Advanced Persistent Threats and Zero-Day Exploits in Industrial Internet of Things

Ioannis Stellios, Panayiotis Kotzanikolaou, and Mihalis Psarakis

Abstract Manufacturing industry, electricity networks, supply chain, food production and water treatment plants have been heavily depended on Industrial Automation and Control (IAC) Systems. Integration of Information and Communication Technology (ICT) played a significant role in the evolution of these systems. New emerging trends and technologies, such as Internet-of-Things (IoT) interact with traditional, isolated IAC systems. Sectors such as manufacturing, electric grids, pharmaceuticals, and water treatment facilities incorporate part of these "smart" technologies in order to increase efficiency, performance and reduce production costs. But despite of its benefits, interconnectivity between smart and legacy IAC systems also creates complex interdependencies, which in turn, make imperative the need for more safety and security countermeasures. This rapid evolution has also affected greatly the threat landscape. In order to comprehend this radical change we present and analyze recent, well documented attacks that target mission critical IAC systems, which incorporate Industrial IoT technologies. In particular, we focus on highly profiled, sophisticated attacks against interconnected automation and monitoring field devices, related software platforms and systems (e.g., Programmable Logical Controllers – PLCs, industrial robots) installed on industrial facilities and smart grid generation, transmission and distribution networks and systems.

Keywords Security · Privacy · IIoT · Advanced persistence threats · Cyberattacks

I. Stellios (✉) · P. Kotzanikolaou
SecLab, Department of Informatics, University of Piraeus, Piraeus, Greece
e-mail: jstellios@unipi.gr; pkotzani@unipi.gr

M. Psarakis
ESLab, Department of Informatics, University of Piraeus, Piraeus, Greece
e-mail: mpsarak@unipi.gr

C. Alcaraz (ed.), *Security and Privacy Trends in the Industrial Internet of Things*,
Advanced Sciences and Technologies for Security Applications,
https://doi.org/10.1007/978-3-030-12330-7_3

1 Introduction

Industrial Control Systems (ICSs) have been constantly evolving in terms of efficiency, productivity, quality, manageability and operational security. Rapid evolution in computer science also affected ICS: Supervisory Control and Data Acquisition (SCADA) systems enabled manufacturers to remotely control complicated production lines via Human-Machine Interfaces (HMIs) placed in central management stations. Initially developed in the early 1950s, first generation of ICSs consisted mainly of Wide Area Networks (WANs) used to communicate with Remote Terminal Units (RTUs). The second generation ICSs utilized smaller and cheaper devices that were connected via Local Area Networks (LANs) whereas in the third generation interconnectivity with third-party peripherals was introduced.

SCADA systems were mainly built from customized hardware, controlled with the use of specialized software, utilized domain specific or proprietary network protocols and, until recently, these systems were, mostly, isolated from the outside world. But the constant need for improving efficiency, interoperability, manageability and production cost reduction introduced the fourth generation of SCADA systems which included, among others, new evolutionary technologies such as Industrial Internet of Things (IIoT) technologies, thus widening significantly their attack surface [3]. Modern SCADA systems that utilize relative IIoT technologies have been widely adopted in almost every critical aspect of our modern lifestyle, ranging from manufacturing industry, power generation, transmission and distribution, water treatment and reservoir, intelligent transportation and smart city/building systems. Real cyberattacks that utilize IIoT technologies have been on the rise throughout the world during the recent decade [7, 27, 34, 35].

High impact attack scenarios usually involve refined exploitation methods named after the term *Advanced Persistence Threats* (APTs). The term *advanced*, mainly corresponds to the fact that the adversaries utilize sophisticated attack techniques, that take advantage the full spectrum of publicly available exploits against well-known vulnerabilities, as well as custom payloads and delivery methods (*zero-day*) depending on the target's response. The term *persistence*, corresponds to the continuous interaction between the adversary and the compromised systems until the goal of the attack is achieved. In short, APTs are well planned, stealthy attacks, that use advanced exploitation techniques against a particular target, designed to be effective for a large period of time.

In a recent survey paper [44], we presented how relative Internet of Things (IoT) technologies, applied in different sectors (industry, smart grid, intelligent transportation systems, medical and smart home), can be utilized by adversaries to create new, hard to identify attack paths in order to launch high impact attacks against critical infrastructures and services. In order to further understand the threat landscape of IIoT ecosystem, we herein describe the different phases that take place in an APT attack scenario and analyze the applicable exploitation techniques used on each phase for several real/PoC attacks. In particular, we analyze APT attack scenarios against Internet facing field devices (e.g., PLCs, industrial robots) [17, 30, 42] as

well as indirect attacks that mostly utilize spear-phishing techniques and existing connectivity paths between corporate and industrial networks [16, 25, 34]. The latter include recent, high profile APT attack scenarios against smart grid's transmission [18] and distribution [7] IIoT devices and systems.

The outline of the chapter is as follows: Sect. 2 presents an overview of ICS software, protocols and architectures. Then, Sect. 3 categorizes and presents *zero-day* exploits found on HMI software [5], some of which can/have be used to APT attack scenarios. Furthermore, Sect. 4 defines the basic phases of an APT attack, which in turn, are used to analyze several attack scenarios against IIoT field devices and smart grid SCADA networks in Sects. 4 and 5, respectively. Finally, Sect. 6 presents an overview of the attack scenarios' characteristics and proposes security countermeasures and best practices.

2 SCADA Related Protocols and Architectures

Due to the diversity of SCADA systems there is plethora of open industrial network communication protocols, such as Modbus/TCP, Distributed Network Protocol 3 (DNP3), Profibus, IEC-104, DeviceNET, ControlNET, Ethernet/IP, wireless IEEE 802.15.4x [12, 20, 28, 45] as well as proprietary ones. Since most of these protocols have been designed with no security features in mind they are susceptible to cyberattacks such as passive/active information sniffing, message spoofing and command injection [15].

Furthermore, in order to reduce production costs, manufacturers utilize off-the-shelf hardware to build industrial field devices, in which, they incorporate vulnerable communications protocols (e.g., 802.15.4x). Most of these devices are capable of communicating directly to the Internet via embedded network protocols (e.g., 6LoWPAN [33]) and are equipped with software that facilitates the integration of IoT technologies such as commercial cloud-computing services. The latter are mainly used to improve data accessibility, reduce the operational costs and increase flexibility, optimization and scalability. On the other hand, many of these cloud platforms [46] come with a plethora of vulnerabilities [38] on both system software and Application Programming Interfaces (APIs) that adversaries may use as an enabler to attack mission critical SCADA systems.

Modern SCADA systems consist (see Fig. 1) of a large number of Intelligent Electronic Devices (IEDs), such as sensors, actuators (e.g., circuit breakers), smart meters, robotics and motors that are controlled through Programmable Logical Controllers (PLCs) and Remote Terminal Units (RTUs). Both RTUs/PLCs utilize wireless/wired network interfaces and protocols in order to communicate to each other and to Master Terminal Units (MTUs). PLCs/RTUs are used to acquire a device's status, (e.g., valve open/closed), read and re-transmit operational variables (e.g., pressure, voltage) as well as control industrial equipment (e.g., robotics) by sending commands [23].

