

Stealthy Deception Attacks Against SCADA Systems

Amit Kleinmann¹ Ori Amichay¹ Avishai Wool¹
David Tenenbaum² Ofer Bar² Leonid Lev²

¹ Cryptography and Network Security Lab, School of Electrical Engineering
Tel-Aviv University, Ramat Aviv 6997801, Israel

`a.b.kleinmann@gmail.com, oriamich@gmail.com, yash@acm.org`

² Israel Electric Corporation, 1 Netiv Ha'Or Haifa 3100001, Israel
{ `david.tenenbaum | br.ofer | leonid.lev` }@iec.co.il

Abstract. SCADA protocols for Industrial Control Systems (ICS) are vulnerable to network attacks such as session hijacking. Hence, research focuses on network anomaly detection based on meta-data (message sizes, timing, command sequence), or on the state values of the physical process. In this work we present a class of semantic network-based attacks against SCADA systems that are undetectable by the above mentioned anomaly detection. After hijacking the communication channels between the Human Machine Interface (HMI) and Programmable Logic Controllers (PLCs), our attacks cause the HMI to present a fake view of the industrial process, deceiving the human operator into taking manual actions. Our most advanced attack also manipulates the messages generated by the operator's actions, reversing their semantic meaning while causing the HMI to present a view that is consistent with the attempted human actions. The attacks are totally stealthy because the message sizes and timing, the command sequences, and the data values of the ICS's state all remain legitimate.

We implemented and tested several attack scenarios in the test lab of our local electric company, against a real HMI and real PLCs, separated by a commercial-grade firewall. We developed a real-time security assessment tool, that can simultaneously manipulate the communication to multiple PLCs and cause the HMI to display a coherent system-wide fake view. Our tool is configured with message-manipulating rules written in an ICS Attack Markup Language (IAML) we designed. Our semantic attacks all successfully fooled the operator and brought the system to states of blackout and possible equipment damage.

Keywords: SCADA, Stealthy Deception Attacks, IDS, NIDS, ICS

1 Introduction

Industrial Control Systems (ICS) are used for monitoring and controlling numerous industrial systems and processes. Supervisory Control And Data Acquisition (SCADA) system is a type of ICS that typically comprises of different

stations distributed over large geographical areas. Anomaly-based intrusion detection approaches are based on “the belief that an intruder’s behavior will be noticeably different from that of a legitimate user” [33]. The main types of anomaly detection approaches that are applied to SCADA systems [3,4,10] are: Network-aware detection in which the anomaly detection models only consider network and OS-level events; Protocol-aware detection in which modeling the normal traffic relies on deep-packet-inspection and considers the SCADA control protocol’s meta-data (message sizes, timing, argument addresses, command sequence); and Process-aware approaches which are based on process invariants, mathematical relationships among physical properties of the process controlled by the PLCs.

1.1 Contributions

In this work we present a class of semantic network-based attacks against SCADA systems, that are undetectable by either protocol-aware or process-aware anomaly detection. After hijacking the communication channels between the HMI and PLCs, our attacks manipulate the traffic so as to cause the HMI to present a fake view of the industrial process, thus deceiving the human operator into taking inappropriate and damaging manual actions. Our most advanced attack also manipulates the messages generated by the operator’s actions, reversing the semantic meaning of commands (‘Close’ becomes ‘Open’ and vice-versa) while causing the HMI to present a view that is consistent with the attempted human actions—thus inducing real damage on the cyber-physical system.

Our attacks are totally stealthy to SCADA-aware anomaly detectors since the message sizes and timing, and also the command sequences (including the command arguments) are all 100% legitimate in every way. Furthermore, our attacks are undetectable even by process-aware anomaly detection, since the observed data values of the ICS’s state are completely legitimate: they appear as natural fault conditions that the SCADA system is designed for, and the human operator is trained to handle. Even the operator’s manual command sequences are the expected actions when responding to a natural fault.

We implemented and tested several attack scenarios in the test lab of our local electric company, against a real HMI and real PLCs, separated by a commercial-grade firewall. To do so we developed a real-time security assessment tool, that can simultaneously manipulate the Modbus communication between the HMI and multiple PLCs, and cause the HMI to display a coherent system-wide fake view.

Our tool is configured with message-manipulating rules written in an ICS Attack Markup Language (IAML) we designed. Our tool is near-stateless—it only replaces message contents, without injecting, deleting, extending, or shortening messages. In one scenario we even used the tool in a “half-duplex” mode, wherein it only manipulates the HMI-to-PLC queries, while the PLC-to-HMI responses are unaltered. Despite these self-imposed limitations, our multi-stage semantic attacks all successfully fooled the operator and brought the system to states of blackout and possible equipment damage.

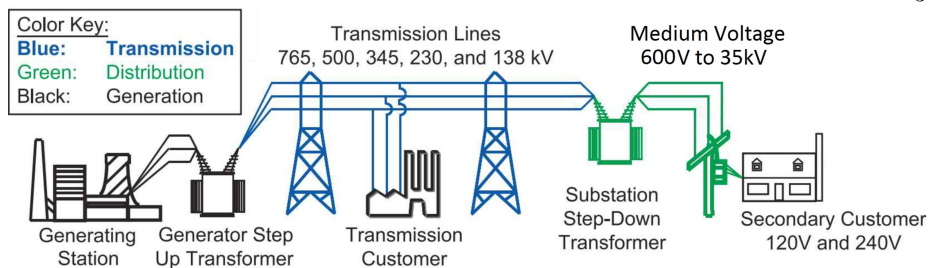


Fig. 1: Basic Structure of the Electric System (following [1])

The rest of this paper is organized as follows: Section 2 explains some fundamentals and assumptions. Section 3 describes the security assessment tool. In Section 4 we present three attacks we tested with their impact on the HMI and the PLCs. Section 5 succinctly reviews related work and in Section 6, we suggest possible countermeasures and conclusions.

