Cyber-Physical Systems Security Education Through Hands-on Lab Exercises

Charalambos Konstantinou, *Member, IEEE*

***Abstract*—As information and communication technologies become more embedded into our daily life, cyber-physical systems (CPS) have attracted extensive attention from industry, academia, and the government because of their wide impact to the environment, economy, and society. One key problem in the area is the separation of cybersecurity education in different science and engineering disciplines. Cybersecurity expertise typically comes from the computer science domain lacking engineering practices, while engineers often do not understand cybersecurity in-depth. Hence, training and education in CPS security has become a priority in order for students to acquire more knowledge and skills in this critical field. Course material delivered in traditional lecture formats may not grab the attention of students. We foster interest in cybersecurity education as well as enable engagement and participation in this area through hands-on lab exercises which encourage experimentation in the context of CPS security. We plan to improve these labs in the next years and release them to interested institutions.**

***Index Terms*—Cyber-physical systems security, education, lab exercises.**

I. Introduction

CYBER-PHYSICAL systems (CPS) are largely referred to as "the next generation of engineered systems with the integration of communication, computation, and control to achieve the goals of stability, performance, robustness, and efficiency for physical systems" [1]. CPS are rapidly applied in critical domains such as smart grids, desalination plants, and chemical factories. The integration of CPS into critical infrastructure has reduced the isolation of physical systems and increased their connectivity to the outside world. In addition, CPS often incorporate commercial off-the-shelf products. This opens up a new range of opportunities for attackers. Similar to information technology systems, the threat landscape of CPS is in a constant state of evolution. Products upon which such critical systems depend are not effective against malicious attacks, leaving them vulnerable to adversary interference.

Due to CPS complexity and interconnections, attack strategies often employ multi-layer approaches. The security assessment of physical processes requires the identification of the vulnerability sources. Attacks on the hardware and firmware of embedded devices (e.g., programmable logic controllers (PLCs)) that control CPS aim to cause failure to the CPS process (e.g., denial of service, reconfigure parameters, etc.). CPS employ also a variety of software packages. Vulnerabilities in software interfaces may range from buffer overflows to poor control access and insecure authentication. Every CPS implementation contains a number of hardware and software systems interacting on a network layer. Vulnerabilities within the network can be due to implementation or specification flaws in the communication protocols (e.g., BACNET, MODBUS, and DNP3). In addition, vulnerabilities may emerge on the operation and process layer of the CPS. Adversaries could compromise sensors measurements in order for monitoring and protection decisions to produce inaccurate results.

There is a significant shortage of security professionals in the area of CPS able to understand and effectively thwart the growing cyber-threats. This dearth of competent and diverse professionals is a recognized problem. More than 3,400 ISACA members in 129 countries found that 86% of respondents see a global cybersecurity skills gap – and 92% of those planning to hire expect to have difficulty finding a skilled candidate [2]. Education in CPS security needs to serve two vital but sometimes opposing interests. It must introduce students to the body of scholarship and theory, giving them the understanding they need to eventually build on existing work and make their own contributions, i.e., it needs to prepare the new generation of cybersecurity scholars able to design and implement optimal and secure CPS. At the same time, it is undeniable that most students will seek jobs in industry, and it is therefore the duty of educators to inculcate practical skills and hands-on research habits [3]. The successful instruction in security and CPS must balance these two interests, and seek out educational experiences that can serve both.

Our primary goal is to foster interest in CPS security education with hands-on lab exercises. This can enable engagement and participation in this field, especially from underrepresented and minority groups. Women make up just 11% and minorities are less than 12% of the cyber-workforce [4]. Furthermore, the separation of cybersecurity education in the computer science area from that of CPS education in engineering disciplines, often results in misunderstanding between professionals. As a result, they typically remain divided within the corporate enterprise impeding the implementation of cybersecurity solutions. To address this problem, a new educational course in CPS security has been developed that incorporates hands-on CPS security experiments. The course is designed for graduate and senior undergraduate computer, electrical and mathematical sciences and engineering students. The goal is to educate them in the unique aspects of CPS security, expose them to fundamental security primitives, and help them understand the challenges in designing and securing CPS. A necessary prerequisite towards this direction is students to gain theoretical knowledge of CPS vulnerabilities and defense strategies via experiential learning and acquire practical skills through actively participating in hands-on exercises.