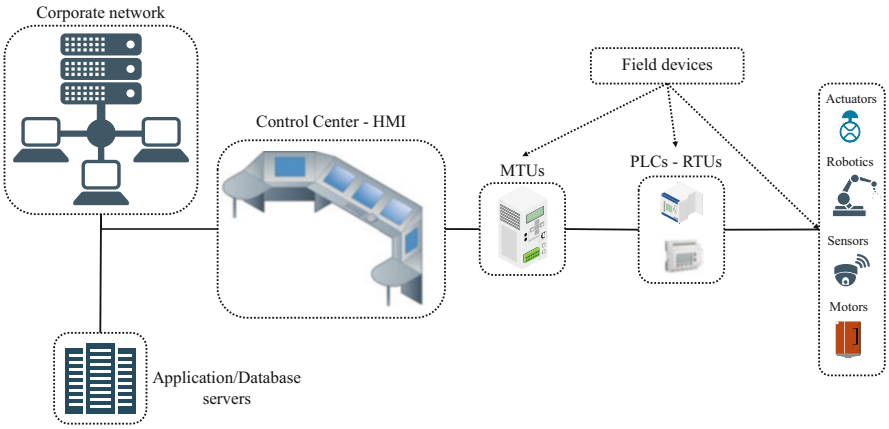

Fig. 1 A typical SCADA architecture

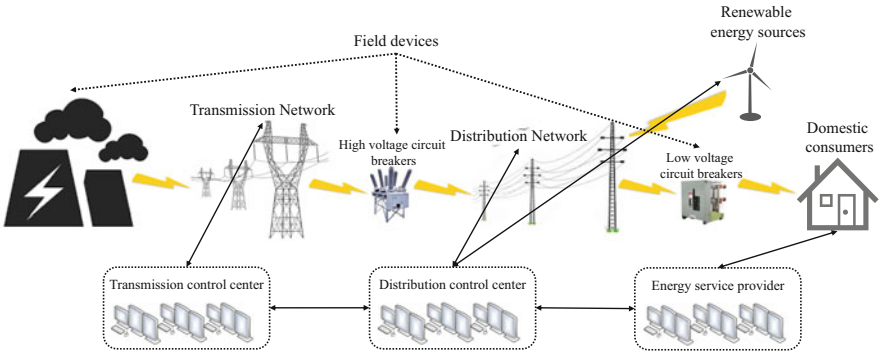

Fig. 2 A typical smart grid architecture

Typically industrial SCADA systems rely on a predefined structure. RTUs are usually interconnected using a hierarchical model to MTUs which, in turn, are connected to Command-and-Control (C&C) centers. Then, using software applications, human operators can administer, regulate and remotely control entire production lines via graphical HMIs from PC type workstations. Integration of IIoT technologies in SCADA systems, requested that C&C rooms must also be connected to the company's corporate network and/or to the Internet. Furthermore, international corporations also interconnect several C&C centers to master regional stations.

Smart grid adopts a hierarchical model similar to the industrial one, but far more complex. Its cornerstone is mainly the generation systems that produce electricity (see Fig. 2). Then, the electric current is transmitted using the backbone of smart grid, consisting of the transmission network and its substations. From there, the electricity is delivered to both home and industrial consumers through the distribution network. The latter mainly consists of the Advanced Metering Infrastructure (AMI) [31] and domestic renewable energy sources (e.g., solar panels).

Energy optimization and control are achieved with the use of Energy Management Systems (EMS) located in strategical places throughout the distribution network, whereas independent systems operators are responsible to manage the electricity flow between service providers and customers [32].

3 Zero-Day Exploits on Human-Machine Interface Applications

HMI software is considered to be the most critical application in IIoT ecosystem since it is installed on control rooms' workstations and its main purpose is to administer mission critical SCADA systems. Compromising an HMI system may lead to a series of attacks ranging from information gathering, deactivation of notification systems (e.g., alarms), notifications to operators up to physically damage industrial equipment. To make things worse, HMI vendors do not always enforce security best practices on the controlling software, thus focusing only on the managed devices. In this section we present the findings of an extensive research conducted by the *Zero Day Initiative* (ZDI) team of *Trend Micro* security company that took place throughout a two-year period (2015–2016) [5] and successfully identified 250 *zero-day* vulnerabilities on HMI applications. During the disclosure process, researchers observed that the average time period for the vendors to release a corresponding patch of a *zero-day* exploit averaged to 150 days. This actually meant that mission-critical SCADA systems were vulnerable for almost five months before a patch was available from software vendors. The exploitation techniques were classified into 4 main categories: (i) Memory corruption, (ii) Credential harvesting, (iii) Insecure installation, authentication and authorization procedures and (iv) Code injection. These exploitation techniques which can be used in various APT attack scenarios are described in detail in the following sections.

3.1 Memory Corruption

Memory corruption issues accounted for the 20% of the total number of vulnerabilities found. The majority were stack/heap-based buffer overflows [13] and out-of-bounds read/write vulnerabilities. In a particular vendor, the software Advantech WebAccess HMI Solution was proved to have a vulnerable `sprintf` function and no protection mechanisms such as stack cookies, Address Space Layout Randomisation (ASLR) [40] and SafeSEH [19]. Due to the absence of ASLR protection an adversary needs only to overwrite the return address to a controlled Return Oriented Programming (ROP) chain, in order to execute malicious code with elevated privileges. Even though the vendor issued a large number of patches these corrected only specific issues and did not address the problem globally or replaced other problematic functions.

3.2 Credential Harvesting

Vulnerabilities found in credential management represented the 19% of the overall vulnerabilities found. These included the use of hard-coded passwords as well as insecure storage and/or protection of passwords (e.g., stored clear text/with reversible encryption algorithms). Furthermore, in a particular case study of General Electric (GE) MDS PulseNET, a software that is used to monitor industrial equipment and communication networks deployed in energy, water, and waste water sectors globally, they managed to identify an embedded account with full privileges apart from the administrator and user account (CVE-2015–6456 [2]). By utilizing *HeidiSQL* tool they managed to extract the `ge_support` account as well as the password's MD5 hash value (`PulseNET`). Notably, even after a successful logging process of the discovered account its username did not appear in the user management screen.

3.3 Insecure Installation, Authentication and Authorization Procedures

This category represents the 23% of the total vulnerabilities found, including unencrypted communications, such as the transmission in plaintext of sensitive information (e.g., usernames or passwords), as well as vulnerable ActiveX controls which where marked as 'safe'. In another case study concerning Siemens SINEMA Server, a network management software for monitoring and diagnostics, a misconfiguration allowed standard authenticated users to have full access to Windows sensitive system folders (CVE-2016–6486). In addition, the binary code used to start the SINEMA service run at local system level thus allowing an adversary with local access to the workstation, to replace the legitimate binary code with a malicious one. Then, triggering a reboot allowed the adversary to execute the malicious code with system privileges.

