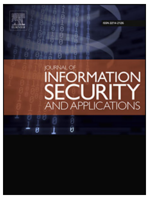
Journal of Information Security and Applications 54 (2020) 102535

Contents lists available at ScienceDirect

Journal of Information Security and Applications

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



Flow based monitoring of ICS communication in the smart grid

Petr Matoušek*, Ondřej Ryšavý, Matěj Grégr, Vojtěch Havlena

Faculty of Information Technology, Brno University of Technology, Božetěchova 1/2, Brno, Czech Republic

ARTICLE INFO

Keywords:
IEC 104
Smart grid
Security monitoring
SCADA
Probabilistic automata

ABSTRACT

A smart grid network is a part of critical infrastructure, and its interruption or blackout may cause fatal consequences on energy production, distribution, and eventually lives of people. Smart grid networks can be a target of cyber attacks coming from the outside or the inside of the network. Traditional smart grid protection includes firewalls and IDS/IPS devices that are usually deployed on edges of the network where they inspect incoming and outgoing traffic. This approach is adequate to cope with external threats. In case of internal threats caused by, for instance, the malware infecting the control station, it is not easy to detect malicious activity commonly masked as legitimate communication at the network edge. For the successful identification of cyber security attacks, two essential elements are necessary.

The first is the visibility of the Industrial Control System (ICS) communication, which enables a smart grid operator to see real-time transmissions in the network. The second important part is an anomaly detection system that analyzes monitoring data and identifies possible cyber security attacks. This paper presents a novel system for monitoring ICS/SCADA protocols based on IP flows extended with application layer data obtained from ICS packet headers. The monitoring system provides an in-depth insight into ICS communication. By applying statistical-based methods or creating communication profiles using probabilistic automata, common security attacks, as well as unknown threats, can be identified. The proposed approach is demonstrated on IEC 60870-5-104 communication.

1. Introduction

Industrial Control System (ICS) plays an essential role in monitoring and controlling industrial devices, processes and events. The main function of ICS is to gather real-time data from industrial devices, realize device automation, and supervise the system [1]. The ICS concept covers a variety of control systems including distributed control systems (DCS), supervisory control and data acquisition (SCADA) system and others.

ICS communication was originally designed for serial data links that were physically separated from external networks. Recently, ICS communication has been adopted to operate over Ethernet with the Internet Protocol (IP) and UDP/TCP transport. This solution opened possibility to interconnect ICS networks over wide-area networks (WANs) in order to provide remote control and online updates.

Fig. 1 shows various industrial control protocols like MMS, Goose, Modbus, SV, or IEC 104 that operate within a smart grid substation and interconnect the control system with Intelligent Electronic Devices (IEDs) and power equipment such as circuit breakers, bay controllers, or relays.

Interconnection of ICS systems with IP networks revealed serious security flaws in industrial protocols design. ICS system is vulnerable to cyber security threats as revealed by attacks against Ukraine's power grid in December 2015 [2] and 2016 [3,4]. The ICS system of Ukrainian power grid was infected by a malware installed on operator's control station without noticing it. The malware called Industroyer by ESET or CrashOverride by Dragos, Inc. Dragos [5] masqueraded as a legitimate process and communicated with Remote Terminal Units (RTUs). At first, the malware scanned the internal network for RTUs and learnt their addresses and functions. Following that it tested ability to switch on and off RTU devices. Since the malware was communicating inside the network, it was unnoticed by an IDS system and firewall located at the edge of the network.

This case unveiled missing visibility of ICS communication within the power grid as mentioned in Assante and Lee [6]. Without advanced monitoring of ICS communication, it is not easy to detect common cyber attacks like scanning, command injection, data spoofing, etc., when launched from the inside of the network.

Insufficiency of ICS network monitoring was highlighted in the Report of European Union Agency for Network and Information Se-

* Corresponding author.

E-mail addresses: matousp@fit.vutbr.cz (P. Matoušek), rysavy@fit.vutbr.cz (O. Ryšavý), mgregr@fit.vutbr.cz (M. Grégr), ihavlena@fit.vutbr.cz (V. Havlena).

https://doi.org/10.1016/j.jisa.2020.102535
2214-2126/© 2020 Elsevier Ltd. All rights reserved.

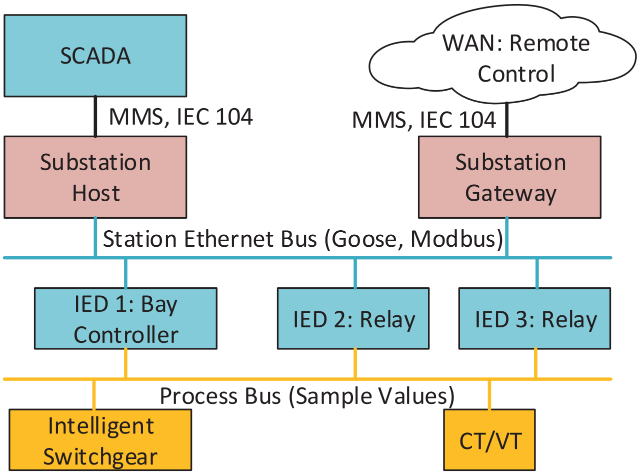


Fig. 1. SCADA communication in the substation.

curity (ENISA) [7] which states that *without active network monitoring, it is very difficult to detect suspicious activity, identify potential threats, and quickly react to cyber attacks.* NIST Guide to ICS Security [8] recommends network segregation, application of firewall rules, redundancy, etc. to secure ICS networks. However, these techniques provides only limited security against internal cyber security attacks. In our approach, we propose a monitoring solution that provides extended L7 visibility of ICS communication that can be used for detection of both internal and external cyberattacks.

The approach was inspired by IP networks where security monitoring is well established. IP monitoring techniques include SNMP monitoring [9] IP flow monitoring [10], and system logging [11]. These approaches can be applied to a certain extent also on ICS systems. Since IP monitoring relies on IP layer, it cannot be easily transferred to ICS protocols that run directly over link layer, e.g., Modbus or GOOSE. Due to the restricted hardware and firmware of RTUs and IEDs, it is not easy to implement SNMP agents or Syslog clients on these devices.

Feasible option for ICS networks is a passive flow monitoring based on IPFIX protocol [12]. Flow monitoring system includes IPFIX probes that observe ICS traffic and create flow records extended with ICS application protocol data. The big advantage of this solution is that probes can be deployed anywhere in the smart grid network. They provide L7 visibility of ICS communication also inside the network. Unlike IDS systems, flow monitoring probe processes packet headers only which is fast and less demanding comparing to the deep packet inspection (DPI) implemented in IDS systems. The probe aggregates extracted header values into flow records that are later sent to the collector. The IPFIX collector stores monitoring data from all probes. Collected data provides a global view on the network communication in multiple points of the network. The data can be further analyzed anomaly detection methods and visualized on the dashboard.

IPFIX standard supports flexible definition of monitoring information using templates. For any ICS protocol we can defined a set of headers that can be extracted from packets and transmitted to the collector using IPFIX records. In case of IEC 104 communication, we can observe the cause of transmission (COT), ASDU type or information object address, from GOOSE packets we can extract control block reference, ID, or status number, etc.

The proposed approach is demonstrated on monitoring of IEC 60870-5-104 (aka IEC 104) communication [13,14]. The paper describes how IEC 104 packets are processed by the probe and IEC 104 flow records created. We also show how to analyze flow data using statistical-based approach which is able to identify majority of common cyber security attacks described by NISTIR 8219 Report [15] as showed in the paper.

Unlike common Internet traffic, ICS communication shows two typical features: stability and periodicity [16–18]. Based on this features, ICS flow data can be used to create communication profiles. The paper shows how typical communication profile can be described using probabilistic automata and used for anomaly detection.

1.1. Structure of the text

The paper is structured as follows. Section 2 gives an overview of published work related to the monitoring and security of smart grid communication. Section 3 describes a flow monitoring system for ICS communication, gives details about packet extraction and ICS flow record creation. It defines four levels of ICS visibility depending on details extracted from ICS headers. The approach is demonstrated on IEC 104 protocol. Section 4 reviews common security threats against ICS systems as described in NISTIR 8219 recommendation [15]. The section shows how selected threats can be identified in ICS flows. Section 5 presents how ICS communication sequences can be represented by probabilistic automata and used for anomaly detection based on learnt communication profiles. The last section concludes this article.

1.2. Contribution

The main contribution of this paper is an extension of IPFIX monitoring system for ICS/SCADA protocols. ICS flow monitoring provides enhanced visibility of smart grid networks and creates data that can be used for common security incident detection as described in Section 4 and Section 5.

This article is an extended version of original authors' work presented at the 6th International Symposium for ICS & SCADA Cyber Security. This extended version adds implementation details about creating ICS flows and shows how it can be applied to any ICS/SCADA protocol. Since the whole monitoring system is based on standardized IPFIX architecture, it is straightforward to incorporate flow monitoring to any Security Information and Event Management (SIEM) system with IPFIX support. Section 4 extends description of cyber security incidents in smart grid networks by presenting typical security scenarios defined in NISTIR 8219 recommendation [15]. Section 5 brings a new unpublished work that demonstrates how ICS communication profiles can be represented by probabilistic automata.

2. Related work

Protection of smart grid networks against cyber attacks has been researched by many authors [19–22]. NIST Guide to ICS Security [8] presents a large overview of past attacks on ICS systems with recommendation how to secure ICS architecture using network segregation, firewall rules, NAT translation and other techniques.

Security of ICS/SCADA networks is often implemented by proprietary IDS systems with deep-packet inspection (DPI) that analyze selected ICS protocols. Generally, an IDS system parses ICS packets and extracts selected data from ICS protocol header and payload. The data is subject to further signature-based or behavior-based analysis. If a suspicious communication is detected, an alert is risen and the traffic is filtered out. Real-time scanning and analysis of ICS packets demand high processing power and fast memory. Each ICS protocol requires an ICS pre-processor (parser) to be a part of an IDS system [23]. IDS systems are usually located at the edge of ICS/SCADA networks. This limits protection to external threats only. The proposed IPFIX flow-based monitoring system can observe communication both at the edge of the substation network as well as in the inside of the network.

Distributed monitoring system for protecting SCADA communication in power grid was proposed by Jarmakiewicz et al. [24]. The authors deployed SCADA probes over the power grid network which observed IEC 104 and IEC 61850 communication. The probes had Snort and Bro software with SCADA protocol analysis installed. When a security incident was detected, the probe sent an alarm to the SIEM system. In contrary to this system, the proposed ICS flow monitoring system gathers data from IPFIX probes and stores them in the IPFIX collector where it can be analyzed using various method. Detection does not depend on Bro or Snort rules but it can be implemented using various anomaly detection methods. IDS systems for power grids based on Snort rules are also presented by Yang et al. [25], Hong et al. [26]. Such systems can also implement multi-attribute analysis [27].

Barbosa et al. [28] propose flow-based monitoring for whitelisting. Unlike our solution, their approach is based strictly on IP flows. They observe packets and extract four properties to build a flow: client address, server address, server-side port and transport protocol. During the learning phase, the system creates an initial white-list of legitimate flows. In detection phase, when a new flow is detected that was previously not white-listed, an alarm is raised. Our approach observes also L7 packet headers in addition to IP addresses and ports which enhances visibility of ICS transmissions by displaying important details, e.g., ASDU types, GOOSE IDs, and also provides ICS-specific data for anomaly detection.

Anomaly detection of smart grid communication mostly implements rule-based methods as mentioned in the previous text. There are also multiple works that observe ICS communication behavior like delays, periodicity, stability and other patterns. Such behavior-based IDS systems create communication profiles during the learning phase by observing normal ICS traffic. In monitoring or detection phase, behavior-based IDS systems compares passing traffic with learnt profiles. If a new traffic does not match any learnt profile, it is marked as anomalous and alarm is raised. The proposed ICS flow monitoring may provide data for behavior-based or statistical-based system, see Sections 4.1.2 and 5.

Anomaly detection systems may observe various communication features. Authors in Lin and Nadjm-Tehrani [29] observe inter-arrival times of IEC 104 spontaneous events. Based on time patterns, communication is classified into five previously defined groups. Kleinmann and Wool [30] model Modbus traffic streams using Deterministic Finite Automata (DFA) with following characteristics: a symbol is defined as a concatenation of message type, function code and address range, and a state is defined for each message in the periodic traffic pattern. During the learning phase, the pattern length is revealed and DFA is built for each HMI-PCL channel. In the detection phase, traffic is monitored for each channel using its DFA and proper events triggered.

Caselli [31] models semantics of Modbus communication using discrete-time Markov chains (DTMC). He demonstrates his approach on a Modbus sequence which is defined as a time-ordered list of events where the event is a 3-tuple of transaction ID, operation code and data. The sequence is represented by a state of DTMC where transitions model time relation. This approach is similar to probabilistic automata that we use for modelling IEC 104 communication, see Section 5.

