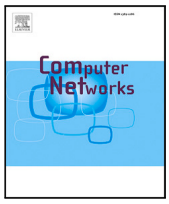
Computer Networks 255 (2024) 110804

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Data article

Malware communication in smart factories: A network traffic data set

Bernhard Brenner [a,*], Joachim Fabini [a], Magnus Offermanns [b], Sabrina Semper [b], Tanja Zseby [a]

[a] TU Wien, Gusshausstrasse 25/E389, Vienna, 1040, Austria
[b] TÜV AUSTRIA, Tüv-Austria-Platz 1, Brunn am Gebirge, 2345, Lower Austria, Austria

ARTICLE INFO

Keywords:
Operational technology
OT
Network traffic
Data set
Industrial control systems
ICS
Intrusion detection system
IDS
Network security
Internet of Things
IoT
IIOT

ABSTRACT

Machine learning-based intrusion detection requires suitable and realistic data sets for training and testing. However, data sets that originate from real networks are rare. Network data is considered privacy sensitive and the purposeful introduction of malicious traffic is usually not possible. In this paper we introduce a labeled data set captured at a smart factory located in Vienna, Austria during normal operation and during penetration tests with different attack types. The data set consists of 173 GB of Packet Capture (PCAP) files, which represent 16 days (395 h) of factory operation. It includes Message Queuing Telemetry Transport (MQTT), OPC Unified Architecture (OPC UA), and Modbus/TCP traffic. The captured malicious traffic was originated by a professional penetration tester who performed two types of attacks: (a) aggressive attacks that are easier to detect and (b) stealthy attacks that are harder to detect. Our data set includes the raw PCAP files and extracted flow data. Labels for packets and flows indicate whether packets (or flows) originated from a specific attack or from benign communication. We describe the methodology for creating the data set, conduct an analysis of the data and provide detailed information about the recorded traffic itself. The data set is freely available to support reproducible research and the comparability of results in the area of intrusion detection in industrial networks.

1. Introduction

The prevailing use of communication technology in Industrial Control Systems (ICSs) creates a new demand for the deployment of sophisticated Intrusion Detection Systems (IDSs). It therefore increases the demand for appropriate data sets from industrial environments to evaluate IDSs. Suitable data sets need to include representative network traffic with machine-to-machine communication, modern attack patterns and, ideally, labeled instances. Machine learning-based classifiers need data with typical patterns for training and testing. IDS detection performance profits if representative data recorded from productive operations in real-world networks can be used.

However, representative and recent public network data sets are rare already for classical Information Technology (IT) networks. The situation is even worse for ICSs, (which are Operational Technology (OT) networks), where only few data sets exist as of today.

Most of those rare OT data sets originate from simplified testbeds, are artificially generated in simulations (see Section 2), or do not provide labels. Data privacy concerns are a primary reason for not publishing such data. Data captured in a real network can contain sensitive information about the network or the operating company. Removing this information (known as data set anonymization) is challenging and error-prone.

In addition, most data sets do not contain attack traffic. This is especially true for datasets captured in real networks since it is usually problematic to introduce attacks in production environments.

In summary, we observed a significant gap between the demand for realistic ICSs network traffic by the scientific community and existing datasets. This work aims to reduce this gap by publishing a representative labeled data set from a real production environment in a pilot factory. It contains 173 GB of raw PCAP OT network traffic captured over two weeks and four days in the pilot factory.

This factory is owned by TU Wien. It is unique because, in addition to being experimental, it produces individual items or small series of custom parts in response to client demands. Therefore, the factory is only temporarily operational, which can be observed in the benign data set "TF A" (cf. Fig. 7). However, the benefit of this non-continuous operation is that the variety of events observed in the data set is more heterogeneous than if it were consistently producing the same workpieces.

The contributions of this work include the following:

* Corresponding author.
E-mail addresses: Bernhard.Brenner@tuwien.ac.at (B. Brenner), Joachim.Fabini@tuwien.ac.at (J. Fabini), Magnus.Offermanns@tuv.at (M. Offermanns), Sabrina.Semper@tuv.at (S. Semper), Tanja.Zseby@tuwien.ac.at (T. Zseby).

https://doi.org/10.1016/j.comnet.2024.110804
Received 4 September 2024; Accepted 11 September 2024
Available online 9 October 2024

- Collection of a network data set containing two weeks of ICS communication during normal operation in a smart pilot factory and four days of attack traffic.
- Introducing a set of attacks by a professional penetration tester across two different scenarios with multiple attack types.
- Losslessly recording both operational traffic and attacks using buffered capture hardware.
- Extracting flow and packet information.
- Labeling the extracted packets and flows.
- Analyzing the data set to support future users.

We make the original data set (PCAP) as well as aggregated information available to the research community.[1]

In addition, we provide the following analyses for the data set:

- Distribution of observed protocols during normal operation and for each attack day (cf. Section 7.1)
- Time series of observed packets and bytes per minute during the two weeks of normal factory operation (cf. Section 7.1.1)
- Time series of observed packets per minute for the days with attacks (cf. Section 7.1.2)
- Distributions of flow durations for benign and attack traffic (cf. Section 7.1.3)

2. Related work

When evaluating available ICS data sets, it is essential to differentiate between the collection of data about the control process itself (process data) and the collection of ICS-related communication network data.

Process data, on the one hand, is the information exchanged at the application layer between ICSs like, e.g., values of sensors, time series, or control commands. This information is of relevance mainly for ICS state simulation or analysis. Network data, on the other hand, includes addressing information (link-layer-, network-layer-, and transport-layer headers) in addition to potentially transport-layer (TLS) encrypted application layer data. In our paper, we provide such a network data set that aims at supporting the development and configuration of network-based IDS or anomaly detection algorithms. Since we publish an ICS network data set, this section also focuses on network data sets.

Morris et al. published several OT data sets that are popular in the community [1–4]. The data originates from testbeds and artificial generation, not from real OT systems. Some of these data sets are network data sets and some are process data sets. The Electra Railway ICS data set [5] contains network data from a simulated railway substation network. The authors describe the simulated network as resembling a realistic railway substation containing seven network devices (of which four communicate via Modbus, and five communicate via the S7 protocol). The data set is split into two parts, containing S7 (1.7 GB) and Modbus (56 MB) traffic. Lemay et al. also created a data set in 2016. Their data set focuses on packet captures from Modbus/TCP-based communication. It includes benign and malicious traffic and also Comma Separated Values (CSV) files for the labels. The data set is created by a simulated testbed environment running on virtual machines. The data set is available for download [6].

Alatram et al. published an OT network data set in 2023. This data set also originates from a testbed consisting of Raspberry Pi microcomputers that continuously publish values obtained from Internet of Things (IOT) sensors to an MQTT broker. The paper itself focuses on the MQTT protocol for the use case of the IOT and contains attack traffic of Denial of Service (DoS) attacks [7]. Specific to this paper is the variety of sensors used.

