IEEE Open Journal of the
Industrial Electronics Society

Received 19 April 2023; revised 6 July 2023; accepted 17 July 2023. Date of publication 19 July 2023;
date of current version 4 August 2023. The review of this paper was arranged by Associate Editor Yang Shi.

Digital Object Identifier 10.1109/OJIES.2023.3297057

Better Safe Than Sorry: Risk Management Based on a Safety-Augmented Network Intrusion Detection System

BERNHARD BRENNER [1], SIEGFRIED HOLLERER [2], PUSHPARAJ BHOSALE [2], THILO SAUTER[3,4] (Fellow, IEEE), WOLFGANG KASTNER [2], JOACHIM FABINI [1], AND TANJA ZSEBY [1]

[1]Institute of Telecommunications, TU Wien, 1040 Vienna, Austria
[2]Institute of Computer Engineering, TU Wien, 1040 Vienna, Austria
[3]Institute of Computer Technology, TU Wien, 1040 Vienna, Austria
[4]Center for Distributed Systems and Sensor Networks, Donau-Universität Krems, 2700 Wiener Neustadt, Austria

CORRESPONDING AUTHOR: BERNHARD BRENNER (e-mail: bernhard.brenner@tuwien.ac.at).

ABSTRACT Interconnected industrial control system (ICS) networks based on routable protocols are susceptible to remote attacks similar to classical information technology (IT) networks. However, addressing ICS security in an isolated view is dangerous since ICSs have to ensure safety measures for people, processes, and the environment. The safety and security of ICSs are often addressed separately, without considering their important interrelation. Safety measures can violate security policies (e.g., an emergency stop function accessible by anyone); likewise, a security incident can violate safety policies (e.g., by increasing reaction time). In this article, we propose a network-based intrusion detection system with the interrelation between safety and security in mind. It detects security incidents while evaluating possible safety-related consequences of both the detected attack and possible countermeasures. We evaluate our approach with a Proof of Concept (PoC). The alerts generated by the PoC prototype serve as the basis for a risk management strategy proposed in this article. Our approach provides a basis for safety-aware intrusion detection in smart factories and other cyber-physical systems.

INDEX TERMS Industrial control systems (ICSs), incident response, information technology (IT) / operational technology (OT) convergence, OT security, risk management, safety.

I. INTRODUCTION

The fourth industrial revolution, as well as the increased use of interconnected commodity hardware and remote services in the production system environment, introduced the need for secure communication and monitoring of communication networks. As opposed to conventional computers, industrial components provide a connection to the physical world. They need to be protected against attacks or misconfigurations that risk the injury of operators, damage the equipment, or endanger the operation. Attacks such as TRITON, LockerGoga, WannaCry, and other ransomware are increasing in prevalence and relevance, and so are targeted cyberattacks, e.g., the attacks on the Ukrainian power grid in 2015 and 2016 [1], [2].

An intrusion detection system (IDS) is an important alert tool for operators and administrators. It serves as an active security monitor in addition to conventional measures, such as firewalls or antivirus applications. Several IDS concepts for information technology (IT) and also for operational technology (OT) networks exist. A small yet increasing number use machine-learning-based concepts at the time of writing (cf. Section II).

In this article, we present a concept for an IDS that also assesses safety risks in combination with the detection of security incidents, prioritizes attacks, and suggests appropriate countermeasures according to the security and safety situation. Some existing approaches (e.g., [3]) have a safety focus but operate on an operational level only. However, to the best of our knowledge, none of the currently existing IDS concepts considers security incidents with their possible safety-related consequences on a risk management level.

The IDS serves as an indicator of risks that originate from the security realm and impact the system's safety. A failure or even a countermeasure targeting a safety weakness can be a threat to the industrial control system (ICS)'s security and vice versa. This relation is formalized as follows (ordered most to least favorable w.r.t. functionality and advantages of the relation) in [4].

1) *Mutual Reinforcement:* An implemented security measure enhances safety or vice versa. Mutual reinforcement is the most favorable relation as it enables resource optimization and, thus, cost reduction. For instance, incidents detected by an IDS can be used to derive the possible safety impact on the ICS, or a safety anomaly (e.g., failure of relevant hosts) is also detected by the IDS as an anomaly.
2) *Independent:* Safety and security measures are independent of each other. It is the next best option after mutual reinforcement because it does not enhance or restrict the overall relation. For instance, under normal working conditions, no dependence between safety and security exists.
3) *Conditional Dependency:* If an incident occurs, a safety measure introduces some condition on a security measure or vice versa. This relation is a less favorable relation as one measure puts a restriction on the other. For instance, in an incident (*fire*), a safety measure (*open doors, which are normally closed, to evacuate*) is implemented. It weakens security measures (*provide physical access to an attacker*) and may lead to a malicious attack, thus increasing the risk.
4) *Antagonism:* When safety and security are considered together, some requirements result in conflicting goals. This relation is the least favorable as safety requirements can contradict security requirements and vice versa. For instance, consider the *response time*. As a safety measure, the response time should be immediate. In contrast, security measures may demand first to authorize *with username and password* and then implement safety measures, thus increasing the *response time*.

The contribution of this work is an approach to identify security incidents in ICS networks, estimate possible safety and security-related consequences of these incidents, and suggest countermeasures by considering the interdependencies between safety and security in a reactive manner. Security incidents are detected by analyzing network traffic in the ICS environment, and safety risks emerging from these incidents are evaluated. Risk assessment based on the incident analysis provided by the underlying IDS is undertaken to determine risks and suggest possible treatment measures.

The objective of the underlying IDS is, therefore, split into the following three:

1) Detect and identify security incidents;
2) assess and quantify resulting risks;
3) list available countermeasures and suggest them according to an assessed priority.

The rest of the article is organized as follows. Section I provides an introduction to the problem and lists our contributions. Section II describes the current state of the art. The concept is described in Section III. Our use case is described in Section IV, and our methodology is described in more detail in Section V. Section VI lists and discusses the results. Finally, Section VII concludes this article.

II. STATE OF THE ART

A. INTRUSION DETECTION SYSTEMS

Network security systems are becoming smarter, cheaper, and more prevalent. For example, major network device vendors, including Cisco, Huawei, and Juniper, offer commercial IDS solutions. Although most vendors still specialize in classical IT-based IDSs and intrusion prevention system (IPS), there exist a few IDS and IPS already focusing either on OT or being at least (claimed) eligible for OT use.

Many open-source IDS solutions have been built for either IT or OT environments. Some commercial products already support detection in OT traffic (e.g., Darktrace, Otorio Ram^2, Nozomi, Claroty). Based on information from their website, Darktrace appears to be using machine learning algorithms.

In [5], three types of IDS are distinguished: 1) *signature-based IDS*; 2) *anomaly-based IDS*; and 3) *specification-based IDS*.

The most popular type of intrusion detection is the *signature-based* approach. Here, network traffic is examined for predefined patterns evident for attacks: For example, traffic with known malicious patterns or suspicious packet content can trigger alerts. Such patterns are added to the systems in the form of rules, similar to firewall rules. The rule sets are regularly updated from a known, trusted source, similar to malware signatures in antivirus software. Systems such as Snort, Zeek, and Suricata fall into the signature-based category [6], [7], [8]. While signature-based detection systems offer a good detection performance and are comparably lightweight in terms of their computational load, their main weakness is that they can only detect attacks that have previously been identified and converted into specific machine-readable patterns [9], [10].

As opposed to signature-based approaches, *anomaly-based* approaches define a model for normal traffic and detect deviations as potential suspicious traffic. Depending on the particular approach, suspicious network traffic patterns can be further classified according to the type of anomaly. These approaches can be implemented in various ways. Two possible ways are using descriptive statistics or machine learning. In both cases, it is necessary to define thresholds for defining a maximum deviation that is to be tolerated for normal traffic. In the case of descriptive statistics, an IDS can use simple statistics such as packet counts, packet rates, average packet size, average destinations per source, maximum flow duration in milliseconds, the standard deviation of the interpacket time, etc., as the basis for the detection. If one or more of these values exceed the threshold, then an alert could be raised. One

remarkable result in this regard was achieved by Mantere et al. in 2012 [11]: The authors observed that inspecting only the average packet size was sufficient to detect anomalies reliably in their use case.

More complex anomaly detection systems often rely on more sophisticated statistical techniques such as hidden Markov models [12] or machine-learning-based techniques, such as neural networks, support vector machines (SVMs), etc.

Conceptually, machine learning aims at finding a relation between the input vector and label where human beings can only assume the existence of such a relation. In this case, chosen network traffic characteristics that are called features, such as the packet sizes, sending patterns, or flags, trigger the classification of observed network traffic into "normal" and "suspicious" traffic or even categorizing it as the most probable type of attack. There are three classes of machine learning methods for detecting attacks: 1) Supervised; 2) semisupervised; and 3) unsupervised. Supervised methods learn relations based on labeled data, i.e., data that have all features and the corresponding label, such as binary labels "Benign"/"Malicious," or labels that identify an attack "DDoS"/"Portscan"/"Botnet," etc.

Therefore, signature-based and supervised machine-learning-based approaches are similar in that they require predefined knowledge, although they are completely different approaches. The difference between signature-based and supervised machine-learning-based approaches from the user's perspective is that training based on network traffic, although being labeled, is sufficient. No additional rules, configurations, etc., are necessary.

In contrast, unsupervised methods do not need labels and just find patterns that, depending on the specific algorithm, allow us to distinguish between two or more different result classes. Semisupervised approaches are "hybrid" approaches where classifiers are first fitted with a small amount of labeled data and then improved using unlabeled data. Unsupervised approaches usually have a lower classification performance than supervised methods but do not need labeled data as input and provide the ability to detect unknown attacks that do not show a similar feature pattern to any known attack.

Many experiments exist that use machine learning, and machine-learning-based anomaly and intrusion detection in OT networks seems to become a highly promising concept. Examples include the recent works of Mühlburger et al., which can detect network attacks based on traffic metadata, meaning that they can identify attacks in encrypted traffic as well. They tested the approach using the 21 features from the APG dataset, an IEC 60870-5-104 dataset, which was created based on traffic captured from a power grid substation network [13]. Their approach is an autoencoder with long short-term memory (LSTM) memory cells. Colelli et al. provide a recent publication using a Scikit-learn-based random forest applied for an IDS in a self-provided test environment. The authors pointed out the strengths of random forests: Little dependence between associated models and the training set is needed, as there will be a reduction in variance and in the classification error due to the use of an ensemble of many weak classifiers at once [14].

Specification-based intrusion detection continuously checks the state of a system by observing incoming commands and compares it against a specification-based system, such as a state machine or even a virtual replication of the system (digital twin). If unauthorized states are reached on this specification-based system, the commands are not forwarded to the real actuator; instead, an alert is thrown. One example of such a system is the approach of Carcano et al. [3].

The concept of specification-based intrusion detection typically focuses on safety-relevant limits. It has the benefit that safety breaches from complex and sophisticated security attacks can be detected, too, before any damage occurs, since it is not necessary to interpret the attack, but only its effect. Furthermore, even attacks based on hijacked authorized accounts as well as plain handling errors, leading to an unwanted state, can be mitigated.

The challenges of specification-based IDS are that the completeness and correctness of the specification are of vital importance and that all commands supported by the real system must also be supported by the virtual system. In addition, the virtual system must be permanently kept up to date.