Combination of rule-based anomaly detection and IP flow analysis is described in Kwon et al. [21] where the authors use IP flow statistics like packet rate and packet size to classify traffic into the four behavior characteristics in power equipment. Besides, they observe GOOSE communication and detects selected security incidents using rule-based IDS.

Martinelli et al. [32] describe SCADA communication using timed automata. Attack vectors are represented by temporal logic formulae. Using employ model checking they verify if a property is valid on the model. This is an interesting approach which is, however, not easy to implement for real-time communication.

Statistical-based approach on IEC 61850 communication is presented by [33]. The authors extract features from GOOSE and MMS messages and create statistical model created using mean, variance and standard deviation classified using support vector machine algorithm. Their approach can be also adopted to our flow monitoring system, especially on the collector where anomaly detection can be implemented using various methods.

The article presents a novel flow-based monitoring system that gathers ICS flow data from monitoring probes using IPFIX protocol. ICS flow data are later analyzed using statistical based approaches and probabilistic automata representing L7 communication between ICS devices.

3. Flow based monitoring of smart grid communication

ICS communication in the smart grid includes industrial protocols like IEC 61850 GOOSE [34], Modbus, IEC 60870-5-104 [13], DNP3, IEC 61850 MMS [35], DLMS [36] and others.The protocols transmit control and status data from industrial processes running on RTUs or IEDs. Protocols like GOOSE and Modbus implement *publish-subscribe* mechanism where an application (publisher) writes the values into a local buffer that is periodically transmitted to subscribing agents using L2 multicast. ICS protocols like IEC 104, DNP3, MMS or DLSM communicate using *client-server model*. In this model, controlled station (RTU slave) is monitored or commanded by a master station. Controlling station (PC with SCADA system, RTU master) performs control of outstations. ICS client-server communication can be delivered in the *monitoring direction* (from controlled station to the controlling station) or in the *control direction* (RTU master sends commands toward the RTU slave), see Fig. 2.

Smart grid security requires awareness of active communication in the network, e.g, what nodes are sending or receiving data, what ICS protocols are active in the network, what commands are issued, how many packets were transmitted between two devices within a given time window, etc. Traditional network monitoring systems provides visibility of network communication using SNMP statistics, syslog events or flow based records. This paper shows how flow based monitoring can be enhanced by ICS-specific meta data.

3.1. Architecture of ICS flow monitoring system

Flow based monitoring system is composed of monitoring probes that observe packets on the link, extract meta data from passing flows and creates so called flow records that are sent to the flow collector. Traditional IP flow is defined as *a sequence of IP packets passing the observation point during a certain time interval* [10]. Packets belonging to a given flow have a set of common properties. IP flow properties include source and destination

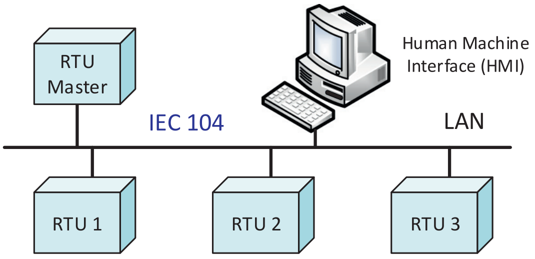


Fig. 2. IEC 104 topology.

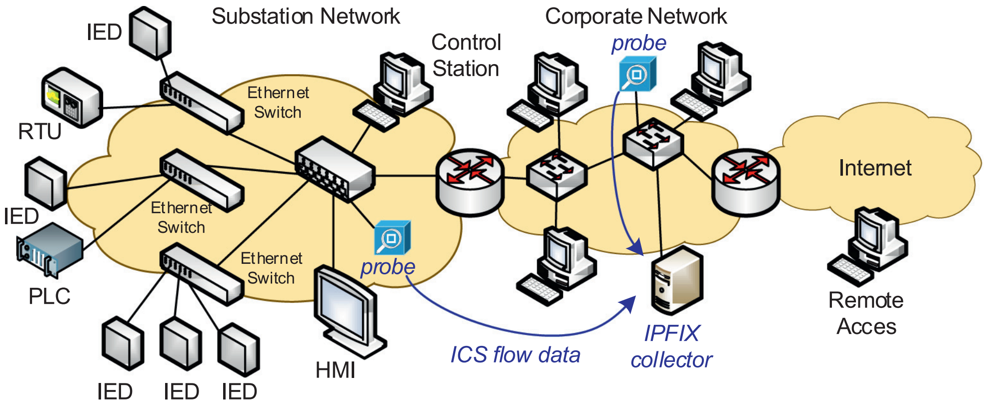


Fig. 3. ICS flow monitoring system.

IP addresses, source and destination port numbers and the protocol type. An example of the IP flow is a HTTP request from a client (identified by the src IP address and port) to a specific server (identify by the dst IP address and port). For each flow the monitoring probe collects meta data, e.g., timestamp of the occurrence of the first packet of the flow, the number of packets, the number of transmitted bytes, duration of the flow, etc. Meta data together with flow properties form a *flow record*. The probe creates flow records of all flows that were transmitted over the link where the probe is connected. The flow records are then deliver to the IPFIX collector, see Fig. 3.

3.2. ICS Flow

For ICS monitoring, we extend definition of IP flows by adding property values extracted from ICS protocol headers. Let P be an IP packet with a set of IP header fields, i.e. $P = \{p_1, p_2, \ldots, p_n\}$, T be a transport layer (L4) protocol data unit (PDU) with L4 protocol headers, i.e., $T = \{t_1, t_2, \ldots, t_m\}$, and A be an application layer (L7) PDU with L7 headers, i.e., $A = \{a_1, a_2, \ldots, a_o\}$. We define *flow property Fprop* as a subset of L3, L4 and L7 headers. The flow property *Fprop* will be mapped to specific L3, L4, and L7 protocols, e.g., IP protocol on Layer 3, TCP or UDP on Layer 4, and IEC 104 on L7.

$$Fprop(ICS) \subseteq P(IP) \cup T(UDP/TCP) \cup A(ICS) \tag{1}$$

For ICS protocols transmitted directly over the link layer, e.g., GOOSE or Modbus, the probe can skip L3 and L4 layer in the *Fprop* or it can create virtual IP and TCP/UDP layers where, for example, IPv6 link local addresses will be generated from MAC address as recommended in RFC 4291 [37].

In case of IEC 104 protocol monitoring, L3 layer properties are typically source IP, destination IP and protocol type, L4 layer properties include source and destination ports, and L7 properties of IEC 104 may include APDU frame type, ASDU type, cause of transmission (COT), number of information objects, origination address (ORG) and ASDU address (COA). Thus, flow properties of IEC 104 can be expressed as follows:

$$\begin{aligned} Fprop(IEC104) = \{&SrcIP, DstIP, IPprot, SrcPort, DstPort, APDUtype, \\ &ASDUtype, COT, Items, ORG, COA\} \end{aligned} \tag{2}$$

Definition (1) is flexible in selection of protocol headers that will be used for flow monitoring. This is important especially for L7 protocols where each protocol has different protocol structure and protocol headers. Thus, *Fprop* will be mapped to any ICS protocol similarly as shown in formula (2).

Let define *ICS flow* as a *sequence of ICS packets passing the observation point during a certain time and having the same flow property Fprop*. ISC probe parses ICS packets, extracts selected header values and creates ICS flow records composed of these values. For ICS monitoring, the ICS flow record is a basic set of data obtained during ICS traffic monitoring. The ICS flow record contains *Fprop* data that identifies the flow and statistical data *Fstat* that describes behavior of the flow. *Fstat* set is computed by the probe and includes start time of the flow, end time, number of packets of the flow, total size of packets in the flow, etc., i.e., $Fstat = \{t_{start}, t_{end}, packets, size, \ldots\}$. Following that definitions, ICS flow record *Frec* can be described as an union of ICS flow property values and statistical behavior of the flow. ICS flow record is then mapped to a specific ICS protocol, e.g., IEC 104, MMS, etc.

$$Frec(ICS) = Fprop(ICS) \cup Fstat \tag{3}$$

An example of IEC 104 flow record is in Fig. 4.

The Figure shows an IEC 104 flow composed of five packets. he IEC 104 flow record contains values from L3 layer (IP addresses), L4 layer (ports) and L7 layer (selected IEC 104 headers). All these values (properties) identify an IEC 104 flow. The flow record contains property values plus statistical values related to the flow. The IEC 104 flow record is transmitted by IPFIX protocol to the IPFIX collector where is stored. A set of flow records is then analyzed and visualized through network management system. Technical details about creating IEC 104 flows are mentioned at [38].

Due to flexible definition of IPFIX flow record format, $Frec(ICS)$ values can be easily mapped into IPFIX flow records using IPFIX templates as defined in RFC 7011 [12]. For each ICS protocol, a unique template with specific fields related to the protocol shall be defined. For example, GOOSE template may include APPID, control block reference, data set or status number extracted from GOOSE header.

ICS monitoring requires an ICS monitoring probe to implement a ICS protocol parser for each supported protocol and definition of an IPFIX template that maps $Frec(ICS)$ fields into IPFIX record format. Parsers for common ICS protocols as GOOSE, IEC-104, MMS and DLSM were implemented by the authors of this paper in frame of research project IRONSTONE (2016–2019) and are available at the project web site.[1]

3.3. Collecting ICS flow data

A big advantage of ISC flow-based monitoring is that ICS monitoring probes can be deployed anywhere in the smart grid network as shown in Fig. 3, thus providing monitoring data from overall the network. The ICS-enabled monitoring probe passively observes passing traffic. ICS flow records are transmitted via IPFIX protocol to the IPFIX collector for further analysis and visualization.

[1] See https://www.fit.vut.cz/research/project/1101/.en [Dec 2019].

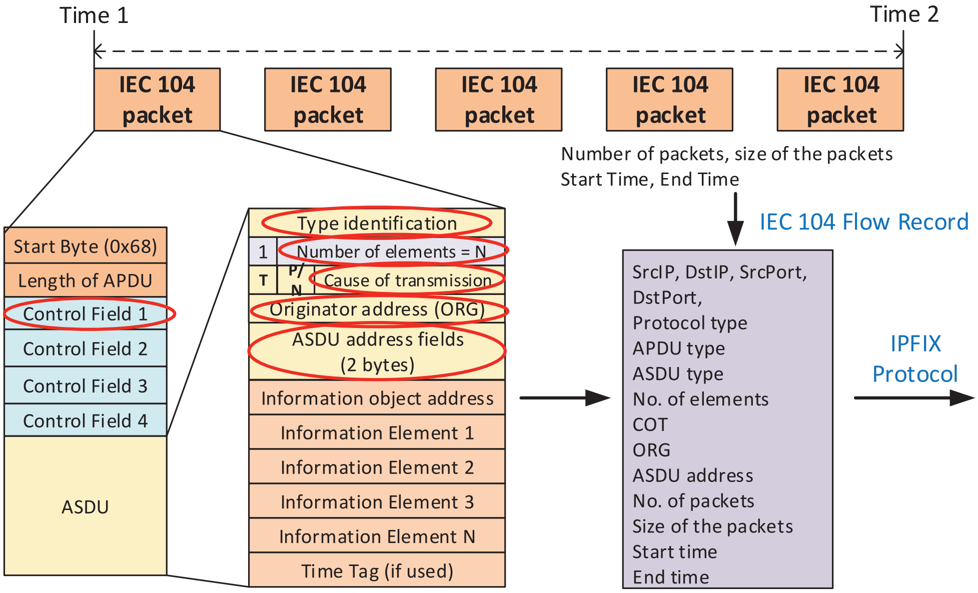
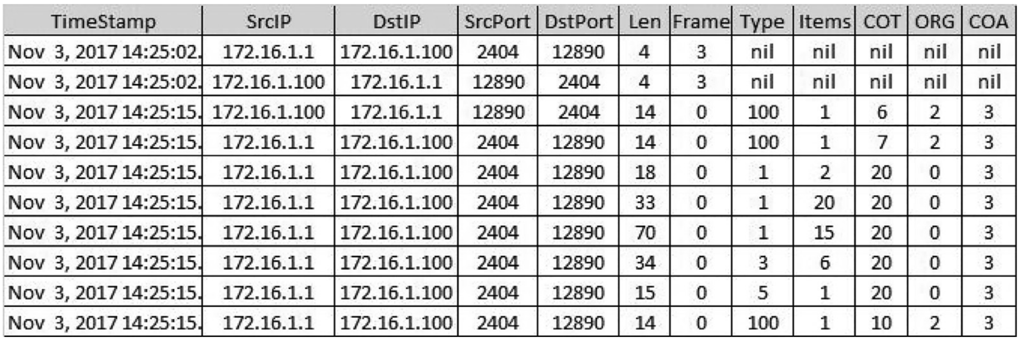


Fig. 4. Building an IEC 104 flow record from IEC 104 packets.

Table 1
Example IEC 104 flows (selected fields only).