2 Preliminaries

2.1 Electrical Distribution

An electricity supply chain is usually divided into three subsystems: generation, transmission, and distribution, as depicted in Figure 1. Electricity is transported along high voltage transmission lines (the transmission network) over long distances, from generation sites to major distribution points. The transmission lines are connected to distribution substations. At a distribution substation, a substation transformer takes the incoming transmission-level voltage (138 to 765 kV) and steps it down to several distribution primary circuits (“medium-voltage” circuits, 600V to 35kV), which fan out from the substation. Close to each end user, a distribution transformer takes the medium-voltage and steps it further down to a low-voltage secondary circuit (commonly 120/240V). In this paper we focus on the distribution subsystem, between the substation and distribution transformers—which is precisely the subsystem impacted during the Ukrainian cyber-attacks [27,28].

For improved reliability, distribution circuits are often provided with “tie switches” to other circuits which are normally open (i.e., disconnected). If a fault occurs on one of the circuits, the tie switches can be closed (connected) to let electricity flow into the faulted circuit, and to allow some portion of the service to be restored. The tie switches can be operated either manually, or automatically from the SCADA system interface. These switches, also called switchgears, may be simple open-air isolator switches or may be gas-insulated.

2.2 Adversary Model

Our underlying threat model is loosely based on the Dolev-Yao threat model [13]: The adversary may overhear and intercept all traffic regardless of its source and destination. More precisely, we assume the adversary has a Man-In-The-Middle (MITM) position between the HMI and *all* the PLCs. The adversary can inject, delete, and delay arbitrary packets with any source and destination addresses on the communication channels it controls. Consequently, the adversary can also

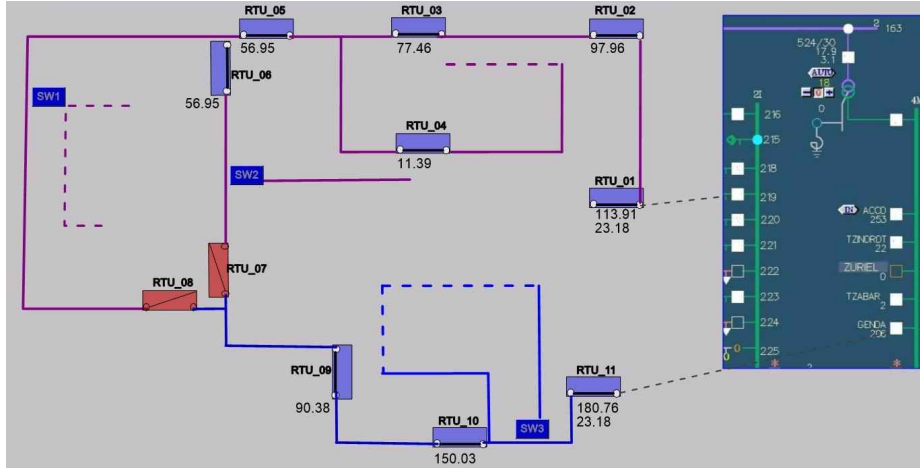


Fig. 2: A screenshot of the HMI panel. We see the substation is on the right, and 2 radial distribution lines: the top line (purple, via RTU 01–06) and the bottom line (blue, via RTU 11,10, 09). RTU 07 and 08 control the switchgears that tie the two lines: both are shown as open (disconnected). The numbers below each RTU show the current (113.91A and 180.76A at the line heads, dropping to 11.39A at RTU 04). At the two line heads (RTU 01, 11) the HMI also shows the voltage (23.18KV) displayed.

replay previously overheard messages, or manipulate messages in transit. The objective of the adversary is to manipulate the SCADA network to achieve an impact on the physical world.

We further assume that the adversary has in-depth knowledge of the architecture of the SCADA network and the various PLCs as well as sufficient knowledge of the physical process and the means to manipulate it via the SCADA system. Thus the adversary has the ability to fabricate messages that would result in real-world physical damage.

In our experiments we implemented a somewhat weaker type of attacker: Our attack tool is near-stateless and does not track or modify the TCP sequence numbers. Hence our attacks do not inject fabricated messages or drop legitimate ones: our attack tool only modifies the contents of pre-existing messages. Despite this self imposed restriction, our attacks are all successful, and undetectable by suggested anomaly-detection systems.

2.3 The Test Lab

The electric company test lab consists of 11 Unitronics V130 PLCs, controlled by a Cimplicity version 9.5 HMI running on a Microsoft Windows 2012 server. The HMI and the PLCs are connected to separate VLANs and separated by Check Point 4000 appliance R77.30 firewall.

The test lab emulates a substation with 2 radial distribution lines, that are interconnected by tie switches. Moreover, along each line there are several additional switchgears, see Figure 2. In total there are 11 PLCs, each controlling a

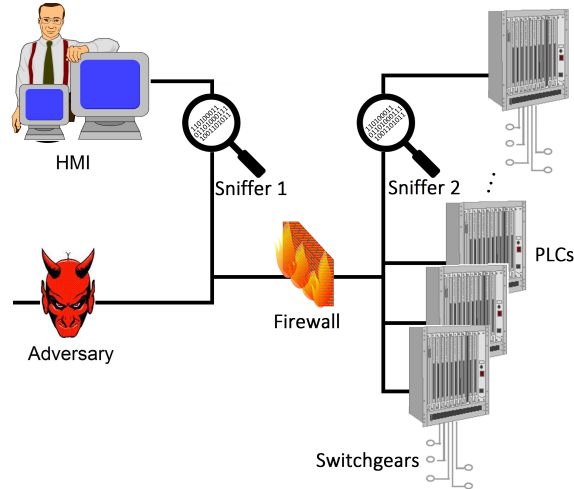


Fig. 3: Network diagram of the test lab

switchgear: each PLC reports back to the HMI the voltage and current flowing through it, and the switchgear state (open/closed), and accepts commands to open or close the switchgear.

