networks such as traffic regularity, Anomaly Detection Systems (ADS) are of great interest to researchers for SCADA security.

Evaluating and validating the detection capabilities of ADS requires quality datasets. Whereas generating regular Internet traffic, such as campus traffic and backbone network traffic for ADS evaluation has been widely studied [3], the tools or data proposed for Internet security can hardly be used for SCADA security research. The biggest difference between regular Internet networks and SCADA networks is the existence of industrial control devices and processes. The tools designed for Internet security lack capabilities to simulate the impact of attacks on industrial control devices and processes. Additionally, the sensitivity of critical infrastructure environments, the use of specialized protocols, and attack types make the SCADA data generation a unique problem.

Getting access to realistic attack datasets is difficult for researchers in the SCADA security field. Real SCADA systems of critical infrastructure can't be used for experiments. Most researchers use datasets collected in simulated environments or real utilities with self-injected attacks [4], [5]. The problem with this approach is that the attack traffic cannot reflect the impact on the processes. Some researchers use datasets with attacks generated in testbeds [6], [7]. However, previous research [8] has shown that traffic from testbeds may be overly repetitive due to the use of a simulation program, which repeats deterministic workflows, and lack of interaction with human operators.

To generate realistic SCADA traffic with attacks, this work implements an attack-bot in the national Swedish testbed RICS-el [9], a testbed that is made using virtual machines and a simulated electricity production and distribution system. The use of virtual machines and a simulated production environment enables safe and repeatable experiments for data generation and attack observations. Also, the system provides two bots that can be utilized to automate experiments in the testbed. The OT-bot can be programmed to mimic some activities of a human operator, running the power grid from the control room and the scenario-bot controls the testbed and collects the data generated during the experiment.

The contributions of this study are (1) design and implementation of an attack-bot that is integrated into the RICS-el testbed, (2) use of the attack-bot to generate cyber attacks in the testbed and collection of the RICSel21 dataset for further studies, and (3) observations of the impact of inserted attacks on the SCADA systems in the RICS-el testbed. The observations can be used as an aid in the process of assessing the risk posed by each attack.

II. Background and Related Work

This section presents the RICS-el testbed and IEC 60870-5-104 (IEC-104) protocol. The section also presents other data generation approaches as related work.

A. *RICS-el*

RICS-el [9] is a virtual testbed consisting of an office network, a control network, and a power grid simulator representing low-level field devices. The control network is referred to as Operation Technology (OT) Local Area Network (LAN) and its setup contains some twenty substations, two control servers, a virtual Wide Area Network (WAN) with 15 nodes interconnecting the control room, the substations, and the emulated Remote Terminal Units (RTU) controlling the power grid. The OT LAN and RTUs connect through pseudo communications with IEC-104.

On top of the emulated system, the OT-bot and the scenario-bot are used to create and execute more realistic scenarios and for data collection. The scenario-bot, which connects to both the control network and power grid, allows users to execute pre-programmed events such as resetting the environment, running scripts, setting up the recording of network traffic, and downloading experiment data as pcap and log files. The OT-bot, which resides in the control network, can be configured to mimic an operator that issues commands from the Human Machine Interface (HMI) to the RTUs. More details about the control flow will be illustrated in Section III-A.

There are three major differences between RICS-el and a real power network. First, though the RTUs and control servers generate traffic and communicate about the status of the system with each other, the power grid simulator does not rely on such traffic. The power grid and control servers update their status via two internal databases. Second, each node in the testbed is virtualized and is not physically distanced. Finally, the simulated power grid values are sampled every five seconds, which may or may not apply in a real configuration depending on how operators set up their SCADA system.

B. IEC-60870-5-104

IEC-60870-5-104 is a communication protocol widely used in SCADA systems, especially within Europe. The protocol is defined in an international standard and this section briefly introduces the concepts and terminology of the protocol mentioned in this paper.

In the application layer, the basic frame format is called Application Protocol Data Unit (APDU). An APDU always contains an Application Protocol Control Information (APCI) block, which contains basic information such as the length of the packet and sequence number. Optionally, an APDU may contain an Application Service Data Unit (ASDU) block for more detailed information. An APDU can be in U, S, or I format. The U format is used to start (STARTDT) or stop (STOPDT) the data transfer. The S format controls the transport of APDUs. The I format containing an ASDU carries the SCADA instructions. Instructions with the prefix M are for measurements from monitored sensors, and the prefix C stands for commands. Table I list the SCADA instructions mentioned in this paper.

