Gamifying ICS Security Training and Research: Design, Implementation, and Results of S3

Daniele Antonioli
Singapore University of Design and Technology (SUTD)
Singapore, Singapore
daniele_antonioli@sutd.edu.sg

Hamid Reza Ghaeini
Singapore University of Design and Technology (SUTD)
Singapore, Singapore
ghaeini@acm.org

Sridhar Adepu
Singapore University of Design and Technology (SUTD)
Singapore, Singapore
sridhar_adepu@sutd.edu.sg

Martin Ochoa
Singapore University of Design and Technology (SUTD)
Singapore, Singapore
martin_ochoa@sutd.edu.sg

Nils Ole Tippenhauer
Singapore University of Design and Technology (SUTD)
Singapore, Singapore
nils_tippenhauer@sutd.edu.sg

ABSTRACT

Our work considers the challenges related to education and research about the security of industrial control systems (ICS). We propose to address those challenges through gamified security competitions. Those competitions should target a broad range of security professionals (e. g., from academia and industry). Furthermore, they should involve both attack and defense components. This could include the development of new attack techniques and evaluation of novel countermeasures. Our gamification idea resulted in the design and implementation of the *SWaT Security Showdown (S3)*. S3 is a Capture-The-Flag event specifically targeted at Industrial Control Systems security. We developed ICS-specific challenges involving both theoretical and applied ICS security concepts. The participants had access to a real water treatment facility and they interacted with simulated components and ICS honeypots.

S3 includes international teams of attackers and defenders both from academia and industry. It was conducted in two phases. The online phase (a jeopardy-style capture the flag event) served as a training session and presented novel categories not found in traditional information security CTFs. The live phase (an attack-defense CTF) involved teams testing new attack and defense techniques on SWaT: our water treatment testbed. During the competition we acted as judges, and we assigned points to the attacker teams according to a scoring system that we developed internally. Our scoring system is based on multiple factors, including realistic ICS attacker models and effectiveness of the detection mechanisms of the defenders. For each phase of the S3 we present the results and relevant statistics derived from the data that we collected during the event.

CPS-SPC'17, November 3, 2017, Dallas, TX, USA.

ACM ISBN 978-1-4503-5394-6/17/11…$15.00
https://doi.org/10.1145/3140241.3140253

CCS CONCEPTS

• **Security and privacy → Human and societal aspects of security and privacy**; *Systems security*;

KEYWORDS

Gamification; Capture-The-Flag; ICS Security; Education; Training

1 INTRODUCTION

Recently, it has been widely argued that one of the fundamental issues in securing industrial control systems (ICS) lies in the cultural differences between traditional IT security and ICS engineering [19, 28]. Therefore, education has been advocated as a means of bridging the gap between these cultures [18, 19]. However, recent surveys indicate that although general IT security education efforts have risen in ICS, there is still need for more targeted education combining both security and ICS specific knowledge [13].

Typically, those willing to do research on ICS security are facing severe problems, such as lack of understanding of a real ICS, and the inability to test (new) attacks and countermeasures in a realistic setup. ICS testbeds constitute a convenient environment to study ICS security, however their deployment is rare because of many (reasonable) costs, such as infrastructure and manpower costs [7, 31]. Another common issue in ICS security is resulting from the intrinsic inter-disciplinary nature of the subject. It is difficult to bring together people from different expertise domains, such as control theory, information security, and engineering.

In this work we propose a solution to the ICS security education problem, based on gamified security competitions. By education we mean both training of new ICS security professionals, and helping researchers to advance the state-of-the-art of ICS security. Our gamification idea evolves around four key points. Firstly, the competition has to be domain-specific (targeted education). Secondly it has to involve people from academia and industry possibly with different expertises (addresses the cultural differences). Thirdly, the contest has to be fun to play to motivate the participants (gamified). Finally it has to present interaction with real ICS components using real ICS tools (realistic attacks and countermeasures).

The result of our efforts is the *SWaT Security Showdown (S3)*, a Capture-The-Flag (CTF) targeted to industrial control systems

security. This paper focuses on the design, implementation and results from the first S3 edition of 2016. S3 was hosted by our institution the Singapore University of Technology and Design. S3 is divided into two phases: an online training CTF, and a live attack-defense CTF. During the online phase the attackers participated in a Jeopardy-style CTF. The online CTF challenges included novel ICS-specific categories, involving for example real-time interactions with ICS simulations, and remote access and programming of real ICS devices. During the live CTF both the attacking and defending teams had access to our water distribution testbed (SWaT). They deployed a wide range of attacks, while two academic attack detection systems were in place.

We summarize our contributions as follows:

- We identify several issues that currently hinder industrial control systems security education and research.
- We propose a solution to address those issues, focusing on a gamified Capture-The-Flag (CTF) competition, using simulated and real ICS infrastructures.
- We present the design and implementation of the *SWaT Security Showdown (S3)* competition. S3 uses a combination of Jeopardy-style CTF and attack-defense CTF to provide a novel and hands-on learning experience for ICS security professionals.

This work organized as follows: in Section 2, we provide brief background on industrial control systems (ICS), the Secure Water Treatment testbed, and Capture-The-Flag events. In Section 3, we present the current challenges for ICS security education and research, our problem statement, and the design of S3. The details about S3 online and live phases are presented in Section 4 and Section 5. Related work is summarized in Section 6, and we conclude the paper in Section 7.

2 BACKGROUND

2.1 Industrial Control Systems Security

Industrial control systems (ICS) are autonomous systems composed of heterogeneous and interconnected devices. ICS are deployed to monitor and control different types of industrial processes, such as critical infrastructures (water distribution and treatment), and transportation systems (planes and railways).

ICS security is a major challenge for many reasons. Firstly, the complexity and diversity of devices involved in an ICS increases the attacker surface. For example, an attacker might attack the cyber-part, the physical-part or both parts of the ICS. Additionally, modern ICS are embracing standard Internet communication technologies, such as TCP/IP based industrial protocol, resulting in ICS that can be controlled (and attacked) from the Internet. Arguably, threats to ICS focus on impacting the physical world, instead of attacks on the confidentiality and the integrity of the information. As such, the damage by those attacks is expected to cause high financial and human costs due to destroyed property and decreased operational availability of commercial systems. Famous examples of high-impact attacks on ICS are the recent attack on the Ukraine power grid [11], the Stuxnet worm [12], and the attack on a wastewater treatment facility in Maroochy [29].

