CLEI ELECTRONIC JOURNAL, VOLUME 22, NUMBER 3, PAPER 7, DECEMBER 2019

Automatic Challenge Generation for Teaching Computer Security

Ricardo de la Rocha Ladeira
Catarinense Federal Institute, Teaching Department, Campus Blumenau,
Blumenau, Santa Catarina, Brazil, 89070-270
ricardo.ladeira@ifc.edu.br

and

Rafael R. Obelheiro
State University of Santa Catarina (UDESC), Graduate Program in Applied Computing (PPGCA),
Campus Joinville, Joinville, Brazil, 89219-710
rafael.obelheiro@udesc.br

Abstract

Computer Security is an increasingly important area, considering the sophistication and spread of threats present in the digital world. The need for information protection contrasts with the lack of professionals and the limited space dedicated to the area in Information Technology (IT) courses. Games and competitions have been used to motivate computing students to improve their practical knowledge on the subject and also to foster the interest of potential students and professionals in Security. The creation of these games requires specialized knowledge to develop new problems, since the novelty of these games is important to reach the desired level of difficulty and to ensure competitiveness. This work proposes the use of randomization to generate problems and entire competitions in an automated way, obtaining exclusive instances of problems for each player. A tool for generating challenges was developed as proof of concept to evaluate the proposal. Competitions with automatically generated problems were held with undergraduate and continuing education students, at two different institutions. The performance in the competitions and the perception of satisfaction, interest and learning of the involved students were analyzed. The results show that the automatic generation of challenges is feasible and the use of competitions to teach Computer Security is motivating and effective for didactic purposes.

Keywords: Automatic Problem Generation. Computer Security. Teaching. Treasure Hunt. Game.

1 Introduction

Computer Security is an increasingly important and necessary area because of technology ubiquity and the need to protect information. This protection involves not only technologies but also knowledge and adoption of good security practices by those who develop, manage and use computer systems that manipulate information [1].

Studies have shown that building and sustaining a Cybersecurity culture is important in times of radical change [2]. However, the need to disseminate Cybersecurity is often neglected by many institutions. In addition, individuals trained in Computer Security are needed, but the market lacks this type of professional [3], and there is a great demand for training in the area [4].

Formal education, especially at the undergraduate level, has struggled to fill this gap. One reason is the lag of curricula, which still need to be adapted to the latest guidelines. In Brazil the Ministry of Education instituted curricular guidelines for Computing courses only in 2016, considering Security one of the topics that must be present among the skills and competencies for professional training [5]. The report for curricular guidelines for IT courses, developed in 2017 by ACM (Association for Computing Machinery) and IEEE-CS (Institute of Electrical and Electronics Engineers Computer Society), suggests that Cybersecurity should emerge in new curricula [6]. The current reality is that the few courses that deal with Computer Security have their classes designed to discuss many issues, in a generic and conceptual way, or to cover a few topics, but with reasonable depth. Many are taught in a traditional way, using books and lectures, with a theoretical approach [7]. It is thus perceived the need to intensify Security training, emphasizing not only theory but also practice to arouse students' interest in the subject.

To attract students and future professionals to the Cybersecurity field it is important to teach using methodologies that encourage the student. Games and competitions are motivating ways to introduce Cybersecurity concepts. Due to the extra motivation provided by a competitive environment, Security competitions have become increasingly popular [8].

Challenge (or treasure hunt) is one of the most common types of Computer Security games. The goal is to find secret words (flags) hidden in (set of) files using computational techniques and tools. In challenges the problems usually are created by hand, which imposes some difficulties. Problem creation is a laborious task and requires specialized technical knowledge, which limits the popularization of this type of game due to the scarcity of available human resources. It is common for challenges to be published on the Internet after they are held, often with solutions to problems, making it infeasible to reuse exercises, because part of the game difficulty lies in finding a strategy to solve each problem. In addition, manually created problems often have a unique flag, which is the same for all players. As flags are used as proof of problem solving, this uniqueness allows them to be copied and/or shared between players, allowing players to score points even without actually solving the corresponding problems. Finally, it is observed that the aforementioned difficulties reinforce each other: the complexity of creating problems is compounded by the loss of value after its publication and the sharing of flags.

In order to reduce the mentioned difficulties and facilitate the creation and promotion of challenges, this work proposes the automatic generation and composition of problems for this type of game. The idea is to automate the problems generation from a list of techniques that can be applied in an individual or composite way and then generate whole competitions formed by equivalent sets of problems, but with unique flags for each player. Automatic problem generation has already been addressed in other works [9, 10]. Our contributions are the more extensive use of compound problems and the generation of entire competitions (problem sets), not only of individual problems.

This paper is organized as follows: Section 2 presents a general review of games used in Security teaching and learning. Section 3 discusses the main related work. Section 4 details the automatic generation of Security problems. Section 5 shows the evaluation performed and the results obtained, and Section 6 concludes the paper.

2 Computer Security Games

Inserting games to raise awareness in Cybersecurity is a practice that has been gaining traction [11, 12]. Regarding the pedagogical value, it is not in the competitions that most of the skills are acquired [8], but in preparation time through training, studies and exchange of experiences. There are indications of an increase in the technical skills of game participants described in various works [13, 14, 15], but the advantages of these activities involve other aspects such as leadership, teamwork (for non-individual games) , decision-making and time control [16].