Sarica and Angin [8] released a data set in 2020. Their data set consists of two parts with several million records and is focused on an IOT network. It is obtained from a testbed.

Our evaluation of related work yields that publicly available ICS data sets have been captured within simulation or testbed environments, not in actual OT networks. Beyond that, fewer data sets focusing on network data of ICSs are available than data sets focusing on IOT network data. Our paper is one of the few that are accompanied by a network data set created in a real factory, containing attack data.

Flood et al. analyze seven network datasets and highlight six key shortcomings to avoid: Poor data diversity, highly dependent features, unclear ground truth, traffic collapse, artificial diversity, and wrong labels [9]. These issues and their consequences are detailed in the paper, especially in Section 4. Recommendations for avoiding these mistakes and improving datasets are provided in Section 6, focusing on generalization, feature selection, and overall design.

Of the seven datasets analyzed, five were IT datasets, and two were IOT datasets, namely Ton IoT [10] and Bot-IoT [11], both recorded from testbed networks. Both IoT datasets suffered from unclear ground truth, highly dependent features, traffic collapse, artificial diversity, and poor data diversity.

Their work is a valuable contribution for us during the creation and especially documentation of the data set since this paper was published after we recorded the data in the factory.

3. Technical setup and methodology

This section details on the technical setup, the methodology of the data set collection, the attacks, and some descriptive statistics about the data set itself. The attack setup can be seen in Fig. 1. All tests are performed within the turning cell of the pilot factory located in Vienna, Austria in autumn 2023, with the turning machine as its most crucial production asset (cf. Fig. 2). The factory is equipped with comparatively recent equipment, such as the EMCO MAXXTURN 45 turning machine, Siemens SENTRON PAC power sensors, and a Siemens 840D SL PCU/NCU pair. Fig. 1 depicts the cell network.

The following systems (hosts, addressed by dedicated IP addresses in the network capture) are used during normal productive operation:

- A Sinumeric PCU (human interface device which is a host on its own and is built into the turning machine).
- A Sinumeric NCU (controller, contains a Programmable Logic Controller (PLC)) as part of the turning machine.
- The turning machine itself as an actuator, which is only connected to the PLC
- Three power sensors attached to the turning machine
- A firewall for this network segment
- An enterprise-grade network switch
- Other hosts such as an MQTT broker

In addition, we attached the following hosts as part of our experiments:

- Our IDS, attached to the network segment's switch. The IDS is a common-of-the-shelf PC equipped with an Endace Data Acquisition and Generation (DAG) hardware capture card as the main component. The host is used exclusively as a capture device, i.e., to capture and record the network data directly from the mirror port of the switch and store it in a lossless and persistent manner.
- Additional hosts (colored black in Fig. 1) were connected to the Local Area Network (LAN) as attackers and attack targets.

The technical administrators of the pilot factory prohibited attacks beyond reconnaissance on productive hosts. This is why we connected three additional hosts to the cell as targets: Three virtual machines running (intentionally) vulnerable Linux distributions on a Windows 10

[1] The dataset is published with the DOI 10.48436/vs6hv-1vs74 at researchdata.tuwien.ac.at.

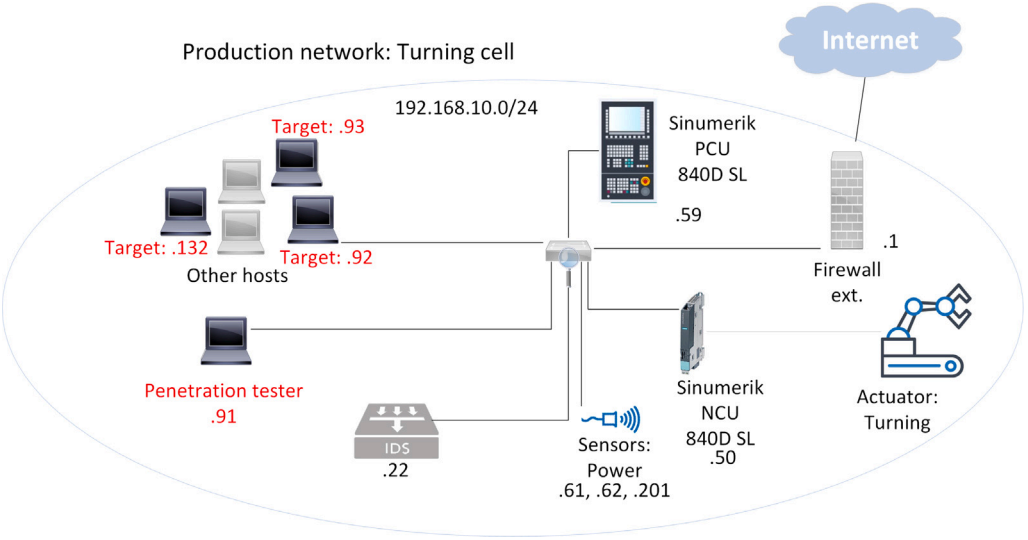


Fig. 1. Penetration test topology. (For interpretation of the references to color in all figures, the reader is referred to the web version of this article.)

Laptop with an Intel Core i7 6700HQ and 12 GB RAM were deployed in the factory as targets for all but the reconnaissance attacks. The three attacked hosts are running Metasploitable Linux 2 (IP .132), Metasploitable Linux 3 (IP .132), and a Kubuntu installation using Damn Vulnerable Web Application (IP: .93) [12–14]. All listed hosts are in operation during all days of the attack experiment, with one exception: on day one, the turning machine itself (including PCU and NCU) is turned off. In fact, this provides for a more realistic scenario, since systems in the pilot factory are typically only turned on when they are needed in order to save energy and wear.

3.1. Factory operation: Benign

During regular operation, the observable interaction between the machines includes the following: The MQTT broker is in operation. The turning machine is sometimes, but not always, turned on. The power sensors are constantly sending information about the turning machine's power supply and consumption via Modbus/TCP. Furthermore, local hosts and cloud services synchronize their state using Hypertext Transport Protocol - Secure (HTTPS) based protocols. Figs. 4(a) and 4(b) show the protocol distributions during the two weeks of regular operation.

3.2. Factory operation: Benign and malicious

During the four experiment days, the turning machine was turned on, except for day A1 (cf. Fig. 3). Except for the three additional target hosts and the attacker's host, the entire factory cell operated normally. The protocol distributions are similar to the ones of the regular operation (cf. Figs. 5(a), 5(b), 6(a) and 6(b)).

4. Description of the data set

The data was collected from the turning cell that hosts and connects the turning machine depicted in Fig. 2. Table 3 shows details about every of the available PCAP files.

The data set comprises 173 GB of PCAP files, which have been collected during 395.2 h of smart factory operation. The data set also comprises various modern IP-based factory communication protocols such as MQTT, OPC UA, and Modbus/TCP.