TABLE I
INSTRUCTIONS USED IN THIS STUDY

Instruction	Description
M_SP_NA_1	Single point information
M_ME_NA_1	Normalized measured value
M_SP_TB_1	Single point information with time tag
M_DP_TB_1	Double point information with time tag
M_ME_TF_1	Measured short floating point value with time tag
C_DC_NA_1	Double command
C_DC_TA_1	Double command with time tag
C_SE_NC_1	Set-point command, short floating point value

C. Data Generation Approaches

There are two common approaches to generate SCADA datasets for security research without access to real SCADA systems: the testbed approach and the synthetic data approach. The testbed approach builds a testbed mimicking a real-world SCADA system, while the synthetic data approaches create datasets in accordance with a set of rules found from previous captures.

Conti et al. [10] in a survey on SCADA testbeds and datasets categorized testbeds into physical, hybrid, and virtual testbeds. Of the presented testbeds, the Maynard SCADA [11] was the most relevant since it is an open-source, scalable framework for deploying a virtual testbed. The framework supports multiple protocols including IEC-104. One dataset was documented with the setup of 5 RTUs, 1 HMI, and 1 historian. RICSel21 exploits a library provided by the Maynard SCADA framework. Except for the size of the testbed, the main difference between the Maynard testbed and RICSel21 system is the realism of the RICSel21 system due to the deployment of OT-bot and the scenario-bot.

While synthetic data generation is an active research area for regular IT systems, synthetic data generation research for SCADA systems is still in its infancy. Most of the studies in this strain are based on the packet crafting tool Scapy. Kobayashi et al. [12] extended Scapy for the Modbus protocol. Al-Dalky et al. [13] proposed a malicious traffic generator according to the rules of an intrusion detection system. Lopes et al. [14] proposed a traffic generator in IEC 61850 (GOOSE) networks. In addition to these Scapy-based approaches, Ndonda et al. [15] proposed a SCADA traffic generator based on statistical properties of empirical traces. Compared to testbed approaches, these generation methods may not be able to reflect the impact of the attack on the process.

While other approaches, such as privacy-preserving machine learning [16] and network traffic anonymization [17], can also contribute to solving the problem of SCADA data for ADS evaluation, the tradeoff between privacy and utility is a challenge that needs to be addressed. This paper focuses on testbed SCADA data generation in presence of attacks, where information about the testbed is shareable.

III. Data Generation Process

This section outlines how data can be generated and collected from RICS-el regarding the system architecture and data flow. Details, such as the full list of functions and used commands, can be found in the thesis by Fundin [18].

A. System Architecture

Figure 1 illustrates the data generation system including part of the RICS-el testbed (OT LAN and power grid) and the attack-bot created to define and execute the attacks. The orange arrows show the normal IEC-104 traffic flow. The red arrows represent the re-routed traffic between the HMI and the firewall, passing through the attack-bot. The yellow arrows show the control flow from the scenario-bot, which establishes a Secure Shell (SSH) connection to different nodes where actions will be executed and sends a command to each node. Finally, the dark blue arrows illustrate the non-IEC-104 internal communication as presented in Section II-A.

The design of the attack-bot is a major contribution of this study. The attack-bot runs on an Ubuntu 16.04 machine located in the OT LAN. The Man in the Middle (MitM) library developed in Maynard SCADA [11] is installed to enable