2.2 Secure Water Treatment (SWaT) Testbed

For the experimental part of this work we target the *Secure Water Treatment (SWaT)*. SWaT is a state-of-the-art water treatment testbed available at our institution since 2015 [20]. SWaT is composed of six stages and includes advanced filtering equipment such as: ultrafiltration and reverse osmosis sub-systems. We now briefly describe the six stages of SWaT:

(1) *Supply and Storage* pumps raw water from the source to the Raw water tank.
(2) *Pre-treatment* chemically treats raw water controlling electrical conductivity and pH.
(3) *Ultrafiltration (UF) and backwash* purifies water using ultrafiltration membranes, collects ultra-filtrated water in the Ultra-filtration tank, and periodically cleans the UF membranes.
(4) *De-Chlorination* chemically and/or physically (UV light) removes chlorine from ultra-filtrated water.
(5) *Reverse Osmosis (RO)* purifies water using RO process, separates the result into permeate (purified) and concentrate (dirty) water.
(6) *Permeate transfer and storage* store permeate water into the RO permeate tank.

Figure 1 shows a schematic view of SWaT architecture. Starting from the bottom we can see six gray boxes representing the six water treatment stages. Each stage involves two Programmable Logic Controllers (PLCs) configured in redundant mode, and a Remote Input-Output (RIO) device that interfaces the PLC with the sensors and actuators. The *field* networks (Layer 0) use an Ethernet ring topology. The rings are established and maintained using the device level ring (DLR) protocol. The data is exchanged using EtherNet/IP over UDP. Every PLC is connected to the *control* network (Layer 1). The control network has a star topology, and it includes the PLCs, a SCADA server, an HMI, and a historian server. Other network devices (e. g., in the DMZ network) access the SWaT control network through an industrial firewall. EtherNet/IP over TCP is used in the control network to carry data about commands, sensors, and actuators. EtherNet/IP is an object oriented industrial protocol. In particular, it is an implementation of the common industrial protocol (CIP) [22] on top of the TCP/IP protocol stack.

2.3 Capture-The-Flag (CTF) Events

Capture-The-Flag (CTF) events are cyber-security contests organized by universities, private companies and non-profit organizations. CTF competitions can be classified in two categories: *Jeopardy-style* and *attack-defense*. A Jeopardy-style CTF usually is hosted on the Web, and includes a set of tasks to be solved divided by categories (e. g., cryptography, exploitation and reverse engineering). Each task is presented with a description, a number of hints and an amount of reward points. The solution of a challenge comprises finding (or computing) a message (the flag) with a prescribed format, such as `ctf{foo-bar}`, and submitting it to the CTF scoring system. An attack-defense CTF, also called red team (the attackers) blue team (the defenders), is organized both offline and online. Each team is given an identical virtual machine containing some vulnerable services. The teams are connected on the same LAN,

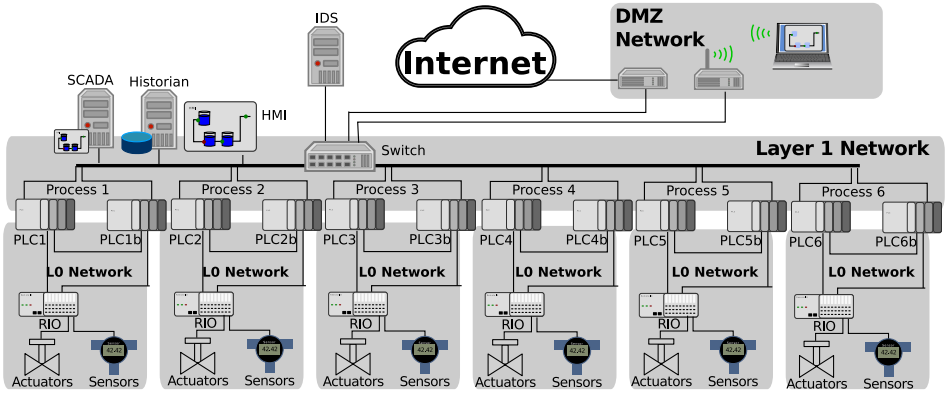

Figure 1: The Secure Water Treatment (SWaT) testbed architecture.

and their goal is both to have an high service runtime and to tamper with the services of the other teams. For example, finding and exploiting a vulnerable service has two benefits: it allows a team to patch its service to be more resilient to the attacks from the other teams, and to attack other teams vulnerable service. Both Jeopardy-style and attack-defense, CTF have time constraints (e. g., increase level of realism), and the team who scored most points wins the competition.

3 GAMIFYING ICS SECURITY

We start this section by summarizing the current main challenges related to ICS security education. Then, we set the problem statement according to those challenges. Finally, we propose our solution based on gamified CTF competitions and we discuss its design points.

3.1 ICS Security Education Challenges

In recent years, experts have argued extensively about the criticality of securing industrial control systems (ICS). Many have pointed out that one fundamental challenge in achieving this task lies in cultural and educational differences between the fields of (traditional) information security and ICS security. According to Schoenmakers [28]: *"Differences in perspectives between IT and OT specialists can cause security issues for control systems. It is important for organizations to keep in mind that different values between groups can influence the perception of issues and solutions."*, which emphasizes the cultural clashes still existing between traditional IT security and ICS specialists.