The pilot factory granted permission to capture a data set in their real OT communication network and release it to the public. For this purpose, a penetration tester from [company] conducted a series of attacks within two different scenarios consisting of two dedicated sets of attacks:

Fig. 2. The turning machine of the pilot factory in Vienna, Austria (in operation). In the blue box, there is the PCU. The NCU is mounted inside, on the rear.

- Aggressive (referred to as **attack set A** in the following): The attacker is informed that there is no IDS or monitoring system deployed in this network, so he can run any attack without the risk of being detected.
- Stealth referred to as **attack set S**: Assuming an IDS is in use, the attacker is advised to "stay below the radar" and conceal all attacks. Furthermore, the attacker's Internet Protocol (IP) address (.91) is included in the training data week by temporarily assigning the address to a trusted host (training file S).

5. Adversary model and attack scenarios

For this experiment, the attacker is connected to the local network and uses a dedicated IP address. However, the earlier-mentioned two scenarios are subject to distinct prerequisites:

- In the aggressive scenario:
  - the attacker is informed that there is no operational IDS in this network.
  - the attacker is not limited regarding the rate or "aggressiveness" of attacks.
  - the attacker's IP address has not been used before by authorized hosts.
- In the stealth scenario:

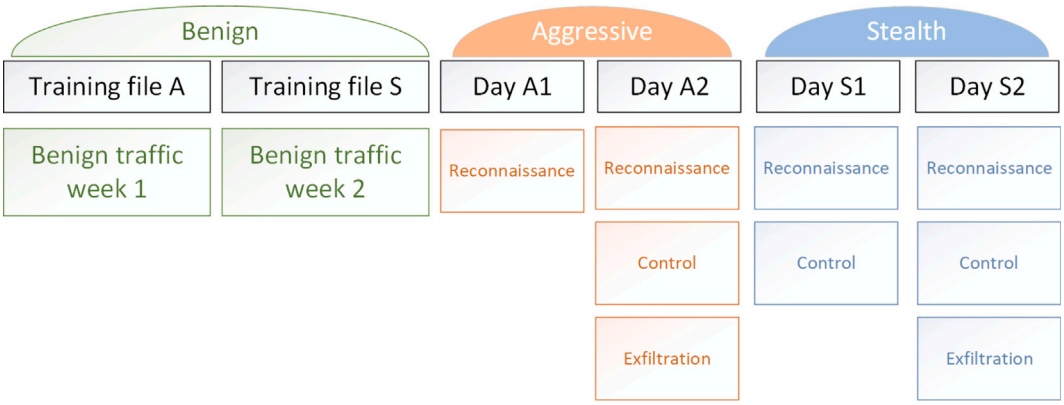
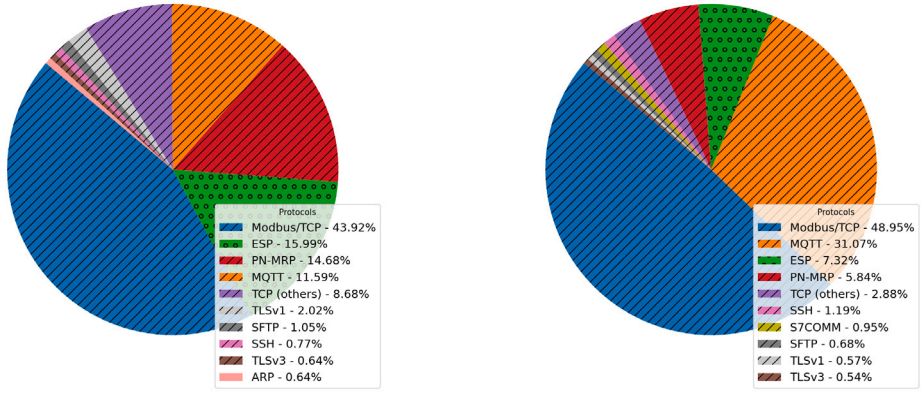


Fig. 3. Workflow of the data set creation with emphasis on the attacks.



(a) Training file A (b) Training file S

Fig. 4. Protocol distribution during the two weeks of training data acquisition (training files A and S, respectively).

- the attacker is aware of the presence of an IDS deployed in the network.
- the attacker uses low-rate and concealed attacks, contacting only a few hosts and transferring fewer data at a time compared to the aggressive scenario.
- the attacker uses an IP address that authorized hosts have used before. This is done on purpose to test the capability of the IDS that, for instance, cannot rely on trusted IP addresses.

The attacker's resources are restricted to one business laptop, built in 2023, with an Intel Core i7 and 32 GB of Random Access Memory (RAM). The attacker, furthermore, is assumed to possess knowledge and skills on the level of a single activist/hacker, referring to Table 11.2 of [15].

The attacker is furthermore assumed to not know credentials, technical details, nor additional information about assets such as IP addresses, host names, etc., with one exception: The IP addresses of the three target hosts were provided to the attacker (pentester) to ensure that only those are attacked with active attacks. Machines that are part of the production system are only included for reconnaissance activities.

The successful setup in the factory was followed by five days of attacks. Note that the entire data of the second day (which would be A2) (the DoS attack) is omitted from the data set due to the large file size of 86.1 GB. We therefore named the third day A2 for simplicity, therefore having two aggressive and two stealth days of network attack data.

The data set itself can be found in our repository (cf. Section 1), where the PCAP files, as well as all extracted and labeled flows, are provided.

The data set can be used either by extracting desired information from the PCAP files or by using the labeled flows found in the repository. The flows are extracted using the tool go-flows [16] and labeled based on information that is available for us, such as attacker's IP and Media Access Control (MAC) addresses, time frames, and other details such as packet sizes and Time To Live (TTL) using a custom written Python script.

For the extraction of flows, a subset of the multikey feature vector by Meghdouri et al. is used [17]. Our repository contains a list of all extracted features.

The experiments were conducted in October and November 2023, with one week of preparation and two days of attacks each. During the preparation week, the operational traffic of the factory cell was recorded.

5.1. Files and sizes

The data set consists of six PCAP files and has a total size of 173 GB (intentionally excluding the DoS attack). There are four days in total that contain malicious traffic (cf. Fig. 3), with an aggregated file size of 28.4 GB. The other two files are training files containing one week of traffic right before the experiment days each. Sizes are: 53.5 GB (**TF A**) and 93.2 GB (**TF S**) for the training files. These training files of usual operational traffic can be used for instance to train the "default" state that anomaly detectors rely on.

Table 1
Attacks performed during the four attack days: A1, A2, S1, and S2.