Different types of games are used for teaching Cybersecurity, including

- video games [17, 18];
- board and card games [19, 20];
- attack and defense competitions [8, 11, 14]; and
- challenges (treasure hunt) [12].

Security challenges, also called treasure hunts, consist of sets of problems that need to be solved with processes and tools, typically without interaction with other players. They motivate players to find secret resources and at the same time consist of activities that can be easily reproduced because they require a limited number of tools to enable their solution, such as network traffic analyzers and binary file editors.

Problems in treasure hunting competitions do not involve direct interaction between players/teams, but promote competition based on scoring systems. For example, players/teams that first accomplish a task receive more points than those that finish later, following a decreasing scale [8], or everyone that solves a problem inside a given time limit is awarded the same points. Establishing fixed targets for all teams promotes competition in a healthier way than occurs in games in which teams attack each other [12]. In addition, challenges are flexible enough to enable individual practice as well.

Challenges can be linear, non-linear or mixed. In a linear challenge, problems need to be solved in order, and usually have progressive difficulty [21]. In a non-linear challenge, problems are independent and can be solved in any order. In this type of challenge, problems can have different scores depending on their difficulty [22]. The format can still be mixed by grouping problems into sequential (hence linear) phases, with each phase having a set of problems that can be solved in any order [23].

Challenges have been applied for some years outside the academic context. The US Department of Defense promoted the DC3 Digital Forensics Challenge [24] between 2006 and 2013, a challenge open to participants from around the world. Experiences in Brazil include Cryptorace[1] since 2015 and Roadsec's Hackaflag[2] and SBSeg's Computer Forensics Workshop in 2015 and 2016. The target audience for these challenges, however, is not well-defined and can blend students and professionals.

Comparing the types of games, video games and board and card games are generally more concerned with raising awareness of players through concepts than with current technologies. Attack and defense games require more in-depth knowledge, such as developing codes to exploit vulnerabilities and fixing vulnerable services, and their implementation involves complex logistics, including sophisticated computational infrastructure and highly specialized staff to create a game and maintain infrastructure [8, 12, 25]. Challenges require practical knowledge with less depth than attack and defense, and are more flexible in that they can involve problems with varying levels of complexity. In this way, challenge-type competitions can offer competitiveness to skilled players and at the same time allow beginners to make progress and stay motivated.

Novelty is a motivating factor when present in games but also brings a difficulty to game creators. Problem solutions can often be found online after a competition is held, making the problems practically disposable due to losing their surprise factor. This issue is called problem sharing [25]. In addition, during a competition, teams may submit flags that were actually found by other teams, which is called flag sharing [9]. This occurs because some participants might not be interested in winning, just in making progress without regard for their score. One way to overcome these difficulties is to automate the generation of problems that make up a challenge, using randomization to make the generated problems unique.

3 Related Work

In this section we discuss more closely works that apply Automatic Problem Generation (APG) to Computer Security challenges.

PicoCTF [9, 23] is a student-focused challenge, held annually. Developed since 2014 in Carnegie Mellon University, it has used APG for individual problems to mitigate flag sharing, and has been identified as the pioneering work in this area. The papers that discuss PicoCTF [9, 23] do not describe the tools needed to create and solve the problems proposed.

MetaCTF [10] is a challenge game that incorporated APG from 2015 onwards to lower the odds of cheating. The competition goal was to teach Reverse Engineering and malware analysis, requiring expertise in tools such as *objdump*, *readelf* and *gdb*. The work evaluated the quality and usefulness of pre-competition extra-class tasks, resulting in a significant increase in students' performance on those tasks when the game was applied.

SecGen [26] is a tool capable of generating random challenges in virtual machine sets (VMs), called rich scenarios. The work has a great diversity of problems, such as network services, steganography and system vulnerabilities. Each player receives a VM with a different operating system and distinct sets of problems in a competition, which minimizes the sharing of techniques and flags. The results indicate that the tool was well received by the students, and that the problems had an adequate level of difficulty. However, the fact that the problems are not uniform can generate an imbalance between the competitors, causing certain players to be favored or impaired due to their knowledge and the problems drawn in their respective challenges. For example, a player with a broad set of knowledge can be harmed if the challenge is focused on a sub-area with which he is unfamiliar; similarly, a player with limited knowledge can benefit if he is challenged with problems concentrated in an area with which he is familiar.

All these related works have scoreboards, adopt a non-linear game flow and provide unique problem instances. However, although PicoCTF uses APG, its problems are restricted to a single technique. MetaCTF has used APG restricted to a class of problems and has the possibility of encoding the flag of a given problem, thus

[1] http://roadsec.com.br/cryptorace/

[2] https://roadsec.com.br/hackaflag/

providing limited composition of techniques. In SecGen, the only way of nesting techniques is to encode a flag before hiding it using another technique. Our proposal applies composition more broadly avoiding only unworkable combinations of techniques. In addition, instances of problems have the same complexity while SecGen and MetaCTF allow players to receive totally different problems, affecting the score and the effort required to solve the exercises

Table 1 presents a comparison between the related works and TreasureHunt, which is our proposal. The table shows the differences and similarities regarding automatic generation, composition of problems, uniformity of problems and classes of problems addressed. The automatic generation of problems is seen in all works, with the proviso that PicoCTF generates only isolated problems, not entire competitions. Regarding the generation of problems using technique composition, PicoCTF does not have this characteristic, whereas MetaCTF and SecGen limit this composition to flag encoding after generating a problem using another technique (reason why they are represented with the symbol ± in Table 1). TreasureHunt applies composition more broadly, avoiding only unfeasible combinations of techniques, as will be shown in the following sections.