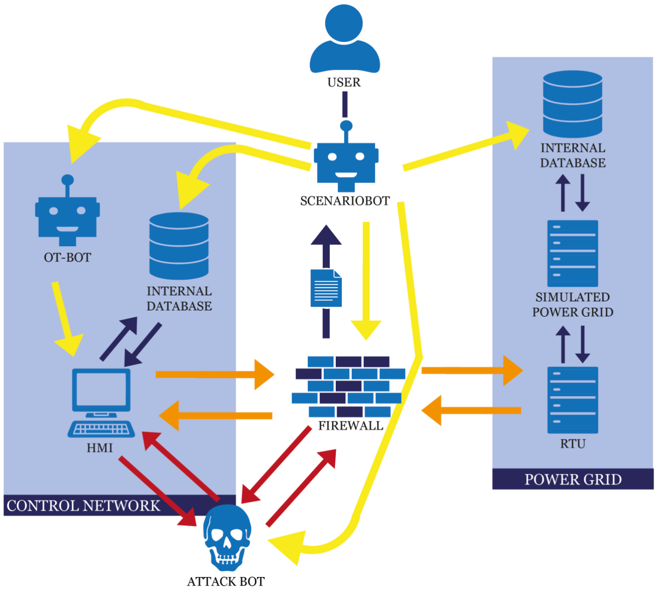

Fig. 1. A section of the RICS-el testbed depicting the attack-bot and the major data flow.

re-routing of traffic by Address Resolution Protocol (ARP) spoofing. The attacks are written in C and the attack-bot is written with BASH and Python3. It provides a command-line interface that the scenario-bot can use to initiate attacks. Via the command-line interface, it is possible to control the attack-bot by using arguments, where for instance **-a** specifies which attack to run and **-t** specifies for how long the attack should run. The complete set of parameters of the command-line interface can be found in [18].

B. Data Generation Workflow

Figure 2 presents the data generation cycle. Before starting the data generation cycle, all scenarios are put in a queue for processing. For each scenario in the queue, there are six steps to complete the data generation process as follows.

Script preparation. The user first needs to prepare an attack script and enter it into the attack-bot. Then, the user adds a command to initiate the attacks in the scenario-bot.

System configuration. The step resets and stabilizes the system. The scenario-bot sends a reset command and reverts the system to normal operation by (1) shutting down the attack-bot, (2) setting internal databases with predefined values, (3) unification and synchronization of internal databases, (4) enabling connecting requests from RTUs and HMI, and (5) resetting communications between RTUs and HMI. This step temporarily stops the IEC-104 traffic for about 1 minute, which puts the system into a stable state and makes it ready when the traffic resumes.

Running an attack scenario. The scenario-bot initiates the following actions after the system is put in a stable mode. (1) Start recording. (2) Tell the OT-bot to load and execute a set of commands. (3) Tell the attack-bot to start an attack. (4) Halt the attack. (5) Stop recording and download the pcap file.

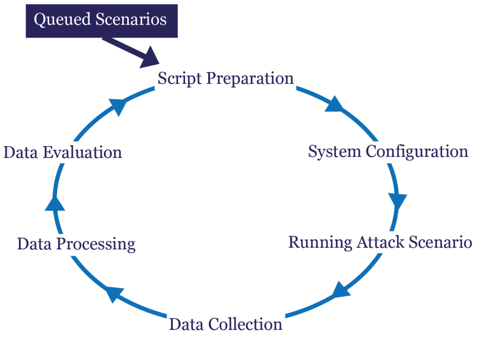

Fig. 2. The data generation workflow.

Data collection. All network traffic is recorded at the firewall between the RTUs and HMI using Tcpdump[1]. Tcpdump records packets in their raw format with all headers and layers stored in a pcap file on the hard drive of the machine running the scenario-bot. At the same time, the scenario-bot downloads the OT-bot configurations and its actions as well as process values in 5-second resolution when the testbed is active. Then, it logs every action, together with timestamps, in a CSV file. This log can be used to label the generated datasets.

Data processing. The data processing includes the following steps:

1) The total number of network packets in the pcap file is counted using Wireshark.
2) The number of IEC-104 packets in the pcap file seen as legitimate by Wireshark is counted.
3) Attack-bot logs are analyzed to check if there are any run-time errors.
4) The attack-bot logs are also parsed for attack-specific data, such as the number of packets recorded in a replay attack.
5) The overviews in the HMI and the simulated results of power grid are compared to check for attack impacts or distortions.
6) Network traffic characteristics are examined.
7) The user manually checks whether dataset labels are present and accurate, especially attack intervals and attack type.

