Enabling Multi-Layer Cyber-Security Assessment of Industrial Control Systems through Hardware-in-the-Loop Testbeds

Anastasis Keliris[1], Charalambos Konstantinou[1], Nektarios Georgios Tsoutsos[1], Raghad Baiad[2], and Michail Maniatakos[2]

[1]New York University School of Engineering, New York, USA
[2]New York University Abu Dhabi, Abu Dhabi, UAE

Abstract

Industrial Control Systems (ICS) are under modernization towards increasing efficiency, reliability, and controllability. Despite the numerous benefits of interconnecting ICS components, the wide adoption of Information Technologies (IT) has introduced new security challenges and vulnerabilities to industrial processes, previously obscured by the systems' custom designs. Towards securing the backbone of critical infrastructure, selection of the proper assessment environment for performing cyber-security assessments is crucial. In this paper, we present a layered analysis of vulnerabilities and threats in ICS components, that identifies the need for including real hardware components in the assessment environment. Moreover, we advocate the suitability of Hardware-In-The-Loop testbeds for ICS cyber-security assessment and present their advantages over other assessment environments.

I. Introduction

Industrial Control Systems (ICS) are systems used in industrial environments for interconnecting, monitoring and controlling physical processes. The generic term ICS encompasses several types of industrial automation and control systems. Depending on the nature and topology of the industrial process, two major types of ICS are [1]:

- 1) Supervisory Control and Data Acquisition (SCADA) for centralized control over large geographic areas
- 2) Distributed Control Systems (DCS) in the case of decentralized distributed subsystems, each responsible for their own local process

A generic ICS with typical components (controller, sensors, actuators, Human-Machine Interface) is depicted in Fig. 1.

The industrial environments controlled and monitored by ICS are often parts of critical infrastructure. Examples of such industrial sectors are oil and gas, electric power, chemical plants, factories, refineries, waste-water treatment facilities and nuclear plants. The control of such industrial processes is safety-critical, as disruptions in operation can have devastating effects. They can lead to environmental disasters, impinge public health and safety, and have severe financial implications [2].

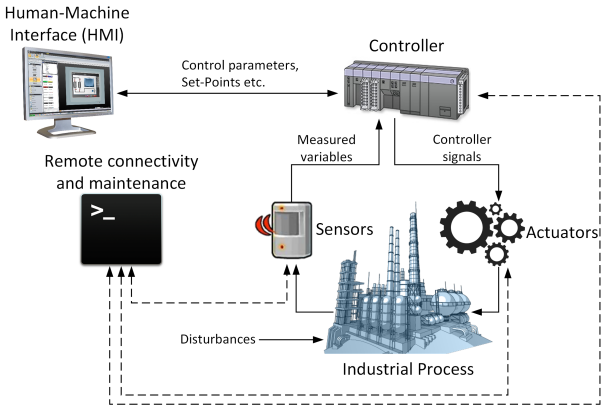

Fig. 1. Generic ICS architecture

Over the past years, industrial environments and critical infrastructure are under major restructuring [3]. The main reason behind this trend is the promising potential for increased efficiency, reliability and lower production and maintenance costs stemming from the inter- and intra-connectivity of ICS components. The enablers of industrial systems modernization are the computation and networking abilities of the comprising components. ICS nowadays include embedded microprocessors, and "smart" sensors and actuators. These components offer a multitude of advanced features, such as web-servers with GUIs for monitoring and configuration, FTP access and remote maintenance capabilities [4]. The majority of these features are made possible by the use of Commercial-Off-The-Shelf (COTS) devices and open standards, shifting away from traditionally custom and proprietary designs. Modern ICS include common microprocessor-based designs (e.g. ARM-based) and widely deployed operating systems such as VXworks and Unix-based Real-Time Operating Systems (RTOS), sup-

porting a plethora of communication protocols [5].

While the shift to COTS components offers stable, readily available and robust hardware and software, it vastly increases the attack surface against ICS [6]. The pervasive integration of Information Technologies (IT) and Internet Protocol (IP) devices, introduces new vulnerabilities in these safety-critical processes. For example, new vulnerabilities found in the COTS components can be promptly ported to industrial environments. Standardized and open IT protocols used for ICS communications have known vulnerabilities which enable direct use of existing exploitation techniques. The case is similar for IT software and hardware deployed in ICS. This allows for orchestration of elaborate attacks. Moreover, the interconnection of the corporate and control networks breaks the assurances of previously considered "air-gapped" networks and increases the accessibility of the control system.