The distribution system controlled by the PLCs is simulated by a separate PLC (the S-PLC) - which the switchgear PLCs interface with. Whenever a switch attempts to read a sensor value (e.g., current), the S-PLC provides the required value. The sensor values reported by the S-PLC are based on measurements taken at real switchgears that are deployed at a certain radial circuit of our local electric company's network.

The switchgears that are controlled by RTU_07 and RTU_08 in Figure 2 are the tie switches and are initially *disconnected*. The initial state of all the other switchgears is *connected*.

The HMI runs two separate polling threads that monitor the eleven PLCs using the Modbus protocol. The polling threads repeatedly send the same three read-requests to get register values from each one of the PLCs. The PLCs all run the same control logic and expose the same Modbus memory layout: in particular the current value is stored in register #130 and the voltage value is stored in register #131.

In this environment, and indeed in the electric company's real HMI, switchgears are only operated manually from the HMI: If the operator observes a fault, such as current and voltage dropping to zero, she can open or close the switchgears by clicking on the HMI screen. Such operator actions are realized by corresponding Modbus 'write' packets that are sent from the HMI to the proper PLC. Note that in the test lab, the S-PLC reacts to such write events by updating all the subsequent current and voltage values that will be reported to all the relevant PLCs to be consistent with the system state following the operator action.

3 The Attack Tool

3.1 Gaining Network Access

There are many ways for an attacker with network access to place itself in a MITM position. In our attack tool we chose to implement a well known ARP

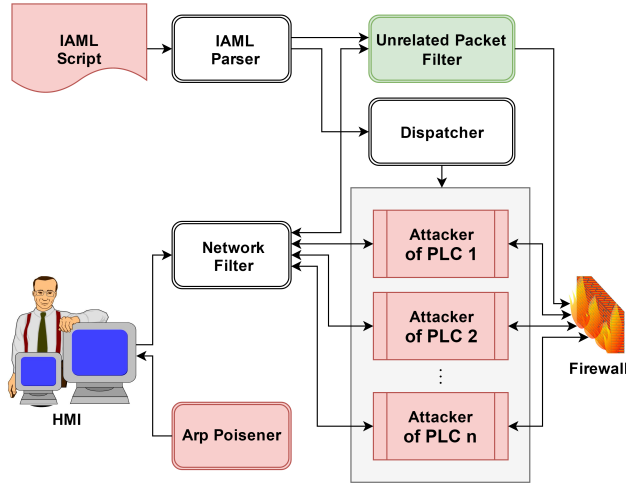


Fig. 4: The architecture of the attack tool

poisoning attack (cf. [2]). Using ARP poisoning obviously makes the attack tool detectable to standard low level Network Intrusion Detection Systems (NIDS). However, our focus is on semantic SCADA-aware anomaly detection, so we assume the attacker is able to bypass the NIDS somehow.

In order to record the attacks’ progress, we placed 2 traffic sniffers, one on each of the VLANs (Sniffer 1 and Sniffer 2 in Figure 3).

The architecture of the attack tool is depicted in Figure 4. It comprises of: an *Arp poisoner* that crafts and sends the spoofed-ARP messages to the HMI, an *Unrelated Packet Filter (UPF)* that just forwards these packets (untouched) to their original destination, a *Dispatcher* that creates a *PLC attacker* (composed of 2 threads, sharing state, per each direction of the traffic) per each PLC that needs to be attacked. An *IAML parser* that accepts an IAML script, parses it and transfers the needed parameters to the *Dispatcher* and the *UPF*, and a *Network filter* that filters proper packets from the HMI VLAN and sends them to the appropriate *PLC attacker*. The Network filter is based on the Pcap.Net .NET wrapper for WinPcap. In our full technical report [21] we describe the attack tool in more details.

3.2 Near-Stateless Manipulation of Modbus

Modifying the message length in the Modbus stream is a relatively “noisy” attack action. SCADA-aware anomaly detection that has even minimal Modbus understanding can flag messages with unusual lengths. Further, injecting messages into a hijacked connection is also detectable by either a network-aware approach (if the message timing is unusual) or by a protocol-aware approach (if the function code or arguments are unusual). Therefore, to demonstrate the power of our attacks we elected to make our attack tool stateless at the transport layer: it does not track or modify the TCP sequence numbers at all. Note that this introduces a significant limitation on the attacker: e.g., it precludes replacing a Modbus “read” query by a “write” query (which is the usual way to

implement commands)—simply because a “write” message is longer, due to the additional written-values payload.

However our tool is not totally stateless: As we shall see, we wish to modify the values reported to the HMI in PLC responses of selected messages. Importantly, the Modbus response messages do not carry the read register addresses; they carry only the read data values, while the register addresses are present in the HMI’s query message. Thus when the attack tool matches a query message of interest, it places that query’s Modbus Transaction ID (TID) into a state variable; when the corresponding response message, with the same TID, is matched on the same connection, the attack tool modifies its content.