Attack command	Attack category	Day	Attack description
fping	Reconnaissance	A1, A2, S2	Simple host discovery in local network
nmap	Reconnaissance	A1, A2, S1	Port scans in local network
sqlmap	Reconnaissance	A1	Automated SQL injection attempts
gobuster	Reconnaissance	A2	Automated search for web directories
revshell	Control	A2, S1	Reverse shell
file exfiltration(FTP)	Exfiltration	A2	Exfiltration of file via FTP upload
meterpreter	Control, exfiltration	S2	Metasploit-based reverse shell

5.1.1. Time frames and timestamps

The series A and S come in a bundle with the appropriate training and attack files. The training files contain one week of traffic each and were captured the week before the attacks were conducted. For example, training data was captured from Monday to Sunday. The attacks were conducted on Monday and Tuesday (attack set S) and Monday and Wednesday (attack set A). On Tuesday, a prolonged DoS attack was performed and therefore the resulting data set is relatively large (97.5 GB). We decided not to label or analyze this data set. However, the data of Tuesday of set A is included in the publicly available data as well, and ready for download (cf. Section 1). The data sets were created and recorded in October 2023 (attack set A) and November 2023 (attack set S).

All timestamps are recorded as utc+1 local time. Enterprise-grade DAG hardware with Pulse Per Second (PPS) synchronization was used and attached to our DAG system to obtain highly precise timestamps. This accurate timing allows the correlation of distributed network captures with potential external (application-layer) events.

In Fig. 1, the sensors, actuators, and network gear are part of the factory cell network. At the same time, the black hosts with red descriptions underneath are involved in the penetration testing scenarios.

The switch of the turning cell is an enterprise-grade gigabit Ethernet switch. One port of the switch is configured as a mirror port. This mirror path sends a copy of every frame leaving the switch to the DAG server's capture interface.

During the four measurement days, the target hosts were attacked in two modes: two days for aggressive attacks and two days for stealth, concealed attacks.

On the one hand, option "-T5" is the fastest scanning mode and the easiest to detect in the network. On the other hand, the option "-T1" takes a very long time to deliver results but is harder to detect, e.g., by an IDS.

6. Attacks

In both of the scenarios, the attacker first obtains knowledge of the local assets in the network and then attacks by obtaining control of the target hosts and stealing confidential data through exfiltration. Apart from the target hosts, it is assumed that the attacker has no knowledge of the attacked infrastructure and network at the beginning of any scenario.

6.1. Data set A - Aggressive scenario

In this scenario, the adversary is allowed to run arbitrary attacks, as there is no caution needed to remain undetected. The first days (A1) focused on reconnaissance, for which the attacker uses the tools fping [18], nmap [19] and sqlmap [20] on Day A1 with maximum aggressiveness. Then, on the second day A2, the web application of host .92 (Metasploitable 3) is scanned using the tool gobuster [21]. On the same day, a PHP-based reverse shell is used, and lastly, a file exfiltration attack using the FTP protocol is performed on the same host.

6.2. Data set S- Stealth scenario

In this scenario, the attacker is tasked to remain as undetected as possible by the IDS deployed within the network and attack in a way as concealed as possible. The number of ports scanned was adjusted to the duration of the scan. The following scans were performed using the tools nmap and fping on days S1 and S2:

- SYN scan of all ports
- TCP scan of all ports
- nmap SYN scan with speed T0 of the ports: 21, 22, 80, 443, 445
- nmap SYN scan with speed T1 of the top 20 ports
- nmap SYN scan with speed T2 of the top 100 ports

On day S1, a reverse shell is run on host .93 (Kubuntu with DVWA). On day S2, a meterpreter-based remote shell is used to execute commands and exfiltrate files from the target host in a slower and more concealed way compared to days A1–A2.

Labeling was performed considering the recommendations of Flood et al. [9]: We did not only rely on the attackers IP address to label the traffic as malicious but several individual parameters of every single attack such as packet size, TTL, receiver address, protocol identifier, source port, etc. which led to a more accurate labeling: while packets that are part of the attack itself in forward direction were labeled as malicious, the target's responds traffic was not — with all reverse shells as only exception. Furthermore, in the PCAP files it can be observed that the attacker's host sent administrative traffic such as NETBIOS [22] hostname resolution, etc., sent Domain Name System (DNS) requests or did research on the internet. All these actions have been labeled as benign traffic since they are not part of attacks.

The File Transfer Protocol (FTP) protocol and a Metasploit remote shell are used for the exfiltration process. The exhaustive list of the conducted attacks can be found in Table 1. Days marked in red contain only aggressive modes of attacks, and days marked in blue contain only stealth modes of attacks.

7. Analysis of the data set

This section contains detailed properties of the provided data set.

7.1. Protocols

Figs. 4(a) and 4(b) show the network traffic's protocol distribution during 14 days of normal factory operation. The legend shows the top 10 observed protocols by prevalence. For the full list of protocols identified, however, please refer to the file " identified_protocols.txt" within the repository of the provided data set(cf. Section 1). Please note the meaning of the patterns, too: User Datagram Protocol (UDP)-based protocols are marked with circles, whereas Transmission Control Protocol (TCP)-based protocols are marked with stripes within the diagrams. The legend shows the ten most common protocols during the recording in terms of the number of packets transmitted — from the most common protocol to the least common protocol.

What is observable here is e.g. that OPC UA is only among the top 10 protocols on days S1 and A2. beyond that, SSH is only found in the

Table 2
Starting times/Sequence numbers of packet, of performed attacks (time zone: Central European).

Data set	First attack appearance in data set		
	Tool	Timestamp	Sequence number
A1	fping	2023-10-23 10:46:41	1 489 166
	nmap	2023-10-23 11:01:48	2 124 001
	sqlmap	2023-10-23 15:09:03	14 064 302
	meterpreter	2023-10-23 16:53:09	18 483 608
A2	fping	2023-10-25 10:07:22	495 722
	nmap	2023-10-25 10:08:40	555 316
	gobuster	2023-10-25 10:08:40	555 317
	revshell	2023-10-25 12:16:51	7 143 051
	exfiltration(ftp)	2023-10-25 13:40:31	11 115 494
S1	fping	2023-11-21 09:23:26	64 029
	meterpreter	2023-11-21 09:31:00	423 486
	nmap	2023-11-21 09:31:00	423 487
	revshell	2023-11-21 10:01:34	2 025 510
	exfiltration	2023-11-21 10:34:29	3 645 668
S2	fping	2023-11-20 09:31:26	7 133 768
	nmap	2023-11-20 10:24:20	10 035 491
	gobuster	2023-11-20 10:24:20	10 035 741
	revshell	2023-11-20 14:33:52	22 587 712
	meterpreter	2023-11-20 15:00:23	23 947 386

Table 3
Data set files, their content, and details.