This work is supported in part by Cyber Florida under Collaborative Capacity Award #3910-1010-00-B.

Charalambos Konstantinou is with the Department of Electrical and Computer Engineering, FAMU-FSU College of Engineering and the Center of Advanced Power Systems, Florida State University, Tallahassee, FL 32310 USA (e-mail: ckonstantinou@fsu.edu).

II. Learning Objectives and Course Approach

The 'CPS security' course offers six hands-on exercises which allow students to interact in an environment which they can assess the security level of a CPS, or any part of it. The designed labs capitalize on: (1) the intellectual strengths of engineering and sciences students that are consistent with the broad suite of professional needs in the cybersecurity field, (2) the vibrant research and education program of many colleges aimed at advancing energy systems and other CPS such as those found in naval and transportation sectors, and (3) the developed methodologies able to integrate diverse philosophies to education. Towards achieving the aforementioned objectives, the exercises demonstrate attacks and defenses in CPS testbed designs at various levels of abstraction from the hardware to the process layer. These activities aim in better student learning and engagement which should over time translate into a larger and more skilled workforce. The educational purpose of the designed course has the potential to increase the size of cybersecurity workforce and reach student audiences that would not otherwise receive such education, making them consider cybersecurity as a possible career.

The six bi-weekly lab exercises give students the possibility to understand different offensive cybersecurity activities, to detect ongoing attacks, and also to perform defensive actions in the context of CPS and critical infrastructure domains. Due to the adversarial nature of cybersecurity, the study of hacking techniques facilitates student learning and help them develop stronger defensive cyber-skills. In addition, the labs can potentially increase student enrollment and expand existing curricula. The objectives of this fifteen-week semester course is to teach how security is integrated and managed within CPS applications. The course is largely self-contained and introduces the necessary technologies required for a qualitative (rather than quantitative) understanding of the security landscape of CPS. Undergraduate and graduate courses in power systems, computer architecture, and networking are preferred but not required. It is assumed that the students are familiar and have good background using C/C++ programming. Table I gives the lecture topics and associated lab assignments for each of the fifteen weeks of the course. The design, development, and implementation of the hands-on exercises cover a wide spectrum of principles and technologies along with well-developed tools, essential for both undergraduate and graduate education of CPS security.

Each lab exercise includes background material which is the prerequisite knowledge required before the hands-on demonstration. The background information is provided as part of the: (1) course lectures, (2) lab manual distributed in the beginning of the semester, and (3) lab introduction, as a presentation, during the occurrence of the lab. The exercises take place at the lab every two weeks in a three-hour window which includes the following: (1) a background material presentation, (2) lab demonstration by the instructor and lab assistant, (3) hands-on experiments by the students to collect results for the required deliverables. The students are asked to provide a report discussing the process, code, and experimental and evaluation results before the next lab.

TABLE I: Course Outline and Lab Exercises.

Week	Lecture Topic	Lab Exercise
1	Intro to CPS, Security Mindset and Ethics	No Lab
2	Design Principles, Threat Modeling	No Lab
3	CPS Security Characteristics, History, Threats	No Lab
4	Cyber-Physical Energy Systems Infrastructure	Lab 1 - Process Security: PLC Testbed
5	Cryptography, Passwords, Authentication	No Lab
6	Network Attacks and Mitigations	Lab 2 - Network Security: Penetration Testing
7	Software Vulnerabilities	No Lab
8	Security Mechanisms (Access Control, Malware Detection)	Lab 3 - Software Security: Buffer Overflow
10	Hardware Security: Supply Chain and Trojans	No Lab
11	Hardware Security: Side Channel Attacks	Lab 4 - Hardware Security: Performance Counters
12	Industrial CPS Operational Standards & Facility Tour	No Lab
13	Attacking Utilities and Smart Devices	Lab 5 - Cryptography: Side-Channel Attacks
9	Social Engineering and Open Source Intelligence	No Lab
14	Advanced Topic: Multi-Party Computation in CPS	Lab 6 - Cryptography: Multi-party Computation
15	Recap and Final Exam	No Lab