In another case study considering Advantech WebAccess, a cross-platform user interface management based in HTML5, an authenticated user was able to retrieve the passwords of other platform's users including the administrator.

3.4 Code Injection

Although Structured Query Language (SQL) type and Operating System (OS) command injections occupy a small fraction (9%) of the overall vulnerabilities discovered, the impact of such threats on HMI systems is considered to be very high, especially those injections that apply to domain-specific languages for SCADA software solutions. In a particular case study, 'Cogent DataHub', a real time

visualization software for complex SCADA systems, was evaluated. The application incorporates Gamma script language, a domain-specific language that has built-in features and functions for SCADA systems. Cogent DataHub also includes a database, that resides in server's memory providing interchange of data for Object Linking and Embedding (OLE) for Process Control (OPC) and other Windows applications. Researchers discovered that it is possible for an attacker to take advantage a flaw in the `EvalExpression` method of Gamma script language and enable the insecure processing mode in the Asynchronous JavaScript and XML (AJAX) web server, resulting in a remote code execution on the server.

4 APT Attack Scenarios on Industrial IoT Field Devices

Industrial systems usually attract well-funded, high-skilled and strongly motivated adversaries that seek to gain substantial economic profit (e.g., cybercriminals) or to disrupt a nation's Critical Infrastructures (CIs) (e.g., nation state adversaries). These attacks are considered of high impact due to the effect that SCADA systems have on our every day life. APTs' attack vectors, that exploit relative IIoT technologies, may include but not limited to, the following basic phases:

- **Reconnaissance/Data gathering and host discovery phase** Gathering valuable information regarding corporates' employees and executives, enumerating the targeted company's web presence and compromising corporate email accounts to launch a series of spear phishing campaigns [7, 18, 34] are considered to be the most prevailing methods in the early stages of an APT attack scenario. In addition, web search engines (e.g., Shodan) are also used to locate web exposed industrial equipment that then can be enumerated for vulnerabilities before the exploitation/ initial infection phase begins [5, 17, 30, 42].
- **Initial infection phase** Since corporate users must communicate with the outside world and, at the same time, are usually connected (directly or indirectly) to mission critical industrial control systems are considered to be the prime target for adversaries. This is usually accomplished by launching spear-phishing campaigns, which include the process of sending malware infected, office documents and malicious web links from hijacked corporate/legitimate accounts (e.g., [7, 18]). Another more direct approach is to exploit the web interfaces of modern industrial equipment, that utilize IoT enabling technologies, in order to be able to be operated, managed and updated remotely (e.g., [5, 17, 30, 42]). In addition, it is common practice for manufacturers as well as companies that provide technical support to industrial equipment, to distribute essential software components and/or updates (e.g., IIoT devices' firmware and relative management software) via vulnerable websites and unsecured methods (e.g., HTTP), with devastating consequences on IIoT ecosystem [34]. Finally, off-line exploiting techniques can be also used, as presented in [16, 25].

- **Establish and maintain remote access** Asynchronous communication, data masquerade and encryption, Intrusion Prevention/Detection System (IDS/IPS) evasion and privilege escalation are some of the techniques used in order to achieve stealthiness. To ensure access persistence, payloads are made so as to withstand power loss/reboot processes and equipped with auxiliary communication modules for redundancy.
- **Lateral movement and propagation phase** In APT attack scenarios, adversaries utilize several enumeration and pivoting techniques (e.g., probing nearby systems for open ports, connect to default drive shares, spread to different network segments) in order to locate and exploit other mission critical vulnerable ICT equipment such as control rooms' workstations and IIoT devices.
- **Remote control and device manipulation** Attackers must incorporate a series of well established and new industrial network protocols in order to remotely communicate and ultimately take control the IIoT device(s). The payloads installed on IIoT devices must be able to run with minimum resources and hide their code so as to avoid detection from machine operators.
  Functionality plays an essential role when designing payloads that target industrial equipment, since, adversaries must be able to issue arbitrary commands and even control all functions and features of the IIoT device/system. The latter enables adversaries to lock out legitimate operators thus preventing them from responding to the threat accordingly [7].

In many cases of APT attack scenarios the adversaries include payloads that are used to renter the devices and systems affected unusable and/or hide their footprints (e.g., [7, 18, 34]).

4.1 Stuxnet

The most well-known APT attack against SCADA systems, that managed to infect the software of at least 14 industrial sites in Iran, including a uranium enrichment plant, is considered to be *Stuxnet* [16, 25]. This 500 KB computer worm utilized four 0-day vulnerabilities to compromise two digital certificates, inject code into industrial control systems and hide the code from the operator. Its main goal was to sabotage industrial facilities by reprogramming network connected field devices to operate out of their specified boundaries. Although it required a victim to unintentionally install it in the network (e.g., via an infected external usb drive) the code was extremely stealth and sophisticated. Its main target was Siemens Steps7 software used for controlling industrial centrifuges. The worm operated autonomously by using self-replicating techniques to spread out to the internal network. It was equipped with advanced exploitation payloads that targeted Windows operating machines used to control specialized industrial equipment, thus enabling the adversaries to spy on the infected devices and even cause the destruction of the fast-spinning centrifuges. Although the authors of Stuxnet have

not been officially identified, the sophistication of the discovered code indicates the involvement of nation state adversaries [6]. In particular the vector of the attack can be described as follows:

1. **Reconnaissance phase** Nation state adversaries create malware infected usb drives which, then, place in strategically chosen sites (e.g., at the Iran's industrial sites' entrances) so as to allure industrial workers to plug them to their computers.
2. **Initial infection phase** The worm is designed to infect Windows operating machines by taking advantage of auto-execution features in removable drives (Microsoft Windows Shortcut LNK/PIF Files Automatic File Execution Vulnerability – Bugtraq ID 41732). Then, it takes advantage of two *zero-day* Windows vulnerabilities to perform privilege escalation. In order to avoid detection, it utilizes a rootkit to hide its binaries so as to evade antivirus products.
3. **Lateral movement and propagation phase** Module *Export 22* was the main payload responsible for network communications and propagation. In particular:
   - Infects any newly inserted removable drives.
   - Utilizes peer-to-peer networks in order to connect to C&C servers.
   - Uses hardcoded credentials to infect WinCC devices [4].
   - Connects to all available default network shares.
   - Exploits a *zero-day* vulnerability (MS10-061) in Microsoft Windows print spooler service.
   - Exploits MS08-067 Windows Server Service Vulnerability.
4. **Establish and maintain remote access** Adversaries utilize peer-to-peer networks for communicating to C&C centers and updating purposes, whereas during the final infection process the malware was designed to hide its code on PLCs using a specially crafted rootkit.
5. **Remote control and device manipulation** The adversaries were able to remotely adjust the spinning rate of the network enabled centrifuges and, at the same time, falsify the information sent back to the operators. The latter enabled them to increase the spinning rate at a level where centrifuges started to fail without anyone noticing.