TimeStamp	SrcIP	DstIP	SrcPort	DstPort	Len	Frame	Type	Items	COT	ORG	COA
Nov 3, 2017 14:25:02.	172.16.1.1	172.16.1.100	2404	12890	4	3	nil	nil	nil	nil	nil
Nov 3, 2017 14:25:02.	172.16.1.100	172.16.1.1	12890	2404	4	3	nil	nil	nil	nil	nil
Nov 3, 2017 14:25:15.	172.16.1.100	172.16.1.1	12890	2404	14	0	100	1	6	2	3
Nov 3, 2017 14:25:15.	172.16.1.1	172.16.1.100	2404	12890	14	0	100	1	7	2	3
Nov 3, 2017 14:25:15.	172.16.1.1	172.16.1.100	2404	12890	18	0	1	2	20	0	3
Nov 3, 2017 14:25:15.	172.16.1.1	172.16.1.100	2404	12890	33	0	1	20	20	0	3
Nov 3, 2017 14:25:15.	172.16.1.1	172.16.1.100	2404	12890	70	0	1	15	20	0	3
Nov 3, 2017 14:25:15.	172.16.1.1	172.16.1.100	2404	12890	34	0	3	6	20	0	3
Nov 3, 2017 14:25:15.	172.16.1.1	172.16.1.100	2404	12890	15	0	5	1	20	0	3
Nov 3, 2017 14:25:15.	172.16.1.1	172.16.1.100	2404	12890	14	0	100	1	10	2	3

Table 1 shows a raw format of IEC 104 flows extracted from IEC 104 communication. For space limit, not all flow items are displayed. Besides IP addresses and ports, IEC 104 flow record includes APDU length (Len), frame format (Frame), ASDU type (Type), number of information objects (Items), cause of transmission (COT), originator address (ORG) and ASDU address (COA).

Fields with `nil` value in first two rows indicate IEC 104 APDUs without the ASDU payload, i.e., U-frames (frame type = 3). Only I-frames (frame type = 0) transmit ASDUs as showed in Fig. 4.

From the list of IEC 104 flows above, we can notice that there is a controlling station with originator address ORG = 0 running on IP address 172.16.1.100. The third flow record describes an I-frame (Frame = 0) that encapsulates ASDU data sent by a controlling station with ORG = 2 to the controlled station with ASDU address 3. Type of this ASDU is 100 (interrogation command) and cause of transmission (COT) is 6 (activation). The controlled station responses with COT = 7 (activation confirmation). Then we see packets transmitted in monitoring direction from station 172.16.1.1 with originator address ORG = 0 to station 172.16.1.100. TypeID = 1 means `single point information`, typeID = 3 `double point information`, and typeID = 5 `step position information`. The station sends monitoring data of active information objects with cause of transmission COT = 20 (interrogation). Value `Items` gives a number of information objects transmitted within the ASDU.

ICS flow records contain a list of active stations and commands that were exchanged. Having ICS flows enhances visibility in of ICS communication in the smart grid.

3.4. Comparison of IP flows and ICS flows

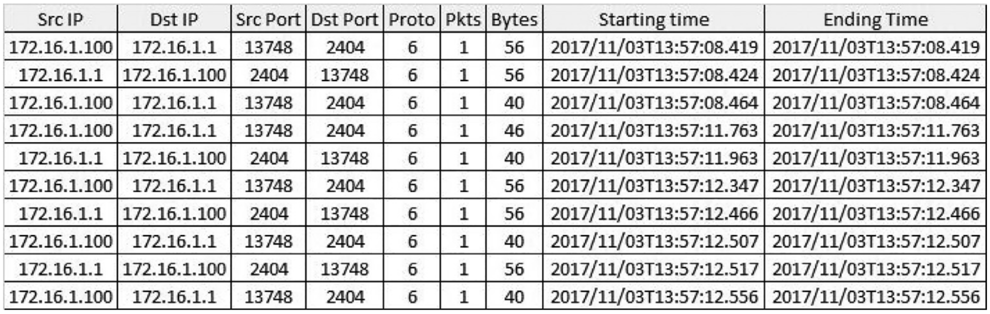
In this section we show benefits of ICS flows monitoring approach in comparison to traditional IP flows. The section discusses what levels of details can be obtained from ICS protocol headers and put into a flow record. Naturally, with the increased number of details, more processing power and time is required from the probe hardware. On the other hand, by reducing number of ICS headers processed by the probe, communication details will be lost. The following part demonstrate how various levels of details in ICS flow records impacts the visibility of ICS communication.

The case will be demonstrated on IEC 104 traffic that was captured during the attack against the IED device. The attacker sent multiple IEC 104 activation commands in order to put a device out of order. The attack is expressed by repetition of ASDUs with cause of transmission 6 (Activation), 7 (Activation Confirmation), and 10 (Activation Termination) and `double command` ASDU that invoked a switch on/off function on the target device.

3.4.1. IP flows

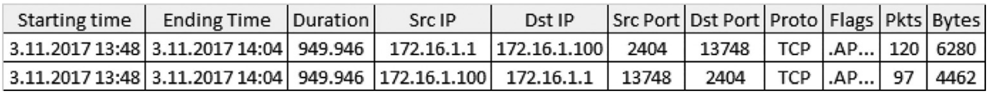
As mentioned before, traditional IP flow monitoring focuses only on Layer 3 and Layer 4, e.g., IP addresses and ports. Table 2

Table 2
IP flows during the attack obtained by Silk.



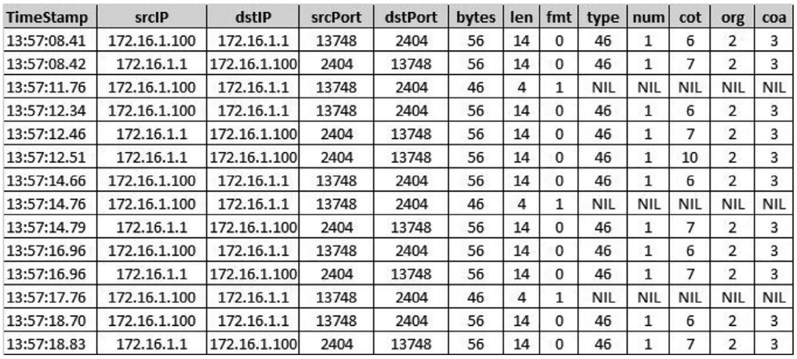
Src IP	Dst IP	Src Port	Dst Port	Proto	Pkts	Bytes	Starting time	Ending Time
172.16.1.100	172.16.1.1	13748	2404	6	1	56	2017/11/03T13:57:08.419	2017/11/03T13:57:08.419
172.16.1.1	172.16.1.100	2404	13748	6	1	56	2017/11/03T13:57:08.424	2017/11/03T13:57:08.424
172.16.1.100	172.16.1.1	13748	2404	6	1	40	2017/11/03T13:57:08.464	2017/11/03T13:57:08.464
172.16.1.100	172.16.1.1	13748	2404	6	1	46	2017/11/03T13:57:11.763	2017/11/03T13:57:11.763
172.16.1.1	172.16.1.100	2404	13748	6	1	40	2017/11/03T13:57:11.963	2017/11/03T13:57:11.963
172.16.1.100	172.16.1.1	13748	2404	6	1	56	2017/11/03T13:57:12.347	2017/11/03T13:57:12.347
172.16.1.1	172.16.1.100	2404	13748	6	1	56	2017/11/03T13:57:12.466	2017/11/03T13:57:12.466
172.16.1.100	172.16.1.1	13748	2404	6	1	40	2017/11/03T13:57:12.507	2017/11/03T13:57:12.507
172.16.1.1	172.16.1.100	2404	13748	6	1	56	2017/11/03T13:57:12.517	2017/11/03T13:57:12.517
172.16.1.100	172.16.1.1	13748	2404	6	1	40	2017/11/03T13:57:12.556	2017/11/03T13:57:12.556

Table 3
IP flows during the attack obtained by softflowd.



Starting time	Ending Time	Duration	Src IP	Dst IP	Src Port	Dst Port	Proto	Flags	Pkts	Bytes
3.11.2017 13:48	3.11.2017 14:04	949.946	172.16.1.1	172.16.1.100	2404	13748	TCP	.AP...	120	6280
3.11.2017 13:48	3.11.2017 14:04	949.946	172.16.1.100	172.16.1.1	13748	2404	TCP	.AP...	97	4462

Table 4
IEC 104 flows during the attack.



TimeStamp	srcIP	dstIP	srcPort	dstPort	bytes	len	fmt	type	num	cot	org	coa
13:57:08.41	172.16.1.100	172.16.1.1	13748	2404	56	14	0	46	1	6	2	3
13:57:08.42	172.16.1.1	172.16.1.100	2404	13748	56	14	0	46	1	7	2	3
13:57:11.76	172.16.1.100	172.16.1.1	13748	2404	46	4	1	NIL	NIL	NIL	NIL	NIL
13:57:12.34	172.16.1.100	172.16.1.1	13748	2404	56	14	0	46	1	6	2	3
13:57:12.46	172.16.1.1	172.16.1.100	2404	13748	56	14	0	46	1	7	2	3
13:57:12.51	172.16.1.1	172.16.1.100	2404	13748	56	14	0	46	1	10	2	3
13:57:14.66	172.16.1.100	172.16.1.1	13748	2404	56	14	0	46	1	6	2	3
13:57:14.76	172.16.1.100	172.16.1.1	13748	2404	46	4	1	NIL	NIL	NIL	NIL	NIL
13:57:14.79	172.16.1.1	172.16.1.100	2404	13748	56	14	0	46	1	7	2	3
13:57:16.96	172.16.1.100	172.16.1.1	13748	2404	56	14	0	46	1	6	2	3
13:57:16.96	172.16.1.1	172.16.1.100	2404	13748	56	14	0	46	1	7	2	3
13:57:17.76	172.16.1.100	172.16.1.1	13748	2404	46	4	1	NIL	NIL	NIL	NIL	NIL
13:57:18.70	172.16.1.100	172.16.1.1	13748	2404	56	14	0	46	1	6	2	3
13:57:18.83	172.16.1.1	172.16.1.100	2404	13748	56	14	0	46	1	7	2	3

shows IP flows related to the attack as obtained by the Silk NetFlow/IPFIX probe.[2]

Silk creates a unique IP flow for each ASDU packet without parsing application data. By monitoring the attack, we may notice intensive activity on the link using this approach but we cannot determine ICS operations that were requested.

Creating flows for each ASDU is not typical IP flow behavior. The standard IP flow includes five key properties (srcIP, dstIP, srcPort, dstPort, Protocol). Thus, the attack communication would yield two IP flows only, see Table 3. The result was obtained using softflowd probe.[3]

This example demonstrates limits of traditional IP flow monitoring which is not able to provide higher visibility of ICS communication and reveal ICS-specific attacks.

3.4.2. ICS flows with standard ICS headers

ICS flows monitoring adds selected L7 header values into the flow records as described in Section 3.2. Considering IEC 104 flows with headers described in Fig. 4, we obtain L7 monitoring data that can reveal activity of the attacker. Table 4 with IEC 104 flows collected during the attack shows a sending node with IP 172.16.1.100 that sends `double command` (type = 46) operation to the IEC device with address 3. This level of details does not provide information which object on the device was requested, however, such extended data are valuable source of information for statistical-based anomaly detection as described in Section 4.1 and behavior-based anomaly detection as described in Section 5.

3.4.3. ICS flows with extended headers

ICS-enabled probe can also implement advanced ICS protocol pre-processing that extracts additional ICS headers from the packets. New headers extends a set of ICS flow record values *Fprop*. In case of IEC 104 packets, the probe adds the IOA address of an information object that is involved in communication, see Table 5.

Pre-processing can go further, e.g., we can analyze IEC 104 information elements in ASDUs and IEC 104 operations applied on them, see Table 6.

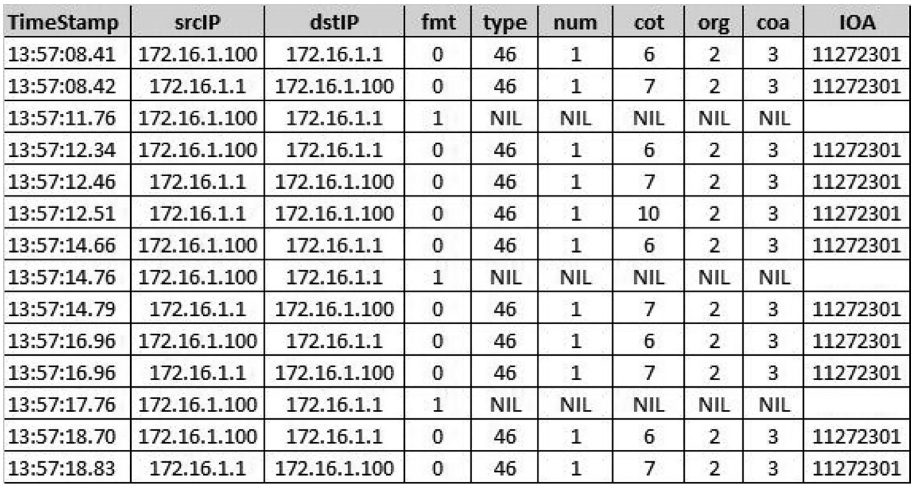
However, detailed pre-processing of packet headers requires higher computational power on the probe, more fields in IPFIX template transmitting ICS flows and also more space on the collector to store monitoring data. Thus, it is necessary to find balance between the level of details obtained by ICS monitoring and implementation demands.