Data set	Time captured	Duration	No. Pkts.	No. bytes	No. flows	Type	Attack-actions
TFA	2023-10-15 22:30:01 2023-10-22 12:09:26	6 d 13 h 39 m	176.615.865	56 470 492 187	769 903	Normal operation	–
A1	2023-10-23 10:11:35 2023-10-23 20:31:58	10 h 20 m	27.847.171	5 574 769 640	370 363	Aggressive attacks	Reconnaissance
A2	2023-10-25 09:56:40 2023-10-25 16:29:09	6 h 32 m	18.882.114	3 913 818 540	348 195	Aggressive attacks	Reconnaissance, control, exfiltration
TFS	2023-11-12 19:23:44 2023-11-19 19:23:44	7 d 0 h 0 m	493.159.029	97 417 854 436	553 675	Normal operation	–
S1	2023-11-20 06:59:49 2023-11-20 21:41:47	14 h 41 m	51.258.814	10 925 600 828	199 553	Stealth attacks	Reconnaissance, control
S2	2023-11-21 09:22:00 2023-11-21 01:02:05	08 h 19 m	44.574.874	11 937 655 812	166 231	Stealth attacks	Reconnaissance, control, exfiltration
Sum:		15 d 5 h 31 m	812.337.867	186.240.191.443	2.407.920		

training files' top 10 protocols — indicating that the attacker never used the SSH protocol. Interestingly, the relative amount of Modbus/TCP traffic did not differ between training files and attack files as expected: some of the attack files (A1 for example) have a relatively high amount of Modbus/TCP traffic — Even though the applicant did not introduce a single Modbus/TCP packet himself. For us, this is a hint that the attacks generally made up a minority of traffic.

Lastly, we found that a significant amount of Transport Layer Security (TLS) v1 traffic (between 0.5 and 2.02) is observable in the training data (TF A and TF S).

Fig. 5 depicts the protocol distributions for the aggressive attack days A1 in Fig. 5(a) and A2 in Fig. 5(b), whereas 6 illustrates the protocol distributions on the two stealthy attack days S1 in Fig. 6(a) and S2 in Fig. 6(b), respectively. For both figures, the following labeling is applied: Segments with stripes build upon the TCP protocol whereas segments with rings use UDP.

7.1.1. Progression of traffic over time

Figs. 7 and 8 show the progression of the network traffic over time, i.e., the number of bytes and packets exchanged in the local network in the form of a bar plot. Each bar depicts the accumulated amount of bytes/packets within one minute. Since the two fingers belong to the training files (TF A and TF S), the total time frame is around one week, around 10,000 min. while the traffic was very regular during the second training week (TF S), traffic during the first training week (TF A) was meager at around 60% of the time.

7.1.2. Progression of the attacks over time

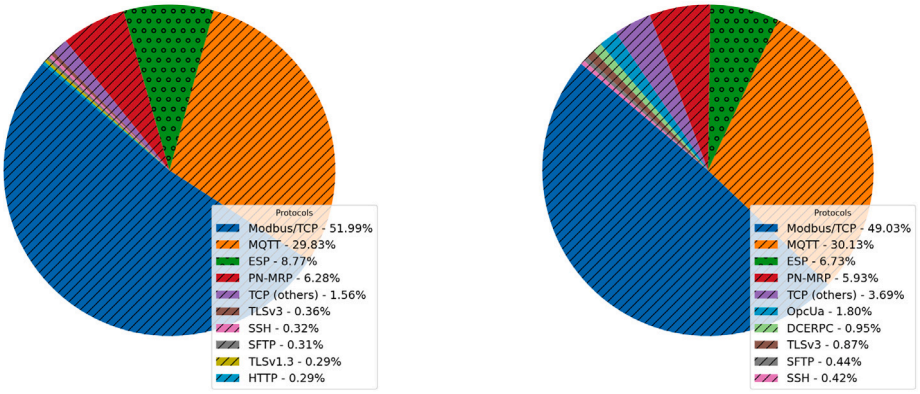
Figs. 9–12 show the courses of attacks over time. Some benign traffic peaks are caused by attacks, e.g., from reflections, i.e., replies of benign hosts to attack traffic. Those are marked as benign traffic.

7.1.1 furthermore allows us to identify which traffic peaks are caused by attacks since both are bar plots with a time block size of one minute (cf. also Table 2).

The Y-axis represents the number of packets accumulated over a one-minute time block, while the X-axis indicates the time. The legend is a crucial tool, as it shows the specific color associated with each attack, aiding in the interpretation of the graphs.

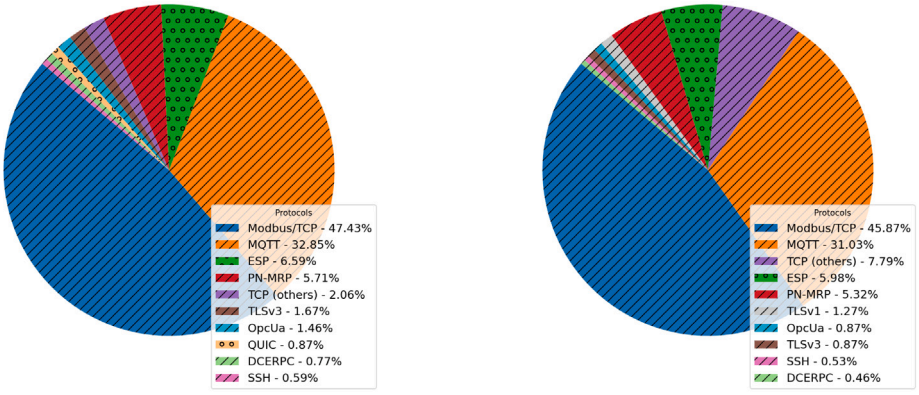
7.1.3. Histograms of flow durations

This section discusses the distribution of flow durations of the data set. Note that the two training data sets (TF A and TF S) contain benign traffic only. The four days A1, A2, S1, and S2, on the other hand, show the flow durations of benign traffic in blue and the flow durations of malicious traffic in red, slightly transparent so that both distributions are visible at the same time. Figs. 13 and 14 show the distributions of flow durations for the days A1 and S2. All figures for all of the data sets can be found in our data set repository. We chose to include these



(a) Day A1 (b) Day A2

Fig. 5. Protocol distribution of operational traffic and attack traffic on days A1 and A2, respectively.



(a) Day S1 (b) Day S2

Fig. 6. Protocol distribution of operational traffic and attack traffic on days S1 and S2, respectively.

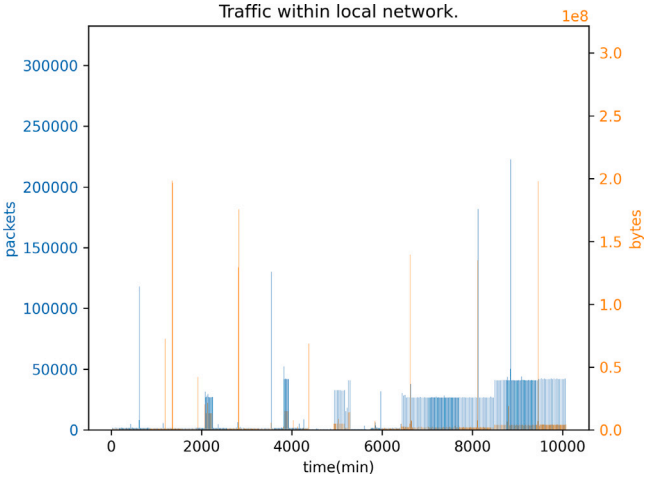


Fig. 7. Packets and bytes over time, training file A (TF A).