Although its main target were Iran's enrichment uranium plants the worm managed to spread throughout the world. In September 2010, approximately 100,000 hosts were infected (40,000 unique external IP addresses from over 155 countries), 60% of which, were located in Iran.

4.2 Dragonfly

A group of well-funded, highly-skilled adversaries launched a cyber-espionage campaign, the first advanced attack after Stuxnet that targeted ICS equipment [34]. The group behind the attack was named 'Dragonfly' by Symantec or 'Energetic

Bear' by other security firms. Initially, the targeted systems were aviation and defense industries located in the US/Canada but afterwards the attacker's group showed interest for industries of the energy sector. Using the *watering hole* attack technique [9] the adversaries managed to infect with malware several company networks. Furthermore, they managed to inject malicious payloads on available ICS vendor software found on official websites. The attack was staged in three phases: Firstly, spear-phishing campaigns were launched and remote access was established via a Remote Access Trojan (RAT) horse. Then, Havex software was used in *watering hole* attacks against official vendor websites thus redirecting users to servers with malware infected ICS software.

1. **Reconnaissance phase** Retrieval of corporate information from aviation, defense and energy industries' web presence.
2. **Initial infection phase** Via spear-phishing techniques 'Dragonfly' group infected employees' workstations with HAVEX malware. Initially, the malware harvested data, such as emails, contact lists and documents.
3. **Establish and maintain remote access** HAVEX malware served as a means for installation of other malware sent from Dragonfly servers (e.g., Karagany RAT, password stealer module, etc.). It consisted of a remote access Trojan and a server module written in PHP. After installation, the malware communicated with C&C server in order to download and execute other malicious payloads, such as an OPC scanning module, that utilized specific TCP ports used by Siemens and Rockwell automation systems, to retrieve information for ICS equipment.
4. **Propagation phase** By exploiting vulnerabilities in the vendors' websites the Dragonfly group was able to place its payloads in three major ICS vendor websites. In the case of the first vendor's website (eWon) the adversaries managed to change a download link so as to point to a modified package of a VPN application (Talk2Me) that provided access to PLCs. The second compromised website belonged to a European manufacturer of PLC devices whereas the third website was owned by a company that manufactured ICS for energy sector, including wind turbines. None of the websites affected, enforced any authorization mechanisms for accessing the ICS software.
5. **Remote control and device manipulation** Havex's main target was ICS communication interfaces and especially OLE for Process Control information. In all three cases described the attackers successfully managed to inject malicious code into the vendor's driver package. Investigators were able to identify 88 different versions of Havex, 146 C&C centers (mainly vulnerable blog websites) and 1,500 IP addresses of potential victims, most of which, in Europe.

Although Dragonfly attack did not disturb any industrial control process or lead to a severe energy outage, the adversaries manage to collect a large amount of valuable information that could potentially assist them in launching future attacks [34]. Furthermore, the OPC scanning module could be used to compromise ICS maintenance suppliers' services such as eWon, which utilizes approximately a

million remote connections in order to provide remote support on ICS equipment. Based on later investigations, it was discovered that the Dragonfly group had also targeted the pharmaceutical industry aiming at stealing valuable information such as medicine recipes, batch production sequence steps as well as manufacturing plant volumes and capabilities.

4.3 Attacking Internet Facing PLCs: PLC-Blaster

In BlackHat 2015, security researchers presented a malware that targets network enabled PLCs [24], while a similar attack was presented the next year at the same conference [42]. The latter was mainly consisted of a self-replicating worm that could infect specific manufacturer PLCs such as Siemens SIMATIC. The malware was able to probe port 102/TCP in order to identify PLC devices (in this case S7-1200). Then, after establishing connection with the target the exploitation phase begins. The worm mimicked TIA-portal, a platform supported by Siemens for remote management, to implement the manufacturer's proprietary binary protocol named `S7CommPlus`. The latter utilized both TPKT and ISO8073 [14], to remotely infect and control the PLC. The functionality features of the protocol included configuration of the device, start/stop its operation, modify its processes' variables, uploading/downloading of programs as well as debugging & alerting. In order to issue commands from one PLC to another an analysis of the protocol's message structure was conducted and vulnerabilities such as insufficient integrity protection mechanisms, password originated from hash encryption keys and disabled default access protection settings were found. In summary the attack vector included the following steps:

1. **Reconnaissance phase** The adversary scans the Internet using Shodan (or similar) search engine and locates vulnerable PLCs.
2. **Initial infection phase** Since the PLC lacks of integrity protection mechanisms, the malware mimics TIA portal to issue commands and transfer its malicious payload in order to take over the PLC(s).
3. **Propagation phase** The warm sends messages to 102/TCP communication port. Then, by using the proprietary Siemens protocol (`S7CommPlus`), tests the target and tries to download a copy of itself. If no connection is established after 200 prob cycles the IP address is incremented.
4. **Establish and maintain remote access** Using an embedded Socks4proxy the worm communicates to an external C&C center. The worm is stored on the device so it can survive a restart or even a power loss.
5. **Remote control and device manipulation** The worm can alter any outputs of the compromised PLCs as well as force them to enter an endless loop thus triggering an error condition.

4.4 Attacking Industrial Robots

According to the International Federation of Robotics forecast [30], 1.3 million industrial robot units will be installed in factories located all over the globe until the end of 2018. Robots are used in almost every critical industrial sector such as automotive, aerospace, defense, plastics, electronics and electrical, metal fabrication, pharmaceutical, railway and many more. Several security firms and researchers have pointed out vulnerabilities in both domestic and industrial robots [8, 30]. The latter are usually of large volume used in complex manufacturing processes and play an essential role in production lines. Industrial robots are exceptionally complex cyber-physical systems that include actuators, sensors, human-robot interfaces and are constantly connected to computer networks primarily for operation, programming and maintenance purposes. In [30] researchers mainly focused on industrial robots by analyzing protocols and relative software. The impact of a single software vulnerability could have serious consequences, since, it could enable an adversary to inflict a massive financial damage and/or even threat human lives. After the Industry 4.0 [26] was introduced, almost all new models of industrial robots tend to incorporate IoT technologies such as connectivity and operational features that expose them to a much broader attack surface. The researchers in [30] utilized an actual robot (ABB six-axis IRB140) in order to demonstrate a series of attacks such as alter or introduce minor defects in the manufactured products, physically damage the robot, steal industrial secrets and/or cause human injuries.

