Teaching Industrial Control System Security Using Collaborative Projects

Thuy D. Nguyen[✉] and Mark A. Gondree

Department of Computer Science, Naval Postgraduate School,
Monterey, CA 93943, USA
tdnguyen@nps.edu, mgondree@nps.edu

Abstract. In this work, we discuss lessons learned over the past three years while supporting a graduate capstone course centered on research projects in industrial control system (ICS) security. Our course considers real-world problems in shipboard ICS posed by external stakeholders: a system-owner and related subject matter experts. We describe the course objectives, format, expectations and outcomes. While our experiences are generally positive, we remark on opportunities for curricula improvement relevant to those considering incorporating realistic ICS topics into their classroom, or those working with an external SME.

Keywords: ICS · SCADA · Ship-board ICS · Education · Capstone project

1 Introduction

As mandated by Executive Order 13636, "Improving Critical Infrastructure Cybersecurity" [6], the National Institute of Standards and Technology (NIST) published its *Cybersecurity Framework* document to provide guidelines for managing security risks that could affect the national critical infrastructure [18]. This NIST framework recognizes that information technology (IT) systems and industrial control systems (ICS) differ in term of operational environment and potential risk. It also identifies cybersecurity education as a core requirement to protect the critical infrastructure services.

The insecurity of industrial control systems (ICS) is a pressing and tangible problem, prompting the formation of the Industrial Control Systems Cyber Emergency Response Team and various working groups on critical infrastructure protection, like the Critical Infrastructure Partnership Advisory Council. In the security education community, several groups have proposed curricula to address the needs to educate students and professionals about critical infrastructure protection and ICS security [7,8,15]. SCADA (supervisory control and data

---

acquisition) security is a knowledge unit highlighted as contributing to the academic requirements for the designation as the Center of Academic Excellence in Cyber Operations from the U.S. government [19].

In keeping with the broader goal of preparing our graduates for real-world challenges, we introduced a capstone course that aligns with the observation that IT security and ICS security are different and thus teaching ICS security needs to be tailored for the ICS domain and taught as a separate course [13]. Our capstone course focuses on shipboard ICS because of its relevance to the mission of our institution.

The pedagogical approach of *learning by doing* assumes that the learner can work independently, leveraging on prior experience, to develop an effective solution to a complex problem. *Collaborative learning* and *problem-based learning* are two synergistic instructional methods in which learners work in small groups to solve real-world problems, utilizing knowledge, skills and abilities learned prior in the classroom. Our course follows these teaching strategies, while involving participation of a shipboard ICS system owner (i.e., the stakeholder) and ICS subject matter experts (SME).

In the remaining sections, we discuss previous work on ICS security education and describe the objectives and structure of the capstone course. We then enumerate the class projects that addressed a number of shipboard ICS security concerns, ranging from a comparative analysis of version control systems for ICS development to a table-top vulnerability assessment of a notional ICS target system. We highlight lessons learned and recommendations for future course enhancements.

2   Related Work

Several educators describe the need for security curricula leveraging practical laboratories for ICS security, as a motivating context for undergraduate education and for professional development. These prior efforts include the discussion of curricula [4,7,9], the design of teaching laboratories [10,16,22] and the development of various teaching modules [8,11,13]. For example, McGrew and Vaughan describe a set of exercises demonstrating software vulnerabilities associated with a commercial HMI product [14]. Generally, education research related to industrial control security has focused on the description of a specific laboratory environment or the potential hands-on course exercises using this environment. In contrast, we do not describe a target, instructional laboratory environment or exercises for use in direct-instruction courses. The course described here explores very different intentions, audiences and delivery strategies.

3   Course Description

The course presents a capstone experience in which students are immersed in an operational or policy challenge of interest to an external stakeholder. Student teams develop courses of action (COA) that address legal, ethical, political,

technical, tactical, operational and strategic implications. At the end of the course, the recommended COA is presented by students to the stakeholders. The outcome of this research is a solution to the stakeholder's technical problem. Students draw upon their classroom knowledge and research skills to analyze the problem space, derive potential solutions, and communicate these to subject matter experts and stakeholders. However, the stakeholder's problems require students exercise critical thinking in a problem domain with which they have no prior curricular experience, i.e., industrial control systems.