The technical objectives of each hands-on lab are outlined in the following section. Below we provide a comprehensive list of the learning goals in regards to the desired students outcomes for all laboratory experiences:

- Enhance mastery of subject matter. Lab experiences enhance student understanding of cybersecurity concepts (explore the strategies, tactics, and tools of CPS attackers and defenders) and give them real-world and hands-on experience in a live CPS testbed environment.
- Understand the complexity of CPS. Lab experiences help students to understand the multi-layer threat model of many attack incidents in the CPS domain.
- Develop practical skills. Lab exercises teach students the hands-on skills necessary to facilitate their transition into the workforce of cybersecurity professionals.
- Understand the nature of information security science and engineering disciplines. Lab experiences help students to understand how the concepts of information security can be applied in attacking and securing critical infrastructure such as the electric power grid, air traffic control system, and communication networks.
- Cultivate interest in cybersecurity. As a result of lab exercises which make the topic of CPS security "come alive", students may become interested in learning more about cybersecurity and see the subject relevant to everyday life.
- Develop teamwork abilities. Lab experiences promote students abilities to collaborate with others and understand the way red and blue teams form and coordinate.
- Understand the ethical considerations of cybersecurity. The exercises maintain a strong emphasis on the ethical conduct and professional responsibilities associated with the field.

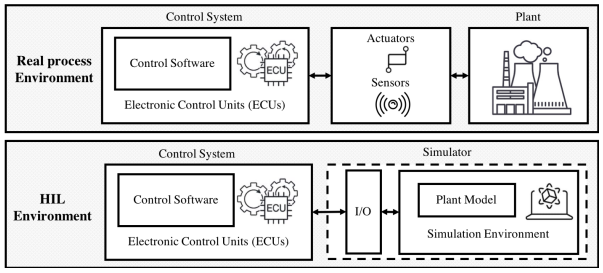

Fig. 1: Real CPS Vs. hardware-in-the-loop simulation.

III. Labs Description

The lab exercises introduce students to CPS security and give them an overview of common and state-of-the-art offensive and defense practices. The following parts elaborate on the background as well as the technical and student objectives of each lab exercise. The course materials and lab manuals will be freely available in the near future to interested institutions.

A. Lab 1 - Process and Application Security: PLC Testbed

a) Background: As shown in Fig. 1, hardware-in-the-loop (HIL) simulation aims to mimic the actual CPS behavior as closely as possible. It adds the control module in a loop. The physical process as well as the sensors and actuators are simulated in a host computer. Due to the real-time process requirements and the layer complexity of CPS, HIL simulation is often more efficient in testing the effects and the resiliency of CPS against attack scenarios. Dynamic CPS processes are characterized by the control elements, their centralized or decentralized architecture, and the underlying application behavior (e.g., discrete or continuous). While there is a plethora of feedback control structures, the proportional-integral-derivative (PID) controller is the most popular in CPS.

Process-aware attacks: use existing knowledge of the dynamic controlled process, control formulas, and operational schemes in order to cause an adverse effect in the closed-loop system [5]. Even though the attack payload is awareness-based, to access the system one shall use vulnerabilities of commercial-off-the-shelf products.

b) Lab Overview: In this lab, a CPS process-aware attack is demonstrated. The Tennessee Eastman chemical process is used as the benchmark for the dynamic and complex CPS process. Hardware PLCs are integrated in a HIL testbed in order to add realistic interference to the simulation. Students examine in the lab two process-aware attacks: sensor and actuator attacks. The reactor pressure and temperature are demonstrated as a case study for causing process instability.

PLC Vulnerability: In order to compromise the PLC, an attacker has to find an entry point that does not disrupt the normal operation of the system. The Wago 750-881 PLC is used in the HIL testbed: (1) it has the same vulnerability disclosed for other Wago systems (CVE-2012-3013), and (2) the PLC FTP service does not require any authentication.