For our work, we decided to use an anomaly-based approach, using machine-learning-based traffic classification, due to the following reasons.

1) OT traffic patterns are typically more regular [15], [16] and changes in network infrastructure occur less frequently. Therefore, it is easier to define models for normal traffic behavior.
2) As opposed to signature-based schemes, also unknown anomalies can be detected just by identifying divergent traffic patterns. This is especially valid for approaches such as autoencoders, which are trained using only operational traffic and are able to detect deviating network traffic patterns. In this case, the IDS is able to raise alerts also for anomalies that have not been caused by cyberattacks but by a failure of devices or connections, additional hosts, unusual behavior of hosts, etc.
3) They do not require the creation and permanent maintenance of a flawless specification or a virtual copy of the system.
4) If a supervised classification is applied, then automated identification of (pretrained, i.e., known) anomalies is possible without network-specific configuration, rule updates, etc.

The approach we selected is a random forest. This is a supervised machine learning approach based on anomaly detection that needs labeled data for training. For successful training, the training data need to include malicious traffic similar to the expected attacks. This approach enables us to identify anomalies and attacks similar to the expected ones, which is helpful when identifying resulting risks. To explain our concept, we use a pure supervised method in our proof of concept. However, in practice, we recommend combining

supervised and unsupervised algorithms to cope with similar as well as completely new attack patterns.

B. MACHINE-LEARNING-BASED INTRUSION DETECTION IN OT NETWORKS

Supervised approaches have been used for OT intrusion detection, e.g., by Anton et al. [17], who used SVMs and tested them in their experiments on two OT datasets: one with attacks and one with anomalies. The datasets are based on Modbus/TCP captures. Other supervised approaches for OT settings include neural networks [18], [19], [20], [21], [22], [23], [24], LSTM [20], [25], [26], [27], decision trees [21], [28], random forests [29], [30], and KNN [21], [28].

Unsupervised approaches for networks have been used, e.g., by Schuster et al. [15], who determined the potentials of One-Class SVMs (OCSVMs) for the detection of anomalies in OT traffic. Their experiments are based on real OT network traffic data, and they conclude that OCSVMs are a viable approach for OT anomaly detection. OCSVMs are also used by other authors [31], [32], [33], [34]. Other unsupervised approaches applied in OT IDS concepts have been self-organizing maps [35], autoencoders [26], [36], [37], [38], [39], [40], and clustering [41].

In general, the literature suggests that random-forest-based anomaly detectors produce good or even very good results in supervised settings [14], [17], [28], [42]. Therefore, we decided to use them in our work. Nevertheless, our concept is modular and works equally well with an unsupervised or hybrid approach (i.e., supervised and unsupervised classification combined)—especially if known attacks are still identifiable.

The novelty of our concept is the consideration of the safety-security interplay on the IDS level (cf. Section I) reactively and on the management level.

This work presents an anomaly-based IDS using a supervised machine-learning-based approach for detecting incidents. In our approach, multiclass detection is possible to predict the particular attack type, such as *portscan*, *Denial of Service (DoS)*, botnet traffic, and *benign*.

C. SAFETY-AWARE INTRUSION DETECTION

Mitchell et al. [43] developed a method for intrusion detection that combines rules and system behavior in smart grid networks, outperforming anomaly-based approaches created for the same purpose. The authors also provide a formal specification of possible unsafe states. Their concept differs from ours with regard to risk management; however, as they do not assign values to assets or safety-relevant consequences and do not suggest/prioritize countermeasures.

Wasicek et al. developed a context-aware approach to intrusion detection called CAID. Their system evaluates data on the sensor level of a car via onboard diagnosis 2 link, observing possible impacts on passenger safety. It uses a reference model created based on sensor data and evaluates the plausibility of commands for the controller's operation using an unsupervised neural network [44].

Johnson [45] evaluated the impact of intrusion detection systems themselves on OT safety in 2015, concluding with possible dangers of allowlist and denylist approaches and countermeasures to these.

D. RISK MANAGEMENT

A general risk management concept is a part of many fields, such as finance, safety engineering, health monitoring, enterprise, transportation, security, and supply chain management. For ICSs, there are various standards, frameworks, and best practices available for the safety or security needs of the industry. For instance, IEC 62443 and NIST SP 800-82 specifically target security aspects of an ICS. At the same time, IEC 61508 and ISO 12100 address only safety. IEC TR 63069 explains and guides the common applications of IEC 61508 and IEC 62443 in the area of industrial process measurement, control, and automation.

Risk management is defined as a continuous process of providing risk assessment (the process of risk identification, analysis, and evaluation) based on risk treatment options. The individual characteristics of risk management used in this article are shown in Fig. 1. The risk identification phase answers the questions: *Who/what is the risk agent?* (e.g., attackers, failure, workers), *why is the agent motivated?* (e.g., malicious intent, unintentional mishap, component efficiency), *what is at risk?* (e.g., assets, people, process, environment), *how will the attack take place?* (e.g., scanning, attacker capability), *where is the component located?*, and *when is the component going to fail?* (e.g., exploited vulnerabilities). The common methods for risk identification include brainstorming, documentation review, operation impact review, assumption analysis, Delphi technique, root cause analysis, strengths, weaknesses, opportunities, and threats analysis, and expert judgment [46].

Risk identification is followed by risk analysis. Tixier et al. [47] reviewed different risk analysis methods used in plants and highlighted their types of inputs: *Plans or diagrams, Process and reactions, Asset function and quantity, Probability, and frequency, Implemented policy, Environment, and Documentation and Historical data*. Moreover, the following types of common methodology are named in the publication: *1) deterministic; 2) probabilistic; 3) qualitative; and 4) quantitative*.

The risk assessment methods categorized as safety [e.g., fault tree analysis, failure mode and effects analysis, hazard and operability] and security (e.g., attack tree analysis, system-theoretic process analysis for security) have been in practice in research and industry. The methods (e.g., Boolean-logic-driven Markov processes, and Bayesian belief network) are required to facilitate integrated risk assessment [48], [49]. Other methods that have been used include analytic hierarchical process and Monte Carlo simulations to determine the risk value. The standards for risk management also provide some best practices for risk treatment. In [50], a defense in depth approach for securing ICS is mentioned. The *as low as reasonably practical (ALARP)* principle is a common risk-reduction principle, which is based on risk-informed

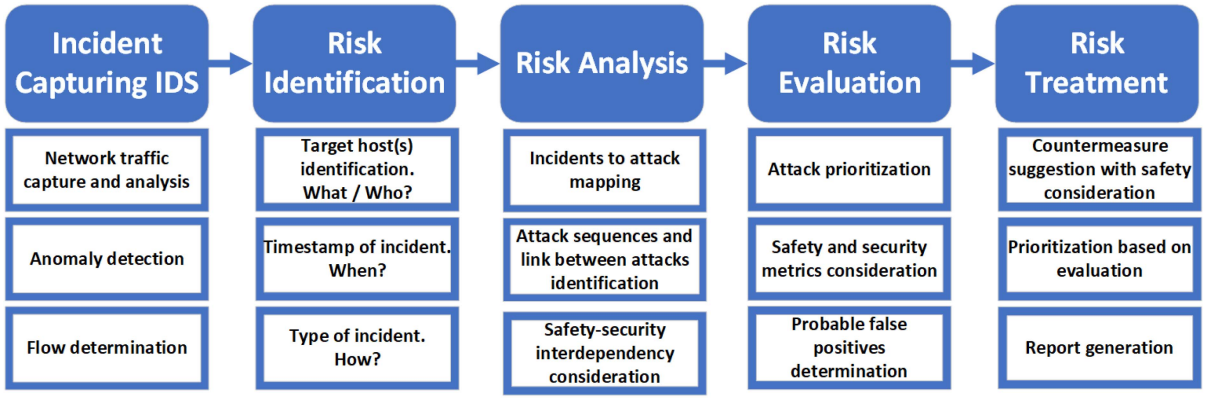


FIGURE 1. NIDS-based risk management approach.

and cautionary/precautionary thinking [51]. Each mentioned risk assessment method is a proactive measure and, therefore, considers risks before their occurrence. In contrast, our introduced risk assessment method is reactive due to the self-developed IDS, which enables the live reaction to attacks occurring during the operation of the ICS network.

A network-based intrusion detection system (NIDS) is a security measure and has been in use for risk assessment for security analysis based on attack patterns [52]. An approach mentioned in [53] proposes reaction selection, quantifying effectiveness, and providing minimum side effects of measures in [54] while assessing attacker skill and knowledge. In [55], a taxonomy of intrusion response systems is proposed. The article furthermore explains the challenges in its development. In our approach, the patterns are used to recognize the attacks and possible targets. The risk evaluation consists of the security and safety impact of a successful attack. The risk treatment has not only security measures but also safety measures. In this article, an IDS is tested for assessing the system's safety in case of a security attack.

III. CONCEPT

As described in the introduction (cf. Section I), the interplay between safety and security measures and threats introduces a challenge. The idea behind this work is to consider the safety consequences of threats and security measures when deciding on priorities of alerts and countermeasures. Therefore, the IDS is no longer an isolated security appliance that alerts solely security-related incidents and lists them in chronological order but also takes over a strategic job to assist factory operators in decision-making, including both safety and security implications. It highlights the mutual reinforcement relation (cf. Section I) between safety and security, where a security measure enhances the safety of the system. Therefore, the goal is to create a risk management method built on the results provided by a developed IDS and addresses the following objective.

Given the ability to find and identify security incidents, find and identify anomalies that may have an effect on the safety at the local premises.

TABLE 1. Predefined Asset Value Table, Stored in the CSV File

Device name	IP address	Device description	Device asset value
Firewall	172.16.1.1	Firewall	5
MaxxTurn45 NCU	172.16.1.70	NCU	9
MaxxTurn45 PCU	172.16.1.50	PCU	9
Experiment Host 1	172.16.1.20	Initial attacker	6
Experiment Host 2	172.16.1.11	Infected host	6
Experiment Host 3	172.16.1.32	Thin client	3
Experiment Host 4	172.16.1.33	Thin client	3

In addition to detecting attacks, the following takes into account:

1) possible security and safety-related consequences of the corresponding attack;
2) possible security and safety-related consequences of mitigation measures;
3) priorities of countermeasures, depending on the asset's value.

For this, parameters about the value of the asset and dependencies need to be predefined and stored in a knowledge database. Hence, a link is created between the following:

1) asset;
2) value of asset;
3) consequences if an asset is attacked;
4) consequences of countermeasures.