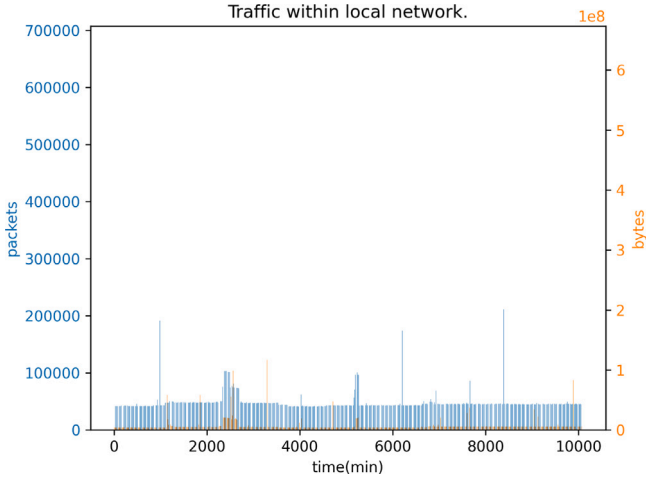


Fig. 8. Packets and bytes over time, training file S (TF S).

two examples since they show a clear difference between benign and malicious traffic is observable.

8. Findings

This section points out the most relevant findings of this paper.

One important finding is the remarkable traffic regularity that we could observe in the factory network. This regularity is generally beneficial for attack detection because traffic peaks (cf. Section 7.1.1) as well as flow amount peaks (cf. Section 7.1.2) and also differences in flow duration distribution were observable for benign and attack traffic — in both the aggressive and the stealth data sets.

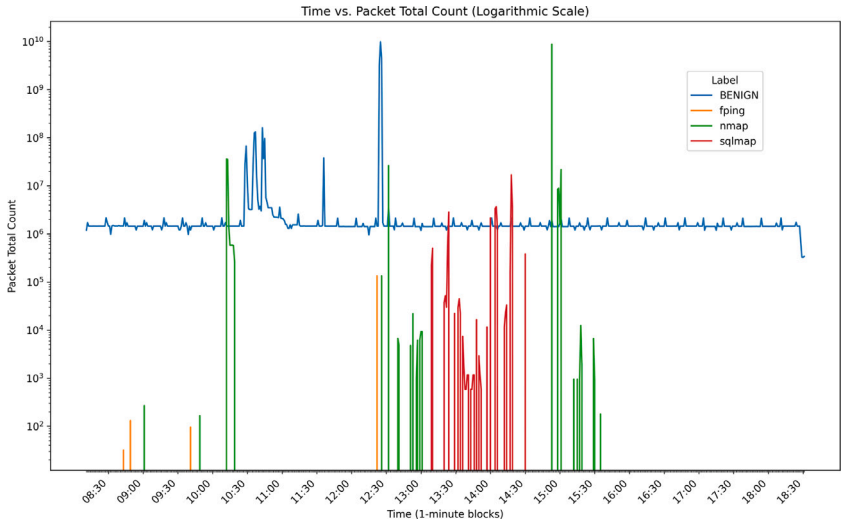


Fig. 9. The graph shows the attacks performed by the pentester over time on day A1.

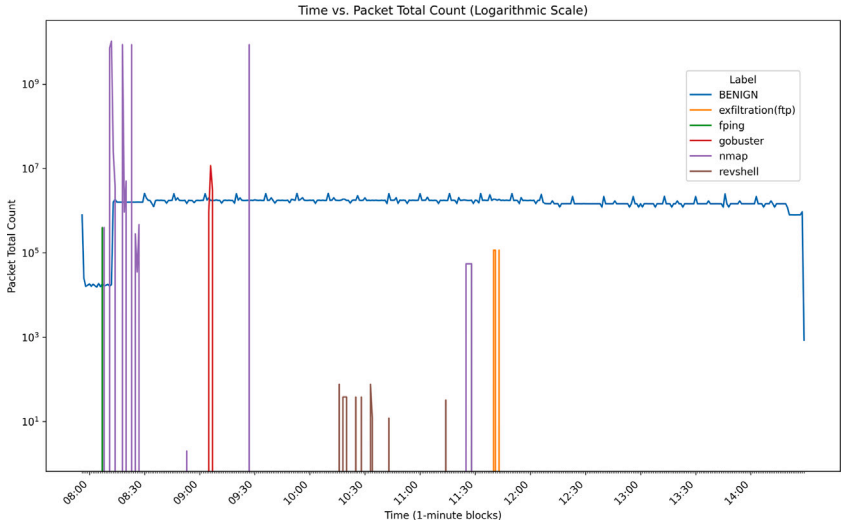


Fig. 10. The graph shows the attacks performed by the pentester over time on day A2.

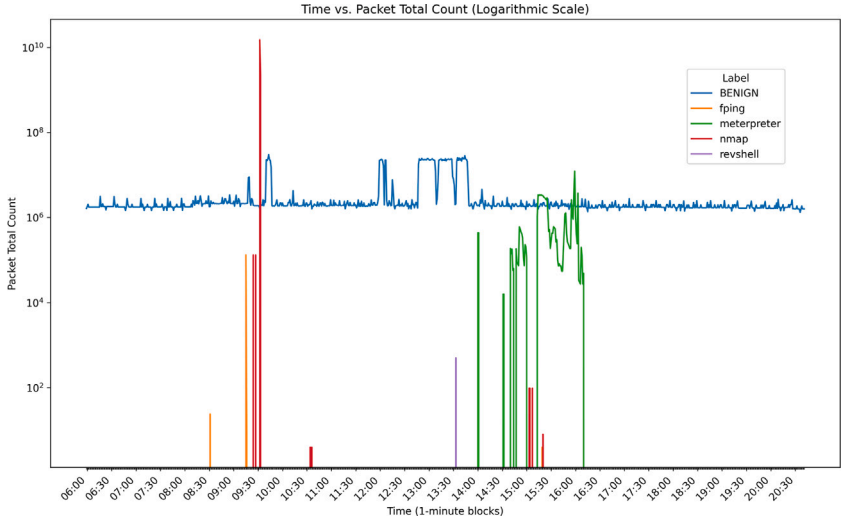


Fig. 11. The graph shows the attacks performed by the pentester over time on day S1.