Deliverables: Students are asked to examine the impact of process-aware controller attacks. They follow the payload delivery steps demonstrated in the lab for sensor and actuator attacks in order modify the controller PID gains. The aim is to influence the controller performance and cause deterioration of the overall process performance. Specifically, students connect to the PLC through the Ethernet port and establish a communication link after logging into the PLC using the exploited firmware-based credentials (CVE-2012-3013). Then, they download the ladder logic over FTP and modify PID variables. After calculating the checksum of new firmware image, students are asked to send the files to PLC via FTP and force-reload the boot ladder logic.

c) Student Learning Outcomes: The lab exercise allows students to develop the ability to interact with CPS components, understand the cyber and physical coupling of CPS via process-aware attacks, realize that vulnerabilities discovered in commercial-off-the-shelf products can be ported to industrial and critical CPS environments, and gain understanding of the practical applicability of PLCs in a HIL testbed.

B. Lab 2 - Network Security: Penetration Testing

a) Background: Network security is the process of protecting via physical and software methods the networking infrastructure of a system from unauthorized access, misuse, malfunction, modification, destruction, or improper disclosure. Due to the various entry points in the network layer, there is a large number of vulnerabilities connected with CPS operation at this level. Typically, networks are compromised using vulnerabilities in their hosting services and software. Entry points include control networks and protocols that link the CPS to lower-level control equipment. Network devices such as firewalls and modems can also be entry point candidates.

Network Attack Methodology: As with any ambitious attempt, a successful cyber-attack requires planning and precise execution. In network-related attacks, adversaries typically follow a series of steps: (1) Reconnaissance: information gathering; (2) Scanning: enumeration of identified hosts; (3) Vulnerability identification; (4) Exploitation: gain access into the network, get elevated privileges, perform application/web level attacks; and (5) Post-Exploitation: maintain access, remove forensic evidence, exfiltration.

b) Lab Overview: The main goal of this lab is for students to use basic exploitation techniques utilized by both penetration testers and hackers as well as understand the post-exploitation possibilities. The underlying CPS application used to motivate the lab content is supervisory control and data acquisition (SCADA) systems [6]. Students get familiar with Kali Linux, Metasploitable, Metasploit, nmap, and netcat tools in order to scan and exploit the discovered hosts such as those existing in real-world SCADA systems including substations and remote terminal units. The penetration testing tools (enumeration, vulnerability scanning, and passive/active reconnaissance) are tested within a private virtual network.

Deliverables: Students are asked to demonstrate the steps employed during a network attack: (1) scan Metasploitable machine using nmap from Kali Linux, (2) identify an open TCP port (e.g., 3632/tcp with service distccd – CVE-2004-2687), (2) use Metasploit to achieve a daemon shell, and (4) escalate privilege from daemon to root using the 141 local privilege escalation exploit via netcat (CVE-2009-1185).

c) Student Learning Outcomes: The lab helps students to understand the fundamentals of network security concepts and

mechanisms in the context of CPS, get hands-on experience on the educational practices and tools used for attacking networks and thus understand the significance of defense strategies, and learn the importance of penetration testing.

C. Lab 3 - Software Security: Buffer Overflow Vulnerability

a) Background: Buffer overflow is an anomaly where a program stores more data in a buffer than it can handle and overwrites adjacent memory locations. This could allow an attacker to take control of the system. One of the several propagation techniques used by Morris worm in 1988 was a buffer overflow. In 1996, Aleph One in Phrack magazine explained in a concise manner how it is possible to exploit vulnerabilities that arise from buffer overflows.

Protection against buffer overflows: Modern operating systems (OS) often include preventive technologies that reduce the attacker's options of exploiting a software bug, or at least reduce the chance that a program can be constructed to reliably exploit a program vulnerability. Such countermeasures include address space layout randomization (ASLR) and stack execute invalidation (NX bit). Also, there are compiler and linker protections such as StackShield and StackGuard. Higher-level programming languages can also used to disallow direct memory access. Furthermore, developers can validate inputs to prevent unexpected data from being processed and avoid using functions that allow user input to contain control information.

b) Lab Overview: In this lab, students are given a program with a buffer overflow vulnerability. Their task is to develop a scheme to exploit the vulnerability and finally to gain root privilege. In addition, the lab exercise guides students through several schemes of modern OS to protect against such attacks. Students are asked to evaluate these protection schemes. The lab is based on a modified version of the SEED buffer overflow lab [7]. The buffer overflow vulnerability is tested on a Raspberry Pi (RPi) machine. The exploited vulnerability allows students to get admin access on the machine and control a critical CPS signal. The status signal is the relay controller control signal (trip/close) as part of a controller HIL power grid testbed environment [8].

