# Developing Battery of Vulnerability Tests for Industrial Control Systems

Radek Fujdiak[1,2], Petr Blazek[1], Petr Mlynek[1], and Jiri Misurec[1]

[1]Brno University of Technology, Department of Telecommunications, Brno, Czech Republic
Email: {fujdiak,xblaze24, mlynek,misurec}@vutbr.cz
[2]Trustport, Brno, Czech republic
Email: {Radek.Fujdiak}@trustport.com

***Abstract*—Nowadays, the industrial control systems (ICS) face many challenges, where security is becoming one of the most crucial. This fact is caused by new connected environment, which brings among new possibilities also new vulnerabilities, threats, or possible attacks. The criminal acts in the ICS area increased over the past years exponentially, which caused the loss of billions of dollars. This also caused classical Intrusion Detection Systems and Intrusion Prevention Systems to evolve in order to protect among IT also ICS networks. However, these systems need sufficient data such as traffic logs, protocol information, attack patterns, anomaly behavior marks and many others. To provide such data, the requirements for the test environment are summarized in this paper. Moreover, we also introduce more than twenty common vulnerabilities across the ICS together with information about possible risk, attack vector (point), possible detection methods and communication layer occurrence. Therefore, the paper might be used as a base-ground for building sufficient data generator for machine learning and artificial intelligence algorithms often used in ICS/IDS systems.**

***Index Terms*—Security, Information security, Intrusion detection, Industrial control, Industrial communication.**

## I. Introduction

The Industrial Control Systems (ICS) were for a very long time isolated from traditional information networks [1]. Moreover, the disconnection from the internet was for long time considered as sufficient protection against cyber-attacks [2]. However, the new concepts such as Smart Grid or Internet of Things connect these systems more than ever [3], [4]. Even not necessarily connected to the internet, the threats to the control systems grew over the past years exponentially [5]. This was proven for example by the Stuxnet attack, which targets programmable logic controllers and deviates their expected behavior [6], [7]. In response to growing cyber-criminal activities, Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) were introduced [8]. These systems were created for classical information technology area, but over the recent years evolved to be efficient also in the industrial control systems by incorporating the detailed knowledge of protocols [9].

This paper focuses on the cybersecurity issue in ICS systems. First, we highlight the essential parts of ICS security together with related researches. Second, the requirements for the test environment are introduced together with tools, which are implemented in our cyber-physical testbed. The attack vectorization and vulnerability analysis over layers L1-L7 containing more than twenty different threats are provided. The results should be used for creating a battery of vulnerability tests in ICS systems as we provide not only information about the attacks, but also possible detection methods. In the case of the cyber-physical testbed, the testbed might be used to provide a sufficient amount of data and test battery could be used as an anomaly and security incident generator to obtain sufficient broad training data. These data are the primary input in the IDS/IPS commonly used in defense of ICS.

The rest of the paper is organized as follows: Section II provides state of the art for the ICS testbeds and ICS security related papers. Followed by Section III, which provides valuable information about the requirements for the test environment together with a suggestion for hardware and software. Section IV brings our analysis of vulnerabilities in ICS systems together with relevant information about their detection, vectorization and layer appearance. Finally, Section V provides conclusion and suggests future work.

## II. Related work

The ICS is a major part of operational technologies (OT), which refers to computing systems that are used to manage industrial operations in the industrial environment, in contrast to information technologies refering to information processing in enterprise networks. The first vulnerabilities of ICS systems that date from 1997 are based on Kaspersky analysis [5]. From that time, the number of vulnerabilities found in ICS systems exponentially grew. Nowadays, the ICS systems face a high number of threats at many different levels. To study, research and provide sufficient defense against these threats, testbed environments have been created across the whole world.

