Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

The Smart Extension approach for securing industrial control systems

Riccardo Colelli * **Chiara Foglietta** * **Stefano Panzieri** *
Federica Pascucci *

* *University of Roma Tre, Rome, Italy (e-mail: {riccardo.colelli, chiara.foglietta, stefano.panzieri, federica.pascucci}@uniroma3.it).*

Abstract: Industrial Control Devices are one of the major targets for hackers due to their exposure to threats. The principle of "air gaps" (disconnecting the Industrial Control Network from the operational networks) is not anymore feasible in a connected world. In this paper, a host anomaly detection system for Critical Infrastructures networks is presented. The device, called Smart Extension, also implements a filtering strategy in order to secure a single host reacting to cyber threats. Therefore, it is positioned in the network between PLC (Programmable Logic Controller) and the SCADA (Supervisory Control and Data Acquisition) control centre, more precisely just in front of the PLC. Finally, experimental results are shown in order to explain the internal working procedures in a possible case study.

Keywords: Industrial Control System, security, Smart Extension, Intrusion Detection System, filtering

1. INTRODUCTION

Critical Infrastructures (CIs) are a vital set of physical systems for our well-being. Among them, we remember power grids, telecommunications, water pipelines and transport networks. Those geographically distributed physical processes are continuously monitored and controlled by means of Industrial Control Systems (ICSs). Recently, the ICSs witness a massive integration between information and communication technologies. The old generation of monolithic Supervisory Control And Data Acquisition (SCADA) systems made way for the new network-based ones. Since then, industrial networks have been undergone numerous technical transformations to segment and protect the operational and manufacturing processes, leading to a new industrial revolution today, as can be seen in (Rubio et al., 2017). As described in (Chen et al., 2018), the Industrial Internet of Things (IIoT) paradigm foresees the use of remote monitoring and control exploiting network communication unifying the Operational Technologies (OTs) with the novel Information Technologies (ITs), such as edge and fog computing. Although this technical evolution has been very remarkable for the evolution of control processes, new security challenges concerning industrial facilities are constantly arising: vulnerabilities typical of the cyber domain have also emerged in ICSs.

To cope with these new challenges, the H2020 ATENA project (Adamsky et al., 2018) developed a complex architecture to prevent and properly react to attack on the ICSs of the CIs. One of the component of this architecture is the Smart Extension (SE). The SE is a device devoted to protect single appliance, such as a Programmable Logic Controller (PLC), a Remote Terminal Unit (RTU), or an industrial PC. The proposed architecture represents the evolution of the Smart Behavioural Filter (Corbò et al., 2018), since it is implemented over an industrial PC and is able to cope with different industrial protocols. The main aim of the SE is to protect the connected device by analyzing the incoming traffic: it is able to detect anomalies, to send alerts to Security Operation Center or to an Intrusion Detection System (IDS), and to set some mitigation actions. The innovation of the Smart Extension is the possibility to detect attacks on the protocol, exploiting vulnerability on the protocol such as an ARP spoofing, and to detect advanced attacks, such as data modification where the packets are well formed but the payload of the messages can damage the physical process controlled by the PLC. The Smart Extension would help in very complex attacks, where the damage is due to the sequence of commands that reach the PLC, as described in (Corbò et al., 2018).

The paper is organized as follows. In Sec. 2, the state of the art is reviewed, by considering OT solutions from IT networks. In Sec. 3, the architecture of the SE is proposed and in Sec. 4 the implementation is shown. In Sec. 5 preliminary experimental results are proposed, while some conclusive remarks can be found in Sec. 6.

2. RELATED WORKS

The introduction of the Industrial Internet of Things (IIoT) changed the scenario for securing the industrial plant. According to the IIoT paradigm, above the field network, there are two levels: the edge computing and the fog computing (Chiang and Zhang, 2016). Within the edge computing, industrial PC and embedded system are considered for the control operations. The fog computing level brings together information from the edge with a distributed intelligence processing before store data in cloud or in data centre. These paradigm allows more horizontally connected plant, that are more prone to fail to malicious agents: in this approach, more vulnerabilities can be found since the access points for attack increase.