The uniformity of problems concerns the application of problems and techniques of similar complexity to all players. In SecGen, players can receive totally different virtual machines, with distinct operating systems and classes of exercises and problems. This means that familiarity or difficulty with a certain technique can influence the scoreboard. MetaCTF generates problems with the same technique, since it only works with Reverse Engineering, but they are not uniform with respect to complexity. In this way, a player receives an instance that may be more or less complex than the instances of other players, also influencing the score and the amount of effort required to solve the exercise.

The list of problem classes addressed in each work consider only the classes actually implemented, as described in the references, and not all classes that could be implemented. It is observed that the MetaCTF is restricted to Reverse Engineering problems, while the other works approach a more extensive set of classes of problems.

Table 1: Comparison between the related works and the proposed tool.

Work	Automatic Generation	Composing Problems	Uniformity of Problems	Classes of Problems
PicoCTF	problems	×	✓	● Reverse Engineering ● Web ● Miscellaneous ● Encoding/Encryption
MetaCTF	competitions	±	×	● Reverse Engineering
SecGen	competitions	±	×	● Web ● Forensics ● Miscellaneous ● Encoding/Encryption
TreasureHunt	competitions	✓	✓	● Reverse Engineering ● Forensics ● Miscellaneous ● Encoding/Encryption

4 Automatic Challenge Generation

Security competitions usually only attest the players' knowledge, bringing the ideas of competition and prevailing. In other words, the target audience is usually composed of individuals with knowledge in the subject and who only seek to win the competition [27]. In an educational setting, competing is not the main objective, but a means of positively influencing the teaching and learning processes, taking into consideration that the players are also students. Thus, encouraging players to deepen their knowledge of Security and pursue a career in the field is just as important as recognizing the most capable ones.

In Cybersecurity competitions, players can participate as individuals or in teams. In this work we opted for the first case, to be able to evaluate each player individually. However, it is possible to play in teams, at the discretion of the organizer. The proposed competition is non-linear, that is, problems can be solved in any order. For scoring purposes, all problems are worth one point, thus, only the number of correct flags is counted. For players who find the same number of flags, the time of the last correct submission is used as a tiebreaker.

For building a prototype tool for automatic challenge generation, we analyzed 84 competitions between 2016 and 2017 described in a repository of challenges maintained by the community [28], totaling 1250 problems covering about 200 different techniques. The 40 most frequent techniques were identified, of which eight were chosen. The criteria for selecting techniques were the educational goal of the work and the profile of our target audience, formed by students without advanced knowledge in Linux and Computer Security. We deliberately chose techniques that could produce problems with easy-to-medium difficulty, so that even beginners could expect to solve at least one problem in a competition. However, the tool can be easily adapted to generate more advanced problems, if so desired. This would require that the implementer figures out how to randomize problem instances.

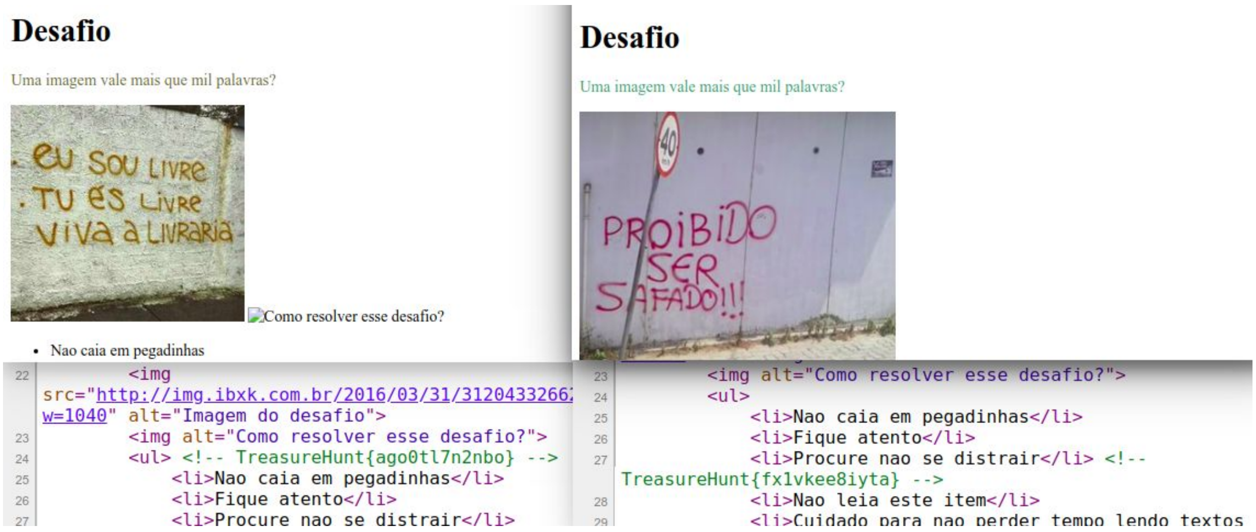
The techniques implemented in the prototype were:

