electronics

Article

The Vulnerability of the Production Line Using Industrial IoT Systems under DDoS Attack

Tibor Horak [1,*], Peter Strelec [1], Ladislav Huraj [2], Pavol Tanuska [1], Andrea Vaclavova [1] and Michal Kebisek [1]

[1] Institute of Applied Informatics, Automation and Mechatronics, Faculty of Materials Science and Technology in Trnava, Slovak University of Technology in Bratislava, 91724 Trnava, Slovakia; peter.strelec@stuba.sk (P.S.); pavol.tanuska@stuba.sk (P.T.); andrea.vaclavova@stuba.sk (A.V.); michal.kebisek@stuba.sk (M.K.)

[2] Department of Applied Informatics, University of SS. Cyril and Methodius, 91701 Trnava, Slovakia; ladislav.huraj@ucm.sk

* Correspondence: tibor.horak@stuba.sk

Abstract: Internet of Things (IoT) devices are not only finding increasing use in ordinary households, but they have also become a key element for the Industry 4.0 concept. The implementation of industrial IoT devices into production streamlines the production process and reduces production costs. On the other hand, connected IoT devices bring new security risks to production and expose an industrial environment to new types of attacks. The article analyzes the vulnerability of the production line with implemented industrial IoT devices with consideration of a possible Distributed Denial-of-service (DDoS) attack led by attackers from the internet. Various types of DDoS attacks abusing the presence of IoT devices in the system were performed on an automated production line implementing sorting, preparation, and dosing of bulk and liquid materials for filling into containers. The leading attacks caused failure of the production line during the production, as well as the dysfunction of communication with IoT devices. The article also demonstrates the implementation of countermeasures against DDoS attacks and possible strategies to protect and mitigate such attacks on the production line.

Keywords: industrial IoT; DDoS attack; production line

Citation: Horak, T.; Strelec, P.; Huraj, L.; Tanuska, P.; Vaclavova, A.; Kebisek, M. The Vulnerability of Production Line Using Industrial IoT Systems under DDoS Attack. *Electronics* **2021**, *10*, 381. https://doi.org/10.3390/electronics10040381

Academic Editors: Stavros Shiaeles, Bogdan Ghita and Nicholas Kolokotronis

Received: 29 December 2020
Accepted: 29 January 2021
Published: 4 February 2021

1. Introduction

Production lines are one of the basic elements of industrial production and enable an effective management of the production process, and therefore, it reduces costs. In order to improve the efficiency of the production, and the effective interconnection of production units, it is necessary to place emphasis on the monitoring and maintenance of individual components involved in the production process. Monitoring and maintenance of the production line are currently done using tablets or operator panels, where operators can remotely set various parameters of the production line or check the status of the line. The Industrial Internet of Things (IIoT) finds its use in this kind of activity. IIoT is already commonly used to monitor large production systems and to diagnose errors, but also to identify machines, e.g., via RFID (Radio-Frequency IDentification) and for many other uses for various industries, such as power plants, water supplies, or oil and gas refineries [1,2].

The introduction of IIoT into traditional control systems is a natural consequence of the onset of modern production trends. The rapid development of communication networks has enabled the application of network management systems to production, while allowing key components of production to be distributed and controlled through the communication network. The increase in the technological functionality of IoT devices, as well as the reduction of their prices, has led to the introduction of IoT technology into industrial production systems and subsequently to the introduction of the IIoT concept [3]. The IIoT concept enables the use of the Internet of Things to connect various components of the production establishment to the internet from anywhere in the world and their easy control via mobile applications.

The integration of an IIoT technology into the industrial environment brings better flexibility and efficiency of the production system and new services. The data obtained using IIoT are more accurate and can be collected continuously. The more accurate and continuous data will enable a better monitoring system to prevent dangerous situations from occurring, whether it is a nuclear power establishment or another production establishment. In addition, the implementation of IIoT in production lines, or other industrial projects, aims to reduce production or maintenance costs and improve efficiency, stability, safety, etc. [4].

The integration of IIoT technology into the industrial environment, on the other hand, brings new challenges to be faced as well. The manufacturers place emphasis on low-cost IoT devices, and because of this, the security of IoT devices is neglected. A typical IoT device includes a processor with relatively low performance, low RAM capacity, and the ability to connect to the internet. The IoT device often does not have sufficient resources to ensure its own safety. The effort to implement a security feature directly in an IoT device will generally rapidly reduce the performance of the IoT device and increase its cost. It is the security of the Internet of Things that is one of the biggest weaknesses hampering the adoption of IIoT on a massive scale.

Therefore, it is important to pay attention to the security and to the potential of the misuse of IoT devices in Industry 4.0, as well as to design efficient and cost-effective platforms secured against cyber-attacks. All the more so, as the implementation of digital security measures in industrial control, a system environment is often considered a secondary priority. The never-ending race for higher yields and productivity on the production line also contributes significantly to the concept of safety in environments, such as the noncritical component [5].

The IT manufacturing industry relies on three basic security requirements: Confidentiality, integrity, and availability. Confidentiality is the preservation of the privacy of the flow of information in all flows of the production chain. A loss of confidentiality can mean that significant losses for the company, customer data, intellectual property, trade secrets, and so on can be compromised. The integrity in production systems represents the consistency, accuracy, and reliability of information going through the production chain, but also the consistency and reliability of physical components throughout the product life cycle. The integrity of the manufacturing system in Industry 4.0 can be easily affected by cyber-attacks due to the closer connection of operating technologies with the IT infrastructure of the manufacturing system. The availability of the production system may be compromised by various cyber and physical attacks by attackers on that specific system. A service outage caused by these attacks can cause different components of the product lifecycle to be deactivated simultaneously, which can lead to a halt in the entire process chain [6].

The experiments performed in this study focus primarily on disrupting the availability of the production system in the form of Distributed Denial-of-Service (DDoS) attacks performed on the production line from the internet. Denial-of-service (DoS) attacks deplete the system resources and then process them slowly until the system is disabled or shut down. Distributed DoS (DDoS) attacks are designed to use multiple devices to attack by sending a large amount of traffic to the network and consuming computer resources.

A DDoS attack can also reduce the functionality of the Internet of Things (IoT), as these devices are not designed to handle this type of attack. In some environments, it can cause production line interruptions, and the correct system can only be restored by human assistance.

1.1. Related Works

Examples of real practice also point to the seriousness of attacks on industrial and production systems. One of the most famous cyber-attacks on production systems was the attack on the Ukrainian electricity grid in December 2015. The attack took place as a combination of several hacking techniques, including DDoS attacks, malicious e-mails, internet virus worms, and malware. The attackers managed to disrupt the entire distribution company and the power outage for more than 200,000 people throughout the whole of Ukraine. The situation became even more serious because, in the winter, Ukrainians had a problem with household heating. Malicious malware was used to infect the Ukrainian energy system via infected emails and networks. The DDoS attack was used to prevent the proper functioning and recovery of infected industrial systems, and the internet virus worm denied access to industrial technicians by logging them out of the system and changing their password to their user accounts [5].

The cyber-attacks on the production line have been known for a long time already. For example, already in 2005, an attack on production lines was recorded using the Zotob internet virus worm, which caused a production outage at DaimlerChrysler's automotive enterprises in the USA for 5 up to 50 min on various production lines. The Zotob worm exploited a vulnerability in the Windows 2000 operating system and some earlier versions of the Microsoft Windows operating systems associated with a buffer overflow on the TCP port. The operating systems became unstable after being attacked by the Zotob worm, which resulted in unplanned system shutdowns and restarts [7].

It is the implementation of IoT devices into the production system that can increase the vulnerability of the system and allow an attacker to perform an attack in a way that would not have been possible before. An example is the use of an implemented IoT device as a reflector for a reflected DDoS attack. Many publications appeal to the vulnerability of DDoS to attacks by production and the production line connected to IoT devices, e.g., [5,8,9]; however, specific impacts on the production line are not described or tested in more detail anywhere.

In 2017, Knudsen et al. [10] reported a cyber-attack on the production line but did not implement a DDoS attack, so the production line was endangered in other ways. Their main findings and shortcomings were the SSH (Secure Shell) service with support for insufficient RC4 encryption (Rivest Cipher 4) and the use of default passwords for various services.

The latest state-of-the-art research has been conducted in the field of cyber-security mitigating DDoS attacks [11–19]. Table 1 summarizes research on different approaches to detecting, defending, and mitigating DDOS attacks.

Table 1. Summary of related research for detecting, mitigating, and defending against Distributed Denial-of-Service (DDoS) attacks.

Research	Method	Description
S.N. Shiaeles, et al., 2012 [11]	DDoS detection using fuzzy estimators	detecting a DDoS and identifying the malicious IPs
S.N. Shiaeles, et al., 2015 [12]	fuzzy hybrid spoofing detector	based on source MAC address, hop count, GeoIP, OS passive fingerprinting, and web browser user agent
M. Siracusano, et al., 2018 [13]	detection of LDDoS attacks	based on characteristics of malicious TCP flows
Q. Yan et al., 2018 [14]	DDoS mitigation	a multi-level DDoS mitigation framework for IIoT including the edge computing level, fog computing level, and cloud computing level
B. Saridou, et al., 2019 [15]	DDoS mitigation	a machine learning-based system promoting high availability of DNS services during DDoS attacks
D.J. Prathyusha and K. Govinda 2020 [16]	DDoS mitigation	based on network flow analysis at the targeted side against virtual services
W.L. Costa, et al., 2020 [17]	DDoS detection	detection mechanism based on machine learning
D.V.V.S. Manikumar and B.U. Maheswari 2020 [18]	DDoS mitigation	system uses machine learning algorithms to identify the incoming packet and uses blockchain technology to store the blacklist
B. Wang and X. Zhang 2020 [19]	DDoS mitigation	defense strategy based on dynamic IP packet filtering technology

1.2. Contributions

Industrial IoT devices bring various changes to a production system. However, they also add various safety challenges to the product life cycle of the production system.

In this study, a production line, combined with industrial IoT devices, is tested for DDoS attacks to determine the security vulnerabilities of the IIoT application in the production process. The real production line containing components necessary for the production process such as programmable logic controllers (PLCs), a robot, a QR code reading camera that identifies the pallet number, and an RFID system that identifies the type of packaging and transport system has been extended by common industrial IoT devices, specifically a smart thermostat and a Fibaro control unit. These IoT devices were installed in a common computer network such as the production line. The role of the thermostat is to sense the temperature and then regulate the room temperature to match the production conditions on the production line. The Fibaro control unit, with appropriate sensors such as a fire sensor, flood sensor, and motion sensor, has the task of protecting the production line from fire, floods, and unwanted people.

Various types of DDoS attacks on this real production line, from the WAN network and also at the level of the local computer network of the production line, were performed and tested. First, a reflected DDoS attack was implemented, and it was using added IoT devices as reflectors. Direct DDoS attacks, on specifically selected components in the production line, were also tested. The results from the experiment show that such conducted DDoS attacks have a devastating effect on the operation of the production line and cause the cessation of the entire production process taking place on the line. The performed analysis of attacks made it possible to identify, describe, and generalize system vulnerabilities that caused the system to fail.

The study proposed and implemented possible countermeasures against the selected attacks, taking into account the trade-off between connectivity and security, which is an important challenge in integrating IoT into the production. Two possible variants of countermeasures were tested. The first variant was a transparent firewall, which protects IoT devices connected to the WAN network; and the second variant was the creation of a communication map and preventing degradation of the communication of infected IoT devices at the switch level toward the production line. The testing of the countermeasures has shown the effectiveness and the robustness of the proposed countermeasures to stop or mitigate DDoS attacks on the production line and, therefore, maintain the continuity of the production process.

The results of this study will find use in the implementation of IoT devices into the production process built on the production line, and they will help to increase the overall safety of the industrial environment connected with the paradigm of the Internet of Things.

Real DDoS attacks with an impact on a production line with the IoT systems have not yet been reported in the literature. The vulnerability of the production line with IoT systems is usually only assumed, but not confirmed for real. Siemens PLCs define that even in the event of network anomalies, they can continue to operate. Research has shown that this is only valid if integration with other devices is not done. Sensors, cameras, and other elements of the production line do not really solve security requirements, and solutions from the point of view of security are absent in the design of production lines. The article shows how specific conditions can and do manifest themselves in the production process. By deploying the described design in practice, it was possible to immediately detect the problematic behavior of the IoT device. Real testing of production line vulnerabilities for DDoS attacks is the main novelty of this article, as well as finding that the mitigation of the production line is possible with relatively cheap devices. However, countermeasures must be targeted at IoT systems. Research has shown that a DDoS attack on a production line has had devastating effects and that, by default, securing a production line against DDoS attacks is not sufficient at all. The research results are widely applicable in Industry 4.0.

1.3. Recommendations for Securing IIoT and IoT

The implementation of IoT devices, in industrial production, makes it possible to significantly reduce the costs of operation and maintenance of industrial equipment. Even older devices can be upgraded with smart sensors that provide more features, intelligence, and connectivity between devices. Overall, the implementation of IoT in industries helps to reduce the error rate and increase the safety and efficiency of the production process [20].

The limitations of IoT devices in terms of static configurations and computing resources, as well as the lack of safety mechanisms in IoT devices, are the main shortcomings of IoT ecosystems. These limitations make IoT devices an easy target for various threats directed at them from the internet [21].