The cyber-security threat to ICS is rising, as evident by the increasing number of cyber-attacks against such systems [7]. In 2014, the ICS Cyber Emergency Response Team (ICS-CERT) received and responded to 245 security incidents for U.S. ICS [8]. Dell's Annual Threat report shows that global SCADA attack efforts increased from 91,676 in January 2012 to 163,228 in January 2013, and 675,186 in January 2014 [9]. The situation is exacerbated by the fact that reported incidents are just a fraction of the actual attacks. A lot of incidents go unreported, either because the operators do not realize their infrastructure has been, or is, under attack, or because corporations do not report attacks, especially in extortion cases, in fear of hurting their public image and losing their customer base. The actual cost of cyber-attacks against ICS to the global economy is estimated to be more than $400 billion [10]. In response, the ICS cyber-security market is expected to grow to $8.73 billion by 2019 [11].

Comprehensive and thorough cyber-security assessment of ICS is imperative given the criticality of the processes controlled by these systems. A common approach for ensuring ICS resiliency, performing penetration tests, and validating mitigation techniques is the use of evaluation testbeds. In this paper we enumerate possible attack paths and existing vulnerabilities for the different layers of ICS architecture. This layered analysis provides insights to the requirements of the evaluation environment. More specifically, it highlights the need to include real hardware components in the testbed. We thus advocate the suitability of Hardware-In-The-Loop testbeds and present the advantages they offer in cyber-security assessment environments.

The rest of the paper is organized as follows: Section II presents the main differences between conventional IT systems and ICS cyber-security and selected cyber-attacks on ICS. A layered analysis of ICS components and cyber-security threats and vulnerabilities for each layer are presented in Section III. ICS cyber-security assessment is discussed in Section IV along with a description of Hardware-In-The-Loop testbeds. Section V concludes the paper.

II. ICS Cyber-Security

A. ICS Security vs. IT Security

Traditional IT systems and ICS share many similarities, as the former increasingly adopt IT solutions to enhance their performance and efficiency. However, the security goals of these two systems differ given the nature and purpose of ICS, rendering security solutions designed for IT systems not adequate to protect ICS [12, 5]. The main differences of IT and ICS security goals are listed below:

Physical interaction: ICS are cyber-physical systems, where control signals sent to actuators lead to physical actions that interact in a complex manner with the industrial process. In contrast, typical IT systems do not have physical interactions.

Security Objectives: ICS security focuses on availability of the industrial components and integrity of the industrial process, whereas IT security focuses on confidentiality and integrity of data.

Software changes: Software updating in IT systems can be done frequently and in an automated manner. In the case of ICS, the continuity of industrial processes does not allow downtime for software changes. In addition, exhaustive testing is required to ensure correctness of the update.

Communication Protocols: Networks in IT systems use well-known open protocols. ICS communications on the other hand, typically use a variety of different non-standard protocols, usually proprietary.

Component Lifespan: Typical lifetime of IT systems is approximately 3-5 years. On the contrary, ICS are designed with a lifespan of 15-20 years. This is also one of the reasons behind the large number of legacy devices in the field.

Criticality of response time: Response to emergencies is safety-critical in the case of ICS - information flow must not be interrupted or obstructed (e.g. authentication delays). IT systems do not have critical emergency interactions and can implement stricter access control mechanisms.

B. ICS Cyber-Attacks Examples

A significant number of cyber-security related incidents in ICS have been reported up to date. A large number of these cyber-attacks significantly affected the production process and safety of ICS. We elaborate on two well-known and high impact attacks, which resulted in physical damages:

Stuxnet: The most sophisticated malware targeting an ICS is the Stuxnet worm, discovered in 2010 [13]. Stuxnet is believed to be a state-developed cyber-weapon that infected at least 14 industrial sites in Iran, including a

uranium-enriching plant. It targeted Windows computers that had Step7 installed - a specific software for programming Programmable Logic Controllers (PLCs). The worm reconfigured PLCs in control of the spin speed of centrifuges. By manipulating the spin speed, attackers destroyed several centrifuges. Since this cyber-attack, ICS-targeting malware like Duqu, Flame and Gauss have been found, sharing many similarities to Stuxnet [14].

Baku-Tbilisi-Ceyhan (BTC) incident: In 2008, the oil pipeline of BTC in Turkey faced a cyber-attack [15]. The attackers gained access to the pipeline's control and monitoring system by exploiting vulnerabilities in the camera communication software, disabled the alarm system and manipulated the pressure of the pipeline. The result was an explosion on the pipeline that had severe financial implications and environmental impacts. More than 30,000 barrels of oil were spilled in an area above a water aquifer, while the incident cost British Petroleum (BP) and its partners $5 million/day in transit tariffs.

Cyber-attacks against ICS have targeted various industry sectors. In 2000, a disgruntled employee used inside knowledge to attack a sewage treatment plant and pump 800,000 liters of sewage in a river in Queensland, Australia [16]. The Nachi virus shut down production in a U.S. petrochemical company for five hours in 2003 [17]. The Zotob worm infected automotive manufacturing plants in the U.S. in 2005 [18]. Slammer infected the computer system of an Ohio nuclear plant in 2003, disabling the monitoring system for almost five hours [19]. More recently, in 2014, attackers used an ICS-targeting water-holing attack and attempted to extract sensitive process data using the Havex Trojan [20].