Holm et al. [10] provide a broad survey of 30 testbeds and laboratories focused on ICS across the world, where more than half are focused on vulnerabilities. Even so, the vulnerabilities are not closely discussed; the survey shows the limitations of the current testbeds. Implementation approaches are identified as physical, simulated/emulated and virtualized. Most of the testbeds focus on simulation/emulation and/or physical approach. Moreover, the ICS is mostly implemented only as a control center and the communication is limited to a very narrow class of field devices including mostly

only RTU (Remote Terminal Unit), MTU (Master Terminal Units) or PLC (Programmable Logic Controller). Further, the older communication protocols such as DNP3 or ModBus are dominating across the testbeds with very few focusing on newer IEC 60870 or IEC 61850. Last but not least, the crucial parameters of testbed are identified as: (i) fidelity, (ii) repeatability of experiments, (iii) measurements accuracy, and (iv) safe execution of tests.

Another very extensive survey of 37 testbeds (with only a few repetitions compared to previously mentioned survey) were published by Cintuglu et al. [11], where again more than half are focused on security and privacy awareness with dominating (D)DoS (Distributed Denial of Service) and MITM (Man-In-The-Middle) vulnerabilities. The most crucial features identified by this survey is communication heterogeneity of the testbeds, where only seven testbeds were identified as heterogeneous. This means that most of the highlighted testbeds are focused mostly on a single solution approach. Also, 24 from 37 testbeds were marked as they supporting multiple protocols, but generally this means to support only DNP3 and ModBus (and C37.118) with very few exceptions.

However, only very little information about the cybersecurity approach was highlighted in these two surveys. Humayed et al. [12] published a comprehensive survey of cyber-physical systems security. However, this survey selected high-level threats/risk approach without a detailed explanation of the specific vulnerabilities, attacks or threats. On the other hand, Drias et al. [13] published an analysis of cybersecurity for industrial control systems, where protocols such as DNP3 and ModBus are introduced. Moreover, a brief introduction to the mitigation via NIST and IEC standards is provided together with cryptographic countermeasures analysis. Compared to this survey, we bring more general threat approach, which might be used over different protocols and infrastructures compared with specific-protocol approach introduced in the paper of Drias et al. This may give many different security incident scenarios without communication protocol dependency. Gao et al. [14] provide extensive general level cyber-threat analysis for ICS systems, but the paper is focused on the network layer whereas our paper focuses more on the ICS protocols/layer. Other works are focused mostly on particular vulnerability, threat or attack of ICS such as German Steel Mill Cyber Attack via specific malware [15], Australian Maroochy Water Services Attack Case Study [16], Ukraine power grid attack via phishing, malware and manipulation [17], ICS DoS/Injection/Reconnaissance attacks [18] and others.

Based on our best practice, modern ISC test environments should focus on non-single solutions and approaches, which should provide the broadest data, scenarios and functions. Limiting to only a single solution might end-up with insufficient quality of data, narrow set of possibilities and much less efficient security testing. This paper improves the current state of the art by bringing together information for creating an efficient cyber-physical testbed for ICS systems together with highlighting the most significant threats and summarizing the detection techniques. These should provide a sufficient environment for generating necessary data for IPS/IDS, conducting security assessment tests.

## III. Cyber-physical environment

A crucial property of the testbed is a possibility to implement communication and attacks on any network protocol. This led us to develop an advanced cyber-physical environment that simulates ICS environment. The structure of the entire environment is shown in Figure 1.

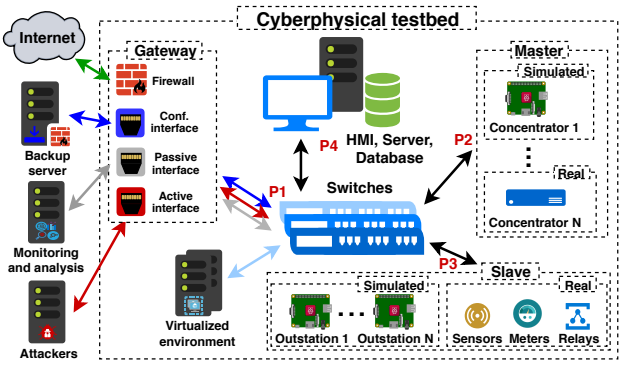


Fig. 1: Diagram of considered cyber-physical ICS environment for security assessment.