Relying on the isolation of the environment is not a safety measure. Industrial management and control systems SCADA ICS/DCS form a large, and until recently, independent area from the point of view of cybersecurity [22]. However, the cyber-attacks that target industrial facilities are a serious threat. Remote access to machines brings clear advantages for a production. Up to 63% of maintenance work on the machine is either a routine inspection, or it is found that the problem simply does not exist. In addition, 30% or more of these corrections can be made remotely by adjusting the parameters via the internet or with a little help from a person at the place [23]. IoT devices in the industry, or even for home use, are often operated on simple hardware, which is deployed across different environments. This wide deployment leads to new challenges, which are unique to IoT devices. The software updates and vulnerability management are very important, and they require different strategies than other IT applications. When designing secure IoT devices, the following areas need to be considered:

- data classification
- physical security
- secure installation of the device
- secure operating systems
- application security
- credential management
- encryption
- network connections
- software updates

When designing IIoT and IoT devices, it is necessary to build a secure Internet of Things from the early stage of the process design or through consultations and audits at later stages. With these measures, it is possible to protect and prevent various cyber attacks from penetrating through IoT devices, for example, into the network infrastructure, in which the production line is served, where these attacks could cause great damage in the production process [24].

2. Materials and Methods

The role of industrial IoT devices is more specific than the role of common IoT devices. IIoT devices are usually limited to industrial production or controllers, and not to all types of sensors in IoT. Devices, at the level of IIoT sensors, form a special communication with others based on the requirements of production and supervision. For example, to achieve synchronization, the PLC may communicate with another through self-defined protocols; a camera is always used to monitor the smart meter in the production line; images or videos are transformed from higher-layer IIoT sensors by using internal protocols. Besides this, configuration tools should be equipped with efficient performance and excellent stability, especially for industrial control and real-time data collection [25].

We decide to test two IIoT devices and check the impact on the production line, which were used in the factory.

2.1. IoT Thermostat

The IoT device is the control unit of the Honeywell system. It is a multi-zone controller that enables control of the temperature of individual rooms by using individual time programs. The system is designed for radiator heating, under-floor heating, and also for controlling the charging of the hot water tank. The Honeywell control unit has a universal use. It can be operated as a multi-zone controller or as a simple room thermostat with the possibility of later expansion to a multi-zone system [26]. The controller is equipped with a sophisticated Fuzzy logic with the ability to adapt to local operating conditions and has optimization functions such as preheating or cooling. This ensures accurate and efficient temperature control throughout the industrial building. Honeywell allows people to control heating more efficiently while saving business costs. It will ensure the right temperature in the right room and at the right time. The Honeywell thermostat can be controlled via a mobile device via a network connection. It connects to the network infrastructure wirelessly via WI-FI [27].

IoT Fibaro Security System

An IoT security system, in the form of a miniaturized (small) control unit, is the brain of the Fibaro system. The control unit listens to and controls all wireless modules, which are motion, smoke, flood, door sensor, and wireless socket, through the network with modern Z-Wave technology also being a gateway to the internet and Wi-Fi network [28]. With its help, it is possible to know about all events in a modern company (including history), and it can manage all settings in one place. It can also allow people to connect via a mobile application or web interface anywhere. Advanced technology can control a modern business from anywhere and save time and money [29].

2.2. DDoS Attack

Nowadays, DDoS attacks are one of the most serious threats that companies face. The severity and frequency of these attacks are constantly increasing and can be directed against all types of companies [30]. They come from many sources at the same time, in a way that an attacker infects a great number of computers, and this creates a so-called botnet. The botnet is a network of infected computers, called zombies or bots, which are the attacking devices. In this case, it would be done with the potential to completely flood the service provider's network infrastructure [31]. The purpose of such an attack is to disable and cause dysfunction to the provided service such as servers, websites, enterprise systems, and IoT devices by CPU overload, and also by a RAM memory overload [32]. The motive for the attack can be, for example, the demand for blackmailing, a rivalry of companies, and political motives. The data, sent from the attacking computers, are many times indistinguishable from normal data from a usual user [33]. Nowadays, there are several attempts known to detect a DDoS attack, mitigate the attack, or even stop the attack. However, there are still some DDoS attacks existing against it where there is no effective defense possible [34].

2.2.1. A Reflective DDoS Attack

The reflective DDoS (DRDoS) attack is the most dangerous implementation possible of a DDoS attack. Detecting an attacker is the most challenging of all DDoS attacks because this attack preserves the anonymity of the attacker via an IP address using a potentially innocent third party. The innocent third party is involved indirectly and through it the attacker forwards the flow of attacking data to the target victim. A huge multiplicity makes IoT nodes an interesting amplification tool for attackers [9]. The attacker sends packets with a spoofed source IP address set to the victim's IP address to the reflector devices, thus indirectly overloading the target with packets [35]. The reflector may be a common legitimate device, e.g., IoT device, which did not have to be compromised at all. The advantage of a DRDoS attack is that when tracing the source of the attack, the sent packets do not go directly to the attacker, but they go only to the reflector, and thus to the device that forwards the packets [36]; therefore, security attacks cause automation processes to fail and are all the more insidious because their origin does not have to be revealed using standard monitoring tools [37].

2.2.2. Types of DDoS Attacks

A UDP flood is the most common form of attack. The acronym UDP is an internet protocol that takes care of communication in the network similarly to TCP, but unlike it, it does not use any handshaking, meaning the verification of the establishment, progress, and expiration of communication time. One of the examples is a UDP flood, which works on a very simple principle: Attackers send a large number of UDP packets to random ports on a target server. The target server must respond to packets. First, it checks to see if any of its applications are monitoring these ports. If the server finds that it is not happening, it must reply with the information that the destination is unavailable. It sends this via the ICMP Internet Protocol packet, which is used to send error messages [38]. It must respond to each UDP packet that the server receives according to the communication rules. When the target server gets a huge number of them, it tries to answer them all, thus exhausting its internet connectivity and sometimes other resources as well. A huge amount of processed data will prevent the server from other communication. The mentioned UDP attack is popular probably because the defense against it is quite demanding. The server owners themselves can do the minimum against it. It can, perhaps, only limit the number of packets processed at the same time. However, this does not solve the load on the line toward the server, which can also be significant in the case of a UDP flood [39].

A TCP SYN flood is the second most common attack that uses the key internet protocol TCP, specifically, one of its characteristics. In contrast to simpler, faster, but less accurate UDP, the TCP monitors the course of communication between two parties [40]. The whole process consists of SYN messages, then SYN-ACK, and finally, ACK. However, in the third point, the approach in the case of a SYN flood differs. Although the target server receives a SYN message and responds with its SYN-ACK, it will no longer receive the required confirmation ACK message. It is happening this way on purpose. This way, the server does not know the state of the communication, and it is still waiting for the receiving of this message, which could be delayed, for example, only due to network overflow. Therefore, the server must leave the connection half-open for a period of time, which will result in the depletion of its resources. The whole situation can lead to the depletion of server resources, which leads to its malfunction or direct malfunction [41].

An ICMP flood attack is also known as a Ping flood attack. It is one of the most common DoS attacks. An attacker floods the target of the ICMP ping, which is actually echo-requests. The ICMP generates an echo-request and echo-reply message, and if the target device is capable of responding, it is possible to determine whether the device is responding or to see the basic response times [42]. In the case of a flood attack, packets of this type can force the network to respond to all incoming requests, causing traffic overflow and unavailability. The load is visible, for both, at the input and output of the monitored channel. Regardless of this fact, the target status leads to a denial of service, due to the consumption of the allocated communication zone or device resources [43]. However, an assumption for such an attack is at least a basic knowledge of network infrastructure. Published, targeted, placed—in this type of attack, the ping flood targets a specific computer on the local network. In this case, the attacker must obtain the target IP address in advance [44]. The revealed router—here, the ping flood focuses on routers in order to interrupt communication. Blind ping—this involves using an external program to detect the IP address of the target computer or router before the attack begins. As the ICMP attack floods the network connections of the target device with spoofed traffic, it prevents legitimate requests to execute them [45]. This scenario creates a DoS threat, or even more dangerous, as it is in this case, a more concentrated DDoS attack. In the past, attackers forged a spoofed IP address to mask a sending device. However, with today's sophisticated botnet attacks, attackers do not even bother masking an IP of attacking bots. Instead, they use an extensive network to overwhelm the victim server [46].

2.3. A Production Line

The production line, where all of the attack scenarios were tested, shown in Figure 1, consists of five zones. Each zone represents a certain part of the production process. In addition, each zone consists of several stations. Every station is controlled by its own PLC S7-300.

The family of S7-300 controllers consists of a series of PLCs. These controllers cover a wide range of requirements from simple to very complex. Even though these drivers differ in size and overall capabilities, they are equally similar in operating characteristics, data structure, addressing, memory organization, instruction sets, and programming languages [47]. The products of the processor family, Siemens SIMATIC S7-300, have been designed for discrete and continuous control in industrial environments such as food and beverage production, industrial production, and the worldwide chemical industry [48].

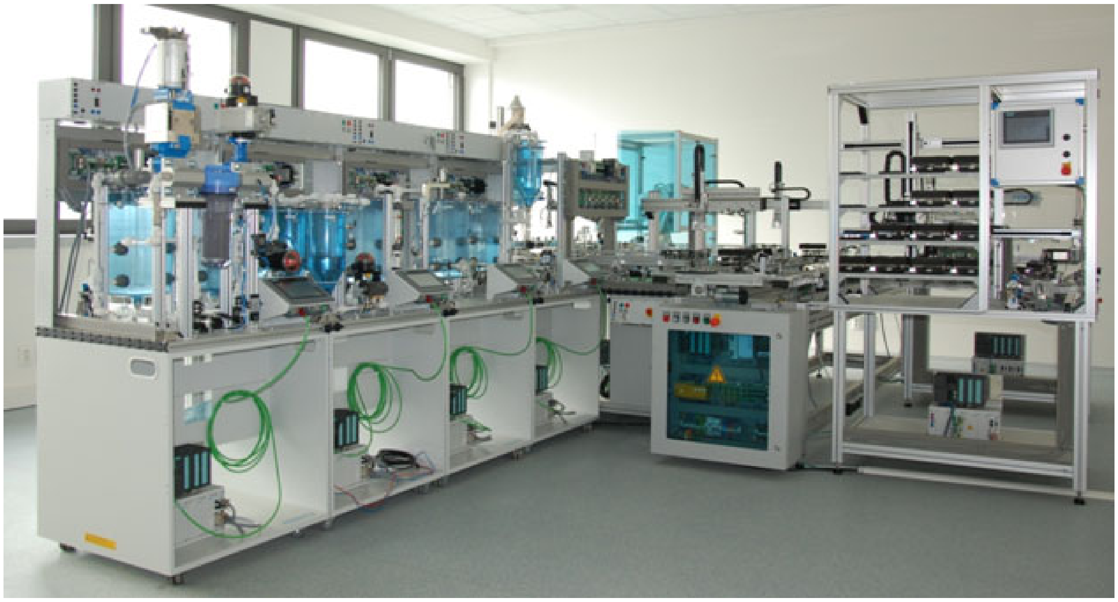


Figure 1. Real production line.

The first zone is used for the collection and dosing of discrete materials in part B by using a Shaker conveyor station, Quality sorting station, and Corn dosing station. A Filtration station, Mixing station, Reactor station, and Quality-probe station are used in order to control the heat when mixing the liquids and mixing that is according to the selected recipe and specific, exact ratios with the specified preparation time in part A. After the liquids are mixed and the discrete materials are dosed, the production process will continue in the second zone. In the second zone, the bottles are filled with liquid or discrete material at the filling (bottling) station. Next, the bottles are closed with a screw cap on the rotating table. After that, the bottles are transported to the filling position by a conveyor belt. These bottles are next separated by a pneumatic separator. The bottles can be filled regardless of the required volume that is specified in the recipe. The bottles that are already filled are transported by a conveyor belt to the packaging station. In this packaging station, bottles are placed in crates with a capacity of 2 × 3 bottles by means of a precise, two-axis automated industrial manipulator. The correctness of the filling of the crate is checked by using a high-speed camera.

In the fourth zone, the crates are further transported between the individual stations by a conveyor belt. This is a modular system with four segments. Each segment has a separate drive, more specifically, a three-phase AC asynchronous motor. The crates, prepared in this way, can be further stored in a storage station, where the crates are stored on four floors of four crates by using a three-axis Cartesian manipulator. Or, additionally, the crates continue along the conveyor belt in the in/out station. There are two unloading ramps for storing crates, at hand, for dispensing by using a three-axis pneumatic manipulator with a pneumatic linear clamp, and an input conveyor for returning empty crates.

The fifth zone is used for unloading bottles from crates by using a biaxial, industrial manipulator that contains a clamp, which is capable of holding three bottles at once and is able to transport them from the unpacking station to the Robot station, called the Recycling station. The Recycling station includes a pump for emptying the liquid contents of the bottles and a vacuum pump for emptying discrete materials. In this station, bottles are being opened by using an industrial Robot with six degrees of freedom. After them being opened, their content is emptied, and then the empty bottles are transported back to the rotary filler by a conveyor belt. The removed cap is stored in a gravity hopper for semi-finished products in the third zone, specifically in a distribution station [49]. This distribution station is used for dosing the semi-finished caps. A Buffer station is used to create stocks and separate semi-finished products in the production process. The insertion of semi-finished products into the container is detected by an optical sensor, and its operation is controlled by light barriers that are located in front of and behind the separator. The light barriers are able to separate one piece of a semi-finished product while releasing the exit position. The two-axis pneumatic manipulator with a pneumatic chuck in the handling station sorts two types of semi-finished products according to color, and then it stores them in the appropriate container [50]. The following Figure 2 shows the model of the described line, Figure 3 shows the functional overview of production line, and the individual parts of the production line are listed in Table 2.

By decommissioning a production line by using a DDoS attack, significant costs for delay can occur in the factory. In addition, the effects of this single attack can cascade and spread further in the factory if the attacked production line is connected to a larger production system spread across many production centers [51].