III. ICS Architecture Layers

Towards gaining better understanding of the requirements of a cyber-security assessment, we dissect ICS architecture into its comprising layers. These are the hardware layer, the firmware layer, the software layer, the network layer and finally the industrial process layer, as depicted in Fig. 2. In this layered organization, each layer builds on top of the previous one. In addition, compromising one layer can lead to compromising all the layers above it. For example, an attacker that gains control of the network can send spoofed measurements to the controller, thus controlling the industrial process to preference. Similarly, an attacker exploiting a firmware vulnerability or modifying the firmware can gain control of the layers above (e.g. attack the software implementation of a network protocol and send malicious commands to the controller, thus affecting the process). A discussion of each layer's characteristics, possible threats and vulnerabilities follows, summarized in Table I.

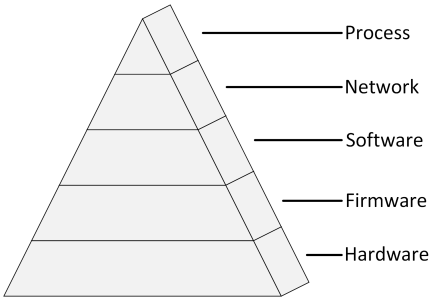

Fig. 2. ICS architecture layers

A. Hardware Layer

The hardware layer of an ICS in the cyber-security context includes the microcontrollers, microprocessors and other electronic equipment. The role of the hardware layer is crucial, as other layers rely and build upon it to perform all operations. A possible attack vector against this layer is physically manipulating the hardware. However, that requires prolonged physical access to the device and in-depth knowledge of the specifics of the system and its architecture.

Another attack vector is via the hardware supply chain. Due to supply chain globalization, it is possible for an adversary to inject malicious logic, i.e. Hardware Trojans, in a design during one of the fabrication stages [21]. Examples of malicious logic are backdoors that enable remote control of the device and/or leak confidential information and time-bombs that disable the circuit or cause performance degradation at a given time or when a condition is activated [22].

The computation results of hardware devices can be manipulated using fault injection attacks [23]. Fault injection attacks deliberately inject faults during the device's normal operation. The faults can be transient or permanent, depending on the fault injection method. They also range in accuracy. Low accuracy fault injection attacks can be achieved by manipulating the environment temperature of the device, its power supply and clock, or using electromagnetic (EM) pulses, among others. More accurate attacks require more sophisticated and costly tools, such as focused ion beams, but can in order achieve a granularity of single bit-flips [24]. A down site to these attacks is that some are destructive to the circuit and require physical proximity to the hardware device.

Information can be extracted from the device through its side-channels [25]. Side-channel attacks target the physical implementation of an algorithm or process and exploit information leakages. The timing and power consumption are typical side channels, that led to timing attacks and simple and differential power analysis (SPA and DPA) attacks against cryptographic algorithms [26, 27]. Other side-channels are the EM emanations of a device and its acoustic vibrations [28].

B. Firmware Layer

The firmware layer sits between the hardware and software layers and acts as a bridge between them. The firmware supports higher level operations and controls the basic functionality of the device, including communications, execution of compiled binary programs and device initialization. The large degree of control it offers makes it an attractive target for an attacker. Firmware can be exploited through reverse engineering. There are three basic steps in reverse engineering a firmware image [29]:

Firmware image acquisition: The binary can be obtained via vendor websites, social engineering or extracted from the JTAG port. If all the above fail, it can be extracted from the device's ROM memory by desoldering it and directly reading its contents.

Binary analysis: This step tries to get information about any file-systems, presence of encryption or checksum fields etc. Visual inspection, binwalk and binary differentials are typically used tools and methods.

Binary disassembly: In this step, the target processor architecture is identified and ASCII strings examined. Disassembly software like IDA Pro can then be used to get information regarding the functionality of the firmware image [30].

By reverse engineering a firmware image, attackers can get insight to the low-level details of the device. They can then identify weaknesses in the firmware, such as poor implementations of protocols, possible candidates for buffer overflow attacks and improper authentication techniques [31]. Following the reverse engineering process, it is possible to modify the firmware image by adding malicious code [32].

PLCs, the main control components in an ICS, are typically microprocessor-enabled devices controlled by firmware. Although firmware updates can be disabled, the majority of PLC vendors support them for facilitating the application of bug patches and security upgrades. However, where vendors see features, attackers see entry points. Firmware analysis and modification attacks against embedded devices and PLCs have been carried out successfully in the literature: 38 previously unknown vulnerabilities were discovered in more than 693 firmware images of embedded devices in [33]. The firmware update protocol and checksum algorithm of Wago PLCs were reversed engineered in [34]. Through firmware analysis, hardcoded credentials for a backdoor in Siemens PLCs was demonstrated in BlackHat 2011 [35]. Moreover, the authors of [36] successfully uploaded modified firmware over the network to DCS controllers, enabling arbitrary code execution.