- **HTML comment**: problem where the flag is inserted as a comment in a random location of a valid HTML file. The page loads different styles and images in each instance.
- **Comment in the robots.txt file**: A problem that contains a set of files from a website and, among them, a robots.txt file containing a parameterizable number of comments. All comments are random sequences of characters, and one matches the flag.
- **Base64 encoding/decoding**: problem where the flag is inserted into a text file chosen at random from a user-defined directory. After that the file is encoded in Base64 format.
- **Caesar Cipher encryption/decryption**: problem in which the flag is inserted in a text file chosen at random from a user-defined directory. After that the file is encrypted by Caesar Cipher with a key randomly chosen between 1 and 25.
- **ASCII-to-decimal encoding/decoding**: problem in which the flag is inserted into a text file chosen at random from a user-defined directory. The file is then converted to a sequence of decimal numbers, with each character being replaced by its ASCII value.
- **Java decompilation**: problem in which a Java code, which produces arbitrary text, is created. In this code a flag is inserted into a variable and the code is compiled into byte-code, generating a .class file.
- **Python decompilation**: similar to the previous problem, but using Python instead of Java, and generating a .pyc file instead of a .class one.
- **Image steganography**: problem in which the flag is embedded in an image chosen at random from a user-defined directory using a steganographic tool.

Figure 1 presents two instances of the "*HTML comment*" problem generated by the tool, and their respective source codes. Note that the flags are on different lines (line 24 on the left and line 27 on the right) and have distinct symbol sequences, although they are the same size. The displayed images and page style are random, so each user receives a file with different settings.

Because flags are distinct, sharing responses has no effect. On the other hand, the same strategy can be used to find the answer. The possible sharing of resolution procedures is a good thing, as it allows players to progress through the game and learn new techniques and tools with their peers.

The challenge generator also compounds techniques, generating distinct instances for each player. The composition of techniques consists on applying a technique and then submit the flag or the output file(s) to the application of a second technique — which may be the same again — to then generate new file(s). For example, a problem with a composition between *Base64 encoding/decoding* and *Caesar Cipher encryption/decryption* will apply Base64 encoding to a file, generating an intermediate output file. This file will be encrypted using the Caesar Cipher, generating a new output file that composes the two techniques in a problem, and which is provided to a player. Thus, to solve the exercise, the player must discover the key used in the Caesar Cipher, obtain the file encoded in Base64, decode it and look for the flag.


Desafio
Uma imagem vale mais que mil palavras?
eu sou livre
tu és livre
viva a livraria
Como resolver esse desafio?
Desafio
Uma imagem vale mais que mil palavras?
40
PROIBIDO SER SAFADO!!!

- Nao caia em pegadinhas

```
        <img
src="http://img.ibxk.com.br/2016/03/31/3120433266
w=1040" alt="Imagem do desafio">
        <img alt="Como resolver esse desafio?">
        <ul> <!-- TreasureHunt{ago0tl7n2nbo} -->
            <li>Nao caia em pegadinhas</li>
            <li>Fique atento</li>
            <li>Procure nao se distrair</li>
```

```
        <img alt="Como resolver esse desafio?">
        <ul>
            <li>Nao caia em pegadinhas</li>
            <li>Fique atento</li>
            <li>Procure nao se distrair</li> <!--
TreasureHunt{fx1vkee8iyta} -->
            <li>Nao leia este item</li>
            <li>Cuidado para nao perder tempo lendo textos
```

Figure 1: Two instances of the "*HTML comment*" problem and their respective source code.

It should be noted that not all techniques can be compounded. For example, the composition (*Caesar Cipher encryption/decryption* ◦ *Caesar Cipher encryption/decryption*) is not feasible, since, for this cipher, encrypting a string multiple times is equivalent to encrypting it only once with a different key. Our challenge generator produces only viable compositions.

It is important to mention that the order of application of the techniques influences the generated instance. For example, applying Base64 encoding before a code decompilation problem involves generating a flag, encoding it, and then inserting it into the source code. If the order is reversed, the generated flag is inserted into the source code and, after compilation, the bytecode file is Base64 encoded.

Figure 2 shows an instance of the problem that composes *Image steganography* and *Base64 encoding/decoding*. In it the player receives a file encoded in Base64 and, upon decoding it, obtains an image that contains a text embedded with the *outguess* tool. After extracting the image content with *outguess*, the flag is obtained. The generated flags and the images used in each instance are different, which produces different files when encoding in Base64.

The interaction with the problem generation tool is done by the activity organizer, who supplies the parameters for creating the desired problems, which are hosted on a web server and accessed by a web application. The organizer assigns identifiers (IDs) to players, who use the web application to download their set of problems and to submit flags. The web application also provides basic instructions, and individual and general scoreboards for monitoring the competition.