Using well known search engines (Shodan, ZoomEye and Censys) they managed to discover multiple industrial robots' network interfaces connected directly to the Internet. As of late March of 2017, researchers discovered approximately 84,000 of industrial robots that were exposed to the Internet, 5105 of which did not require any authentication, 59 had known vulnerabilities whereas the researchers were able to identify 6 totally new (*zero-day*) ones. These included the usage of a self-signed certificate for multiple devices, network service banners that disclosed sensitive information (vendor's name, MAC address, firmware version, CPU model, CPU frequency, etc.), outdated software components (application & cryptography libraries, compilers, kernel), default credentials or no/poor authentication mechanisms, static VPN private keys on publicly available firmware images, adoption of symmetric cryptography schemes in VPNs, the use of plain HTTP web interfaces with no/poor input sanitization, default 'as is' use of open software (e.g., REST layer in PHP) and publicly available unstripped firmware images. A realistic APT attack vector against industrial robots includes the following steps:

1. **Reconnaissance phase** Adversaries use search engines to discover and enumerate Internet-exposed robot interfaces by searching for specific strings in the HTTP header (e.g., 'eWON', 'Westermo', etc.). Then, they manage to locate several software vulnerabilities by reading freely available technical documentation, reverse engineering publicly available software (e.g., firmware files, controller software) and even run exploitation tests using available simulation software (e.g., ABB's software suite).

2. **Initial infection phase** Using the vulnerabilities found in previous phases adversaries establish a connection with the device (e.g., authentication bypass in ABB's eWON industrial cellular router, FTP static credentials to access the command driver, memory errors found in the RobAPI). Since no security mechanisms are present and the Internet interface is used, the attack will remain undetected from any IDS/IPS equipment installed in the internal network.
3. **Establish and maintain remote access** Through FTP access, attackers upload custom, malicious software and trigger a reboot using the command shell reboot FTP function. The malicious files are executed and all robot features are now remotely controlled via a C&C center.
4. **Propagation phase** Utilizing connectivity features installed in robot's main computer (e.g., FlexPendant, RobotStudio) attackers manage to discover and attack other robot network interfaces that are connected on the company's internal network.
5. **Remote control and device manipulation** Adversaries are able to launch a series of attacks, which the researchers categorized into five classes, evaluating the potential impact of each one individually. The categorization was made under the assumption that a robot must be able to at least read accurately from its sensors and execute its control logic, perform precise movements, and not harm humans in any circumstance. In particular:

   (a) **Altering the Control-Loop Parameters** This attack includes the modification of the configuration control closed/open loop parameters used to control robot movements. Implications of such attack can lead to safety boundary violation and even breakage of robot parts.
   (b) **Tampering with Calibration Parameters** Repeatedly manipulation of the controller's calibration parameters at runtime could lead to Denial-of-Service (DoS) attacks.
   (c) **Tampering with the Production Logic** In the case where the controller does not enforce *end-to-end* integrity checks a program task could be altered thus leading to the manufacturing of defective products or fully compromising a factory's manufacturing process.
   (d) **Altering the User-Perceived Robot State** In this case the robot's user interface is manipulated in order to hide/misinform the operator of the true robot status so as to fool him/her into making wrong risk evaluations. This kind of attack can put operators at risk and even lead to human injuries.
   (e) **Altering the Robot State** Changing the robot's true state may have major impact especially when combined with other attacks (e.g., manufacture a large amount of defective products).

Realistic threat attack scenarios may include sabotage of an entire production line via product's characteristics alteration followed by a ransomware campaign in order to reveal which product batch was affected, physical damage to industrial equipment, human injuries and/or the use of the device as a means to exfiltrate sensitive industrial data (e.g., industrial secrets such as calibration parameters).

4.5 PLC Ransomware: LogicLocker

In 2017 researchers of Georgia Institute of Technology [17] presented a hypothetical ransomware attack scenario in which, adversaries target network connected PLCs located in a water treatment plant. The targeted PLCs were used to control the valves which, in turn, control the amount of chlorine that is added into the water. In particular, they developed a framework named 'Logiclocker' that then used to attack some of the most popular PLCs in the market such as Schneider Modicon M221, Allen Bradley MicroLogix 1400, and Schneider Modicon M241. The phases described in order to launch a successful ransomware campaign included initial infection, lateral movement within internal SCADA networks, reconnaissance and target discovery, locking and encrypting process and finally the negotiation for the ransom. In their PoC attack the researchers managed to retrieve the device's credential (in this case Modicon M241) either by stealing or using brute force attack techniques. A typical ransomware attack scenario consists of the following phases:

1. **Reconnaissance phase** Adversaries locate Internet facing PLCs via search engines (e.g., Shodan).
2. **Initial infection phase** Using stealing, brute force and dictionary attack techniques they manage to recover authentication information (e.g., user/system credentials) from the discovered Internet facing PLCs.
3. **Propagation phase** Embedded payloads enable the malware to scan the internal SCADA networks of the water treatment plant in order to infect other vulnerable PLCs.
4. **Establish and maintain remote access** Adversaries remotely reprogram the infected PLCs with new passwords thus locking the legitimate operators out.
5. **Remote control and device manipulation** The attackers remotely encrypt the PLCs' software using well known encryption algorithms (e.g., AES) with a newly generated key.
6. **Ransomware phase** Via the LogicLocker framework an email is sent to the water treatment plant that threatens to release chlorine in the water and cause massive human fatalities.

5 APT Attacks on Smart Grid SCADA Networks and Field Devices

Smart grids are not always engineered having in mind the security-by-design principle, thus making them vulnerable to various novel cyber threats. In this section we analyze recent, high profile attacks that utilize APT techniques.

5.1 Attacks on Generation Systems: The Aurora Attack

In 2007, an attack scenario that targeted electric power generators, was demonstrated at the Idaho US National Labs [43, 47]. Network enabled PLCs (circuit breakers) were forced to open and close in a very fast rate (4 times per second) in order to force the affected power generator to desynchronize thus resulting in its physical destruction. In a potential attack scenario described in [47] an attacker compromises the company's corporate network to propagate to the facility's main control center and take advantage of an existing communication link that is used to remotely administer the PLCs.

In order to launch an Aurora-like attack, the attacker would have to overcome intentional delays in switching on and off and synchronization checks that exist to ensure the smooth operation of the system. Assuming that the attacker has compromised a sufficient number of devices, it is possible to inject falsified commands to trip and reclose a circuit breaker in a rapid repetitive way. In particular, an hypothetical Aurora attack scenario can be described as follows:

1. **Reconnaissance phase** Adversaries manage to collect corporate information (e.g., email accounts) that then use to launch a spear-phishing campaigns.
2. **Initial infection phase** Using known and *zero-day* exploits they manage to elevate privileges and install a RAT tool in order to control the infected workstations remotely.
3. **Establish and maintain remote access** Using network pivoting techniques they manage to navigate the facility's internal network and infect a workstation located in the control center. Moreover, using similar exploitation techniques they establish remote access to the workstation and through it to the target PLCs.
4. **Lateral movement and propagation phase** Using NMAP or similar tools they fingerprint the relays' brand name and model (Ethernet and/or Modbus). Then, via passive eavesdropping and vulnerability exploitation techniques (e.g., false data injection attacks [29]) they manage to remotely control the circuit breakers and bypass protection relays.
5. **Remote control and device manipulation**
    - **Step 1** The circuit breaker(s) are opened isolating the generator from the grid.
    - **Step 2** The generator starts to speed up and the frequency of the generator increases.
    - **Step 3** The frequency difference between the grid and the generator increases.
    - **Step 4** After a particular amount of time the circuit breakers are closed, thus connecting back the generator to the grid.
    - **Step 5** The generator is forced into synchronization with *out-of-sync* conditions thus causing substantial electrical and mechanical transients.
    - **Step 6** Steps 1–5 are repeated in a timely manner until the generator is permanently damaged.