Our approach to tackling this challenge is the following: Assets are stored as predefined knowledge in comma separated values (CSV) files (cf. Table 1 for an example). Their value for the ICS is stored in two dimensions: The safety integrity level (SIL), which defines the severity of the consequences of failure from a safety point of view, and the operational criticality (OC), which is a measure of the importance of this OT component for the operation from a security and operation point of view. SIL is a metric defined by the safety standard IEC 61508. A safety risk assessment (e.g., according to the standard ISO 12100) is required before a SIL may be applied to a dedicated OT component to address the identification of hazards, estimation, and evaluation of safety risks in the phases throughout the machine life cycle, and the suggestion of either eliminating hazards or sufficient risk reduction. The identified safety risks are evaluated using EN

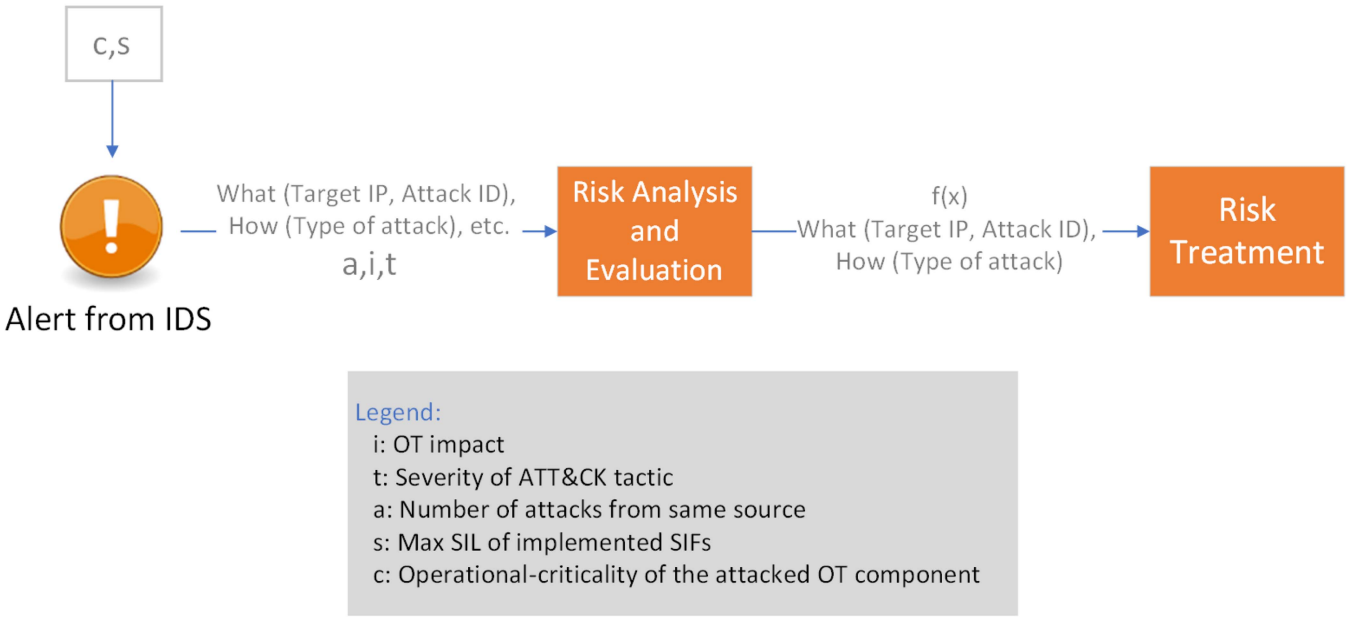


FIGURE 2. NIDS consists of three main components, each serving a distinct objective. The components exchange information as depicted.

TABLE 2. Relation Between PL and SIL Based on the Standard EN ISO 13849

PL	PFH $_D$ (Probability of Dangerous Failure per Hour)	SIL
a	$10^{-5} \leq$ PFH_D $< 10^{-4}$	None
b	$3 \cdot 10^{-6} \leq$ PFH_D $< 10^{-5}$	1
c	$10^{-6} \leq$ PFH_D $< 3 \times 10^{-6}$	1
d	$10^{-7} \leq$ PFH_D $< 10^{-6}$	2
e	$10^{-8} \leq$ PFH_D $< 10^{-7}$	3

ISO 13849, based on the results of the safety risk assessment based on ISO 12100. Each identified safety risk is rated with a performance level (PL), which can be mapped to a SIL, as Table 2 shows.

The OC is a direct proportional numerical value representing the importance of the component w.r.t. the whole industrial architecture. The OC may be applied to all components in the industrial architecture, including network devices.

Fig. 2 depicts the interplay of the three main components of the IDS, which are the detection of incidents, risk assessment, and risk treatment. Beyond that, the graphic shows the information that is stored as predefined knowledge (c, s) and the information that the components deliver to each other [a, i, t, $f(x)$], and information about the incident that the IDS component obtained. The predefined information described in Table 1 is accessible by the first of the three components, incident detection, and identification. Therefore, an alert created by this component comprises information extracted from network data and also the predefined knowledge, e.g., according to the addresses and ports directly obtained from the network traffic. This information is then forwarded to the risk assessment component, where the priority value is calculated. The resulting priority value $f(x)$ plus additional incident information, such as source and destination host and timestamp, is then passed to the risk treatment component, which uses predefined knowledge to infer the best countermeasures and their priority from the given information.

IV. USE CASE DESCRIPTION

A. NETWORK SETUP

Our setup for attack traffic generation is integrated into a real prototype factory in Austria. The factory network is divided into network segments. One of these segments is the turning segment, in which we conducted the experiments. Fig. 4 shows the network setup. This setup consists of a turning machine with a process control unit (PCU)/numeric control unit (NCU) pair. Attached to the turning machine, there are three power sensors that continuously monitor voltage, current, and power consumption and send these values continuously over the network via Modbus Transmission Control Protocol (TCP). The black hosts were added for the experiment while the gray hosts are part of the productive operation.

Safety-relevant aspects of this particular Proof-of-Concept (PoC) network are, for example, these power sensors, as wrong signals (e.g., too high current or too low voltage) may lead to wrong control decisions (e.g., halt production, cut electricity supply, etc.). The PCU and NCU of the turning machine may both influence production safety as well, as both devices can control the turning machine (the PCU indirectly, the NCU directly). Even the security of the hosts in the network is relevant for the safety of the production: All hosts with access to the storage of the PCU—be it a file server or direct access to the PCU's local file storage—can influence the production, and, in the worst case also the PCUs security, since

TABLE 3. Attacks Executed in Real OT Environment

Attack ID	Attack category	Software
A1,A2	Portscan	nmap
A6	DoS	hping3
A5	Remoteshell	Metasploit
A3, A4	Botnet	self-developed

transferred engineering artifacts may contain malware as well (cf. Fig. 4).

The PCU and NCU are assigned with the highest OC in this setup since their failure (or compromise) may stop the operation. The infected host and the initial attacker host have a network connection to the valuable assets PCU and NCU and have, therefore, also increased importance. A low OC is assigned to the firewall since a successful attack would not directly influence the production process. The thin client in this network with limited functionality has the lowest OC. Table 1 also shows the assigned OC as *deviceAssetValue*.

B. ATTACK USE CASE

We set up an IDS in a PoC setup with several example attacks that have implications for safety: In this setup, two types of attacks are generated and recorded (cf. Fig. 4 and Table 3).

The attacks belong to two different types of attacks: The first type is *simple* attacks. For this, we create four attacks: A1, A2, A4, and A5. Simple attacks always have their origin at a single attacker host located in the local network and are directed at one victim host that is also located in the local network. Furthermore, these attacks use standard software. In Fig. 4, the attacker is named *Host 1*, and the victim is named *Host 2*.

In Table 3, the tools are listed that were used for the creation of attack traffic. The second type of attack is a complex botnet attack (A3). It is more complex, as it involves several hosts that are not all located within the local network but can attack from the wide area network (WAN). In addition, the attacker uses a communication channel that tries to conceal both the presence and the content of its communication. A simple botnet simulator has been developed by the authors to conduct this attack. It simulates lateral spread of botnet malware from Host 1 to Hosts 2, 3, and 4 via the SMBGhost exploit,[1] information exfiltration of a confidential document to a host on the Internet (referred to as attacker's storage host) and a DoS attack on the factory network's NCU in this order. The attacker (located in the WAN) controls the Command and Control Center (CnC), which, in turn, controls the infected computers (bots) in the botnet. All simple attacks use only plain text traffic while the more complex attack (A3) relies almost entirely on encrypted traffic that is based on hypertext transport protocol secure, a protocol that is widely used and also deployed in our ICS during normal factory operation. When performing data exfiltration, the botnet simulator compresses the document first and splits it into chunks of less than 200 kb, with the goal of reducing peaks in the network traffic volume.

Fig. 3 shows the data (packets and bytes per minute) exchanged over time for the simple attacks and the complex botnet attack. As expected, the OT traffic shows a high regularity in the time series as the amount of exchanged bytes and packets every minute are similar. The simple DoS attack can be easily identified as it causes a peak in packets that occurred during the early afternoon.

Looking at the bar plot for the complex attack, it can be observed that despite the efforts to reduce peaks in the network traffic, there are still peaks visible in the number of exchanged packets and bytes during the attack.

C. DATA CAPTURE AND ANALYSIS

A monitoring port on the core switch of this network segment has been configured so that the entire traffic is forwarded to the experimental IDS. A data acquisition and generation (DAG) server using Endace hardware (cf. [56]) has been deployed to capture all traffic from this monitoring port of the switch. The DAG server captures all frames in a lossless manner and with microsecond-accurate timestamps using a PPS (pulse-per-second) signal from an external GPS antenna located outside the building. This setup is created to meet real-time requirements as well as possible. Due to the off-path analysis of the IDS, a delay in the size of the configured IDS time window (in our case: 60 s) is introduced to all incident alerts (cf. Section IV-C). The data are captured between the supervisory control and data acquisition (SCADA) and manufacturing execution system (MES) level since the connection between SCADA and programmable logic controller (PLC) is time critical and often uses proprietary (nonroutable) communication protocols such as Siemens S7 communication. The data collected from the experiments described in Section IV-B comprise operational factory traffic and attack traffic and have a total size of 18.1 Gigabyte (GB) for the timeframe containing the simple attack experiments 31.5 GB of traffic for the complex attack (both stored in packet capture (PCAP) format), making up a total dataset of 49.1 GB. The displayed time frames are about 15 h (09:10 A.M. to midnight local time) for the simple attacks and a full 24-h block (midnight to midnight) of network traffic for the more complex (botnet) attack. Therefore, both captures contain a significant amount of benign operational traffic as well. Unfortunately, we are not allowed to publish these data due to factory policy.

In live operation, the IDS captures and analyzes network data simultaneously and off-path. Off-path means that one logical process on the IDS continuously captures data and splits it into chunks of a certain configured window size. Another logical process on the same machine opens these captured data files once they are complete and extract flows (cf. Section IV-D), etc., for analysis. These extracted flows are then test data fed to the machine-learning-based IDS. Incidents that are found are forwarded, together with their

[1][Online]. Available: https://msrc.microsoft.com/update-guide/vulnerability/CVE-2020-0796

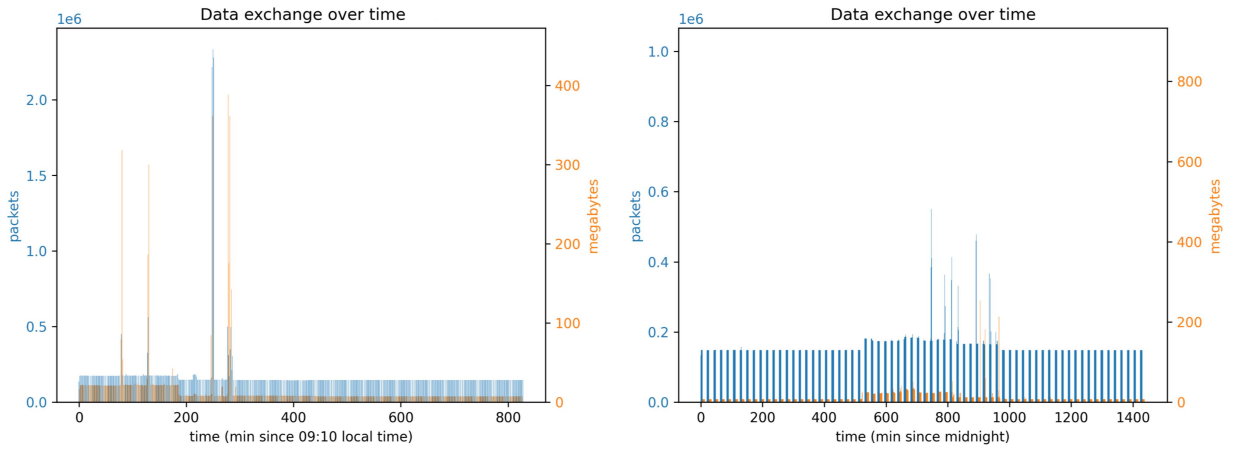


FIGURE 3. Total exchanged packets and megabytes in the network over the time frames of the experiments. Left: Simple attacks. Right: Botnet attack.