The main parts of the cyber-physical environment are:

- **Backup server** - This server should be isolated from the other parts of the network. It is used for self-healing and fast recovery. Therefore, it should not be a part of security testing. The protection of the server is made via the firewall. The Backup server stores the operating system and configuration settings of all devices.
- **Monitoring and Analysis server** - This server is a passive part of the security testing. The server contains specialized monitoring SW MENDEL and traffic logger Wireshark, which ensure the pcap files. As an alternative, several different open-sourced solutions such as SolarWinds, SNORT, Security Onion, Bro Network Security Monitor, WinPatrol or Osquery might be used.
- **Attackers server** - This server contains the batter of tests, presented in this paper and imported in KALI Linux distribution. As an alternative, different frameworks such as [19]: Backtrack, BlackArch, DEFT, HackPorts, Helix, NST, OpenVAS, Ophcrack, Pentoo, SamuraiSTFU, Secmic, Nesus, or others might be used . However, these are mostly general tools used often in IT networks and fail to provide full ICS security assessment environment. The DDoS techniques and load generator might be implemented via JMeter or Cisco TRex, where JMeter seems to be a more developer-friendly environment, which is important for implementing many different communication protocols, which are used in ICS.
- **Virtualization environment server** - The virtualization environment should provide an alternative to the high-cost physical approach and also a secure environment for high-risk tests. Therefore, three main techniques are used:

(i) Containerization, (ii) Simulation, and (ii) Sandboxing. The containerization is a new type of virtualization and provides a simple hardware-light solution for stacking clients and servers, which provide sufficient traffic-noise as in a real environment. The simulation provides a simulated ICS infrastructure. For the simulation purpose, the OMNET++ with implemented SCADA-SST framework is used. As an alternative to SCADA-SST,SCADASim, which is also OMNET++ compatible, might also be selected . However, the full alternatives might be NS2/NS3 or OPNET simulators [20] or solutions such as Green Energy Corp, Total Grid Community, DNP3 Simulator, Mitra Software, RocyLuo IEC Simulator, CONPOT ICS/SCADA, Total Grid Community or Rapid SCADA. However, most of these are very specifically focused. On the other hand, SCADA-SST provides full simulation environment with the possibility to connect it with a physical one. Moreover, the SCADA-SST offers a simple drag-and-drop solution. Selection of the simulation environment should follow these requirements: simplicity, scalability, modularity, friendly traffic logging and friendly customization. Finally, the sandboxing is a highly important technique of testing high-risk scenarios, which might otherwise irreversibly damage the infrastructure components.

- **HMI/Database server** - The selected HMI for our environment is openMUC, which provides friendly customization and has native support for newer protocols such as IEC 61850 and IEC 60870. Other alternatives might be Free Scada 2, IndigoSCADA, openDAX, S.E.E.R. 2, SCADA Process Viewer, ScadaBR, Szarp, OpenSCADA. However, these systems mostly focus only on older protocols and lack the newer ones such as IEC 61850 or IEC 60870. Further, the commercial alternatives such as mySCADA and PROMOTIC are high-cost and non-open with low or no user-expandability.
- **Physical field device** - We recognize two types of physical devices: (i) real field device, and (ii) simulated/emulated field device. The real devices are sensors, quality meters, relays and other physical commercial devices, which provide real environment experience. The real environment also provides the possibility to connect tested devices for security assessment and penetration testing. The simulated devices are in our case Raspberry Pi computers, which contain specific libraries for communication protocols such as DNP3 (library openDNP3), IEC 61850 (library libiec61850), IEC 60870 (library libiec60870-5) and others. This approach of simulating/emulating real devices on Raspberry computers provides a sufficient number of physical communicating devices, which add extra traffic noise.

This multi-solution approach simulates a whole SCADA infrastructure from the communication infrastructure, control center and field devices via different techniques to provide high-quality test environment and training data.