Program Context. Our capstone course is an upper-division class in the Cyber Systems and Operations (CSO) curriculum at the Naval Postgraduate School. The CSO program is an 18-month (six-quarter) multidisciplinary graduate program covering a broad range of cyber operations [12]: computer network attack, defense, and exploitation; cyber analysis, operations, planning and engineering; and cyber intelligence operations and analysis. The program is very practically-focused, employing site visits, wargaming exercises, seminars, guest speakers and practical workshops to complement traditional instruction. The program is designed for students with diverse, non-CS backgrounds: entrance requirements are based on a bachelor's degree in some STEM discipline.

Prerequisites. Prior to enrolling in this capstone course, students have completed most CSO program requirements including basic courses on computer security and network security. It is expected that students can use virtual machines, trial-install unfamiliar software, employ penetration testing tools and reason about systems from an adversarial perspective. Students have already started their Master's thesis projects prior to the capstone course, and are familiar with many on-campus resources for performing background research and literature reviews.

Learning Objectives. Prior courses require both individual and group work, while student thesis projects emphasize individual research; in contract, our capstone course emphasizes collaborative research for an external stakeholder. At the end of the course, students will be able:

– to collaborate on research in self-directed teams;
– to communicate in-progress research results to a technical audience;
– to interpret and respond to outside technical feedback;
– to prepare COA design alternatives;
– to evaluate alternatives from an operational perspective;
– to synthesize final technical recommendations; and
– to communicate technical recommendations to a stakeholder.

These learning objectives are understood to be quite advanced, falling in the highest levels of Bloom's taxonomy (i.e., applying, analyzing, evaluating, creating). Compared to prior group coursework, this capstone project requires significant time management (on the scale of months rather than weeks) and coordination outside the classroom (nearly all group work is performed outside of class).

4   Course Format

The capstone is offered as a four-credit class twice a year. It follows a resident course format with in-class meetings each week. Significant course work occurs in group meetings outside class time and via the course website and forums. The expected time commitment for out-of-class course work is approximately eight hours per week. By the end of the course, students demonstrate an in-depth understanding of project-related material. Grading is based on team accomplishments, SME evaluation of the final COA report and each individual's technical contribution to project tasks including oral presentations and written deliverables (e.g., correspondence with the SME, written reports).

Through a guided inquiry learning process facilitated by the instructor, students work together in small teams to develop a recommended COA to solve a problem provided by the external sponsor or SME. The course is divided into several phases (see Table 1), with phase 0 occurring approximately three months before class begins. Phases 1–5 occur over a ten-week period. Throughout phases 3 and 4, the instructor works with the SME to guide students in research and to oversee interactions between the students and the SME.

Table 1. Course schedule overview.

Phase	Purpose	Primary participants
Phase 0	Project creation	SME, instructor
Phase 1	Technology familiarization	Students, instructor
Phase 2	Initial engagement with SME	SME, students, instructor
Phase 3	Interim progress review	SME, students
Phase 4	Final progress review	SME, students
Phase 5	Project conclusion	Students

4.1   Phase 0 – Project Creation

Prior to each class, the instructor solicits real-world ICS problems from stakeholders, stated in relatively general terms. Additionally, stakeholders and SMEs may propose specific research ideas for certain problem areas. Iteratively, the SME and instructor refine the scope of work to better align with the students' technical background and the course timeframe. The final outcome of this iterative process is a project assignment.

Each project assignment has four key components: a problem statement, including the needs and goals expressed by the SME; a description of the research activities to be performed; the guidelines for team formation; and the expected deliverables from each team. Students are asked to organize their own teams, with some guidelines and final approval from the instructor. To promote the

development of new student leadership and project management skills, students who served as group leaders in prior coursework cannot serve as project team leaders. The instructor approves team membership and leader selection to ensure a balanced distribution of experience, and a good alignment between individual skills and those required for success in the team's project. This strategy has been successful: over three years, only one team (of 12 total teams) has had interpersonal conflict that required some instructor intervention.

4.2   Phase 1 – Technology Familiarization

During the first two weeks, students learn about ICS technology through a relatively traditional lecture-oriented approach, with homework and lab exercises.

Lectures. There are two introductory course modules. The ICS Fundamentals module covers system types (e.g., Distributed Control System and SCADA), components (Human Machine Interface, Programmable Logic Controller, etc.) and industrial protocols. This module explains the *security zone* and *conduit* concepts defined by the ISA/IEC-62443 (formerly ISA-99) security standards, and introduces students to shipboard control systems (e.g., steering, propulsion, electrical plant). The ICS Vulnerabilities module focuses on select ICS attacks. The main objective is to show students the similarities and dissimilarities of cyber exploits between a traditional IT systems and industrial control systems.