Deliverables: Students are asked to complete the tasks described in the SEED lab. In addition, a hands-on demonstration is required in order to validate the exploitation of the vulnerability in the testbed: open the circuit breaker in a HIL simulation controlled by the RPi-based relay controller.

c) Student Learning Outcomes: The lab enables students to gain first-hand experience on buffer overflow vulnerabilities and corresponding protection techniques, understand the impact of software vulnerabilities in critical infrastructure, and appreciate the importance of HIL testbeds for cybersecurity assessment in CPS environments.

D. Lab 4 - Hardware Security: Performance Counters

a) Background: Hardware performance counters (HPCs) are special-purpose registers built into the processor monitoring unit of a microprocessor for performance tuning of applications. HPCs count a variety of hardware-related activities by monitoring hardware events such as total instructions retired and branches taken. HPCs provide in-depth performance data without requiring source code modifications and with much lower overhead than software profilers. The hardware events that can be counted and the number of available HPCs vary from one processor model to another. For example, AMD Opteron, Intel Core and ARM Cortex A-15 have 4, 8 and 6 hardware counters, supporting 125, 200 and 70 events, respectively. HPC-based profiling tools have been built into all popular OS. For example, Perf is an application level HPC profiling tool for Linux.

HPCs for Security: A program is composed of a sequence of various types of instructions which can be seen as low-level operations when executed on a platform. When the sequence of instructions is changed, the occurrences of the low-level operations change accordingly. Therefore, HPCs can be leveraged for anomaly checking in order to determine if an HPC-monitored program is maliciously modified.

b) Lab Overview: In this lab, students learn to use Perf tool and leverage HPCs for anomaly detection. First, students get familiar with Perf to measure events from two different implementations of a matrix multiplication algorithm. Then, they use Perf for anomaly detection in two phases: an offline and online profiling phase. In the offline phase, a clean copy of a monitored program is executed in order to collect and store the corresponding HPC-based signature ("baseline"). In the online-runtime phase, the same HPC configuration is applied and the HPC values are collected and compared with those from the offline phase. In case the runtime HPC-signature does not match to the "baseline" reference, a malicious modification will be reported and appropriate actions can be initiated [9].

Deliverables: Students are asked to implement a malicious implementation of the Underhanded C Contest of 2009 which will be detected using HPCs. The malicious version program must inexplicably misroute a piece of luggage in the CPS design of an airport baggage handling automation system. This happens if the right kind of free text comment is provided by the check-in clerk. The students use Perf tool to measure the performance events for the golden and malicious program pair, e.g., instructions, branches, branch-misses, cache-misses, etc.

c) Student Learning Outcomes: The lab aids students to understand the significance of defensive security, learn about malware detection techniques, and demonstrate how to leverage hardware for enhancing the security of microprocessor-based devices employed in CPS environments.

E. Lab 5 - Cryptography: Architectural Side-Channel Attacks

a) Background: Cryptographic implementations are often considered as "black-boxes". It is unrealistic, however, to consider that these blocks provide perfectly secure solutions. Crypto-primitives rely on both software and hardware and hence this interaction can be monitored by adversaries to infer side-channel leaked information such as power consumption and timing information. This lab gives a practical demonstration of a cache-timing attack: monitoring the movement of data in and out of cache and analyzing the corresponding timings.

Timing Side-Channel Attack: is a side-channel attack in which the attacker attempts to compromise a system by analyzing the required time to execute cryptographic algorithms.

Advanced Encryption Standard (AES): is a symmetric block cipher that operates on a fixed block size of 128-bits and allows for key sizes of 128, 192, or 256 bits. AES has different round keys which are derived from a cipher key. The deriving process uses a schedule called Rijndael's key schedule.