5 Evaluation

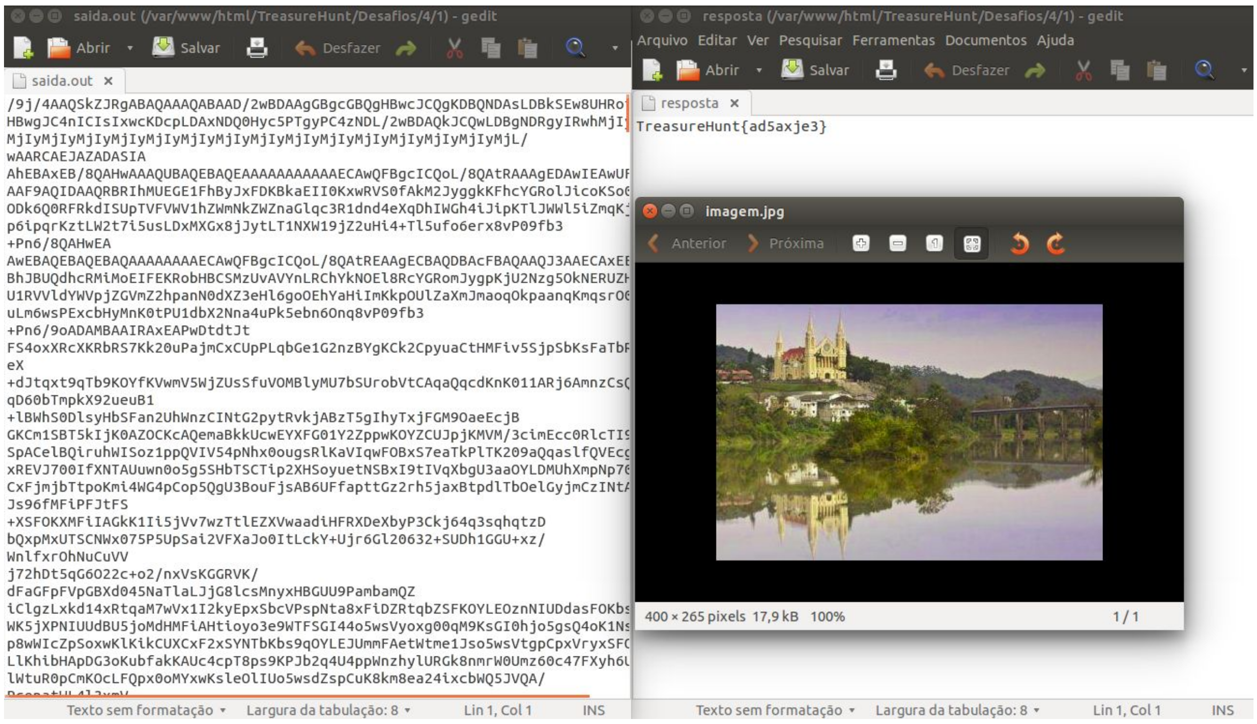
To evaluate the effectiveness of the proposal in generating distinct challenges without trivializing the problems and the perception of students' satisfaction, learning and interest, an experiment was performed, as described below.

5.1 Activity Description

The experiment was a competition in an academic environment where an organizer generated challenges using our prototype tool. The experiment was carried out in three stages, one preparatory class and two competitions. The competitions were individual, lasting 1h30min, held in a computer lab with machines previously configured with the necessary tools.

In the preparatory class[3] we explained the dynamics of the challenge and presented a set of tools that could be used to solve problems, with examples and exercises. At the beginning of the class the students signed a Consent Form about their participation in the experiment, and answered two questionnaires, a profile survey[4] and a pre-test survey[5], identifying their impressions before the competition.

[3] Files available in repository: https://github.com/TreasureHuntGame/TreasureHunt.
[4] Available at: http://bit.ly/2AdEWMW.
[5] Available at: http://bit.ly/2AoVkew.