## IV. Vulnerability analysis of ICS systems

Despite the potential threats, many industrial systems are still not sufficiently secure. These systems were long time considered as isolated secured systems. However, the rise of digital and smart technologies entirely changes the industrial environment and new threats occur. VirusBlokAda discovered the first major sophisticated attack, known as Stuxnet [21], [22]. It was the first known worm focused on industrial systems control. This attack aimed to reprogram the PLC and hide the changes in the system. Another major attack called Night Dragon [23] was carried out in 2009 (fully documented in 2011). This attack, already known from IT networks (SQL - Injection, phishing, password breaks, Windows OS vulnerability, and others), was used to infiltrate the system. The attack was focused on harvesting information and data from more than 70 targeted systems. After these two attacks, the Saudi Arabian Oil Company was the target, where the attackers erased data from more than 35 000 computers. The attack took place in 2012 and is known as Shamoon [24]. In recent years, an attack named BlackEnergy [17], [25] has been performed in three variants. At the outset, it was a modular Trojan horse that could download the necessary components (to the target computer) to perform various tasks in the CI system. In the last variation, it is already a sophisticated attack with several phases and it paralyzes the target system totally. Attacks on ICS protocols are not designed as attacks from outside networks, but assume that an attacker already has access to the system (access is mediated via malicious SW). The above and further complex attacks consist of a series of smaller attacks targeting a specific application or device. An overview of the most common attacks on ICS systems is shown in Table I. The table includes the attack description, the risk of attack, the detection method, and one or more of the detection options. The last part of Table I are points representing the location of the attack. In Figure 1, which was presented in the previous section, there are four points (P1, P2, P3, and P4), where each point represents one point of the attack being executed (or the attacker can only simulate an attack from that point). Each attack in the table has been assigned one or more of these points to visualize the location from which the attack is being executed. Attacks made from network elements are marked with P1. If an attack is implemented from a master station or simulates its use it is marked as P2. The next Point P3 is for attacks from the Slave station and the last point P4 is for attacks from the SCADA or HMI. As mentioned in the previous chapter on the cyber-physical environment, it is possible to connect a real element that communicates using the above mentioned ICS protocols. Based on above description of the testbed and attacks, different scenarios can be created to verify the functionality and weakness of the connected device. Furthermore, thanks to a virtualized environment, it is possible to duplicate a real-world infrastructure and simulate specific conditions. The connected device can be tested as if it were in a specific system. The same simulation scenario can be applied to the entire infrastructure to identify potential shortcomings.

TABLE I: Results of analysis of the most common attacks in ICS systems [16], [18], [26]–[29], [29]–[40].