4 Semantic Attacks against Electric Distribution SCADA

4.1 Zero Values Deception Attack

Our first deception scenario is illustrated in Figure 5. The attacker invokes the attack twice (first for 18 seconds between 151.8 - 169.8 sec, and second for 14 seconds, between 359.53 - 373.53 sec). During each one of these attacks the attacker simultaneously changes the values of the registers (#130 and #131) that hold the current and the voltage reported by six PLCs to be zeros. The victim PLCs are those controlling the top line (recall Figure 2): RTU 01–06. The bottom graph of Figure 5 shows the true current and voltage values as recorded by Sniffer 2 on the PLC VLAN. The top graph shows the values of the same registers as recorded by Sniffer 1 and observed at the HMI.

This attack causes the HMI to display a view in which RTU_01 - RTU_06 all show zero current - as in Figure 6. This view is consistent with a natural fault on the top line – and causes the operator to implement unneeded remediation steps, that are expensive and possibly harmful. Note that the attack is super-stealthy: SCADA-aware anomaly detection is blind to such an attack, since the attack mimics a natural fault, which is a planned-for scenario and does not, in itself, signify an attack.

4.2 A Multi-stage Attack

Our main attack is more elaborate, and aims to interfere with the operator’s reaction to a fake fault. In this attack scenario we implemented a stealthy multi-stage deception attack, see Figure 7. The attack has two main stages: the first stage is the Zero values deception attack described in the previous section. Looking at the HMI pannel (depicted at Figure 6) the operator is deceived to believe that the top radial circuit is disconnected. This motivates the operator to start disconnecting and reconnecting switchgears according to the operating procedures. This is where the second stage of the attack comes into action. In this stage, whenever the operator issues a switchgear open/close command, the attack tool replaces it with the *opposite command*.

The bottom graph of Figure 7 shows the actual values (of current and voltage) recorded at the different PLCs, and the top graph shows the manipulated values observed at the HMI. The two graphs also indicate the open/close actions by small icons: the top graph shows the intended operator actions at the HMI

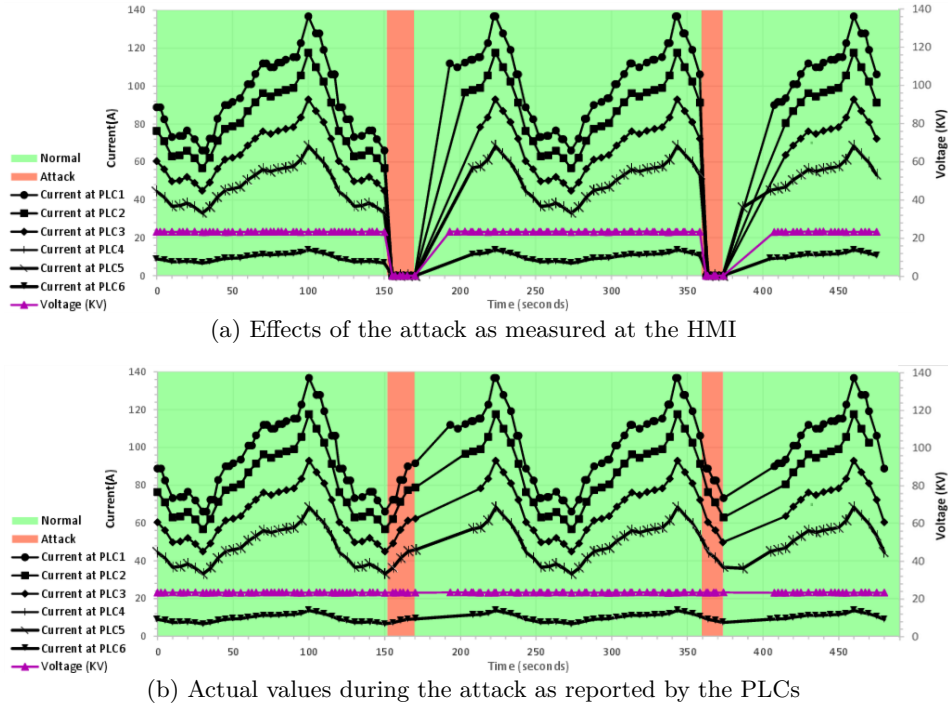


Fig. 5: The *zero values deception attack* against an ICS of an electric distribution system. The attack sends zero values for the current and voltage reported simultaneously by RTU_01 - RTU_06 (denoted here as PLC1–PLC6 respectively), causing the HMI to present a view consistent with a fault on the top line.

(*Open* at time 231.3 and *Close* at time 246.6), and the bottom graph shows the icons for the reversed action received and executed by the PLCs. Note that the attack also fakes current and voltage values that are consistent with the intended operator actions: after the *Open* command the attack starts returning current values computed as “nominal – actual”, where “nominal” is a fixed per PLC constant; this shows the operator that the trouble shooting has some effect. Then after the *Close* command (at time 246.6) the attack tool reports “nominal” current values on all the PLCs—while in fact the circuit is disconnected and customers are experiencing a blackout. Again, note how stealthy the attack is: all Modbus messages are on schedule, using normal functions and arguments, with designed-for, semantically reasonable, data values.