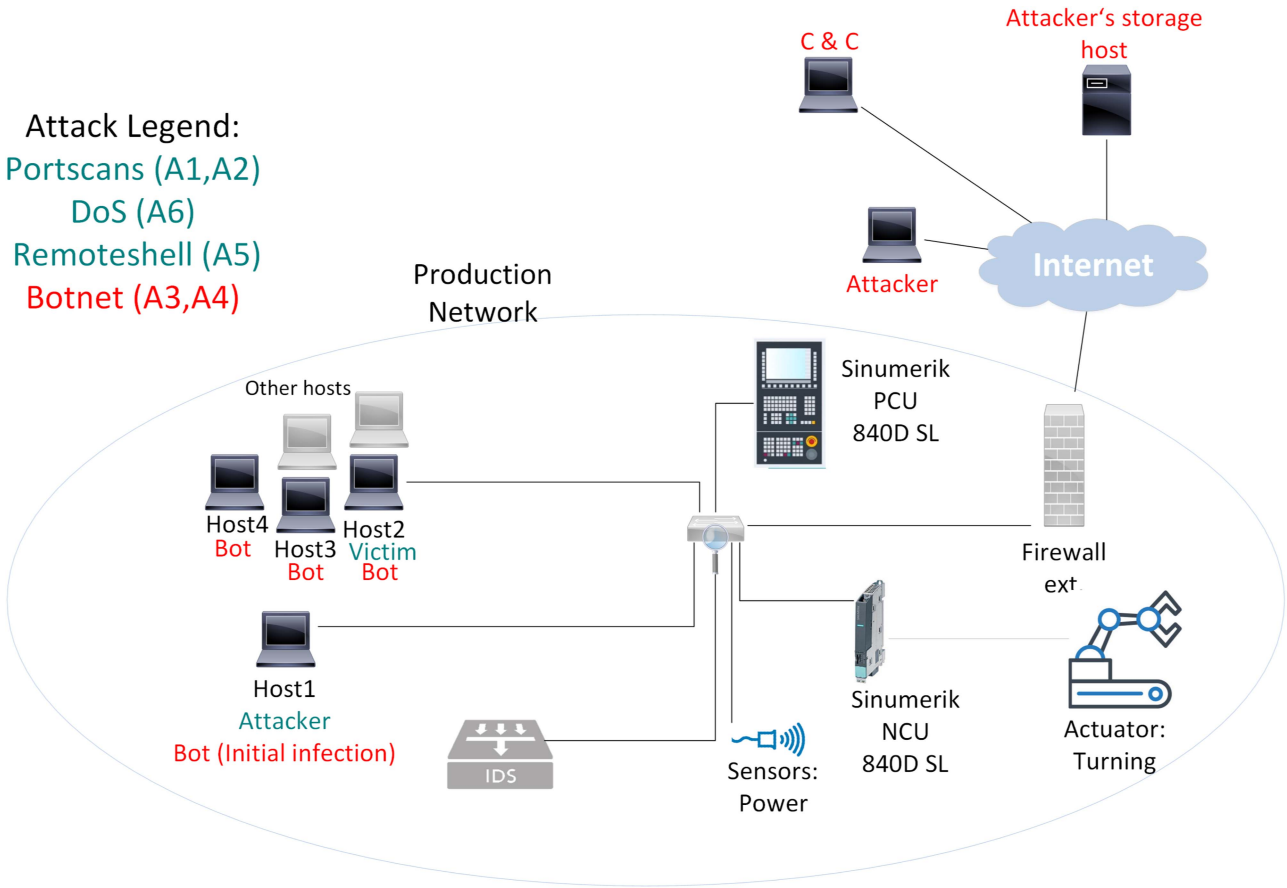


FIGURE 4. Scenario use case. Black connections are Ethernet connections, and gray connections are control connections.

determined context (internet protocol (IP) address, time, etc.), to the higher-level analysis. This approach has two major benefits but also one major drawback: On the one hand, no traffic is overseen by the IDS, and real-time constraints are met, as the delivery of packets is not delayed. On the other hand, alerts are delivered with a certain delay that is determined by the sum of the time window size and required analysis time. This introduces a tradeoff for the time window size: While a short time window introduces shorter delays to incident alerts, a longer time window ensures a more complete analysis of network flows since network flows have a different duration and incompletely captured flows, which are then split into

several smaller flows, may influence the detection of security incidents. Currently, our PoC implementation supports the detection of four attack types. If an incident occurs that differs a lot from those attack types, it can be wrongly classified: If it is wrongly classified as benign traffic, we get a false negative. If it is classified as one of the four supported attack types, it still raises an alarm. However, in our safety NIDS, it may lead to unsuitable suggestions for the risk treatment.

D. DATA PROCESSING

The captured network traffic is extracted into CSV files containing flows. A flow is an aggregation of packets that share a common set of values: The flow key. The flow key, in our case, is the classical 5-tuple defined by source and destination IP address, source, and destination port number, as well as protocol identifier. Using the flow extractor *go-flows* [57], a combination of the following metadata-based feature sets have been extracted (cf. [58], [59], [60], [61]): *CAIA, AGM, TA,* and *Consensus*.[2] This way, a large union feature set consisting of 146 features was created. Source and destination port numbers, addresses, and timestamps were removed prior to classifier training and testing to not bias the IDS to specific sources and destinations. Also, removing particular features is sometimes necessary, for example, for the evaluation of the DoS attack: Since this attack turned out to be the only source to use packets of size 160 B. Since it would be easy for an attacker to adjust this to a more commonly used packet size in this environment, we removed all features related to packet sizes in order not to create bias. Nevertheless, especially the DoS attacks are still easy to identify using round-trip times, off-times (i.e., consecutive milliseconds without packets being sent), and the number of TCP SYN requests within a certain time span, for example. The IDS, once trained, are tested the same way, using labeled test data taken from the same experiment from a different point in time.

E. ASSUMPTIONS

For this scenario, we assume that all basic security measures, such as deploying suitable security policies with secure passwords and appropriate firewall rules, are taken at the factory premises and that the network is secured according to state-of-the-art security methods. We assume that all hosts and nodes have the most recent software updates installed. For the attacker, we assume a skilled attacker with medium to high resources, such as a competitor (i.e., competing company) level attacker or above according to the work in [62].

V. METHOD

A. RISK IDENTIFICATION AND ANALYSIS

Fig. 1 depicts our method beginning with the detection of a network incident (left side of figure). The IDSs output after detecting a suspicious event includes host addresses, type of attack, etc., and serves as an input for the application of a self-developed threat modeling approach [63]. This threat modeling approach includes various characteristics of the underlying system and its OT components, for instance, implemented safety functions of an OT component and the corresponding hazards to counter. Based on this knowledge, multiple hazards caused by one successful attack may be identified.

Risk identification is performed using the introduced IDS in this work, which identifies the type of attack (e.g., botnet, scan, etc.). *Risk analysis* is implemented by using the MITRE ATT&CK framework [64] on the results derived from *risk identification*. In this phase, the perception of the analyzed system and the indicators of the identified potential incidents are analyzed and compared to known tactic, technique, and procedures (TTPs). The perception of the analyzed system consists of the identified hosts, the function of the hosts, network layout, and network communication. This comparison maps incidents to attacks or phases of attacks. For instance, when the incidents "botnet" and "portscan" are identified, and the portscan originates from the hosts belonging to the botnet, this may also indicate lateral movement or preparation to launch further attacks. Furthermore, this perception of the analyzed system and the comparison of the incidents with known TTPs serves as a means to determine potential false positives. For example, if the network communication behavior indicates a potential DoS attack due to the high amount of sending requests, but the source of the requests is a sensor, this event may be the intended update cycle of sensor parameters instead of an attack. False positives can furthermore be determined using predefined knowledge, such as additional lists of allowed incidents.

B. RISK EVALUATION

Risk evaluation defines the criticality of the identified attacks using an adapted evaluation scheme based on [65]. The criticality-rating of the attacks is evaluated according to (1) based on the work in [63] using the following variables:

1) $f(x)$ is the resulting priority after applying the formula;
2) x is the attack (range: $0 - \$attackCount$);
3) i is the OT impact of the attack (range: 1–6);
4) t is the severity of the attack based on MITRE ATT&CK tactic [64] (range: 1–11);
5) a is the number of attacks from the same source (range: $1 - \$attackCount$);
6) s is the maximum SIL of all safety instrumented functions (SIFs) of the OT component if implemented (range: 1–4);
7) c is the operational-criticality of the OT component (range: 1–9);

$$f(x) = w_i \cdot \text{scale}(i_x) + w_t \cdot \text{scale}(t_x) + w_a \cdot \text{scale}(a_x) + w_s \cdot \text{scale}(s_x) + w_c \cdot \text{scale}(c_x). \quad (1)$$

[2]Note that the AGM vector uses a different flow key. Therefore, you can either join the extracted features in the aftermath or use just the compatible features from the AGM, which is what we did.

TABLE 4. Definition OT Security Impact

Severity	Violated security protection goal	CVSS based impact
6	Availability	High
5	Integrity	High
4	Availability	Low
3	Confidentiality	High
2	Integrity	Low
1	Confidentiality	Low

TABLE 5. Definition ATT&CK Tactic Severity

Severity	ATT&CK tactic
11	Command and Control
10	Inhibit Response Function
9	Impair Process Control
8	Lateral Movement
7	Execution
6	Privilege Escalation
5	Initial Access
4	Persistence
3	Evasion
2	Collection
1	Discovery

Our adapted evaluation only considers attacks that are already running. It does not consider possible underlying threats that may lead to attacks since this work focuses on already launched attacks, resulting in suggesting the use of case-tailored reactive countermeasures only in the phase *risk treatment*.

One of the newly introduced risk evaluation factors is the OT impact of the attack. This metric considers the different priorities of the security protection goals (confidentiality, integrity, and availability) in the OT domain. The impact of each security protection goal is measured using Common Vulnerability Scoring System Version 3.1 (CVSS). There are different conceptions of which security goal has the highest importance in the OT domain: Either availability [66] or integrity [67] is considered as most important. However, the mentioned authors of [66] and [67] agree that confidentiality is the least important security protection goal. This work considers availability as the security goal with the highest importance in the OT domain and defines the OT security impact according to Table 4. The resulting OT security impact value is the sum of the violation of security protection goals caused by the attack.

