Chromik *et al. Energy Informatics* (2018) 1:56
https://doi.org/10.1186/s42162-018-0058-7

Energy Informatics

SOFTWARE Open Access

# An integrated testbed for locally monitoring SCADA systems in smart grids

Justyna J. Chromik[1*], Anne Remke[1,2] and Boudewijn R. Haverkort[1]

*Correspondence:
j.j.chromik@utwente.nl
[1]University of Twente, Enschede, Netherlands
Full list of author information is available at the end of the article

**Abstract**

A testbed for evaluating if and how process-aware monitoring may increase the security of decentralized SCADA networks in power grids is presented. The testbed builds on the co-simulation framework *Mosaik*, and co-simulates in an integrated way, the power distribution network on different voltage levels, as well as the control network (Modbus/TCP). The existing simulators were extended to allow topology changes, and a controller (RTU) simulator connected to a SCADA server enabling remote control was implemented. Using the developed testbed, a recently proposed local monitoring approach was investigated. The results show that for so-called interlocks the proposed monitoring approach prevents the execution of 33.3% of the commands, that would result in an unsafe state of the power distribution grid. Furthermore, it is shown that unsafe transformer tap positions can also be avoided. To illustrate the relevance and importance of the proposed testbed, a detailed comparison of related work on process-aware intrusion detection approaches and testbeds combining (parts of) the control network and the power grid is provided.

**Keywords:** SCADA, Process-aware, Monitoring, Smart grid, Testbed, Mosaik, Co-simulation

## Introduction

The ongoing integration of more renewable energy resources and new technology, like energy storage systems, into smart grids requires the full integration of ICT into power transmission and distribution systems (Smart Grids in Distribution Networks 2015). To guarantee a stable power grid, many approaches propose Decentralized Energy Management (DEM), which relies on Supervisory Control and Data Acquisition (SCADA) networks to communicate sensor readings and commands between the individual components and their control server. Due to the increasing number of Distributed Energy Resources (DERs) such as Photo Voltaic (PV) panels, real-time monitoring and control is required also at medium and low voltage levels (Lu et al. 2015). While DEM is promising, recent events, such as disconnecting the Ukrainian distribution substations (ICS-CERT 2018b) through cyber attacks, have shown that also these control networks need to be improved w.r.t. their security and reliability. Moreover, reports show that breaches in the energy domain account for 20% of the reported cyber security incidents in 2016 (ICS-CERT 2016), and new hacking tools are being developed with the energy sector in mind (CRASHOVERRIDE 2017), e.g., abusing vulnerabilities of protocols used in the energy sector.

One way to improve network security is to monitor ongoing traffic and to view it in relation to the current state of the system. Clearly, when doing this for larger networks, scalability becomes a challenge. Hence, this paper evaluates a *decentralized* monitoring approach using a testbed that builds on the co-simulation framework *Mosaik*. In this approach, an additional security measure is taken by inspecting and pre-evaluating network traffic before actually executing commands in the field stations controlling the Medium and Low Voltage levels. The Bro Intrusion Detection System (IDS) (Paxson 1999) combined with the state information of the underlying physical process is used to monitor the SCADA network traffic and to determine if the commands sent through the network are legitimate, as proposed in Chromik et al. (2016a, b). Monitoring the network traffic allows for creating a thorough picture of the power distribution subsystem without interfering with the operation of it. By monitoring locally, the detection of malicious commands is performed directly at remote substations managed by the Distribution System Operators, without involving the central control room. This not only helps to keep the DEM secure, but also avoids a centralized single point-of-failure, thus improving scalability and resilience. The proposed approach is *not* intended to replace the current security mechanisms, but to complement the existing SCADA specific firewalls and IDSes.

The contribution of this paper is twofold. Firstly, the feasibility of the previously proposed monitoring approach is shown in a testbed, which has been adapted for this purpose. It integrates a newly developed simulator of the control network into the co-simulation framework *Mosaik* for the power distribution network. Secondly, a thorough comparison of the presented approach with respect to related work regarding testbeds and process-aware monitoring is provided. The comparison shows that no other approach has yet implemented a dynamic, system state-dependent set of rules in monitoring the traffic in the power distribution field stations.

Regarding the first contribution, this paper presents the integration of the simulation of the physical power distribution with a discrete-event simulation of the Remote Terminal Units (RTUs) used for control purposes. Moreover, this paper shows how the previously proposed local monitoring approach can improve the security of the distributed field stations at different voltage levels. For so-called interlocks, i.e., mutually dependent states of system elements, the proposed monitoring prevents the execution of 33.3% of all commands. Without the proposed approach in place, those commands would have resulted in an unsafe state of the power distribution. The remaining two-thirds of the commands yield a safe state of the power distribution, i.e., all the neighborhoods remain connected to the power grid. Hence, the approach allows the RTU to execute them, even though they might come from an untrusted source. In a second scenario, monitoring is used to identify commands to change the tap switch position of a transformer, which lead the system into an unsafe state. This could either lead to an alert or potentially, to discarding the packet with the malicious command.

*Related work* in the field of process-aware IDS techniques distinguishes between learning- (e.g., (Caselli et al. 2015; Hadžiosmanović et al. 2014)) and specification-based (e.g., (Lin et al. 2016; Urbina et al. 2016; Koutsandria et al. 2014; Nivethan and Papa 2016b; Bao et al. 2016; Mashima et al. 2016)) approaches. The latter then either uses static (e.g., (Nivethan and Papa 2016b)) or dynamic (e.g., (Lin et al. 2016; Urbina et al. 2016)) rules for detecting and/or preventing malicious commands. The specification-based approaches are closely related to the approach presented in this paper. However,

they can either not be used in the field stations (Lin et al. 2016), are able to detect but not prevent malicious commands (Urbina et al. 2016; Nivethan and Papa 2016b), or do not implement a dynamic policy depending on the system state (Koutsandria et al. 2014). Simulation testbeds mainly differ in the power equation solvers. PowerWorld is used, e.g., by Davis et al. (2006); Gunathilaka et al. (2016), Matlab/Simulink is used, e.g., by (Sadi et al. 2015; Koutsandria et al. 2014), and OpenDSS is used, e.g., by (Lévesque et al. 2012; Awad et al. 2016). Existing testbeds either have limited access (Davis et al. 2006; Sadi et al. 2015; Gunathilaka et al. 2016) or do not include SCADA-specific protocols (Lin et al. 2016; Lévesque et al. 2012; Sadi et al. 2015; Awad et al. 2016). Section "Comparison of the proposed system to existing approaches" presents an extensive comparison of related approaches and testbeds.

The paper is further organized as follows. Section "SCADA and monitoring" provides background on SCADA systems and monitoring of the physical process. Section "Local monitoring approach" presents the proposed local monitoring approach, and section "Implementation of the testbed" provides details on the created testbed. Then, section "Improving field stations security" shows the traffic monitoring approach and its influence on the security of field stations. Relevant related literature is discussed and compared extensively in section "Comparison of the proposed system to existing approaches". The paper is concluded in section "Conclusions" with a summary and directions for further work.

## SCADA and monitoring

First, an overview on SCADA systems is provided together with a discussion of the communication protocols used when controlling power grids. Then, section "SCADA security" highlights the vulnerabilities present in such systems.

### Overview and control

Supervisory Control And Data Acquisition (SCADA) systems are crucial for any geographically distributed physical process that needs to be monitored and controlled in a timely manner. A conceptual picture of a SCADA system is shown in Fig. 1. The most important elements are discussed in the following.

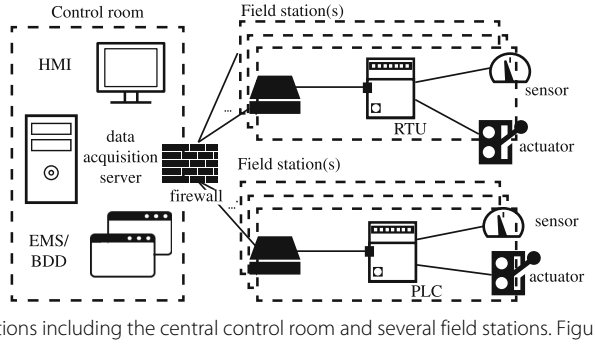


**Fig. 1** SCADA locations including the central control room and several field stations. Figure 1 illustrates a generic SCADA network. On the left the control room, combining the human machine interface (HMI), the data acquisition server and the energy management system (EMS). Separated by a firewall, these components can access the field stations, which in turn control the physical process by means of an RTU or PLC, which are equipped with sensors and actuators

The **control room** contains the **data acquisition server**, which collects the data sent from the field stations over communication channels, processes this information using models of the physical system, and displays the resulting system state on a **Human Machine Interface** (HMI). An operator is able to view the information on the HMI and, if necessary, can request changes in the system by sending commands via the HMI to the field stations. Although possible, this manual intervention does not happen often, as the SCADA system usually has some form of automated control in place. In power distribution, the so-called **Energy Management Systems** (EMS) perform crucial monitoring and correction functions, such as **State Estimation** and **Bad Data Detection**, as well as the controlling functions, such as **Load Balancing**, etc. (Liu et al. 2011; Zambon et al. 2015). The **field stations** are connected with the control room via communication channels, e.g., via GSM or Ethernet. In the field stations, the information about the process is measured using **sensors**, and this information is processed by the **Programmable Logic Controllers** (PLCs) and collected and sent to the central control room by so-called **Remote Terminal Units** (RTUs). These devices form the connection between the power grid's operators and the power grid's process. Any changes requested in the control room, such as changing the state of **actuators**, e.g., switches, which they control, have to pass through these devices.

In the past, the monitoring using SCADA systems was mainly used in transmission of the electricity operating at High Voltage. However, due to increased use of DERs such as PV panels, there is an increased need for implementing such control and monitoring also at Low and Medium Voltage (Lu et al. 2015; Ciocia et al. 2017; Bell et al. 2018).

For the SCADA elements to communicate, the devices need to use a communication protocol. In the past decades, SCADA systems were using proprietary protocols, which made it difficult to integrate with other systems. Next to that, this separation also gave a (false) sense of security, as the protocols were not publicly known. Therefore, these communication protocols were not developed with security measures in mind. Today, protocols are open and standardized in order to enable easier and efficient communication between various equipment vendors and power operators. This standardization eliminates the sense of "security by obscurity" (Nicholson et al. 2012).

One of the widely-used protocols to connect the remote RTUs with a central supervisory computer is Modbus/TCP (Khan and Mauri 2013). Although Modbus is a generally accepted industrial process standard, especially popular in the oil and gas sector, it also plays an important role in power distribution (Bush 2014; Kenner et al. 2016). It is a master/slave type of protocol, where only one of the communicating devices, called master (or "client"), can initiate the communication. The slave (or "server") continuously listens for incoming connections on TCP port 502. Modbus stores either 1 bit values (so-called coils) or 1 byte values (so-called registers). Both coils and registers can be either read-only values (discrete inputs and input registers, respectively) or read/write values (coils or holding registers, respectively). In order to allow for, e.g., floating point variables, some vendors allow for combining registers to hold 32-bit and 64-bit values (Hadžiosmanović et al. 2014). Security extensions for Modbus/TCP protocol have been proposed, e.g., (Fovino et al. 2009; Shahzad et al. 2015; Éva et al. 2018), which, however, do require changes on the protocol level of operating devices. This is expected to be difficult as companies are reluctant to such changes and global standardization. Without a uniform standard, the proposed approaches may be incompatible with existing systems. No dedicated Modbus

security standards exist, however, one could argue that IEC62351 (IEC Webstore 2018) also encompasses Modbus as it is nowadays usually runs over TCP/IP. The proposed testbed uses Modbus/TCP as it is still often used; we propose a network-monitoring approach of securing this protocol, that does not require changes on the protocol level of operating devices.

Apart from Modbus, several other protocols have been developed with power systems in mind. IEC TC57 has developed widely accepted communication standards for power distribution and transmission (Cleveland 2012), which include IEC 60870-5 used in Europe and non-US countries for communication between the SCADA control room and RTUs, DNP3, which is used, among others, in North America for communication between the SCADA control room and RTUs, or IEC 61850, used for interactions with field equipment such as protective relays and substation automation.