3.5. Visibility of smart grid communication

As presented above, the proposed ICS flow monitoring system is flexible because monitoring data can be extracted from any ICS protocol. A great advantage of this approach is that uses standardized IPFIX protocol and IPFIX templates that allows a user to de-

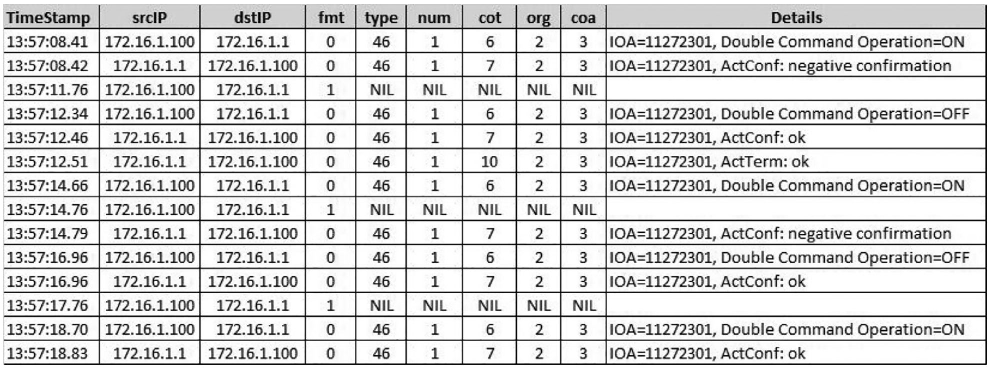
[2] See https://tools.netsa.cert.org/silk/ [May 2019].
[3] See https://github.com/irino/softflowd [May 2019].

Table 5
IEC 104 flows extended by IOA address.



TimeStamp	srcIP	dstIP	fmt	type	num	cot	org	coa	IOA
13:57:08.41	172.16.1.100	172.16.1.1	0	46	1	6	2	3	11272301
13:57:08.42	172.16.1.1	172.16.1.100	0	46	1	7	2	3	11272301
13:57:11.76	172.16.1.100	172.16.1.1	1	NIL	NIL	NIL	NIL	NIL	
13:57:12.34	172.16.1.100	172.16.1.1	0	46	1	6	2	3	11272301
13:57:12.46	172.16.1.1	172.16.1.100	0	46	1	7	2	3	11272301
13:57:12.51	172.16.1.1	172.16.1.100	0	46	1	10	2	3	11272301
13:57:14.66	172.16.1.100	172.16.1.1	0	46	1	6	2	3	11272301
13:57:14.76	172.16.1.100	172.16.1.1	1	NIL	NIL	NIL	NIL	NIL	
13:57:14.79	172.16.1.1	172.16.1.100	0	46	1	7	2	3	11272301
13:57:16.96	172.16.1.100	172.16.1.1	0	46	1	6	2	3	11272301
13:57:16.96	172.16.1.1	172.16.1.100	0	46	1	7	2	3	11272301
13:57:17.76	172.16.1.100	172.16.1.1	1	NIL	NIL	NIL	NIL	NIL	
13:57:18.70	172.16.1.100	172.16.1.1	0	46	1	6	2	3	11272301
13:57:18.83	172.16.1.1	172.16.1.100	0	46	1	7	2	3	11272301

Table 6
IEC 104 flows with information elements.



TimeStamp	srcIP	dstIP	fmt	type	num	cot	org	coa	Details
13:57:08.41	172.16.1.100	172.16.1.1	0	46	1	6	2	3	IOA=11272301, Double Command Operation=ON
13:57:08.42	172.16.1.1	172.16.1.100	0	46	1	7	2	3	IOA=11272301, ActConf: negative confirmation
13:57:11.76	172.16.1.100	172.16.1.1	1	NIL	NIL	NIL	NIL	NIL	
13:57:12.34	172.16.1.100	172.16.1.1	0	46	1	6	2	3	IOA=11272301, Double Command Operation=OFF
13:57:12.46	172.16.1.1	172.16.1.100	0	46	1	7	2	3	IOA=11272301, ActConf: ok
13:57:12.51	172.16.1.1	172.16.1.100	0	46	1	10	2	3	IOA=11272301, ActTerm: ok
13:57:14.66	172.16.1.100	172.16.1.1	0	46	1	6	2	3	IOA=11272301, Double Command Operation=ON
13:57:14.76	172.16.1.100	172.16.1.1	1	NIL	NIL	NIL	NIL	NIL	
13:57:14.79	172.16.1.1	172.16.1.100	0	46	1	7	2	3	IOA=11272301, ActConf: negative confirmation
13:57:16.96	172.16.1.100	172.16.1.1	0	46	1	6	2	3	IOA=11272301, Double Command Operation=OFF
13:57:16.96	172.16.1.1	172.16.1.100	0	46	1	7	2	3	IOA=11272301, ActConf: ok
13:57:17.76	172.16.1.100	172.16.1.1	1	NIL	NIL	NIL	NIL	NIL	
13:57:18.70	172.16.1.100	172.16.1.1	0	46	1	6	2	3	IOA=11272301, Double Command Operation=ON
13:57:18.83	172.16.1.1	172.16.1.100	0	46	1	7	2	3	IOA=11272301, ActConf: ok

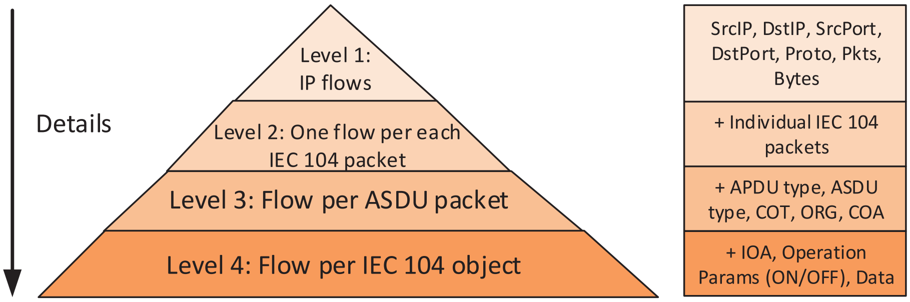

Details
Level 1: IP flows
Level 2: One flow per each IEC 104 packet
Level 3: Flow per ASDU packet
Level 4: Flow per IEC 104 object
SrcIP, DstIP, SrcPort, DstPort, Proto, Pkts, Bytes
+ Individual IEC 104 packets
+ APDU type, ASDU type, COT, ORG, COA
+ IOA, Operation Params (ON/OFF), Data

Fig. 5. Levels of IEC 104 visibility.

fine its own IPFIX flow records. This means that ICS flow monitoring can be incorporated into any SIEM system with NetFlow/IPFIX support. In addition, anomaly detection can be applied on historical and current ICS flow data stored at the IPFIX collector.

The level of visibility of ICS communication depends on implementation of ICS pre-processor and operational requirements. Fig. 5 displays multiple levels of ICS flow monitoring for IEC 104 protocol.

As you can see, traditional IP flow monitoring offers only basic statistics about communication between two hosts on Layer 3 and 4. This level does not provide sufficient visibility into ICS transmissions, as demonstrated on scenario in Table 3 where we cannot see operations used during the attack. Nevertheless, traditional IP flows provide useful data that may reveal a few types of cyber attacks, e.g., detection of a rogue device on the network, DoS attack, network scanning, etc. However, without L7 information, we are not able to disclose details of the attack.

By splitting the IP flow into ICS protocol-based flows as implemented by Silk, see Table 2, we can see detailed statistics about individual L7 packets without knowing what operation is requested, what objects are involved in communication, etc. The flows records include Layer 3 and 4 headers only. No real ICS visibility is provided.

The third level of ICS visibility requires Layer 7 processing of ICS packets by the monitoring probe. In case of IEC 104 ASDU packets, monitoring data includes APDU type, cause of transmission,

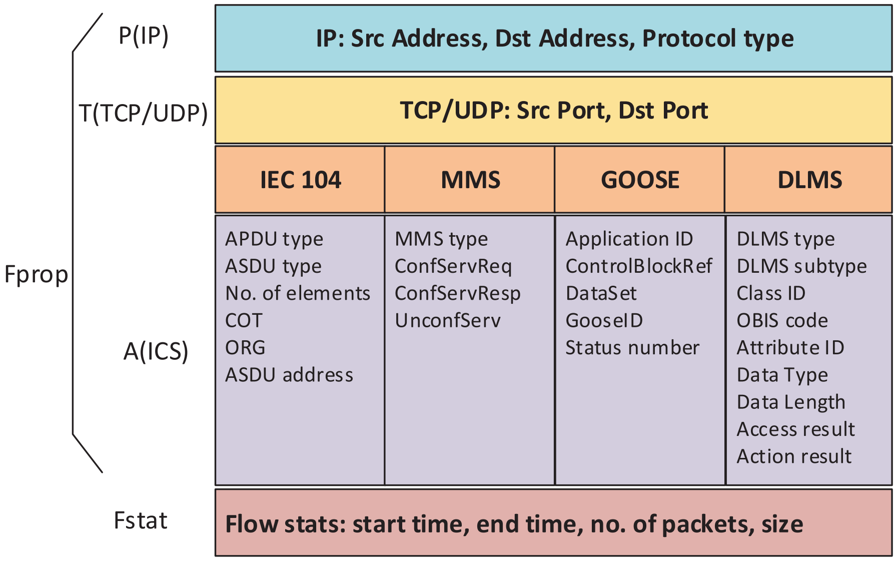


Fig. 6. Recommended ICS header extracted from ICS protocols.

originator address, ASDU address and number of transmitted information objects. This level of visibility is sufficient for observing daily communication of IEC 104 nodes. Using such data, we can create statistical profiles describing communication of information objects, detect IEC 104 resource scanning and provide rich input data for behavior-based anomaly detection.

The most detailed level of ICS visibility is obtained by processing all embedded objects transmitted in ICS communication. In case of IEC 104 communication, it means monitoring of information objects and information elements transmitted in ASDU packets. Such ICS packet processing corresponds to full packet capturing which processes both ICS packet header and payload. Full packet processing requires high CPU performance and big memory, it can be applied only on links with limited bandwidth (around 100 Mb/s) while ICS-enabled probes with level 3 monitoring can process packets on links with tens of Gb/s. Our experiments show that level 3 provides sufficient visibility of ICS communication for most use cases.

The ICS flow monitoring was also implemented in commercial probes.[4]

4. Securing smart grid networks using ICS flows

The imminence of ICS cyber security reflects the characteristics of possible threats, which can range from script kiddies, disgruntled employees, hacktivists, industrial espionage, terrorist or even state-sponsored attacks. Instead of creating a comprehensive threat model that considers all aspects related to these attacks we focus on threat categories as listed in the NISTIR 8219 report [15]. The report aims to evaluation of available techniques for the identification of activities that can be a part of attack scheme. The listed activities thus represent a representative sample of operations used by different types of attackers in the course of an attack. Our goal is to demonstrate that flow-based security monitoring and anomaly detection methods are able to identify such activities that when reported as security events can lead to attack detection.

The presented approach is flexible and can be applied to any ICS protocol by mapping specific protocol headers to ICS flow record fields. Fig. 6 presents L7 headers of common ICS protocols that can be subject of ICS flow monitoring. Recommended headers were chosen based on protocol behavior and application domain of the protocol.

The detection will be demonstrated on IEC 104 datasets using simple statistical techniques. Each header transmits a value from the set of values defined by the protocol standard. For example, `MMS type` defines 13 types of MMS packets, e.g., `confirmed request` (type = 0) that offers 86 services like `getNameList`, `read`, `write`, etc. Using ICS flow monitoring, such values become visible to network administrator and can be analyzed using anomaly detection system.

4.1. Statistical-based anomaly detection using flow data

NIST report *Securing Manufacturing Industrial Control Systems: Behavioral Anomaly Detection* [15] presents practical approaches for strengthening cyber security in the manufacturing processes using behavioral anomaly detection (BAD) tools. This approach is also useful for detecting anomalous behavior related, for example, to equipment malfunctioning. Similar to our approach, the BAD report presents non-intrusive techniques to analyze industrial network communications. Passive monitoring by NIST tools is implemented via port mirroring. The same solution is also used by ICS flow monitoring probes that passively observe the traffic mirrored to the probe.

We argue that ICS flow monitoring data can be successfully used as source data for BAD classes as described in the NISTIR report. The list of BAD capabilities observed by NIST is summarized in Table 7.