Another newly introduced risk evaluation factor is the attack's severity according to the attack's corresponding MITRE ATT&CK tactic [64]. Each identified attack is linked to a suitable ATT&CK technique which is a subset of an ATT&CK tactic. Each ATT&CK tactic is assigned a severity value based on the concept of OT security impact, as Table 5 lists. If several techniques can apply to a specific incident, the corresponding tactic with the highest severity score is used for calculation. For example, *Command and Control* is considered the most severe ATT&CK tactic since it enables remote control of assets in the OT domain, impacting all three protection goals. Again, availability is rated by the highest severity scores followed by integrity and confidentiality as well in Table 5.

TABLE 6. Mapping Quantitative to Qualitative Risk Values

Quantitative value range	Qualitative risk value
$\frac{4}{5} \cdot \sum_k w_k \leq f(x) \leq \sum_k w_k$	Major
$\frac{3}{5} \cdot \sum_k w_k \leq f(x) < \frac{4}{5} \cdot \sum_k w_k$	High
$\frac{2}{5} \cdot \sum_k w_k \leq f(x) < \frac{3}{5} \cdot \sum_k w_k$	Medium
$\frac{1}{5} \cdot \sum_k w_k \leq f(x) < \frac{2}{5} \cdot \sum_k w_k$	Low
$0 \leq f(x) \leq \frac{1}{5} \cdot \sum_k w_k$	Negligible

One factor in (1) is the number of attacks (a) from the same source. This factor indicates a compromised asset.

One factor used for risk evaluation is the maximum SIL (s) of all SIFs of the attacked OT component. An OT component implementing safety functions [e.g., PLCs, safety instrumented systems] may have several SIFs implemented. Since all functions, including SIFs, may be compromised after a successful attack against an OT component, the SIL of the SIF with the highest SIL is taken into consideration for risk evaluation.

The factor of operational criticality is described in Section III.

All factors used for risk evaluation are *scaled* to normalize the resulting values using the function $(X - X.\min)/(X.\max - X.\min)$. The importance of each factor can be increased or decreased by modifying the corresponding w-variables. For instance, w_i modifies the importance of OT security impact, or w_t modifies the importance of the severity according to the MITRE ATT&CK tactic. With this, administrators can tune the system to different priorities.

The quantitative result of (1) can be mapped to a qualitative risk rating using the ranges listed in Table 6.

C. RISK TREATMENT

The risk treatment consists of four different routes, namely: 1) mitigation, 2) avoidance, 3) transfer, and 4) acceptance. Depending on the measures that are usually in place in the organization, one or more of the four routes are considered. In this article, we mainly focus on the mitigation route based on the risk identification and assessment result. The choice of a route depends on factors such as the economic aspect, technical capability, available tools, nature of the attack, and third-party contracts of the organization. Accepting the risk is one of the basic principles of the ALARP method. For example, a portscan against assets with low operational criticality and without safety relevance may be accepted. However, a scan against a critical asset (according to the introduced metric) is to be mitigated because it can lead to targeted attacks resulting in severe consequences (cf. Section VI for a more specific example). The risk mitigation strategies implemented are limited to the security attacks that have an impact on the

TABLE 7. Excerpt of Identified Attacks applying [64]

A-ID	Incident	Possible ATT&CK Techniques	Possible ATT&CK Tactics	Interdependency addressed
A1	Portscan	Remote system information discovery	Discovery	-
A2	Portscan	Remote system information discovery	Discovery	Activity may cause a DoS on the target
A3	Botnet	Lateral tool transfer Exploitation of remote services Exploitation of privilege escalation	Lateral movement Privilege escalation Initial access	Activity may cause a DoS on the target
A4	Botnet	Standard application layer protocol Connection proxy	Command and control	Connected safety-relevant devices may be manipulated
A5	Remoteshell	Command-line interface	Execution	Connected safety-relevant devices may be manipulated
A6	DoS	Denial of Service	Inhibit response function	Execution of safety function may be denied

safety of the system so as to address the interdependency. Therefore, countermeasures based on both industrial security standards (e.g., IEC 62443) and safety standards (e.g., IEC 61058) are provided. The mitigation of such attacks would be to enhance the capability of the firewall or IDS itself, as the IDS in use works best in the current scenario.

The MITRE ATT&CK framework [64] used for the evaluation of the attacks for possible techniques and tactics also provides some mitigation strategies for the used technique and tactic. Other than that, there are online databases maintained such as ICS-CERT,[3] and NIST,[4] which also provide useful information related to the components of risks and guidance for risk management. In most cases, expert opinion and brainstorming are also implemented as a preliminary mitigation strategy. This might help the asset owner to implement strategies to avoid or control the ill effects caused due to the attacks.

VI. DISCUSSION OF RESULTS

The experiments in the pilot factory yielded an authentic experimental OT dataset (cf. Figs. 3 and 4). We use these data for a use case on which the whole concept is applied—from incident detection and identification, over risk assessment to risk mitigation (i.e., suggestion of countermeasures). Fig. 6 shows the confusion metrics of the traffic classification. It has been created with the full data of our experiments. These data were split in a stratified manner into 80% training data and 20% test data. On the X-axis of the confusion matrix, there is the predicted label for each of the observed traffic flows. On the Y-axis, there is the true label for the corresponding flow (i.e., the ground truth). Given the performance in the confusion matrix, we interpret that a higher number of remoteshell traffic would have probably led to higher detection performance. Nevertheless, it was possible to use these data for a simulated use case with all components involved:

Applying the self-developed threat modeling approach [63] to the results from our experiments, the IDS identifies the attacks listed in Table 7. The first observed attack, A1, is a portscan conducted initially to gain information about a dedicated host in the production network. Since this attack was performed on a nonsafety-relevant asset that is not connected to a safety-relevant asset, no interdependencies between safety and security are involved. This attack is linked to the ATT&CK technique remote system information discovery, which is part of the tactic discovery. A2 is another portscan but, this time, aimed against Host 1, which is connected to the PCU, a safety-relevant asset since it controls the production of the turning machine. This attack is linked to the ATT&CK technique remote system information discovery, which is part of the tactic discovery. Depending on the targeted OT component's robustness, a portscan may already lead to a DoS on the target. Using attack A2, the adversary identified a more attractive target for further attacks than by using attack A1. He will then launch targeted attacks on the corresponding target. A3 describes the next attack: The creation of a botnet. The initial infection of Host 1 may have been introduced via a USB drive with malware on it. Since the IDS is not able to detect this initial attack vector of the incident, it is not part of the corresponding attack sequence in Table 7. However, when the botnet software starts to spread over the network, it may be identified by the IDS. The botnet software may also spread to the PCU, which enables the manipulation of the application logic of the NCU to interfere with safety-relevant functions. From the perspective of an IDS, it is not clearly distinguishable if the technique "lateral tool transfer," "exploitation of remote service," or "exploitation of privilege escalation" caused this incident. Therefore, the tactics "lateral movement," "privilege escalation," and "initial access" may be applicable. A4 is the second phase of the botnet attack, where a connection to a CnC server is established. From the perspective of an IDS, it is not clearly distinguishable if the technique "standard application layer protocol" or "connection proxy" is used for this incident. Both techniques belong to the tactic "Command and Control." A5 describes the attempt to spawn a privileged remoteshell on the NCU to manipulate the configuration and behavior of the NCU directly. After a failed attempt to spawn the desired remoteshell, the adversary decides with A6 to launch a DoS attack against the NCU to delay or abort potential executions of the safety function.

Table 8 presents the results when applying the proposed adapted evaluation scheme to the attacks listed in Table 7. The columns of this table are defined as follows:

1) *A-ID:* ID of the identified attack;
2) *CVSS vector:* The overall CVSS vector of the attack;
3) *CVSS base score:* The CVSS base score of the attack;

[3][Online]. Available: https://www.cisa.gov/uscert/ics/advisories
[4][Online]. Available: https://www.nist.gov/

TABLE 8. Evaluation of Identified Attacks

A-ID	CVSS vector	OT impact	Max Severity ATT&CK tactic	# attacks from source	Max SIL of OT component's SIFs	OC	Priority f(x)
A1	CVSS:3.1/AV:A/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N	1	1	1	-	3	Negligible
A2	CVSS:3.1/AV:A/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:H	7	10	1	2	9	High
A3	CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:C/C:L/I:L/A:L	7	8	2	2	9	High
A4	CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:C/C:H/I:H/A:H	14	11	3	2	9	Major
A5	CVSS:3.1/AV:A/AC:H/PR:L/UI:R/S:U/C:L/I:L/A:H	3	7	4	2	9	Medium
A6	CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H	6	11	5	2	9	High

4) *Mapped score:* Resulting score after performing the mapping of CVSS to security level (SL) according to the work in [63];
5) *Attack SL:* Classification of the skill level and resources needed to successfully launch this attack. This value is a result of the mapping introduced in [63];
6) *OT component SIL:* Increase of the potential impact of the resulting risk of the attack (as an SIL increase) if a safety-relevant OT component is affected during the attack;
7) *OC:* The assigned OC to the attacked OT component;
8) Resulting priority considering (1).

The risk evaluation suggests that A1 has the lowest priority, negligible, since the attack was performed against a nonsafety-relevant and nonoperational-critical OT component. Furthermore, there is only a low impact against the security protection goal "confidentiality." A2 has a high priority assigned since a portscan against a legacy safety-relevant component implementing SIFs may result in a DoS situation, where the OT component is overwhelmed by the requests, impacting the availability strongly. The consequence could be, e.g., the inability of this component to be regularly shut down or to react to control commands, such as movements, speed, force, angles, etc., with a very wide spectrum of possible production and safety-critical consequences. Imagine drops of workpieces from a conveyor belt, for example, since no signals are received from light barriers anymore—and the conveyor belt, therefore, does not stop in time.

A3 also results in high priority due to the high availability loss of the safety-relevant OT component. A4 has the highest priority rating, major, because of the loss of availability, integrity, and confidentiality when being able to remote control OT components connected to safety-relevant OT components. A5 has a medium priority since spawning a remoteshell in the context of an unprivileged user may not necessarily lead to the possibility of interfering with the OT component's availability and integrity. For example, stopping a service or changing config-/log-files may require additional privileges. Since A6 has similar consequences as A2, A6 has the same priority.

The risk treatment addresses the primary decision factors, namely *incidents, attack target, attack origin, priority, and false positives* to provide suggestions. Fig. 5 visually explains how these decision factors' input (shown on the left) is used to reach a decision of whether mitigation should be provided for the risk or simply accepting it would be a better choice. False positives (alert is generated even though there is no threat) are considered as an error in detection. The false positives are not assessed in order to save resources. However, one needs a mechanism to filter them out. To identify whether an alert is a false positive, system understanding is required and can only be done after sufficient alerts are analyzed by the user. Beyond that, active filters can be introduced as a mechanism to filter them. If the detected alert is a false positive, the process stops, and the incident is recorded and reported by the user. The next decision is to identify whether the incident or attack is known or not. If the attack is unknown, the incident is recorded for further analysis. Alternatively, priority calculation (cf. Section I) may still be applied with values for "x," "i," and "t" determined based on other factors.