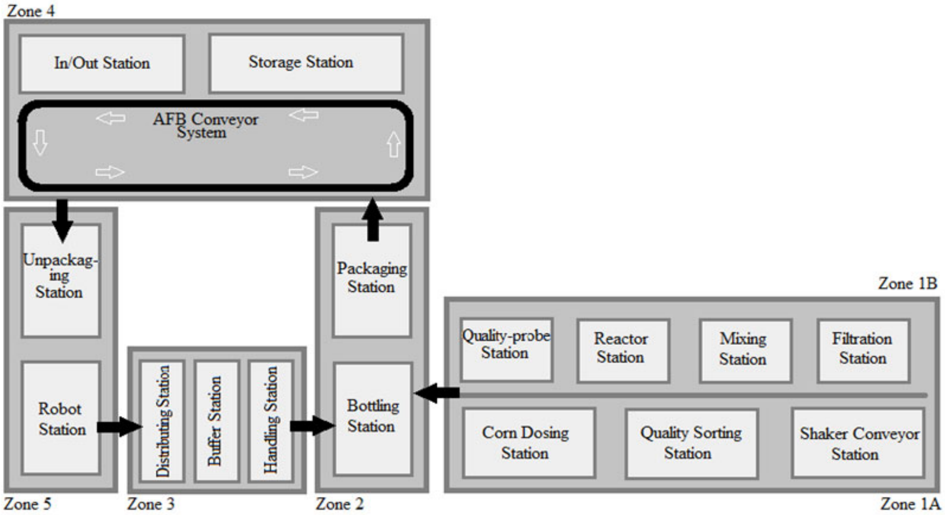


Figure 2. Model of production line.

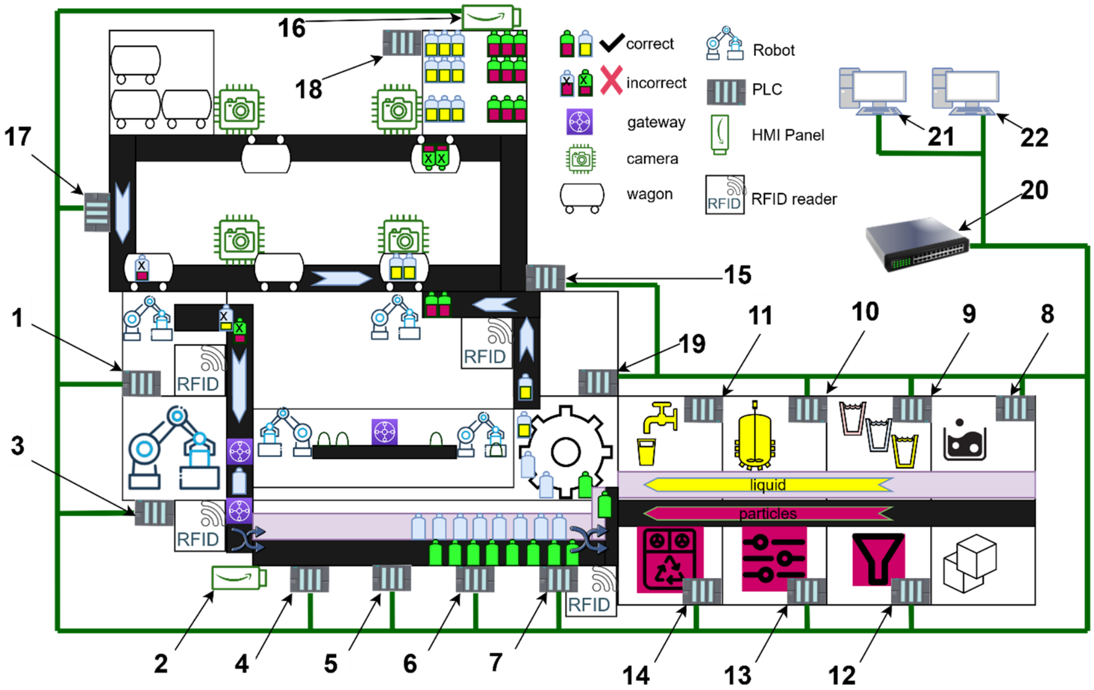


Figure 3. Functional overview of production line.

Table 2. Parts of the production line.

Position	Station	Controller
1	Unpackaging station	PLCS7-314C
2	Bottling station touch panel	Touch panel
3	Robot station recycling	PLCS7-314C/Port 1200
	Robot	Drive Unit/Port 10001
4	Distribution station	PLCS7-314C
5	Buffer station	PLCS7-314C
6	Handling station	PLCS7-314C
7	Bottling station	PLCS7-314C
8	Filtration	PLCS7-314C
9	Mixing	PLCS7-314C
10	Reactor	PLCS7-314C
11	Bottling	PLCS7-314C
12	Quality control	PLCS7-314C
13	Vibration conveyor	PLCS7-314C
14	Dosage	PLCS7-314C
15	IN/OUT station	PLCS7-314C
16	AS/RS station touch panel	Touch panel
17	Packaging station Camera	PLCS7-314C
18	AS/RS station	PLCS7-314C
19	Transport system	PLCS7-314C
20	Switch	Hub
21	PC Control WIN CC	MS Windows
22	PC Control MES	MS Windows

2.3.1. Ethernet Connections

The components of the production line communicate via the Ethernet network. A PLC and HMI panels are assigned a static IP address. Their address space is assigned IPv4 in address class C. PLCs send a message indicating the status of their availability to the network. This activity runs periodically, and it has been found that a condition about the availability is also found in other PLC units. Some parts are connected in a different way, but they are not essential for testing the infrastructure, because they are not connected to the infrastructure switch. PLC devices had open ports 80, 443, and 8080 web, 23 Telnet, 21 FTP, 111 RPC Bind, 389 LDAP, ports 10001 to 10004, and port 10009. The Robot's available ports were: 1200 and 10001, and the Robot used the UDP protocol [52].

2.3.2. Wonderware Manufacturing Execution System (MES) System

A Manufacturing Execution System (MES) is computer systems used in production in order to monitor and document the production process with the raw materials entering it up until the processing. In addition, it also uses the production of the desired product. The MES provides information in order to help manufacturers have a better understanding of how current production conditions can be optimized to improve the efficiency of the production process [53]. The main part that was interesting for the experiments is the Wonderware Historian Server subsystem. This subsystem is a high-performance database of historized information in real-time. The power is combined with the flexibility of a Microsoft SQL relational database (MSSQL) with the speed and compression of a true process historian that integrates the office and manufacturing factory, or any industrial operation. The Historian Server is designed to collect a wide range of data of full-resolution and very high-speed traffic data, ensuring that decision-makers at all levels have the historical information they need to take key productivity initiatives [54]. The Historian Server offers excellent scalability and it is also possible to configure this server as a single system for data collection and aggregation. Additionally, the server can be configured as part of a larger multilevel architecture that offers the ability to implement sophisticated aggregation and replication systems [55]. A Wonderware Historian Client provides reports, and it has the ability to publish historical production information in real-time to the company's website or intranet site by using the Wonderware Information Server.

Thanks to information derived from the Historian Server, it is possible to quickly solve problems, to study the potential inefficiency of processes, and to eliminate the time-consuming process of data localization. Thanks to the Historian client, the delivery and visualization of this information are easy to implement and deploy. These properties were used to monitor the effectiveness of the test scenarios. Each attack scenario that generated alarms was effective. These alarms were detected by the application server at runtime and were stored as historical data in Wonderware Historian for the host engine. In addition, events related to alarm, such as "manual intervention," are stored as historical alarm data as well [56]. The number of alarms can be stored according to the severity level as historical data.

If the Wonderware Historian is turned off, or the network connection is lost while the application is running, the historical data are still stored locally on the computer that hosts the WinPlatform object. When the node of the Wonderware Historian restarts, the data are sent from the local node to the low-priority Wonderware Historian node. If AppEngine loses the connection to the Wonderware Historian node, Wonderware Historian reports poor-quality data to the user. When an object with attributes configured for history is released, the Wonderware Historian Server saves the final data points with poor quality. The operator also has the Wonderware MES/Performance extension, which provides a software solution for collecting, monitoring, and communicating about the performance and the efficiency of devices' information in real-time that are scalable from machine/device-level information to enterprise KPIs (Key Performance Indicators). The MES/Performance provides critical information on outage and device efficiency, which can then take immediate action to improve industry performance and productivity with the most modern operating results. This feature was used to monitor operator intervention during attack scenarios [57].

2.4. Real Network Environment for Performing DDoS Attacks

The article uses a real production network infrastructure in order to implement DDoS attacks. This network infrastructure is connected by two TP-Link switches. These switches connect the production line with the server, IoT devices, and the LAN office subnetwork. The server is equipped with the operating system server, Windows 2008. This server contains a MES system, which is used to manage and control production processes. It contains the subsystem, called Historian, where all historical data about production and production processes are stored. In order to modernize and closely approach the Industry 4.0 concept, the production line in the network infrastructure was also expanded with IoT devices. These added IoT devices are designed to protect the production line and processes, while allowing access from the external WAN network for easy control and informing operators remotely via mobile applications. With this WAN network approach, there is a risk of attacks from the external network, making the entire infrastructure, including the production line, much more vulnerable and also vulnerable to DDoS and DRDoS attacks directed through IoT devices.

The first IoT device, by which the line was extended, is the IoT thermostat. The role of the IoT thermostat is to provide remote temperature control in the room where the production line is located. This device is connected to the network infrastructure wirelessly via Wi-Fi technology with a maximum data transfer rate reach of 54 Mbps. Another IoT device added to the network infrastructure is Fibaro. With its IoT sensors, this IoT device has the task of protecting the production line from a safety point of view against fires and floods, and with the help of a motion sensor, checking the presence of operators in the workplace.

All devices in the network infrastructure are connected via a LAN cable network, reaching a data transfer rate of up to 1 Gbps. In the production line, which is part of the network infrastructure, there are machines that communicate with their PLC device. The brand of these PLC devices is Siemens S7-300 with type 314. Each PLC device is connected by a LAN cable to the switch and the machine. The machines are programmed via PLC devices that control the production process. A part of the test environment is also adding a computer via a LAN cable, which served as a packet generator for performing DDoS attacks on the production line and IoT equipment, Figure 4.

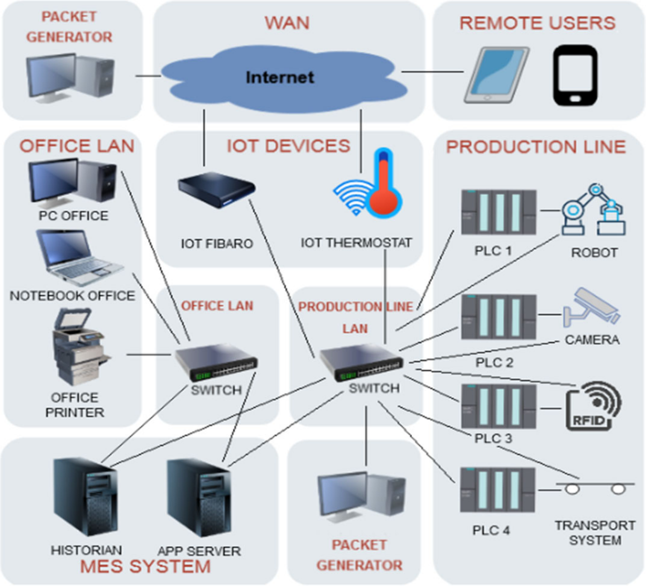


Figure 4. A real network infrastructure used in testing.

3. Results

This chapter describes the experimental attack scenarios used to test the production line communication and vulnerabilities during production with additional IoT devices in a common network infrastructure during DDoS and DRDoS attacks, as well as the results and proposed solutions to best ensure production line operation with IoT devices. The test facility is based on a real-time production environment. As the production line has been expanded with IoT devices to enable communication with the Industry 4.0 concept, which also enables communication via the external WAN environment, it is also more vulnerable to possible attacks from the external network. Another major disadvantage of the Fibaro IoT device and thermostat is that these devices are very poorly secured, as has been demonstrated in recent research on these IoT devices [58]. In order to perform test scenarios in the form of cyber-attacks, the attacks were led from a single personal computer. The personal computer attacked the communication of the network infrastructure composed of IoT devices and the production line in order to test the vulnerability of the production line. This computer employed the Kali Linux operating system. Kali Linux is equipped with Hping3 tools. Hping3 tools can work with network protocols TCP, UDP, and ICMP. The control and operation of packet generation are performed by using commands via the command line. This tool is intended for security analysts who can use this tool to scan the entire network infrastructure and identify security vulnerabilities in the network infrastructure to determine the level of vulnerability risk. In addition, based on the identified shortcomings, the security analysts design effective security mechanisms that reduce the risk of vulnerability [59].

3.1. Scenarios

In performed experiments, five types of attack scenarios were implemented. All attack scenarios were performed with full functionality of the production line and added IoT devices. The attacks were controlled from both the external internet network and the internal LAN network. Two types of attacks were carried out. The first type of attack was DRDoS. It was an indirect type of attack. An attacker sent packets to a nonexistent IoT port on the device. The IoT device reflected these packets to a spoofed IP address that belonged to the selected devices or the PLC controller of the production line. The second type of attack was direct DDoS. An attacker sent packets to individual machines or PLC controllers. The specific types of attacks were used in individual scenarios such as the ICMP echo flood, TCP SYN flood, TCP ACK flood, and UDP flood. The sending of the packets was set to the highest possible speed. The length of these attacks had not been determined. The attacks lasted until the devices, on which the attack was performed, stopped communicating in the network. The goal of the attacks was to compromise the production process by flooding the individual components of the production line, either directly to the components of the production line or indirectly through IoT devices.

3.1.1. Scenario 1

The first attack scenario was implemented as indirectly reflected DRDoS. As shown in Figure 5, the attack was carried out from the external internet network from where the attacker sent ICMP flood packets to the IoT reflectors—Fibaro control unit and smart thermostat. These devices communicate with an external network for their remote management. The attacker routed these ICMP flood packets simultaneously to both IoT devices on their nonexistent port. These reflectors reflected the received ICMP packets on the attacker's spoofed IP address. This spoofed IP address belonged to the devices of the production line. These were the transport system of the production line and its controller (called PLC 4). These devices received the reflected packets from the IoT reflectors until they stopped communicating completely with the network. As shown in Scheme 1, the transport system on the production line stopped communicating with the network after

20 s, and the PLC 4 controller stopped being active in the network after 25 s. After the attack, these devices required a restart, which caused a sudden shutdown in production.