C. Software Layer

At the software layer, the attack surface consists of the Human-Machine Interfaces (HMI) and the terminal systems used to interact with the actual industrial process (top layer). The software that runs on the terminals and implements the corresponding interfaces is deployed over commodity operating systems and thus is prone to wide range of vulnerabilities. Spanning from corruption and disclosed vulnerabilities in the program memory space, to kernel exploits and command injection, such attacks can alter the legitimate control flow of the software running on the system and give full control of the industrial process to adversaries. Recent attack examples applicable to this layer include Return Oriented Programming (ROP) and variants (ret2libc, call oriented, jump oriented, ROP without returns and counterfeit object oriented programming) [37, 38], ret2dir attacks [39], Just-In-time ROP [40], format string attacks and printf-oriented programming [41], pointer violations, integer overflow attacks as well as heap/stack overflows.

Besides attacks applicable to the ICS software itself, additional attacks are possible, especially in case web browsers are used on the same system (as part of the human-machine interaction or as part of multipurpose workstations). For example, cross site scripting attacks (XSS) would allow adversaries to steal session cookies, disclose host files, install further malware, or alter the information send and received through the browser. According to a recent analysis [42], from 1988 to 2012, 35% of all CVE security advisories were either XSS (13%), memory corruptions (14%), or code and command injections (8%), which highlights the prevalence of XSS and memory corruptions. The impact of memory corruptions is emphasized considering that about 45% of "critical" CVEs were either memory corruptions (35%), code and command injections (8%) or format string vulnerabilities (2%).

Exploiting the underlying vulnerabilities at this layer can be accelerated and simplified using specific attack frameworks. Typical examples include Metasploit [43], the Browser Exploitation Framework [44], Xenotix [45], Core Impact [46], while reconnaissance of potential targets is accelerated by vulnerability scanners such as Nessus or Acunetix [47]. Using such frameworks, the time and effort invested by adversaries can be minimized, increasing the attack risk if the ICS software layer is vulnerable. An alternative to reconnaissance, where adversaries attempt to discover know vulnerabilities, is to "purchase" zero-day exploits anonymously from the darknet markets [48, 49]. Since these exploits are unknown to software developers, they are expected to remain unpatched for extended periods of time, while the buyers can be governments/government organizations, security intelligence companies or well-funded individuals.

D. Network Layer

The interaction of ICS components is done via the network layer. In this layer, vulnerabilities can be introduced by its composing elements. Entry points include firewalls, modems and communication equipment, the fieldbus net-

work and communication protocols and their implementations [5]. For example, incorrectly configured firewalls can allow unauthorized access to the control network via the corporate network and the Internet.

There is a large number of ICS communication protocols and deployment varies per industry sector. A report by SANDIA presents an overview of the most commonly used communication protocols per sector [50]. As an example, the most widely used protocols in the electric sector are IEC 60870-5, DNP3, Modbus, FOUNDATION Fieldbus and ICCP. Some of the protocols used in ICS have known vulnerabilities or are insecure by design, while others are proprietary, offering security-through-obscurity assurances. In particular, two widely used protocols in SCADA systems, Modbus and DNP3, have several identified vulnerabilities [51, 52]. The lack of confidentiality, integrity, authentication and antireplay mechanisms in these protocols is the main source of their vulnerabilities. The absence of these mechanisms enables a variety of network attacks, such as Man-in-the-Middle attacks, unauthorized code execution, Denial-of-Service (DoS) and replay attacks.

The extensive use of COTS led to the porting of serial protocols over TCP/IP, for interoperability reasons. However, this has expanded the attack surface of the network layer, by including TCP/IP attacks. In tandem with the less frequent updating schedule of ICS components due to the real-time requirements of ICS, the security threat is increased. Furthermore, several ICS are connected directly to the Internet via TCP/IP as evident by the Shodan search engine. This facilitates reconnaissance and ICS targeting. A study used Shodan to distinguish PLCs connected to the Internet and obtain their PLC code by carefully crafting network requests [53].

E. Process Layer

The interaction of all aforementioned layers contributes in the implementation of industrial processes, apposite to the tasks and goals of the ICS. Attacks on the process layer manipulate the process state, by modifying control logic and process variables and constants. Their aim can be deteriorating the quality of the end product, damaging ICS components causing operation disruptions, DoS and performance degradation. Carefully crafted attacks can go undetected, especially if they do not force components to operate outside their acceptable boundaries.

A prominent example of a process-aware attack is Stuxnet, described in Section II. Another example is the Aurora vulnerability, developed in the Idaho National Lab [54]. The attack causes a diesel generator to explode by rapidly opening and closing the generator's breakers. Furthermore, false data injection attacks in power grids can manipulate the state estimation used in system monitoring, and have a significant impact on the power system [55]. Towards automating process attacks, the authors of [56] have developed an automatic payload generator for PLCs that lowers the prerequisite knowledge required to launch an attack against an ICS. Attacks and their effects to the process layer is a new area of research, where typically an ICS model is used to assess the attacks' effectiveness and impact [57].