#### SCADA security

SCADA systems are not intrinsically secure. Even if deploying security standards, operators cannot protect field stations from malicious commands sent from the control room by, e.g., a disgruntled employee, or by accident. This type of so-called *insider attacks* constitute the majority of targeted computer attacks reported in SCADA systems (Cardenas et al. 2009; Nicholson et al. 2012). For example, in 2000 in Maroochy Shire, Australia, a disgruntled ex-employee hacked into a water control system and flooded the nearby terrains with millions of liters of sewage (Mustard 2005).

SCADA systems are also abused by *outsiders*. In so-called man-in-the-middle attacks, the attacker is able to relay all the communication exchanged between some two devices. While the messages captured by the attacker can be altered, the communicating devices are convinced they communicate directly (Maynard et al. 2014). By hijacking session, attackers are able to display a fake picture of the system state to the operator, or even reverse the semantic meaning of operator's actions, while presenting a consistent picture to the operators (Kleinmann et al. 2017). Stuxnet is a complex malware designed to change values of data sent and received by PLCs. It was most likely introduced to the target environment of Iranian's nuclear facility by an unaware insider or by a third party contractor (ICS-CERT 2010). By spreading malware within operators' networks, hackers are able to maintain connection within those networks and take control over remotely accessible devices (ICS-CERT 2018b).

## Local monitoring approach

This section first motivates the necessity of local monitoring in section "Global monitoring and remote vulnerabilities". Next, a formal description of the monitored system is given in section "Model description". Finally, the proposed local monitoring approach is described in section "Local analysis".

#### Global monitoring and remote vulnerabilities

As explained in section "SCADA and monitoring", a SCADA system is responsible for collecting data from remote field stations and delivering data to the control room, where the SCADA master server is located. As mentioned, in power transmission and distribution, applications like the EMS analyze data, estimate the state of the power system and display an overview of the entire physical system on the HMI. The EMS provides a *global* view

of the power transmission or distribution system. Based on the EMS, commands related to, e.g., load balancing, or system restoration can be sent to the field stations. Although the EMS is able to detect faulty sensors, it is susceptible to stealthy sensor attacks (Teixeira et al. 2011).

In order to manage the future smart grid in an effective, scalable and timely manner, communication with and control of the equipment located in field stations is required. This increased connectivity together with the use of third party software and protocols without security extensions poses quite a large risk to the well-operation of field stations (Oman et al. 2000). Even though the central EMS can correct (some) faulty sensor readings, the system is still at risk if, e.g., the central system is compromised and no extra security checks are performed *locally* at the field stations. Hence, *this paper proposes to additionally secure the communication involving field stations by only using local means.*

### Model description

This section introduces a formal model that allows to unambiguously describe the topology of a power distribution system. The notation previously used in Chromik et al. (2016a) to describe example topologies has now been formalized to allow general specifications. The resulting specification is independent of any programming language, simulation environment or testbed.

The formalism is used in section "Implementation of the testbed" and section "Improving field stations security" to specify the investigated scenarios and to formalize the traffic monitoring policies. Table 1 summarizes all relevant notation, where a set is represented with calligraphic uppercase letters, an element of a set is represented with a normal uppercase letter with a subscripted index, and a vector is represented in bold.

Formally, (a part of) the power distribution system is described as a tuple $\Omega = (\mathcal{P}, \mathcal{B}, \mathcal{L}, \mathcal{S}, \mathcal{M}, \mathcal{T}, \mathcal{R}, \mathcal{F})$, where $\mathcal{P} = \mathcal{P}^G \cup \mathcal{P}^L$ is a set of power generators $(\mathcal{P}^G)$ and consumers $(\mathcal{P}^L)$, $\mathcal{B}$ is a set of buses, $\mathcal{L}$ is a set of power lines, $\mathcal{S}$ is a set of switches, $\mathcal{M}$ is a set of sensors, $\mathcal{T}$ is a set of transformers, $\mathcal{R}$ is a set of protective relays, and $\mathcal{F}$ is a set of fuses.

Even though the formal model is general enough to capture a large part of the power grid, in the following, smaller models that only represent individual substations controlled by a single RTU are used. Depending on the scenario, not all elements included in $\Omega$ will be part of the local system, since, for example, not every substation contains a transformer.

#### *System elements*

**Power lines** (or branches) labelled $L_i$ for $i \in \{1, ..., |\mathcal{L}|\}$ connect power generators (also called sources) and consumers (also called loads) with each other, or with buses and transformers. They are defined as follows: $\mathcal{L} \subseteq ((\mathcal{P} \times \mathcal{B}) \cup (\mathcal{T} \times \mathcal{B}) \cup (\mathcal{B} \times \mathcal{B}) \cup (\mathcal{B} \times \mathcal{T}) \cup (\mathcal{B} \times \mathcal{P}))$. **Buses** are labelled $B_i$ for $i \in \{1, ..., |\mathcal{B}|\}$. The physical characteristics of a power line impose a maximum current on the power line, i.e., $L_i.I_{max}$. Exceeding this maximum value may damage the power line, e.g., by wearing it off much faster. The maximum current capacity is provided as a vector over all power lines using *dot-notation*: $\mathbf{L}.I_{max} = [L_1.I_{max}, L_2.I_{max}, ..., L_{|\mathcal{L}|}.I_{max}]$. The set of other characteristics of power lines and buses can be found in Table 1.

Each power line can be connected to or disconnected from the bus by a **switch**. For each switch $S_i$, where $i \in \{1, ..., |\mathcal{S}|\}$, the state of the switch is denoted as $S_i.st \in \{0, 1\}$,

**Table 1** List of the symbols of the system elements

| Element | Property | Symbol |
|---|---|---|
| Power network | Model | $\Omega$ |
| | State | $T$ |
| Power generators and consumers | Combined set | $\mathcal{P} = \mathcal{P}^G \cup \mathcal{P}^C$ |
| | Set of power generators | $\mathcal{P}^G = \left\{P^G_1, ..., P^G_{\lvert\mathcal{P}^G\rvert}\right\}$ |
| | Set of power consumers | $\mathcal{P}^C = \left\{P^C_1, ..., P^C_{\lvert\mathcal{P}^C\rvert}\right\}$ |
| | Position | $P^x_i.pos = L_j$ for $x \in \{G, C\}$ |
| | Power value | $P^x_i.pv$ for $x \in \{G, C\}$ |
| | Vector of all power values | $\mathbf{P}$ |
| Buses | Set of buses | $\mathcal{B} = \left\{B_1, ..., B_{\lvert\mathcal{B}\rvert}\right\}$ |
| | Vector of incoming lines | $\mathbf{B}_i.in = [L_j, ..., L_n]$ |
| | Vector of outgoing lines | $\mathbf{B}_i.out = [L_c, ..., L_h]$ |
| Transformers | Set of transformers | $\mathcal{T} = \left\{T_1, ..., T_{\lvert\mathcal{T}\rvert}\right\}$ |
| | Transformer rate | $T_i.r$ |
| | Tap switch position | $T_i.p$ |
| | Vector of all tap positions | $\mathbf{T}$ |
| Power lines | Set of power lines | $\mathcal{L} = \left\{L_1, ..., L_{\lvert\mathcal{L}\rvert}\right\}$ |
| | Position | $L_i.pos = (Bk, B_n)$ |
| | Maximum current | $L_i.I_{max}$ |
| | Reference voltage | $L_i.V_{ref}$ |
| | Meter (side of $B_k$) | $L_i.B_k.M = M_d$ |
| | Vector of meters on line $L_i$ | $\mathbf{L}_i.M = [M_d, ..., M_h]$ |
| | Switch (side of $B_k$) | $L_i.B_k.S = S_e$ |
| | Vector of switches on line $L_i$ | $\mathbf{L}_i.S = [S_e, ..., S_o]$ |
| | Fuse (side of $B_k$) | $L_i.B_k.F = F_u$ |
| | Vector of fuses on line $L_i$ | $\mathbf{L}_i.F = \left[F_u, ..., F_y\right]$ |
| Meters | Set of meters | $\mathcal{M} = \left\{M_1, ..., M_{\lvert\mathcal{M}\rvert}\right\}$ |
| | Position | $M_i.pos = L_i.B_n$ |
| | Measured current | $M_i.I$ |
| | Measured voltage | $M_i.V$ |
| | Vector of states of all readings | $\mathbf{M}$ |
| Switches | Set of switches | $\mathcal{S} = \left\{S_1, ..., S_{\lvert\mathcal{S}\rvert}\right\}$ |
| | Position | $S_i.pos = L_i.B_n$ |
| | State of the switch | $S_i.st$ |
| | Vector of states of all the switches | $\mathbf{S}$ |
| Fuses | Set of fuses | $\mathcal{F} = \left\{F_1, ..., F_{\lvert\mathcal{F}\rvert}\right\}$ |
| | Position | $F_i.pos = L_i.B_n$ |
| | State of the fuse | $F_i.st$ |
| | Vector of states of all fuses | $\mathbf{F}$ |
| Protective relays (circuit breakers) | Set of protective relays | $\mathcal{R} = \left\{R_1, ..., R_{\lvert\mathcal{R}\rvert}\right\}$ |
| | Position (on a switch) | $R_i.S = S_j$ |
| | Cutting current | $R_i.I_{max}$ |

representing an open (disconnected) and a closed (connected) switch, respectively. The vector $\mathbf{S}$ collects the states of all the switches and is of size $|\mathcal{S}|$. The summary of the properties of the switches can be found in Table 1.

Next to the switches each power line has **meters** $M$ (sensors) within the substation where the bus is located. The sensor $M_i$ measures usually at least the current in the line $M_i.I$, and the voltage between the line and the ground $M_i.V$. The readings from a sensor

are written as a pair of current and voltage: $(M_i.I, M_i.V)$. The vector $\mathbf{M}$ collects all the sensors' readings and is of size $|\mathcal{M}|$. The properties of the meters can be found in Table 1.

A simpler version of a switch is a **fuse**, which melts when an overcurrent occurs. It is not possible to turn the fuse back on, it can only be replaced. The fuse is denoted as $F_i$, where $i \in \{1, ..., |\mathcal{F}|\}$ and the state of the fuse is either one or zero, i.e., $F_i.st \in \{0, 1\}$. Vector $\mathbf{F}$ collects the states of all the fuses and is of size $|\mathcal{F}|$. Again, the properties of the fuses are summarized Table 1.

**Protective relays** are mechanical or digital controllers, which control a connected switch. In case the current measured on the line exceeds some pre-defined value $I_{max}$, the switch will be opened, disconnecting the line with over-current. They are denoted as $R_i$ for $i \in \{1, ..., |\mathcal{R}|\}$, and are assigned to a switch, i.e., for relay $i$, which is positioned at switch $j$, $R_i.S = S_j$. The properties of protective relays are available in Table 1.

**Transformers** connect parts of the power system that operate at different voltage levels. A transformer $T_i$ for $i \in \{1, ..., |\mathcal{T}|\}$ has the following properties: transformation rate $T_i.r$, which defines the voltage ratio (e.g., the ratio 1000:1 transforms voltage from 400 kV to 400 V), and the transformer tap position $T_i.p$. The position of the tap switch of a Medium to Low Voltage transformer has to be chosen such that the secondary voltage, that is delivered to the customers, equals 230 V. The measurements are not taken directly on the windings of the transformer, but on the incoming and outgoing lines, which results in an accurate approximation. All properties of the transformers are listed in Table 1.

#### *System state*

The so-called state in the system refers to all the actual values which can change in the system over time. The system state can be described by five vectors indicating: (i) the states of the switches, (ii) the state of the fuses, (iii) the sensor readings, (iv) the power consumption and production, and (v) the position of the transformer taps.

- Vector $\mathbf{S} = \left[S_1.st, S_2.st, ..., S_{|\mathcal{S}|}.st\right]$ of size $|\mathcal{S}|$ denotes the state of all switches in the system.
- Vector $\mathbf{F} = \left[F_1.st, F_2.st, ..., F_{|\mathcal{F}|}.st\right]$ is of size $|\mathcal{F}|$ and summarizes the states of all fuses present in the system.
- The readings from one sensor can be written as a pair of the measured current and voltage: $(L_i.M.I, L_i.M.V)$. Vector $\mathbf{M}$ collects those pairs for all sensors: $\mathbf{M} = \left[(L_1.M.I, L_1.M.V), ..., \left(L_{|M|}.M.I, L_{|M|}.M.V\right)\right]$, and is of size $|\mathcal{M}|$.
- Vector $\mathbf{P} = \left[P^G_1.pv, ...P^G_{|\mathcal{P}^G|}.pv, P^C_1.pv, ..., P^C_{|\mathcal{P}^C|}.pv\right]$ for $|\mathcal{P}^G|$ sources and $|\mathcal{P}^C|$ consumers, denotes the loads and sources of power.
- Finally, the set of positions of the transformer tap is denoted as vector $\mathbf{T} = \left[T_1.p, T_2.p, ..., T_{|\mathcal{T}|}.p\right]$ of size $|\mathcal{T}|$.