To improve the security of the IIoT devices, distributed tools need to be developed to protect the single device in the network. Motivated by these issues, several solutions have been proposed both by industry and academy. Threats could come from passive or active actors. The first ones data are introduced in the system by the attacker (e.g., viruses, worms, trojan). The passive attacks aim to learn information on the system from the data. The most common attacks are given below (Borkar et al., 2017):

Pharming: Pharming aims to steal sensitive information at the expense of the user (Aslam et al., 2010). Private information are captured by the malicious actor. Pharming attacks use Internet vulnerabilities (e.g., DNS-Domain Name Server-servers, DNS resolvers) to direct the target to a malicious website.

Denial Of Service (DoS): DoS attack aims to temporarily or permanently disrupt the service of a host connected to the network. A DoS attack could be implemented by a flooding strategy: sending to the victim a large amount of network traffic workload (Carl et al., 2006). In this way, the target is no longer able to communicate in the network.

Eavesdropping Attack: Man-In-The-Middle (MITM) is the most common example of eavesdropping attack. The attacker is positioned between the communication of two devices. Thus, the malicious actor could intercept all the traffic generated by the victims. A technique used by the attacker for the MITM is the ARP (Address Resolution Protocol) spoofing, whereby, a weakness of the ARP protocol is exploited.

Ransomware: Ransomware works by obfuscating the contents of user files, often through the use of encryption algorithms. Victims have to pay the attacker to reverse this process (Scaife et al., 2016). This type of malware denies the use of the infected devices.

In order to protect devices, security tools are implemented in the network. Over time, many solutions for Information Technology (IT) are now consolidated. Otherwise, in Operational Technology (OT) area, those tool are not usually adapted. The different priority order of Confidentiality, Integrity and Availability in IT, it is reversed in OT. Thus, different approach must be used to secure industrial network.

A passive process for monitoring traffic on the network is the intrusion detection. Intrusion Detection Systems allow to identify anomalous behaviour with two different strategies: signature-based and anomaly-based. The signature-based detection uses well-know attack patterns in order to identify intrusions. Moreover, the anomaly-based IDS analyzes the deviation from the nominal established behaviour (Scarfone and Mell, 2007). The most relevant existing non-commercial IDS tools are:

- Snort uses a set of rules to identify malicious traffic in the analyzed network. Thus, Snort is signature-based IDS and it can send alerts to operators.
- Zeek (new name for the long-established *Bro* system) provides a comprehensive platform for more general network traffic analysis. While it supports such standard functionality as well, Zeek's scripting language indeed facilitates a much broader spectrum of very different approaches to find malicious activity, including semantic misuse detection, anomaly detection, and behavioral analysis.
- Suricata is an open source IDS that inspects traffic using an extensive set of rules.

IDSs are not the only defensive tools:

- Firewall (Stouffer et al., 2011) is usually implemented between the operational and corporate (or IT) networks. Firewalls provide several tools to block malicious communication or to enforce secure authentication of all users seeking to gain access to the ICS network.
- Honeypots (Spitzner, 2003) are information system used in order to induce the malicious actor to interact with them. As described in Simoes et al. (2013), it is possible apply the honeypot strategy also in the ICS networks.
- Wireshark is a protocol analyzer that includes the deep inspection of hundreds of protocols. Wireshark is used by operators in order to analyze traffic in online and offline mode.

The proposed Smart Extension (SE) is designed to protect a single element of the industrial network, i.e., a PLC, a RTU, an industrial PC. It intercepts the flow incoming to the protected device and analyzes it in order to passively and actively secure the node. To this end, it implements both an intrusion detection system based on the analysis of statistical network parameters and an anomaly based detection system exploiting deep packet inspection. Thus, the SE is tightly connected to the device that protect, since it is supposed to know the process behind the control device and the industrial protocols used for communicating to the network. The SE actively secure the protected device implementing also a firewall that filters incoming packets. Thus, the SE can drop some packets if they are considered dangerous.