Iterative dataset evaluation. Each generated dataset is evaluated based on the outcome of the data processing and predefined goals. For example, the user needs to evaluate if there is a substantial increase in network load when executing flooding attacks. If the evaluation results show that the attack needs to be altered, the user needs to prepare new scripts for this scenario and re-run the whole cycle. Otherwise, the next scenario will be executed.

IV. Attack Scenarios

The work contains twelve different attack scenarios that are representative of attacks found in the literature. In addition to

[1] A packet analyzer: https://www.tcpdump.org/

the scanning activities, seven of the attack scenarios successfully generated attack traffic. These attacks target the IEC-104 traffic between the HMI and an RTU as listed in Table II. In the base scenario, the attacker has gained access to a computer connected to the control plane, illustrated by the attack-bot in Figure 1.

This section briefly introduces the attack scenarios and their results. The detailed setup and issued commands can be found in the thesis by Fundin [18].

TABLE II
OVERVIEW OF IMPLEMENTED ATTACK SCENARIOS

#	**Attack description**	**Type**
1	Scanning and ARP spoofing of identified targets	Scanning
2	Dropping all IEC-104 packets between two targets	DoS
3	SYN flooding of an RTU	DoS
4	Network traffic flooding, targeting the HMI and RTU.	DoS
5	Intermittent dropping of IEC-104 packets	Sequence
6	Altering sequence numbers of IEC-104 packets	MitM
7	Altering sensor values in IEC-104 packets	MitM
8	Altering time tags of IEC-104 packets	MitM
9	Altering IOAs of IEC-104 packets	MitM
10	Replay of IEC-104 packets in control station	Reply
11	Injection of setpoint commands	Injection
12	Masquerade as a second HMI	Injection

A. Scenario 1 - Establishing foothold

In the first scenario, the attacker reconnoitered the network by investigating the ARP table and by scanning the environment with Nmap[2]. Then the attacker connected to one of the found hosts via SSH [3] and identified IEC-104 communications with Tcpdump. Last, Ettercap[4] was used to intercept the traffic between the firewall and the RTU, establishing the attacker as a MitM. This capability was then used in the following scenarios.

B. Scenario 2 - IEC-104 packet drop

In this scenario, the attacker dropped all IEC-104 packets between the HMI and the RTU. This attack effectively cut the IEC-104 communication between the HMI and the RTU.

C. Scenario 3 - SYN Flood

In scenario three, the attack-bot was used to create a SYN flood attack, targeting all ports ranging from 0 to 4096 on the RTU. This setup includes the port 2404 used by IEC-104, but since the RTU was configured to only accept traffic from the HMI on this port, the attempt to disrupt the traffic was unsuccessful.

D. Scenario 4 - IEC-104 Flood

The IEC-104 flood was created by first recording 20 seconds of the 104-traffic. The recorded packets were then replayed to the source as fast as the attack-bot would manage, resulting in an average of 11783 packets sent per second, which disrupted the real IEC-104 traffic. Even though some measurements got through to the HMI, the attack would probably have been perceived as a communication outage by an operator. When the attack lasted for more than 20 minutes, the HMI also stopped trying to maintain the communication with the RTU, making this a successful attack.

E. Scenario 5 - IEC-104 Sequence attack

In this scenario, an attempt was made to create a sequence attack using ARP spoofing from the attack-bot to intercept IEC-104 packets and then dropping them. But, as the sender never received any TCP acknowledgment, it continued to send the packet which eventually got through to the intended receiver. Therefore the attack was unsuccessful.

F. Scenario 6 - Altered payload (Sequence numbers)

In this attack, IEC-104 S-format packets sent to the RTU were captured by the attack-bot and had their sequence numbers in the APCI lowered before being sent through. The offset of the sequence numbers started at one, but was gradually incremented for every 16 APCI captured. The goal of this attack was to get the RTU to start resending old packets by denying it to clear its buffers, thereby creating an altered payload. At first, there was no reaction from the RTU. But once the offset was larger than three it began to close the connection, effectively creating a denial of service condition and thereby making this attack successful.