AES Cache-Timing Attack: Bernstein has found that many block ciphers, specifically AES, may leak timing information during cache hits/misses [10]. Then, he demonstrated a cache-based timing attack which can reconstruct the key by observing the data flow of different cache levels. AES uses four 1024-byte tables: T0, T1, T2, and T3. In order for AES to get enough speed, those 4KB information should be stored in the cache. However, even for a large enough cache, other processes will eventually cause the cache misses. Those misses cause the encryption to occur at a variable rate.

b) Lab Overview: The experiments test cache-timing attacks on different AES-based servers employed in a healthcare CPS application domain. The goal is to extract a complete AES key from a server. The attack is demonstrated on three different servers so that students will be able to compare the results among them. The lab exercise provides a step-by-step tutorial for performing the attack in one of the servers storing personal information and medical records of patients. Students are later provided with the final results of the attacks in the other two servers in order to draw their conclusions.

Deliverables: Students are given the final results for each server test and asked to correlate the preparation and the attack phase of the experiment. Based on the correlation results, students are asked to find for which bit the range of possibilities for each key bit $k[i]$ is lower than 256. For that specific $k[i]$ index, students need to explain how the input bit $n[i]$ affects OpenSSL AES timings for $k = i$ on the assigned processor inside the targeted server, and how $n[i]$ affects OpenSSL AES timings for random selection of k.

c) Student Learning Outcomes: The lab enables participants to gain first-hand experiences on basic architectural cache-timing attacks, understand block ciphers and the basic concepts of cryptography, and use leaking side-channel data to extract a secret value used by the victim function.

TABLE II: Learning and Experience.

Category	A/A	Question	$\bar{x}$	σ
Organization and Clarity	Q1	The lab instructions are well prepared.	6.22	0.36
	Q2	I know what is expected of me in this lab. I know what the task is.	6.07	0.31
	Q3	The lab explains the procedures clearly. I understand the lab purpose.	6.15	0.28
	Q4	The lab promotes good use of hands-on and experimental time.	6.12	0.32
Enthusiasm and Intellectual Stimulation	Q5	The lab assignments are interesting.	6.38	0.35
	Q6	The lab instructions/content motivate me to do well in the lab.	6.12	0.37
	Q7	The lab reinforces what I have learned in the lecture.	5.87	0.37
Student Perceptions of Learning	Q8	The lab content advances my knowledge in this lab section.	6.08	0.29
	Q9	The lab content makes me more curious about the subject matter.	6.38	0.29
	Q10	The lab content helps me learn important techniques in this course.	6.23	0.30
Open-Ended Questions	Q11	Describe the best aspects of this lab.	–	–
	Q12	Describe changes that could be made to improve this lab experience.	–	–

F. Lab 6 - Cryptography: Multi-party Computation

a) Background: Secure multi-party computation (MPC) addresses how n parties, each with a private input x_i, $i \leq n$, can securely and jointly evaluate an n-party functionality f over x. An MPC scheme ensures that the functionality is computed correctly, and each party will not learn or infer any information from the interaction other than its output and what is inherently leaked from it.

Yao's Millionaires' Problem: Alice and Bob have a private number, a and b respectively. Their goal is to solve the inequality $a \leq b$? without revealing the values of a or b, or more stringently, without revealing any information about a or b other than $a \leq b$ or $a > b$.

Secure Multiparty Set Intersection (MPSI): In [11], Du and Atallah presented several open problems for secure MPC. One of their problems, "Privacy Preserving Geometric Computation", consists of two parties that both have a shape. The two parties wish to determine whether their shapes intersect without revealing their shapes.

Homomorphism: At the heart of at least one branch of secure MPC is the principle of homomorphism. That is, certain encryption schemes allow some mathematical operations to be performed on their encrypted values without losing the ability to recover the resulting plaintext.

b) Lab Overview: In this lab, we present a solution to at least a portion of the above MPSI problem: given two semi-honest parties, each having a line segment (within the same coordinate plane) and without revealing their line segments to each other, or any potential eavesdropper, the parties would like to know if their line segments intersect [12]. The application of a solution to this problem is presented in the context of a CPS military application in which allies wish to determine whether their aircrafts trajectories will intersect (or collide) without revealing their inputs (location and trajectories). Students are given the geometric principles for the MPSI problem, the background of Paillier cryptosystem, and also provided with the source code that solves the problem.