3. SMART EXTENSION SYSTEM

The SE is designed for increasing the security of PLC, RTU, industrial devices such as an industrial PC and its operation is described in Fig. 1. Conceptually, it is composed by a probe directly connected to the protected device and responsible for monitoring and reporting events regarding underlying process and potential incidents. Therefore, the main task of a SE is the analysis of the traffic incoming to the protected device in order to prevent damage caused by intrusion. To this aim, its position is between the switch of the sub-network and the protected device. It is able to detect threats both at transport layer (e.g., Man-In-The-Middle attack) and at application layer (e.g., false data injection). Moreover, it is able to filter malicious traffic. In order to ensure the maximum compatibility and flexibility, the SE is designed

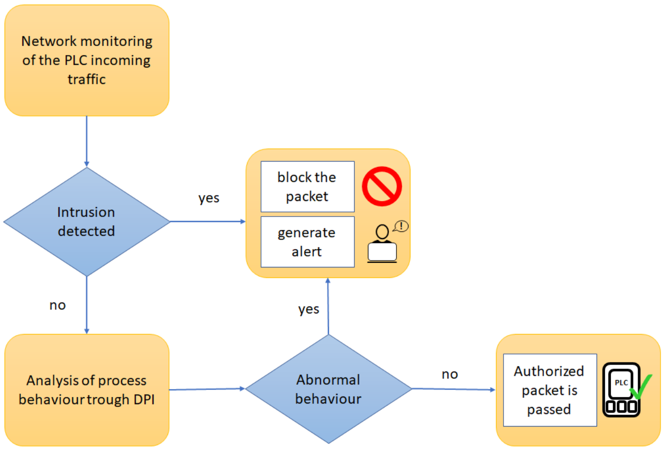

Fig. 1. Overview of the proposed system for filtering packets in the industrial environment

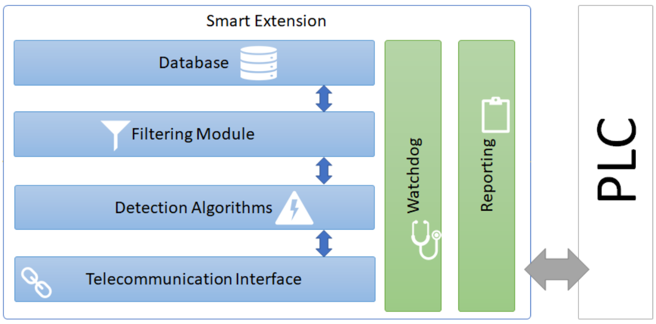

Fig. 2. Smart Extension architecture

to be a transparent device. The devices connected to the protected one do not notice that the packets go through the SE. Furthermore, the SE can be considered like a proxy server in a computer network: the connected devices are able to communicate each others without any additional configuration.

The architecture of the SE is shown in Fig. 2 and it is composed by the following functional layers:

- *Telecommunication Interface*: represents the logical layer that connects the SE to the protected device and the subnetwork. Thus, the Telecommunication Interface is able to provide data according to the requirements of the receiver. The Smart Extension can be part of the Intrusion Detection System, exploiting the Intrusion Detection Message Event Format (IDMEF - RFC 4765) (Debar et al., 2007) to provide information.
- *Detection Algorithms*: are a set of algorithms devoted to analyze the traffic flows generated by the protected device. They can be regarded as an intrusion and anomaly detection system developed for a single host. Since the SE is responsible for detection of intrusions targeted at the protected device, Detection Algorithms are a crucial layer inside the implementation of SE. Indeed, Detection Algorithms allow to intercept the attack and to respond accordingly. It is composed by a probe able to learn some statistical parameter of the network and able to analyze the behaviour of the system in order to detect anomalies.

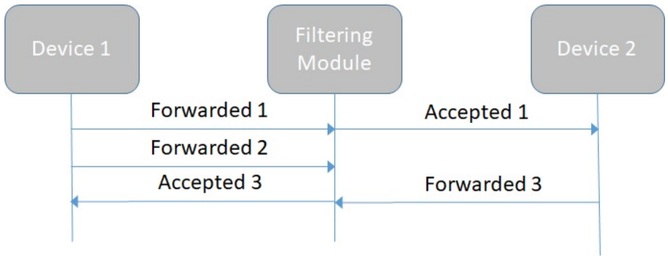

Fig. 3. Filtering Module allows to forward packets (e.g., packets 1 and 3) or to drop them if the packets are considered malicious (e.g., packet 2)