G. Scenario 7 - Altered payload (Sensor values)

This scenario targeted the sensor values being sent from the RTU to the HMI. Once captured by the attack-bot, the I-format IEC-104 packets were analyzed and had their payload altered to decimal zero if they were of the types M_ME_NA_1, M_SP_NA_1, or M_DP_TB_1. The altered NA_1 packets were interpreted as the voltage in the power grid having been lowered from 420 volts to 250 volts. The altered TB_1 packets always set the breaker switch to zero, stopping the operator from knowing which states the breaker switches had. These changes successfully resulted in the operator having an incorrect view of the state of the power grid from the HMI.

H. Scenario 8 - Altered payload (Time tag)

Currently, RICS-el doesn't include any in-line Network Time Protocol (NTP) to supply a shared system time since the virtual hosts get the time from their physical node. Therefore, an attack on the timing in the power grid was attempted by altering the time tag in captured IEC-104 packets. In a first simulation, the time tag was increased by two hours, mimicking the clock in the RTU being desynchronized. In a second simulation, the time tag is decreased by ten seconds to make it appear as if the packet had arrived late. None of the altered time tags appeared to have an effect on RICS-el making these attacks unsuccessful.

[2] The Network Mapper: https://nmap.org/

[3] Secure SHell: https://www.unixtutorial.org/reference/ssh

[4] A suite for MitM attacks: https://www.ettercap-project.org/

I. Scenario 9 - Altered payload (Breaker IOAs)

In this scenario, the attacker used the attack-bot to switch the destination of the commands sent by an operator to affect another breaker than intended. To mask the attack, the subsequent measurement was also changed, making the operator unaware of the actual state of the breakers. The attack was achieved by first monitoring the IEC-104 communication to identify suitable breaker pairs. These were then swapped once an operator issued a command, bringing the system into an unsafe state. This attack was successful.

J. Scenario 10 - Replay attack

The goal of this scenario was to have the HMI accept old data sent by the RTU, effectively making the operator blind to the current state in the power grid. The scenario was initiated by the attacker recording 20 seconds of the IEC-104 packets sent from the RTU to the HMI. For the packets of the type M_ME_NA_1, M_SP_NA_1, M_DP_TB_1, M_SP_TB_1, and M_MF_TF_1, the ASDU-data was stored. Once the recording was complete, the ASDU-field in any subsequent IEC-104 packet of the mentioned types were replaced with the stored data, making the HMI display an inaccurate state of the power grid. This attack was successful.

K. Scenario 11 - Packet injection between the RTU and HMI

The adversary's objective was to masquerade as the HMI and send false setpoint commands to a generator controlled by the RTU. This was done by using the attack-bot to gain access to the established TCP stream between the HMI and the RTU. With packet inspection, a setpoint command of the type C_SE_NC_1 was detected as well as the corresponding IOA of a generator. The adversary created a replica of the command's APDU and changed the setpoint value of zero. A TCP header was then created with values from the most recent transmission. The fabricated C_SE_NC_1 payload was added to the TCP header and sent to the RTU. Until the RTU confirmed the new setpoint no packets were allowed to reach the HMI. This attack was successful.

L. Scenario 12 - Packet injection to the RTU

The adversary's objective was to gain access to and get privileges for the RTU. The method to do that was to first identify an RTU by sending broadcasting interrogation commands and then extract the used port and IP address from the broadcast answers. A STARTDT-frame was then sent to the RTU to initiate IEC-104 communication. After the RTU had acknowledged the STARTDT, packet types of C_DC_TA_1 and C_SE_NC_1 were sent to various IOAs until an affirmative response was received. That response reveals a valid IOA. This was repeated with the purpose to detect all actuators in the substation. This attack was unsuccessful since the RTU had a white list that only accepted packets from the HMI.

V. Results and evaluation

A. Attack scenario evaluation

As described in the iterative dataset evaluation part of Section III-B, the above attack scenarios were evaluated using different criteria depending on the attack goal. The summary of the success criteria for the performed attacks are summarised in Table III. The evaluation process showed that most attacks achieved their expected goals.

In a couple of cases the evaluation found room for improvement in the attack-bot, e.g. where the attack packets timing fields (TTL) had not been adjusted to reflect a consistent value compared to the baseline normal data. In other cases, e.g. attack scenario 8, it was found that the setup of the testbed itself prevented the RTU to execute commands that were seemingly delayed due to lack of time synchronisation protocols. Rectifying this would improve the authenticity of the testbed itself in future use cases.