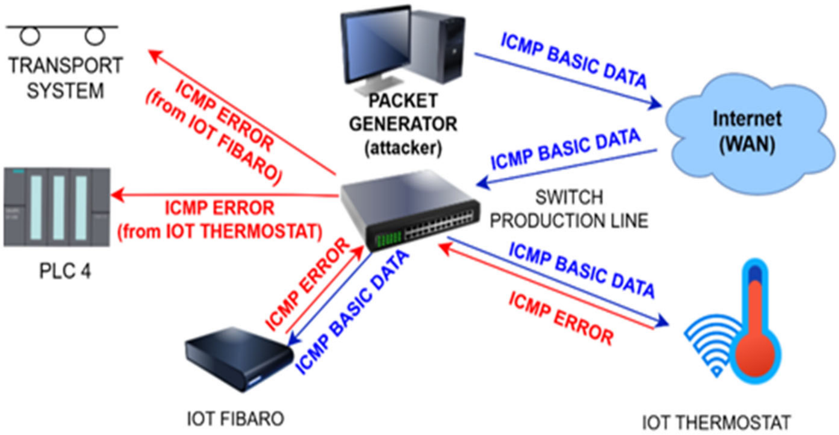
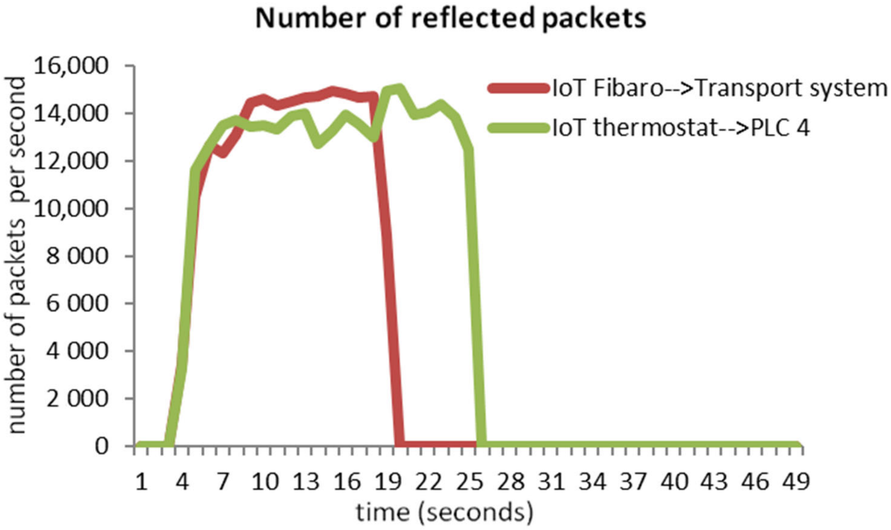


Figure 5. A visual illustration of the scenario 1 attack.



Scheme 1. Scenario 1—communication timeline.

3.1.2. Scenario 2

The second attack was very similar to the first scenario. It was also indirectly reflected by DRDoS. As shown in Figure 6, the attack was carried out from the external internet network from where the attacker sent ICMP flood packets to the IoT reflectors—Fibaro control unit and smart thermostat. This scenario was used to determine whether devices on the production line were vulnerable regardless of their PLC controllers being attacked, but directly only on the devices on the production line. Furthermore, this attack determined how many devices on the production line IoT reflectors can effectively compromise when reflecting packets. In this second scenario, the attacker also directed these ICMP flood packets simultaneously to both IoT reflectors to their nonexistent port. These reflectors reflected the received ICMP packets on the attacker's spoofed IP address. The Fibaro control unit and smart thermostat simultaneously reflected the attack packets on three devices in the production line during the production process, namely on the transport system, on the RFID reader, and on the camera.

As can be seen from Scheme 2, devices stopped communicating in the network without having to attack their PLC controllers. When reflecting attack packets from IoT reflectors, the RFID reader was the first one to stop communicating in the network after 10 s. The camera stopped working after 15 s. Finally, the transport system stopped communicating with the network after 25 s. This number of devices was most effective in reflecting attacks by the IoT reflectors because the speed of packets reflection by these devices was not decreased. However, with a larger number of devices, the attack did not occur until a long time, and therefore, the effectiveness of the attacks was not so effective. After the end of the attacks, it was not possible to work with the devices on the production line, and so it was necessary to restart these devices, which resulted in a significant slowdown in the production process.

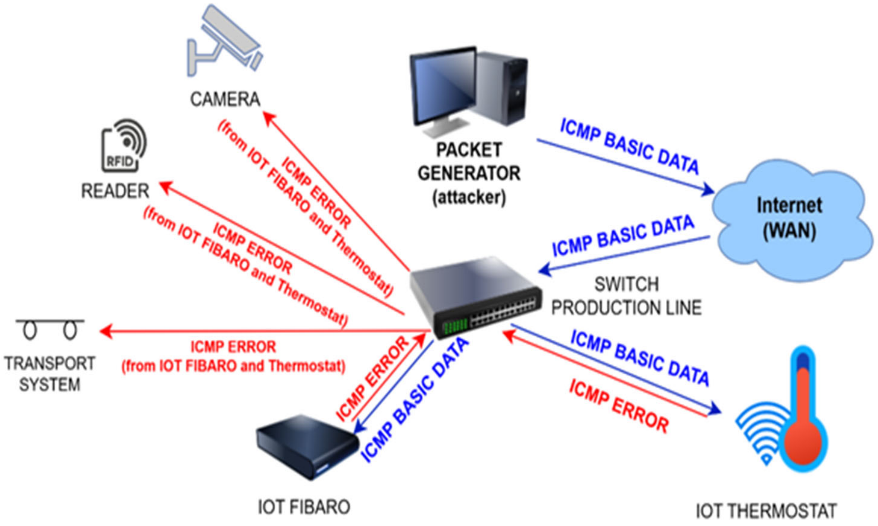
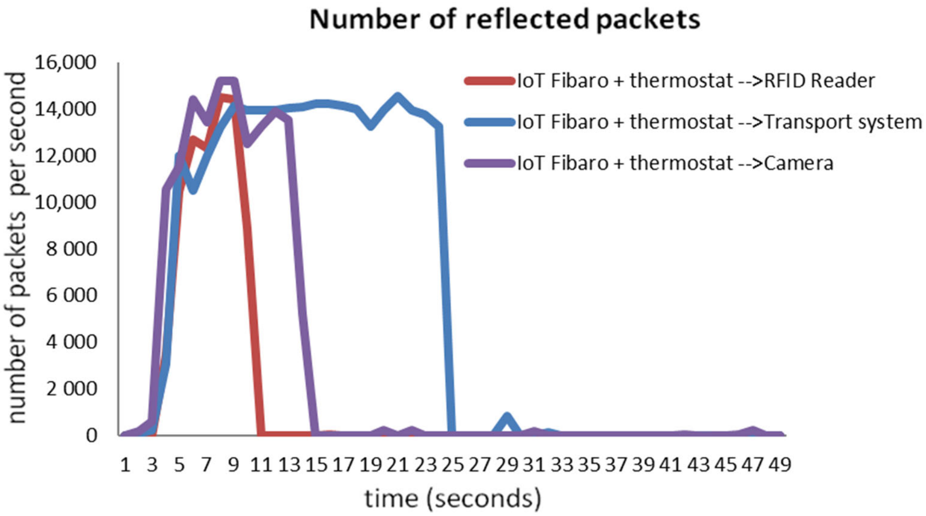


Figure 6. A visual illustration of the scenario 2 attack.



Scheme 2. Scenario 2—communication timeline.

3.1.3. Scenario 3

The production line was also equipped with a Robot, to which the PLC controller transmits the instructions. The Robot and the PLC controller communicate only with each other and work on the UDP network protocol. The PLC controller has open port 1200, and the Robot works with port 10001. The third scenario also describes an indirect reflected DRDoS attack, as shown in Figure 7. The attack was carried out from the external internet network from where the attacker sent UDP flood packets to the IoT reflectors—Fibaro control unit and smart thermostat. These reflectors reflected the received UDP packets on the attacker's spoofed IP address. This spoofed IP address belonged to the production line devices. It was the PLC controller (PLC 1) that instructed the robot and worked on port 1200. The IoT Fibaro reflected the attack packets to the spoofed IP address on port 1200 that the PCL controller contained. Another device on the production line was the Robot. The smart thermostat reflected the attacking UDP packets on the spoofed Robot's IP address with port 10001. As can be seen in Scheme 3, the PLC 1 and Robot devices stopped communicating in the network after 38 s. In this case, only a UDP flood attack on device ports 1200 and 10001 was able to successfully compromise the Robot and the Robot´s PLC device. The ICMP and TCP flood attacks were ineffective for these devices. After a successful UDP flood attack, it was necessary to restart these devices for full functionality.

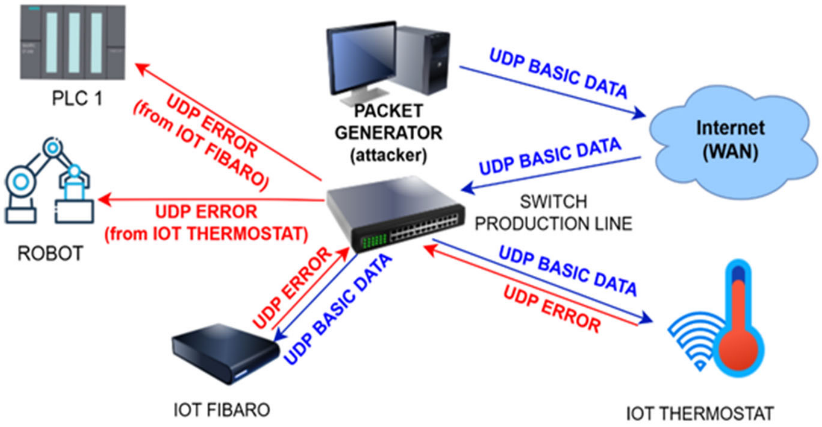
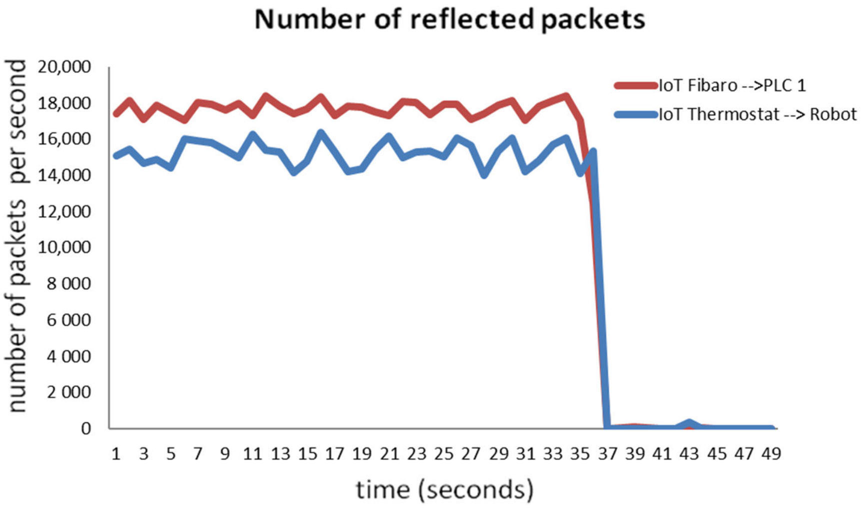


Figure 7. A visual illustration of the scenario 3 attack.



Scheme 3. Scenario 3—communication timeline.

3.1.4. Scenario 4

Unlike the previous attack scenarios, there was a direct DDoS attack implemented in this case. As seen in Figure 8, the attack was performed in the internal network. The attack was carried out by the attacker generating spoofed IP addresses that sent TCP SYN flood packets on the internal network. The attacker sent these packets from randomly generated IP addresses to production line devices, as well as to additional IoT devices. On Scheme 4, it can be seen how the devices on the production line and the additional IoT devices stop communicating in the network. After 11 s of the TCP SYN flood attack, an RFID reader was the first one to stop communicating with the network. After 13 s of the attack, the camera on the production line was disabled. After 14 s, the smart thermostat also became inactive in the network. Then, after 21 s of the attack, the second IoT device, Fibaro control unit, stopped communicating in the network. The last device that was the subject of the attack was a transport system on the production line. This device stopped communicating in the network only after 26 s of the attack. All these devices did not perform any production processes after the attack, so it was necessary to restart the devices. The same situation occurred with the additional IoT devices; they stopped communicating with the operator via a remote application, and the operator had no control over them, so it was not possible to restart them remotely. The TCP SYN flood attack, from randomly generated sources launched on the local network, was effective; all attacked devices were out of operation within 30 s.