Education and training have been advocated to bridge this gap, but there still work to do in this domain. Luijif [18] describes the security of ICS as a societal challenge, and recommends: *"Many of these challenges have to be overcome by both end-users, system integrators and ICS manufacturers at the long run: (…) proper education and workforce development"*. Despite the problem of education being widely acknowledged, according to a recent report published by SANS Institute [13]: *"It is clear from our results that most of our respondents hold security certifications, but the largest number of these (52%) is not specific to control systems (…) IT security education is valuable, particularly with the converging technology trends, but it does not translate directly to ICS environments."*

In order to effectively improve the security of ICS it is thus crucial to educate researchers and practitioners such that they are able to understand the domain-specific requirements and constraints of ICS security. As recently pointed out by Luijif in [19]: *"(…) ICS and (office) IT have historically been managed by separate organizational units. ICS people do not consider their ICS to be IT. ICS are just monitoring and control functions integrated into the process being operated. ICS people lack cyber security education. The IT department, on the other hand, is unfamiliar with the peculiarities and limitations of ICS technology. They do not regard the control of processes to have any relationship with IT. Only a few people have the knowledge and experience to bridge both domains and define an integrated security approach. Organizations that have brought the personnel from these two diverse domains together have successfully bridged the gap and improved the mutual understanding of both their IT and ICS domains. Their security posture has risen considerably."*

3.2 Problem Statement

Based on the challenges from Section 3.1 we isolated four key points summarizing the main issues of ICS security education and research:

(1) Lack of ICS-specific security education
- Control engineers not trained for cyber-security
- Cyber-security experts not trained for industrial control systems

(2) Cultural differences among involved professionals
- Academia vs. Industry
- IT vs. OT security

(3) Lack of motivations and incentives
- Different communities speak different technical languages
- Frictions from interdisciplinary collaborations

(4) ICS infrastructures are difficult to access
- Production ICS cannot be touched for education and research
- Ad-hoc (academic) testbeds are rare and costly

Based on our experience we can translate those problems to requirements for IT and OT security professionals. For example, IT professionals need more information about common device classes

(e. g., PLC, HMI, SCADA), network topologies (e. g., DLR), and industrial protocols (e. g., EtherNet/IP, Modbus). Furthermore, they need more knowledge about physical process specifications (e. g., set points and interlocks), control theory models (e. g., state observer) and ICS-specific design methodologies (e. g., Purdue model). On the other hand OT professionals have another set of necessities. For example they have to be familiar with modern penetration testing and reconnaissance tools (e. g., metasploit, nmap). They have to be acquainted with basic cyber-security hygiene concepts (e. g., confidentiality, integrity, dependability and availability). They have to know modern Internet communication technologies (e. g., Ethernet, TCP/IP, NAT, Web), and common security challenges and standard solutions (e. g., MitM attacks, TLS, firmware and software update schedules).

3.3 Our Idea: the S3 Competition

Gamification in education has already been advocated as a means to enrich learning experiences [15]. In particular, within IT security, the development of Capture-The-Flag like competitions have been argued to be advantageous for education and training [30]. Inspired by the gamified nature of CTF, we propose to address the issues of ICS security education and research with the *SWaT Security Showdown (S3)* competition. One of our main goal is to create an ICS security competition where participants are encouraged to think and act like real industrial control systems attackers and defenders. For example, an attacker would have to bypass advanced intrusion detection systems deployed by a defender in a simulated or real ICS environment. Such a setup would stimulate the participants to use creative attacks and defense strategies, and would potentially unveil the limitations and the advantages of those strategies. Another key point is to give the participants access to a real ICS infrastructure. For the first edition of S3 we decided to focus on the water treatment industrial process and we centered the competition around SWaT, our water treatment testbed (see Section 2.2).

To get the most out of interactions with a real ICS testbed, it is important to learn fundamental concepts of ICS security. However, this learning phase should be as hands-on and gamified as possible to motivate the participants. To this extent, we propose an *online* training phase, where attackers could get familiar with ICS security notions by means of a Jeopardy-style CTF. The S3's online phase is different from traditional IT CTF events because the challenges are tailored to highlight ICS security concepts. For example some challenges involve remote interactions with both simulated and real ICS sub-systems. In this phase, attackers are evaluated by looking at the number of solved challenges and the defenders are not participating. More details about the online phase are presented in Section 4.

After the online preparation phase, the attackers are invited to attack a real water distribution testbed that is being monitored by the defenders in a *live* attack-defense CTF phase. In this scenario, the attackers should have concrete goals to achieve, and their scoring should be influenced by realistic factors, such as the number of defenses triggered during an attack. In order to properly score an attack we provide to the attacker teams different attacker-model choices that set the attacker's capabilities. For example an insider would have administration capabilities, while an outsider would only have network access. More details about the live phase are presented in Section 5.

Due to organizational constraints, and in order to maximize the learning experience, we decided to limit the participants to SWaT Security Showdown to selected invited teams from academia and industry. We invited twelve (12) teams (6 attackers, 6 defenders, of which 3 academic and 3 industrial teams respectively). Teams were not limited in size, but only a maximum of 4 members could participate physically in the live event whereas remaining team members could join remotely.

4 ONLINE PHASE OF S3

In this section we introduce the setup of the SWaT Security Showdown online event and we list the presented challenges and their categories. We describe more in detail several challenges from the MiniCPS, Trivia, and Forensics categories. We conclude the section with a summary of the collected results.

4.1 Setup and Challenges

The aim of the online event was to provide an adequate training to the attacker teams in preparation for the live phase. Before this event, we gave to the attackers the relevant documentation to get familiar with the Secure Water Treatment testbed. The online event was structured as a *Jeopardy-style* CTF, and did not require physical access to SWaT. The contest was divided into two 48-hours CTF sessions where each session involved three teams. The two sessions presented the same challenges. Table 1 summarizes the information about the twenty (20) challenges that we design. They were divided into five categories: MiniCPS, Trivia, Forensics, PLC, and Misc, for a total of 510 points. We balanced the number of challenges and the amount of points to accomodate different types of attacker with different levels of expertise. It is worth stressing that MiniCPS, Trivia and PLC categories are novel in the domain of Jeopardy-style CTFs, and their design is part of the contributions of our paper.

Our Jeopardy-style CTF was designed following the best-practices from state-of-the-art information security CTFs. The flag format was set to `s3flag{foo-bar}`. Each group of challenges was presented in increasing order of difficulty. Within each category (where possible) a challenge was a prerequisite for the next one (e. g., solving challenge number x helped to solve challenge number $x + 1$). For each challenge we included some hints in its the description (e. g., you might use `tool_x` to solve problem x).

The CTFs were hosted using an internally developed web application based on flask [26]. Our website contained a basic web page listing the challenges divided by categories (see Figure 2), and a web page showing live chart and notification messages. Each member of a team logs in to S3's Webapp (using the provided credentials), then navigates to challenge X's Web page, then enters the flag on an HTML form. If the flag is correct, she receives N reward points, otherwise a submission error appears on screen. The web pages were served over HTTPS using Let's Encrypt [14] certificates. A basic brute-force detection mechanism based on user input logging was put in place on the backend side. During the two CTF sessions we offered live help through a dedicated IRC channel, and via email (e. g., each challenge web page pointed to the email address of its author(s)).