4.3 A Half-duplex Attack

Recall that the zero-values attack of section 4.1 required the attacker to modify values in responses from the PLC to the HMI. In this section we describe an attack that is equivalent to the zero-values attack, except it requires a simpler setup by the attacker: Here the attacker acts as a MITM only on the traffic that flows from the HMI to all the PLCs while the PLC-to-HMI responses are

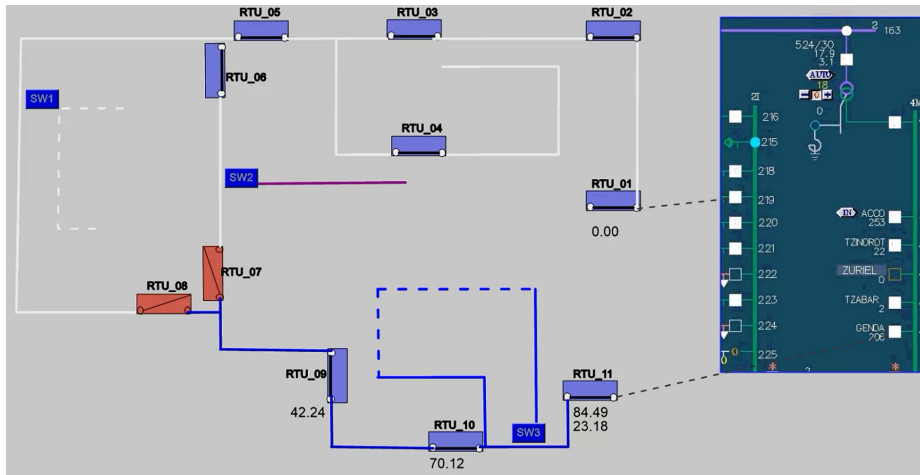


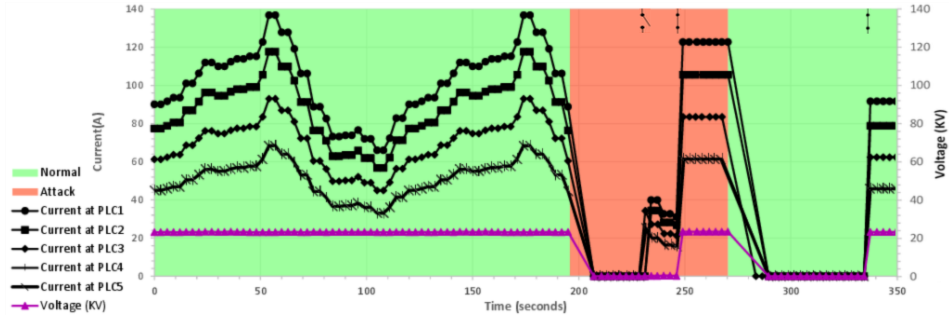
Fig. 6: The HMI panel showing a disconnection in the top radial line. Note the voltage of 0.0KV at RTU_01, the missing current value at RTUs 02–06, and the white color of the line. In such a scenario the operator would typically open the switchgear at RTU 01 and close the ties at RTU 07, 08 to attempt to supply the top line from the bottom line.

unaltered. Hence the attacker is unable to directly set data values to zero in PLC responses. To achieve an attack in this scenario, query messages (from the HMI to the PLC) reading the current and voltage values are modified to access registers whose addresses are shifted by 1. As a result, during the attack, the shifted register-values sent by the PLCs are interpreted at the HMI as the values of the corresponding preceding registers—the HMI interprets the voltage value as the current, and the content of register #132 (which is 0) is interpreted as voltage-value, again fooling the operator into deducing that a fault occurred. We omit the graphs.

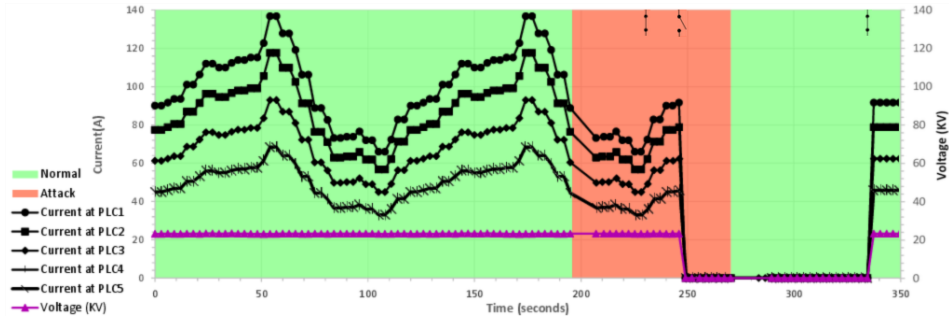
Note that this attack can be detected by a protocol-aware anomaly detection [18,24] as long as the detector is located at the network segment where the PLCs reside, since the attacker modifies the accessed register addresses, which creates unknown symbols in the GW model [18].

4.4 The ICS Attack Markup Language

We defined a new formalism for specifying a concrete execution of an attack, an ICS Attack Markup Language which we called an IAML. IAML enables planning and implementation of a multi-stage, multi-PLC, simultaneous attacks on an ICS without any a priori programming knowledge. It uses a modular approach, where a module represents an attack that changes a certain type of packet under certain conditions. Modules can be linked together to create multi-stage attacks. The relationships among modules are specified through the conditions and changes of local and global stages. IAML is accompanied by a library of predicates, which function as a vocabulary to describe the properties of attack modules and stages. In our full technical report [21] we describe the details of IAML and include complete usage example.



(a) Effects of the attack as measured at the HMI



(b) Actual values during the attack as reported by the PLCs

Fig. 7: The multi-stage attack – the attack starts at time 195.8 with a zero-values attack. The second stage executing the “opposite operation” attack starts at 231.3 and ends at 270.4, with operator commands at times 231.3 and 246.6. The icons at the top edge of the graphs indicate the *Open* and *Close* commands to the switchgears.