Now, the system state $T$ can be written as a tuple that consists of the above five vectors: $T = (\mathbf{S}, \mathbf{F}, \mathbf{M}, \mathbf{P}, \mathbf{T})$ and can be used in the following to determine whether the system state is consistent and safe, to be explained in the section "Local analysis".

#### *Events*

The system state can change upon receiving any new information, e.g., information from the sensors with different voltage readings result in an updated state. Different power values of the sources or loads also update the state. Moreover, a command to open

or close any of the switches, or changing the tap switch position brings the system to another state. For constant power sources and loads, for now, only two types of events are considered: (i) *readings*, and (ii) *commands*. Readings update the state to a new state $T' = (\mathbf{S}', \mathbf{F}', \mathbf{M}', \mathbf{P}', \mathbf{T}')$, whereas a command will result in a new state $T'$ with an updated vector $\mathbf{S}'$, collecting the states of the switches, or/and new vector of transformer states $\mathbf{T}'$.

### Local analysis

The previously presented ideas (Chromik et al. 2016a; b) propose to extend the existing monitoring systems for power distribution and perform additional monitoring in the field stations. This is achieved by (i) monitoring the traffic exchanged between the field station and the control room, in order to maintain the current state of the physical process at the field station, and (ii) based on the obtained commands from the control room, predict the command outcome for this subsystem.

In order to determine whether the sensor readings comply to the laws of physics, the readings are compared to a set of physical constraints, as listed in Table 2.

To determine whether the state of the physical system is *safe*, the readings are checked against the set of safety requirements, as listed in Table 3. Note that the physical constraints in Table 2 and the safety requirements in Table 3 are examples of possible rules that can be analyzed and they depend on the investigated system.

The monitoring process located at field stations analyses the content of the incoming and outgoing packets. The flow chart in Fig. 2 illustrates the procedure as performed by the local monitoring algorithm. The left part of Fig. 2 illustrates the actions taken when receiving new sensor readings. New readings mean that a new system state To' has been reached, which could be unsafe and/or inconsistent. Therefore, two checks need to be performed: (i) the safety check, which compares To' to the restrictions listed in Table 3, and (ii) the consistency check, according to the physical constraints listed in Table 2. If the system state is consistent and safe, the new system state is stored by the monitoring tool. Otherwise an alert is generated, and the state To' is stored as To.

The right part of Fig. 2 shows the actions triggered when a new command is received. Such a new command is first "executed" in the model - based on the previously stored knowledge of the current state Tc. If the predicted new state Tc' is safe, the command can be executed on the actual system, and Tc' can be stored as the current state Tc. Otherwise, if the predicted state is unsafe, an alert is sent to the operator and the command is discarded or at least delayed until explicitly approved by the operator via a secure channel.

**Table 2** Physical consistency constraints

| Physical consistency constraint | Explanation |
|---|---|
| $\forall B_i \in \mathcal{B}\left(\sum_{L_j \in B_i.in} L_j.B_i.M.I = \sum_{L_k \in B_i.out} L_k.B_i.M.I\right)$ | Kirchoff's current law |
| $\forall L_i \in \mathcal{L}\left(\forall S_j \in L_i.S\left(\left(S_j.st = 0\right) \Rightarrow \left(\forall M_k \in L_i.M\left((M_k.I = 0) \wedge (M_k.V = 0)\right)\right)\right)\right)$ | If all switches on a line are open, the values of current and voltage on this line have to be zero |
| $\forall T_i \in \mathcal{T}, M_x, M_y : M_x = L_x.T_i.M \wedge M_y = L_y.T_i.M$ $\left(T_i.r = M_x.V/M_y.V = M_y.I/M_x.I\right)$ for $M_x.V > M_y.V$ | Assuming no losses, the transformer changes the voltage and current value with its predefined ratio $r$ |
| $P = V \cdot I$, e.g., $P_1^G.pv = P_1^G.pos.P_1^G.M.I \cdot P_1^G.pos.P_1^G.M.V$ | Electric power is equal to voltage times current |

**Table 3** Safety requirements

| Safety requirement | Explanation |
|---|---|
| $\forall L_i \in \mathcal{L}\ \forall M_h \in L_i.M$ $M_h.V \in [0.9L_i.V_{ref}; 1.1L_i.V_{ref}]$ | Voltage on all lines stays between the boundaries of the reference voltage value ±10% |
| $\forall L_i \in \mathcal{L}\ \forall M \in L_i.M\ M_h.I \leq L_i.I_{max}$ | Current in a power line does not exceed its maximum allowed current |
| $\sum_{\mathcal{P}^G} P_i^G + \sum_{\mathcal{P}^C} P_j^C = 0$ | Power produced by the sources equals the power consumed by the loads |
| Interlocks (defined based on topology), e.g., $S_i.st \mid\mid S_j.st = 1$ | Some switches cannot be opened simultaneously for safety reasons |

The lower cycle in Fig. 2 compares the current state of the system, as seen by the operator (To), to the previously calculated system state (Tc). If these two states are not the same (within an error margin $\epsilon$), this has to be reported to the operator, since it indicates a potentially dangerous situation. The proposed algorithm cannot provide a meaningful prediction when working with imprecise or even incorrect data. Therefore, the operator will be notified about any such inconsistency until the situation is resolved, e.g., by replacing a faulty sensor.

## Implementation of the testbed

Research on critical infrastructures requires either a dedicated physical testbed or a simulation testbed. Since the former is often expensive, not very flexible or hard to access, the goal of this paper was to develop a flexible and accessible simulation testbed. From the available simulation testbeds, described in detail in section "Comparison of the proposed system to existing approaches", the co-simulation framework *Mosaik* seemed most flexible. Through including several specifically developed simulators, *Mosaik* was extended with communication network capabilities. This section explains

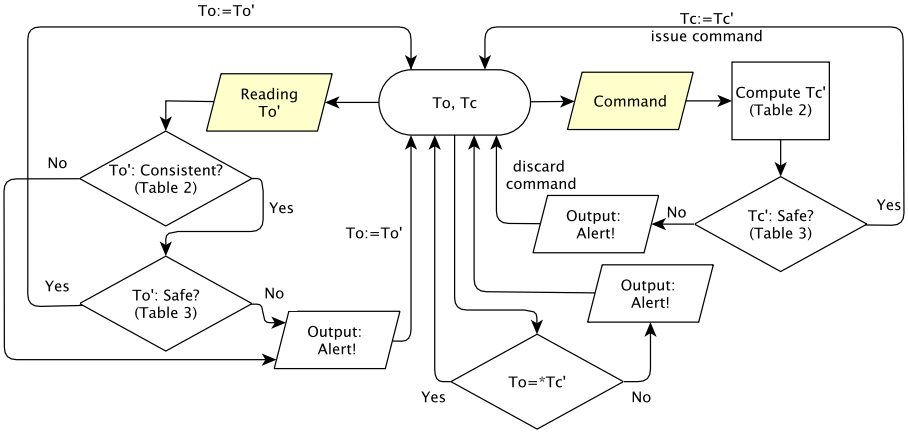


**Fig. 2** Flow chart representing the local monitoring algorithm. Input events are highlighted in yellow. Figure 2 presents the steps the local monitoring algorithm takes to decide whether a command should be executed or whether a sensor reading is consistent. It consists of two main loops: one is triggered by the input event of a command and the other by the input event of a sensor reading. Event checks are triggered upon the occurrence of the respective input, as described in Tables 2 and 3. Depending on the outcomes of those checks, different paths are taken in the flow chart. If a reading is consistent and safe, the internal state of the model is updated. Otherwise an alert is additionally triggered. For commands, if the precomputed state is unsafe, the command is discarded and an alert is issued. Only if the checks yield that the command can safely be executed, it is issued to the respective actuators

the elements of the proposed testbed: the *Mosaik* framework is discussed in section "Mosaik co-simulation framework", the power system simulator is addressed in section "Power distribution system description in Mosaik", the control network is explained in section "SCADA system", and the overall monitoring approach is discussed in section "Traffic monitor".

### Mosaik co-simulation framework

*Mosaik* is an open source co-simulation framework written in Python (under GNU LGPL) (OFFIS 2017), using a discrete-event simulation library based on SimPy. With the provided API, different existing simulators can be connected, while *Mosaik* interfaces their data transfer and tracks the execution order.

Figure 3 illustrates the general scheme of the proposed testbed, with *Mosaik* presented as a box marked with Number 1. The black elements above the horizontal dashed line indicate the *physical* elements of the testbed. They are simulated here, but they refer to the physical parts of the power distribution. The values provided by this part are considered the "ground truth", i.e., if a sensor value on the cyber side will deviate from the one on the physical side, then the one on the physical side is considered true. The most significant parts co-simulated in *Mosaik* are: a household and a PV panel profile simulator (Number 2), which are available in the *Mosaik* example scenario[1]; a power distribution simulator (Number 3), and the RTU simulator (Number 4), enabling communication with the (cyber) Modbus RTU device.

The power distribution simulator solves the power flow equations using the PyPower package (PYPOWER 2018) implementing the Newton-Raphson AC power flow method,

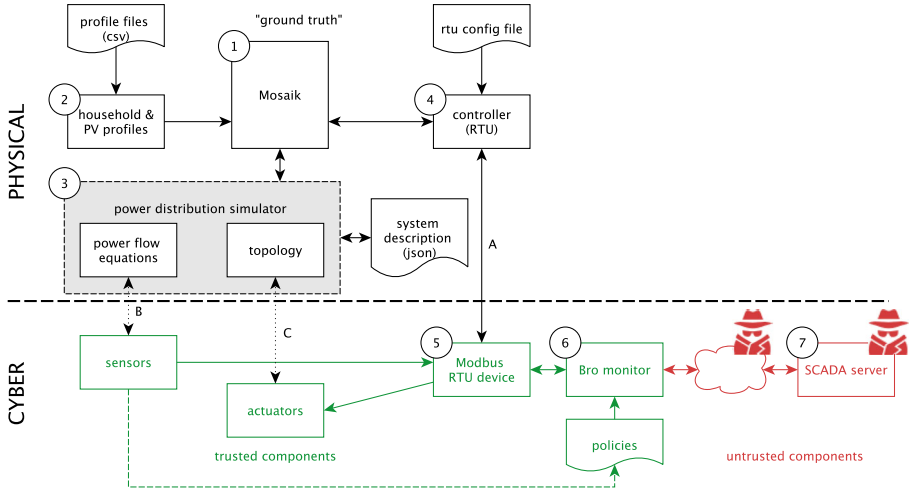


**Fig. 3** Scheme of the testbed: the simulated part and the network components are shown. Figure 3 outlines the testbed. The simulated physical components are used as "ground truth" and depicted above the dashed line. The network components, including the source of malicious commands, are illustrated below the dashed line. The trusted parts are colored green and the untrusted parts red. The physical and the cyber parts are connected via a single physical connection (denoted A). The correspondence between the sensors and the actuators in the cyber part, to their values output by the power distribution simulator (power flow equations and topology), are indicated by dashed arrows, labeled B and C respectively. Traffic generated by the hacker reaches the Modbus RTU device only via the Bro monitor. The monitor then applies the safety and consistency checks before the command or sensor reading is put forward to the simulated physical part and included in the power distribution simulator

which has been adapted to allow for topology changes. The proposed extensions and adjustments are described in detail in the following sections.

Below the horizontal dashed line in Fig. 3, the *cyber* elements of the testbed are presented: the control network, which consists mainly of a Modbus/TCP (The Modbus Organization 2012) RTU device (Number 5), the monitoring device (Number 6), and the SCADA server (Number 7).