Adversaries (e.g. terrorists, nation state) could launch concurrent attacks against multiple generators, in order to destabilize large areas of a country's smart grid thus maximizing the potential impact of the attack.

5.2 Attack on the Ukraine's Smart Grid Distribution Network (2015)

One of highest impact, highly coordinated, stealthy APT attack against the smart grid is considered to be the one that took place on December 23, 2015 against an Ukraine regional electricity company named "Kyivoblenergo". The attack resulted in massive outages that affected approximately 225,000 customers for several hours [7], whereas substation control (e.g., circuit brakers) was switched to manual for weeks.

The adversaries utilized a variety of attack techniques including the use of spear-phishing campaigns (they impersonated an email message from the Ukrainian parliament), variants of *BlackEnergy* 3 and *KillDisk* malware as well as the manipulation of Microsoft Office documents in order to gain an initial foothold to the company's internal network. The attackers possessed specialized knowledge of ICS network connected devices such as Uninterruptible Power Supplies (UPSs), HMI interfaces, credential harvesting techniques, and SCADA client software. The attack vector can be described as follows:

1. **Reconnaissance phase** Nation-state adversaries launched a spear-phishing campaign with malware-infected Microsoft Office documents against corporate users.
2. **Initial infection phase** By exploiting Windows well known and *zero-day* vulnerabilities they managed to install key-loggers and retrieve user credentials.
3. **Lateral movement and propagation phase** Initially, the adversaries performed a reconnaissance of internal SCADA network and devices. Then, pivoting throughout different network segments enabled them to locate and infect SCADA dispatch workstations and servers. In particular, they managed to gain access to operators' workstations, located in control rooms, that run HMI software.
4. **Establish and maintain remote access** Using existing, legitimate remote administration tools, installed on operators' workstations, they managed to remotely connect to the aforementioned workstations and lock the legitimate operators out. In addition they uploaded malicious firmware in field communication devices to prevent any recovery attempts.
5. **Remote control and device manipulation** In order to magnify the impact of the attack the adversaries proceeded with the following actions:
   (a) Remotely opened multiple circuit breakers to cause massive outages. (attack's main target)
   (b) Reconfigured UPS systems to cause outages in company's buildings.

(c) Launched a remote telephonic denial of service on the energy company's call center to frustrate the impacted customers.
(d) Utilized a modified version of *KillDisk* malware to destroy forensic evidence and render workstations inoperable.

5.3 Attack on the Ukraine's Kiev Transmission Station (2016)

In December 2016, the Ukrainian's smart grid SCADA systems were targeted for a second year in a row [18]. The target of the attack was a 200 Megawatt transmission station located near the city of Kiev. Similar to the previous attack, the adversaries launched spear phishing campaigns in which they wrapped in a word document attachment the malware *CrashOverride/Win32/Industroyer* [27] in order to infect the employees' workstations. This time the attack techniques used were far more sophisticated and stealthier than the first attack. The malicious code was capable of being preprogrammed to launch an attack against multiple targets, at a future time, without any intervention from the attackers. The malware was modular and included, among others, the main program that ensured communications with C&C centers and IIoT equipment, four different malicious payloads that correspond to industrial control protocols IEC 101, IEC 104, IEC 61850, OPC Data Access (OPC DA) and a DoS tool that targeted a particular family of protection relays (Siemens SIPROTEC). Figure 3 depicts the basic functionality of the malware. A more detailed description of the software components as well as a walkthrough of the attack [10] is presented here:

1. **Reconnaissance phase** Using publicly available information found on the Internet (e.g., YouTube) the adversaries were able to enumerate substation's ICS. Having selected their target, then they launched a spear-phishing campaign (July 2016) against corporate users.
2. **Initial infection phase** Using advanced exploitation techniques they managed to gain a foothold to the substation's internal network. In particular, after they managed to infect corporate workstations and/or servers, the malware installed the main backdoor program responsible also to control all other SCADA modules. The latter could be programmed to communicate with the attackers at a specific time every day via C&C servers (active TOR nodes). Initially, it authenticated with a local proxy (TCP port 3128) and then utilized an HTTPS channel to connect to external C&C servers. After a successful privilege escalation process, the backdoor was masqueraded as a legitimate windows service program to avoid any detection.
3. **Lateral movement and propagation phase** The adversaries incorporated highly customized, sophisticated SCADA communication modules in order to interact with IIoT equipment. The purpose of the SCADA communication modules was twofold: Initially, they were used in the enumeration/propagation phase, in which specific commands were issued to fingerprint IIoT devices, and as a means of launching the main attack by issuing the necessary control commands.

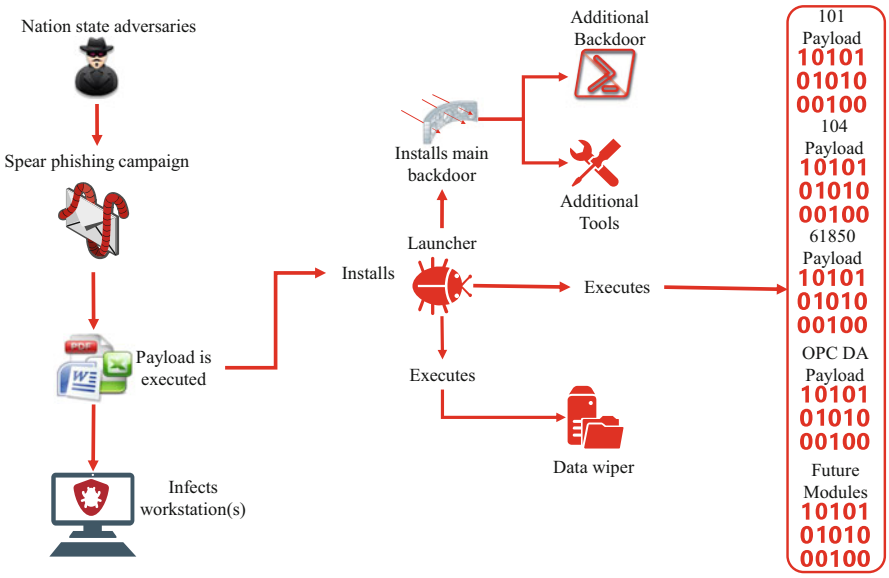

Fig. 3 Attack vector of the malware (CrashOverride/Win32/Industroyer) on Ukraine's smart grid (December 2016)