Table 1: SWaT Security Showdown (S3) online challenges ordered by novelty in the context of ICS specific Jeopardy-style CTFs. 20 challenges, worth 510 points, exercising different domains of ICS security.

Category	Challenges	Points	Exercised ICS Security Domains	Novelty
MiniCPS	5	210	Simulated tank overflows, industrial network mapping, MitM attacks	High
Trivia	6	45	SWaT's physical process, devices and attacks	High
PLC	3	60	Remote access to real PLCs, Ladder logic programming	High
Forensics	4	105	Packet manipulation and cryptography	Medium
Misc	2	90	Web authentication, steganography	Low
Total	**20**	**510**		

4.2 MiniCPS Category

The online phase presented five challenges in the MiniCPS category. MiniCPS [5] is a framework for Cyber-Physical System security research. It uses real-time simulation of physical processes and control devices, it is open-source software [3] and it builds on top of a network emulator called mininet [17]. The aim of the MiniCPS challenges was to present a realistic and interactive simulation environment where the attacker could discover and attack a virtual water treatment ICS without harming our real testbed.

Figure 3 shows the setup of each MiniCPS simulation instance (e. g., one for each attacking team). Each instance can be thought as a virtual high-interaction ICS honeypot [4]. The attacker is provided with administrative credentials of a virtual gateway SSH server. Once connected the attacker could interact with other simulated SWaT's devices in the simulated control network (e. g., four PLCs and an HMI connected in a star topology). As an example, an attacker might alter the state of the simulated water treatment process affecting the two simulated water tanks (the Raw water tank and the Ultra-filtration tank). We now present more details about the five MiniCPS challenges:

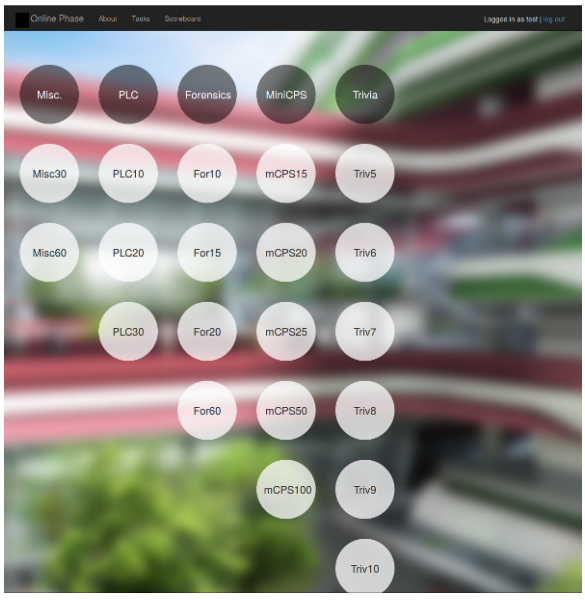

Figure 2: S3 online challenges web page.

1 - Network warm up. The goal of the challenge is to perform a passive ARP-poisoning MitM attack between PLC2 and PLC3. The attacker has to perform a network scanning to discover the hosts addresses and then use `ettercap` to read the flag on the wire.

2 - EtherNet/IP warm up. The goal of the challenge is to read the flag stored in PLC2's EtherNet/IP server, and addressable with the name `README:2`. Remember that EtherNet/IP is the industrial protocol used by SWaT. The attacker has to understand which PLC owns the `README:2` tag, and how to use `cpppo`, the suggested EtherNet/IP's Python library [16].

3 - Overflow the Raw water tank. The goal of the challenge is to overflow the simulated Raw water tank. The attacker has to understand the simulated dynamic of a water tank (e. g., who drives the water tank pump), and send malicious actuation commands over the network to increase the water level above a fixed threshold.

4 - Denial of Service HMI. The goal of the challenge is to disrupt the communication between HMI and PLC3, and then change a keep-alive value to 3 in the EtherNet/IP server of PLC3. In normal working condition the keep-alive value is periodically set by the HMI to 2. The attacker has to perform an active MitM attack that drops all the packets between the HMI and PLC3 and then set the keep-alive value to 3.

5 - Overflow the Ultra-filtration tank. The goal of the challenge is to overflow the Ultrafiltration water tank. The attacker has to use advanced packet manipulation techniques in an active MitM attack (e. g., using `ettercap` and `etterfilter` to change the packet payload on the fly).

4.3 Trivia Category

The online phase presented six challenges for the Trivia category. These challenges are divided into two sub-categories: SWaT-related knowledge (SK) and ICS security research papers (RP). We now present the trivia questions divided by sub-category:

SK 1. The goal of the challenge is to identify the chemical analyzer that is used by the PLC to control a specific dosing pump. In order to identify the device, the participant has to understand the control strategy applied to that particular dosing pump. As the PLC uses a number of different inputs to control the dosing pump,

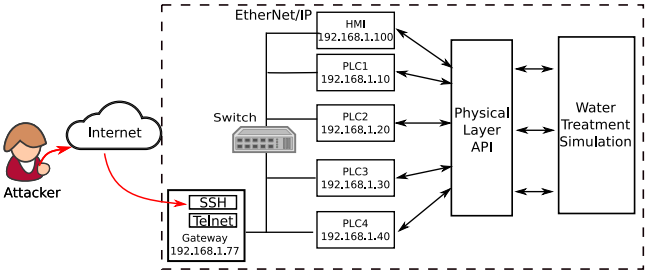

Figure 3: MiniCPS simulation environment for the S3 online event (e. g., virtual high-interaction ICS honeypot).

the participant has to trace the signals and identify the particular analyzer.

SK 2. The goal of the challenge is to find out the set point (e. g., threshold) that triggers the start of SWaT backwash process. During the filtration process, small particles clot the Ultrafiltration membrane. To remove them and clean the Ultrafiltration membrane, a backwash process is started after reaching a specific value. In order to answer this question, the participant needs to revise and understand the backwash process.