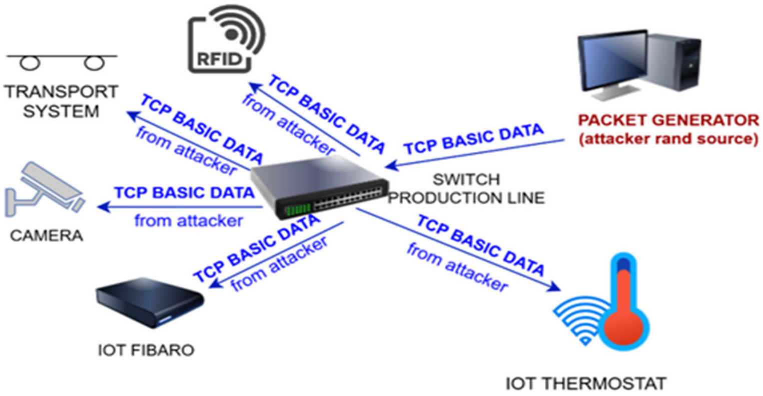
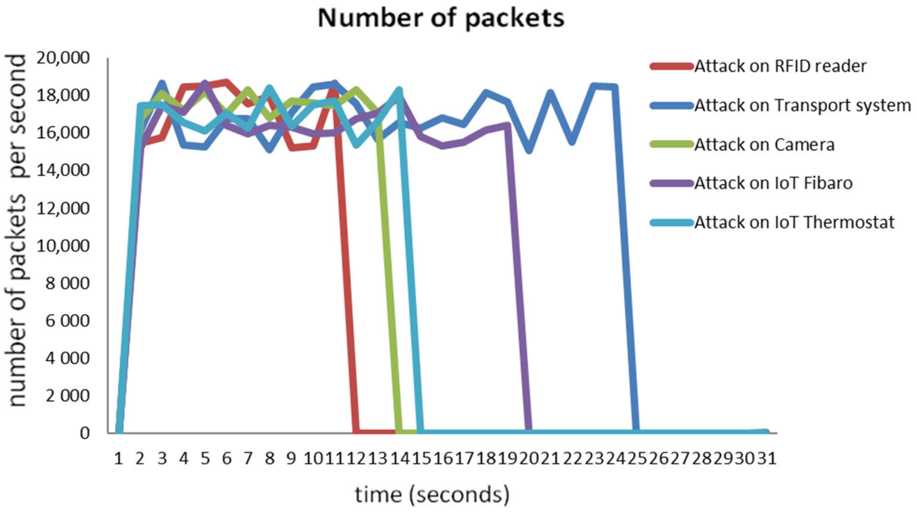

RFID
TRANSPORT SYSTEM
TCP BASIC DATA from attacker
TCP BASIC DATA from attacker
TCP BASIC DATA
PACKET GENERATOR (attacker rand source)
SWITCH PRODUCTION LINE
CAMERA
TCP BASIC DATA from attacker
TCP BASIC DATA from attacker
TCP BASIC DATA from attacker
IOT FIBARO
IOT THERMOSTAT

Figure 8. A visual illustration of the scenario 4 attack.



Scheme 4. Scenario 4—communication timeline.

3.1.5. Scenario 5

The last attack scenario shows a direct DDoS attack. As seen in Figure 9, the attack was carried out from the internal network. In this case, a TCP ACK flood performed the attack. The target was not defined in the attack and, therefore, the attacks were aimed at a random victim in the network infrastructure. A camera and IoT device (Fibaro) were used as accidental victims of the attack. Moreover, during this attack, some of the attack packets overflowed through the Office LAN switch into the Office subnetwork. In the secondary subnetwork of the Office, the overflowing attack packets found a random victim, an Office printer. As shown by Scheme 5, the TCP ACK flood attack packets found their first victim only 8 s after the attack was launched. That victim was the camera on the production line, which stopped communicating in the network after 13 s of the attack. Another accidental victim of the attack was IoT Fibaro that was attacked by the attacker 12 s after the start of the TCP ACK flood. After 15 s of attacking IoT Fibaro, the device was unavailable and did not communicate in the network. After 22 s of launching the attack, the attack packets overflowed into another subnetwork. The subnetwork was connected via an Office LAN switch, through which attack packets flowed and directed to an accidental victim, which was the Office printer. The Office printer was receiving overflowed packets from the subnetwork for 51 s. After this time, the Office printer stopped communicating in the network and could not be requested with any kind of request. The TCP ACK flood attack on the random target thus also affected the device outside the production line.

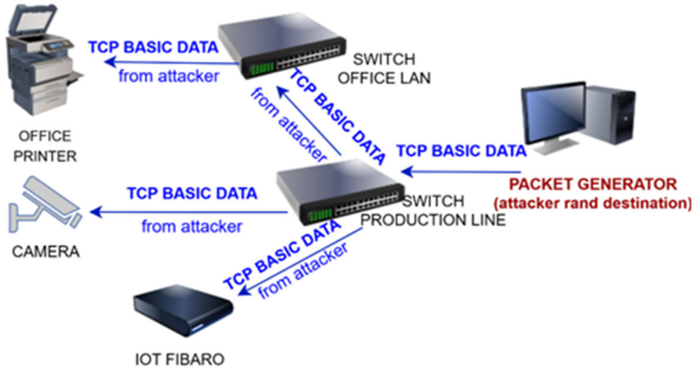
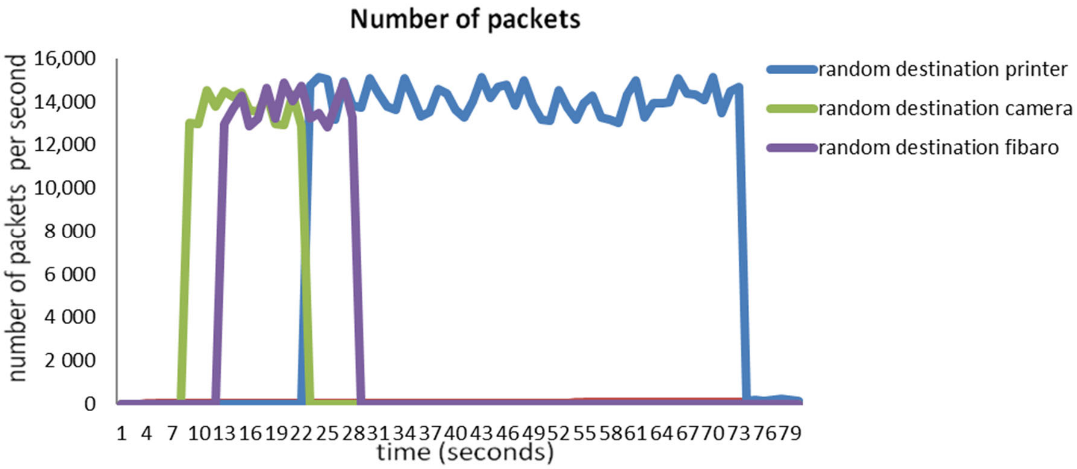


Figure 9. A visual illustration of the scenario 5 attack.



Scheme 5. Scenario 5—communication timeline.

3.1.6. The Overview of Attack Scenarios

As Table 3 shows, reflected ICMP flood attacks were performed for the first two scenarios. These attacks were led by attackers from the external internet network to IoT devices on nonexistent ports, and they reflected attack packets to a spoofed IP address on the production line devices in order to compromise the production process. In the third scenario, a reflected attack from the external internet network was also performed, but it was based on the UDP protocol because the Robot with the PLC controller communicates only on this protocol and on specific ports. The Robot used port 10001, and its PLC controller used port 1200. In the fourth scenario, a direct attack was carried out from the internal local network on the devices on the production line and also on the additional IoT devices. The attack was performed from randomly generated attacker's IP addresses, which sent TCP SYN flood attack packets to the devices. The fifth scenario demonstrated a direct attack from the internal network as well. In this case, a TCP ACK flood was performed. The TCP ACK flood attacked random targets in the network infrastructure. The individual attack scenarios have proved that they can do great damage to the production line together with its IoT devices. The next chapter presents the problems that arose during the performed attacks during the actual production process.

Table 3. A brief overview of the types of attacks that were used in each scenario.

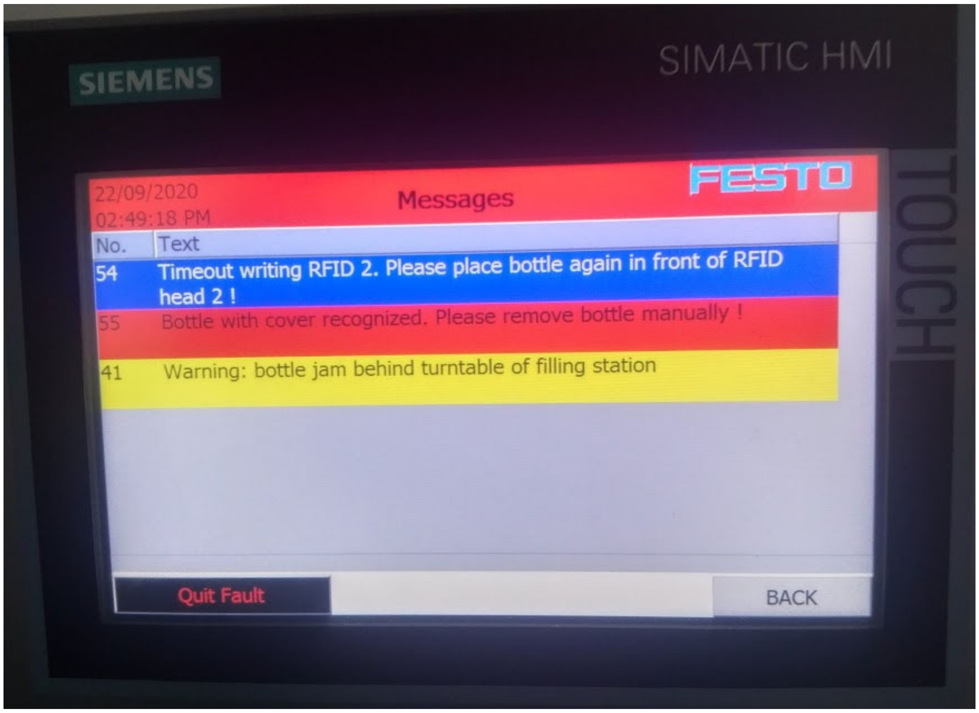
Scenario	Type of Attack	Attack
Scenario 1	Reflected (DRDoS)	ICMP flood
Scenario 2	Reflected (DRDoS)	ICMP flood
Scenario 3	Reflected (DRDoS)	UDP flood
Scenario 4	Direct (DDoS)	TCP SYN flood
Scenario 5	Direct (DDoS)	TCP ACK flood

3.2. Impacts of DDoS and DRDoS Attacks on the Production Process

In the previous chapters, five scenarios of implemented attacks were demonstrated. The attacks were performed during the full operation of the production process. The goal of the targeted attacks was to compromise the production process on the production line. The attack scenarios have clearly shown that production facilities and additional IoT devices stopped communicating in the network infrastructure and became unmanageable, which resulted in the following problems:

1. The first serious problem in the production during the attack was the failure to transport the lid and storing it in the bottle. The carrier stopped in the middle of the track or did not complete the arm release operation when placing the lid on the bottle. At the same time, the production process was interrupted, which ensured the closing of the bottles. The production operators assumed that the occurred error was of mechanical origin, and error messages on the panels led them to misidentify the problem. A cap position placer sensor reported an error, as shown in Figure 10. The cause was the PLC unit of the lid carrier. The PLC, that controlled this operation and was hit by an attack, stopped working during the lifting of the lid while moving along the rail in the middle of the track. The operators started to solve the transport arm and its lubrication quality, as it looked as if the operation of moving along the rail always got stuck at the same point. The error message led the operators to incorrectly evaluate the reasons for the failure.
2. RFID readers are used in the production line, where the second problem could be observed. It was created while reading RFID codes, which can also be seen in Figure 10. The attack caused the RFID reader at the Robot's output to incorrectly read and sort the bottles because the signal to the sorting gate was missing. With this error, the expected exception occurred on the control panel, and the operators tried to start the production process by repeating the last step and reloading the data from the element that was reloaded by the RFID sensor. However, this approach did not lead to anything, and so the operators reported a hardware problem with the RFID reader. Each RFID sensor was prone to failure, and its error rate during the attack generated a large number of different alarms. If the RFID1 reader was hit at the inlet of the bottle into the production line and a reading timeout occurred, it was possible to observe

the discharge of the wrong type of bottle, and another type of material was filled into an incompatible package. For example, granulate was dosed into the liquid container or vice versa. When the RFID2 output did not load the passing bottle, an alarm appeared that informed about the jamming of the rotary table of the filler. As this situation can realistically occur, the production operators tried to read the RFID code by manually attaching the bottle. In the case of high overflow, this activity was unsuccessful, and the operators' assessment was an incorrect reader or an incorrectly set conveyor belt speed. Additional readers are installed in the recycling station, where the status at the input and output of the recycling station was incorrectly identified. Sometimes, the Robot did not run the material recycling action at all, and the contents were not soaked up, or the bottle did not open. This disrupted the work cycle.



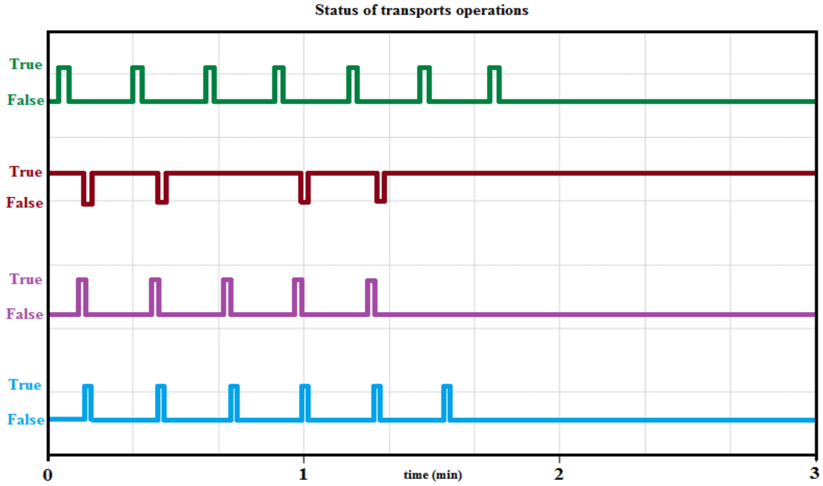
22/09/2020
02:49:18 PM
Messages

No.	Text
54	Timeout writing RFID 2. Please place bottle again in front of RFID head 2 !
55	Bottle with cover recognized. Please remove bottle manually !
41	Warning: bottle jam behind turntable of filling station

Figure 10. Error report on HMI panel.