- *Filtering Module* is another important layer that allows the SE to block a packet before it reaches the protected device. It is designed to avoid the possibility of malicious commands towards a specific component of the network. According to the rules inside the detection algorithms layer, SE identifies malicious packet (e.g., improper values) and is able to do not forward it to the protected device, as depicted in Fig. 3. In this way, SE is close to the functions of a firewall, since it can deny or allow the transit of packets. However, SE improves this functionality using the deep packet inspection to filter packets containing malicious data detected at command level or through the statistical analysis.
- *Database* allows to track the rules of the Detection Algorithms and the reports produced by the SE. It is organized in different table, collecting rules and reports. This layer is tightly related with the process controlled by the protected device, since it collects the nominal behaviour of the system. This layer can be also implemented outside the SE, however, pushing out sensitive information would lower security levels of the whole SE and would increase the response time.
- *Watchdog* is implemented to avoid failure by the SE. In particular, this timer is always active during the SE operation and the restart function is triggered when something goes wrong in the control cycle of the traffic in SE. This failure could happen, for example, during a denial of service attack, as described in (Stajano and Anderson, 2000). During this malfunction, the filtering functions are bypassed according to a Fail-To-Wire policy as shown in Fig. 4 to avoid single point of failure.
- *Reporting* is the layer devoted to collect the information about threats. Reporting becomes fundamental for both a passive response by the operator and for further investigation about the attack. Reports are forwarded to the Cyber Detection System or to the specific cyber operator. SE provides also output to elaborate a response to the attack through anomaly detection alerts.

The device described above is able to capture a packet directed to the protected device and then analyze the content of this packet thanks to the deep packet inspection by specifying the industrial protocol. At this point, SE could decide to forward the packet or block it. Whenever the packet is considered suspicious but not dangerous for the monitored device, a report is created within information about the anomaly. Information, rules and

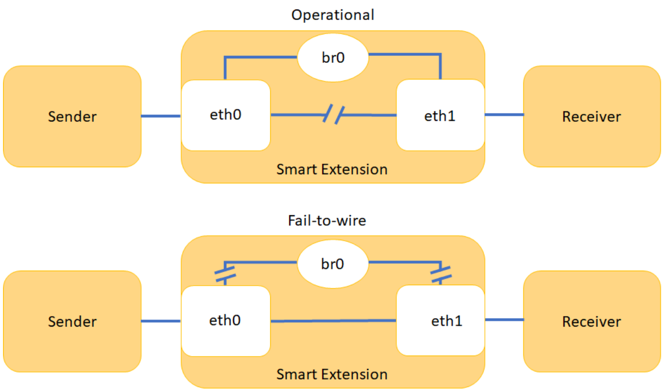

Fig. 4. The upper picture represents the operational functioning SE, the two network interfaces may communicate due to the application; the lower picture is the scheme of the SE with the Fail-to-Wire emergency system: the two Ethernet ports are connected via software to allow the straight-through connection in event of hardware or software failures.

behaviour reported within SE are always available through the Database. This feature allows a proper maintenance to the section of the network.

The requirements that the SE meet are the following. The SE was born for improving the security at low levels as the last chance to mitigate the effect of cyber attacks. Therefore, the SE must ensure a high-level of hardware/software security and must guarantee fail-safe procedures in event of hardware or software failures. The SE is an automatic reaction device that alerts in real-time also the operator to allow him to maintain the knowledge of what is happening in the field. The SE should not alter or interfere with the normal operations of the CI systems.

4. SMART EXTENSION IMPLEMENTATION

The SE has been implemented using C6015 produced by Beckhoff, an ultra-compact industrial PC. It is a multi-core industrial PC, that integrates Intel Core i processors and provides a new level of computing power to suit even the most demanding requirements.

For implementing the SE, a Unix operating system is used, even though the physical device produced by Beckhoff is provided with a Windows Operating System. The decision of changing the operating systems is derived from the flexibility of the Unix operating system. In Linux environments packet filtering and redirecting are available by programming firewall rules. In fact, the Linux kernel firewall provides the use of Iptables. More specifically, Ubuntu 18.04 is used in this proof of concept through the use of a bootable and permanent data storage device. The USB memory allows more flexibility to the proof of concept due to moving it on another device, exporting the solution. Moreover, Unix operating system allows the bridge mode between the two network interfaces. This method is fundamental for forwarding packet from a network interface to another when the filtering module is activated. Windows operating system allows also to bridge two network interfaces, but in this case the packets manipulation is not completely available. Moreover, the control over the traffic forwarding can be achieved through a tool for packets manipulation (e.g. in Python code) having low performance not suitable in real time environments.