TABLE I
SUMMARY OF ICS LAYERS THREATS & VULNERABILITIES

Layer	Vulnerabilities & Threats
Hardware	• Hardware Trojans • Fault Injection Attacks • Side-Channel Attacks
Firmware	• Firmware reverse engineering • Firmware vulnerabilities • Firmware modifications
Software	• Memory corruption & control flow attacks • Web attacks on multipurpose workstations • Zero-day vulnerability markets
Network	• Firewall misconfiguration • Protocol vulnerabilities • Internet-facing ICS
Process	• Process-aware manipulation of control logic & process variables • False data injection attacks • Automatic payload generation

IV. ICS CYBER-SECURITY ASSESSMENT

Cyber-security assessment aims in identifying vulnerabilities and risks of a system and its procedures. For ICS, the assessment methodology is different from IT systems since the two systems have different cyber-security objectives, as explained in Section II. The assessment requires careful planning and implementation in order to capture the different layers of ICS architecture in a cost-efficient and comprehensive manner. There are several steps that must be followed in the assessment process, such as documentation analysis, resources prioritization, vulnerabilities identification, assessment environment, validation testing, threat mitigation and monitoring [58]. In this paper we focus on the assessment environment, typically referred to as testbed.

In IT systems, assessment methodologies that include the production environment, or test individual components can be effective. On the contrary, the interaction of ICS with the physical world renders this approach inefficient and inherently hazardous. Consequently, we only discuss lab-based setups.

Designing a testbed has several challenges, stemming from the nature of ICS. The complex behavior of the system must be as closely as possible captured in the assessment environment, both for operational and non-operational conditions. The testbed should address scal-

ing, since it is a scale down model of the physical system. A crucial requirement is the support of both legacy and modern components and protocols, enabling realistic security and compatibility testing. Cost is also a challenge. The best assessment environment is a complete duplication of the original ICS, however that is not economically feasible nor flexible. Other considerations that must be taken into account are system flexibility, IT boundaries, configuration settings, etc.

Testbeds are necessary for studying, understanding and improving the cyber-security of ICS. Furthermore, they can be used as platforms for ensuring the interoperability of components or training engineers. A large number of industrial testbeds exist for a variety of purposes and industries. For example, over 35 smart-grid applicable testbeds have been developed in the U.S. [59]. For domains other than security (e.g. testing), testbeds are already an indispensable tool to the control industry.

The assessment environment can be solely comprised of software models and simulations [60]. This approach has the main advantage of keeping the design cost low. However, software-only techniques with models and simulations fail to capture the complexity and recreate the real-world conditions of physical systems, since they include only one of the architectural layers. In addition, a software simulation can not include every possible state of the cyber-physical system [61]. The hardware platform on top of which the virtual assessment environment executes may also restrict the analysis. Furthermore, the software simulator itself can introduce delays and simplifying assumptions, like the theoretical implementation of communication protocols. In conclusion, the assessment results obtain with a software-based testbed heavily depend on the simulation model's accuracy and quality and in most cases will give its users a false sense of security. For these reasons, software-only environments are not a frequently adopted tactic.

From the above discussion, it is clear that comprehensive cyber-security assessment requires the presence of real hardware in the assessment environment. Including hardware components in the testbed can capture the information flow throughout the layers of the ICS architecture. In particular, the bottom layers of firmware and hardware can not be otherwise represented in the testbed.

A. Hardware-In-The-Loop Testbeds

The requirements of ICS make Hardware-In-The-Loop (HITL) testbeds attractive as assessment environments. HITL methodologies for ICS connect the system in a loop, as shown in Fig. 3. One (or more) hardware components, typically a PLC, are connected to a host computer that runs the simulation environment. This connection can be done with a Serial Interface Board (SIB) that acts as a hardware translation layer and converts digital to analog signals and vice versa for communication between the

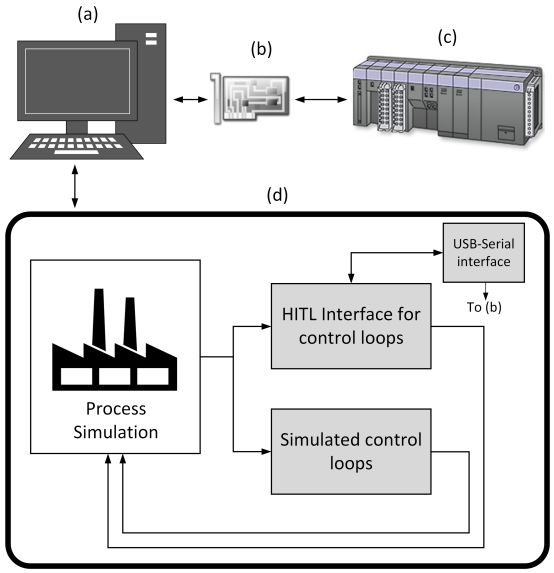

Fig. 3. HITL testbed configuration. a) Host PC running simulation model, b) Serial interface board, c) PLC and d) Simulation environment