5 Related Work

5.1 Attacks against ICS

Digital attacks that cause physical destruction of equipment do occur [19]. Most recently, cyber-attacks on SCADA systems controlling electrical distribution have caused wide-spread blackouts in Ukraine [27,28]. Perhaps most well known is the attack on an Iranian nuclear facility in 2010 (Stuxnet) to sabotage centrifuges at a uranium enrichment plant. The Stuxnet malware [15,26] worked by changing centrifuge operating parameters in a pattern that damaged the equipment – while sending normal status messages to the HMI. In 2014, the German Federal Office for Information Security announced a cyber attack at an unnamed German steel mill, where hackers manipulated and disrupted control systems to such a degree that a blast furnace could not be properly shut down, resulting in “massive”-though unspecified-damage [12].

Byres et al. [7] describe the application of the attack tree methodology to SCADA communication systems based on the common Modbus protocol stack. The authors identify eleven possible attacker goals with their respective technical difficulty, possible severity of impact and likelihood of detection. In particular they noted that an attacker can perform a Man-In-The-Middle (MITM) attack

between a PLC and HMI and “feed” the HMI with misleading data, allegedly coming from the PLC – which is what we implemented.

In 2009 Fovino et al. [16] showed that malware was able to disrupt or even seize control of vital sensors and actuators. Semantic attack scenarios on ICSs are described by [17,31] for a system with a pipe in which high pressure steam or fluid flows. The pressure is regulated by two valves. An attacker capable of sending packets to the PLCs can force one valve to complete closure, and force the other to open. Each of these ICS commands is perfectly legal when considered individually, however when sent in an abnormal order they can cause a ‘water hammer’ and bring the system to a critical state.

5.2 Anomaly Detection in ICS

Surveys of techniques related to learning and detection of anomalies in critical control systems can be found in [3,4,10]. While most of the current commercial network intrusion detection systems (NIDS) are signature-based, i.e., they recognize an attack when it matches a previously defined signature, anomaly-based NIDS “are based on the belief that an intruder’s behavior will be noticeably different from that of a legitimate user” [33]. All anomaly detection approaches below do not distinguish between malicious events and faulty events — and none of them is able to detect our attacks.

Network–Aware Detection. Basic anomaly detection models for SCADA systems only consider network and OS-level events. Yang et al. [38] used an Auto Associative Kernel Regression (AAKR) model coupled with the Statistical Probability Ratio Test (SPRT) and applied them on a SCADA system. The model used numerous indicators representing network traffic and hardware-operating statistics to predict the ‘normal’ behavior.

Barbosa et al. [5,6] analyzed SCADA traces they collected at two different water treatment and distribution facilities. They concluded that SCADA traffic presents remarkably regular time series, due to the fact that the majority of the traffic sources generate data in a periodical fashion. They selected only the high energy frequencies for the anomaly detection phase.

Protocol–Aware Detection. More advanced anomaly detection systems rely on deep-packet-inspection and consider the ICS control protocol’s meta-data, modeling command sequences, and argument addresses.

Model-based anomaly detection for SCADA systems, and specifically for Modbus traffic, was introduced by Cheung et al. [11]. They designed a multi-algorithm intrusion detection appliance for Modbus/TCP with pattern anomaly recognition, Bayesian analysis of TCP headers and stateful protocol monitoring, complemented with customized Snort rules [35]. In subsequent work, Valdes et al. [37] incorporated adaptive statistical learning methods into the system to detect for communication patterns among hosts and traffic patterns.

Goldenberg & Wool [18] developed a model-based approach (the GW model) for Network anomaly detection based on the normal traffic pattern in Modbus SCADA networks using a Deterministic Finite Automata (DFA) to represent the cyclic traffic. The SCADA messages are modeled both in isolation and also by

their sequence order. Subsequently, Kleinmann et al. [22,23] demonstrated that a similar methodology is successful also in SCADA systems running the Siemens S7 protocol.

Caselli et al. [9] proposed a methodology to model sequences of SCADA protocol messages as Discrete Time Markov Chains (DTMCs). Based on data from three different Dutch utilities the authors found that only 35%-75% of the possible transitions in the DTMC were observed. This strengthens the observations of [5,18,22] of a substantial sequentiality in the SCADA communications. However, unlike [18,22] they did not observe clear cyclic message patterns. The authors hypothesized that the difficulties in finding clear sequences is due to the presence of several threads in the HMI's operating system that multiplex requests on the same TCP stream.

Kleinmann et al. [23,24,25] introduced a modeling approach for multiplexed SCADA streams, using *Statechart DFAs*: the *Statechart* includes multiple DFAs, one per cyclic pattern. Each DFA is built using the learning stage of the GW model. Following this model, incoming traffic is de-multiplexed into sub-channels and sent to the respective DFAs.

Process-Aware Detection. These anomaly detection methods are based on process invariants: mathematical relationships among physical properties of the process controlled by the PLCs. Several publications [8,30,36] explain that an IDS that models only the protocol meta-data (commands and arguments) is not sufficient, since attacks can be mounted using legitimate control commands, but with attacker-selected data values. To combat such attacks they suggest modeling both the physical process and the continuous control function. Based on measurements of the state of the process, the models predict the control's response and its effect on the state, and flag deviations from the predicted state.

Fovino et al. [17] use detailed knowledge of the industrial process' control to generate a system virtual image representing the PLCs of a monitored system. The virtual image is updated using a periodic active synchronization procedure and via a feed generated by the intrusion detection system (i.e., known intrusion signatures).

Hadziosmanovic et al. [20] used the logs generated by the control application running on the HMI to detect anomalous patterns of user actions on process control application.