saida.out (/var/www/html/TreasureHunt/Desafios/4/1) - gedit
Abrir
Salvar
Desfazer
saida.out
Texto sem formatação
Largura da tabulação: 8
Lin 1, Col 1
INS
resposta (/var/www/html/TreasureHunt/Desafios/4/1) - gedit
Arquivo Editar Ver Pesquisar Ferramentas Documentos Ajuda
resposta
imagem.jpg
Anterior
Próxima
400 × 265 pixels 17,9 kB 100%
1 / 1

```
/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw8UHRo
HBwgJC4nICIsIxwcKDcpLDAxNDQ0Hyc5PTgyPC4zNDL/2wBDAQkJCQwLDBgNDRgyIRwhMjI
MjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjL/
wAARCAEJAZADASIA
AhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUF
AAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo
ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKj
p6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3
+Pn6/8QAHwEA
AwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEE
BhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZH
U1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO
uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3
+Pn6/9oADAMBAAIRAxEAPwDtdtJt
FS4oxXRcXKRbRS7Kk20uPajmCxCUpPLqbGe1G2nzBYgKCk2CpyuaCtHMFiv5SjpSbKsFaTbF
eX
+dJtqxt9qTb9KOYfKVwmV5WjZUsSfuVOMBlyMU7bSUrobVtCAqaQqcdKnK011ARj6AmnzCs
qD60bTmpkX92ueuB1
+lBWhS0DlsyHbSFan2UhWnzCINtG2pytRvkjABzT5gIhyTxjFGM9OaeEcjB
GKCm1SBT5kIjK0AZOCKcAQemaBkkUcwEYXFG01Y2ZppwKOYZCUJpjKMVM/3cimEcc0RlcTI
SpACelBQiruhWISoz1ppQVIV54pNhx0ougsRlKaVIqwFOBxS7eaTkPlTK209aQqaslfQVEcg
xREVJ700IfXNTAUuwn0o5g5SHbTSCTip2XHSoyuetNSBxI9tIVqXbgU3aaOYLDMUhXmpNp7
CxFjmjbTtpoKmi4WG4pCop5QgU3BouFjsAB6UFfapttGz2rh5jaxBtpdlTbOelGyjmCzINt
Js96fMFiPFJtFS
+XSFOKXMFiIAGkK1Ii5jVv7wzTtlEZXVwaadiHFRXDeXbyP3Ckj64q3sqhqtzD
bQxpMxUTSCNWx075P5UpSai2VFXaJo0ItLckY+Ujr6Gl20632+SUDh1GGU+xz/
WnlfxrOhNuCuVV
j72hDt5qG6O22c+o2/nxVsKGGRVK/
dFaGFpFVpGBXd045NaTlaLJjG8lcsMnyxHBGUU9PambamQZ
iClgzLxkd14xRtqaM7wVx1I2kyEpxSbcVPspNta8xFiDZRtqbZSFKOYLEOznNIUDdasFOKbs
WK5jXPNIUUdBU5joMdHMFiAHtioyo3e9WTFSGI44o5wsVyoxg00qM9KsGI0hjo5gsQ4oK1Ns
p8wWIcZpSoxwKlKikCUXCxF2xSYNTbme1Jso5wVtgpCpxVrxxSFC
LlKhibHApDG3oKubfakKAUc4cpT8ps9KPJb2q4U4ppWnzhylURGk8nmrW0Umz60c47FXyh6L
lWtuR0pCmKOcLFQpx0oMYxwKsleOlIUo5wsdZspCuK8km8ea24ixcbWQ5JVQA/
```

Figure 2: Instance of the compound problem (*Image steganography* ∘ *Base64 encoding/decoding*).

Competition 1 (C1) took place in the next class meeting. The rules of the challenge were explained, the player IDs and the access address to the web application were distributed, and then the competition was started. All players were given instances of problems with the same techniques. The choice of problems was based on the possible compositions, in order to use all the techniques that were implemented in the challenge generator tool. Six problems were generated, two simple problems (using a single technique) and four compound problems (combining two techniques). The choice of simple and compound exercises was made to be able to analyze the performance difference between these types. The available time for the activity was decisive in choosing the number of exercises, so each problem could be solved in 15 min on average. The exercises of this competition were:

- Caesar Cipher encryption/decryption ∘ HTML comment
- Python decompilation
- ASCII-to-decimal encoding/decoding
- Comment in the robots.txt file ∘ Image steganography
- Image Steganography ∘ Image Steganography
- Base64 encoding/decoding ∘ Java decompilation

Competition 2 (C2) was held in the third class meeting. Again the challenge was explained, and new IDs were distributed to the players. The challenge was similar to that of Competition 1, but half of the students solved the same problems (Group C2.1) that were in C1, in a different order, and the other half (Group C2.2) solved problems with the same techniques, but with different compositions. Students were not told about the difference in the exercises. In order to keep the groups balanced, students were divided according to their performance in C1, so that groups C2.1 and C2.2 had approximately the same total number of points in C1. In case of an odd number of players, group C2.2 would have one player more. The new set of exercises was chosen by the organizers to contain two simple problems and four compound ones, with similar complexity. At the end of this session, students answered a post-test questionnaire[6], recording their impressions after the two competitions. The exercises of this competition for Group C2.2 were:

- ASCII-to-decimal encoding/decoding ∘ HTML comment
- Caesar Cipher encryption/decryption ∘ Comment in the robots.txt file
- Java decompilation
- Base64 encoding/decoding ∘ Python decompilation

[6] Available at: http://bit.ly/2lYORDL.

```
TreasureHunt{ad5axje3}
```

- Image Steganography
- Base64 encoding/decoding ◦ Image Steganography

The activities were carried out with three classes:

- **BCC**: students enrolled in the Network Security course of the Computer Science undergraduate degree at the State University of Santa Catarina (UDESC);
- **TADS**: students from the sixth (last) semester of the Technology in Analysis and Systems Development undergraduate degree at the Catarinense Federal Institute (IFC); and
- **FIC**: continuing education students enrolled in the Computer Security course of the Professional Qualification Course in Computer Networks at the IFC.

In all classes the set of exercises of the two competitions was the same. There were 30 players in total, 7 in the FIC class, 10 in the TADS and 13 in the BCC. The activities were carried out by the authors of this work, in the condition of organizers, during November, 2017.

5.2 Student Profile

The profile questionnaire was answered by all 30 students. The mean age was 23.0 years, with predominance of male students (93.3%) from public schools (76.7%).