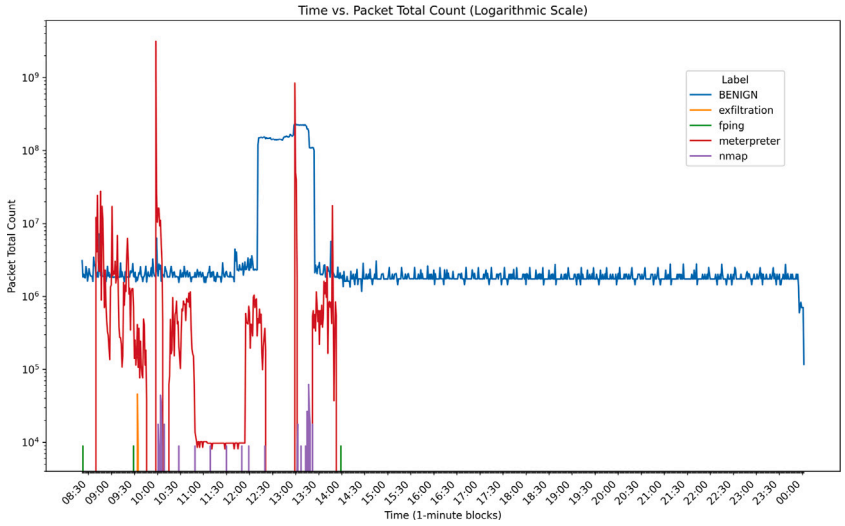


Fig. 12. The graph shows the attacks performed by the pentester over time on day S2.

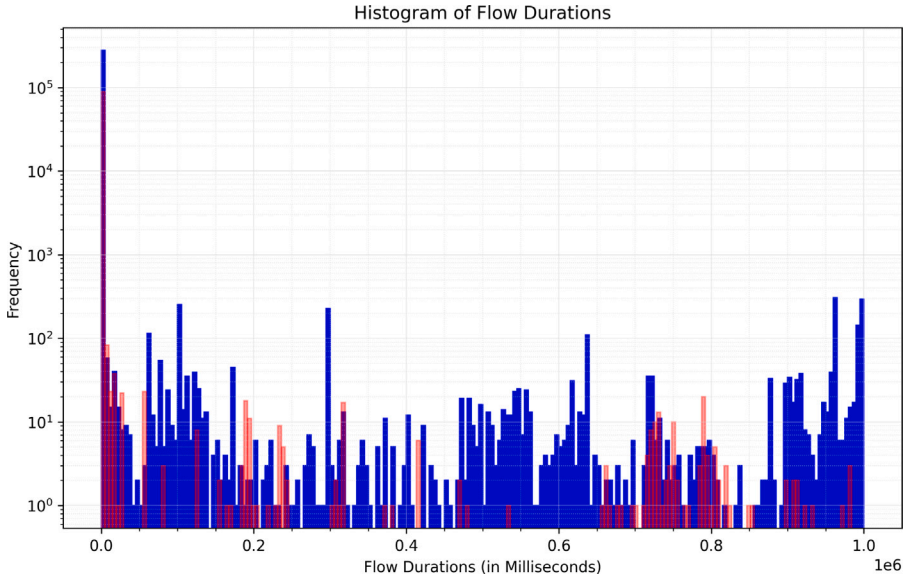


Fig. 13. Histogram of flow durations. Traffic: Day A1. Blue: Benign traffic. Red (transparent): Attack traffic.

At first glance, it seems that the protocol distribution did not change significantly when attacks were introduced into the network (cf. Section 7.1). But in fact, this is due to the amount of benign traffic in comparison to the amount of malicious traffic.

Another finding was that the difference in the distribution of the flow durations between operational and malicious traffic was significant enough to be even easily observable with the plane eye for every one of the attack days.

Note that the graphs shown in this paper are the only a selection. We created all graphs seen in this paper for every of the PCAP files. The complete collection of grass is also found on the repository (cf. Section 1).

9. Conclusion

In our related work, we have shown that most data sets available either is generated traffic or obtained it from a testbed environment (cf. Section 2). The data set provided with this paper, in contrast, is obtained from a real factory OT network. It is one of very few OT data sets that contain both operational traffic and authentic attack traffic in an authentic OT network.

The paper itself describes the data set in detail. We document the methodology and assumptions made during the data capturing along with the documentation of our technical setup and present the experiments and attacks that led to the data set.

The data set consists of two weeks of benign operational traffic and four days in which different attacks are conducted by a pentester. We include two scenarios: an aggressive scenario where no effort is put into concealing attacks, and a stealthy scenario in which the attacker tries to avoid detection.

The data set was created in October and November of 2023. It contains modern Machine to Machine (M2M) communication protocols such as OPC UA, Modbus/TCP, and MQTT and stems from modern

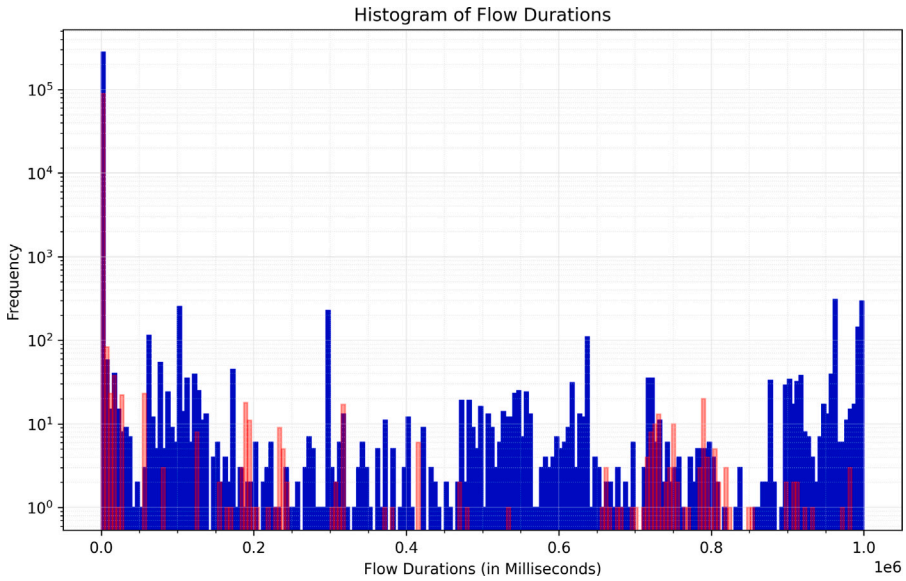


Fig. 14. Histogram of flow durations. Traffic: Day A1. Blue: Benign traffic. Red (transparent): Attack traffic.

industrial control network equipment. The data set can be downloaded and used under the CC BY 4.0 license. (cf. Section 1).

CRediT authorship contribution statement

Bernhard Brenner: Conceptualization, Data curation, Methodology, Validation, Writing – original draft, Formal analysis, Investigation, Visualization. **Joachim Fabini:** Conceptualization, Data curation, Formal analysis, Project administration, Resources, Validation, Writing – review & editing. **Magnus Offermanns:** Conceptualization, Data curation, Investigation, Writing – review & editing. **Sabrina Semper:** Data curation, Investigation, Methodology, Supervision, Validation, Writing – review & editing. **Tanja Zseby:** Conceptualization, Formal analysis, Methodology, Project administration, Resources, Supervision, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no competing financial interests or personal relationships that could have appeared toinfluence the work reported in this paper

Data availability

The data is available at the URL Provided within the paper: Footnote in introduction.