SK 3. The goal of the challenge is to identify the set point of the hardness analyzer used by a PLC to shut down the reverse osmosis (RO) filtration process. The set point is a desired value of a particular sensor which is periodically queried by the PLC. In order to solve this challenge, attacker should understand the set points and control strategy of the RO filtration process.

RP 4. The goal of this challenge is to familiarise the attacker with possible attacks on SWaT and their potential impacts on the testbed. We provided a research paper that presented an experimental investigation of cyber attacks on water treatment ICS. In order to solve the challenge, the participant needed to read the paper, understand it and answer to a specific question.

RP 5. The goal of this challenge is to familiarise the participants with a security analysis of a water treatment ICS. We provided another research paper that presented a security analysis of ICS using a formal model. In order to solve the challenge an attacker should read the paper, understand it and answer to a specific question.

RP 6. The goal of this challenge is to familiarise the attacker with multi-point attacks on ICS. We provided a third research paper that discussed multi-point attacks. A multi-point attack leverages more than one entry point, (e. g., two or more communications links), to disturb the state of an ICS. In order to answer the challenge, the participant needed to read the paper, understand it and answer to a specific question.

4.4 Forensics Category

The forensics challenges focused on the analysis and process of capture files from industrial control systems networks. We distributed *pcap* files because they are easy to read and write using programs such as `wireshark` and `tcpdump`. The target industrial protocol was EtherNet/IP [22], the traffic was pre-recorded and sometimes post-processed (e. g., add packets to confuse the attackers) We now provide some details about three (out of four) Forensics challenges:

1 - Identify the ICS hosts. The goal of the challenge is to perform an analysis of the ICS hosts from a pcap file. To solve the challenge, the attacker should search for the hosts inside the captured traffic, classify them based on their IP addresses, identify whether a host is inside the ICS network or not and enumerate them.

2 - Finding the poisoning host. The goal of the challenge is to search for a host that had performed ARP poisoning Man-in-the-Middle attack, from a pcap file. Then, the attacker has to identify the start and end points (e. g., packets) of the ARP poisoning attack. As an example, the flag for this challenge could be `s3flag{A-B}`, where the A is the start TCP sequence number and B is the end TCP sequence number.

3 - Understanding the CIP protocol structure. The goal of the challenge is to find a particular pattern inside the payload of CIP messages. CIP is the Common Industrial Protocol and EtherNet/IP is an implementation of the CIP application layer specifications over TCP/IP. In this case, the attacker has to recognize that a CIP payload contains encrypted data and then he has to decrypt it. The attacker obtains the plaintext by XORing the ciphertext with the key provided in the packet payload or by a brute-force attack.

4.5 Online Phase Results

Table 2 presents the final results form the S3 online event (a Jeopardy-style CTF). We decided to anonymize the team names for privacy concerns. The table has one row for each attacking team (six in total) and shows the number of flags captured per category, total number of captured flags and final scores. The last column contains an estimation of the time spent by each team on the tasks. Each value is computed as the difference between the timestamps from the last and the first flag submitted by a team. As we can observe from the table, two teams were able to fully complete all tasks, with Team 6 being by far the most efficient. On average teams spent 25.67 hours to solve the challenges (53% of the maximum of 48 straight hours), with a standard deviation of 13.06 hours. The teams scored an average of 268.83 points (52.7% of the maximum of 510). We believe that both the time invested and the percentage of challenges solved shows a notable investment in the game, and provides evidence on the engagement generated by the gamification strategy.

Table 2: SWaT Security Showdown (S3) online event (Jeopardy-style CTF) results summary. Category names: C=MiniCPS, T=Trivia, F=Forensics, P=PLC, M=Misc.

	Flags per category							
Team	C	T	F	P	M	Flags	Score	Time
T1	2	6	4	0	1	13	250	30h
T2	5	6	4	3	2	20	510	44h
T3	0	4	2	0	1	7	86	27h
T4	4	4	2	0	0	10	161	28h
T5	0	4	2	0	1	7	66	21h
T6	5	6	4	3	2	20	510	4h

5 LIVE PHASE OF S3

In this section we focus on the SWaT Security Showdown live event, and the details about its setup, goals, and scoring system. We describe ARGUS and HAMIDS, two of the academic detection mechanisms that were used during the S3 live event. We conclude the section providing a summary of the collected results and describing four selected attacks from the live competition.

5.1 Setup and Goals

The S3 live phase is structured as an *attack-defense* CTF. It was held at our institution (SUTD) over the course of 2 days in July 2016. We invited the six attacker teams who participated in the online phase, and six defender teams (four from industry and two from academia. Each attacking team visited SWaT for one working day before the live competition. During the competition each attacking team had three hours to test and deploy a range of attacks. The live phase had two main goals. Firstly, allow the team to learn more about ICS (security) by letting them access a real plant. Secondly, test a number of academic and commercial detection mechanisms that were deployed in SWaT. We note that in this paper we describe and evaluate only our internally developed SWaT detection mechanism.

5.2 Scoring and Attacker Profiles

We designed the scoring system for the S3 live phase with the following goals:

- Incentivise sophisticated attacks to better evaluate the countermeasures.
- De-incentivise re-use of same attack techniques.
- Accomodate attackers with different expertises.
- Correlate the attack score to an adequate attacker model.
- Minimize damages to the participants and the system.

We now briefly summarize the scoring system we devised. Points were only be awarded if the attack result could be undone by the attacker (to minimize the risks of permanent damages). Equation 1 defines how to score an attack attempt:

$$s = g \cdot c \cdot d \cdot p \tag{1}$$

The final score s is the product of four factors: g represents a base value that depends on the attacker goal. c is a control modifier used to measure the level of control the attacker has over her goal. d is a detection modifier and it is used to proportionally lower the score of an attack who triggered one or multiple detection mechanisms. p is the attacker profile modifier. Most modifiers were in the range $[1, 2]$, while the base value was in the range $[100, 200]$. We now present more in detail the four factors from Equation 1:

Attack base value (g). The attack base value depends on the attacker goal. The goal could be chosen from two sets: *physical process* goals or *data readings* goals. For the physical process goals the attacker has to demonstrate control over sensors, actuators, and the physical process (water treatment in this case). We weighted the score according to the target device affecting the physical process:

- 100 points: Motorised valves (open/close).
- 130 points: Water pumps (on/off).
- 145 points: Pressure sensors.
- 160 points: Tank fill levels (false water level).
- 180 points: Chemical dosing.