PLC and the host PC. The system dynamics can be captured by replacing the physical part and dynamics of the system (actuators, sensors, industrial process) with a simulation. That simulation can for example be a Simulink model of an industrial process. One or more control loops can be offloaded to the actual hardware PLC for control, with the host PC orchestrating the communication between the simulation model and the actual measurements, obtained from the PLC's I/O modules. A commonly used model in the literature is the Tennessee Eastman process, due to extensive existing research on this model and availability of open-source simulation models [62].

Hardware-In-The-Loop (HITL) testbeds are more effective by providing a symbiotic relationship of physical and virtual components [63, 64]. Because of the requirements of ICS, this hybrid environment can more accurately simulate real-world interfaces and represent all the layers of ICS architecture. For cyber-security assessments, the HITL environment can be successful in addressing vulnerabilities for each layer and across layers.

HITL platforms have numerous advantages, inheriting benefits from hardware, software and network tools. Foremost, they have high accuracy and provide comparable results with the real system. They achieve this in a non-destructive way, without the hazards of testing a production environment. In addition, HITL testbeds are cost effective as they combine software and hardware and reduce implementation costs compared to the hardware-only technique. They also offer flexibility as the architecture of the environment is reconfigurable and several application scenarios and components can be assessed. They are modular and can integrate multiple types of control components, as well as facilitate linkages between other interfaces and testbeds. An advantage they inherit from

software, is that the simulation is performed faster than the real physical systems. This enables extensive experimentation, especially in the case of processes with long cycle times. Moreover, network protocols and standards are integrated in the environment and can be directly evaluated and assessed.

For these reasons, HITL setups are frequently used in cyber-security assessment environments. The National Institute of Standards and Technology (NIST) has developed a cyber-security testbed to study several application scenarios for ICS [65]. The ENEL SPA testbed studies and evaluates the impact of cyber-attacks on power plants [66]. PowerCyber is a testbed that integrates communication protocols, industrial devices and software and real-time simulators, trying to capture the inderdependencies of smart-grids [67]. Furthermore, NYU has developed a cyber-security testbed for demonstrating firmware modification attacks on relay controllers [61].

V. Conclusion

The modernization of ICS greatly increases the attack surface of safety-critical operations. The growing number of cyber-attacks against ICS and the criticality of the processes controlled by them are clear indicators of the need for in-depth cyber-security assessment of these systems.

In this paper, through a layered presentation of threats and vulnerabilities of ICS architecture, we show that all layers can be targets of cyber-attacks. We advocate that the assessment environment must include hardware components, in order to capture the information flow across all layers. To that end, HITL testbeds can effectively and efficiently represent all layers and capture the complexity of the physical systems, providing useful assessment results, as well as insights towards securing ICS.

References

[1] E. H. GICSP, M. Assante, and T. Conway, "An abbreviated history of automation & industrial controls systems and cybersecurity," 2014.

[2] J. Falco, J. Gilsinn, and K. Stouffer, "IT security for industrial control systems: Requirements specification and performance testing," in *NDIA Homeland Security Symposium & Exhibition*. Citeseer, 2004.

[3] B. Galloway and G. P. Hancke, "Introduction to industrial control networks," *IEEE Communications Surveys & Tutorials*, pp. 860–880, 2013.

[4] R. Leszczyna, E. Egozcue, L. Tarrafeta, V. F. Villar, R. Estremera, and J. Alonso, "Protecting industrial control systems: Recommendations for europe and member states," 2011.

[5] K. Stouffer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ICS) security," *NIST special publication 800–82*, 2011.

[6] E. Byres and J. Lowe, "The myths and facts behind cyber security risks for industrial control systems," in *Proceedings of the VDE Kongress*, vol. 116, 2004, pp. 213–218.

[7] R. Piggin, "Industrial systems: Cyber-security's new battlefront," *Engineering & Technology*, vol. 9, no. 8, pp. 70–74, 2014.

[8] ICS-CERT, "ICS-CERT Year in Review: 2014," [Online]. Available: https://ics-cert.us-cert.gov/sites/default/files/documents/Year_in_Review_FY2014_Final.pdf, 2014.

[9] Dell Security, "Dell Annual Threat report 2015," [Online]. Available: https://software.dell.com/docs/2015-dell-security-annual-threat-report-white-paper-15657.pdf, 2015.