The integration of the RTU device into the physical system is enabled by making the following connections, as indicated in Fig. 3 by black vertical lines: the controller (RTU) API invokes a thread which creates a simulation of the Modbus RTU device (Connection A). This connection is the *actual* link between the *cyber* and *physical* part of the testbed, therefore in Fig. 3 it is indicated with a solid line. It allows for the following relations: based on the values obtained from the power flow equation solver via the *Mosaik* interface, the Modbus RTU device determines the sensor measurements and forwards them to the control network (Correspondence B, marked with a dashed line); upon a command received from a SCADA server in the Modbus RTU device, this device applies the changes on the actuators in the testbed by changing the topology in the power distribution simulator (Correspondence C, marked with a dashed line).

With the physical and cyber system co-simulated within the *Mosaik* framework, it is possible to include all elements necessary to describe the system $\Omega$ as explained in section "Model description". The power buses, branches, transformers are described within the PyPower simulator, meters and switches are described within the controller simulator, power sources and loads are taken from the household and PV panel simulators, or represented as the reference bus.

Due to the interaction of several simulators, commands that are issued within the network simulation part of *Mosaik* first need to be handled by the simulated controller, before they are propagated to the power distribution system. This corresponds to a delay of two steps in the simulation framework, which does not occur in real systems, as commands that have been processed by the controller directly impact the distribution system. Hence, it is important to choose small step sizes for the simulators that directly change the system state and avoid local control loops between simulators. The step-size for all the simulators has been set to 60 s, except for the household and PV panels profile simulators, which have a time step of 15 min. Together with the *Mosaik* co-simulation real-time factor of 120, this results in a simulation duration of around 720 s (12 min) when simulating 24 h.

#### Power distribution system description in Mosaik

The power distribution system description is based on the previously discussed *Mosaik* example scenario which consists of houses, PV panels and a distribution network built from buses, branches and transformers. The simulator for houses and PV panels, cf. Number 2 in Fig. 3, uses historic consumption profiles, with samples collected every 15 min and stored in the form of CSV files. The power distribution system simulator (cf. Number 3 in Fig. 3) solves the power flow equations using the Newton-Raphson power solving method and processes the topology changes. It uses a system description stored in a human-readable JSON file. The description formalism includes buses, i.e., a reference bus, PQ buses, and isolated buses, branches (or: power lines) and transformers, which are a special kind of branch connecting the medium and low voltage buses. An example of a branch description is shown in Table 4. As can be seen, a power line is defined by its ID

**Table 4** Example of a branch description

| Name | From | To | Length [km] | $R'\left[\frac{\Omega}{km}\right]$ | $X'\left[\frac{\Omega}{km}\right]$ | $C'\left[\frac{nF}{km}\right]$ | $I_{max}$ [A] | *Online* |
|---|---|---|---|---|---|---|---|---|
| $L_{13}$ | $B_9$ | $B_4$ | 0.35 | 0.2542 | 0.080425 | 0.0 | 240.0 | True |

(name), the IDs of the buses it connects (*from bus* and *to bus*) and its physical properties such as its length, resistance, reactance, capacitance and maximum allowed current. The description of power lines is expanded to include their state: *online* (all switches on the power line are closed) or *offline* (at least one of the switches on the branch is opened).

The power distribution system simulator was extended to take into account changes in the topology as follows. The initial PyPower simulator is enhanced with topology functions, which identify isolated buses based on information about the state of switches on the branches. This information is obtained from the controller and is then adjusted in the power distribution (topology) model, which in turn is stored in the JSON file. This new model is then forwarded to the power flow equation simulator.

An example of the description of the power grid is explained below. The power system used in the following to validate the monitoring approach is based on the topology of a small Dutch town and is shown in Fig. 4. Figure 4a shows the power system model in *Mosaik*, with the bus $B_5$ marked with a red circle, and the nodes corresponding to the parts of the transformer are marked with a green oval. These nodes are highlighted, as they will be further used for the analyses. Figure 4b shows bus $B_5$ in more detail, where the rest of the grid is abstracted to a load and a generator.

### SCADA system

In the presented scenario, the Modbus/TCP SCADA system consists of one RTU located in the field station and one SCADA server located in the control room, cf. Numbers 5 and 7 in Fig. 3. The RTU and SCADA server communicate over an untrusted network. Note that the central SCADA server is assumed to be an untrusted component as well, because of the possibility of the presence of insider attacks. The RTU reads the measurements from the sensors on power lines directly connected within the substation

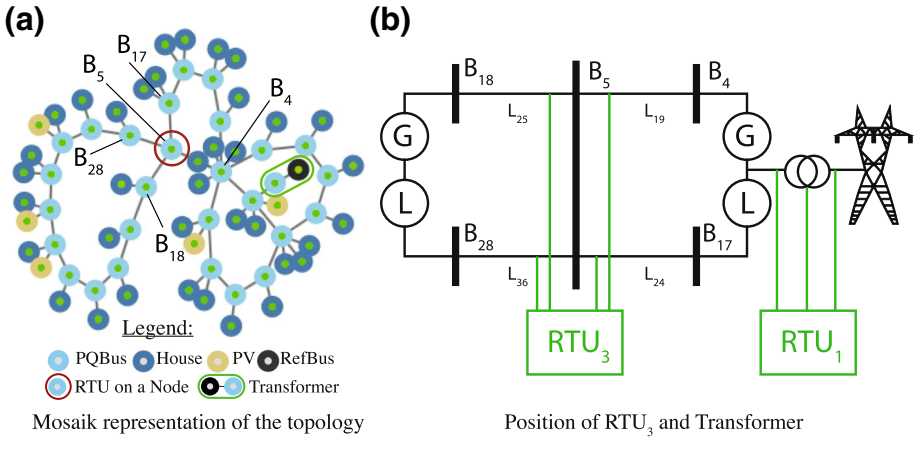


**Fig. 4** Power distribution system under analysis in Mosaik notation and simplified as one-line diagram. Figure 4 illustrates the topology simulated in Mosaik for the first Scenario. Figure 4a uses the Mosaik notation, where elements are denoted as dots in different colors and where power lines connecting different elements are denoted with lines. Figure 4b emphasizes the part of the simulated scenario, which is analyzed in this paper. $RTU_3$ controls bus B_5, which is connected via power lines to four other buses. $RTU_1$ controls a transformer connecting the High and Medium Voltage levels

on bus $B_5$, and it controls a set of actuators (switches) connecting power lines attached to that bus, cf. Fig. 4b. In the proposed testbed, the *Mosaik* controller (RTU) simulator creates a Modbus RTU device, which is a Modbus server listening on TCP port 10502 on the host machine. It uses the PyModbus library[2] to implement the Modbus/TCP protocol (The Modbus Organization 2012). SCADA server is a Modbus/TCP client created in a Virtual Machine.

The RTU controlling the bus $B_5$ stores the values of the state of the switches as coils and the rest of the values (voltage, current) as holding registers. Once a command to change the switch state arrives from the SCADA server, this change is saved on the proper coil within the simulated RTU. The *Mosaik* controller (RTU), upon every simulator step, checks whether the coil value of the RTU device has changed as compared to the stored value. If it has, this triggers the RTU to send the information about the commands to the power distribution simulator. This is the simulator event represented in Fig. 5 as the purple triangle, which further issues the following simulator events.

As an example, consider executing a command in the proposed testbed for bus $B_5$, as presented in Fig. 4b. The command is sent from the SCADA server to $RTU_3$ to open the switch located at power line $L_{25}$. A detailed analysis is shown in Fig. 5. The upper graph shows simulator events occurring in the controller and the power distribution simulators. The lower graph shows the influence of the command on the current readings at $RTU_3$. For clarity, constant values of house consumption and PV panel production are used. The time given on the $x$-axis refers to the simulation time, which is running with the real-time factor of 120 (i.e., 120 times faster). At the beginning, the current reading of power line $L_{19}$ (orange line) equals 0.153 A, the current of power line $L_{25}$ (green) equals 0.078 A, the current of power line $L_{36}$ (red) equals 0.067 A, and the current of power line $L_{24}$ (dark blue) equals 0.007 A. The simulator events (upper) graph shows recurring simulator event of recalculating power flow equations (green crosses X). At a time point just after 12.3 s, the power flow equations are recalculated. Soon after this, the controller simulator receives a command (purple triangle) which has to be passed to power distribution simulator, because the values of the switch state(s) changed. This information is sent to the power

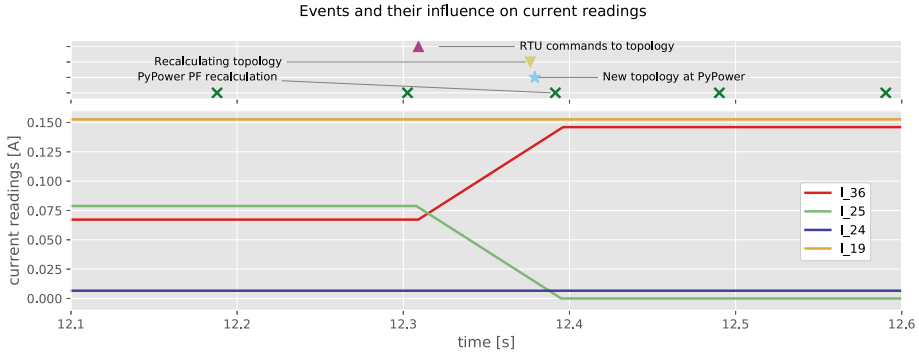


**Fig. 5** Illustration of the effect on current when executing a single event in the testbed. Figure 5 depicts the current (in Amperes) for different simulation time points (in seconds). The current on line 36 is depicted in red, the current on line 25 in green, the current on line 24 in blue and the current on line 19 in yellow. Furthermore, the figure illustrates the delay between different events in the simulation testbed, as indicated in the top part of the figure. The power flow equations are recomputed 5 times (indicated by green crosses). Furthermore, when a command arrives at the the RTU (indicated by a red triangle), it triggers a recalculation of the topology (indicated in yellow), which then leads to a new topology in PyPower after a short delay

distribution simulator and at the next step of that simulator, the topology is recalculated (yellow triangle) and the power flow equations are recalculated using PyPower again. This last event has direct influence on the readings of the current seen in the graph below. Since power line $L_{25}$ is now opened, the current value on that line decreases to zero. To compensate for that, the current on power line $L_{36}$ increased to 0.145 A.

Note that the delay between receiving a command to change the tap switch position and its influence on the voltage value is influenced by the inter-dependencies of the various simulators, as previously shown for the currents in the interlock scenario.

#### Traffic monitor

Among the available open-source network monitoring tools which are used for SCADA protocols, the most popular are Snort (Roesch 1999) and Bro (Paxson 1999; Lin et al. 2016; Udd et al. 2016). While Snort allows for pattern matching within packets to determine their legitimacy, Bro provides various frameworks, which allow rule-based evaluation of packet content, as explained below.

Bro includes a Modbus/TCP parser, that generates events upon parsing packets of this protocol. The parser, for example, generates a *modbus_write_single_coil_request* event when parsing a Modbus/TCP packet containing a "write single coil request". By creating a custom event handler, new policies that use the semantic information extracted from the parsed packet(s) can be instantiated in order to determine proper actions and alerts. By including this traffic monitoring, instead of directly storing the new value of a command from the SCADA server in the respective coil, as explained in section "SCADA system", this command is first checked against a corresponding Bro policy. In the proposed testbed, the monitoring device is placed between the Modbus RTU device and the rest of the network; in Fig. 3, Bro is indicated with Number 6.

To enable process-aware policies in Bro, among others, the requirements and restrictions from Tables 2 and 3 are used in combination with local measurements. First, the system at hand (shown in Fig. 4b) has to be described using these rules. Then, this description is used to produce relevant Bro policies. This is explained in detail in the section "Improving field stations security".

Monitoring maintains an overview of the system state at all times and compares the observed values to a pre-defined set of rules.

The local monitoring algorithm as explained in section "Local analysis" is implemented for both readings and commands:

(i) Upon a *new reading*, the Bro policy tests whether the safety requirements hold and whether physical consistency is maintained, as indicated in Tables 2 and 3. In case no violations are detected, the observed values are stored in the local model of the physical system. If violations are detected, an alert is additionally sent to the operator.
(ii) Upon receiving a *new command*, the Bro policy precomputes the outcome of executing such a command based on the constraints in Table 2, and performs safety checks according to Table 3.