To understand if the exercises were in accordance with the students' background and expectations, we included questions about their experience with Linux and Computer Security and their motivation for enrolling in the course. 43.3% reported moderate familiarity with Linux, feeling reasonably comfortable with the command line interface and having written simple scripts. 36.7% answered being somewhat familiar, 13.3% slightly familiar and 6.7% extremely familiar. No one answered to be anything familiar. Twenty students (66.7%) said they did not have experience with Cybersecurity. Six (20.0%) reported having studied on their own, four (13.3%) said they were taking the course again and one (3.3%) took a course outside the institution. The results confirm the assumption that the players did not have advanced knowledge, but would be able to solve the proposed tasks.

Regarding the motivation for attending the course, 56.7% answered that they found the subject interesting, but they did not have much knowledge, 6.7% intended to enhance their formation, 13.3% did to fulfill degree requirements, 13.3% wanted to pursue a career in Cybersecurity or in a related area, and 10.0% gave other answers.

5.3 Performance Results

To evaluate students' performance, we analyzed the number of correct answers, the proportion of correct answers, the hit hate per problem, the hit hate per technique and the time to solve all problems. As the variables were not normally distributed, two nonparametric statistical tests were used to compare the performance in the competitions, the Wilcoxon rank-sum test and the Wilcoxon signed-rank test. The first is a test for unpaired samples, which compares groups, and was used to compare the data of groups C2.1 and C2.2 in Competition 2, since there is no intersection between these groups. The second is a test for paired samples, which compares the evolution of individuals, and was used to compare the competitions C1 and C2. In all tests, the level of significance $\alpha = 0.05$ was adopted.

5.3.1 Number of Correct Answers

Figure 3 shows the score (number of correct answers) of the three classes in the two competitions, using boxplots. The plots show that the median for the FIC class was 3.00 points, for the TADS class was 4.50 points, and for the BCC class was 6.00 points. In all groups there were students that achieved the maximum score, whereas there was one player who scored zero points in each of the TADS and BCC classes. The overall result shows that 75.0% of students solved at least half of the problems, and that 25.0% solved all of them. This performance allows to conclude that the complexity of the exercises was adequate for the FIC and TADS classes. For the BCC class the complexity was less adequate, judging by the number of players (seven) who had already scored maximum points in the first competition. The plots also show that from C1 to C2 the median for the FIC class increased by 2.00 points and for the TADS class increased by 1.50 points, while for the BCC group it remained the same. Based on the general results, it is noticed that the median was 6.00 points. In all, 60.0% (18 of 30) of the students reached the maximum score. Figure 3 shows that student performance has improved and, for most, it has been easy to solve the exercises.

The Wilcoxon test for paired samples identified a statistically significant difference between the number of hits in Competitions 1 and 2 ($V = 4.5$, $p = 0.00093 < 0.05$). Therefore it has that the players' performance in the C2 was different and superior to the performance in C1.

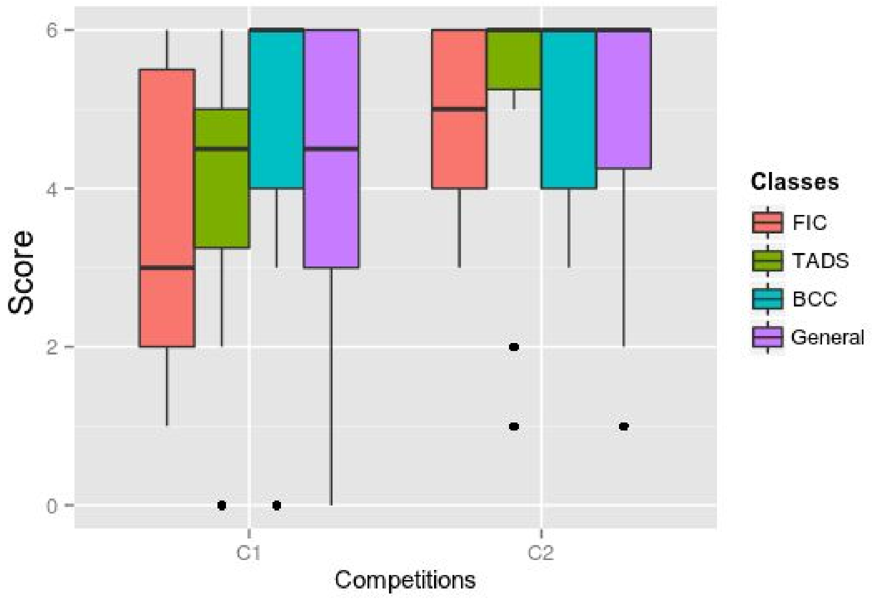


Figure 3: Boxplots of scores in the two competitions.

Figure 4 shows the score comparison for the two groups in Competition 2 (C2.1 and C2.2) in all classes. Note that in both C2.1 and C2.2, most players obtained the maximum number of correct answers since the median in all cases was 6.00 points, except for the FIC class. Group C2.2 presented greater variability in the number of correct answers and a larger number of low scores, but in general, the performance was considered high for all groups. The Wilcoxon test for unpaired samples in C2.1 and C2.2 did not indicate a statistically significant difference ($W = 125.5$, $p = 0.54 > 0.05$). Therefore the difficulty decreased in the second competition, regardless of the group (C2.1 or C2.2).