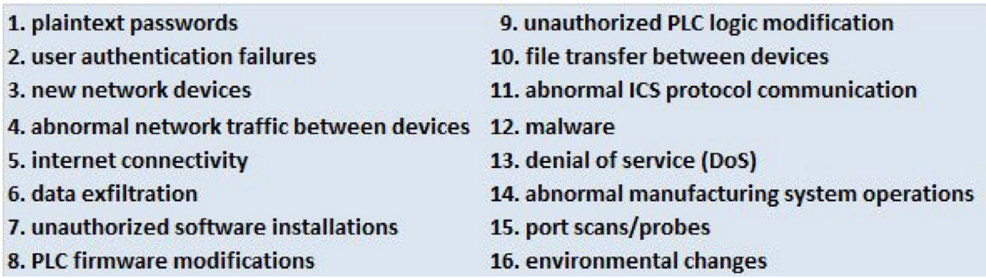
The report documents the use of BAD capabilities in two environments: a robotics-based manufacturing system and process control system in chemical industry. In the following text, we will apply selected scenarios of the report on IEC 104 communication.

4.1.1. Rogue device detection

The ICS flow data provides visibility that enables to detect rogue devices on the network. Generally, ICS networks show signs of stability in the number of connected devices as observed by Barbosa et al. [39]. Using flow based monitoring, active ICS nodes can be learnt from TCP handshake [28]. This helps to determine which station is a client, which is a server and what type of communication is established. The authors apply whitelisting on flow data

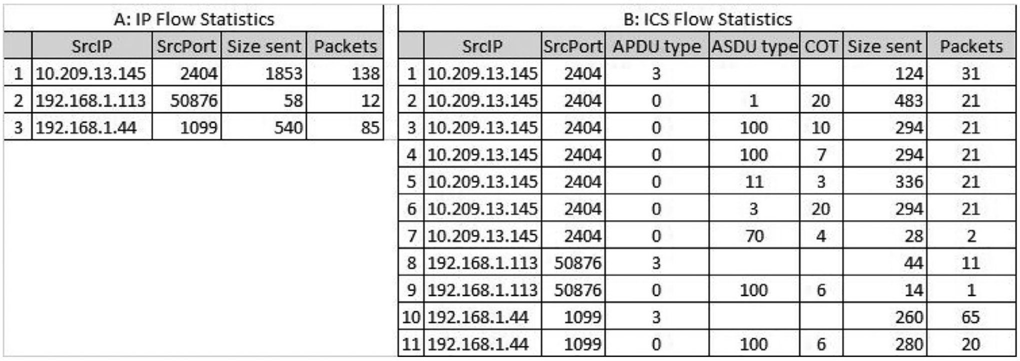
[4] See https://www.flowmon.com/en/blog [March 2017]

Table 7
Behavioral Anomaly Detection (BAD) classes [15].



1. plaintext passwords	**9. unauthorized PLC logic modification**
2. user authentication failures	**10. file transfer between devices**
3. new network devices	**11. abnormal ICS protocol communication**
4. abnormal network traffic between devices	**12. malware**
5. internet connectivity	**13. denial of service (DoS)**
6. data exfiltration	**14. abnormal manufacturing system operations**
7. unauthorized software installations	**15. port scans/probes**
8. PLC firmware modifications	**16. environmental changes**

Table 8
Analyzing ASDU types of IEC 104 communication.



A: IP Flow Statistics

	SrcIP	SrcPort	Size sent	Packets
1	10.209.13.145	2404	1853	138
2	192.168.1.113	50876	58	12
3	192.168.1.44	1099	540	85

B: ICS Flow Statistics

	SrcIP	SrcPort	APDU type	ASDU type	COT	Size sent	Packets
1	10.209.13.145	2404	3			124	31
2	10.209.13.145	2404	0	1	20	483	21
3	10.209.13.145	2404	0	100	10	294	21
4	10.209.13.145	2404	0	100	7	294	21
5	10.209.13.145	2404	0	11	3	336	21
6	10.209.13.145	2404	0	3	20	294	21
7	10.209.13.145	2404	0	70	4	28	2
8	192.168.1.113	50876	3			44	11
9	192.168.1.113	50876	0	100	6	14	1
10	192.168.1.44	1099	3			260	65
11	192.168.1.44	1099	0	100	6	280	20

which compares newly detected devices with the list of known devices. This approach works well for ICS protocols over TCP but it cannot be applied to L2 protocols like GOOSE or Modbus. Our approach creates ICS flows also for ICS protocols directly encapsulated in Ethernet, so it can be applied to protocols like GOOSE or Modbus.

Table 8 shows how rogue devices can be easily identified using ICS flow data.

Part A represents IP flow statistics aggregated by source IP address and port. We can see three communicating nodes without being able to identify their roles. Part B is created from ICS flows. It reveals communication details through APDU type, ASDU type, and cause of transmission (COT). Flows with ASDU type 1 (single point of information), 2 (double point of information, 11 (measured values), or 70 (end of initialization) are initiated by the RTU slave. A flow with ASDU type 100 (interrogation command) is sent in control direction, i.e., from the RTU master to the RTU slave. Thus, we can see also roles of devices. The anomaly detection system can learn IP addresses and roles of all communicating nodes in the system during the learning phase. When an unknown device or a device with an unknown role is detected using regular monitoring, an alarm is raised with details about the rogue device.

4.1.2. Abnormal network traffic

ICS traffic exhibits long-term stability and periodicity [17,18,40] which can be expressed in terms of communication patterns [29]. Any traffic with unusual behavior, e.g., an invalid sequence of commands, exceeding numbers of packets sent within a given time window, or an atypical combination of values in packet headers is considered as anomalous.

An example of the communication pattern is depicted in Fig. 7: activation of an IEC 104 device. The pattern includes four ASDUs exchanged between RTU master and slave. Such pattern is extracted directly from ICS flow records as seen in Table 9.

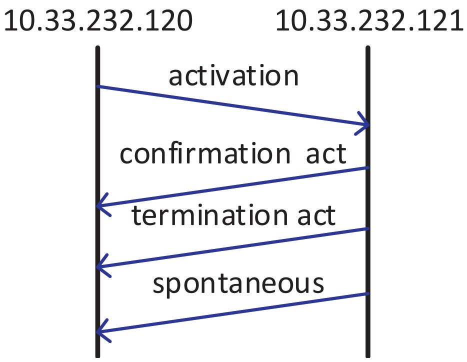


Fig. 7. IEC 104 node activation command sequence.

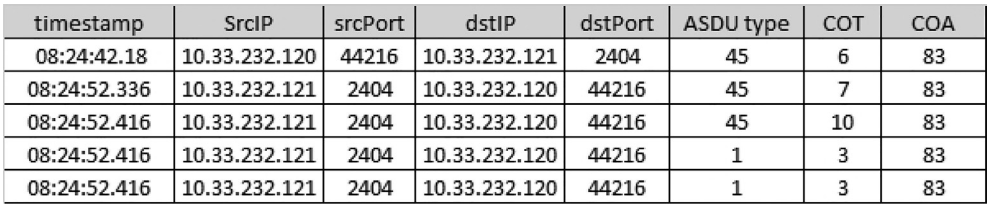
Unlike traditional IP flows, we get sequences of ICS commands exchanged between two ICS devices which can be used to create communication patterns. Patterns can be expressed by statistical models as described below or by probabilistic automata as explained in Section 5.

Statistical techniques have been considered as a resource efficient anomaly detection techniques (see, for instance, Proto et al. [41] and Caberera et al. [42]). The statistical model is formally defined as a pair $(S, \mathcal{P})$, where:

- S is the sample space of the model that comprises the set of all possible tuples of features considered in the model, e.g., number of ICS packets exchanged between ICS devices, their sizes, inter packet delay, etc.
- $\mathcal{P}$ is a set of probability distributions on S.

The statistical model is computed from the sample of ICS flows representing normal behavior of the system as follows:

Table 9
IEC 104 node activation.



timestamp	SrcIP	srcPort	dstIP	dstPort	ASDU type	COT	COA
08:24:42.18	10.33.232.120	44216	10.33.232.121	2404	45	6	83
08:24:52.336	10.33.232.121	2404	10.33.232.120	44216	45	7	83
08:24:52.416	10.33.232.121	2404	10.33.232.120	44216	45	10	83
08:24:52.416	10.33.232.121	2404	10.33.232.120	44216	1	3	83
08:24:52.416	10.33.232.121	2404	10.33.232.120	44216	1	3	83

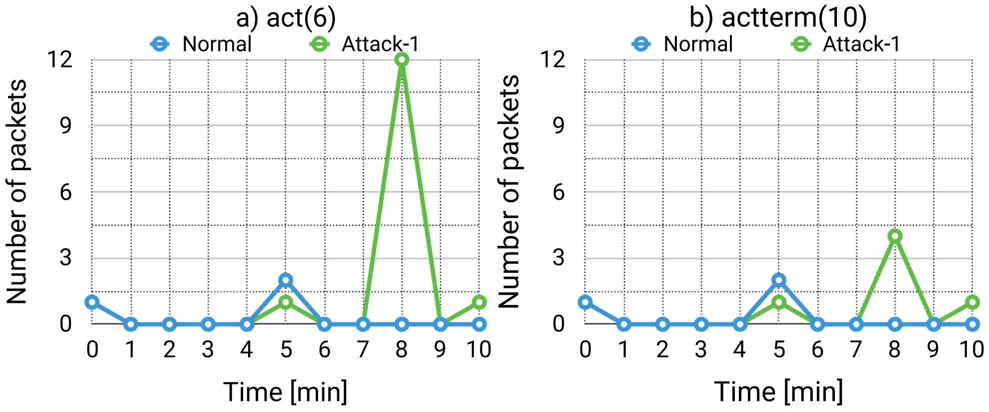


Fig. 8. Distribution of (a) Act and (b) ActTerm packets during the normal usage and the attack.

1. The flows are grouped in time windows of the predefined fixed size, e.g., 60 s.
2. For flows within the given window, features that are part of the sample space of a model are extracted and added as a new tuple to the set of samples.
3. Step 2 is repeated until all windows are processed.
4. Finally, the set of probability distributions $\mathcal{P}$ are determined from all collected samples using a statistical inference method.

In the demonstration scenario, we observe only the number of specific ICS packets transmitted during a given time period, however, additional features can be also added as shown in Crotti et al. [43]. For anomaly detection, different statistical techniques can be applied. In this scenario, we consider a simple method that computes a threshold to define the anomaly of traffic. The threshold value is computed as follows:

$$T = Q_3 + t \cdot (Q_3 - Q_1)$$

where Q_i is the ith quartile of the set of samples. Constant t is used to adjust the threshold value. Fig. 8 depicts a scenario where an attacker manipulates with the IED node by sending IEC 104 Activation and Activation Termination ASDUs.

The blue line represents the number of Act commands (a), and ActTerm commands (b), respectively, sent during 10 minutes of the normal traffic. The green line depicts the number of Act and ActTerm commands during the attack. We can see a peek where unexpected number of Act commands was sent to the IED devices. By comparing the number of sent Act ASDUs with the number of received ActTerm ASDUs we notice that the number of Acts and ActTerms messages is equal for normal communication which means that all Act ASDUs are correctly confirmed by ActTerm ASDUs. However, during the attack, some Act commands are ignored due to the high number of requests. The presented attack simulates the behavior of the Industroyer malware [5] when an IED device was continuously switching on and off within a few seconds interval. Using ICS flow monitoring we are able to detect such behavior.

4.1.3. Data exfiltration between ICS devices and file transfer

Data exfiltration describes an attempt to download data from an ICS system without proper authorization. Observing communication patterns, as mentioned above, can detect these types of attacks too. Fig. 9 shows long term communication between IEC 104 nodes within 3 days. We can see the stable number of ASDUs transmitted over the communication link.

The green line represents ASDUs with COT = 3 (spontaneous event) sent in monitoring direction to the RTU master. The blue line depicts file transfers between two nodes (COT=13, data transmission). Using ICS flows, we can detect unauthorized data transfer by observing ICS flows with COT = 13 and ASDU type 120–127 (file transfer). Without ICS visibility, we can only identify abundant ICS transmission without understanding its meaning.

4.1.4. Resource scanning

Using ICS flow monitoring, we can detect not only new devices but also unknown or invalid resources. ICS device and port scanning is usually a preparatory phase before the cyber attack when the attacker tries to map network resources using various tools. A typical device scanning attack on the IP layer is performed by `nmap` tool. Some tools provide resource scanning on the L7 layer, which is more difficult to detect.

Scanning attacks can be easily revealed by flow monitoring. IP flows show a scanning attack based on the enumeration of IP addresses and ports. Using ICS flows, we can detect resource scanning on the L7 layer. Typically, the resource scanning attack yields a large number of packets targeting one device within a short time and getting invalid responses. Such behavior can be detected using statistical patterns, as mentioned above.

For example, by observing COT values in IEC 104 communication, see Table 10, we can detect packets with unknown addresses (COT = 46), unknown information object (COT = 47) or unknown COT type (COT = 45).

All these packets indicate either resource scanning attack or misconfiguration. In both cases, it is necessary to take action.