Homework. Module homework assignments consist of about 5–6 assigned readings, including academic papers, trade articles, SME-provided background materials and video recordings on ICS security research. For each assignment, students are asked to provide a written synopsis, including a constructive evaluation discussing the material's strengths and weaknesses in terms of reasoning, logic and evidence. A 90-minute video on PLC vulnerabilities and exploit tools [24] is a class favorite since it explains how ICS security researchers uncover design and implementation deficiencies in popular PLC products, e.g., Rockwell Automation ControlLogix and MicroLogix.

Laboratory. Students learn about ICS vulnerabilities using a "SCADA-in-a-box" lab environment [21] simulating a realistic natural gas compression system. It includes a commercial PLC, HMI software, a commercial ICS firewall and malware demonstrating a ModBus-based PLC exploit. Student exercises using this environment consist of two activities. First, students conduct an attack on the unprotected PLC using a malicious payload delivered via opening a PDF on the HMI system. Second, students add and configure a firewall for the system, allowing only select ModBus traffic between HMI and PLC to block attack traffic. Although introductory in nature, the exercise provides hands-on experience with different components and operational aspects of a SCADA system—i.e., as an operator using the HMI, as an attacker and as a security administrator. Students develop a short report explaining their understanding of the ICS components in the environment, the protection mechanisms, any problems they encounter during the activities and how they solve these. Student feedback on the exercise

has been positive, and the lab report provides an opportunity for formative feedback, to correct misunderstandings or confusions.

4.3   Phase 2 – Initial Engagement with SME

In this phase, the class assignment work begins, with team formation and background research to prepare for the first meeting with the SME. This meeting is either in-person or via video teleconference. At this meeting, students interview the SME to collect information and ask questions about scope and expected outcomes. Early interaction with students allows the SME to clarify research needs, ratify assumptions, provide insights on the operational setting and gauge students' technical strengths.

During this phase, the instructor and SME coordinate a ship tour for students to learn about the inner workings of shipboard control systems. Guided by a SME, the students can see the physical layout of various ICS equipment and gain additional knowledge on how these systems are operated and maintained. Although most students are U.S. Navy officers who have served on ships, very few have managed these types of systems. Information obtained from the trip is documented in individual trip reports, used as a basis for in-class discussions to clarify misunderstandings relevant to the project assignments. Seeing ICS systems in context during a tour is extremely valuable, highlighted in nearly all course feedback.

4.4   Phase 3 – Interim Progress Review

Students begin an iterative research process to develop the COA following a traditional prototyping systems development methodology, i.e., working versions of the COA are developed, deliberated and refined cyclically. Project management uses web-based tools through the course website. In particular, each team maintains a wiki that contains a work plan and a set of individual activity logs. Communication with the SME is via email and teleconference. This phase ends with a progress review in which students present emerging ideas and potential approaches informed by the on-going research.

Team Work Plan. This is a lightweight, free-form artifact (as opposed to the traditional work breakdown structure) describing tasks assigned to each team member, the objectives and outcomes that the team plans to accomplish weekly, and the research methodology used to complete the identified work items (interviewing the SME, reaching out to professional contacts, etc.). It is the responsibility of the team leader to update the work plan regularly to reflect changes as the project progresses. Naturally, the level of detail and freshness of this artifact depend on the project management experience of the team leader. A common trend that has been observed is that the work plan tends to lag behind actual work.

Individual Research Log. Each student maintains a running log describing their weekly accomplishments and research findings. The log acts as evidence

and an agenda for required in-class briefs where students discuss their activity and open issues. In their log, students describe the status of each assigned task, the time spent on each task, and the outcomes of each task. Students also keep track of problems encountered during the implementation of each work item, the resolution to these problems, and any rationale for technical decisions made. The soundness of the technical discussion is partially judged based on the supporting materials attached to the contents, e.g., web links to reputable sources, citation to academic papers and technical articles.

Given the variety of assignments and tasks, there is no formal rubric for the evaluation of this log. At a minimum, however, students must demonstrate the following: their understanding of the tasks, any issues to resolve, interactions with the SME, and how they respond to or incorporate advice from the SME. An interesting observation is that, for some classes, the quality of the research log was related to team competition: friendly team rivalry caused activity logs to be more complete and in-depth.