- **IEC 60870-5-101 module** It utilized the file 101.dll to implement the IEC 101 protocol so as to communicate with compatible RTUs. Upon execution, the payload located and terminated the legitimate process used to communicate with IEC 101 devices. Then, a new process was started in order to take over the control of the RTUs.
- **IEC 60870-5-104 module** Since IEC 104 extends the IEC 101, the module utilized TCP/IP network as its main communication channel. It also supported a configuration file for customization and operated in a similar way as the IEC 101 payload.
- **IEC 61850 module** Unlike the previous modules this one consisted of both an executable file (61850.exe) as well as a DLL file. When executed, the malicious program enumerated all IP addresses and tried to connect to TCP port 102. Then, Manufacturing Message Specification (MMS) commands were used to enumerate and control all discovered devices, such as circuit breakers.
- **OPC DA module** OLE, Component Object Model (COM) and Distributed Component Object Mode (DCOM) are Microsoft technologies that are used for real-time data exchange, based on a client/server model. Similar to IEC 61850 payload, the malicious program consisted of a .EXE and a .DLL file that, incorporated both 61850 and OPC DA functionalities. Upon execution, enumeration of all OPC servers and devices was performed (ABB solutions). Then, the OPC's state was altered using the `IOPCSyncIO` interface.

- **Port scanner and DoS tools** Additionally, a custom-made port scanning program and a DoS tool were included in the malware. The latter could be used against SIPROTEC Siemens devices by utilizing a known vulnerability (CVE-2015-5374).

4. **Establish and maintain remote access** Aside the main backdoor, the attackers utilized a trojanized version of the Windows notepad application, to serve as a back-up persistence mechanism, in order to regain access in the case of the main backdoor was found and disabled. To avoid detection, the embedded malicious code was heavily obfuscated and utilized different C&C servers than the one used from the main backdoor program.
5. **Remote control and device manipulation** To launch the attack, the adversaries utilized the 'Launch' module in which they had embedded specific time and dates (17 and 20 December). Once one of the dates was reached the module was programmed to execute two processes in high priority. In particular:

   - **Payload.DLL** The actual name of the DLL file that contained the main payload was not hardcoded into the module but had to be supplied from the adversaries along with a configuration file. Upon execution, the payload used the functionality embedded in aforementioned modules to issue commands to located RTUs and PLCs, such as turn the device off or change their status (e.g., open/closed).
   - **Data wiper module** This payload was scheduled to launch with a delay of 1–2 h from the first payload. It included the file `haslo.exe/dat` that when executed, it modified the registry value `ImagePath` with an empty string thus rendering the system unusable. In addition, it deleted specific files by overwriting them twice and terminated all running process in order to make the system crash. The list with the file extensions for deletion included, among others, Windows binaries as well as MS SQL server and ICS configuration (ABB PCM600) files.

6 Conclusions

In this chapter, we analyzed recent, high impact, real APT attacks on IIoT ecosystem, as well as PoC attack scenarios that utilize APT techniques based on related work of security researchers. In the case of real APT attack scenarios, malicious actors mainly focused in exploitation techniques that took advantage of indirect attack paths, that exist between corporate and industrial networks. From the analysis of the attack techniques used in real cyberattacks against Ukraine's smart grid [7, 18], one can ascertain that adversaries are constantly evolving their attack techniques in terms of customization, stealthiness and user interaction. Adversaries mainly target systems that are being used by corporate users, such as mail servers, to infiltrate to the company's internal network. Then, via privilege escalation techniques, they manage to fully control the infected workstations and propagate

throughout the network so as to locate and exploit IIoT equipment such as SCADA HMI software. Finally, using vulnerabilities found on legacy/new network industrial protocols they infect and remotely control field devices (e.g., circuit breakers).

On the other side the majority of PoC attack scenarios utilize search engines, such as Shodan, to locate and enumerate exposed web interfaces of IIoT equipment. Such attack scenarios are quite realistic. Indications of real attacks on Internet facing IIoT equipment can also be found in [22]. Another interesting finding is that a vast amount of publicly available industrial software, such as firmware files, is available to adversaries. Such information may allow them to extract valuable information and to refine their exploitation methods. In the attack scenarios examined [8, 17, 30, 42], security researchers managed to successfully exploit *zero-day* and well known vulnerabilities of the IIoT ecosystem (e.g., hardcoded credentials on firmware files, remote code execution on web interfaces) to remotely control and manipulate mission critical industrial equipment such as sensors, actuators and robotics.

In order to mitigate the risks that involve relative IIoT technologies, organizations/companies should always ensure that their mission critical ICSs can survive a large scale APT attack. Disaster recovery plans should always include well defined incident response procedures that correspond to several attack scenarios which, in turn, are thoroughly planned and tested. Furthermore, specialized security equipment, such as IDS/IPS equipped with advanced detection techniques (e.g., YARA rules [21]), should be applied throughout industrial networks. Developers of HMIs and other relative SCADA applications should adopt the secure software life cycle practices used by operating system and other application developers for over a decade. Security-by-design [36] should also be adopted by IIoT equipment manufacturers and industrial software developers to ensure strong authentication, integrity protection and authentication mechanisms are in place. Various critical procedures, such as the over-the-air updating process of IIoT equipment, should be properly implemented. In addition, network administrators should always protect industrial web interfaces via dedicated firewall devices, properly segment and sometimes isolate mission critical SCADA systems, especially those that utilize outdated, vulnerable industrial network protocols. Finally, other security best practices that promote defense-in-depth include anomaly detection systems [1], as well as message authentication, integrity and encryption [11, 37, 39, 41].

References

1. Alves T, Das R, Morris T (2018) Embedding encryption and machine learning intrusion prevention systems on programmable logic controllers. IEEE Embed Syst Lett 10:99–102
2. Andrea M (2015) GE MDS PulseNET hidden support account remote code execution vulnerability. https://www.zerodayinitiative.com/advisories/ZDI-15-440/
3. Antón SD, Fraunholz D, Lipps C, Pohl F, Zimmermann M, Schotten HD (2017) Two decades of scada exploitation: a brief history. In: 2017 IEEE Conference on Application, Information and Network Security (AINS). IEEE, pp 98–104