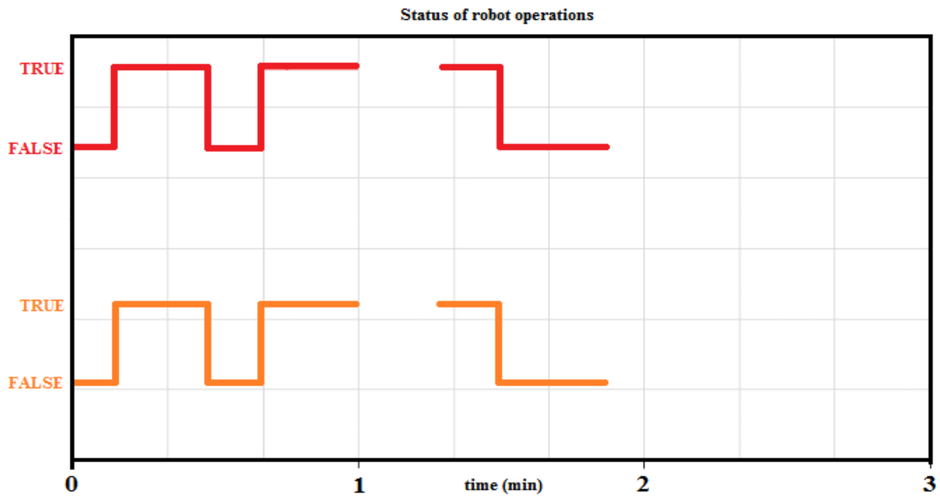
3. The third problem arose when controlling the movement of the conveyor belt, where the PLC control unit ceased to be able to operate the drives. The result was a malfunction of the conveyor belt. Processed products could not be transported to the warehouse, and alarms were generated on the production line. The error, generated in this way, appeared to the operators as a technical failure of some component of the production line without a more precise specification. The operators determined that there was a problem with the power supply to the motors that provided the movement.
4. The camera, which was another element in the production line, was not able to read QR codes from the transport trolley. As the result, the conveyor belt was flooded with fully loaded pallets, and therefore, unloading and loading did not take place. The pallets were still running on the conveyor belt, but it was not until the monitor of unloading reported an error message, stating that no goods had been delivered in the last time interval. The operators again reported a hardware problem with the camera. These operations were usually visible in the MES system, and so the data provided by the Historian system were examined, where this type of problem manifested itself visibly on Scheme 6. Again, the operators deduced a hardware problem with the camera and later with the production line infrastructure. Specifically, the intention

was to replace the switch. After exchanging for the same type after repeating the experiment, it was able to successfully implement this error again.



Scheme 6. Historian report during the attack for wagon transport.

5. The Robot and its PLC were very stable elements of the production line, and because of this, the result after attacking them was only slightly visible. One of the observed degradations was slowing the work down or jamming the Robot's movement. The Robot and its control PLC failed only in a targeted attack on precisely specified ports and a defined UDP protocol. After the attack, the Robot stopped in the middle of the operation, but this moment could have occurred at any time. The Robot's tools fell down, the recycling operation stopped, and the Robot stopped while the arm was moving.
6. The MES system was able to continue to perform the assigned task, but the problematic area was mainly data collection from the production line. In this case, there was the opportunity to analyze the behavior of the MES system, called Wonderware. The basic component that is responsible for data collection is the part of the Historian. This is divided into two sub-systems. The first sub-system takes care of the communication with the control elements of the production line. The second one is responsible for processing and writing data to a relational database, in this case—Microsoft SQL. During the attacks, it was possible to monitor changes in the quality of read data or their complete outages. These events were then interpreted by the MES system, and alarms were generated that could confuse production operators. A change in the quality of the collected data can be understood as their delayed processing, a change in the data parameters on the MES side of the system. Although the Wonderware implementation uses an internal protocol to ensure data delivery, the affected infrastructure has made it impossible to read real data, so the Historian only generated empty default data in the periodic cycles. In the case of loading error, it was possible to observe nonexistent values for the given parameters. Some outages were visible only with a considerable delay compared to the real state on the production line, as can be seen in Scheme 7.



Scheme 7. Historian report for Robot operation during the attack.

7. All listed attack scenarios were later wrongly identified as hardware failures of individual production line components, or they were assessed as incorrectly entered production process parameters. The initial analysis of the problem, which relies only on monitoring the production process, was insufficient and led to incorrect conclusions. In the further search for failure, the components were randomly selected and, finally, a system and production line restart was performed. Even the monitoring of the infrastructure did not lead to the finding that it was an attack from the outside, as the reflected packets were visible in the network as packets that were generated by IoT devices. These were identified only with a delay, and also, they were first evaluated as a hardware error or incorrect configuration. For network monitoring, their behavior showed attributes of delicately used IP addresses or incorrectly set network parameters.

The test attacks that were performed had a degrading impact on the production process. Each attack scenario was initially misidentified and almost always assessed by an experienced operator as a failure that could be justified by a hardware problem with the components. Incorrect conclusions cannot be attributed to the operator’s lack of experience, as the behavior of the production process during attacks appeared to be a normal operational failure. The monitoring system and the integration of the production process in the MES system did not reveal the unusual behavior of the production line. To use an example, all of the carts remained full, and the camera was unable to read QR codes, which led to production delays. The variability of the problems and the randomness of time, as demonstrated by the attack scenarios, show that it is possible to marginally influence the production process without clearly identifying the source of the problems. This also points to the fact that an attacker does not have to know the infrastructure to be compromised or to be able to degrade the production process. In addition to the scenario for the Robot and its PLC, this is generally true as it has been proved in the tests. In this area, the IoT devices security has been identified as a priority part, which should be integrated into the production line.

4. Discussion

Using the scenarios and subsequent analyses of the attacks, the behavior of the production line in terms of communication resilience, and possible threats or disruptions to the production process, that may occur with the added IoT sensor was discovered.

Proposed Solutions to Mitigate the Effects of DDoS Attacks

In previous chapters, it was clearly shown that DDoS attacks can cause major problems and huge damage in the network infrastructure and, especially, in the production process. Based on the findings, it was necessary to design appropriate solutions that prevent or mitigate the DDoS attacks into the network infrastructure. There was a comprehensive approach needed to address this issue. It was decided to focus on two possible proposals that take into account the real deployment of IoT devices in the manufacturing factory, considering the existing solution, which will not jeopardize the manufacturing process. It can be assumed that the company is not willing, or it is not possible, to change the existing operation, either due to the budget or due to the fact that the proposal would jeopardize the delivery of agreed contracts.

The first proposal is based on the assumption that the company has a running production process and production line that has been operational for a long time, and no change of infrastructure is possible; however, at the same time there is a need to integrate IoT devices in order to increase the line quality or efficiency. The second solution is a proposal on how to change the infrastructure in order to make the integration of IoT devices into the production process safe.

In the case of an existing production line, it is risky to change the existing infrastructure that is deployed in the company. This operation can cause unwanted production interruptions. In such an operation, the risk management takes into account whether the added value of the IoT devices has a significant impact on the management, or whether the integration of the device itself does not pose a risk to the production process. A transparent bridge firewall was chosen as the first solution, taking into account the impossibility of changing the infrastructure. A transparent bridge firewall is a device that allows the deployment of the protection of added IoT devices and effectively isolates them from the existing infrastructure. If the communication of IoT devices is known, rules can be defined that will ensure the communication requirements and all other communication will be blocked. The protection and the price of the solution itself should not exceed the price of the IoT device, and for this reason, the use of a mini-computer seems to be a suitable solution. For simplicity, it was decided to configure such a device with a Raspberry Pi mini-computer that has two network cards. These cards have been configured for transparent bridge wiring, Figure 11. The bridge works at the MAC (Ethernet address) level, and the communication is forwarded between the two cards. It could be said that it is a switch, but it contains only two ports. The added value is that the bridge connected in this way can also serve as a firewall. A prerequisite for such an implementation is that both network cards must support promiscuous mode. This requirement arises from the nature of the implementation, as the bridge will have to be able to work with other addresses from the network, and the network communication itself will not be rewritten to the addresses of the physical cards that are physically present on the mini-computer. The bridge, implemented in this way, is called transparent, precisely because of this feature that it may not be directly visible in network traffic. The Linux operating system for ARM Armbian processors was used. The operating system must have the bridge module enabled. The interesting thing about this solution is that the bridge itself does not even have to have an assigned IP address. Usually, this setting is not used, as it is good to have access to the firewall and possibly make the necessary configuration or monitor the status of the device.

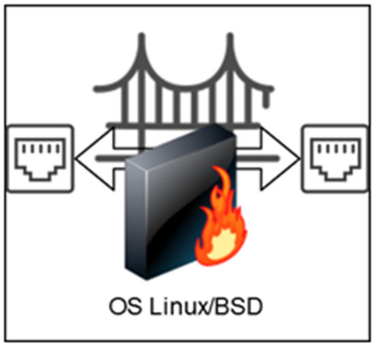


Figure 11. Configuration for transparent bridge for OS Linux.

When implementing and configuring the mini-computer, it is necessary to create rules for Iptables Table 4.

Table 4. IPtables firewall configuration.

Rule Description	Rule Definition
default rules check established, related	-F FORWARD -P FORWARD DROP -A FORWARD -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 -m state --state INVALID -j DROP
enable DHCP UDP packets if needed	-A FORWARD -p udp --dport=67:68 --sport=67:68 -J ACCEPT
Limit ICMP	-A FORWARD -p icmp -m limit --limit 4/s -j ACCEPT
Rule description	**Rule definition**
Force SYN packets check	-A FORWARD -p tcp ! --syn -m state --state NEW -j DROP -A FORWARD -p icmp --icmp-type 8 -d 0/0 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT -A FORWARD -p icmp --icmp-type 0 -s 0/0 -m state --state ESTABLISHED,RELATED -j ACCEPT
Reject spoofed packets	-A FORWARD -s 10.0.0.0/8 -j DROP -A FORWARD -s 169.254.0.0/16 -j DROP -A FORWARD -s 172.16.0.0/12 -j DROP -A FORWARD -s 127.0.0.0/8 -j DROP -A FORWARD -s 224.0.0.0/4 -j DROP -A FORWARD -d 224.0.0.0/4 -j DROP -A FORWARD -s 240.0.0.0/5 -j DROP -A FORWARD -d 240.0.0.0/5 -j DROP -A FORWARD -s 0.0.0.0/8 -j DROP -A FORWARD -d 0.0.0.0/8 -j DROP -A FORWARD -d 239.255.255.0/24 -j DROP -A FORWARD -d 255.255.255.255 -j DROP
Stop smurf attacks	-A FORWARD -p icmp -m icmp --icmp-type address-mask-request -j DROP -A FORWARD -p icmp -m icmp --icmp-type timestamp-request -j DROP -A FORWARD -p icmp -m icmp -j DROP
Drop all invalid packets	-A FORWARD -m state --state INVALID -j DROP -A FORWARD -m state --state INVALID -j DROP -A FORWARD -m state --state INVALID -j DROP
Block sync flood	

	-N udp-flood -A FORWARD -p udp -j udp-flood -A udp-flood -p udp -m limit --limit 50/s -j RETURN -A udp-flood -j LOG --log-level 4 --log-prefix ‘UDP-flood: ‘ -A udp-flood -j DROP
Enable other traffic which is needed—ssh/web	-A FORWARD -p tcp -s 0.0.0.0/0 -d X.X.X.X/32 --dport 80 -j ACCEPT -A FORWARD -p tcp -s 0.0.0.0/0 -d X.X.X.X/32 --dport 22 -j ACCEPT

When testing the vulnerability, the first tested thing was whether or not the mentioned attack scenarios will have any impact on the production process at all. Only when the attacks were seen as observable and manifested was a method of measurement proposed. The two problems were solved. The first problem was the time when the operation of the production line device failed, and whether such a manifestation occurred at all. The second problem was finding out when a device that was under attack, stopped communicating with the environment, if at all.

The first task seemed to be very simple. It is clear that any cessation of the production operation is observable and, thus, the basic monitoring system will sooner or later detect a failure in the affected sector of the production line. It was assumed that in the Historian system, there would be an observable decrease in the collected data or directly visible outages of the production process. During the implementation, anomalies that did not always accurately show the actual state of the production process were encountered, as the elements of the infrastructure that ensured data transmission entered the network communication. The switches that provided the Ethernet network communication were supplied with the production line. The production line uses several protocols such as Modbus, which is connected via proxy to the Ethernet converter. The data collection does not have to run in real-time. In this case, it has been observed that the data, which the Historian system processes, may be missing. However, if the attention is not focused directly on the error rate of reading data, it is easily possible that the attack that is taking place is not clearly observable. In other words, unless it is known that some type of attack is taking the place, the Historian system is not intended to report erroneously read data, only in the case if something else is explicitly defined in the system.

Monitoring of production lines is usually performed on the basis of events that currently take the place. In the case of a production line without Industry 4.0, considering the fact that all elements of the production line perform a given task, but they do not share data on the state of the production process with a description of the product life cycle, it is almost impossible to determine the actual state of the operation. Only the input and output of each operation can be monitored. What can also be monitored as well is whether the activity lasts for a defined time or not. If one of these three parameters does not occur, an alarm is generated, and the indication of an error status is then possible. Only when this condition is labeled as an error event in the Historian database can it be resolved by the operator. In the case of Industry 4.0, there are several options to choose from. One of the choices, in particular, would be the ability to monitor the events generated during the production process in order to be able to rely on the accuracy of the data. The problem was solved by defining the time in which the given operation on the production line was to be performed. Now, it is good to take into consideration that the start of the production or the treatment of the failure must be removed from the preparation of the times that are used to set-up these time limits. After defining the duration of the operation and their integration into the MES system, it was possible to use data from the Historian database and, without interfering with the production system, to define views at the database level, which will read data and evaluate the operation times. If it is available and possible to integrate such a solution into a MES or SCADA system, it is greatly recommended to modify the system. If this is considered risky, it is possible to make “read-only” access to the database and rely just on the monitoring system, without the risk of any modifications.