Lin et al. [29] combine system knowledge of both the control network (extracting control commands from SCADA network packets) and the physical infrastructure in power grid (obtaining measurements from sensors in substations) to help IDS to estimate execution consequences of control commands, thus to reveal attacker's malicious intentions. The authors claimed that their semantic analysis provides reliable detection of malicious commands with a small amount of analysis time.

Erez et al. [14] developed an anomaly detection system that detects irregular changes in SCADA control registers' values. The system is based on an automatic classifier that identifies several classes of PLC registers (Sensor, Counter and Constant registers). Parameterized behavior models were built for each class. In

its learning phase, the system instantiates the model for each register. During the enforcement phase the system detects deviations from the model.

Mo et al. [32] as well as Pasqualetti et al. [34] investigated the detection and prevention of deception and replay attacks. They concluded that certain types of attacks are undetectable by using their attack models.

6 Conclusions and Counter Measures

This work presented a class of semantic network-based attacks against SCADA systems which are undetectable by both protocol-aware and process-aware anomaly detection. After hijacking the communication channels between the HMI and PLCs, our attacks cause the HMI to present a fake view of the industrial process, deceiving the human operator into taking manual actions. Our most advanced attack also manipulates the operator's commands reversing their semantic meaning while causing the HMI to present a view that is consistent with the attempted operator directions. The attacks are totally stealthy since the message sizes and timing, the command sequences, and the data values of the ICS's state all remain legitimate. They appear as natural fault conditions that the SCADA system is designed for, and the human operator is trained to handle.

We implemented and tested several attack scenarios in the realistic test lab of our local electric company. We developed a real-time security assessment tool, that can simultaneously manipulate the communication to multiple PLCs, and cause the HMI to display a coherent system-wide fake view. Our tool is configured with a new IAML language we designed. Our multi-stage semantic attacks all successfully fooled the operator and brought the system to states of blackout and possible equipment damage.

We argue that current intrusion detection and anomaly detection systems are fundamentally unable to detect the stealthy deception attacks we described. In fact, once the attacker is positioned as MITM, the traffic at both the HMI and the PLC sides looks perfectly normal—because it is perfectly normal. In our opinion the only real countermeasure against such attacks is to secure the communication channel via cryptographic means. E.g., by adding data integrity protections such as digital signatures or message authentication codes to block the attacker's ability to modify packets. Nevertheless, we believe that ongoing research into anomaly detection for ICS is still very valuable: with such systems in place, the attacker is restricted to *only* mount super-stealthy deception attacks like ours, and cannot mount simpler and more direct attacks without risk of detection.

References

1. Final report on the August 14, 2003 blackout in the United States and Canada: Causes and recommendations. U.S.-Canada Power System Outage Task Force, U.S. Secretary of Energy and Minister of Natural Resources Canada (April 2004)
2. Abad, C.L., Bonilla, R.I.: An analysis on the schemes for detecting and preventing arp cache poisoning attacks. In: 27th International Conference on Distributed Computing Systems Workshops, ICDCSW'07. pp. 60–60. IEEE (2007)
3. Alcaraz, C., Cazorla, L., Fernandez, G.: Context-awareness using anomaly-based detectors for smart grid domains. In: International Conference on Risks and Security of Internet and Systems. vol. 8924, pp. 17–34. Springer International Publishing, Trento (2014)
4. Atassi, A., Elhajj, I.H., Chehab, A., Kayssi, A.: The State of the Art in Intrusion Prevention and Detection, chap. 9: Intrusion Detection for SCADA Systems, pp. 211–230. Auerbach Publications (January 2014)
5. Barbosa, R., Sadre, R., Pras, A.: A first look into SCADA network traffic. In: IEEE Network Operations and Management Symposium (NOMS). pp. 518–521 (April 2012)
6. Barbosa, R., Sadre, R., Pras, A.: Towards periodicity based anomaly detection in scada networks. In: 17th IEEE Emerging Technologies Factory Automation (ETFA). pp. 1–4 (Sept 2012)
7. Byres, E.J., Franz, M., Miller, D.: The use of attack trees in assessing vulnerabilities in SCADA systems. In: Proceedings of the International Infrastructure Survivability Workshop (2004)
8. Cárdenas, A.A., Amin, S., Lin, Z.S., Huang, Y.L., Huang, C.Y., Sastry, S.: Attacks against process control systems: risk assessment, detection, and response. In: Proceedings of the 6th ACM symposium on information, computer and communications security. pp. 355–366. ACM (2011)
9. Caselli, M., Zambon, E., Kargl, F.: Sequence-aware intrusion detection in industrial control systems. In: Proceedings of the 1st ACM Workshop on Cyber-Physical System Security. pp. 13–24. New York, NY, USA (2015), <http://doi.acm.org/10.1145/2732198.2732200>
10. Chen, C.M., Hsiao, H.W., Yang, P.Y., Ou, Y.H.: Defending malicious attacks in cyber physical systems. In: IEEE 1st International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA), 2013. pp. 13–18 (Aug 2013)
11. Cheung, S., Dutertre, B., Fong, M., Lindqvist, U., Skinner, K., Valdes, A.: Using model-based intrusion detection for SCADA networks. In: Proceedings of the SCADA Security Scientific Symposium. pp. 127–134 (2007)
12. De Maizière, T.: Die Lage Der IT-Sicherheit in Deutschland 2014. The German Federal Office for Information Security (2014), https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2014.pdf?__blob=publicationFile, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2014.pdf?__blob=publicationFile
13. Dolev, D., Yao, A.C.: On the security of public key protocols. Tech. rep., Stanford, CA, USA (1981)
14. Erez, N., Wool, A.: Control variable classification, modeling and anomaly detection in modbus/tcp scada systems. International Journal of Critical Infrastructure Protection 10, 59–70 (2015)