Acknowledgments

This work was enabled by TÜV AUSTRIA #safeseclab Research Lab for Safety and Security in Industry, a research cooperation between TU Wien and TÜV AUSTRIA.

References

[1] T. Morris, A. Srivastava, B. Reaves, W. Gao, K. Pavurapu, R. Reddi, A control system testbed to validate critical infrastructure protection concepts, Int. J. Crit. Infrastruct. Prot. 4 (2) (2011) 88–103.

[2] T. Morris, W. Gao, Industrial control system traffic data sets for intrusion detection research, in: International Conference on Critical Infrastructure Protection, Springer, 2014, pp. 65–78.

[3] T.H. Morris, Z. Thornton, I. Turnipseed, Industrial control system simulation and data logging for intrusion detection system research, in: 7th Annual Southeastern Cyber Security Summit, 2015, pp. 3–4.

[4] T. Morris, Tommy morris - industrial control system (ICS) cyber attack datasets, 2015, URL https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets. (Accessed 09 June 2021).

[5] Á.L.P. Gómez, L.F. Maimó, A.H. Celdrán, F.J.G. Clemente, C.C. Sarmiento, C.J.D.C. Masa, R.M. Nistal, On the generation of anomaly detection datasets in industrial control systems, IEEE Access 7 (2019) 177460–177473.

[6] GitHub, GitHub - antoine-lemay/modbus_dataset: Modbus dataset from CSET 2016, 2016, URL https://github.com/antoine-lemay/Modbus_dataset. (Accessed 15 June 2024).

[7] A. Alatram, L.F. Sikos, M. Johnstone, P. Szewczyk, J.J. Kang, DoS/DDoS-MQTT-IoT: a dataset for evaluating intrusions in IoT networks using the MQTT protocol, Comput. Netw. 231 (2023) 109809.

[8] A.K. Sarica, P. Angin, A novel sdn dataset for intrusion detection in iot networks, in: 2020 16th International Conference on Network and Service Management, CNSM, IEEE, 2020, pp. 1–5.

[9] R. Flood, G. Engelen, D. Aspinall, L. Desmet, Bad design smells in benchmark NIDS datasets, in: Proceedings of the European Symposium on Security and Privacy, EuroS&P, IEEE, Edinburgh, United Kingdom and Leuven, Belgium, 2024, *University of Edinburgh* and *KU Leuven*.

[10] N. Moustafa, A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets, Sustainable Cities Soc. 72 (2021) 102994.

[11] N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull, Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset, Future Gener. Comput. Syst. 100 (2019) 779–796.

[12] Sourceforge, Metasploitable - browse /metasploitable2 at SourceForge.net, 2019, URL https://sourceforge.net/projects/metasploitable/files/Metasploitable2/. (Accessed 13 June 2024).

[13] GitHub, GitHub - rapid7/metasploitable3: Metasploitable3 is a VM that is built from the ground up with a large amount of security vulnerabilities, 2024, URL https://github.com/rapid7/metasploitable3. (Accessed 13 June 2024).

[14] Vulnhub, Damn vulnerable web application (DVWA): 1.0.7 VulnHub, 2011, URL https://www.vulnhub.com/entry/damn-vulnerable-web-application-dvwa-107,43/. (Accessed 13 June 2024).

[15] B. Brenner, A. Ekelhart, in: S. Biffl, M. Eckhart, A. Lüder, E. Weippl (Eds.), Security Analysis and Improvement of Data Logistics in AutomationML Based Engineering Networks, Springer, 2019, p. 31.

[16] G. Vormayr, J. Fabini, T. Zseby, Why are my flows different? a tutorial on flow exporters, IEEE Commun. Surv. Tutor. 22 (3) (2020) 2064–2103.

[17] F. Meghdouri, F.I. Vázquez, T. Zseby, Cross-Layer profiling of encrypted network data for anomaly detection, in: 2020 IEEE 7th International Conference on Data Science and Advanced Analytics, DSAA, IEEE, 2020, pp. 469–478.

[18] Fping, Fping homepage, 2024, URL https://fping.org/. (Accessed 28 July 2024).

[19] Nmap, Nmap: the network mapper - free security scanner, 2024, URL https://nmap.org/. (Accessed 10 October 2024).
[20] Sqlmap, Sqlmap: automatic SQL injection and database takeover tool, 2024, URL https://sqlmap.org/. (Accessed 28 July 2024).
[21] GitHub, GitHub - OJ/gobuster: directory/file, DNS and vhost busting tool written in go, 2023, URL https://github.com/OJ/gobuster. (Accessed 21 July 2024).
[22] Datatracker, RFC 1001 - protocol standard for a netbios service on a TCP/UDP transport: Concepts and methods, 1987, URL https://datatracker.ietf.org/doc/html/rfc1001. (Accessed 12 September 2024).

Bernhard Brenner received a B.Sc. in Medical Informatics from TU Wien, Austria, and an M.Sc. from Denmarks Technical University (DTU), Denmark, in Computer Security. He is now working on his Ph.D. degree at TU Wien, focusing on cybersecurity in OT networks.

Joachim Fabini is Senior Scientist with the Institute of Telecommunications at TU Wien. He received his Diploma (Dipl.-Ing.) degree in technical computer sciences and Ph.D. degree in electrical engineering from TU Wien. His research interests include measurement methodologies and metrics in packet-switched networks, time synchronization, and network architectures for secure communication in critical infrastructures.

Magnus Offermanns received a B.Sc. in Information Technology and an M.Sc. in Information and Communication Engineering from the University of Klagenfurt, Austria. He is now working as a penetration tester and cybersecurity consultant.

Sabrina Semper, an IT professional, has spent the last 20 years in the IT Industry. She holds a Bachelor's Degree in Information- and Communication systems and a Master's Degree in IT-Security at the University of Applied Science Technikum Wien. She began her career as operations engineer at a major provider of government solutions. Eight years ago, she transitioned to IT-Security and becoming a white hat hacker. Recently her focus has shifted to OT-Security, gaining expert knowledge in the IEC 62443 standards. At the TÜV Austria TRUST IT she works as a Senior Consultant and helps customers to protect and secure their (critical) infrastructure.

Tanja Zseby is a full professor of communication networks at the Faculty of Electrical Engineering and Information Technology at TU Wien. She received her diploma and her Ph.D. degree at TU Berlin, Germany in electrical engineering. Before joining TU Wien, she led the Competence Center for Network Research at the Fraunhofer Institute for Open Communication Systems (FOKUS) in Berlin and worked as visiting scientist at the University of California, San Diego.