On the other hand, for the data readings goals the attacker should demonstrate control over sensor readings at different components. As in the previous case we weighted the score according to the target device (e. g., the closer the device is to the ICS field network the higher is the score):

- 100 points: Historian values.
- 130 points: HMI/SCADA values.
- 160 points: PLC values.
- 200 points: Remote I/O values.

Control modifier (c). The control modifier determines how much control the attacker has over her attack outcome (e. g., over the values that she is able to modify). As a guideline the modifier was 0.2 if the attacker could randomly influence a process value over time, up to 1.0 if the attacker could precisely influence the process value to a target one chosen by the judges.

Detection modifier (d). The detection modifier decreases the score of an attack proportionally to the number of detection mechanism that are able to identify the attack in real-time. Not triggering a detection mechanism while the attack is executed would increase the detection modifier using the following formula: $2 - x/6$, where x is the number of triggered detection mechanisms.

Attacker profile modifier (p). For each attack attempt, the attacking team had to inform the judges about the chosen attacker model before the attack is started. The attacker profiles are based on our research paper [25]. During the S3 live event we used three attacker profiles: the cybercriminal, the insider, and the strong attacker. In general, a weak attacker profile yields an high multiplier for the final score. For example, a successful attack performed as insider results in an higher score than the same attack performed as the strong attacker. The cybercriminal attacker had $p = 2$. He was assumed to have remote control over a machine in the ICS network. He was able to use standard tools such as `nmap`, and `ettercap` and to develop his own tools. The cybercriminal did not have access to ICS specific tools, such as Studio 5000 (IDE to configure SWaT's PLCs), or access to administrator accounts. The insider attacker had $p = 1.5$. He represented a disgruntled employee with physical access to the plant. The insider had a good knowledge of the plant, no prior attack experience, and only limited computer science skills. In particular, the insider was not allowed to use pentesting tools such as `nmap` or `ettercap`, but he had access to engineering tools (such as Studio 5000), and to an administrator account. The strong attacker was a combination of the cybercriminal and the insider attacker profiles. It was the strongest attacker model, hence it had the lowest modifier factor of $p = 1$. Attackers could earn points for one or more attacks. If more than one attack was successfully performed, the highest final scores from each attack were summed together. However if the same attack (e. g., same goal) was performed with two different attacker model then only the highest of the two score was counted.

5.3 Our SWaT Detection Mechanisms

As discussed in Section3, we decided to include both attack and defense components in the S3 competitions. We now briefly describe ARGUS and HAMIDS, two SWaT detection mechanisms developed at SUTD:

Table 3: SWaT Security Showdown (S3) live event (attack-defense CTF) results summary. d_{rate} computed using ARGUS and HAMIDS.

Team	Successful Attacks	d_{rate}	Score
T1	4	1	666
T2	2	1	458
T3	3	1	642
T4	1	1	104
T5	5	$\frac{6}{5}$	688
T6	3	$\frac{4}{3}$	477

ARGUS. The ARGUS detector is based on physical invariants derived from the design of the SWaT. A "Process invariant," or simply invariant, is a mathematical relationship among "physical" and/or "chemical" properties of the process controlled by the PLCs in an ICS. The invariants serve as checkers of the system state. Those invariants are translated into control code and each PLC is then re-programmed to include the checking code, without affecting the original control logic (e. g., additional layer of protection). The PLC executes the code in a cyclic manner. In each cycle, data from the sensors is obtained, control actions computed and applied when necessary, and the invariants checked against the state variables or otherwise. Distributing the attack detection code among various PLCs helps to scale the implementation of ARGUS. More information about ARGUS can be found in [1, 2].

HAMIDS. The HierArchical Monitoring Intrusion Detection System (HAMIDS) framework is designed to detect network-based attacks on Industrial Control Systems. The framework leverages a set of distributed Intrusion Detection System (IDS) nodes, located at different layers (segments) of an ICS network. The role of those nodes is to extract detailed information about a network segment, combine the information in a central location, and post-process it for real-time security analysis and attack detection. Each node uses the Bro Intrusion Detection System [23]. More information about HAMIDS can be found in [24].

5.4 Live Phase Results and Selected Attacks

Table 3 presents a summary of the results from the S3 live phase (an attack-defense CTF). As for the online phase, we decided to anonymize the team names for privacy concerns. There is one row for each participating team. The second column shows the number of successful attacks. The third column shows the cumulative detection rate d_{rate}, that was computed as the average number of detection mechanisms triggered during a successful attack considering only ARGUS and HAMIDS detectors. The last column shows the final scores for each attacking team. During the competition we noted that the majority of the attacking teams took advantage of the knowledge gained during the online phase presented in Section 4.

To show more in depth insights from the attack-defense CTF, we now provide details about four selected attacks that were conducted by the participants during the S3 live event. We classify those attacks in two types. The *cyber* attacks were conducted over the network using either the cybercriminal, or the strong attacker model. On the other hand the *physical* attacks were conducted having direct access to the SWaT using either the insider, or the strong attacker model. Table 4 presents a summary of the four selected attacks with their score, attack type and detection statistics (considering only ARGUS and HAMIDS). We now describe the four selected attacks:

DoS PLC1 by TCP SYN flooding. The first selected attack is a cyber attack, and the attacker used the insider attacker model. As a reminder, the insider has access to the SWaT administrator account and the engineering tools. The attacker performed a TCP SYN flooding attack on the EtherNet/IP server of the first PLC. SYN flooding is a denial of service attack, where the attacker (the client) continuously tries to establish a new TCP connection sending SYN requests to the target (the PLC EtherNet/IP server). The EtherNet/IP server then responds with a TCP ACK packet, however the attacker never completes the TCP three-way-handshake and continues to send TCP SYN packets. As a result of this DoS attack, the HMI was unable to obtain current state values to be displayed, and it displayed 0 or * characters instead. Such effects is dangerous because it impedes the real-time supervision of the plant. Fortunately, the attack did not interrupt the physical process itself. The HAMIDS detector was able to detect the attack by observing the high number of TCP SYN requests without follow-up. The ARGUS detector was not able to detect the attack, as the physical process was not impacted.