Interim Review. The class is structured to include checkpoints for the SME to review interim results and provide guidance on challenges encountered. Typically, there is only one interim review; however, if problems affect the assignment goals or research direction, an additional review of subsequent findings take place, if deemed necessary. When multiple feasible COA alternatives exist, the team and SME confer to select the most promising path.

For each review, students prepare a written report to the SME and, immediately following this, a formal presentation to the SME based on the report. Before submission to the SME, each team's report is reviewed by all members of the other team. From the regular in-class briefs and activity logs, most students have an adequate understanding of the other team's work to provide constructive comments during peer review.

4.5   Phase 4 – Final Progress Review

Teams have established their research direction by this phase, and students can concentrate on generating the final COA. The predominant activity in this phase is COA refinement, where questions to the SME are more detailed and the analyses are more focused. If the project requires implementation of some selected technology, students must demonstrate a working prototype before the final progress review with the SME.

This phase culminates in a draft final report from each team for review by the SME. The report describes the recommended COA for the team assignment and rationale for its selection over any alternatives. Procedures to build and operate the prototype are fully documented in this report. The rubric to assess the report includes the following characteristics:

– Content: purpose, literature review, technical content, critical thinking, and organization;
– Communication: tone and writing mechanics; and
– References: usage and quality.

This phase ends with a final progress review in which the SME examines the validity and feasibility of the actionable recommendations. Recommendations with solid technical analysis or prototypes will be considered for implementation while ideas that are relevant to the problem but are not fully developed will be considered for future work.

4.6   Phase 5 – Project Conclusion

During the final week, teams finalize their COA report and perform peer assessments of their own team's members. The objective of this peer assessment is to review and evaluate each team member's effort, contribution to the project and interaction with the team. The peer-assessment rubric employs a 4-point Liekert-scale, with each item accompanied by a justification explaining the score. The peer assessment is feedback to the instructor only, used as an aid in course grading decisions. The completed assessment is itself evaluated based on its fairness and usefulness to the instructor.

5   Project Description

To date, our capstone course has been offered six times across three years, involving three different SMEs at different times over this period. We provide synopses of past projects to illustrate the variety of ICS security aspects—from developmental security to operational security—addressed in the course.

5.1   Software Subversion via Portable Memory Devices

This project addressed the threat of inappropriate use of portable memory devices to introduce malicious code into a shipboard ICS environment. The threat landscape of modern shipboard ICS architectures has grown significantly because of the dissimilarity in ICS hardware and software technologies used in different ship designs. Hence, two different ships were used as case studies for this project. The assigned tasks included:

– Review existing policies and operational practices for using portable memory devices on shipboard control systems; and
– Propose changes to allow the use of these devices while safeguarding the integrity of the system.

One team examined the list control system and the other team studied the ventilation control system. The selection of these two systems exposed students to different system architectures and ICS technologies; the list control system used Allen Bradley equipment while the ventilation control system utilized Siemens equipment.

5.2 Network Security

This project investigated technologies for network isolation to control unauthorized traffic between an ICS network and the external shipboard network. The project used the same ventilation control and list control systems employed in the prior project (see Sect. 5.1) as case studies. The project assignment tasks included:

– Survey existing DMZ architectures and perimeter control technologies (e.g., firewall and intrusion detection systems) used in land-based SCADA systems;
– Propose a relevant shipboard ICS design that incorporates these technical measures; and
– Propose a concept of operations on security incident response, including detection and analysis, containment, eradication and recovery.

Teams reviewed best practices for implementing perimeter control as recommended by ICS-CERT and NIST. Students learned about preprocessors and signatures for the Snort IDS that were designed to support industrial protocols such as DNP3, Modbus and Ethernet/IP [5,20].

5.3 Protection of Multicast IPsec Messages

This project investigated the use of IPsec with manual keying to provide message authentication and replay protection for multicast communications. The target ICS used IP multicast to conserve bandwidth, i.e., status update messages could be sent to pre-defined groups of HMI systems instead of broadcasting them to all HMI systems. The tasks included:

– Survey IPsec products that support multicast;
– Make recommendations for a bump-in-the-wire (BITW) appliance;
– Make recommendations for a bump-in-the-stack (BITS) appliance;
– Propose an IPsec-based ICS design that can provide integrity and anti-replay protection for data transiting the ICS network; and
– Propose a key management design that addresses the entire life cycle of cryptographic keys and other keying material, and is resilient to unauthorized key disclosure.