The second part of the problem was to identify when the components of the production line stopped working. It was decided to first monitor network traffic on one of the

PLC devices. A mini-computer with three network cards was used, and it was configured as a NetFlow collector alongside a transparent bridge, Figure 12.

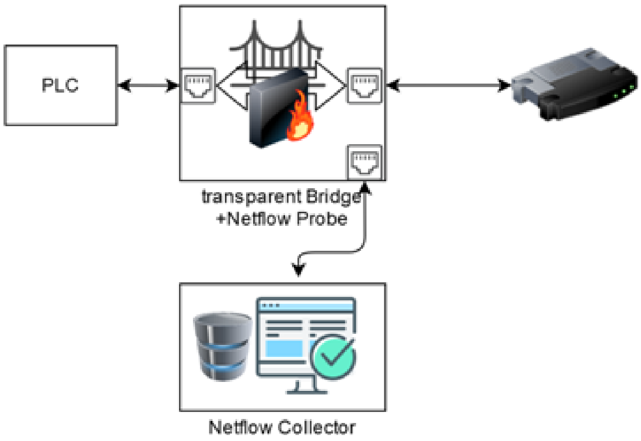


Figure 12. Scheme of net flow data acquisition.

Two network cards are connected in transparent bridge mode, and the third network card is used for NetFlow traffic, which is sent to the computer. This computer serves as a NetFlow collector. This way, basic data about network traffic can be obtained even when the NetFlow version 5 is used. Therefore, it is able to see how the devices communicate. The production line uses the IPv4 version, so the NetFlow basic parameters are obtained. These basic parameters are:

- a source IP address
- a destination IP address
- a source port for UDP, TCP, and other protocols of 0
- a destination port for UDP, TCP, and ICMP of type and code, and for others, 0
- an IP protocol
- an IP type of service
- recording date and time
- an amount of data transferred.

In case it is necessary to implement a NetFlow collector for IPtables configurations, it is important to implement these rules for NetFlow, as the first ones, in order to avoid the loss of the packets. This will prevent modification of the packets by additional rules. Next, it was necessary to configure netflow.conf and to specify to which IP address and port the data will be sent. It is necessary to define the IP address and the version of the protocol. If possible, the use of NetFlow protocol version 5 is highly recommended. In Linux distributions, these settings are located in the netflow.conf configuration file. Either the NetFlow must be enabled in the operating system core, or the ipt_NETFLOW core module must be enabled. The destination IP address can also be defined as a parameter when loading the core operating system module. The configured mini-computer was used to measure and to record packets. The advantage of such a solution is the simplicity and the accuracy of the measurement and the possibility to monitor how the attack takes place, with the fact that it is possible to see when the device stops responding and the network communication visibly decreases. Another advantage would also be that in the existing infrastructure, it is possible to track individual devices without a high risk of production line malfunction. The disadvantage of this solution is that each PLC should have its own mini-computer, which in the case of the presence of a switch, is eliminated. The switch has the support of the NetFlow, and the network communication can be mapped directly by using the switch. The second disadvantage of the mini-computer is that there can be a problem with the attack and a lack of system resources. This can lead to distortion of the values of the examined network communication. In this case, this phenomenon during the network communication produced by the infected IoT devices was not encountered. Another disadvantage may also be the complexity of configuring the mini-computer or securing multiple network cards on the mini-computer.

After creating a network map of communication of production line components, the step, where we replaced the switch with a manageable one with support for L2 (switch) and L3 (router), was proceeded. One problem occurred because the network communication with default permissions did not work. In this case, however, the reason was easily revealed, as the data from the NetFlow showed the need to enable multicast UDP communication that the PCL S-300 needed. As there was a switch with L3 support in hand, where it was possible to apply firewall settings, it was also possible to make the exact collected communication rules from the basic collected communication, which limits the LAN options to the necessary protocols and ports to ensure the maximum security. Basically, the only communication necessary is allowed, and everything else is forbidden. Therefore, it is good to see if, for example, some parameters such as the enabled multicast do not pose a potential security risk.

From the findings, it was possible to create two types of possible solutions. The starting point is the same for both of them: It is necessary to get to know the communication that takes place on the production line, and the expansion of IoT devices is also wanted. In principle, it is quite simple to put the transparent bridge firewall solution ahead of IoT devices and allow only the necessary communication in both directions to the WAN. The IoT device connected in this way, although vulnerable, will not jeopardize the production process, only if the mini-computer does not have security issues itself. The second solution is the need to intervene in the infrastructure of the production line, either by configuring a manageable switch or by replacing the flood switch with a manageable one and applying the prepared rules. There are rules prepared for the production line in Table 4 that are the same as for the firewall. These rules are applicable for the configuration of a transparent firewall, as well as a manageable switch. However, the solution, that uses a switch, cannot ensure the failure or attack of the IoT device from the WAN network. It can only isolate the source of problems that may occur at the network level of the LAN production line. The actual connection of the IoT device to the WAN network can also be handled by similar rules as it was stated for the transparent bridge, but it can also be handled on devices that provide a connection to the internet, only if they allow it, however.

As could be seen, the attack can be implemented from the external and, possibly also, from the local network. The prepared scenarios that were used have proved this impact. However, the overall impacts still depend on the quality of the integration and the components used in the production line. On the other hand, it needs to be added that for some devices, a random attack without the knowledge of the infrastructure would probably not be as effective as could be observed in Robot attack scenarios. However, the vulnerability of the production line, which is extended by IoT devices, is clearly possible. As part of the measures it tried to implement, there were the limits of the mini-computer's performance encountered. Specifically, these were the DRDoS attacks. There was a case where the actual collapse of the transparent bridge firewall was observed. It was assumed that if there is a need to block DDoS attacks from IPtables, it is very important that the rules of IPtables are processed fast enough and that their performance is sufficient to prevent this type of attack. Attacks based on TCP DDoS usually have a high packet rate, and the device that is supposed to filter these packets could shut down and fail. The system that is to perform this function must be sufficiently dimensioned, and here comes the question of the suitability of the mini-computer, whether it is able to handle this function or not. Most sources report the treatment of DDoS attacks on the INPUT side of IPtables. The problem is that the INPUT rules are processed after the PREROUTING, and the FORWARD rules and the INPUT chain rules are applied only after going through the previous two sets of rules.

This can result in a system overload. The first chain that handles packets in IPtables is the PREROUTING chain, but that one is not possible to use for a configuration of a standard filter table. However, it is possible to use a mangle table, which is usually used for packet fragmentation. It is possible to apply almost all rules for this table, as well as for the INPUT table. For the first solution for a transparent firewall bridge, the FORWARD rules were used. However, this can be further improved by just changing the PREROUTING rules by using the mangle table. Both rules for the IoT device worked; considering the performance of the IoT device, the solutions of both its vulnerability and power with which it can reflect the packets are satisfactory.

It should be noted that by default, the production line was delivered with two simple switches that have been replaced by a manageable one. After an analysis, it came to the conclusion that the switches in the original configuration were a weak point, and the communication on their ports could be blocked. Therefore, scenarios were repeated that partially confirmed this. There were two states. In the first one, the ports from which the attack took a place were blocked. In the second state, the ports on which the attack was targeted were blocked. Both cases led to a malfunction of the system. When blocking the ports of the targeted switch, the attack occurred especially when the device was able to respond to packets. During the tests, the production process in each scenario had to be restarted because the partial resets of individual systems were not enough to resume production.

The process of the scenarios was also very remarkable from the point of view of the production operator, where the problems caused by the attacks were almost never identified as a cyber-attack. All conditions and error messages generated by the attacked production line were known. There was only a problem in eliminating these errors. Documented procedures did not resolve the error conditions. Production operators considered the situation to be a new experience. In addition, they stated that, based on the behavior of the system, they would not be able to detect that this was a security breach. During DRDoS attacks, although the basic performance monitoring of the network was applied, it was only after searching for the source that the excessive data flow from the IoT device was identified; however, it was still not possible to identify the source that generated this traffic. These problems were finally solved by replacing the switch and turning the NetFlow on, and expanding the network monitoring of the production line.

From the performed tests, it is possible to recommend providing an IoT device that is to provide data to another network separately. The solution can be quite simple if the network infrastructure can be configured. However, if it is impossible, or there is a concern to change the settings of the production line, as could be seen in this case, it is possible to extend the communication of IoT devices with a module that provides network communication, and also, possibly provides data for monitoring network activities.

The recommendations, resulting from the research performed during the introduction of industrial IoT devices into the production line from the point of view of defense against DDoS attacks, are mainly:

1. If it is necessary to make the IoT device freely available on the internet, its protection and isolation must be ensured by appropriate rules, which are defined in the firewall rules table. The firewall rules must be applied, if necessary, by using the proposed transparent bridge firewall.
2. Performing analysis, mapping a network communication, and not relying only on the documentation that comes with the production line, for example, the method of mapping the network communication, as it was implemented in this study by using the net flow technique.
3. Restriction of communication rules to what is necessary, if possible, within the production line.

4. Ensure that the security systems that are used to protect devices are regularly updated to address potential security vulnerabilities to ensure the maximum possible removal of any known vulnerabilities.
5. To include the network monitoring with data from the net flow and possibly detect any unusual activity during the production process.
6. To specify whether IoT devices must necessarily provide data to the internet, or if it is possible to use another infrastructure and allow the integration into, for example, the intranet, where access is conditional like in a VPN connection, thus indirectly providing access to IoT devices.

The mentioned methodology and tests were applied in a real production process. The advantage of collected information provides a possibility to analyze the communication of added IoT devices. Firewall protection against attacks was also applied and it was discovered to already be running attacks on the IoT device used in the production line. Feedback from customers and tests in real environments gave us commitment of the described design.

Testing and analysis of other defense methods against DDoS attacks and their impacts on the production line are planned as future research. Many of the techniques for detecting, mitigating, and defending against DDoS attacks listed in Table 1 appear to be suitable methods applicable to mitigating DDoS attacks on a production line. An example is the method described in [15] using DNS requests and machine learning, where machine learning is able to detect reflective attacks, and applying this method to the production line can increase the resistance of the production line against DDoS attacks.

5. Conclusions

In this paper, an experimental analysis of the vulnerability of a production line enriched with industrial IoT devices during a real DDoS attack was presented. Two different types of attack were tested—direct DDoS flood attack and DDoS reflective attack. The experimental analysis highlighted the increase in production line vulnerabilities through the implementation of modern IoT systems. The tested DDoS attacks caused the production line to malfunction during its production process, which could potentially endanger the system itself and, indirectly, the people who interact with it as well. Finally, the implemented countermeasures, as well as possible strategies, for protection and mitigation of a DDoS attack on the production line with integrated industrial IoT devices were summarized.

Author Contributions: Conceptualization, T.H., P.S. and L.H.; methodology, T.H., P.S. and L.H.; validation, T.H., P.S. and A.V.; formal analysis, T.H., P.S. and L.H.; investigation, T.H., P.S. and L.H.; resources, P.T. and M.K.; writing—original draft preparation, T.H., P.S., L.H. and A.V.; writing—review and editing, T.H. and L.H.; visualization, T.H. and P.S.; supervision, L.H., P.T. and M.K.; funding acquisition, P.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Scientific Grant Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic and the Slovak Academy of Sciences, grant number VEGA 1/0272/18 Holistic approach of knowledge discovery from production data in compliance with Industry 4.0 concept.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Derhab, A.; Guerroumi, M.; Gumaei, A.; Maglaras, L.; Ferrag, M.A.; Mukherjee, M.; Khan, F.A. Blockchain and Random Subspace Learning-Based IDS for SDN-Enabled Industrial IoT Security. *Sensors* **2019**, *19*, 3119, doi:10.3390/s19143119.
2. Bucci, G.; Ciancetta, F.; Fiorucci, E.; Fioravanti, A.; Prudenzi, A.; Mari, S. An IoT condition monitoring system for resilience based on spectral analysis of vibration. In Proceedings of the IEEE International Workshop on Metrology for Industry 4.0 & IoT, Roma, Italy, 10 July 2020; pp. 38–43, doi:10.1109/metroind4.0iot48571.2020.9138177.