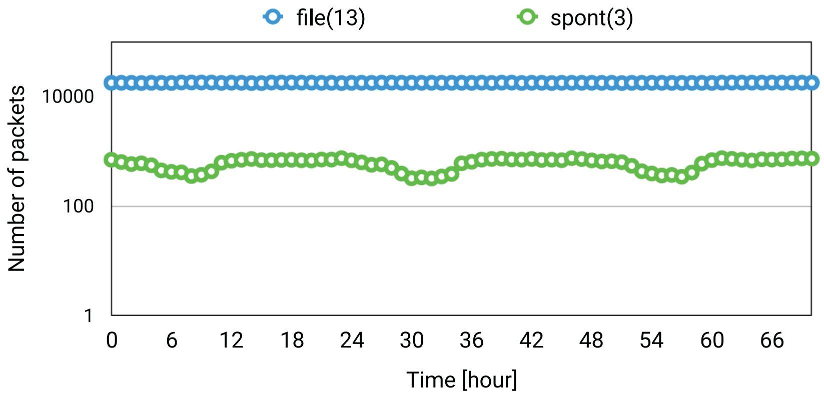
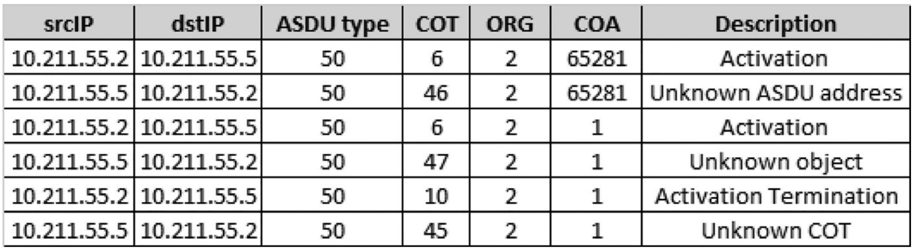


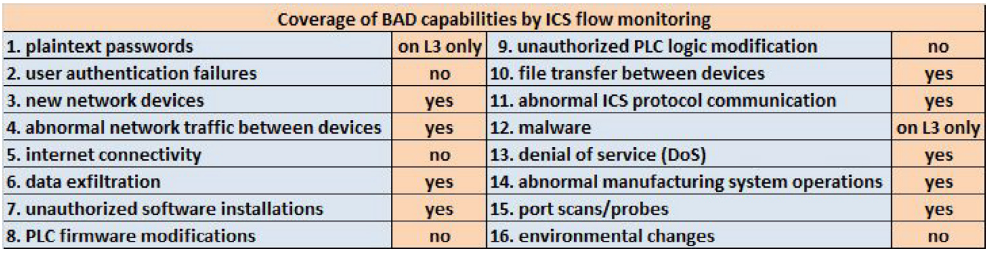
Fig. 9. IEC 104 communication.

Table 10
IEC 104 wrong data .



srcIP	dstIP	ASDU type	COT	ORG	COA	Description
10.211.55.2	10.211.55.5	50	6	2	65281	Activation
10.211.55.5	10.211.55.2	50	46	2	65281	Unknown ASDU address
10.211.55.2	10.211.55.5	50	6	2	1	Activation
10.211.55.5	10.211.55.2	50	47	2	1	Unknown object
10.211.55.2	10.211.55.5	50	10	2	1	Activation Termination
10.211.55.5	10.211.55.2	50	45	2	1	Unknown COT

Table 11
Coverage of BAD capabilities by ICS monitoring .



Coverage of BAD capabilities by ICS flow monitoring			
1. plaintext passwords	on L3 only	9. unauthorized PLC logic modification	no
2. user authentication failures	no	10. file transfer between devices	yes
3. new network devices	yes	11. abnormal ICS protocol communication	yes
4. abnormal network traffic between devices	yes	12. malware	on L3 only
5. internet connectivity	no	13. denial of service (DoS)	yes
6. data exfiltration	yes	14. abnormal manufacturing system operations	yes
7. unauthorized software installations	yes	15. port scans/probes	yes
8. PLC firmware modifications	no	16. environmental changes	no

4.2. *ICS flows and BAD capabilities*

As demonstrated above, ICS flows provide a valuable source of monitoring data for the successful detection of common security threats. In comparison with network-based commercial tools like CyberXand and SecurityMatters SilentDefence, the monitoring is based on a standardized IPFIX framework with extended ICS protocol headers extraction. Unlike IP flow monitoring, it gives more details about ICS communication, which are essential for cyber threat detection.

Since the system is based on passive monitoring of ICS traffic, some anomaly detection (BAD) capabilities as defined in NIST report [15] can be covered only partially or not at all. For example, flow monitoring is not able to detect plain text password transmission (BAD no. 1) in ICS protocols, however, it is able to detect protocols that may transmit plain passwords, e.g., SMB, telnet or FTP. ICS flow monitoring is not able to detect failed internet connectivity (BAD no. 5) because it uses passive traffic observation and this capability requires active testing. Also, it is not able to detect malware transmission since it does not process packet payload. However, the system can recognize unusual HTTP transfers between two devices, e.g., between an external site and the control station, which can be an indicator of malware activity.

Table 11 shows how ICS flow monitoring and analysis covers BAD capabilities.

Many BAD capabilities can be fully covered using ICS flow data (value `yes`). In classes like plain text passwords detection or malware detection, we cannot detect these threats from ICS flows but it is possible to identify suspicious activity by analyzing L3 values in flow records. There are also threats related to the hardware, e.g., BAD capabilities no. 8 and 9, which cannot be detected by flow-based monitoring.

5. Anomaly detection using probabilistic automata

This part presents a proof-of-concept model of profiling ICS communication using probabilistic automata. The main idea behind this concept comes from observation that machine to machine communication between two ICS devices or RTU master and slave is stable and communication patterns composed of exchanged commands do not change very often. Thus, by observing typical communication sequences between two ICS devices we can create a finite state automaton (FSM) representing typical communication traces. FSM is augmented by probability value on the edge which says that several traces share a common sub-string.

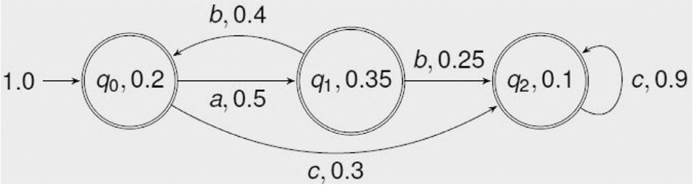


Fig. 10. An example of simple DPA.

This idea has been already explored in Wang et al. [44] where the authors infer probabilistic time automaton (PTA) from network communication traces. They apply the approach on SMTP protocol and P2P file sharing. Similar approach was also used by Krueger et al. [45] who create deterministic finite automata (DFA) for SIP, DNS and FTP communication.

5.1. Deterministic probabilistic automata (DPA)

The probabilistic automaton is generalization of a non-deterministic finite automaton (NFA) [46] with state and transition probabilities.

Definition 5.1. Probabilistic automaton (PA) is a tuple $\mathcal{A} = (\Sigma, Q, \delta, \mathbb{I}, \mathbb{F})$ where

- Σ is an alphabet,
- Q is a finite set of states,
- $\delta : Q \times \Sigma \times Q \to \mathbb{Q} \cap [0, 1]$ is a (total) transition function assigning probabilities to each transition. If $\delta(q_1, a, q_2) = 0$ the transition from q_1 to q_2 via symbol a is not actually there,
- $\mathbb{I} : Q \to \mathbb{Q} \cap [0, 1]$ is a mapping assigning initial-state probabilities, and
- $\mathbb{F} : Q \to \mathbb{Q} \cap [0, 1]$ is a mapping assigning final-state probabilities.

Moreover the following conditions hold:

$$\sum_{q \in Q} \mathbb{I}(q) = 1,$$

and $\forall q \in Q$,

$$\mathbb{F}(q) + \sum_{a \in \Sigma, r \in Q} \delta(q, a, r) = 1.$$

Definition 5.2. A PA $\mathcal{A} = (\Sigma, Q, \delta, \mathbb{I}, \mathbb{F})$ is called deterministic, if

- $\exists q \in Q.\ \mathbb{I}(q) = 1$ (unique initial state),
- $\forall q \in Q, \forall a \in \Sigma.\ |\{r \mid \delta(q, a, r) > 0\}| = 1$ (for each $q \in Q$ and $a \in \Sigma$ there is a unique successor).

Further we aim at the definition of the probability of a word w. In the following, let $\mathcal{A} = (\Sigma, Q, \delta, \mathbb{I}, \mathbb{F})$ be a PA and $w = a_1 \ldots a_n \in \Sigma^*$ be a word. A trace π of the word w is a sequence $\pi = (q_0, a_1, q_1) \ldots (q_{n-1}, a_n, q_n)$ where $\delta(q_{i-1}, a_i, q_i) > 0$, $\mathbb{I}(q_0) > 0$, and $\mathbb{F}(q_n) > 0$ for $1 \leq i \leq n$. (informally, it is a path through automaton via w ending in a state with non-zero accepting probability). The set of all traces of a word w we denote as Π_w. Probability of the path π is then given as $\mathcal{P}_{\mathcal{A}}(\pi) = \mathbb{I}(q_0) \cdot \delta(q_0, a_1, q_0) \ldots \delta(q_{n-1}, a_n, q_n) \cdot \mathbb{F}(q_n)$ (we multiply probabilities of transitions occurring in the sequence). Probability of a word w is then given as $\mathcal{P}_{\mathcal{A}}(w) = \sum_{\pi \in \Pi_w} \mathcal{P}_{\mathcal{A}}(\pi)$ (it is a sum of probabilities of all traces for a given word). This is demonstrated on Example 5.1.

Example 5.1. Consider a PA from Fig. 10 where states are labeled with a name and the accepting probability, and transactions with a symbol and the probability taking the transaction. Then probability of accepting word abc is as follows: $\mathcal{P}_{\mathcal{A}}(abc) = 1.0 \cdot 0.5 \cdot 0.25 \cdot 0.9 \cdot 0.1 + 1.0 \cdot 0.5 \cdot 0.4 \cdot 0.3 \cdot 0.1$.

In our approach, DPAs represent communication sequences between two ICS devices obtained from ICS flows. Then, anomaly detection has two phases: the learning phase where DPAs are created from samples of ICS communication, and the detection phase where unknown flow sequences are classified using DPAs. Both phases are described in the following text.

5.2. Learning DPA using ICS flows

The learning phase uses algorithm *Alergia* originally described in Carrasco and Oncina [47] and de la Higuera [46]. The algorithm takes as an input multiset of strings S and outputs a deterministic probabilistic automaton corresponding to S, see Algorithm 1.

Algorithm 1: The algorithm Alergia.

```
Input: A multiset of strings $S$, $\alpha > 0$, $t_0 > 0$
Result: A DPA $\mathcal{B}$
$\mathcal{A} \leftarrow \mathsf{Fpt}(\mathcal{S})$;
$Red \leftarrow \{q_\varepsilon\}$;
$Blue \leftarrow \{q_a \mid a \in \Sigma \cap \mathsf{Pref}(S)\}$;
while $Choose\ q_b\ from\ Blue\ s.t.\ \mathcal{C}(q_b) \geq t_0$ do
    if $\exists q_r \in Red$ : $\mathsf{Compatible}(\mathcal{A}, \sqcup_{\nabla}, \sqcup_{\lfloor}, \alpha)$ then
        $\mathcal{A} \leftarrow \mathbin{\wedge\!\!\wedge}(\mathcal{A}, \sqcup_{\nabla}, \sqcup_{\lfloor})$;
    else
        $Red \leftarrow Red \cup \{q_b\}$;
    $Blue \leftarrow \{q_{ua} \mid ua \in \mathsf{Pref}(S) \wedge q_u \in Red\} \setminus Red$;
return $\mathcal{B} = \mathsf{Normalize}(\mathcal{A})$;
```

The algorithm proceeds in the following steps:

1. Create a prefix tree with numbers of strings from S. The resulting DPA will be created by iterative merging of states of this prefix tree.
2. From the root state of the prefix tree iteratively search for "similar" states. For this maintain two sets of states *Red* and *Blue* set. The *Blue* set contains still unprocessed states of the prefix tree. In each iteration take a blue state. If there is a "similar" red state merge these two states together (and update values on the transitions starting at the states). Finally update the *Red* and *Blue* set.
3. The automaton constructed in the previous steps contain an integer on each transition. Normalize these values to obtain a DPA with probabilities on transitions.

Details about construction of the DPA, including computation of a set of prefixes $\mathsf{Pref}(S)$, merging states and normalization is written in report [48].

For learning DPAs we use tool `Treba` [49] that implements algorithm Alergia [47] with three parameters: α that determines when to merge two states (the bigger the value, the less merges are made), t_0 which determines the minimum number of strings to a state that is being considered for merging, and *prior(p)* which gives an amount of probability that is used to complete the resulting DPA transition function.