| Name | Attack point | Layer | Risk | Process | Detection | Detection methods |
|---|---|---|---|---|---|---|
| Network Mapping | P1 | L2, L3 | Identifying possible targets for further attacks | Scanning services within a network segment or multiple services within a single device | Signature detection or anomaly detection | L2 Traffic analysis (ARP request) |
| Firmware detection | P2, P3 | L7 | Identifying a specific version to which a particular type of attack can be executed | System version query | Signature detection or anomaly detection | Device queries analysis (Protocol dependent) |
| Configuration Error | P2, P3, P4 | L7 | Access control to device or application resources (data, conf. information, user data) | It occurs for each type of communication differently or as a specific character list | Known - signature detection; Unknown - behavioral analysis, anomaly detection (Difficult to detect) | Known - Protocol-based signatures are defined for detection; Unknown - Communication irregularities or unusual commands |
| Application Error | P2, P3, P4 | L7 | Code injection, data steal or denial of service | It occurs for each type of communication differently or as a specific character list | Known - signature detection; Unknown - anomaly detection (Difficult to detect) | Known - Protocol-based signatures are defined for detection; Unknown - Communication irregularities or unusual commands |
| Man-in-the-Middle (MITM) | P1 | | Communication manipulation | Communication of devices goes through a point controlled by an attacker | Behavioral analysis, anomaly detection | Detecting a communication out of standard time (Difficult to detect) |
| (D)DoS | P2, P3 | L2, L3, L4, L7 | Denial of service availability for users | Increased communication focused on resource depletion | Communication analysis or signature detection | Traffic increase, unusual regular traffic, signature for specific attacks |
| Changing of Database Conf. | P2, P3 | L7 | Shutdown devices that are controlled by the configuration database | Different for each type of communication, not recognizable from the normal behavior | Unauthorized access, or abnormality of the given communication in terms of time distribution | Access to the device out of standard time |
| Changing parameters | P2, P3 | L7 | A change in the behavior of the servicing device | Changing in communications or sending unexpected variables | Unauthorized access, Behavior Analysis of Administrator, Time analysis | Access to the device out of standard time |
| Zero-day attacks | P1, P2, P3, P4 | L2 to L7 (mainly) | Unauthorized access to resources | The attack is based on the attacked application/configuration | Anomaly detection, behavioral analysis | Attack-related activities - anomalous data transfers, abnormal behavior |
| Response Delay | P1 | L2, L3 | Communication interruption | An attacker delays frames at state protocols to terminate a connection | Anomaly detection, behavioral analysis | Specific packet comm. out of standard time |
| NTP Spoofing | P1 | L3, L7 | Reject a legitimate request | An attacker injects a NTP response on legitimate NTP time request of device (legitimate response is undelivered). Due to a badly set time, the device does not get response to request | Communication analysis | Evaluate packet dropping based on device behavior pattern |
| Protocol Rule Exploitation | P2, P3 P4 | L2, L3, L4, L7 | Using a protocol weakness for device control | An attacker manipulated with a process or a service that is used to complete a control process (uses MITM) | Known - signature detection; Unauthorized access; Unknown - Behavioral analysis (Difficult to detect) | Known - set of signatures for specific protocol; Unknown - Analysis of abnormal communication |
| Fake Master | P2 | L7 | Device control, communication manipulation | An attacker simulates the mater station and sends legitimate requests to the slave station | Behavioral analysis, abnormal behavior | Detecting comm./commands out of standard time. Monitor changes in network (Difficult to detect after replacing the station) |
| Manipulation Injection | P3 | L7 | Device control, communication manipulation | An attacker simulates message about change on device (e.q., exceeded threshold value), the master station responds | Behavioral analysis, abnormal behavior | Evaluation of the device behavior based on analysis. Messages about change sent out of standard time |
| Malicious HW/SW | P1, P4 | L7 | Backdoor through the infected device (usb, computer, etc) | The person connects / opens infected device, which leads to backdoor for attackers | Known- signature detection; Unknown - behavioral analysis | Access monitoring from the outside network. Set signatures for known malicious software (Firewall) |
| Auto Replay message | P1 | L2, L3, L4, L7 | Communication manipulation | An attacker stores legitimate messages that are transmitted unchanged for manipulation with the recipient (only for protocol without ACK or sequence number) | Behavioral analysis | Detecting a communication out of standard time. Multiple messages out of standard time |
| Connection Hijacking | P1 | L2, L3, L4, L7 | Communication interruption, communication manipulation | An attacker sniffs a communication and interrupts a legitimate communication. Then establishes new connection with device. The attack uses state protocols (e.g. TCP) | Behavioral analysis, abnormal behavior | Detecting a communication out of standard time (Difficult to detect after a connection establishing) |
| Buffer Overflow | P2, P3, P4 | L7 | Data steal, data change, files damage, etc. | An attacker overwrites adjacent memory locations with specific instructions for actions intended by a hacker | Known - Signature detection; Unknown - Behavioral analysis (difficult to detect) | Known - database of signature from IPS system; Unknown- based on communication pattern are detected anomalies |
| Function code | P2, P3 | L7 | Device manipulation | An attacker uses the function codes to affect the device (Reset, data change, reboot etc) | Behavioral analysis (difficult to detect) | Based on communication pattern are detected anomalies |
| Memory Corruption | P3 | L7 | Device failure, device manipulation | For example, an attacker modifies ladder logic (only on PLC), which affects the functionality of the program | Known - Signature detection; Unknown - Behavioral analysis (difficult to detect) | Known - database of signature from IPS system; Unknown- based on communication pattern are detected anomalies |
| Access on System-level | P4 | L7 | Unauthorized access | An attacker gains access to a system with administrative privileges | Behavioral analysis (difficult to detect) | Based on communication pattern are detected anomalies |

## V. Conclusion

The paper describes the current status of progress in the area of cybersecurity for ICS systems. We bring together the main surveys in the field and highlight the possible shortcomings. Among that, we provide clear requirements for ICS testbed, which should be used for security assessment methods, data generation, and research. The main parts of ICS systems are introduced together with possible implementation alternatives. Moreover, we introduce the best choices also selected in our testbed. Last but not least, the vulnerability analysis of ICSs is provided with more than twenty cyber-threats.