Deliverables: Students are asked to evaluate the computational complexity and overhead of the scheme: (1) evaluate the complexity of the protocol from Alice's perspective: calculate the number of encryptions and decryptions (required by Alice according to the protocol) as well as the number of homomorphic operations, and (2) evaluate the communication overhead of the implemented application: calculate the size of each protocol instance, i.e., how many bits are required in order for the two parties to exchange their line position.

c) Student Learning Outcomes: This lab introduces students to the concepts of trust, confidentiality, MPC, and homomorphism. It also exposes participants to crypto-protocols for solving Yao's millionaires' problem, and help them to evaluate computational complexity and communication overhead.

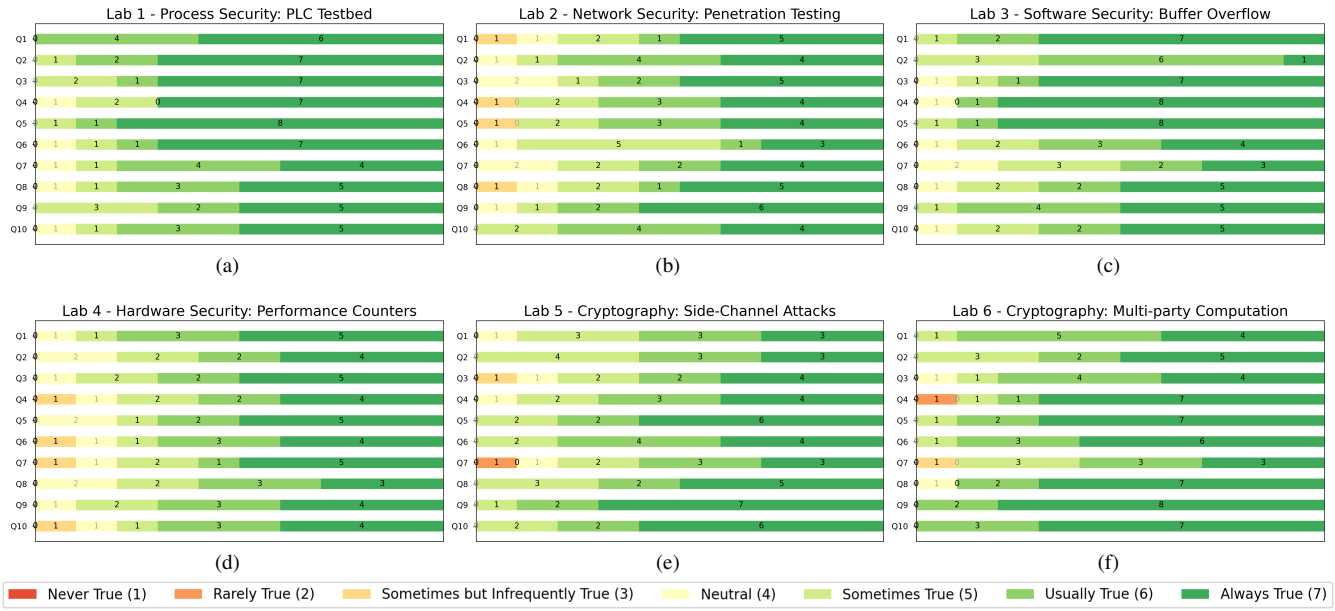

Fig. 2: Survey results by students for (a) Lab 1, (b) Lab 2, (c) Lab 3, (d) Lab 4, (e) Lab 5, and (f) Lab 6.

IV. Labs Evaluation

The exercises were rated by the students after each lab demonstration. The author designed a lab assessment survey, besides the standard course evaluation, to examine the effectiveness of the lab contents and demonstrations, determine the knowledge acquisition experience as part of the course, and measure how the lab activities contribute to students' understanding. The primary purpose of the survey is to improve future offerings of the labs and evaluate the design of the current offering. Overall, students found the additional challenge and realism of hands-on lab exercises to be valuable. Detailed questions are presented in Table II.