15. Falliere, N., Murchu, L., Chien, E.: W32. stuxnet dossier. White paper, Symantec Corp., Security Response (2011)
16. Fovino, I.N., Carcano, A., Masera, M., Trombetta, A.: An experimental investigation of malware attacks on SCADA systems. *International Journal of Critical Infrastructure Protection* 2(4), 139–145 (2009), <http://www.sciencedirect.com/science/article/pii/S1874548209000419>
17. Fovino, I., Carcano, A., De Lacheze Murel, T., Trombetta, A., Masera, M.: Modbus/DNP3 state-based intrusion detection system. In: 24th IEEE International Conference on Advanced Information Networking and Applications (AINA). pp. 729–736. Ieee (2010)
18. Goldenberg, N., Wool, A.: Accurate modeling of Modbus/TCP for intrusion detection in SCADA systems. *International Journal of Critical Infrastructure Protection* 6(2), 63–75 (2013), <http://www.sciencedirect.com/science/article/pii/S1874548213000243>
19. Gorman, S.: Electricity grid in U.S. penetrated by spies. *The Wall Street Journal* p. A1 (April 8th 2009), <http://www.wsj.com/articles/SB123914805204099085>
20. Hadziosmanovic, D., Bolzoni, D., Hartel, P.H., Etalle, S.: MELISSA: Towards automated detection of undesirable user actions in critical infrastructures. In: Proceedings of the European Conference on Computer Network Defense, EC2ND, Gothenburg, Sweden. pp. 41–48. USA (September 2011)
21. Kleinmann, A., Amichay, O., Wool, A., Tenenbaum, D., Bar, O., Lev, L.: Stealthy deception attacks against scada systems. In: arXiv:1706.09303 [cs.CR] (June 2017)
22. Kleinmann, A., Wool, A.: Accurate modeling of the siemens S7 SCADA protocol for intrusion detection and digital forensic. *JDFSL* 9(2), 37–50 (2014), <http://ojs.jdfs1.org/index.php/jdfs1/article/view/262>
23. Kleinmann, A., Wool, A.: A statechart-based anomaly detection model for multi-threaded scada systems. In: International Conference on Critical Information Infrastructures Security. pp. 132–144. Springer (2015)
24. Kleinmann, A., Wool, A.: Automatic construction of statechart-based anomaly detection models for multi-threaded SCADA via spectral analysis. In: Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy. pp. 1–12. CPS-SPC '16, ACM, New York, NY, USA (2016), <http://doi.acm.org/10.1145/2994487.2994490>
25. Kleinmann, A., Wool, A.: Automatic construction of statechart-based anomaly detection models for multi-threaded industrial control systems. *ACM Transactions on Intelligent Systems and Technology (TIST)* 8(4) (February 2017)
26. Langner, R.: Stuxnet: Dissecting a cyberwarfare weapon. *Security & Privacy, IEEE* 9(3), 49–51 (2011)
27. Lee, R.M., Assante, M.J., Conway, T.: Analysis of the cyber attack on the Ukrainian power grid. Tech. rep., SANS E-ISAC (March 18 2016), https://ics.sans.org/media/E-SAC_SANS_Ukraine_DUC_5.pdf
28. Liang, G., Weller, S.R., Zhao, J., Luo, F., Dong, Z.Y.: The 2015 Ukraine blackout: Implications for false data injection attacks. *IEEE Transactions on Power Systems* (2016)
29. Lin, H., Slagell, A., Kalbarczyk, Z., Sauer, P.W., Iyer, R.K.: Semantic security analysis of SCADA networks to detect malicious control commands in power grids. In: Proceedings of the first ACM workshop on Smart energy grid security. pp. 29–34. ACM (2013)
30. Liu, Y., Ning, P., Reiter, M.K.: False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security (TISSEC)* 14(1), 13 (2011)

31. Marsh, R.T.: Critical foundations: Protecting America's infrastructures - the report of the president's commission on critical infrastructure protection. Tech. rep., President's Commission on Critical Infrastructure Protection (October 1997)
32. Mo, Y., Kim, T.H.J., Brancik, K., Dickinson, D., Lee, H., Perrig, A., Sinopoli, B.: Cyber-physical security of a smart grid infrastructure. *Proceedings of the IEEE* 100(1), 195–209 (2012)
33. Mukherjee, B., Heberlein, L.T., Levitt, K.N.: Network intrusion detection. *Network*, *IEEE* 8(3), 26–41 (1994)
34. Pasqualetti, F., Dörfler, F., Bullo, F.: Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control* 58(11), 2715–2729 (2013)
35. Roesch, M.: Snort - lightweight intrusion detection for networks. In: *Proceedings of the 13th USENIX Conference on System Administration*. pp. 229–238. LISA '99, USENIX Association, Berkeley, CA, USA (1999), <http://dl.acm.org/citation.cfm?id=1039834.1039864>
36. Urbina, D.I., Giraldo, J.A., Cardenas, A.A., Tippenhauer, N.O., Valente, J., Faisal, M., Ruths, J., Candell, R., Sandberg, H.: Limiting the impact of stealthy attacks on industrial control systems. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. pp. 1092–1105. ACM (2016)
37. Valdes, A., Cheung, S.: Communication pattern anomaly detection in process control systems. In: *IEEE Conference on Technologies for Homeland Security (HST)*. pp. 22–29 (2009)
38. Yang, D., Usynin, A., Hines, J.: Anomaly-based intrusion detection for SCADA systems. In: *5th Intl. Topical Meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies*. pp. 12–16 (2006)