### Improving field stations security

This section describes how monitoring the safety of the state of the physical system can improve field station security. Section "Threat model and attack scenario" discusses the threat model and attack scenarios. Section "Interlocks" applies monitoring to identify

attacks on the system's interlocks, and section "Transformer tap switch" applies them to a transformer tap switch. Then, section "Advantages of monitoring in a simulation testbed" lists the advantages of using the proposed testbed.

#### Threat model and attack scenario

In the following, an attacker can either perform a *man-in-the-middle attack* (cf. section "SCADA security") and inject false messages between the Modbus RTU device and the SCADA server, or can directly take control over the SCADA server, as illustrated in Fig. 3. Both attacks result in a corrupted communication channel to the field station. Hence, both the network and the SCADA server cannot be trusted. Assume that an adversary sends well-formatted packets from the control room to the remote stations and has all necessary privileges to perform the requested commands. This means that other security mechanisms, such as standard Network IDS would not recognize such packets as potentially malicious.

In the initial attack scenario an attacker attempts to disconnect power lines controlled by the $RTU_3$ (cf. Fig. 4b), one by one. That RTU initially does not perform any of the safety checks as defined in Table 3, i.e., it directly executes the received command. Then the attack scenario is changed, such that the attacker attempts to change the tap switch controlled by $RTU_1$ (cf. Fig. 4b) to an unsafe position.

#### Interlocks

Interlocks are used to manage mutually dependent elements. This logic is supposed to work locally and independently from the central control room. However, distribution operators were concerned, that for some solutions, checks are not performed locally, but only in the central control room. This means, that it is possible to bypass interlocks by injecting a command via an outside communication channel, which is not analyzed by the central EMS. Consider the interlocks that are required for the system from Fig. 4b, where bus $B_5$ is a node operating at medium voltage. When disconnecting either the two power lines $L_{19}$ and $L_{24}$, or the two power lines $L_{25}$ and $L_{36}$, the neighborhood behind bus $B_5$ is left without electricity. Hence, there are two groups of interlocks, where at least one switch has to be connected (closed).

##### *Implementation of the interlocks*

The interlocks are configured in a Bro policy as follows. First, the state of the switches is stored in a global policy table, as shown in Listing 1. This is the vector mentioned in section "Local monitoring approach" and it is part of state T, as indicated in Fig. 2. These values will be updated each time a read command is parsed by Bro.

**Listing 1** Table with status of the switches.

```
global S: table[string] of bool = {
["S_19.st"] = True,
["S_24.st"] = True,
["S_25.st"] = True,
["S_36.st"] = True
};
```

Secondly, the sets of interlocked switches have to be determined, that is, the sets of switches which should not be disconnected simultaneously. This corresponds to the last safety requirement from Table 3. This description is added to the Bro policy that will be configured in RTU on bus $B_5$, as shown in Listing 2.

**Listing 2** Table with sets of interlocks.

```
global interlock: table[count] of set[string] = {
[1] = set("S_19.st", "S_24.st"),
[2] = set("S_25.st", "S_36.st")
};
```

Thirdly, updating the switch states upon receiving a new read command has to be implemented. Since the switch states are stored on the RTU as Modbus coil values, the event handlers for the read coil request and response events are created, as shown in Listing 3. Line 2 stores the address and number of requested coils in a temporary table *temp*, identified by a string with the connection identifier and transaction identifier. Line 5 checks whether a connection with the defined connection and transaction identifiers is stored in the temporary table. If such a connection is present, the value of the switch is stored in Line 6, and in Line 7 the element from the temporary table is deleted.

**Listing 3** Event handlers for Modbus read request and response.

```
1: event modbus_read_coils_request(c: connection, headers: ModbusHeaders,
   start_address: count, quantity: count) {
2: temp[fmt("%s-%s", c$id, headers$tid)] = vector(start_address, quantity);
3: }
4: event modbus_read_coils_response(c: connection, headers: ModbusHeaders, coils:
   ModbusCoils) {
5: if ( fmt("%s-%s", c$id, headers$tid) in temp ) then
6:     S[switches_address[temp[fmt("%s-%s", c$id, headers$tid)][0]]] = coils[0];
7:     delete temp[fmt("%s-%s", c$id, headers$tid)];
8: end if
9: }
```

Finally, the safety requirements checked upon receiving a new command are implemented, according to Listing 4. Upon a write coil request and response, similar handlers as shown in Listing 3 are created. Additionally, the function shown in Listing 4 tests whether the outcome of the command does still satisfy the interlock constraints. Line 4 checks whether the switch that is supposed to be opened is part of any of the interlock sets. If so, the number of closed switches in that set is counted and if this number is at least 2, the switch can be opened.

**Listing 4** Function testing the interlocks.

```
1:  function CHECK_INTERLOCKS((address: count, value: bool): bool)
2:      local amt:count = 0;
3:      for i in interlock do
4:          if switches_address[address] in interlock[i] then
5:              for sw in interlock[i] do
6:                  if S[sw] == True then
7:                      ++amt;
8:                  end if
9:              end for
10:         end if
11:     end for
12:     if ( then amt ≥2 )          ▷ # At least 2 lines are connected to allow this action
13:         return True;
14:     else
15:         return False;
16:     end if
17: end function
```

#### *Example attack without local monitoring*

In the example shown in Fig. 4b, a successful attack is performed by disconnecting a pair of lines: either $L_{19}$ and $L_{24}$, or $L_{25}$ and $L_{36}$. An example of the effect of such a successful attack on $RTU_3$ is shown in Fig. 6. In this attack, the SCADA server sends three commands to open switches on power lines $L_{25}$, $L_{19}$ and $L_{24}$, respectively. Similar to Fig. 5, the upper graph shows events in the co-simulation framework, and the lower graph shows the effect of those events on the current readings in the power lines that are directly connected to bus $B_5$. Again, the profiles of power demand in houses and production of PV panels are set as constant for the sake of better visibility, and the time on the $x$-axis refers to the simulation time.

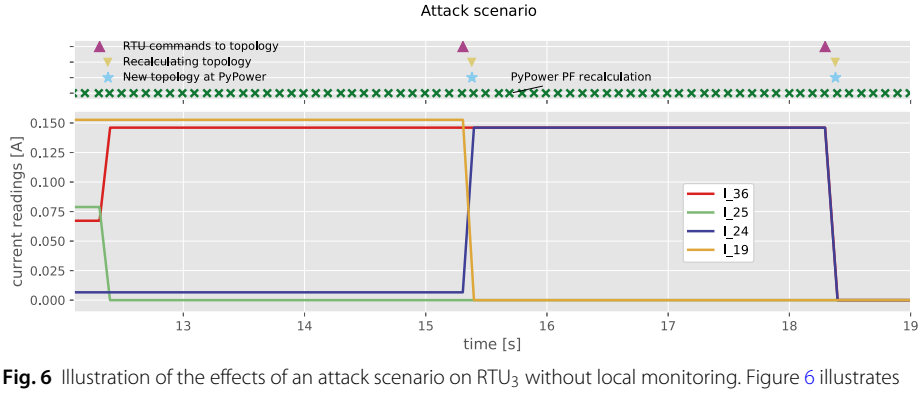


**Fig. 6** Illustration of the effects of an attack scenario on $RTU_3$ without local monitoring. Figure 6 illustrates the effects of an attack on the RTU in the simulation testbed if no local monitoring is applied. The current on the lines is shown in the same colors as in Fig. 5, however, different events are observed. Three commands to change the topology are issued, which each trigger a recalculation of the topology and a renewed analysis of the (new) power flow equations in PyPower. Once all the commands are executed, the resulting disconnection of certain power lines causes isolation of a part of the neighborhood

In Fig. 6 the current reading of the current in power line $L_{19}$ is shown in orange, $L_{24}$ in dark blue, $L_{25}$ in green and $L_{36}$ in red. Initially, the current readings on the lines have a constant value. After the first event, i.e., opening power line $L_{25}$, the current which was carried by line $L_{25}$ is then taken by power line $L_{36}$. After opening the switch on power line $L_{19}$, the bus $B_{28}$ and the rest of the neighborhood is now only connected via lines $L_{36}$ and $L_{24}$. The current on lines $L_{36}$ and $L_{24}$ is therefore equal (in the Fig. 6, the dark blue line (for $L_{24}$) overwrites the red line (for $L_{36}$)). Finally, opening power line $L_{24}$ causes isolation of part of the neighborhood and all the power lines around $RTU_3$ have zero current (orange overwrites green).

Although disconnecting power lines $L_{25}$ and $L_{19}$ influences the power flow in the distribution system, it does not disrupt the operation of the distribution system, as all the houses can still be connected to a source of power.

#### *Results*

In the following, the influence of the proposed local monitoring approach on the security of the field stations for *all* possible initial settings is investigated. The left part of Table 5 shows all possible initial (safe) values of vector **S** describing the state of the switches in the subsystem controlled by $RTU_3$. In this context, *safe* means that all houses are still connected to the source of electricity.

The right side of the table, under column "command", shows all possible commands that can be sent to $RTU_3$. These commands could be sent from the control room either by the operator or by an attacker. The outcome of each of the 4 commands for each of the nine safe initial states is tested and the output of the detection mechanism is presented. Mark '–' means that the system does not execute a requested command, as the current state of the switches already matches the requested one. Mark 'safe' indicates that the command is safe to perform and allowed. Mark 'alert!' means that the command is not safe to perform, an alert is raised and the command is discarded.

Out of a total of 36 cases, 12 cases are marked with "–", as the execution of the command would not change the state of the system. An operator should still be notified about such an incident, since the command could have been sent by an attacker who is unaware of the current state of the system and performs an attack in a opportunistic or random way. Another 12 cases are marked as safe. This means, that after performing the attack, the resulting vector **S** indicating the switch states is also one of the 9 listed safe vectors. This possible type of attack (if sent by an attacker) is unnoticed, but also does not harm the system. The remaining 12 cases were marked as attack. Here it is clear that the resulting

**Table 5** Safe values of vector **S**

| Safe **S** | | | | Command | | | |
|---|---|---|---|---|---|---|---|
| $L_{19}$ | $L_{24}$ | $L_{25}$ | $L_{36}$ | $S_{19}.st = 0$ | $S_{24}.st = 0$ | $S_{25}.st = 0$ | $S_{36}.st = 0$ |
| 1 | 1 | 1 | 1 | Safe | Safe | Safe | Safe |
| 0 | 1 | 1 | 1 | – | Alert! | Safe | Safe |
| 1 | 0 | 1 | 1 | Alert! | – | Safe | Safe |
| 1 | 1 | 0 | 1 | Safe | Safe | – | Alert! |
| 1 | 1 | 1 | 0 | Safe | Safe | Alert! | – |
| 0 | 1 | 0 | 1 | – | Alert! | – | Alert! |
| 0 | 1 | 1 | 0 | – | Alert! | Alert! | – |
| 1 | 0 | 0 | 1 | Alert! | – | – | Alert! |
| 1 | 0 | 1 | 0 | Alert! | – | Alert! | – |

vector of switch states **S** is not safe for the system. All these alerts are cases which would otherwise go unnoticed, thus stressing the extra security and safety precautions provided by the local monitoring approach.

#### Transformer tap switch

The previous scenario was analyzing the situation of an RTU controlling a bus operating at a Medium Voltage level. The following scenario monitors an RTU that controls different voltage levels, namely High and Medium Voltage. This is done via so-called tap switches; by changing their setting, the transformer changes the ratio of the voltage values on its primary and secondary side. This ratio change results in changing the value of the secondary voltage, while the voltage on the primary side remains the same. The transformer marked in Fig. 4a connects the High and Medium Voltage levels and contains a controllable tap switch. The operator can send commands from the control room in order to change the value of the voltage on the secondary side of the transformer.

The main safety requirement that is tested when changing the tap switch position is the voltage value on the secondary windings of the transformer. The safety requirement defined in Table 3 defines that the voltage has to be equal to the nominal value $\pm 10\%$. This is defined for the Low Voltage areas (CENELEC 1988), however, in the proposed approach it is also possible to perform the same check for Medium Voltage, like proposed in Isozaki et al. (2014). The implementation of the monitoring tool on $RTU_1$ that controls the transformer needs to be done similarly like shown in section "Implementation of the interlocks" for interlocks (and is not shown here in detail). In the following, only the outcome of the performed tests are shown.