To train DPAs for IEC 104 communication, we employed several datasets with normal IEC 104 traffic, see Table 12. Some dataset

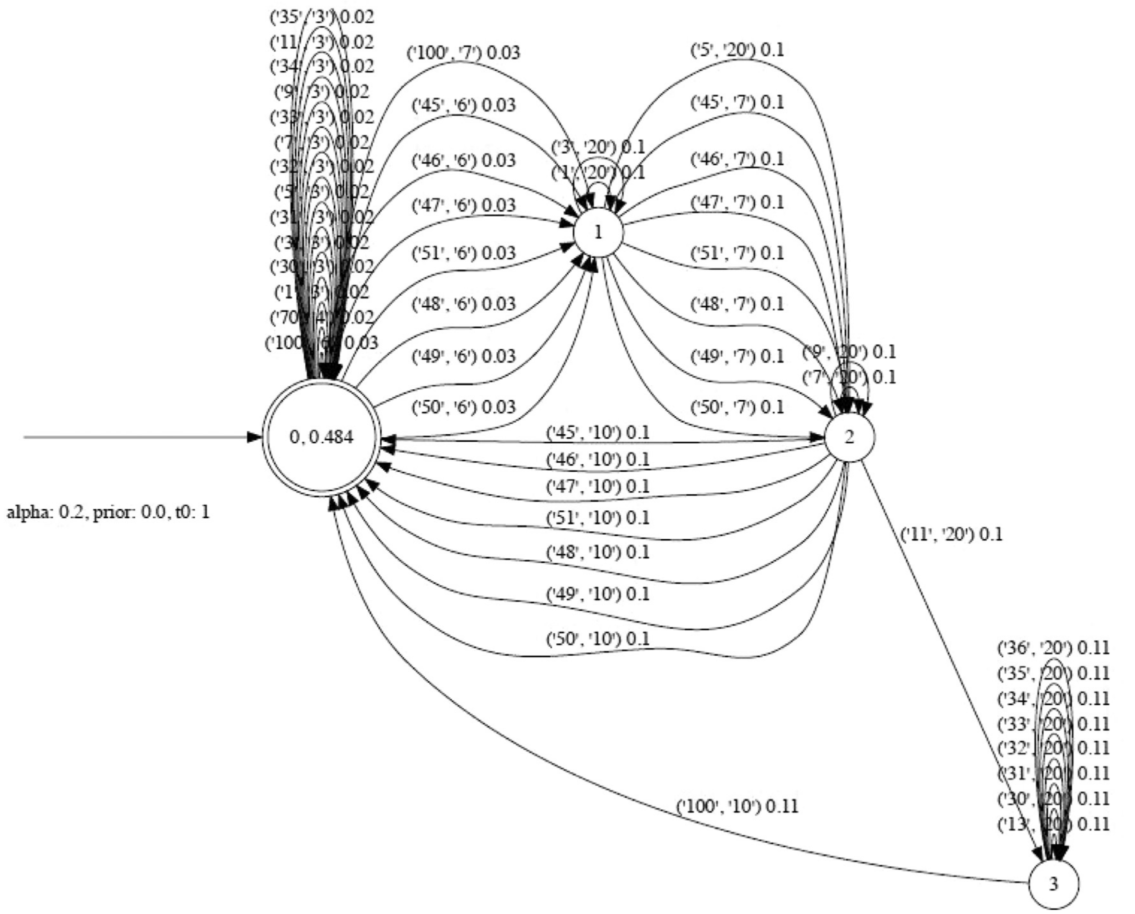


Fig. 11. DPA with ASDU and COT.

Table 12
IEC 104 datasets used for training.

Dataset	Packets	ICS flows
`iec104`	115	91
`10122018-104Mega-ioa` (part 0)	9905	8876
`10122018-104Mega-ioa` (part 1)	3011	206
`10122018-104Mega-ioa` (part 2)	91,617	82,460
`13122018-mega104-ioa` (part 0)	74,205	4798
`13122018-mega104-ioa` (part 1)	62,040	55,772
`mega104-14-12-18-ioa`	14,597	9657
`mega104-17-12-18-ioa`	58,930	9657
`SCADA-normal-ioa`	127	27

files are split into several parts so that each part contains only one end-to-end communication.

From captured data, we obtain ICS flows. For each end-to-end data communication, we extract a multiset of training strings S which are in fact pairs `(ASDU type, COT)` representing abstraction of IEC 104 messages exchanged between these two entities. A time sequence of such pairs is based on timestamps and represents a *logical conversation* between two nodes. Thus, the conversations are strings of a language representing the communication.

During learning phase we also tested combination of *Alergia* parameters α and t_0 where $\alpha \in \{0.05, 0.1, 0.15, 0.2\}$ and $t_0 \in \{2, 5, 10, 20, 50, 100, 200\}$.The parameter *prior* was set to 0.0.

Resulting DPA automata contain states in form `{state-id,accepting probability}` and transitions in form `{symbol, probability}`, where the symbol is a pair (`asduType`, `cot`).

Graphical representation of a DPA built from IEC 104 communication extracted from `iec104` dataset is depicted in Fig. 11.

In the automaton, we can see many spontaneous conversations (single messages with `cot` = 3). Beside the spontaneous conversations the model contains conversations transferring values. These conversations begin with activation command (`cot` = 6) and ends with termination activation (`cot` = 10). The model represents the communication learnt from IEC 104 flows extracted from `iec104` dataset.

Another DPA representing file transfer using IEC 104 messages is depicted in Fig. 12.

From the automaton we can conclude that a half of all conversations are spontaneous and a half were file transfer messages. The file transfer begins with an initialization phase (`asduType` = 122;120;122), followed by an arbitrary number of file segments (`asduType` = 125), and finished by an acknowledgement (`asduType` = 123;124).

Results of other experiments with our datasets and details about creating DPAs are described in the technical report [48].

5.3. Anomaly detection using DPA

Here, we present our first results with detection of an attack using IEC 104 communication using deterministic probabilistic automata. We use the same scenario as described in Section 4.1.2 where we applied statistical approach. Here, instead of observing time delays and number of transmitted packets, we focus on conversations, i.e., sequences of transmitted messages sent between IEC 104 nodes.

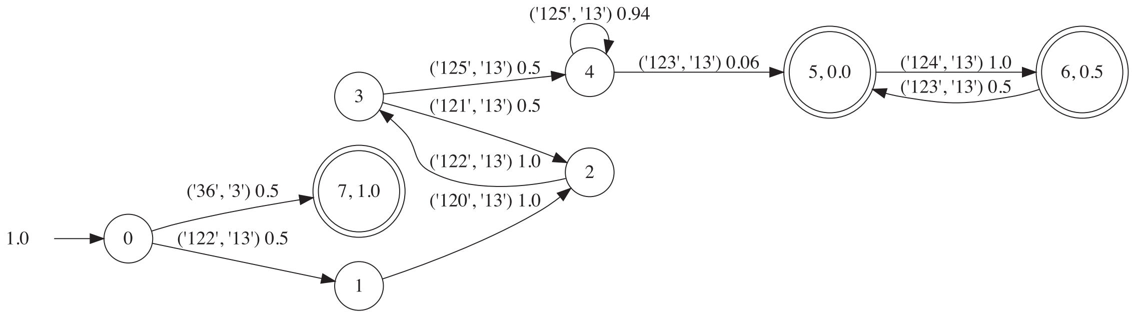


Fig. 12. DPA representing file transfer.

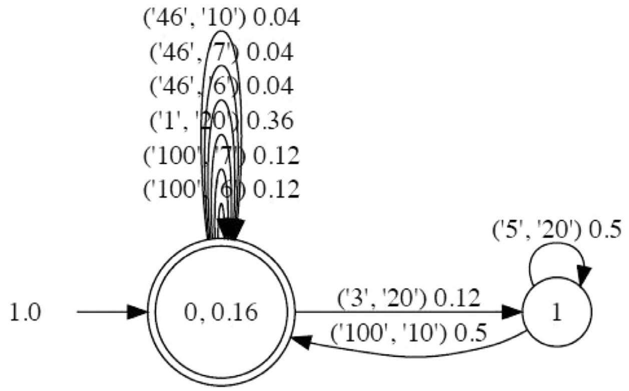


Fig. 13. DPA with $\alpha = 0., t_0 = 1$.

We demonstrate detection using two datasets SCADA-normal-ioa and SCADA-attack1-ioa2. The former contains normal traffic while the latter includes an attack. The attack consists of repetition of initialization messages. The anomaly detection is performed in the following steps:

1. In the first step, we split ICS flows from SCADA-attack1-ioa2 into a multiset of conversations $\{c_1, \ldots, c_n\}$.
2. We take a DPA $\mathcal{A}$ representing communication sequence of normal traffic learnt from SCADA-normal-ioa.
3. For each c_i where $1 \leq i \leq n$ we compute probability of the conversation c_i wrt. $\mathcal{A}$. In other words, we compute $\mathcal{P}_{\mathcal{A}}(c_i)$. If the probability is zero, the communication sequence c_i ends in an unknown state. It means unexpected communication sequence which is anomaly.

First we show an unsuccessful detection using overgeneralized DPA. Consider a DPA obtained by learning algorithm Alergia with parameters $\alpha = 0.1,\ t_0 = 1$ using dataset SCADA-normal-ioa. The automaton is show in Fig. 13.

If we apply the detection steps described above over this DPA we did not get any anomaly. The reason is excessive generalization (over-approximation) of the model, which is caused by a small number of conversations for learning combined with a small value of t_0.

If we increase parameter $t_0 = 2$, we obtain a new DPA with more states, see Fig. 14. During detection phase, we receive an alert because conversation {(46, 6)(46, 7)(46, 6)(46, 7)(46, 10)} has a zero probability over this DPA. By further analysis we can see that the conversation contains multiple initialization messages which is not a legitimate behavior learnt from the training dataset.

We also provided experiments with other datasets. We used the one third of each dataset for training and the two thirds for detection. Since the datasets are regular and did not contain any unusual communication sequences, no anomaly was found during detection phase. We also tested various parameters. Selected resuls are in Table 13. The table show datasets used for learning (the first third of the file) and detection (two thirds of the file). Only the last row mentions a file used for detection while learning was done using SCADA-normal-ioa. The table also shows the number of conversations used for training, the number of states of a DPA created from samples and representing communication in the dataset, and the results of detection. We can see that even for higher number of training strings the number of DPA states is acceptable and does not increases rapidly. This is because of stability of communication and predictability of exchanged communication sequences between IEC 104 nodes.

Full results of our experiments are in Matoušek et al. [48].

5.4. Discussion

As said before, Section 5 extends the research of flow based monitoring of ICS communication toward applications, namely

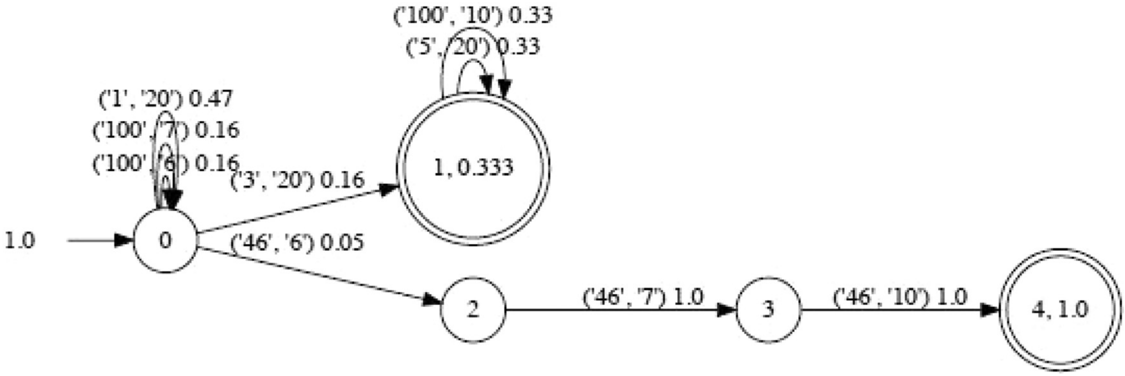


Fig. 14. DPA with $\alpha = 0., t_0 = 2$.

Table 13
Results of learning and detection for pairs (`asduType`, `cot`).

Dataset	Parameters	Conv.	States	Anomaly
`iec104`	$\alpha = 0.2,\ t_0 = 1$	31	3	No
`10122018-104Mega-ioa`	$\alpha = 0.05,\ t_0 = 200$	6927	7	No
`10122018-104Mega-ioa` (part 0)	$\alpha = 0.1,\ t_0 = 10$	503	7	No
`13122018-mega104-ioa`	$\alpha = 0.1,\ t_0 = 5$	91,957	7	No
`13122018-mega104-ioa` (part 1)	$\alpha = 0.2,\ t_0 = 200$	3603	7	No
`mega104-14-12-18-ioa`	$\alpha = 0.1,\ t_0 = 20$	9125	7	No
`mega104-17-12-18-ioa`	$\alpha = 0.15,\ t_0 = 200$	37,661	2	No
`SCADA-normal-attack1`	$\alpha = 0.1,\ t_0 = 2$	7	4	Yes

anomaly detection based on ICS sequences. Our primary interest of this experiment was to find out if it is feasible to apply DPAs for anomaly detection of ICS sequences and Alergia algorithm for building DPA automatically. Preliminary results show that such research is promising and the number of states and edges is acceptable for describing ICS communication.