Students were divided into three teams: two teams focused on BITW and BITS implementations and the third team worked on key management. A number of functional requirements were levied on the BITW implementation: fast Ethernet support (at least two ports), memory (256 MB), physical size (6" × 6" × 8"), cost ($2000), operating temperature (0–65 degrees Celsius), power (12–24V), DIN rail mountable, ruggedized. A candidate BITW appliance must also conform to the IETF RFC 5374 which extends IPsec to support multicast addressing [23].

5.4   Continuous Monitoring

This project examined the use of two security information and event management (SIEM) and network monitoring tools—OSSIM [1] and Zabbix [2]—to provide continuous monitoring and real-time analysis of a shipboard ICS environment. The project assignment tasks included:

– Acquire a full understanding of the tools being investigated. This includes installing the tools and running experiments to gain insight on each tool's capabilities, system architecture, software design, and dependencies. This task also includes a survey of comparable commercial products; and
– Propose how the tools can be used in a shipboard ICS. The proposed design must identify the modules (plugins) that must be developed or customized for the afloat environment.

This was a hands-on project in which students built two test environments and ran a series of functional tests on the target tools. One environment was an isolated "practice network" for learning about the tools, and the other was a mock ICS environment modeled after an actual ICS network provided by the SME. Packet captures provided by the SME were replayed to test the tools.

5.5   Smart Card Authentication

This project explored the use of PKI-based smart cards for user authentication in shipboard control systems. Both contact and contactless smart card technologies were investigated. The assigned tasks were:

– Survey existing smart card technologies and products, including their utilization in ICS domain.
– Develop a concept of operations for PKI-based user authentication in a land-based ICS architecture, informed by DoD regulations on using smart cards for Personal Identity Verification (PIV); and
– Recommend a smart card product for use in an ICS on ships.

The recommendation also addressed: operational scenarios including different classes of users, e.g., operators, maintainers, security administrators; system life cycle management from initial deployment through retirement or disposal; and system boundaries and interconnections.

5.6   Code Repository Security

This project investigated security threats and defenses related to revision control systems for ICS software. Two revision control systems were examined: Apache Subversion (SVN) and Git. The assigned tasks included:

- Survey known threats against revision control systems;
- Survey known attacks against SVN and Git services, including how such attacks could theoretically damage the life cycle maintenance of ICS software artifacts;
- Recommend how to secure an SVN server and a Git server for use in ICS development, including eliciting functional and security requirements from ICS software developers; and
- Recommend a revision control system and methods for configuring and hardening it for use in ICS.

It was important to the SME that the recommended system addressed challenges for using a revision control system in a disconnected development environment, i.e., the revision control system resides in a disconnected laboratory and the corporate LAN is where non-developers (systems engineers, managers, auditors, etc.) view related artifacts.

5.7   Backplane Intrusion Detection

Modern PLCs are modular, consisting of multiple modules that communicate via a backplane. Mulder et al. perform several analyses on PLC hardware, firmware, and backplane activities to look for low-level information about PLC design and software that attackers can exploit, e.g., hardware properties and backplane traffic [17]. These efforts lead to the development of the WeaselBoard, a PLC backplane analyzer that captures backplane traffic and forwards it to an external system for intrusion analysis. The objective of this project was to perform a tabletop vulnerability assessment of WeaselBoard using a hypothetical ICS. The project consisted of the following tasks:

- Perform a threat analysis to create a threat profile for the target ICS;
- Perform a vulnerability analysis of the target ICS; and
- Develop potential attack scenarios in which persistent payloads are utilized to disrupt the operation of the target ICS without triggering an alarm from the WeaselBoard.

This project was the most challenging project to-date. Project tasks included searching the National Vulnerability Database for known vulnerabilities related to the platforms and software used in the target ICS.

6   Discussion

In this section we share a number of lessons learned related to delivering our project-based course about ICS security. We follow a learner-centric paradigm, resembling in many ways a flipped classroom where background and preparatory work occurs outside classroom contact hours and student-instructor interactions are reserved for interactive problem solving and planning. Over three years of delivering this capstone course, several modifications in course format and assessment have led to its current incarnation.