DDoS by distributed ARP spoofing. The second selected attack is a cyber attack, and the attacker used the cybercriminal attacker model. As a reminder, the cybercriminal has access to the SWaT network and common penetration testing tools. The attacker performed a distributed ARP poisoning man-in-the-middle attack, that redirected and dropped all the traffic addressed to the HMI. The attack drove the HMI to an unusable state, and it took a while to restore the system state after the attack. We did not allow the attack to run long enough to affect the physical process. HAMIDS detected the attack because of the amount of traffic redirected to a single host and the presence of malicious ARP traffic. In contrast, ARGUS did not detect the attack, as the physical process continued to operate without impact.

Spoofing over the field network. The third selected attack is a physical attack and it involved an on-site interaction with the field network. The attacker used the strong attacker model and he focused on one of the L0 network segments of the SWaT (see Figure 1). The attacker demonstrated control over the packets sent in the L0 DLR Ethernet ring and he was able to manipulate the communication between the PLC and the RIO in real-time, altering the content of the EtherNet/IP packets. ARGUS was able to detect the attack due to the sudden changes in reported sensor values. In addition, the HAMIDS framework detected the attack by observing the change in data reported from the PLC to the SCADA (and potentially, in L0 as well).

HMI tampering. The fourth selected attack is a physical attack, and the attacker used the insider attacker model. The attacker was able to alter the chemical dosing in the second stage (Pre-treatment) of the SWaT by interacting directly with the HMI interface. The attacker set the PLC in manual mode and overwrote its set of commands by tampering with the HMI. The attack would have resulted

Table 4: SWaT Security Showdown (S3) live phase (attack-defense CTF) selected attacks and detections summary: ○ = Undetected, ● = Detected.

Selected Attack Description	Type	ARGUS	HAMIDS	Score
DoS PLC1 by TCP SYN flooding	Cyber	○	●	396
DDoS by distributed ARP spoofing	Cyber	○	●	104
Spoofing over the field network	Physical	●	●	324
Chemical dosing pump manipulation	Physical	●	○	360

in an eventual degradation of the quality of the water, however we stopped the attack before that case occurred. ARGUS was able to detect the attack because the updated setpoints (sensor values) diverged from their hard-coded counterpart in the detection mechanism. The HAMIDS detection was unable to detect this scenario as the network traffic did not show unusual patterns or changes.

6 RELATED WORK

In [21] Mink presents an empirical study that evaluates how exercises based on gamification and offensive security increase the motivation and the final knowledge of the participants. Our work tries to extend this message to ICS security, while Mink's paper focuses on traditional Information security.

DEFCON [9] is an annual hacking conference organized by information security enthusiasts. The DEFCON CTF is part of the main event, and it is one of the most well known, and competitive CTF contest worldwide. Like S3, it has a Jeopardy-style qualification phase, and an attack-defense final phase. However ICS security is not the main focus of DEFCON's CTF. Several other similar CTF competitions are listed in [10]. In [30] Vigna proposes to use gamified live exercises to teach network security. The motivations and philosophy of this work are similar to ours. However the focus of the paper is on IT network security (e. g., gain root privileges on a webserver or steal data from a SQL database) and not on OT network security (industrial network devices and protocols). Inspired by [30], in [8] authors of the iCTF event presented two novel, live, and large-scale security competitions. The first is called "treasure hunt" and it exercises network mapping and multi-step network attacks. The second is a "Botnet-inspired" competition and it involves client-side web security tasks such as Web browsers exploitation. Unlike the presented paper, both competitions focus on traditional client-server IT network architectures and attack-only scenario.

The MIT/LL CTF [32] was an attack-defense CTF with a focus on web application security. The main goal of the event was to attract more people towards practical computer security exercises. The CTF takes inspiration from Webseclab [6], a web security teaching Virtual Machine that is packed with an interactive teaching web application, a sandboxed student development environment, and a set of useful programs. Both are interesting projects but they are not covering the ICS security domain, even though they share some of the presented goals. BIBIFI [27] is a cyber-security competition held mainly in academic environments that combines in the same contest: secure development (Build-it), attacks development (Break-it) and patch development (Fix-it). This effort was targeted at improving secure software construction education, and thus the exercises proposed in this competition do not cover the ICS security domain.

7 CONCLUSIONS

In this work we discussed problems faced by security experts and ICS engineers in the context of ICS security education and research. In particular, security experts require access to real ICS infrastructures to learn about ICS (security) and practise applied attacks and defenses. In addition, ICS engineers require additional training focused on basic cyber-security concepts and offensive and defensive security techniques. We propose to use gamified security competitions such as online and live Capture-The-Flag to address and mitigate those problems. To demonstrate the feasibility of such events, we designed and implemented the *SWaT Security Showdown (S3)*, leveraging the Secure Water Treatment (SWaT) water treatment testbed.

To the best of our knowledge, the S3 event was the first security competition involving access to live and virtual ICS infrastructures (e. g., MiniCPS). The online phase consisted of a Jeopardy-style CTF that included novel challenges specifically designed for ICS security. For example, we gave to the attacker remote access to a real PLC programming environment (e. g., Studio 5000) and we asked them to understand a ladder logic program. Overall, six participating attacker teams submitted 77 correct flags in the online phase of the S3 event.

In the live phase (an attack-defense CTF), the participating teams performed 18 successful attacks in SWaT within a limited time frame. The timing was an important factor because it increased the level of realism of the competition. During the S3 live phase the teams demonstrated a wide range of different attack approaches, and adapted their attacks to challenges posed by the complexity of the real testbed. In addition we also evaluated several (novel) detection mechanisms including two internally-developed ones (e. g., ARGUS and HAMIDS). Most of the attacks were detected by at least one of our detection mechanisms.

In summary, S3 was an enriching experience for everybody, including us (the organizers). We hope that such event provides a foundation to enable others to run similar ICS security educational experiments in the near future.

8 ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their helpful feedback and suggestions. This work was partially supported by SUTD startup grant SRIS14081, and the iTrust research centre.

REFERENCES

[1] Sridhar Adepu and Aditya Mathur. 2016. Distributed Detection of Single-Stage Multipoint Cyber Attacks in a Water Treatment Plant. In *Proc. of the Asia Conference on Computer and Communications Security (ASIACCS)*.