It's worth mentioning that scenario 1 is partially successful because all the hosts in RICS-el are password-protected. The emulated attacker needed to ask the administrator for passwords to complete the SSH connections in step 2.

B. Generated datasets

Table IV summarizes the generated datasets [5] for the fully successful scenarios. In the Table, time is specified in mm:ss format, denoting time passed since RICS-el had stabilized. All the traces are 30-minute long and every scenario lasts the entire duration except for scenario 11, in which the attack lasts for three minutes.

The published datasets also include one baseline traffic (row 1 in Table IV) for the users to compare with the different attack traffic sequences and investigate the impact of the attacks. With a first look at the baseline and attack traffic, we observe significant differences, such as a sudden burst of traffic in a certain flow, or a full stop of traffic. This shows the proposed system can reflect the changes of the simulated process caused by attackers on the generated traffic in the current settings.

Note that the datasets from scenarios 7 and 10 may not contain malicious traffic since the traffic alteration inside the OT LAN could not be captured by the recording software at the firewall and the impact of attacks was mainly observed at the HMI.

VI. Conclusion and Future Work

This paper presents an attack generation framework based on RICS-el, which is a virtual testbed for a power grid network. The paper also presents twelve different attacks against RICS-el and their results. Eight out of the twelve attacks succeed and seven traffic streams have been collected as the RICSel21 dataset for further studies by other researchers. The framework can be used to study and analyze the impact of new attacks in power networks and the datasets can be used for ADS evaluations or dynamic risk assessment.

[5] https://gitlab.liu.se/ida-rtslab/public-code/2021_attack_rics

TABLE III
ATTACK SUCCESS CRITERIA AND OUTCOMES

Scenario #	Criteria of success	success?
1	If a machine running IEC-104 was found.	Yes (partially)
2	If the connection between HMI and RTU was lost from the operator's perspective.	Yes
3	If the connection between HMI and RTU was lost or if latency increased by 100%.	No
4	If the connection between HMI and RTU was lost or if latency increased by 100%.	Yes
5	If a command was ignored or executed in an unintended way.	No
6	If the RTU stopped all of its transmissions to the HMI, started to repeat itself or lost data.	Yes
7	If the observed values in the HMI differed from the actual values in the power grid.	Yes
8	If the commands from the HMI were ignored by the RTU.	No
9	If the operator saw a breaker change in the HMI, but another breaker than the intended got changed in the power grid.	Yes
10	If the HMI stopped receiving new values from the RTU and instead accepted the replayed packets.	Yes
11	If the setpoint of a generator was altered without the involvement of an operator.	Yes
12	If the setpoint of a generator was altered without the involvement of an operator.	No

TABLE IV
RECORDED DATASETS FOR SUCCESSFUL ATTACKS

#	Type	# Pkt. (IEC-104)	Comments
	Baseline	7144 (1571)	No attacks present
2	DoS	4366 (603)	Attack briefly stopped 22:17-23:05
4	DoS	16488794 (1121)	1121 legitimate IEC-104 packets
6	MitM	2955 (520)	Communication line service crashed
7	MitM	6850 (1519)	-
9	MitM	6991 (1540)	-
10	Replay	6967 (1504)	14 packets recorded
11	Injection	6718 (1446)	Cover-up successful for ten seconds

The proposed system contains an OT-bot that operates a selected set of commands to enhance the realism of the testbed traffic. One obvious future work is to study how operators act in different control networks through analysis of real operation traffic or interviews so that the system generates datasets in a wider context. The proposed systems can also be used to study more advanced attack types and defense mechanisms such as advanced persistent attacks and defense in depth.

ACKNOWLEDGMENTS

This work was supported by the Swedish Civil Contingencies Agency (MSB) within the project Resilient Control and Information Systems (RICS), in its extended second phase (www.rics.se).

REFERENCES

[1] N. Falliere, L. O. Murchu, and E. Chien, "W32.Stuxnet dossier," Symantec, Mountain View, Tech. Rep., 2011.

[2] R. M. Lee, M. J. Assante, and T. Conway, "Analysis of the cyber attack on the ukrainian power grid: Defense use case," Electricity Information Sharing and Analysis Center (E-ISAC), Tech. Rep., 2016.