##### *Attack scenario*

A successful attack is performed by changing the tap switch to such a position that the value of the secondary voltage exceeds the maximum bounds. Since the nominal value of the secondary voltage is 10 kV, this means the voltage must stay within 9 kV and 11 kV. The initial ratio of the transformer, i.e., the ratio of the primary to secondary voltage is 11 in the following scenario. The transformer has 3 tap switch positions, resulting in ratios 11 (position 1), 10.5 (position 2) and 10 (position 3), respectively. If the primary voltage equals the nominal value of 110 kV, then setting the transformer's tap switch to position 3 results in violating the bound of the secondary voltage. The attacker opportunistically changes the tap switch position to different values, aiming to disturb the physical process. The lower part of Fig. 7 shows the voltage value on the secondary side of the transformer. It can be seen that at 16s (simulation time; $x$-axis), the attacker changed the position from 2 to 1, resulting in a voltage of 10 kV. This is a failed attack attempt, as the resulting voltage is well within bounds. Next, at around 32s another change is made: the tap position is changed back to 2, as the attacker does not know the initial value of the tap switch. Finally, at around 48s, the attacker changes the tap switch to position 3 which results in an undesired voltage value of 11 kV. If the attacker continues to perform changes, the monitoring approach will continue to filter out actions that lead to unsafe states. However, our approach is not able to detect the attacker.

##### *Results*

While the previous scenario covered all initially safe configurations, this section focuses on the analysis of the interaction in the testbed between receiving commands and issuing

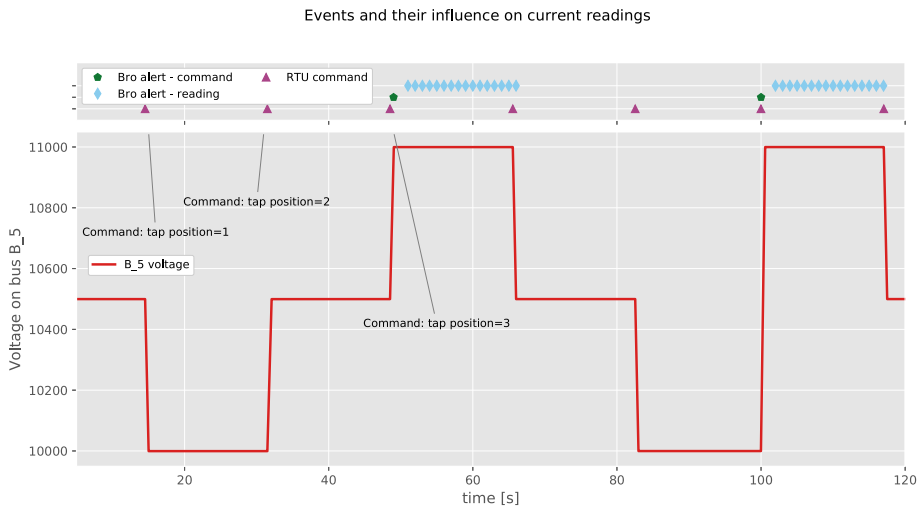


**Fig. 7** Illustration of the effects of an attack scenario on $RTU_1$ with local monitoring. Figure 7 illustrates the effects of an attack on a transformer controlled by $RTU_1$. Here, the traffic monitoring is enabled, however the commands are not discarded in order to present the outcome of those commands. The voltage on bus B_5 is shown in red on the y-axis for different simulation times (in seconds). It can be seen that Bro issues alerts (marked in green) when receiving commands to change the tap switch position, such that the secondary voltage would become to high. Once the command has been executed, a Bro warning is issued upon every sensor reading (marked with the blue diamonds)

alerts, as presented in Fig. 7. The upper part of Fig. 7 indicates the time when commands are sent by the attacker and the reaction of the monitoring tool to these commands. The events marked with a green pentagon represent alerts issued by Bro upon receiving the command to change the tap switch to a position that would result in a too high secondary voltage. This is a result of implementing the voltage safety requirement (cf. Table 3) upon receiving a new command (cf. the right-side loop of Fig. 2). Note that Fig. 2 indicates that the command that may bring the system to an unsafe state should be discarded. Here, only an alert was given in order to analyze the further behavior of the system.

The blue diamonds represent the warnings issued by Bro due to violations of the voltage safety constraint upon receiving a new reading (cf. Fig. 2, the left-side loop).

#### Advantages of monitoring in a simulation testbed

Section "Interlocks" and section "Transformer tap switch" presented how the proposed testbed can be used to investigate the effect of the proposed process-based monitoring on the security in field stations. In both cases the testbed has shown that the monitoring tool responds accurately to the processed command, e.g., generates alerts for commands that would bring the system to an unsafe state.

Furthermore, using a simulation testbed, allows to investigate the consequences of executing a malicious command versus discarding it or simply issuing an alert. This would not be possible in real infrastructures and still very difficult in a physical testbed.

Moreover, the proposed co-simulation testbed lends itself to stress tests, e.g., regarding the frequency of reading commands and how this influences the number of alerts and the accuracy of the monitoring tool.

Also the real-time capabilities of the proposed approach can be evaluated for the presented test cases. The first investigated scenario, i.e., monitoring the interlocks, focused

on 4 elements in the switch vector describing part of the system state, and on the sensor measurements on the 4 connected power lines. The second scenario investigated the transformer tap switch position vector with a single element and the sensor readings of two power lines on the primary and secondary side of the transformer. In these scenarios, calculating the resulting system state and the policy checking within Bro caused message delays of only 0.002 ms on average.

Clearly, a more thorough investigation of the real-time performance is needed for different sizes of field stations, before bringing this approach to market. However, as the approach is meant to work locally at field stations, the models should not become much larger than for the scenarios analyzed here. Hence, scalability should not be a problem in this distributed approach.

## Comparison of the proposed system to existing approaches

In the following, the related work on process-aware monitoring in SCADA (section "Process-aware monitoring") and on testbeds for the control of power distribution (section "Testbeds") is discussed.

### Process-aware monitoring

Traditional IDSes, even if they provide support for SCADA, rely on the detection of unusual packets: *whitelisting* relies on knowledge of the source/destination host and ports (Barbosa and Pras 2010); rules can be implemented in network intrusion detection system to check whether packet formatting and packet content match protocol specification (Roesch 1999; Cheung et al. 2007). However, by analyzing only the properties of the exchanged packets, a system is not able to detect well-formatted legitimate packets which could nevertheless harm the underlying physical system.

Using the state of both the *control network* and the *state of the physical process* to improve security has been proposed before under different names: (Lin et al. 2016; Wain et al. 2016) discuss semantic-based security analysis, (Bao et al. 2016) describes a similar approach as behavior-based detection, and (Urbina et al. 2016; Koutsandria et al. 2015) introduce physics-based attack detection. Hadžiosmanović et al. (2014) characterize the types of the variables in the network traffic based on their behavior over time and model the resulting regularity. This approach assumes that the process variables remain consistent over time. Moreover, this approach does not predict the outcome of an incoming command, it rather detects whether process variables deviate from their normal value. This approach has been shown to be 98% accurate in real-life traffic. Lin et al. (2013; 2016) propose an intrusion detection system for SCADA systems controlling the power grid, targeting attacks that send commands that potentially harm the physical system but are hidden in a legitimate format. Although accurate, this approach heavily relies on the assumption that the monitoring system, i.e., a central Master IDS, remote Slave IDSes, and the communication link are not compromised themselves.

Urbina et al. (2016) study the detection of stealthy attacks in a system controlling the acidity level of a fluid in a tank. Using real-time measurements from the tank and a physical model of the process being controlled allows detecting malicious behavior if the observations are significantly different from the model-based predictions. The authors present both, a stateful and a stateless approach. Koutsandria et al. (2015) investigate the so-called "physics aware" Hybrid Control Network IDS (HC-NIDS), which checks a set

of cyber-physical security policies on the communication traffic obtained from a network tap. This HC-NIDS is tailored to the protection of digital relays (Koutsandria et al. 2014) and can also be used in automated power distribution systems when adjusting the rules accordingly (Parvania et al. 2014).

Caselli et al. (2015) do not take process information into account, directly. However, they investigate the importance of sequences of commands in the ICS setting. The violation of pre-defined sequences of commands can directly impact the process negatively. Sequences of packets are modeled as a discrete-time Markov chain and compared to a pre-computed reference model, which represents normal traffic behavior. Nivethan and Papa (2016a) propose a SCADA IDS framework that incorporates process semantics, by implementing extra warning notifications in case process variables exceed some threshold values. A system description language and a mapper for turning requirements into actual Bro policies is also provided. This approach is considered *static*, as it computes policies and thresholds, but only once. This approach is not validated and to some extent duplicates the work of Human Machine Interfaces (HMIs) in SCADA. Moreover, the authors in Nivethan and Papa (2016b) analyze the use of open source firewalls in SCADA/ICS and propose to use `iptables`}for filtering SCADA traffic. Using string matching they detect, e.g., unauthorized write commands and test this approach on Modbus/TCP traffic. Bao et al. (2016) use rules obtained from physical properties of the system, which are then translated into state machines. Based on measurements from the system, the state machines are updated continuously and when reaching a critical state a warning is issued to the operator. Mashima et al. (2016) propose to implement an *active command mediation* mechanism in the electrical substations. Their approach builds on the idea to actively inspect and pre-process the command sent to the remote station before executing it on the physical power system devices. The authors provide an example implementation of this mechanism, the so-called *command delaying mechanism*. In this mechanism, a command could be delayed by a number of proxies so the central system has the opportunity to cancel such a command.

Table 6 summarizes and compares the related work discussed above. The table indicates whether the used approach is specification-based or learned from the traffic. It mentions the sector to which the approach has been applied: PG indicates the Power Grid, while ICS indicates a more generic approach and refers to Industrial Control Systems in general. The validation method used in the literature is listed either as TB - physical TestBed, SIM - SIMulation or RS - Real System (Real Traffic). An approach is capable of detecting attacks or can also prevent attacks, as indicated in the table. Moreover, the detection rules used in the approach are compared. They are either static - generated only once - or dynamically adapt to the current system state. The combination of a learned approach with static rules means that the approach investigates only one-time learning for the proposed mechanism. Finally, the location, which is the placement of the detection mechanism, is compared. It either uses local information and protects a single station, is distributed and relies on information from multiple controllers, or centrally works with information from the entire network, protecting the whole system.

Table 6 shows that most approaches tailored for the power grid are based on specifications of the power grid. Approaches that only detect but cannot prevent attacks mainly duplicate the work of the HMI, as operators are notified about values exceeding pre-defined thresholds. Furthermore, adapting models of the physical process during

**Table 6** Comparison of process-aware IDS techniques

| Work | Approach | Sector | Validation | Detection | Prevention | Rules | Location |
|---|---|---|---|---|---|---|---|
| Hadžiosmanović et al. (2014) | Learned | ICS (tank) | TB, RS | Yes | No | Static | Local |
| Lin et al. (2013; 2016) | Specification | PG | SIM | Yes | Yes | Dynamic | Central |
| Urbina et al. (2016) | Specification | ICS, PG | SIM, TB, RS | Yes | No | Dynamic | Local |
| Koutsandria et al. (2015; 2014), Parvania et al. (2014)) | Specification | PG | TB | Yes | Yes | Dynamic (not yet) | Local |
| Caselli et al. (2015) | Learned | ICS | RS | Yes | No | Static | Local |
| Nivethan and Papa (2016b); Nivethan and Papa (2016a) | Specification | PG | None | Yes | No | Static | Local |
| Bao et al. (2016) | Specification | PG | SIM | Yes | No | Dynamic | Distributed |
| Mashima et al. (2016) | Specification | PG | SIM | Yes | Yes | Dynamic | Distributed Delay, Central Detection |
| The proposed approach | Specification | PG | SIM | Yes | Yes | Dynamic | Local |

ICS - Industrial Control Systems, PG - Power Grid, SIM - Simulation, TB - TestBed, RS - Real System (or trace)

run-time is not done often to prevent attacks. Summarizing, the proposed approach is close to (Koutsandria et al. 2014), however, it operates on a simulation engine and also the proposed system implementation already updates rules dynamically, based on the state of the physical process.