## VI. Acknowledgement

The National Sustainability Program under grant no. LO1401 and the Ministry of the Interior of the Czech Republic under grant no. VI20172019057 financed the research described in this article. Our research and the idea of the paper is coming from the research conducted and supported by research project Aggregated Quality Assurance for Systems (AQUAS H2020-EU.2.1.1.7 ID: 737475). For the research, the infrastructure of the SIX Center was used.

## References

[1] I. Zolotova, R. Hosak, and M. Pavlik, "Supervisory control sustainability of technological processes after the network failure," *Elektronika ir Elektrotechnika*, no. 9 (125), pp. 3–7, 2012.

[2] S. Karnouskos, "Stuxnet worm impact on industrial cyber-physical system security," in *IECON 2011-37th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2011, pp. 4490–4494.

[3] S. Karnouskos and A. W. Colombo, "Architecting the next generation of service-based scada/dcs system of systems," in *IECON 2011-37th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2011, pp. 359–364.

[4] J. Milosevic, N. Sklavos, and K. Koutsikou, "Malware in iot software and hardware," in *Workshop on Trustworthy Manufacturing and Utilization of Secure Devices (TRUDEVICE'16)*, 2016, pp. 1–4.

[5] O. Andreeva, S. Gordeychik, G. Gritsai, O. Kochetova, E. Potseluevskaya, S. I. Sidorov, and A. A. Timorin, "Industrial control systems vulnerabilities statistics," *Kaspersky Lab, Report*, 2016.

[6] N. Falliere, L. O. Murchu, and E. Chien, "W32. stuxnet dossier," *White paper, Symantec Corp., Security Response*, vol. 5, no. 6, p. 29, 2011.

[7] A. Matrosov, E. Rodionov, D. Harley, and J. Malcho, "Stuxnet under the microscope," *ESET LLC (September 2010)*, 2010.

[8] M. Papadaki and S. Furnell, "Ids or ips: what is best?" *Network Security*, vol. 2004, no. 7, pp. 15–19, 2004.

[9] M. Spear, "Industrial cyber security 101," *Honeywell Users Group Europe, Middle East and Africa*, 2015.

[10] H. Holm, M. Karresand, A. Vidström, and E. Westring, "A survey of industrial control system testbeds," in *Secure IT Systems*. Springer, 2015, pp. 11–26.

[11] M. H. Cintuglu, O. A. Mohammed, K. Akkaya, and A. S. Uluagac, "A survey on smart grid cyber-physical system testbeds," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 446–464, 2017.

[12] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems securitya survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, 2017.

[13] Z. Drias, A. Serhrouchni, and O. Vogel, "Analysis of cyber security for industrial control systems," in *2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC)*. IEEE, 2015, pp. 1–8.

[14] Y. Gao, Y. Peng, F. Xie, W. Zhao, D. Wang, X. Han, T. Lu, and Z. Li, "Analysis of security threats and vulnerability for cyber-physical systems," in *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology*. IEEE, 2013, pp. 50–55.

[15] R. M. Lee, M. J. Assante, and T. Conway, "German steel mill cyber attack," *Industrial Control Systems*, vol. 30, p. 22, 2014.

[16] M. Abrams and J. Weiss, "Malicious control system cyber security attack case study–maroochy water services, australia," *McLean, VA: The MITRE Corporation*, 2008.

[17] D. U. Case, "Analysis of the cyber attack on the ukrainian power grid," *Electricity Information Sharing and Analysis Center (E-ISAC)*, 2016.