4. Berger H (2014) Automating with SIMATIC S7-400 inside TIA portal: configuring, programming and testing with STEP 7 Professional. Wiley
5. Brian G, Fritz Sands TTMZDI Hacker machine interface: the state of scada HMI vulnerabilities. White paper, Trend Micro
6. Broad WJ, Markoff J, Sanger DE (2011) Israeli test on worm called crucial in Iran nuclear delay. NY Times 15:2011
7. Case DU (2016) Analysis of the cyber attack on the Ukrainian power grid. Electricity Information Sharing and Analysis Center (E-ISAC)
8. Cerrudo C, Apa L (2017) Hacking robots before Skynet1. IOActive Website
9. Chen P, Desmet L, Huygens C (2014) A study on advanced persistent threats. In: IFIP International Conference on Communications and Multimedia Security. Springer, pp 63–72
10. Cherepanov A (2017) Win32/industroyer: a new threat for industrial control systems. White paper, ESET, June 2017
11. Cherifi T, Hamami L (2017) A practical implementation of unconditional security for the IEC 60780-5-101 scada protocol. Int J Crit Infrastruct Prot 20:68–84
12. Clarke GR, Reynders D, Wright E (2004) Practical modern SCADA protocols: DNP3, 60870.5 and related systems. Newnes
13. Cowan C, Wagle F, Pu C, Beattie S, Walpole J (2000) Buffer overflows: attacks and defenses for the vulnerability of the decade. In: DARPA Information Survivability Conference and Exposition, 2000, DISCEX'00. Proceedings, vol 2. IEEE, pp 119–129
14. Devarajan G (2007) Unraveling scada protocols: using sulley fuzzer. In: Defon 15 Hacking Conference
15. Drias Z, Serhrouchni A, Vogel O (2015) Taxonomy of attacks on industrial control protocols. In: 2015 International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS). IEEE, pp 1–6
16. Falliere N, Murchu LO, Chien E (2011) W32. stuxnet dossier. White paper, symantec corporation. Secur Response 5(6):29
17. Formby D, Durbha S, Beyah R (2017) Out of control: ransomware for industrial control systems. http://www.cap.gatech.edu/plcransomware.pdf
18. Goodin D (2017) Hackers trigger yet another power outage in Ukraine. https://arstechnica.com/security/2017/01/the-new-normal-yet-another-hacker-caused-power-outage-hits-ukraine/
19. Gruber E (2014) Verifying ASLR, DEP, and safeSEH with powershell. Blog, NetSPI 23
20. Gutierrez JA, Naeve M, Callaway E, Bourgeois M, Mitter V, Heile B (2001) IEEE 802.15. 4: a developing standard for low-power low-cost wireless personal area networks. IEEE Netw 15(5):12–19
21. Hurd CM, McCarty MV (2017) A survey of security tools for the industrial control system environment. Technical report, Idaho National Laboratory, Idaho Falls, ID
22. Israel B, Ross R (2018) ICS threat broadens: nation-state hackers are no longer the only game in town. https://www.cybereason.com/blog/industrial-control-system-specialized-hackers
23. John KH, Tiegelkamp M (2010) IEC 61131-3: programming industrial automation systems: concepts and programming languages, requirements for programming systems, decision-making aids. Springer, Heidelberg
24. Klick J, Lau S, Marzin D, Malchow JO, Roth V (2015) Internet-facing PLCs-a new back orifice. Black Hat USA, pp 22–26
25. Kushner D (2013) The real story of stuxnet. IEEE Spectr 50(3):48–53
26. Lasi H, Fettke P, Kemper HG, Feld T, Hoffmann M (2014) Industry 4.0. Bus Inf Syst Eng 6(4):239–242
27. Lee RM, Assante, MJ, Conway T (2017) CRASHOVERRIDE: analysis of the threat to electric grid operations. Dragos Inc. https://dragos.com/wp-content/uploads/CrashOverride-01.pdf
28. Lian FL, Moyne JR, Tilbury DM (2001) Performance evaluation of control networks: ethernet, controlnet, and devicenet. IEEE Control Syst 21(1):66–83
29. Liang G, Weller SR, Zhao J, Luo F, Dong ZY (2017) The 2015 Ukraine blackout: implications for false data injection attacks. IEEE Trans Power Syst 32(4):3317–3318

30. Maggi F, Quarta D, Pogliani M, Polino M, Zanchettin AM, Zanero S (2017) Rogue robots: testing the limits of an industrial robot's security. Technical report, Trend Micro, Politecnico di Milano
31. Mohassel RR, Fung A, Mohammadi F, Raahemifar K (2014) A survey on advanced metering infrastructure. Int J Electr Power Energy Syst 63:473–484
32. Momoh J (2012) Smart grid: fundamentals of design and analysis, vol 63. Wiley, Hoboken
33. Mulligan G (2007) The 6LoWPAN architecture. In: Proceedings of the 4th Workshop on Embedded Networked Sensors. ACM, pp 78–82
34. Nelson N (2016) The impact of dragonfly malware on industrial control systems. SANS Institute, Bethesda
35. Pagliery J (2015) The inside story of the biggest hack in history. CNN, 5 Aug 2015
36. Radvanovsky R, Brodsky J (2016) Handbook of SCADA/control systems security. CRC Press, Boca Raton
37. Rrushi JL (2017) Defending electrical substations against 0-day malware through decoy I/O in protective relays. In: Dependable, Autonomic and Secure Computing, 15th International Conference on Pervasive Intelligence & Computing, 3rd International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), 2017 IEEE 15th International. IEEE, pp 486–493
38. Sadeghi AR, Wachsmann C, Waidner M (2015) Security and privacy challenges in industrial Internet of things. In: Proceedings of the 52nd Annual Design Automation Conference. ACM, p 54
39. Saxena N, Grijalva S (2017) Efficient signature scheme for delivering authentic control commands in the smart grid. IEEE Trans Smart Grid 9:4323–4334
40. Shacham H, Page M, Pfaff B, Goh EJ, Modadugu N, Boneh D (2004) On the effectiveness of address-space randomization. In: Proceedings of the 11th ACM Conference on Computer and Communications Security. ACM, pp 298–307
41. Shahzad A, Lee M, Lee C, Xiong N, Kim S, Lee YK, Kim K, Woo SM, Jeong G (2016) The protocol design and new approach for scada security enhancement during sensors broadcasting system. Multimed Tools Appl 75(22):14641–14668
42. Spenneberg R, Brüggemann M, Schwartke H (2016) PLC-blaster: a worm living solely in the PLC. Black Hat USA, Singapore
43. Srivastava A, Morris T, Ernster T, Vellaithurai C, Pan S, Adhikari U (2013) Modeling cyber-physical vulnerability of the smart grid with incomplete information. IEEE Trans Smart Grid 4(1):235–244
44. Stellios I, Kotzanikolaou P, Psarakis M, Alcaraz C, Lopez J (2018) A survey of IoT-enabled cyberattacks: assessing attack paths to critical infrastructures and services. IEEE Commun Surv Tutorials 20:3453–3495
45. Tovar E, Vasques F (1999) Real-time fieldbus communications using profibus networks. IEEE Trans Ind Electron 46(6):1241–1251
46. Wan J, Tang S, Shu Z, Li D, Wang S, Imran M, Vasilakos AV (2016) Software-defined industrial Internet of things in the context of industry 4.0. IEEE Sens J 16(20):7373–7380
47. Zeller M (2011) Myth or reality does the aurora vulnerability pose a risk to my generator? In: 2011 64th Annual Conference for Protective Relay Engineers. IEEE, pp 130–136