### Testbeds

The proposed approach aims to locally monitor and perform detection analysis at field stations, hence, there is no need to simulate the entire control network. However, a simulation engine for the controller, e.g., an RTU or PLC, which receives information from the SCADA network and sends commands and requests to the physical process is required. Hence, the controller is the main interface between the physical process and the control network including the control room.

In contrast, current co-simulation environments focus on simulating the *entire* network using Omnet++ (Awad et al. 2016; Lévesque et al. 2012), ns2 (Lin et al. 2011), RINSE (Davis et al. 2006) or OPNET (Sadi et al. 2015), and evaluate, e.g., denial of service attacks on the control network *only*. These fully simulated approaches are highly flexible, while more advanced testbeds (Koutsandria et al. 2015; Kang et al. 2015; Gunathilaka et al. 2016; Sadi et al. 2015), may require a connection to emulate real hardware or the use of proprietary software. Non-virtualized testbeds at Distribution System Operators (DSOs) are less flexible and often difficult to access. All simulation-based approaches require a power simulator, like Power World (Davis et al. 2006; Gunathilaka et al. 2016), OpenDSS (Awad et al. 2016; Lévesque et al. 2012), PSFL (Lin et al. 2011), or MATPOWER-based *Matlab/Simulink* (Sadi et al. 2015; Koutsandria et al. 2015) or *Mosaik* (Schloegl et al. 2015). The latter easily integrates existing simulators in the smart grid co-simulation framework. Moreover, if needed, new simulators can be attached to the *Mosaik* co-simulation framework by using the provided API. This is the main reason why *Mosaik* was chosen in the proposed testbed and integrated with

(part of) the Modbus/TCP based control network and the monitoring tool, as shown previously.

Table 7 summarizes the characteristics of the investigated co-simulation environments. For each framework the availability is specified: either it is available under an Open Source license (OS), or tools are openly available, but the source code is not (OS*). The table indicates if a paid license is required for an element used in the co-simulation environment (LIC), or if it is a physical Test Bed (TB) or uses some other Hardware In the Loop (HIL). Next, the integration of the simulator is discussed: If a programming language (such as Python, Java or C++) is specified in the table, the approach has a dedicated interface written in that language which enables the integration of simulators. Two of the approaches use other communication protocols, such as HTTP requests (Lévesque et al. 2012) or VPN connections (Davis et al. 2006). One approach uses a physical testbed, which requires physical connections between hardware components (Kang et al. 2015). The table shows which simulator is used for the SCADA network and for the power grid system. The extensibility of the co-simulation testbed is specified, and all available communication protocols are listed.

While older approaches mostly do not investigate particular SCADA protocols, newer approaches are tailored towards Modbus and or substation automation protocols. The testbed described in this paper is not only flexible, but also easily extensible to include new simulators, e.g., for controllers using other protocols used in power distribution, and only uses open-source software and libraries.

## Conclusions

Detecting potentially malicious commands in systems controlling power distribution is mostly performed in a central control room. However, due to the modernization and

**Table 7** Table comparing related works on testbeds environments

| Work | Availability | Integration | Network | Power | Extensibility | Protocols |
|---|---|---|---|---|---|---|
| Davis et al. (2006) (2008) | LIC | Server requests over VPN | RINSE, TCP/IP | Power-World | HIL | Modbus, TCP/IP |
| Lin et al. (2011) | OS* | C++/Java | ns2 | PSFL | Not discussed | Not discussed |
| Levesque et al. (2012) | OS* | HTTP requests | Omnet++ | OpenDSS | Not discussed | Not discussed |
| Sadi et al. (2015) | LIC | C | OPNET | Matlab/ Simulink | Not discussed | Not discussed |
| Kang et al. (2015) | TB | Physical links | IEC61850 device | PV simulator | Not discussed | Modbus, IEC61850 |
| Koutsandria et al. (2015) | HIL/ TB | C, C++, Python | Modbus | Matlab/ Simulink | Other protocols | Modbus, TCP/IP |
| Awad et al. (2016) | OS | C++ | Omnet++ | OpenDSS | Not discussed | TCP/IP, 802.11, Ethernet |
| Gunthilaka et al. (2016) | LIC | Java | IEC61850 simulator | Power-World | Other protocols | IEC-104, IEC61850 |
| The proposed approach (2017) | OS | Python | Modbus simulator | PyPower | Other protocols; other SG elements | Modbus, TCP/IP |

OS - Open Source, OS* - open source elements, however the source code is not made available, LIC - License required , TB - TestBed, PSLF - Positive Sequence Load Flow, HIL - Hardware In Loop

automation of field stations and the use of standardized protocols, also remote field stations may be the target of (insider) attacks and require improved security. Research in this direction requires a testbed that is capable of simulating both the physical power distribution system and the control network.

This paper presents a co-simulation testbed that can be used to implement and evaluate local monitoring approaches for SCADA systems as proposed before, e.g., (Chromik et al. 2016a; Koutsandria et al. 2014; Urbina et al. 2016; Chromik et al. 2017; Meliopoulos et al. 2017). The presented testbed environment is based on the co-simulation framework *Mosaik* and simulates both the power distribution system and a control network implementing the communication protocol Modbus/TCP. Moreover, a monitoring system based on process-aware policies implemented using the Bro monitoring tool is presented. For better reference, the paper also provides an extensive overview on the related work on approaches for process-aware intrusion detection systems and on testbed environments for power grids.

The paper describes the simulators developed and used in the proposed testbed and presents the influence of cyber commands on the power distribution system. With the simulated Modbus/TCP controller, it is possible to remotely change the topology of the simulated power distribution system. This allows for, e.g., simulating attacks on power distribution and testing the working on the monitoring tool for various initial states of the power system.

This paper also presents an approach to implement policies depending on the system state, using the Bro network intrusion detection system. Knowing the system physical constraints and safety requirements, such as the interlocks, the proposed detection mechanism is configured and updated in order to reject commands that can bring the physical system to an unsafe state. Even though the rules are static, the outcome of a command at a particular moment in time depends on the current state of the physical system. For various examples, it is presented how such local monitoring helps to improve the security of the power distribution field stations, when malicious commands are sent from the control room (insider attacks). This has been illustrated for two different settings, one in the medium voltage area (interlocks) and one between medium and high voltage (tap switch).

Future work will compare the performance and accuracy of this local monitoring approach with a centralized approach (Lin et al. 2016). Furthermore, the amount of local information necessary to perform accurate monitoring will be investigated and the proposed approach will be further evaluated using the IEEE benchmark suite (Distribution Test Feeders 2018).

## Endnotes

[1] http://mosaik.readthedocs.io/en/latest/installation.html

[2] http://pymodbus.readthedocs.io/en/latest/index.html

### Abbreviations

DEM: Decentralized energy management; DSO: Distribution system operator; EMS: Energy management system; HMI: Human machine interface; ICT: Information and communications technology; ICS: Industrial control systems; IDS: Intrusion detection system; PLC: Programmable logic controller; PV: PhotoVoltaic; RTU: Remote terminal unit; SCADA: Supervisory control and data acquisition; TCP: Transmission control protocol

### Acknowledgements

We thank the reviewers for their detailed and constructive comments.

**Funding**
This research is funded through the NWO project ("MOre secure scada through SElf-awarenesS") grant nr. 628.001.012.

**Availability of data and materials**
The graphs presented in the paper are available at: https://github.com/jjchromik/mosaik-events-notebook. The code of the simulators is available here: https://github.com/jjchromik/mosaik-cosim.

**Authors' contributions**
JC and AR designed the setup of the testbed, with discussions with BH. JC developed and adjusted the simulators needed for co-simulation in Mosaik and performed the experiments. JC, AR and BH jointly developed the and wrote the paper. All authors read and approved the final manuscript.

**Competing interests**
The authors declare that they have no competing interests.

## Publisher's Note


**Author details**
[1] University of Twente, Enschede, Netherlands. [2] Westfälische Wilhelms-Universität, Münster, Germany.

Received: 21 March 2018 Accepted: 6 September 2018
Published online: 06 November 2018

**References**
Awad A, Bazan P, German R (2016) SGsim: Co-simulation framework for ICT-enabled power distribution grids. In: Proceedings of the International GI/ITG Conference on Measurement, Modelling, and Evaluation of Dependable Computer and Communication Systems (MMB&DFT). Springer, Münster. pp 5–8. https://doi.org/10.1007/978-3-319-31559-1_2

Bao H, Lu R, Li B, Deng R (2016) BLITHE: Behavior rule-based insider threat detection for smart grid. IEEE Internet Things J 3(2):190–205. https://doi.org/10.1109/JIOT.2015.2459049

Barbosa R, Pras A (2010) Intrusion detection in SCADA networks. In: Proceedings of the Mechanisms for Autonomous Management of Networks and Services. Springer, Zurich. pp 163–166. https://doi.org/10.1007/978-3-642-13986-4_23

Bell M, Berkel F, Liu S (2018) Real-Time Distributed Control of Low Voltage Grids with Dynamic Optimal Power Dispatch of Renewable Energy Sources. IEEE Trans Sust Energ :1–8. https://doi.org/10.1109/TSTE.2018.2800108, early access

Bush SF (2014) Smart Grid: Communication-Enabled Intelligence for the Electric Power Grid. Wiley, Chichester. https://books.google.nl/books?id=C0ecAgAAQBAJ

Cardenas A, Amin S, Sinopoli B, Giani A, Perrig A, Sastry S (2009) Challenges for securing cyber physical systems. In: Proceedings of the Workshop on Future Directions in Cyber-physical Systems Security, Newark. pp 1–5. http://feihu.eng.ua.edu/NSF_CPS/year1/w2_read.pdf

Caselli M, Zambon E, Kargl F (2015) Sequence-aware intrusion detection in industrial control systems. In: Proceedings of the 1st ACM Workshop on Cyber-Physical System Security. ACM, Singapore. pp 13–24. https://doi.org/10.1145/2732198.2732200

CENELEC (1988) Harmonisation Document: Nominal voltage for low voltage public electricity supply systems, HD 472 S1. European Committee for Electrotechnical Standardization, Brussels

Cheung S, Dutertre B, Fong M, Lindqvist U, Skinner K, Valdes A (2007) Using Model-based Intrusion Detection for SCADA Networks. In: Proceedings of the SCADA Security Scientific Symposium, Miami Beach. pp 1–12. http://www.csl.sri.com/papers/scadaIDS07/

Chromik JJ, Remke A, Haverkort BR (2016a) Improving SCADA security of a local process with a power grid model. In: Proceedings of the 4th International Symposium for ICS&SCADA Cyber Security Research, Queen's Belfast University, UK. BCS Learning & Development Ltd. pp 114–123. https://doi.org/10.14236/ewic/ICS2016.13

Chromik JJ, Remke A, Haverkort BR (2016b) What's under the hood? Improving SCADA security with process awareness. In: Proceedings of the Joint Workshop on Cyber-Physical Security and Resilience in Smart Grids. IEEE, Vienna. pp 1–6. https://doi.org/10.1109/CPSRSG.2016.7684100

Chromik JJ, Pilch C, Brackmann P, Duhme C, Everinghoff F, Giberlein A, Teodorowicz T, Wieland J, Haverkort BRHM, Remke AKI (2017) Context-aware local Intrusion Detection in SCADA systems: a testbed and two showcases. In: Proceedings of the International Conference on Smart Grid Communications (SmartGridComm). IEEE, Dresden. https://doi.org/10.1109/SmartGridComm.2017.8340672

Ciocia A, Chicco G, Di Leo P, Gai M, Mazza A, Spertino F, Hadj-Said N (2017) Voltage control in low voltage grids: A comparison between the use of distributed photovoltaic converters or centralized devices. In: Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), 2017 IEEE International Conference On. IEEE. pp 1–6. https://doi.org/10.1109/EEEIC.2017.7977815

Cleveland F (2012) IEC TC57 security standards for the power system's information infrastructure–beyond simple encryption. In: Proceedings of the Transmission and Distribution Conference and Exhibition, Dallas, TX, USA Vol. 2006. pp 1079–1087. https://doi.org/10.1109/TDC.2006.1668652

CRASHOVERRIDE (2017) Analysis of the Threat to Electric Grid Operations. Technical report, DRAGOS. Available online: https://dragos.com/blog/crashoverride/CrashOverride-01.pdf. Accessed 10 Nov 2017