[18] T. H. Morris and W. Gao, "Industrial control system cyber attacks," in *Proceedings of the 1st International Symposium on ICS & SCADA Cyber Security Research*, 2013, pp. 22–29.

[19] "SCADAhacker", "Scada hacker's toolset," 2019. [Online]. Available: https://www.scadahacker.com/tools.html

[20] A. Almadhor, "A survey on generic scada simulators," *International Journal of Computer Applications*, vol. 128, no. 8, pp. 38–43, 2015.

[21] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, vol. 9, no. 3, pp. 49–51, 2011.

[22] T. Chen and S. Abu-Nimeh, "Lessons from stuxnet," *Computer*, vol. 44, no. 4, pp. 91–93, 2011.

[23] M. F. P. S. (Firm), *Global Energy Cyberattacks:" Night Dragon"*. McAfee, Incorporated, 2011.

[24] Z. Dehlawi and N. Abokhodair, "Saudi arabia's response to cyber conflict: A case study of the shamoon malware incident," in *2013 IEEE International Conference on Intelligence and Security Informatics*. IEEE, 2013, pp. 73–75.

[25] R. Samani and C. Beek, "Updated blackenergy trojan grows more powerful," 2016.

[26] B. Morrow, "Byod security challenges: control and protect your most sensitive data," *Network Security*, vol. 2012, no. 12, pp. 5–8, 2012.

[27] S. Bhatia, N. Kush, C. Djamaludin, J. Akande, and E. Foo, "Practical modbus flooding attack and detection," in *Proceedings of the Twelfth Australasian Information Security Conference-Volume 149*. Australian Computer Society, Inc., 2014, pp. 57–65.

[28] R. Amoah, S. Camtepe, and E. Foo, "Formal modelling and analysis of dnp3 secure authentication," *Journal of Network and Computer Applications*, vol. 59, pp. 345–360, 2016.

[29] Z. Drias, A. Serhrouchni, and O. Vogel, "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, 2015, pp. 1–6.

[30] M. Chawki, A. Darwish, M. A. Khan, and S. Tyagi, "Injection of malicious code in application," in *Cybercrime, Digital Forensics and Jurisdiction*. Springer, 2015, pp. 39–51.

[31] P. Kamal, A. Abuhussein, and S. Shiva, "Identifying and scoring vulnerability in scada environments," in *Future Technologies Conference (FTC) 2017*, 2017, pp. 845–857.

[32] M. Long, C.-H. Wu, and J. Y. Hung, "Denial of service attacks on network-based control systems: impact and mitigation," *IEEE Transactions on Industrial Informatics*, vol. 1, no. 2, pp. 85–96, 2005.

[33] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: risk assessment, detection, and response," in *Proceedings of the 6th ACM symposium on information, computer and communications security*. ACM, 2011, pp. 355–366.

[34] B. Zhu, A. Joseph, and S. Sastry, "A taxonomy of cyber attacks on scada systems," in *2011 International conference on internet of things and 4th international conference on cyber, physical and social computing*. IEEE, 2011, pp. 380–388.

[35] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, "The 2015 ukraine blackout: Implications for false data injection attacks," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3317–3318, 2017.

[36] D. Ding, Q.-L. Han, Y. Xiang, X. Ge, and X.-M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674–1683, 2018.

[37] T. Fleury, H. Khurana, and V. Welch, "Towards a taxonomy of attacks against energy control systems," in *International Conference on Critical Infrastructure Protection*. Springer, 2008, pp. 71–85.

[38] K. Stouffer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ics) security," *NIST special publication*, vol. 800, no. 82, pp. 16–16, 2011.

[39] S. Sridhar, A. Hahn, and M. Govindarasu, "Cyber attack-resilient control for smart grid," in *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*. IEEE, 2012, pp. 1–3.

[40] Z. Basnight, J. Butts, J. Lopez Jr, and T. Dube, "Firmware modification attacks on programmable logic controllers," *International Journal of Critical Infrastructure Protection*, vol. 6, no. 2, pp. 76–84, 2013.