Furthermore, the decision-making steps analyze the attack pattern knowledge. If the attack pattern is known, further analysis based on priority is undertaken. If the priority is identified to be negligible, i.e., the impact of the risk on the system or component is negligible, then the risk is simply accepted. If the priority is classified as one of {low, medium, high, major}, then countermeasures are suggested. If there is some anomaly w.r.t. priority and it does not fall in any of the categories of defined priority, then the risk is accepted, and the result is recorded for further analysis.

A1 is assessed to be of *negligible* priority due to no or very low impact on the safety and only low security impact. Furthermore, the target of the attack has a low operational-criticality. Therefore, the suggested risk treatment for this attack is its acceptance. *A2* has a high priority since it is executed against a safety-relevant and operation-critical asset. It is easy to launch (leading to a high probability) and eases the successful execution of further targeted attacks. Mitigation may be achieved by following the least functionality principle and configuration of the remaining services to be as nonverbose as possible. Including the host in an IP blacklist when the IDS identifies a portscan attempt from an attacker would counter the attack during its execution. The mitigation policy would be to perform penetration tests to determine exploitable vulnerabilities, limit information (shutdown unnecessary services), and add an IDS. *A3* and *A4* are both the same type of attack, but according to the risk assessment results, A3 and A4 possess different priorities. Therefore, similar mitigation techniques help to encounter both A3 and A4. One of the security risk mitigation applicable here is strong authentication and authorization. The victim's hosts, especially the PCU and NCU, should have this mitigation strategy implemented. Depending on the performance requirement, encryption of the communication is also a valid mitigation strategy to avoid heavy impact. Implementing similar mitigation strategies as *A3* and *A4* to *A5* will also help to control or avoid the

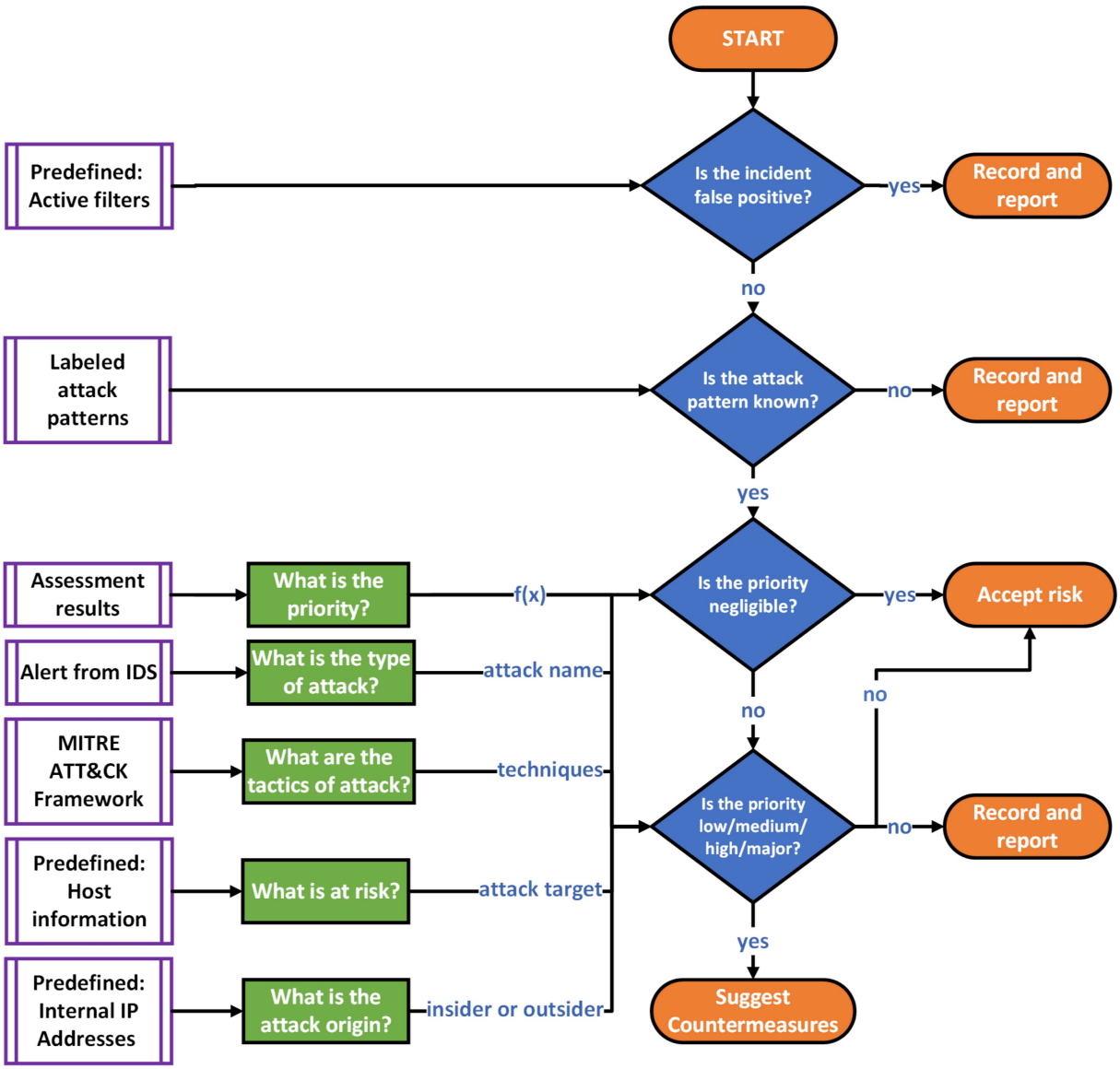


FIGURE 5. Conceptual risk treatment suggestion flow.

attack impacts. For *A5* and *A6*, permitting only the authorized IPs addresses to access the PCU and NCU and limiting requests from a host identified as malicious would mitigate the attack being performed on them. This would, however, require a network setup with packet filtering implemented, e.g., through virtual LANs (VLANs) and a firewall. *A6* may also be countered by introducing rate limiting that blocks requests if the limit is exceeded. In the case of *A3*, from the perspective of safety, the regular up-to-date backup creates a possibility of getting back lost data and configuration. A backup also has a positive impact in countering the effects of *A6* as a reactive measure, along with necessary reliability and robustness tests on the system. The authentication and authorization act as mutual reinforcement to the safety system as authentic illegal login with privileges can alter the processing logic of the PCU and NCU. The combination of the safety measure redundancy (e.g., adding the same safety-relevant and operation-critical OT component twice and activating it if the first fails), a security measure to block IP addresses to prevent further attacks from the compromised host, and safety/security measure backup to revert the compromised OT asset back to its precompromised state as a reactive set of countermeasures addresses *A3*, *A4,* and *A6*. The risk management report would consist of the attack host identification, attack definition, assessed risk with its priority, and mitigation suggested based on the above points. As the system suggests countermeasures and does not necessarily implement them, the report can serve as a good source of useful information for the risk manager or user. The countermeasures proposed may also have an unintentional impact on system operation.

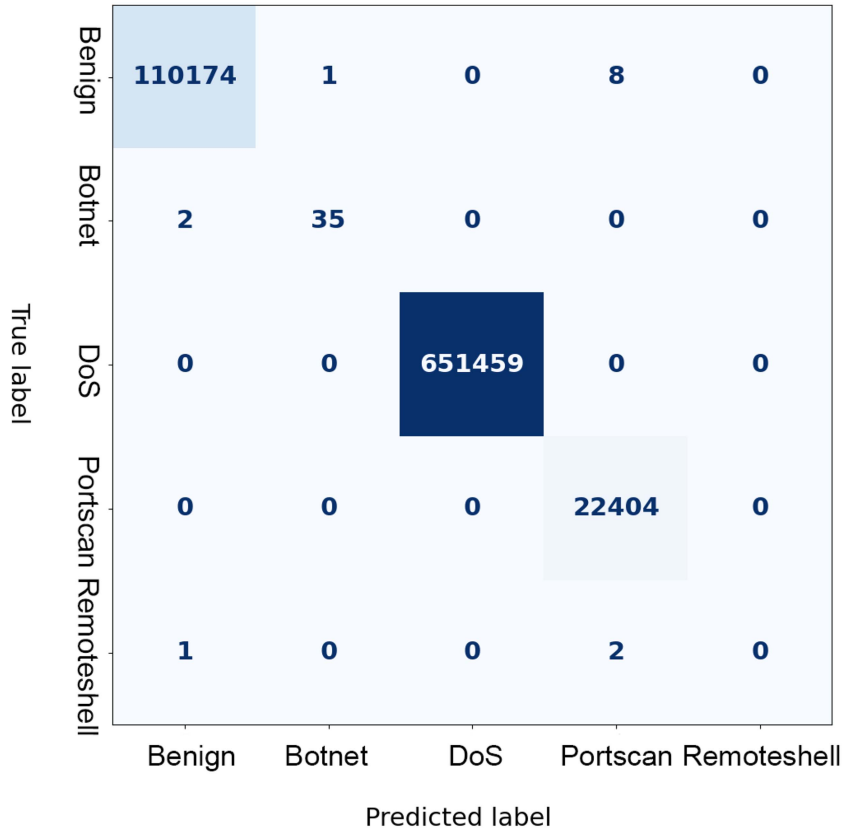


FIGURE 6. **Confusion matrix of the random forest classifier, showing the detection and identification performance for all available attack types.**

Isolating certain hosts from the ICS network, for example, also limits their productive use. The consequences observed with strong authentication and authorization increase response times. Regular backups as a strategy for data safety impose security challenges, infrastructure, efficiency, and cost. Nevertheless, not implementing the countermeasures would have a larger negative impact on the ICS. The risk treatment strategy is suggestive in nature, and hence, only with continuous risk management, one can identify the actual consequence of these suggestions on the system.

Fig. 6 shows the classification performance of the incident detection and identification module evaluated in our pilot factory network. This graph has been created by extracting the 146 chosen features (cf. Section IV-D) from the available 49.1 GB of experiment data, splitting that data into 80% training data and 20% test data, training the random forest classifier with the labeled training data and testing with the remaining 20%. The data have been split in a stratified way, i.e., keeping the ratios between each of the classes the same as in the original dataset. Note that, especially for the remoteshell attack, only very few flows were available. It is possible that classification performance would have been better if more attack data had been created.

Referring to the DoS attack (cf. Section IV-D), the result was suspiciously good, and we assumed the existence of a bias. However, it was, in fact, very easy to detect due to remarkable, yet DoS-typical properties (especially off-times between packets), and so a 100% detection accuracy could be reached. The graphic furthermore shows the limited identification capability regarding the metasploit-based remoteshell: Although the system was able to identify the attack flows as malicious, it misidentified them as "Portscan" traffic.

Information from IDS alerts (i.e., *attacks* targets, meta-information), from predefined knowledge (i.e., whether the attack is from inside or outside the local network), and from the assessment [i.e., priority $f(x)$] as well as the target's functional relevance as provided in Table 1 are used to determine the treatment technique. If the priority is in the negligible range (cf. Table 6), the risk is accepted. Otherwise, the mitigation will be suggested according to the *attack ID*, *framework identified tactics and techniques* (cf. Table 7, *attack target and origin*. A single mitigation strategy might address one or more attacks. For instance, static network configuration limits the use of IT protocols and discovery functions. It proves best with the portscans (A1 and A2). According to MITRE ATT&CK mitigation, it is also effective in avoiding and/or controlling attacks such as network sniffing, man-in-the-middle attack, and system discovery. Table 9 provides a list of mitigation strategies addressing the attacks from Table 3.