In Unix, a new interface *br0* is created. The controlled forwarding is realised at kernel level, therefore the filtering component has high performance, and slow latency.

The SE architecture is designed with parallel processes in order to guarantee a computational effort divided into the different cores of the industrial PC. All parallel processes represent a layer in Fig. 2 and are controlled by the watchdog that is implemented in Python: it realizes the Fail-To-Wire policy.

The Telecommunication Interface is implemented by using Scapy, an interactive packet manipulation program written in Python. It is able to decode and forge packets of a large number of protocols in order to send or capture them in the network Packet crafting for Python2 and Python3. Scapy natively runs on Linux with libcap, libnet and their respective wrapper written in Python, moreover this tool is able to run with Python 2 and Python 3. There are two main functionalities in Scapy: sending packet and receiving answers. It is possible to define and to send a set of packets, to receive the answer and to finally match the couples request/answers. In addition to perform tasks performed by many network tools, such as Nmap, Scapy can also operate many specific tasks that other tools can't handle, like combining techniques as VLAN hopping with ARP cache poisoning or sending invalid framework.

The Detection Algorithms are represented by an executable Python script. Currently, it is able to analyze a packet by inspecting both the information in the header and the payload, implementing an intrusion and anomaly detection system. Currently, the algorithm works on the thresholds and the process variable or a combination of that are bounded to a priori limits.

The Filtering Module is executed by a Python script that use Nfqueue (netfilter-queue), netfilter/iptables project and different chains in Linux OS (Katic and Pale, 2007).

Information about detection rules, intrusion detected and behaviour of the process to be controlled are collected into database, in particular MySQL relational database management system.

The Reporting is a Python script that exploits Scapy to format the alerts according to the protocol adopted by the Cyber Detection System.

5. PRELIMINARY RESULTS

To prove the effectiveness of the proposed approach, the SE has been used to protect the CX2030, an embedded PC produced by Beckhoff. Two types of attack have been considered: the detection methods used to identify the attacks consider both the analysis of the header and the payload of the collected traffic.

The first attack is a Man-In-The-Middle (MITM): the malicious agent is positioned in the middle of the communication between two devices. The purpose of this attack is to observe data in the network without any permission. To this end, the attacker performs an ARP poisoning

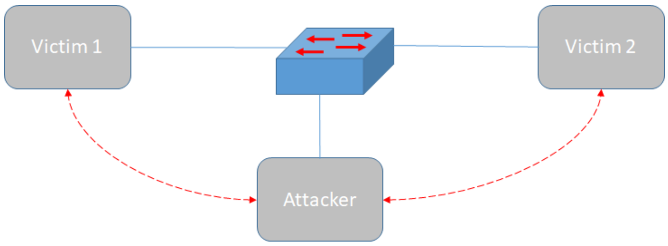

Fig. 5. Man-In-The-Middle attack

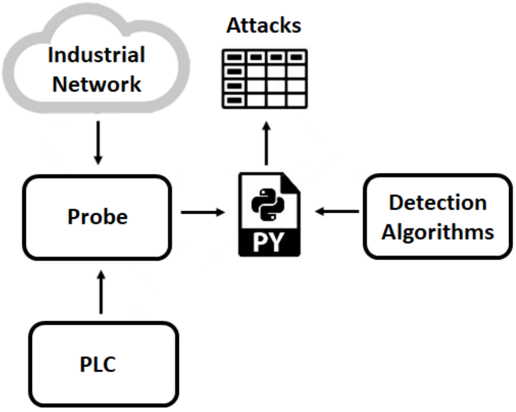

Fig. 6. MITM detection

(table poisoning ARP). The lack of authentication is the structural vulnerability behind this attack thus a protocol without encryption or authentication instruments is more vulnerable. The ARP protocol manages the relationship between IP addresses and MAC addresses. The attacker sends properly constructed ARP packets in order to receive the packet in transit and resulting transparent between the two devices communication. This means that the received packet is forwarded to the original recipient as illustrated in Fig. 5.