[2] Sridhar Adepu, Siddhant Shrivastava, and Aditya Mathur. 2016. Argus: An Orthogonal Defense Framework to Protect Public Infrastructure against Cyber-Physical Attacks. *IEEE Internet Computing* (2016).

[3] Daniele Antonioli. [n. d.]. MiniCPS public repository. https://github.com/scy-phy/minicps. ([n. d.]).

[4] Daniele Antonioli, Anand Agrawal, and Nils Ole Tippenhauer. 2016. Towards High-Interaction Virtual ICS Honeypots-in-a-Box. In *Proc. of the Workshop on Cyber-Physical Systems-Security and/or PrivaCy (CPS-SPC)*. ACM.

[5] Daniele Antonioli and Nils Ole Tippenhauer. 2015. MiniCPS: A toolkit for security research on CPS Networks. In *Proc. of the Workshop on Cyber-Physical Systems-Security and/or PrivaCy (CPS-SPC)*. ACM.

[6] Elie Bursztein, Baptiste Gourdin, Celine Fabry, Jason Bau, Gustav Rydstedt, Hristo Bojinov, Dan Boneh, and John C. Mitchell. 2010. Webseclab Security Education Workbench. In *Proc. of Conference on Cyber Security Experimentation and Test (CSET)*.

[7] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry. 2011. Attacks against process control systems: Risk assessment, detection, and response. In *Proc. of the ACM Conference on Computer and Communications Security (CCS)*.

[8] Nicholas Childers, Bryce Boe, Lorenzo Cavallaro, Ludovico Cavedon, Marco Cova, Manuel Egele, and Giovanni Vigna. 2010. Organizing large scale hacking competitions. In *Proveedings of conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*. https://doi.org/10.1007/978-3-642-14215-4_8

[9] Crispin Cowan. 2003. Defcon Capture the Flag: Defending vulnerable code from intense attack. In *Proc. of DARPA Information Survivability Conference and Exposition (DISCEX)*.

[10] ctftime [n. d.]. CTFtime. https://defcon.org/. ([n. d.]). Accessed: 2016-10-19.

[11] E-ISAC and SANS. 2016. Analysis of the Cyber Attack on the Ukrainian Power Grid. (2016). https://ics.sans.org/media/E-ISAC

[12] Nicolas Falliere, L.O. Murchu, and Eric Chien. 2011. W32. stuxnet dossier (Symantec Security Response). http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf. (2011).

[13] SANS institute. 2015. The State of Security in Control Systems Today. (2015). https://www.sans.org/reading-room/whitepapers/analyst/state-security-control-systems-today-36042.

[14] Internet Security Research Group (ISRG). [n. d.]. Let's Encrypt. https://letsencrypt.org/. ([n. d.]).

[15] Karl M Kapp. 2012. *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons.

[16] Perry Kundert. [n. d.]. Communications Protocol Python Parser and Originator. https://github.com/pjkundert/cpppo. ([n. d.]). [Online; accessed 31-July-2016].

[17] Bob Lantz, Brandon Heller, and Nick McKeown. 2010. A network in a laptop: rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. ACM.

[18] Eric Luiijf. 2015. Cyber (In-) security of Industrial Control Systems: A Societal Challenge. In *International Conference on Computer Safety, Reliability, and Security (SafeComp)*. Springer.

[19] Eric Luijif and Bert Jan te Paske. 2015. Cyber Security of Industrial Control Systems. TNO technical report. (2015). https://www.tno.nl/ics-security/.

[20] Aditya Mathur and Nils Ole Tippenhauer. 2016. A Water Treatment Testbed for Research and Training on ICS Security. In *Proc. of Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater)*.

[21] Martin Mink and Rainer Greifeneder. 2010. Evaluation of the offensive approach in information security education. In *Proc. of IFIP International Information Security Conference (IFIP SEC)*.

[22] ODVA. [n. d.]. Ethernet/IP Technology Overview. https://www.odva.org/Home/ODVATECHNOLOGIES/EtherNetIP.aspx. ([n. d.]). Accessed: 2016-08-01.

[23] Vern Paxson. 1999. Bro: a system for detecting network intruders in real-time. *Computer Networks* (1999).

[24] Hamid Reza Ghaeini and Nils Ole Tippenhauer. 2016. HAMIDS: Hierarchical Monitoring Intrusion Detection System for Industrial Control Systems. In *Proc. of Workshop on Cyber-Physical Systems Security & Privacy (SPC-CPS)*.

[25] Marco Rocchetto and Nils Ole Tippenhauer. 2016. On Attacker Models and Profiles for Cyber-Physical Systems. In *Proc. of the European Symposium on Research in Computer Security (ESORICS)*. https://doi.org/10.1007/2F978-3-319-45741-3_22

[26] Armin Ronacher. [n. d.]. Flask: web development, one drop at a time. http://flask.pocoo.org/. ([n. d.]).

[27] Andrew Ruef, Michael Hicks, James Parker, Dave Levin, Michelle L. Mazurek, and Piotr Mardziel. 2016. Build It, Break It, Fix It: Contesting Secure Development. In *Proc. of the ACM Conference on Computer and Communications Security (CCS)*. http://arxiv.org/abs/1606.01881

[28] Floris A Schoenmakers. [n. d.]. Contradicting paradigms of control systems security: how fundamental differences cause conflicts. http://rpository.tudelft.nl/. ([n. d.]).

[29] Jill Slay and Michael Miller. 2007. *Lessons learned from the maroochy water breach*. Springer.

[30] Giovanni Vigna. 2003. Teaching network security through live exercises. In *Security education and critical infrastructures*. Springer.

[31] S. Weerakkody, Yilin Mo, and B. Sinopoli. 2014. Detecting integrity attacks on control systems using robust physical watermarking. In *Proc. of Conference on Decision and Control (CDC)*. IEEE.

[32] Joseph Werther, Michael Zhivich, Tim Leek, and Nickolai Zeldovich. 2011. Experiences in Cyber Security Education: The MIT Lincoln Laboratory Capture-the-flag Exercise. In *Proc. of the Conference on Cyber Security Experimentation and Test (CSET)*. USENIX Association.