TABLE 9. Mitigation Strategies Addressing Attacks

Sr. No.	Suggested mitigation	Addressed to
0	Accept risk	A1
1	Static network configuration	A1, A2
2	Strong authentication and authorization	A2, A3, A4, A5
3	Disable unnecessary services/ports	A2, A3, A4
4	Network Segmentation	A2, A3
5	Network Traffic Encryption	A2,
6	Network Segmentation	A2, A3, A4
7	Access management	A3, A4
8	Update software	A3, A4
9	Network allow list	A3, A4
10	Vulnerability scanning	A2, A3, A4
11	Software process and device authentication	A4
12	Data backup	A2, A6
13	Reliability and robust tests	A6
14	Limit request on the source	A1, A2, A6
15	Restrict automatic execution	A4, A5
16	Training of employees	Avoiding mishaps
17	Regular maintenance	Safe operation
18	Redundancy	Availability

VII. CONCLUSION

The proposed combination of a safety-augmented NIDS to the suggested risk management framework provides multiple benefits. The incidents identified by the NIDS enable the application of the risk management framework [63], [68] also reactively for these very incidents instead of being able to manage potential risks proactively only. Beyond that, attacks against safety-relevant assets are captured and prioritized according to the introduced evaluation formula despite the number of incidents identified in the same time frame. This prevents losing valuable time in deciding which incident to focus on during incident response and avoids wrong or delayed decisions. The proposed risk analysis maps, based on the results of the NIDS, the identified incidents to attacks which enables to identify attack sequences and links between attacks. This knowledge may help to consider additional proactive measures against attacks that might follow up according to the attack pattern noticed. Nevertheless, this approach is also limited due to its heavy dependence on predefined knowledge. This creates additional effort for the operators, especially if new nodes are added to the network with different specifications/properties. Luckily, most SCADA/Cyber-Physical System (CnC) networks face such changes rarely.

Future work includes the implementation of unsupervised machine-learning-based incident detection so that training is no longer dependent on labeled data. A future iteration of this approach could also consider the dynamic nature of asset priorities: If one out of two components of a parallel system fails, for example, the remaining component would turn into a more valuable asset until the successful recovery of the first. The evaluation with new experiments in different settings is also future work as well as possible extensions of the output, such as recommendation of countermeasures on a more concrete level (e.g., *update service X to version 5.10 or higher*). This can lead to refined recommendations or even automated update and maintenance procedures, further enhancing the value of such an IDS for the operators of the ICS.

ACKNOWLEDGMENT

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

REFERENCES

[1] M. A. Umer, K. N. Junejo, M. T. Jilani, and A. P. Mathur, "Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations," *Int. J. Crit. Infrastructure Protection*, vol. 38, 2022, Art. no. 100516, doi: 10.1016/j.ijcip.2022.100516. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1874548222000087

[2] J.-w. Myung and S. Hong, "ICS malware triton attack and countermeasures," *Int. J. Emerg. Multidisciplinary Res.*, vol. 3, pp. 13–17, 2019, doi: 10.22662/IJEMR.2019.3.2.013.

[3] A. Carcano, I. N. Fovino, M. Masera, and A. Trombetta, "State-based network intrusion detection systems for SCADA protocols: A proof of concept," in *Critical Information Infrastructures Security*, E. Rome and R. Bloomfield, Eds. Berlin, Germany: Springer 2010, pp. 138–150.

[4] L. Pièrre-Cambacédès and M. Bouissou, "Modeling safety and security interdependencies with BDMP (Boolean logic driven Markov processes)," in *Proc. IEEE Int. Conf. Syst., Man, Cybernet.*, 2010, pp. 2852–2861, doi: 10.1109/ICSMC.2010.5641922.

[5] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, 2013.

[6] Snort, "SNORT - Network intrusion detection & prevention system," 2023. Accessed: Sep. 3, 2020. [Online]. Available: https://www.snort.org/

[7] Suricata-ids, "Suricata| Open Source IDS / IPS / NSM engine," 2023. Accessed: May 3, 2020. [Online]. Available: https://suricata-ids.org/

[8] Zeek, "The Zeek network security monitor," 2020. Accessed: Mar. 5, 2020. [Online]. Available: https://zeek.org/

[9] D. Day and B. Burns, "A performance analysis of SNORT and Suricata network intrusion detection and prevention engines," in *Proc. 5th Int. Conf. Digit. Soc.*, 2011, pp. 187–192.

[10] V. Jyothsna, V. R. Prasad, and K. M. Prasad, "A review of anomaly based intrusion detection systems," *Int. J. Comput. Appl.*, vol. 28, no. 7, pp. 26–35, 2011.

[11] M. Mantere, M. Sailio, and S. Noponen, "Feature selection for machine learning based anomaly detection in industrial control system networks," in *Proc. IEEE Int. Conf. Green Comput. Commun.*, 2012, pp. 771–774.

[12] K. Stefanidis and A. G. Voyiatzis, "An HMM-based anomaly detection approach for SCADA systems," in *Proc. IFIP Int. Conf. Inf. Secur. Theory Pract.*, 2016, pp. 85–99.

[13] H. Mühlburger and F. Wotawa, "A passive testing approach using a semi-supervised intrusion detection model for SCADA network traffic," in *Proc. IEEE Int. Conf. Artif. Intell. Testing*, 2022, pp. 42–47.

[14] R. Colelli, F. Magri, S. Panzieri, and F. Pascucci, "Anomaly-based intrusion detection system for cyber-physical system security," in *Proc. 29th Mediterranean Conf. Control Automat.*, 2021, pp. 428–434.

[15] F. Schuster, A. Paul, R. Rietz, and H. Koenig, "Potentials of using one-class SVM for detecting protocol-specific anomalies in industrial networks," in *Proc. IEEE Symp. Ser. Comput. Intell.*, 2015, pp. 83–90, doi: 10.1109/SSCI.2015.22.

[16] R. R. R. Barbosa, R. Sadre, and A. Pras, "Difficulties in modeling SCADA traffic: A comparative analysis," in *Passive and Active Measurement*. N. Taft and F. Ricciato, Eds. Berlin, Germany:Springer, 2012, pp. 126–135.

[17] S. D. D. Anton, S. Sinha, and H. D. Schotten, "Anomaly-based intrusion detection in industrial data with SVM and random forests," in *Proc. Int. Conf. Softw. Telecommun. Comput. Netw.*, 2019, pp. 1–6.

[18] G. S. Sestito et al., "A method for anomalies detection in real-time ethernet data traffic applied to PROFINET," *IEEE Trans. Ind. Informat.*, vol. 14, no. 5, pp. 2171–2180, May 2018.

[19] K. Krithivasanet al., "Detection of cyberattacks in industrial control systems using enhanced principal component analysis and hypergraph-based convolution neural network (EPCA-HG-CNN)," *IEEE Trans. Ind. Appl.*, vol. 56, no. 4, pp. 4394–4404, Jul./Aug. 2020.

[20] M. R. Monfared and S. M. Fakhrahmad, "Development of intrusion detection in industrial control systems based on deep learning," *Iranian J. Sci. Technol. Trans. Elect. Eng.*, vol. 46, no. 3, pp. 641–651, 2022, doi: 10.1007/s40998-022-00493-6.

[21] S. A. Varghese, A. Dehlaghi Ghadim, A. Balador, Z. Alimadadi, and P. Papadimitratos, "Digital twin-based intrusion detection for industrial control systems," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops Affiliated Events (PerCom Workshops)*, 2022, pp. 611–617, doi: 10.1109/PerComWorkshops53856.2022.9767492.

[22] V. Ravi, R. Chaganti, and M. Alazab, "Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system," *Comput. Elect. Eng.*, vol. 102, 2022, Art. no. 108156.

[23] H. Gunjal, P. Patel, and D. D. Ebrahimi, "Smart network intrusion detection system for cyber security of industrial IoT," 2022, doi: 10.36227/techrxiv.21431889.v1.

[24] H. Gu, Y. Lai, Y. Wang, J. Liu, M. Sun, and B. Mao, "DEIDS: A novel intrusion detection system for industrial control systems," in *Neural Comput. Appl.*, vol. 34, no. 12, pp. 9793–9811, 2022.

[25] J. Goh, S. Adepu, M. Tan, and Z. S. Lee, "Anomaly detection in cyber physical systems using recurrent neural networks," in *Proc. IEEE 18th Int. Symp. High Assurance Syst. Eng.*, 2017, pp. 140–145.

[26] J. Tai, I. Alsmadi, Y. Zhang, and F. Qiao, "Machine learning methods for anomaly detection in industrial control systems," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, 2020, pp. 2333–2339, doi: 10.1109/BigData50022.2020.9378018.

[27] A. Terai, T. Chiba, H. Shintani, S. Kojima, S. Abe, and I. Koshijima, "Intrusion detection using long short-term memory model for industrial control system," *Int. J. Saf. Secur. Eng.*, vol. 10, no. 2, pp. 183–189, 2020.

[28] S. Mokhtari, A. Abbaspour, K. K. Yen, and A. Sargolzaei, "A machine learning approach for anomaly detection in industrial control systems based on measurement data," in *Electronics*, vol. 10, no. 4, 2021, Art. no. 28.

[29] O. A. Alimi, K. Ouahada, A. M. Abu-Mahfouz, S. Rimer, and K. O. A. Alimi, "Supervised learning based intrusion detection for SCADA systems," in *Proc. IEEE Nigeria 4th Int. Conf. Disruptive Technol. Sustain. Develop.*, 2022, pp. 1–5.

[30] D.-D. Nguyen, M.-T. Le, and T.-L. Cung, "Improving intrusion detection in SCADA systems using stacking ensemble of tree-based models," *Bull. Elect. Eng. Informat.*, vol. 11, no. 1, pp. 119–127, 2022.

[31] L. A. Maglaras and J. Jiang, "Intrusion detection in SCADA systems using machine learning techniques," in *Proc. Sci. Inf. Conf.*, 2014, pp. 626–631, doi: 10.1109/SAI.2014.6918252,.

[32] B. Stewart, L. Rosa, L. Maglaras, T. J. Cruz, P. Simões, and H. Janicke, "Effect of network architecture changes on OCSVM based intrusion detection system," in *Industrial Networks and Intelligent Systems*, L. A. Maglaras, H. Janicke, and K. Jones, Eds. Berlin, Germany: Springer, 2017, pp. 90–100.

[33] L. A. Maglaras, J. Jiang, and T. J. Cruz, "Combining ensemble methods and social network metrics for improving accuracy of OCSVM on intrusion detection in SCADA systems," *J. Inf. Secur. Appl.*, vol. 30, pp. 15–26, 2016.

[34] W. Yu, Y. Wang, and L. Song, "A two stage intrusion detection system for industrial control networks based on ethernet/IP," in *Electronics*, vol. 8, no. 12, 2019, Art. no. 1545.

[35] M. Mantere, M. Sailio, and S. Noponen, "A module for anomaly detection in ICS networks," in *Proc. 3rd Int. Conf. High Confidence Netw. Syst.* (ser. HiCoNS Association for Computing Machinery), 2014, pp. 49–56, doi: 10.1145/2566468.2566478.