3. Jiang, X.; Lora, M.; Chattopadhyay, S. An experimental analysis of security vulnerabilities in industrial IoT devices. *ACM Trans. Internet Technol.* **2020**, *20*, 1–24, doi:10.1145/3379542.
4. Sari, A.; Lekidis, A.; Butun, I. Industrial Networks and IIoT: Now and Future Trends. In *Industrial IoT*; Springer: Cham, Switzerland, 2020, 3–55, doi:10.1007/978-3-030-42500-5_1.
5. Prinsloo, J.; Sinha, S.; von Solms, B. A Review of Industry 4.0 Manufacturing Process Security Risks. *Appl. Sci.* **2019**, *9*, 5105, doi:10.3390/app9235105.
6. Chhetri, S.R.; Rashid, N.; Faezi, S.; Al Faruque, M.A. Security trends and advances in manufacturing systems in the era of industry 4.0. In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Irvine, CA, USA, 13–16 November 2017; pp. 1039–1046, doi:10.1109/ICCAD.2017.8203896.
7. Tuptuk, N.; Hailes, S. Security of smart manufacturing systems. *J. Manuf. Syst.* **2018**, *47*, 93–106, doi:10.1016/j.jmsy.2018.04.007.
8. Frey, M.; Gündoğan, C.; Kietzmann, P.; Lenders, M.; Petersen, H.; Schmidt, T.C.; Wählisch, M. Security for the Industrial IoT: The case for information-centric networking. In Proceedings of the IEEE 5th World Forum on Internet of Things (WF-IoT), IEEE, Limerick, Ireland, 15–18 April 2019; pp. 424–429, doi:10.1109/WF-IoT.2019.8767183.
9. Apiecionek, L.; Großmann, M.; Krieger, U.R. Harmonizing IoT-Architectures with Advanced Security Features-A Survey and Case Study. *J. UCS* **2019**, *25*, 571–590, doi:10.3217/jucs-025-06-0571.
10. Knudsen, A.H.; Pedersen, J.M.; Sørensen, M.A.M.; Villumsen, T.D. *Security in the Industrial Internet of Things, in Cybersecurity and Privacy: Bridging the Gap*; River Publishers: Gistrup, Denmark, 2017; pp. 119–134.
11. Shiaeles, S.N.; Katos, V.; Karakos, A.S.; Papadopoulos, B.K. Real time DDoS detection using fuzzy estimators. *Comput. Secur.* **2012**, *31*, 782–790.
12. Shiaeles, S.N.; Papadaki, M. FHSD: An Improved IP Spoof Detection Method for Web DDoS Attacks. *Comput. J.* **2015**, *58*, 892–903.
13. Siracusano, M.; Shiaeles, S.; Ghita, B. Detection of LDDoS attacks based on TCP connection parameters. In Proceedings of the Global Information Infrastructure and Networking Symposium (GIIS), Thessaloniki, Greece, 23–25 October 2018; pp. 1–6.
14. Yan, Q.; Huang, W.; Luo, X.; Gong, Q.; Yu, F.R. A Multi-Level DDoS Mitigation Framework for the Industrial Internet of Things. *IEEE Commun. Mag.* **2018**, *56*, 30–36.
15. Saridou, B.; Shiaeles, S.; Papadopoulos, B. DDoS attack mitigation through Root-DNS Server: A case study. In Proceedings of the IEEE World Congress on Services (SERVICES*)*, Milan, Italy, 8–13 July 2019; Volume 2642, pp. 60–65.
16. Prathyusha, D.J.; Govinda, K. Analysis of Network Flow for Mitigation of DDoS Attacks in a Cloud Environment. In *Embedded Systems and Artificial Intelligence*; Springer: Singapore, 2020; pp. 835–841.
17. Costa, W.L.; Silveira, M.M.; de Araujo, T.; Gomes, R.L. Improving DDoS Detection in IoT Networks Through Analysis of Network Traffic Characteristics. In Proceedings of the IEEE Latin-American Conference on Communications (LATINCOM), Santo Domingo, Dominican Republic, 18–20 November 2020; pp. 1–6.
18. Manikumar, D.V.V.S.; Maheswari, B.U. Blockchain Based DDoS Mitigation Using Machine Learning Techniques. In Proceedings of the Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 15–17 July 2020; pp. 794–800.
19. Wang, B.; Zhang, X. Construction of Compound DDOS Network Security System Based on PKI and CA Authentication. In *Data Processing Techniques and Applications for Cyber-Physical Systems (DPTA 2019), Proceedings of the DPTA 2019, Shanghai, China, 15–16 November 2019*; Springer: Singapore, 2020; pp. 375–382.
20. Vijayakumaran, C.; Muthusenthil, B.; Manickavasagam, B. A reliable next generation cyber security architecture for industrial internet of things environment. *Int. J. Electr. Comput. Eng.* **2020**, *10*, 387–395, doi:10.11591/ijece.v10i1.pp387-395.
21. Dantas Silva, F.S.; Silva, E.; Neto, E.P.; Lemos, M.; Venancio Neto, A.J.; Esposito, F. A Taxonomy of DDoS Attack Mitigation Approaches Featured by SDN Technologies in IoT Scenarios. *Sensors* **2020**, *20*, 3078, doi:10.3390/s20113078.
22. Sajid, A.; Abbas, H.; Saleem, K. Cloud-assisted iot-based scada systems security: A review of the state of the art and future challenges. *IEEE Acc.* **2016**, *4*, 1375–1384, doi:10.1109/ACCESS.2016.2549047.
23. Boyes, H.; Hallaq, B.; Cunningham, J.; Watson, T. The industrial internet of things (IIoT): An analysis framework. *Comput. Ind.* **2018**, *101*, 1–12, doi:10.1016/j.compind.2018.04.015.
24. Younan, M.; Houssein, E.H.; Elhoseny, M.; Ali, A.A. Challenges and recommended technologies for the industrial internet of things: A comprehensive review. *Measurement* **2019**, *151*, 107198, doi:10.1016/j.measurement.2019.107198.
25. Sha, L.; Xiao, F.; Chen, W. IIoT-SIDefender: Detecting and defense against the sensitive information leakage in industry IoT. *World Wide Web* **2018**, *21*, 59–88, doi:10.1007/s11280-017-0459-826.
26. Bettayeb, M.; Waraga, O.A.; Talib, M.A.; Nasir, Q.; Einea, O. IoT Testbed Security: Smart Socket and Smart Thermostat. In Proceedings of the *IEEE Conference on Application, Information and Network Security (AINS),* Pulau Pinang, Malaysia, 19–21 November 2019; pp. 18–23, doi:10.1109/AINS47559.2019.8968694.
27. Özgür, L.; Akram, V.K.; Challenger, M.; Dağdeviren, O. An IoT based smart thermostat. In Proceedings of the 5th International Conference on Electrical and Electronic Engineering (ICEEE), Istanbul, Turkey, 3–5 May 2018; pp. 252–256, doi:10.1109/ICEEE2.2018.8391341.
28. Liou, J.C.; Jain, S.; Singh, S.R.; Taksinwarajan, D.; Seneviratne, S. Side-channel information leaks of Z-wave smart home IoT devices: Demo abstract. In Proceedings of the 18th Conference on Embedded Networked Sensor Systems (SenSys'20), Virtual Event, Japan, 16–19 November 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 637–638, doi:10.1145/3384419.3430436.

29. Kaderabek, J. Integration of Fibaro system to intruder and hold-up alarm systems. In Proceedings of the 16th International Scientific Conference Engineering for Rural Development, Jelgava, Latvia, 24–26 May 2017; pp. 1080–1085, doi:10.22616/ERDev2017.16.N228.
30. Xu, Y.; Liu, Y. DDoS Attack Detection Under SDN Context. In Proceedings of the IEEE INFOCOM 2016—The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9, doi:10.1109/INFOCOM.2016.7524500.
31. Manso, P.; Moura, J.; Serrão, C. SDN-Based Intrusion Detection System for Early Detection and Mitigation of DDoS Attacks. *Information* **2019**, *10*, 106, doi:10.3390/info10030106.
32. Yuan, X.; Li, C.; Li, X. DeepDefense: Identifying DDoS attack via deep learning. In Proceedings of the IEEE International Conference on Smart Computing (SMARTCOMP), Hong Kong, China, 29–31 May 2017; pp. 1–8, doi:10.1109/SMARTCOMP.2017.7946998.
33. Hoque, N.; Bhattacharyya, D.; Kalita, J. Botnet in DDoS Attacks: Trends and Challenges. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2242–2270, doi:10.1109/COMST.2015.2457491.
34. Bawany, N.; Shamsi, J.; Salah, K. DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions. *Arab. J. Sci. Eng.* **2017**, *42*, doi:10.1007/s13369-017-2414-5.
35. Lukaseder, T.; Stölzle, K.; Kleber, S.; Erb, B.; Kargl, F. An SDN-based Approach for Defending Against Reflective DDoS Attacks. In Proceedings of the IEEE 43rd Conference on Local Computer Networks (LCN), Chicago, IL, USA, 1–4 October 2018; pp. 299–302, doi:10.1109/LCN.2018.8638036.
36. Vlajic, N.; Zhou, D. IoT as a Land of Opportunity for DDoS Hackers. *Computer* **2018**, *51*, 26–34, doi:10.1109/MC.2018.3011046.
37. Sharafaldin, I.; Lashkari, A.H.; Hakak, S.; Ghorbani, A.A. Developing realistic Distributed Denial of Service (DDoS) attack dataset and taxonomy. In Proceedings of the International Carnahan Conference on Security Technology (ICCST), Chennai, India, 1–3 October 2019; pp. 1–8, doi:10.1109/CCST.2019.8888419.
38. Kolahi, S.S.; Treseangrat, K.; Sarrafpour, B. Analysis of UDP DDoS flood cyber attack and defense mechanisms on Web Server with Linux Ubuntu 13. In Proceedings of the International Conference on Communications, Signal Processing, and their Applications (ICCSPA'15), Sharjah, UAE, 17–19 February 2015, doi:10.1109/ICCSPA.2015.7081286.
39. Barki, L.; Shidling, A.; Meti, N.; Narayan, D.G.; Mulla, M.M. Detection of Distributed Denial of Service Attacks in Software Defined Networks. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, India, 21–24 September 2016, doi:10.1109/ICACCI.2016.7732445.
40. Kumar, P.; Tripathi, M.; Nehra, A.; Conti, M.; Lal, C. SAFETY: Early Detection and Mitigation of TCP SYN Flood Utilizing Entropy in SDN. *IEEE Trans. Netw. Serv. Manag.* **2018**, *15*, 1545–1559, doi:10.1109/TNSM.2018.2861741.
41. Mohammadi, R.; Javidan, R.; Conti, M. Slicots: An sdn-based lightweight countermeasure for tcp syn flooding attacks. *IEEE Trans. Netw. Serv. Manag.* **2017**, *14*, 487–497, doi:10.1109/TNSM.2017.2701549.
42. Gurina, A.; Eliseev, V. Anomaly-Based Method for Detecting Multiple Classes of Network Attacks. *Information* **2019**, *10*, 84, doi:10.3390/info10030084.
43. Galeano-Brajones, J.; Carmona-Murillo, J.; Valenzuela-Valdés, J.F.; Luna-Valero, F. Detection and Mitigation of DoS and DDoS Attacks in IoT-Based Stateful SDN: An Experimental Approach. *Sensors* **2020**, *20*, 816, doi:10.3390/s20030816.
44. Chandel, S.; Yang, G.; Chakravarty, S. AES–CP–IDABE: A Privacy Protection Framework against a DoS Attack in the Cloud Environment with the Access Control Mechanism. *Information* **2020**, *11*, 372, doi:10.3390/info11080372.
45. Polat, H.; Polat, O.; Cetin, A. Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models. *Sustainability* **2020**, *12*, 1035, doi:10.3390/su12031035.
46. Bhandari, A.; Sangal, A.L.; Kumar, K. Characterizing flash events and distributed denial-of-service attacks: An empirical investigation. *Secur. Commun. Netw.* **2016**, *9*, 2222–2239, *doi:*10.1002/sec.*1472.*
47. Mystkowski, A.; Kierdelewicz, A. Fractional-Order Water Level Control Based on PLC: Hardware-In-The-Loop Simulation and Experimental Validation. *Energies* **2018**, *11*, 2928, doi:10.3390/en11112928.
48. Xiao, Y.; Yin, J.; Hu, Y.; Wang, J.; Yin, H.; Qi, H. Monitoring and Control in Underground Coal Gasification: Current Research Status and Future Perspective. *Sustainability* **2019**, *11*, 217, doi:10.3390/su11010217.
49. Vaclavova, A.; Kebisek, M. Design of Virtual Model of Production Line Using Wonderware ArchestrA. In Proceedings of the IEEE 22nd International Conference on Intelligent Engineering Systems (INES), Las Palmas de Gran Canaria, Spain, 21–23 June 2018; pp. 000425–000430, doi:10.1109/INES.2018.8523998.
50. Vaclavova, A.; Kebisek, M. Integration of production line with the Wonderware platform. In *Software Engineering and Algorithms in Intelligent Systems*; Springer: Cham, Switzerland, 2018; pp. 208–215, doi:10.1007/978-3-319-91186-1_22.
51. Panarello, A.; Tapas, N.; Merlino, G.; Longo, F.; Puliafito, A. Blockchain and IoT Integration: A Systematic Survey. *Sensors* **2018**, *18*, 2575, doi:10.3390/s18082575.
52. Tempest. *Production Line Description: User Manual*; Tempest: Ashland, OH, USA, 2014.
53. Mohammed, W.M.; Ferrer, B.R.; Iarovyi, S.; Negri, S.; Fumagalli, N.; Lobov, A.; Lastra, J.L.M. Generic platform for manufacturing execution system functions in knowledge-driven manufacturing systems. *Int. J. Comput. Integr. Manuf.* **2018**, *31*, 262–274, doi:10.1080/0951192x.2017.1407874.
54. Yee, I.; Eren, H. Data Historian. In *Instrument Engineers' Handbook: Process Software and Digital Networks*; CRC Press (Taylor and Francis Group): Boca Raton, FL, USA, 2011; pp. 454–464.

55. Erickson, B.; Manushree, A.; Naryzhny, Y.; Kamath, V.; Lie, C.; Middleton, E. Replicating Time-Series Data Values for Retrieved Supervisory Control and Manufacturing Parameter Values in a Multi-Tiered Historian Server Environment. U.S. Patent 8,676,756, 18 March 2014.
56. Patel, R.J.; Mahesuria, G.; Panchal, P.; Panchal, R.; Sonara, D.; Tanna, V.; Pradhan, S. Implementation of time synchronized cryogenics control system network architecture for SST-1. *Fus. Eng. Des.* **2016**, *112*, 747–751, doi:10.1016/j.fusengdes.2016.05.033.
57. Shipunov, M.V.; Grachev, V.V.; Myshlyaev, L.P.; Ivushkin, K.A.; Fayrushin, S.A.; Makarov, G.V. Creation of a control automation system on the example of the coal processing plant. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2020; doi:10.1088/1757-899x/865/1/012013.
58. Horák, T.; Šimon, M.; Huraj, L.; Budjač, R. Vulnerability of Smart IoT-Based Automation and Control Devices to Cyber Attacks. In *Computer Science On-Line*; Springer: Cham, Switzerland, 2020; pp. 287–294, doi:10.1007/978-3-030-51974-2_27.
59. Liang, L.; Zheng, K.; Sheng, Q.; Huang, X. A Denial of Service Attack Method for an IoT System. In Proceedings of the 8th International Conference on Information Technology in Medicine and Education (ITME), Fuzhou, China, 23–25 December 2016; pp. 360–364, doi:10.1109/itme.2016.0087.