[10] R. Anderson, C. Barton, R. Böhme, R. Clayton, M. J. Van Eeten, M. Levi, T. Moore, and S. Savage, "Measuring the cost of cybercrime," in *The economics of information security and privacy*. Springer, 2013, pp. 265–300.

[11] Infinity Research, "Global Industrial Control Systems (ICS) Security Market," 2015.

[12] A. A. Cárdenas, S. Amin, and S. Sastry, "Research challenges for the security of control systems." in *HotSec*, 2008.

[13] N. Falliere, L. O. Murchu, and E. Chien, "W32. stuxnet dossier," *White paper, Symantec Corp.*, 2011.

[14] B. Bencsáth, G. Pék, L. Buttyán, and M. Felegyhazi, "The cousins of stuxnet: Duqu, flame, and gauss," *Future Internet*, 2012.

[15] J. Robertson and M. Riley, "Mysterious 08 Turkey pipeline blast opened new cyberwar," *Bloomberg Business*, 2014.

[16] J. Slay and M. Miller, *Lessons learned from the Maroochy water breach*. Springer, 2008.

[17] R. J. Turk *et al.*, *Cyber incidents involving control systems*. Citeseer, 2005.

[18] P. Roberts, "Zotob, PnP worms slam 13 DaimlerChrysler plants," 2005.

[19] K. Poulsen, "Slammer worm crashed ohio nuke plant network," *Security Focus*, vol. 19, 2003.

[20] D. Walker, "Havex malware strikes industrial sector via watering hole attacks," [Online]. Available: http://www.scmagazine.com/, 2014.

[21] N. G. Tsoutsos and M. Maniatakos, "Fabrication attacks: Zero-overhead malicious modifications enabling modern microprocessor privilege escalation," *IEEE Transactions on Emerging Topics in Computing*, pp. 81–93, 2014.

[22] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy hardware: Identifying and classifying hardware trojans," *Computer*, no. 10, pp. 39–46, 2010.

[23] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan, "The sorcerer's apprentice guide to fault attacks," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 370–382, 2006.

[24] A. Barenghi, L. Breveglieri, I. Koren, and D. Naccache, "Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures," *Proceedings of the IEEE*, vol. 100, no. 11, pp. 3056–3076, 2012.

[25] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The em sidechannel (s)," in *Cryptographic Hardware and Embedded Systems–CHES 2002*. Springer, 2003, pp. 29–45.

[26] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Advances in Cryptology–CRYPTO96*. Springer, 1996, pp. 104–113.

[27] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology–CRYPTO99*. Springer, 1999, pp. 388–397.

[28] D. Genkin, A. Shamir, and E. Tromer, "RSA key extraction via low-bandwidth acoustic cryptanalysis," in *Advances in Cryptology–CRYPTO 2014*. Springer, 2014, pp. 444–461.

[29] R. Santamarta, "Reversing industrial firmware for fun and backdoors i," [Online]. Available: http://reversemode.com/index.php?option=com_content&task=view&id=80&Itemid=1, 2011.

[30] J. Ferguson and D. Kaminsky, *Reverse engineering code with IDA Pro*. Syngress, 2008.

[31] I. Slochinsky, "Introduction to embedded reverse engineering for PC reversers," in *REcon Conference*, 2010.

[32] X. Wang, C. Konstantinou, M. Maniatakos, and R. Karri, "ConFirm: Detecting firmware modifications in embedded systems using hardware performance counters," in *ICCAD*, 2015.

[33] A. Costin, J. Zaddach, A. Francillon, D. Balzarotti, and S. Antipolis, "A large-scale analysis of the security of embedded firmwares," in *USENIX Security Symposium*, 2014.

[34] M. Gjendemsjø, "Creating a weapon of mass disruption: Attacking programmable logic controllers," 2013.

[35] D. Beresford, "Exploiting siemens simatic S7 PLCs," *Black Hat USA*, 2011.

[36] D. Peck and D. Peterson, "Leveraging ethernet card vulnerabilities in field devices," in *SCADA Security Scientific Symposium*, 2009, pp. 1–19.

[37] H. Shacham, "The geometry of innocent flesh on the bone: Return-into-libc without function calls (on the x86)," in *ACM Computer and Communications Security (CCS)*. ACM, 2007, pp. 552–561.

[38] F. Schuster, T. Tendyck, C. Liebchen, L. Davi, A.-R. Sadeghi, and T. Holz, "Counterfeit object-oriented programming: On the difficulty of preventing code reuse attacks in c++ applications," in *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2015, pp. 745–762.

[39] V. P. Kemerlis, M. Polychronakis, and A. D. Keromytis, "ret2dir: Rethinking kernel isolation," in *USENIX Symposium on Security (SEC)*, 2014, pp. 957–972.

[40] K. Z. Snow, F. Monrose, L. Davi, A. Dmitrienko, C. Liebchen, and A.-R. Sadeghi, "Just-in-time code reuse: On the effectiveness of fine-grained address space layout randomization," in *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2013, pp. 574–588.