The first offering of the course was treated as a graduate-level advanced topics class in which students read papers to gain background knowledge. Direct instruction and a field trip (phase 1) were introduced in the next class, which significantly improved student understanding and reception. The second critical and biggest course improvement came after the introduction of simple introductory SCADA lab exercises, affirming that small, hands-on activities play a critical role in the learning process.

We found students were most actively engaged in research projects when elements of the assignment were familiar or if they had prior experience with the problem domain. For example, one student whose thesis project incorporated ArcSight was able to apply this knowledge in the context of the network security project (see Sect. 5.2). Projects with explicit hands-on experiments also increased student engagement, based on feedback and activity logs.

Student experience was highly variable cohort-to-cohort; although a few students had worked with shipboard ICS or had an undergraduate degree in computer science, most students lacked the technical background required by the projects designed by the SME. This mismatch made the formulation of appropriate project assignments difficult, as the collective experience of a cohort is not fully understood in phase 0, when project assignment areas are proposed.

The course structure had both advantages and disadvantages. It allowed frequent feedback from the SME which, in turn, provided opportunities for students to rise to new challenges. Many students, however, were only familiar with short class projects in which a strategy or approach, once selected, could be worked until completion. Our iterative research and design process was challenging to these students. Many viewed unanticipated problems as impediments, rather than opportunities for amelioration. When strategies required adjustment, students viewed this as time wasted on a poor strategy, rather than a process by which identification of a better strategy was itself a beneficial outcome.

We found that students had trouble in applying prior knowledge in new and unfamiliar contexts. For example, students learned about IPsec in other classes and had the skill to construct IPsec-based virtual channel networks, but they had difficulties researching ways to extend the core IPsec functionality to solve a more complex problem, e.g., using IPsec with multicast addressing. This trouble in horizontal transfer of knowledge led to improved scaffolding during phase 1, leveraging direct instruction in the project assignment domain. Students were similarly challenged in working with entirely new concepts, i.e., those not explicitly covered in prior courses, such as software development and revision control. This was worsened by the perception that these were not relevant to their course of study. More than direct instruction, interaction with SMEs who attested to the relevancy of these topics from a practical, cyber operations perspective was essential in overcoming this perception.

In summary, we found that students were able to demonstrate understanding of ICS security issues successfully, through interpreting and analyzing topics covered in class and in their research assignments. In particular, SME feedback indicated that final COA recommendations were sensible and, in some cases,

targeted for adoption. This was, to us, one of the most essential indicator of student learning, demonstrated through individual presentations, group reports and SME reviews.

Recommendations. We found students felt overloaded when attempting knowledge transfer to the unfamiliar ICS domain; this worsened when the project assignment was itself foreign and in an unfamiliar context. The positive response to the addition of hands-on ICS laboratory exercises in phase 1 largely echoes prior successes reported by others with using hands-on modules for ICS education. A focused class on ICS security leveraging hands-on exercises would be an invaluable prerequisite to any course employing real-world ICS systems as case studies.

One of the most labor-intensive aspects of capstone development was travel logistics (associated with SME visits and field trips) and other phase 0 planning activities. As each course offering focused on disjoint topics, this topic refinement became time-consuming to both instructor and SME. Additionally, each project's learning curve was quite steep for the student teams. As a notable exception, when two capstone classes used identical case studies in different contexts (i.e., the list and ventilation control systems), the second class was able to leverage the prior class' reports very successfully. We believe following this pattern—where team assignments intentionally share context across cohorts—may be a highly successful strategy. In particular, it allows project outcomes from one cohort to inform the next, following an agile research process [3]; classes could review and re-evaluate past projects as case studies; and tech transfer is a recurring process in which team deliverables are transferred to stakeholders, to other teams and across cohorts. Projects may build on past deliverables, improving or reconsidering previous findings.

7   Conclusion

This paper presents an approach to teach ICS security as a capstone course using collaborative research projects designed by ICS experts. We described our course motivation, development efforts and instructor observations. We summarized lessons learned from running the course, based on three years of feedback.

Our observations largely echo those of other educators in reinforcing the importance of hands-on exercises for ICS education. We caution other educators attempting to use real-world ICS case studies, in absence of a prerequisite course on SCADA and ICS. Furthermore, having SME support, system owner's participation and field trips was imperative for reinforcing course content, especially for a practical domain like ICS, where nearly all practical experience is with large, complex legacy systems. We believe other ICS curriculum proposals have completely omitted these instructional aids. For an effective ICS capstone course, we recommend field trips to local industrial facilities, e.g., a waste water treatment plant or electrical substation, if possible.