[3] C. G. Cordero, E. Vasilomanolakis, A. Wainakh, M. Mühlhäuser, and S. Nadjm-Tehrani, "On generating network traffic datasets with synthetic attacks for intrusion detection," *ACM Trans. Priv. Secur.*, vol. 24, no. 2, Jan. 2021. [Online]. Available: https://doi.org/10.1145/3424155

[4] C. Wressnegger, A. Kellner, and K. Rieck, "Zoe: Content-based anomaly detection for industrial control systems," in *Proceedings of 48th Annual International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2018.

[5] C.-Y. Lin and S. Nadjm-Tehrani, "Timing patterns and correlations in spontaneous SCADA traffic for anomaly detection," in *Proceedings of 22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*. USENIX Association, 2019.

[6] S. Adepu, N. K. Kandasamy, J. Zhou, and A. Mathur, "Attacks on smart grid: Power supply interruption and malicious power generation," in *International Journal of Information Security volume 19*, 2020, p. 189–211.

[7] M. A. Teixeira, T. Salman, M. Zolanvari, R. Jain, N. Meskin, and M. Samaka, "SCADA system testbed for cybersecurity research using machine learning approach," *Future Internet*, vol. 10, no. 8, 2018. [Online]. Available: https://www.mdpi.com/1999-5903/10/8/76

[8] C.-Y. Lin and S. Nadjm-Tehrani, "A comparative analysis of emulated and real IEC-104 spontaneous traffic in power system networks," in *Proceedings of Cyber-Physical Security for Critical Infrastructures Protection (CPS4CIP)*. Springer, 2020.

[9] M. Almgren, P. Andersson, G. Björkman, M. Ekstedt, J. Hallberg, S. Nadjm-Tehrani, and E. Westring, "RICS-el : Building a National Testbed for Research and Training on SCADA Security (Short Paper)," in *Critical Information Infrastructures Security (CRITIS)*. Cham: Springer International Publishing, 2019, pp. 219–225.

[10] M. Conti, D. Donadel, and F. Turrin, "A survey on industrial control system testbeds and datasets for security research," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2021.

[11] P. Maynard, K. McLaughlin, and S. Sezer, "An Open Framework for Deploying Experimental SCADA Testbed Networks," in *5th International Symposium for ICS & SCADA Cyber Security Research 2018*. Science Open, 2018, pp. 89–98.

[12] T. H. Kobayashi, A. B. Batista, A. M. Brito, and P. S. Motta Pires, "Using a packet manipulation tool for security analysis of industrial network protocols," in *2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007)*, 2007, pp. 744–747.

[13] R. Al-Dalky, O. Abduljaleel, K. Salah, H. Otrok, and M. Al-Qutayri, "A modbus traffic generator for evaluating the security of scada systems," in *2014 9th International Symposium on Communication Systems, Networks Digital Sign (CSNDSP)*, 2014, pp. 809–814.

[14] Y. Lopes, D. C. Muchaluat-Saade, N. C. Fernandes, and M. Z. Fortes, "Geese: A traffic generator for performance and security evaluation of iec 61850 networks," in *2015 IEEE 24th International Symposium on Industrial Electronics (ISIE)*, 2015, pp. 687–692.

[15] G. K. Ndonda and R. Sadre, "Network trace generation for flow-based IDS evaluation in control and automation systems," *IJCIP*, vol. 31, 2020. [Online]. Available: https://doi.org/10.1016/j.ijcip.2020.100385

[16] L. K. L. Ng, S. S. M. Chow, A. P. Y. Woo, D. P. H. Wong, and Y. Zhao, "Goten: Gpu-outsourcing trusted execution of neural network training," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 17, pp. 14 876–14 883, May 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17746

[17] N. V. Dijkhuizen and J. V. D. Ham, "A survey of network traffic anonymisation techniques and implementations," *ACM Computing Surveys*, vol. 51, p. pp 1–27, 2018.

[18] A. Fundin, "Generating datasets through the introduction of an attack agent in a SCADA testbed: A methodology of creating datasets for intrusion detection research in a SCADA system using IEC-60870-5-104," Master's thesis, Linköping University, 2021.