[36] P. Schneider and K. Böttinger, "High-performance unsupervised anomaly detection for cyber-physical system networks," in *Proc. Workshop Cyber-Phys. Syst. Secur. Privacy* (Ser. Association for Computing Machinery), 2018, pp. 1–12, doi: 10.1145/3264888.3264890.

[37] S. Kim, W. Jo, and T. Shon, "APAD: Autoencoder-based payload anomaly detection for industrial IOE," *Appl. Soft Comput.*, vol. 88, 2020, Art. no. 106017.

[38] J.-R. Jiang and Y.-T. Chen, "Industrial control system anomaly detection and classification based on network traffic," *IEEE Access*, vol. 10, pp. 41874–41888, 2022.

[39] H. T. Truong et al., "Light-weight federated learning-based anomaly detection for time-series data in industrial control systems," *Comput. Ind.*, vol. 140, 2022, Art. no. 103692.

[40] M. Altaha and S. Hong, "Anomaly detection for SCADA system security based on unsupervised learning and function codes analysis in the DNP3 protocol," in *Electronics*, vol. 11, no. 14, 2022, Art. no. 2184.

[41] R. Bhatia, S. Benno, J. Esteban, T. V. Lakshman, and J. Grogan, "Unsupervised machine learning for network-centric anomaly detection in IoT," in *Proc. 3rd ACM CoNEXT Workshop Big Data Mach. Learn. Artif. Intell. Data Commun. Netw.*, 2019, pp. 42–48, doi: 10.1145/3359992.3366641.

[42] P. Laskov, P. Düssel, C. Schäfer, and K. Rieck, "Learning intrusion detection: Supervised or unsupervised," in *Proc. 13th Int. Conf. Image Anal.*, 2005, pp. 50–57.

[43] R. Mitchell and R. Chen, "Behavior-rule based intrusion detection systems for safety critical smart grid applications," *IEEE Trans. Smart Grid*, vol. 4, no. 3, pp. 1254–1263, Sep. 2013.

[44] A. Wasicek, M. D. Pesé, A. Weimerskirch, Y. Burakova, and K. Singh, "Context-aware intrusion detection in automotive control systems," in *Proc. 5th ESCAR USA Conf.*, 2017, pp. 21–22.

[45] C. W. Johnson, "Barriers to the use of intrusion detection systems in safety-critical applications," in *Proc. Comput. Saf. Rel. Secur. 34th Int. Conf.*, 2015, pp. 375–384.

[46] D. Tharanga, "Thesis: Critical review of risk identification techniques," May 2020. [Online]. Available: https://www.researchgate.net/publication/342159877_CRITICAL_REVIEW_OF_RISK_IDENTIFICATION_TECHNIQUES

[47] J. Tixier, G. Dusserre, O. Salvi, and D. Gaston, "Review of 62 risk analysis methodologies of industrial plants," *J. Loss Prevention Process Ind.*, vol. 15, no. 4, pp. 291–303, 2002, doi: 10.1016/S0950-4230(02)00008-6. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950423002000086

[48] X. Lyu, Y. Ding, and S. Yang, "Safety and security risk assessment in cyber-physical systems," *IET Cyber-Phys. Syst. Theory Appl.*, vol. 4, pp. 221–232, 2019, doi: 10.1049/iet-cps.2018.5068.

[49] P. Bhosale, W. Kastner, and T. Sauter, "A centralised or distributed risk assessment using asset administration shell," in *Proc. 26th IEEE Int. Conf. Emerg. Technol. Factory Automat.*, 2021, pp. 1–4, doi: 10.1109/ETFA45728.2021.9613152.

[50] T. Abdelghani, "Implementation of defense in depth strategy to secure industrial control system in critical infrastructures," *Amer. J. Artif. Intell.*, vol. 3, pp. 17–22, 2020, doi: 10.11648/j.ajai.20190302.11.

[51] T. Aven, "Risk assessment and risk management: Review of recent advances on their foundation," *Eur. J. Oper. Res.*, vol. 253, no. 1, pp. 1–13, 2016, doi: 10.1016/j.ejor.2015.12.023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221715011479

[52] Y. Ben Charhi, N. Mannane, E. Bendriss, and B. Regragui, "Intrusion detection in cloud computing based attacks patterns and risk assessment," in *Proc. 3rd Int. Conf. Syst. Collaboration*, 2016, pp. 1–4.

[53] W. Kanoun, N. Cuppens-Boulahia, F. Cuppens, and J. Araujo, "Automated reaction based on risk analysis and attackers skills in intrusion detection systems," in *Proc. 3rd Int. Conf. Risks Secur. Internet Syst.*, 2008, pp. 117–124, doi: 10.1109/CRISIS.2008.4757471.

[54] W. Kanoun, N. Cuppens-Boulahia, F. Cuppens, and F. Autrel, "Advanced reaction using risk assessment in intrusion detection systems," in *Critical Information Infrastructures Security*. Berlin, Germany:Springer, 2008, pp. 58–70.

[55] A. Shameli-Sendi, M. Cheriet, and A. Hamou-Lhadj, "Taxonomy of intrusion risk assessment and response system," *Comput. Secur.*, vol. 45, pp. 1–16, 2014.

[56] A. Hernandez and E. Magana, "One-way delay measurement and characterization," in *Proc. Int. Conf. Netw. Serv.*, 2007, pp. 114–114.

[57] G. Vormayr, J. Fabini, and T. Zseby, "Why are my flows different? A tutorial on flow exporters," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2064–2103, Jul.–Sep. 2020.

[58] F. Meghdouri, T. Zseby, and F. Iglesias, "Analysis of lightweight feature vectors for attack detection in network traffic," *Appl. Sci.*, vol. 8, no. 11, 2018, Art. no. 2196.

[59] N. Williams, S. Zander, and G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 5, pp. 5–16, 2006.

[60] F. Iglesias and T. Zseby, "Time-activity footprints in IP traffic," *Comput. Netw.*, vol. 107, pp. 64–75, 2016.

[61] D. C. Ferreira, F. I. Vázquez, G. Vormayr, M. Bachl, and T. Zseby, "A meta-analysis approach for feature selection in network traffic research," in *Proc. Reproducibility Workshop*, 2017, pp. 17–20.
[62] B. Brenner and E. Weippl, *Security Analysis and Improvement of Data Logistics in Automation ML-Based Engineering Networks*. Berlin, Germany: Springer, 2019, pp. 305–334, doi: 10.1007/978-3-030-25312-7_11.
[63] S. Hollerer, T. Sauter, and W. Kastner, "Risk assessments considering safety, security, and their interdependencies in OT environments," in *Proc. 17th Int. Conf. Availability Rel. Secur.*, 2022, pp. 1–8, doi: 10.1145/3538969.3543814.
[64] E. Blake et al., "Finding cyber threats with ATT&CK-based analytics," MITRE Corp., Annapolis Junction, MD, USA, Tech. Rep. MTR170202, 2017.
[65] "ISO Guide 73:2009: Risk management—Vocabulary," 2009.
[66] W. A. Conklin, "IT vs. OT security: A time to consider a change in CIA to include resilience," in *Proc. 49th Hawaii Int. Conf. Syst. Sci.*, 2016, pp. 2642–2647, doi: 10.1109/HICSS.2016.331.
[67] S. Hollerer, W. Kastner, and T. Sauter, "Safety und security-ein spannungsfeld in der industriellen praxis," *Elektrotech. Informationstechnik*, vol. 138, no. 7, pp. 449–453, 2021.
[68] P. Bhosale, W. Kastner, and T. Sauter, "Automating safety and security risk assessment in industrial control systems: Challenges and constraints," in *Proc. IEEE 27th Int. Conf. Emerg. Technol. Factory Automat.*, 2022, pp. 1–4, doi: 10.1109/ETFA52439.2022.9921517.

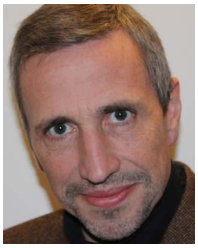
BERNHARD BRENNER received the B.Sc. degree in medical informatics from TU Wien, Vienna, Austria, in 2014, and the M.Sc. degree in computer security from Denmark Technical University (DTU), Kongens Lyngby, Denmark, in 2016. He is currently working toward the Ph.D. degree with TU Wien, focusing on cybersecurity in OT networks.

SIEGFRIED HOLLERER received the B.Sc. degree in 2016, and Dipl.-Ing. (similar to M.Sc.) degree, TU Wein, Vienna, Austria, in 2016. He is currently working towards the Ph.D. degree from TU Wien, Vienna, Austria.

He worked as a penetration tester, conducting assessments on web applications, system hardening checks, social engineering attacks, and received the certificate OSCP (Offensive Security Certified Professional), for 5 years. Furthermore, he does security risk assessments based on the industrial security standard IEC 62443.

PUSHPARAJ BHOSALE received the B.E. degree in electronics and telecommunication from Mumbai University, Mumbai, India, in 2014, and the M.Sc. degree in sensor system technology from the Vellore Institute of Technology, Vellore, India, in 2017, and Hochschule Karlsruhe-Technik und Wirtschaft, Karlsruhe, Germany, under a dual-degree program. He is currently working toward the Ph.D. degree in safety and security in industry for automated risk management in industrial control systems from TU Wien, Vienna, Austria.

THILO SAUTER (Fellow, IEEE) received the Dipl-Ing. and Dr. Techn. degrees in electrical engineering from TU Wien, Vienna, Austria, in 1992 and 1996, respectively. He is currently a Professor with the Institute for Computer Technology, TU Wien, Vienna, Austria. He was the founding Director of the Department of Integrated Sensor Systems, University for Continuing Education Krems. His research interests include intelligent sensors, and industrial communication systems focusing on questions regarding real-time, safety, security, and integration.



WOLFGANG KASTNER received the Dipl-Ing. and Dr. Techn. degrees in computer science from TU Wien, Vienna, Austria, in 1992 and 1996, respectively. He is currently a Full Professor with the Institute of Computer Engineering and leads the Research Unit Automation Systems, TU Wien, Vienna, Austria. His research interests include the design, analysis, and modeling of distributed automation systems and their seamless integration into the Internet of Things in the industrial domain focusing on knowledge representation and safety and security aspects.

JOACHIM FABINI received the diploma degree (Dipl.-Ing.) in technical computer sciences from Technische Universitaet Wien (TU Wien), Vienna, Austria, in 1997, and the master.s degree in computer science management (Mag.rer.soc.oec) and the Dr. techn degree from the Institute of Telecommunications (formerly Institute of Broadband Communications), TU Wien, in 2005 and 2008, respectively. His Ph.D. dissertation was titled "Generic Access Network Modelling for Next Generation Network Applications."

After five years of industrial R&D at Ericsson Austria in the area of Voice over IP, he joined the Institute of Telecommunications, TU Wien, in 2003.

TANJA ZSEBY received the Diploma and Ph.D. degrees in electrical engineering from TU Berlin, Berlin, Germany, in 2005.

She is currently a Full Professor of Communication Networks with the Faculty of Electrical Engineering and Information Technology, TU Wien, Vienna, Austria. Before joining TU Wien, she led the Competence Center for Network Research with the Fraunhofer Institute for Open Communication Systems (FOKUS), Berlin, and was the Visiting Scientist with the University of California, San Diego.