References

1. AlienVault OSSIM: The open source SIEM (2015). https://www.alienvault.com/products/ossim
2. Zabbix: the enterprise-class monitoring solution for everyone (2015). http://www.zabbix.com/
3. Dark, M., Bishop, M., Linger, R.C., Goldrich, L.: Realism in teaching cybersecurity research: The agile research process. In: Bishop, M., Miloslavskaya, N., Theocharidou, M. (eds.) WISE 9. IFIP AICT, vol. 453, pp. 3–14. Springer, Heidelberg (2015)
4. Department of Homeland Security (U.S.). Critical infrastructure and control systems security curriculum, March 2008
5. Digital Bond, Inc.: Quickdraw SCADA IDS (2014). http://www.digitalbond.com/tools/quickdraw/
6. Executive Order no. 13636. Improving Critical Infrastructure Cybersecurity, February 2013. http://www.gpo.gov/fdsys/pkg/FR-2013-02-19/pdf/2013-03915.pdf
7. Foo, E., Branagan, M., Morris, T.: A proposed australian industrial control system security curriculum. In: 2013 46th Hawaii International Conference on System Sciences (HICSS), pp. 1754–1762. IEEE (2013)
8. Foreman, J.C., Graham, J.H., Hieb, J.L., Ragade, R.K.: A curriculum model for industrial control systems cyber-security with sample modules. Technical Report 2012–14, Center for Education and Research, Purdue University (2012)
9. Francia III, G.A.: Critical infrastructure security curriculum modules. In: Proceedings of the 2011 Information Security Curriculum Development Conference (InfoSecCD 2011), pp. 54–58, Sept 2011
10. Francia III, G.A., Beckhouche, N.: Portable SCADA security toolkits. Int. J. Inf. Netw. Secur. (IJINS) **1**(4), 265–274 (2012)
11. Francia III, G.A., Snellen, J.: Embedded and control systems security projects. Inf. Secur. Educ. J. **1**(2), 77–84 (2014)
12. Irvine, C.: A cyberoperations program. IEEE Secur. Priv. Mag. **11**(5), 66–69 (2013)
13. Luallen, M.E., Labruyere, J.-P.: Developing a critical infrastructure and control systems cybersecurity curriculum. In: 46th Hawaii International Conference on System Sciences (HICSS), pp. 1782–1791. IEEE, January 2013
14. McGrew, R.W., Vaughn, R.B.: Discovering vulnerabilities in control system human-machine interface software. J. Syst. Softw. **82**(4), 583–589 (2009)
15. Mishra, S., Romanowski, C.J., Raj, R.K., Howles, T., Schneider, J.: A curricular framework for critical infrastructure protection education for engineering, technology and computing majors. In: 2013 IEEE Frontiers in Education Conference (FIE), pp. 1779–1781. IEEE, October 2013
16. Morris, T., Srivastava, A., Reaves, B., Gao, W., Pavurapu, K., Reddi, R.: A control system testbed to validate critical infrastructure protection concepts. Int. J. Crit. Infrastruct. Prot. **4**(2), 88–103 (2011)
17. Mulder, J., Schwartz, M., Berg, M., Van Houten, J.R., Mario, J.: WeaselBoard: zero-day exploit detection for programmable logic controllers. Technical report SAND2013-8274, October 2013

18. National Institute of Standards and Technology (U.S.): Framework for improving critical infrastructure cybersecurity, February 2014
19. National Security Agency (U.S.): Academic Requirements for Designation as a Center of Academic Excellence in Cyber Operations (2014). https://www.nsa.gov/academia/nat_cae_cyber_ops/nat_cae_co_requirements.shtml
20. The Snort Project. SNORT users manual (2014). http://manual.snort.org/snort_manual.htm
21. Tofino Security Inc.: Tofino SCADA security simulator (TSSS) user's guide, January 2013
22. Vaughn, R.B., Morris, T., Sitnikova, E.: Development & expansion of an industrial control system security laboratory, an international research collaboration. In: CSIIRW 2013: Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop. ACM, January 2013
23. Weis, B., Gross, G., Ignjatic, D.: Multicast extensions to the security architecture for the internet protocol. RFC 5374, November 2008
24. Wightman, R.: S4x12: Project basecamp (2012). http://vimeopro.com/s42012/s4-2012/video/35783988