It is obvious that learnt DPAs can be used for detecting anomalies. For detection, a sequence of input messages obtained from ICS flow records is divided into conversations $\{c_1, \ldots, c_n\}$. Then, we compute probability for each conversation with respect to the learnt DPA $\mathcal{A}$ which represents normal behavior. If probability $\mathcal{P}_{\mathcal{A}}(c_i)$ is lower then threshold, it indicates anomaly or a new legitimate communication sequence that was not present in learning dataset. Having a dataset describing ICS communication over the network within a week, it should be sufficient to use it for learning since ICS communication shows stability and periodicity over the week period [18].

As shown above, parameters α and t_0 also play important role during learning phase because they determine level of abstraction. If abstraction is high (lower values of the parameters), over-approximated DPA is built which also accepts anomalous conversation. It depends on number of strings in a training dataset too. In our future work we plan to find optimal values of these parameters depending on the characteristics of the dataset so that over-approximation is eliminated.

6. Conclusion

Industrial systems are attractive targets for cyber criminals, activists, professional hackers, disgruntled employees, etc. Critical infrastructure is among the most significant concerns for cyber warfare/cyber defense organizations. Several vulnerabilities have been exploited in ICS systems, demonstrating the need to improve the security of these critical systems. Many attacks were not correctly detected due to inadequate or wrongly implemented protection. To provide adequate network security monitoring in ICS systems, visibility into communication is an essential requirement. Although many ICS systems use IP-based networking, standard enterprise security systems cannot analyze ICS application protocols, thus it is not able not to provide the required in-sight to network transactions. In this paper, we have introduced the concept of the ICS monitoring system employing IPFIX flows extended with application-level data extracted from ICS communication protocols. The approach was demonstrated on the IEC 104 communication, which is the standard protocol suite for smart grid networks. The proposed ICS flow monitoring is passive and does not affect network performance.

Anomaly detection techniques can be integrated with the flow-based network monitoring system. Thanks to ICS transaction visibility, it is possible to detect behavior anomaly detection cases as specified in the NISTIR 8219 report. We have demonstrated the use of ICS flow data in statistical-based detection techniques in Section 4. In particular, we have provided examples of the detection of rogue devices, abnormal network traffic, data exfiltration, and resource scanning. The simple statistical model was created based on regular traffic and used for the detection of anomalies in ICS traffic.

Another method for network profiling and anomaly detection has been introduced in Section 5. The network model is represented by deterministic probabilistic automata. Using ICS flow data, we can observe L7 conversation and represent them by DPAs. During the learning phase, DPAs are created from conversations between pairs of communicating network hosts. In the detection phase, we feed the automaton with messages of actual communication. The automaton yields the probability value, which generates an alarm if less than the defined threshold. Automata represent the refined model for a network communication profile by observing typical communication sequences between pairs of ICS devices. As a machine to machine communication between two ICS devices or RTU master and slave is stable and communication patterns composed of exchanged commands do not change very often this model is reliable and precise.

We have demonstrated the advantages of applying flow based network monitoring in the domain of ICS as one of the behavioral anomaly detection security methods that according to the NIST report represents the key element in a complex ICS security solution. The future work focuses on a combination of different anomaly detection techniques for ICS IPFIX data, including presented statistical-based and automata-based methods. Further, we consider employing a machine learning approach for ICS traffic pattern classification.

Declaration of Competing Interest

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

Acknowledgments

This work is supported by Brno University of Technology project “Application of AI methods to cyber security and control systems” (2020–2022), no. FIT-S-20-6293, and project “Security Monitoring of ICS Communication (Bonnet)” (2019–2022), no. VI20192022138, funded by Ministry of Interior of the Czech Republic.

References

[1] Knapp ED, Langill JT. Industrial network security. Securing critical infrastructure networks for smart grid, SCADA, and other industrial control systems. Syngress; 2015.

[2] Lee RM, Assante MJ, Conway T. Analysis of the Cyber Attack on the Ukrainian Power Grid. Defense Use Case. Technical Report. Electricity Information Sharing and Analysis Center (E-ISAC); 2016.

[3] Assante MJ, Lee RM, Conway T. Modular ICS Malware. Technical Report. Electricity Information Sharing and Analysis Center (E-ISAC); 2017.

[4] Cherepanov A. Win32/Industroyer. A new threat for industrial control systems.. Technical Report. ESET; 2017.

[5] Dragos. CrashOverride. Analysis of the Threat of Electric Grid Operations.. Technical Report. Dragos Inc.; 2017.

[6] Assante MJ, Lee RM. The Industrial Control System Cyber Kill Chain. Technical Report. SANS Institute; 2015.

[7] ENISA. Communication Network Dependencies for ICS/SCADA Systems. Technical Report. European Union Agency for Network and Information Security (ENISA); 2016.

[8] Stouffer K, Pillitteri V, Abrams M, Hahn A. Guide to Industrial Control Systems (ICS) Security. Technical Report. National Institute of Standards and Technology; 2015.

[9] Presuhn R, Case J, McCloghrie K, Rose M, Waldbusser S. Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP). IETF RFC 3416; 2002.

[10] Claise B. Cisco Systems NetFlow Services Export Version 9. IETF RFC 3954; 2004.

[11] Gerhards R. The Syslog Protocol. IETF RFC 5424; 2009.

[12] Claise B, Trammel B, Aitken P. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information. IETF RFC 7011; 2013.

[13] IEC. Telecontrol equipment and systems - Part 5–104: Transmission protocols - Network access for IEC 60870-5-101 using standard transport profiles. Standard. IEC 60870-5-104:2006. Geneva: International Electrotechnical Commission; 2006.

[14] Matoušek P. Description and Analysis of IEC 104 Protocol. Technical Report FIT-TR-2017-12. Brno University of Technology; 2017.

[15] McCarthy J, Powell M, Stouffer K, Tang C, Zimmerman T, Barker W, Ogunyale T, Wynne D, Wiltberger J. Securing Manufacturing Industrial Control Systems: Behavior Anomaly Detection. Technical Report NISTIR-8219. National Institute of Standards and Technology; 2018.

[16] Barbosa RRR. Anomaly Detection in SCADA Systems: A Network Based Approach. University of Twente; 2014. Ph.D. thesis.

[17] Wang W, Lu Z. Cyber security in the smart grid: survey and challenges. Comput Netw 2013;57(5):1344–71.

[18] Barbosa RRR, Sadre R, Pras A. Towards periodicity based anomaly detection in SCADA networks. In: Proceedings of 2012 IEEE 17th international conference on emerging technologies factory automation (ETFA 2012); 2012. p. 1–4.

[19] Lu Z, Lu X, Wang W, Wang C. Review and evaluation of security threats on the communication networks in the smart grid. In: Milcom 2010 military communication conference; 2010. p. 1830–5.

[20] Miller B, Rowe DC. A survey of SCADA and critical infrastructure incidents. In: In Proceedings of the 1st annual conference on research in information technology, RIIT '12. ACM; 2012. p. 51–6.

[21] Kwon Y, Lee S, King R, Lim J, Kim H. Behavior analysis and anomaly detection for a digital substation on cyber-physical system. Electron (Switzerland) 2019;8(3):1–24.

[22] Maynard P, McLaughlin K, Haberler B. Towards understanding man-in-the-middle attacks on IEC 60870-5-104 SCADA networks. In: Proceedings of the 2nd international symposium on ICS & SCADA cyber security research 2014. UK: BCS; 2014. p. 30–42.

[23] Horkan M. Challenges for IDS/IPS Deployment in Industrial Control Systems. Technical Report. SANS Institute; 2015.

[24] Jarmakiewicz J, Parobczak K, Maślanka K. Cybersecurity protection for power grid control infrastructures. Int J Crit Infrastruct Prot 2017;18:20–33.

[25] Yang Y, McLaughlin K, Littler T, Sezer S, Pranggono B, Wang HF. Intrusion detection system for IEC 60870-5-104 based SCADA networks. In: 2013 IEEE power energy society general meeting; 2013. p. 1–5.

[26] Hong J, Liu C, Govindarasu M. Integrated anomaly detection for cyber security of the substations. IEEE Trans Smart Grid 2014;5(4):1643–53.

[27] Yang Y, McLaughlin K, Sezer S, Littler T, Im EG, Pranggono B, et al. Multiattribute SCADA-specific intrusion detection system for power networks. IEEE Trans Power Delivery 2014;29(3):1092–102.

[28] Barbosa RRR, Sadre R, Pras A. Flow whitelisting in SCADA networks. Int J Crit Infrastruct Prot 2013;6(3):150–8.

[29] Lin C-Y, Nadjm-Tehrani S. Understanding IEC-60870-5-104 traffic patterns in SCADA networks. In: Proceedings of the 4th ACM workshop on cyber-physical system security. New York, NY, USA: ACM; 2018. p. 51–60.

[30] Kleinmann A, Wool A. Automatic construction of statechart-based anomaly detection models for multi-threaded SCADA via spectral analysis. In: Proceedings of the 2nd ACM workshop on cyber-physical systems security and privacy. New York, NY, USA: ACM; 2016. p. 1–12.

[31] Caselli M, Zambon E, Kargl F. Sequence-aware intrusion detection in industrial control systems. In: Proceedings of the 1st ACM workshop on cyber-physical system security. New York, NY, USA: ACM; 2015. p. 13–24.

[32] Mercaldo F, Martinelli F, Santone A. Real-time SCADA attack detection by means of formal methods. In: 2019 IEEE 28th international conference on enabling technologies: infrastructure for collaborative enterprises (WETICE); 2019. p. 231–6.

[33] Yoo Hzunguk ST. Novel approach for detecting network anomalies for substation automation based on IEC 61850. Multimed Tools Appl 2015;74:303–18.

[34] Matoušek P. Description of IEC 61850 Communication. Technical Report. Brno University of Technology; 2018.

[35] . Industrial automation systems–Manufacturing Message Specification–Part 2: Protocol specification. Standard. Geneva: International Organization for Standardization; 2003.

[36] Matoušek P. Analysis of DLMS Protocol. Technical Report FIT-TR-2017-13. Brno University of Technology; 2017.

[37] Hinden R, Deering S. IP Version 6 Addressing Architecture. IETF RFC 4291; 2006.

[38] Matoušek P, Ryšavý O, Grégr M. Increasing visibility of IEC 104 communication in the smart grid. In: The 6th international symposium for ICS & SCADA cyber security research 2019. BCS Learning and Development Ltd; 2019. p. 21–30.

[39] Barbosa RRR, Sadre R, Pras A. A first look into SCADA network traffic. In: 2012 IEEE network operations and management symposium; 2012. p. 518–21.

[40] Barbosa RRR, Sadre R, Pras A. Difficulties in modeling SCADA traffic: a comparative analysis. In: Proceedings of the 13th international conference on passive and active measurement. Berlin, Heidelberg: Springer-Verlag; 2012. p. 126–35.

[41] Proto A, Alexandre LA, Batista ML, Oliveira IL, Cansian AM. Statistical model applied to NetFlow for network intrusion detection. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010. p. 179–91.

[42] Caberera J., Ravichandran B., Mehra R.. Statistical traffic modeling for network intrusion detection. 2000, p. 466–473. 10.1109/MASCOT.2000.876573

[43] Crotti M, Dusi M, Gringoli F, Salgarelli L. Traffic classification through simple statistical fingerprinting. SIGCOMM Comput Commun Rev 2007;37(1):5–16.

[44] Wang Y, Zhang Z, Yao DD, Qu B, Guo L. Inferring protocol state machine from network traces: a probabilistic approach. In: Lopez J, Tsudik G, editors. Applied cryptography and network security. Berlin, Heidelberg: Springer Berlin Heidelberg; 2011. p. 1–18.

[45] Krueger T, Gascon H, Krämer N, Rieck K. Learning stateful models for network honeypots. In: Proceedings of the 5th ACM workshop on security and artificial intelligence. New York, NY, USA: ACM; 2012. p. 37–48.

[46] de la Higuera C. Grammatical inference: learning automata and grammars. New York, NY, USA: Cambridge University Press; 2010.

[47] Carrasco RC, Oncina J. Learning stochastic regular grammars by means of a state merging method. In: Proceedings of the second international colloquium on grammatical inference and applications. London, UK, UK: Springer-Verlag; 1994. p. 139–52.

[48] Matoušek P, Havlena V, Holík L. Anomaly detection of ICS traffic using detereministic probabilistic automata. Technical Report FIT-TR-2020-xx. Brno University of Technology; 2020.

[49] Hulden M. Treba: efficient numerically stable EM for PFA. In: Heinz J, Higuera C, Oates T, editors. Proceedings of the eleventh international conference on grammatical inference. Proceedings of Machine Learning Research, vol. 21. University of Maryland, College Park, MD, USA: PMLR; 2012. p. 249–53.