Davis C, Tate J, Okhravi H, Grier C, Overbye T, Nicol D (2006) SCADA cyber security testbed development. In: Proceedings of the 38th North American Power Symposium (NAPS). IEEE, Carbondale. pp 483–488. https://doi.org/10.1109/NAPS.2006.359615

Distribution Test Feeders (2018). Available online: http://sites.ieee.org/pes-testfeeders/resources/. Accessed 10 Nov 2017

Éva Á, Gábor J, Tamás SP (2018) Proposal of a secure modbus rtu communication with adi shamir's secret sharing method. Int J Electron Telecommun 64(2):107–114

Fovino IN, Carcano A, Masera M, Trombetta A (2009) Design and implementation of a secure modbus protocol. In: International Conference on Critical Infrastructure Protection. Springer, Berlin. pp 83–96

Gunathilaka P, Mashima D, Chen B (2016) SoftGrid: A Software-based Smart Grid Testbed for Evaluating Substation Cybersecurity Solutions. In: Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy. ACM, Vienna. pp 113–124. https://doi.org/10.1145/2994487.2994494

Hadžiosmanović D, Sommer R, Zambon E, Hartel PH (2014) Through the eye of the PLC: semantic security monitoring for industrial processes. In: Proceedings of the 30th Annual Computer Security Applications Conference. ACM, New Orleans. pp 126–135. https://doi.org/10.1145/2664243.2664277

ICS-CERT (2016) Year in Review, Industrial Control Systems Cyber Emergency Response Team Technical report, NCCIC, ICS-CERT. Available online: https://ics-cert.us-cert.gov/sites/default/files/Annual_Reports/Year_in_Review_FY2016_Final_S508C.pdf. Accessed 10 Nov 2017

ICS-CERT (2010) Advisory (ICSA-10-272-01): Primary Stuxnet Advisory. Available online: https://ics-cert.us-cert.gov/advisories/ICSA-10-272-01. Accessed 10 Nov 2017 (released September 29, 2010)

ICS-CERT (2018b) Alert (IR-ALERT-H-16-056-01) Cyber-Attack Against Ukrainian Critical Infrastructure. Available online: https://ics-cert.us-cert.gov/alerts/IR-ALERT-H-16-056-01. Accessed 10 Nov 2017 (released February 25, 2016)

IEC Webstore (2018) Power systems management and associated information exchange - Data and communications security. Available online: https://webstore.iec.ch/publication/6912. Accessed 20 Aug 2018

Isozaki Y, Yoshizawa S, Fujimoto Y, Ishii H, Ono I, Onoda T, Hayashi Y (2014) On detection of cyber attacks against voltage control in distribution power grids. In: IEEE International Conference on Smart Grid Communications. IEEE, Venice. pp 842–847

Kang B, Maynard P, McLaughlin K, Sezer S, Andrén F, Seitl C, Kupzog F, Strasser T (2015) Investigating cyber-physical attacks against IEC 61850 photovoltaic inverter installations. In: Proceedings of the 20th Conference on Emerging Technologies & Factory Automation. IEEE, Luxembourg. pp 1–8. https://doi.org/10.1109/ETFA.2015.7301457

Kenner S, Thaler R, Kucera M, Volbert K, Waas T (2016) Comparison of smart grid architectures for monitoring and analyzing power grid data via Modbus and REST. EURASIP J Embed Syst 2017(1):1–13. https://doi.org/10.1186/s13639-016-0045-7

Khan S, Mauri JL (2013) Green Networking and Communications: ICT for Sustainability. CRC Press, Boca Raton

Kleinmann A, Amichay O, Wool A, Tenenbaum D, Bar O, Lev L (2017) Stealthy Deception Attacks Against SCADA Systems. CoRR abs/1706.09303. 1706.09303

Koutsandria G, Gentz R, Jamei M, Scaglione A, Peisert S, McParland C (2015) A real-time testbed environment for cyber-physical security on the power grid. In: Proceedings of the 1st ACM Workshop on Cyber-Physical Systems-Security. ACM, Denver. pp 67–78. https://doi.org/10.1145/2808705.2808707

Koutsandria G, Muthukumar V, Parvania M, Peisert S, McParland C, Scaglione A (2014) A hybrid network IDS for protective digital relays in the power transmission grid. In: Proceedings of the International Conference on Smart Grid Communications (SmartGridComm). IEEE, Venice. pp 908–913. https://doi.org/10.1109/SmartGridComm.2014.7007764

Lévesque M, Xu DQ, Joós G, Maier M (2012) Communications and power distribution network co-simulation for multidisciplinary smart grid experimentations. In: Proceedings of the 45th Annual Simulation Symposium. Society for Computer Simulation International, Orlando. pp 1–7

Lin H, Sambamoorthy S, Shukla S, Thorp J, Mili L (2011) Power system and communication network co-simulation for smart grid applications. In: Proceedings of the Innovative Smart Grid Technologies (ISGT). IEEE, Anaheim. pp 1–6. https://doi.org/10.1109/ISGT.2011.5759166

Lin H, Slagell A, Di Martino C, Kalbarczyk Z, Iyer RK (2013) Adapting Bro into SCADA: Building a Specification-based Intrusion Detection System for the DNP3 Protocol. In: Proceedings of the 8th Annual Cyber Security and Information Intelligence Research Workshop. ACM, Oak Ridge. pp 5–154. http://doi.acm.org/10.1145/2459976.2459982

Lin H, Slagell A, Kalbarczyk Z, Sauer P, Iyer R (2016) Runtime semantic security analysis to detect and mitigate control-related attacks in power grids. IEEE Trans Smart Grid 99:1–16. https://doi.org/10.1109/TSG.2016.2547742

Liu Y, Ning P, Reiter MK (2011) False data injection attacks against state estimation in electric power grids. TISSEC 14(1):13–11333. https://doi.org/10.1145/1952982.1952995

Lu S, Repo S, Della Giustina D, Figuerola FA-C, Löf A, Pikkarainen M (2015) Real-time low voltage network monitoring—ICT architecture and field test experience. IEEE Trans Smart Grid 6(4):2002–2012. https://doi.org/10.1109/TSG.2014.2371853

Mashima D, Gunathilaka P, Chen B (2016) An active command mediation approach for securing remote control interface of substations. In: Proceedings of the International Conference on Smart Grid Communications (SmartGridComm). IEEE, Sydney. pp 147–153. https://doi.org/10.1109/SmartGridComm.2016.7778753

Maynard P, McLaughlin K, Haberler B (2014) Towards Understanding Man-in-the-middle Attacks on IEC 60870-5-104 SCADA Networks. In: Proceedings of the 2nd International Symposium for ICS&SCADA Cyber Security Research, St Pölten, Austria. pp 30–42. https://doi.org/10.14236/ewic/ics-csr2014.5

Meliopoulos S, Cokkinides G, Fan R, Sun L (2017) Data Attack Detection and Command Authentication via Cyber-Physical Co-Modeling. IEEE Des Test 34(4):34–43. https://doi.org/10.1109/MDAT.2017.2682233

Mustard S (2005) Security of distributed control systems: The concern increases. Comput Control Eng J 16(6):19–25. https://doi.org/10.1049/cce:20050605

Nicholson A, Webber S, Dyer S, Patel T, Janicke H (2012) SCADA security in the light of Cyber-Warfare. Comput Secur 31(4):418–436. https://doi.org/10.1016/j.cose.2012.02.009

Nivethan J, Papa M (2016a) A SCADA Intrusion Detection Framework that Incorporates Process Semantics. In: Proceedings of the 11th Annual Cyber and Information Security Research Conference. ACM, Oak Ridge. pp 1–5. https://doi.org/10.1145/2897795.2897814

Nivethan J, Papa M (2016b) On the use of open-source firewalls in ICS/SCADA systems. Information Security Journal: A Global Perspective 25(1-3):83–93. https://doi.org/10.1080/19393555.2016.1172283
OFFIS (2017) Mosaik Documentation. Available online: http://mosaik.readthedocs.io/en/latest/overview.html. Accessed 10 Nov 2017
Oman P, Schweitzer E, Frincke D (2000) Concerns about intrusions into remotely accessible substation controllers and SCADA systems. In: Proceedings of the 27th Annual Western Protective Relay Conference. pp 1–16. https://doi.org/10.1.1.20.6519
Parvania M, Koutsandria G, Muthukumary V, Peisert S, McParland C, Scaglione A (2014) Hybrid control network intrusion detection systems for automated power distribution systems. In: Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). IEEE. pp 774–779. https://doi.org/10.1109/DSN.2014.81
Paxson V (1999) Bro: a system for detecting network intruders in real-time. Comput Netw 31(23):2435–2463. https://doi.org/10.1016/S1389-1286(99)00112-7
PYPOWER (2018) PYPOWER. Available online: https://pypi.org/project/PYPOWER/. Accessed 31 Aug 2018
Roesch M (1999) Snort - Lightweight Intrusion Detection for Networks. In: Proceedings of the 13th USENIX Conference on System Administration. LISA '99. USENIX Association, Seattle. pp 229–238. https://doi.org/10.1.1.105.6212. http://dl.acm.org/citation.cfm?id=1039834.1039864
Sadi MAH, Ali MH, Dasgupta D, Abercrombie RK, Kher S (2015) Co-Simulation Platform For Characterizing Cyber Attacks in Cyber Physical Systems. In: Proceedings of the IEEE Symposium Series on Computational Intelligence. IEEE, Cape Town. pp 1244–1251. https://doi.org/10.1109/SSCI.2015.178
Schloegl F, Rohjans S, Lehnhoff S, Velasquez J, Steinbrink C, Palensky P (2015) Towards a classification scheme for co-simulation approaches in energy systems. In: Proceedings of the International Symposium on Smart Electric Distribution Systems and Technologies. IEEE, Vienna. pp 516–521. https://doi.org/10.1109/SEDST.2015.7315262
Shahzad A, Lee M, Lee Y-K, Kim S, Xiong N, Choi J-Y, Cho Y (2015) Real time modbus transmissions and cryptography security designs and enhancements of protocol sensitive information. Symmetry 7(3):1176–1210
Smart Grids in Distribution Networks (2015) Roadmap Development and Implementation. Technical report, International Energy Association. Available online: https://www.iea.org/publications/freepublications/publication/TechnologyRoadmapHow2GuideforSmartGridsinDistributionNetworks.pdf. Accessed 10 Nov 2017
Teixeira A, Dán G, Sandberg H, Johansson KH (2011) A cyber security study of a SCADA energy management system: Stealthy deception attacks on the state estimator. IFAC Proc Vol 44(1):11271–11277. https://doi.org/10.3182/20110828-6-IT-1002.02210
The Modbus Organization (2012) Modbus application protocol specification, ver. 1.1b3. Available online: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf. Accessed 10 Nov 2017
Udd R, Asplund M, Nadjm-Tehrani S, Kazemtabrizi M, Ekstedt M (2016) Exploiting Bro for intrusion detection in a SCADA system. In: Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security. ACM, Xi'an. pp 44–51. https://doi.org/10.1145/2899015.2899028
Urbina DI, Giraldo JA, Cardenas AA, Tippenhauer NO, Valente J, Faisal M, Ruths J, Candell R, Sandberg H (2016) Limiting the impact of stealthy attacks on industrial control systems. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security. ACM, Vienna. pp 1092–1105. https://doi.org/10.1145/2976749.2978388
Wain A, Reiff-Marganiec S, Janicke H, Jones K (2016) Towards a Distributed Runtime Monitor for ICS/SCADA Systems. In: Proceedings of the 4th International Symposium for ICS&SCADA Cyber Security Research, Queen's Belfast University, UK. BCS Learning & Development Ltd., Belfast. pp 132–141. https://doi.org/10.14236/ewic/ICS2016.15
Zambon E, Cairo I, Costante E, Guadagnoli M, Lavernia D, Leon GE, Marin J, Barbosa RRR, Ribak A, Ruiz A, Trilla L (2015) D2.3 Reference Taxonomy on Industrial Control Systems Networks for Utilities. Technical Report. Technical report, PREEMPTIVE. Available online: http://preemptive.eu/wp-content/uploads/2015/07/preemptive_deliverable-d2.3.pdf. Accessed 10 Nov 2017