In order to identify this type of attack, in the SE Database, for each packet retrieved from the network, the IP address and the connected MAC address are registered in the table Attacks of the Database (see Fig. 6). The script Python first captures traffic thanks to the probe and then, with a multiple thread strategy, it is able to write specification about IP address and MAC address. Moreover, this table is used to identify other attacks in the low levels of OSI model (e.g. Port Scan). If there is a mismatch between the two addresses, SE reacts by filtering the packets and by blocking them using iptables and netfilter. In this way, the attacker could abort the eavesdropping, since the suspicious packets are not forwarded.

The second attack is the Data Modification, that can be regarded as one of the most dangerous attack that a malicious actor could implement for the purpose of controlling a process. The attack changes a determined value of a packet depending on the control that the attacker wants to strike. This type of attack can be customized according to the communication protocol. For this reason, Data modification attack is often preceded by a session of eavesdropping. Modifying a data could be used in order to falsify values from a controller gain injection or for sensor data tampering. A data can be changed in different methods depending on attacker position:

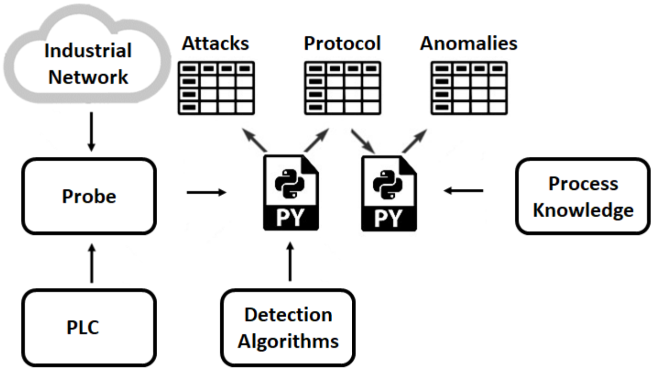

Fig. 7. Tables for Data Modification detection

- Man-In-The-Middle position: striker is positioned between two target (e.g. ARP Spoofing) in order to modify on fly a packet before send it to the original recipient, as shown in Fig. 5
- Insider position: striker is able to modify values in a control process due to authorized account or with a privilege escalation.

In our experiment the attacker gains the control of the network by a MITM attack, however, in this case the filtering on network layer is deactivated. The attacker can modify the data in the payload of the packet. In our set-up this payload contain command and data formatted according to the ADS (Automation Device Specification) protocol by Beckhoff.

Since Scapy does not support ADS, a proper parser has been also implemented using Python. This code within the probe is able to analyze each command of a specific variable to be monitored through the deep packet inspection strategy and then write the values into another table named Protocol. Over the table Protocol another process elaborates the information and the behaviour of the specific variable and then detect a mismatch with the nominal behaviour. The mismatch funded is reported into the third table Anomalies, that is used to generate the alert in the passive mode or block the packets in the active one, as shown in Fig. 7.

6. CONCLUSION

The main requirement of an industrial plant is to guarantee its proper functioning even in case of intrusion by third part. In this paper, devices used in field for reaction strategies are analyzed, first describing devices already existing and exploited for mitigate the critical issues of ICT over critical Infrastructures. This description mentions the analysis of vulnerability of the most used protocol in industrial systems and the resulting weakness of protocols could be exploited by malicious actors. In this respect, security protocols are recommended in order to make intrusions difficult. In this paper is shown how the architecture and the topology of the network could be efficient for avoiding intrusion or avoiding attacks, in particular the task of filtering denies the arrival of suspicious packets to critical component in the network (e.g. PLC).

In order to ensure as much details as possible functioning by the device for mitigation, deep packet inspection (DPI) is analyzed. Because of DPI, it is possible to trace back to each command reported into the packet and reporting or blocking anomaly behaviour of critical variable.

SE works in different function layers implemented over an industrial PC Beckhoff C6015 device in a Unix Environment. Moreover, are illustrated different scenarios used on Roma Tre test-bed, in particular Man-In-The-Middle for eavesdropping first and then data modification.

Combining knowledge and implementation of this task, in particular with the SE solution, it is possible to increase the security level of network section, attempting either to avoid malicious command to the more vulnerable device or alert in good time supervisory levels about the situation in order to contribute to a safer living environment for working people and constantly services for end-users.

In the future work we would like to create a discrete event systems approach to represent process knowledge. These implementation could be useful to better understand dangerous states of the system.