Table II also shows the means ($\bar{x}$) and standard deviations (σ) for questions answered by students for all labs. This is based on a sample size of 10 registered students for the first course offering. Participants responded to each statement using a 7-point Likert-type scale (1 = never true; 2 = rarely true; 3 = sometimes but infrequently true; 4 = neutral; 5 = sometimes true; 6 = usually true; 7 = always true) in terms of several pedagogical characteristics of the labs. Survey results for each lab based on Table II questions are presented in Fig. 2.

As for the open-ended questions, most of the students commented that the "lab content was interesting and instructions were clear". Also, they agreed that the labs helped them to "get insights on how theory works in practice" as well as excite them to "learn more on the topic". One student commented that the "hands-on labs were unique and something that she/he has never experienced before". Students also suggested to allow more time in the background presentation for each lab.

V. Conclusions

The described lab exercises in the context of the CPS security course introduce students to the body of scholarship and knowledge as well as help broaden participation and promote understanding of this critical area. The plan is to continue improving the labs aiming to potentially contribute towards the formation of the next generation of professionals that will have the skills to assume leadership in cybersecurity positions within the industry, the government, and the academia. This effort can also enhance education programs in the area of CPS security, help the development of a cybersecurity-based curriculum, and enhance students experiences.

Acknowledgment

The author would like to thank XiaoRui Liu, Ioannis Zografopoulos, and Ali Sayghe for their help in setting up the testbed environments for the first offering of the lab exercises.

References

[1] C. Konstantinou, M. Maniatakos, F. Saqib, S. Hu, J. Plusquellic, and Y. Jin, "Cyber-physical systems: A security perspective," in *2015 20th IEEE European Test Symposium (ETS)*, 2015, pp. 1–8.
[2] ISACA, "State of cybersecurity," [Online]: https://www.isaca.org/go/state-of-cybersecurity-2020, 2020.
[3] P. Marwedel, H. A. Andrade, T. Mitra, and M. Grimheden, "Survey on education for cyber-physical systems," *IEEE Design Test*, pp. 1–1, 2020.
[4] B. Dinah, "Women, Minorities Largely Absent from Cybersecurity Jobs," [Online]. Available: https://tinyurl.com/qv99b35.
[5] O. Koucham, S. Mocanu, G. Hiet, J.-M. Thiriet, and F. Majorczyk, "Detecting process-aware attacks in sequential control systems," in *Secure IT Systems*. Springer International Publishing, 2016, pp. 20–36.
[6] L. Roepert, M. Dahlmanns, I. B. Fink, J. Pennekamp, and M. Henze, "Assessing the Security of OPC UA Deployments," *arXiv preprint arXiv:2003.12341*, 2020.
[7] W. Du, "Buffer-Overflow Vulnerability Lab," [Online]: http://www.cis.syr.edu/~wedu/seed/.
[8] C. Konstantinou and M. Maniatakos, "Impact of firmware modification attacks on power systems field devices," in *Smart Grid Communications (SmartGridComm), Int'l Conference on*. IEEE, 2015, pp. 283–288.
[9] X. Wang, C. Konstantinou, M. Maniatakos, and R. Karri, "Confirm: Detecting firmware modifications in embedded systems using hardware performance counters," in *2015 IEEE/ACM Int'l Conference on Computer-Aided Design (ICCAD)*, 2015, pp. 544–551.
[10] D. J. Bernstein, "Cache-timing attacks on AES," 2005.
[11] M. J. Atallah and W. Du, "Secure multi-party computational geometry," in *Algorithms and Data Structures*. Springer, 2001, pp. 165–179.
[12] Y. Lindell, "Secure Multiparty Computation (MPC)," Cryptology ePrint Archive, Report 2020/300, 2020, https://eprint.iacr.org/2020/300.

Charalambos Konstantinou (S'11-M'18) is an Assistant Professor in Electrical and Computer Engineering at the FAMU-FSU College of Engineering and the Center for Advanced Power Systems, Florida State University, Tallahassee, FL, USA. He received a Ph.D. in Electrical Engineering from New York University, NY, USA, in 2018. He is the Director of the Decision & Secure Systems Laboratory (dss-lab.github.io) and his research interests include cyber-physical and embedded systems security with focus on power systems.