[41] N. Carlini, A. Barresi, M. Payer, D. Wagner, and T. R. Gross, "Control-flow bending: On the effectiveness of control-flow integrity," in *USENIX Symposium on Security (SEC)*, 2015, pp. 161–176.

[42] Y. Younan, "25 years of vulnerabilities: 1988-2012," Sourcefire Vulnerability Research, Tech. Rep., 2013.

[43] D. Maynor, *Metasploit toolkit for penetration testing, exploit development, and vulnerability research*. Elsevier, 2011.

[44] The BeEF project, "Browser Exploitation Framework," [Online]. Available: http://beefproject.com.

[45] The OWASP Foundation, "Xenotix XSS Exploit Framework," [Online]. Available: https://www.owasp.org/index.php/OWASP_Xenotix_XSS_Exploit_Framework/.

[46] Core Security, "Comprehensive multi-vector penetration testing," [Online]. Available: http://www.coresecurity.com/core-impact-pro/.

[47] N. I. Daud, A. Bakar, K. Azmi, M. Hasan, and M. Shafeq, "A case study on web application vulnerability scanning tools," in *Science and Information Conference (SAI), 2014*. IEEE, 2014, pp. 595–600.

[48] A. Greenberg, "New Dark-Web market is selling zero-day exploits to hackers," [Online]. Available: http://www.wired.com/2015/04/therealdeal-zero-day-exploits/, 2015.

[49] V. Tsyrklevich, "Hacking Team: a zero-day market case study," [Online]. Available: https://tsyrklevich.net/2015/07/22/hacking-team-0day-market/, 2015.

[50] Sandia, "Control system devices: Architecture and supply channels overview," [Online]. Available: energy.sandia.gov/wp-content/gallery/uploads/JCSW_Report_Final.pdf, 2010.

[51] E. J. Byres, M. Franz, and D. Miller, "The use of attack trees in assessing vulnerabilities in scada systems." Citeseer.

[52] S. East, J. Butts, M. Papa, and S. Shenoi, "A taxonomy of attacks on the dnp3 protocol," in *Critical Infrastructure Protection III*. Springer Berlin Heidelberg, 2009, vol. 311, pp. 67–81.

[53] P. M. Williams, "Distinguishing internet-facing ics devices using plc programming information," DTIC Document, Tech. Rep., 2014.

[54] "The aurora attack," [Online]. Available: http://www.secmatters.com/casestudy10, 2007.

[55] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 1, p. 13, 2011.

[56] S. McLaughlin and P. McDaniel, "Sabot: specification-based payload generation for programmable logic controllers," in *Proceedings of the 2012 ACM conference on Computer and communications security*. ACM, 2012, pp. 439–449.

[57] M. Krotofil and A. A. Cárdenas, "Resilience of process control systems to cyber-physical attacks," in *Secure IT Systems*. Springer, 2013, pp. 166–182.

[58] Centre for the Protection Of National Infrastructure, "Cyber Security Assessments of Industrial Control Systems: A good practice guide," [Online]. Available: http://www.cpni.gov.uk/documents/publications/2011/2011020-cyber_security_assessments_of_ics_gpg.pdf, 2011.

[59] NIST, "Measurement challenges and opportunities for developing Smart Grid testbeds," [Online]. Available: http://www.nist.gov/smartgrid/upload/SG-Testbed-Workshop-Report-FINAL-12-8-2014.pdf, 2014.

[60] M. Krotofil and D. Gollmann, "Industrial control systems security: What is happening?" in *Industrial Informatics (INDIN)*, 2013, pp. 670–675.

[61] C. Konstantinou and M. Maniatakos, "Impact of firmware modification attacks on power systems field devices," in *Smart Grid Communications (SmartGridComm)*, 2015.

[62] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Computers & chemical engineering*, vol. 17, no. 3, pp. 245–255, 1993.

[63] H. Fathy, Z. Filipi, J. Hagena, and J. Stein, "Review of hardware-in-the-loop simulation and its prospects in the automotive area," *Ann Arbor*, vol. 1001, pp. 48 109–2125, 2006.

[64] W. Grega, "Hardware-in-the-loop simulation and its application in control education," in *Frontiers in Education Conference*, vol. 2, 1999, pp. 12B6–7.

[65] NIST, "A Cybersecurity Testbed for Industrial Control Systems," [Online]. Available: http://www.nist.gov/customcf/get_pdf.cfm?pub_id=915876, 2014.

[66] I. N. Fovino, M. Masera, L. Guidi, and G. Carpi, "An experimental platform for assessing scada vulnerabilities and countermeasures in power plants," in *Human System Interactions (HSI)*, 2010.

[67] A. Hahn, A. Ashok, S. Sridhar, and M. Govindarasu, "Cyber-physical security testbeds: Architecture, application, and evaluation for smart grid," *IEEE Transactions on Smart Grid*, 2013.