REFERENCES

Adamsky, F., Aubigny, M., Battisti, F., Carli, M., Cimorelli, F., Cruz, T., Di Giorgio, A., Foglietta, C., Galli, A., Giuseppi, A., et al. (2018). Integrated protection of industrial control systems from cyber-attacks: the atena approach. *International Journal of Critical Infrastructure Protection*, 21, 72–82.

Aslam, B., Wu, L., and Zou, C.C. (2010). Pwdip-hash: A lightweight solution to phishing and pharming attacks. In *2010 Ninth IEEE International Symposium on Network Computing and Applications*, 198–203. doi: 10.1109/NCA.2010.35.

Beckhoff (2019a). [acessed October.24,2019]. `https://www.beckhoff.com/`.

Beckhoff (2019b). Ads protocol. [acessed October.24,2019]. `https://infosys.beckhoff.com`.

Borkar, A., Donode, A., and Kumari, A. (2017). A survey on intrusion detection system (ids) and internal intrusion detection and protection system (iidps). In *2017 International Conference on Inventive Computing and Informatics (ICICI)*, 949–953. doi: 10.1109/ICICI.2017.8365277.

Carl, G., Kesidis, G., Brooks, R.R., and Rai, S. (2006). Denial-of-service attack-detection techniques. *IEEE Internet Computing*, 10(1), 82–89. doi: 10.1109/MIC.2006.5.

Chen, B., Wan, J., Shu, L., Li, P., Mukherjee, M., and Yin, B. (2018). Smart factory of Industry 4.0: Key technologies, application case, and challenges. *IEEE Access*, 6, 6505–6519.

Chiang, M. and Zhang, T. (2016). Fog and iot: An overview of research opportunities. *IEEE Internet of Things Journal*, 3(6), 854–864. doi: 10.1109/JIOT.2016.2584538.

Corbò, G., Foglietta, C., Palazzo, C., and Panzieri, S. (2018). Smart behavioural filter for industrial internet of things: A security extension for plc. *Mobile Networks and Applications*, 23(4), 809–816.

Debar, H., Curry, D.A., and Feinstein, B.S. (2007). The intrusion detection message exchange format (idmef).

Katic, T. and Pale, P. (2007). Optimization of firewall rules. In *2007 29th International Conference on Information Technology Interfaces*, 685–690. doi: 10.1109/ITI.2007.4283854.

netfilter/iptables project (2019). [acessed October.24,2019]. `http://www.netfilter.org`.

Packet crafting for Python2 and Python3 (2019). Scapy. [acessed October.24,2019]. `www.scapy.net`.

Rubio, J.E., Alcaraz, C., Roman, R., and Lopez, J. (2017). Analysis of intrusion detection systems in industrial ecosystems. In *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications - Volume 6: SECRYPT, (ICETE 2017)*, 116–128. INSTICC, SciTePress. doi:10.5220/0006426301160128.

Scaife, N., Carter, H., Traynor, P., and Butler, K.R.B. (2016). Cryptolock (and drop it): Stopping ransomware attacks on user data. In *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, 303–312. doi:10.1109/ICDCS.2016.46.

Scarfone, K. and Mell, P. (2007). Nist special publication 800-94, guide to intrusion detection and prevention systems (idps).

Simoes, P., Cruz, T., Proença, J., and Monteiro, E. (2013). On the use of honeypots for detecting cyber attacks on industrial control networks.

Snort (2019). [acessed October.24,2019]. `https://www.snort.org/`.

Spitzner, L. (2003). Honeypots: catching the insider threat. In *19th Annual Computer Security Applications Conference, 2003. Proceedings.*, 170–179. doi: 10.1109/CSAC.2003.1254322.

Stajano, F. and Anderson, R. (2000). The grenade timer: Fortifying the watchdog timer against malicious mobile code. In *in Proceedings of 7th International Workshop on Mobile Multimedia Communications (MoMuC 2000), Waseda*.

Stouffer, K., Falco, J., and Scarfone, K. (2011). Guide to industrial control systems (ics) security. *NIST special publication*, 800(82), 16–16.

Suricata (2019). [acessed October.24,2019]. `https://suricata-ids.org/`.

Wireshark (2019). [acessed October.24,2019]. `https://www.wireshark.org/`.

Zeek (2019). [acessed October.24,2019]. `https://www.zeek.org`.