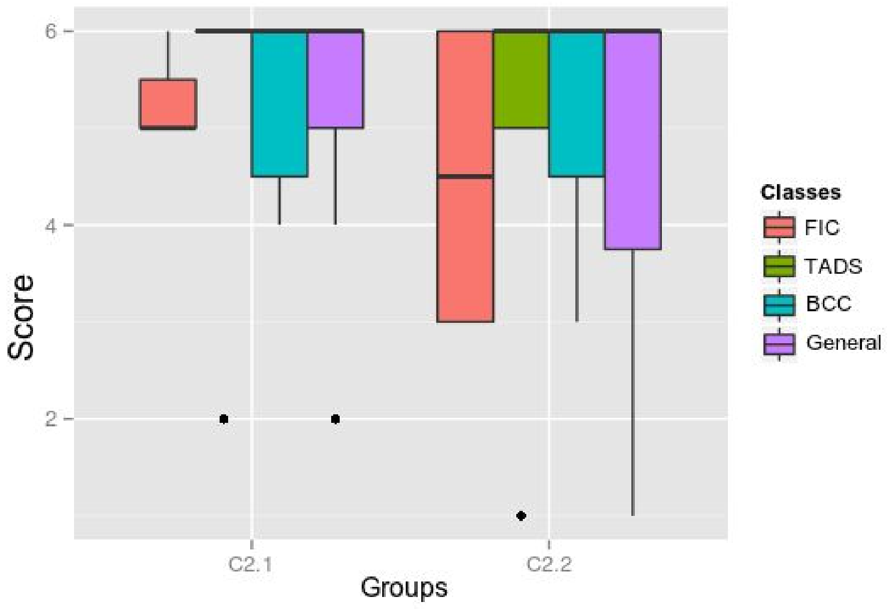


Figure 4: Score comparison between groups C2.1 and C2.2 for all classes.

5.3.2 Proportion of Correct Submissions

The proportion of correct submissions per class was calculated by dividing the total of correct flags by the total number of submissions made. This number may be different from that presented in Section 5.3.1 because players may make a number of attempts until they arrive at the correct answer, which may reveal misunderstanding of game rules, issues with the submission system, or cheating attempts (flag sharing, guessing or even a brute force attack). Figure 5 shows the proportion of correct submissions (in %) for all classes in both competitions.

The overall average in Competition 1 was 75.3% and Competition 2 was 82.1%. From Figure 5 it is possible to notice that the proportion of correct submissions was higher in the second competition, except for the BCC class. Some errors were due to the insertion of white spaces in the flags, and no students submitted flags already sent by other colleagues, that is, there was no flag sharing.

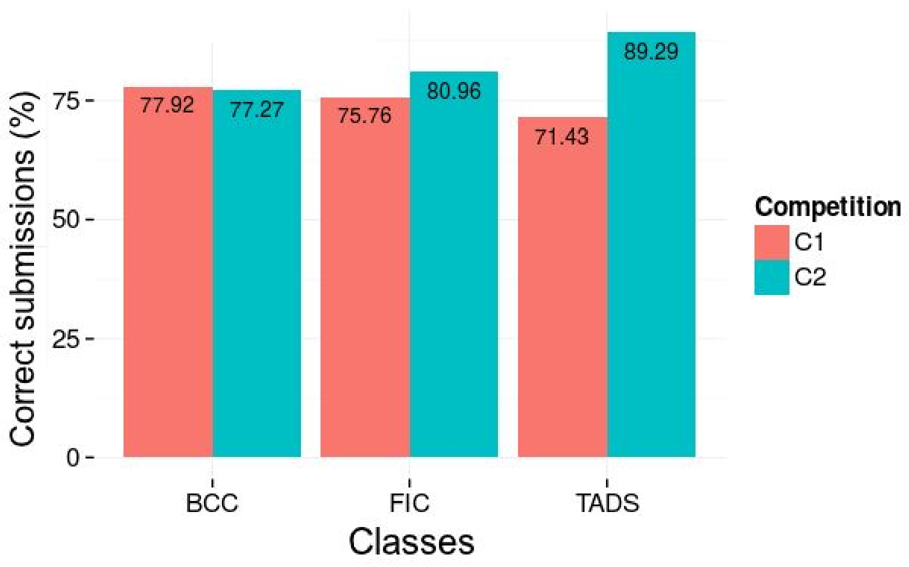


Figure 5: Proportion of correct submissions in the two competitions for the three classes.

5.3.3 Success Ratio per Problem

We analyzed the success ratio (proportion of students with correct answers) for each problem in the two competitions. Tables 2, 3 and 4 show, respectively, the exercises with most hits (correct answers) in Competition 1, Competition 2 for group C2.1 and for group C2.2.

For both groups in Competition 2, the results suggest that simple (non-compound) problems were the easiest. The individual problems of Java and Python decompilation were among the most profitable problems in all games. The most successful exercise in C1 was *Python decompilation*, solved by 93.3% of the players. This exercise was also the most successful in C2.1, along with *Caesar Cipher encryption/decryption ◦ HTML comment*, with 100.0% of hits. In C2.2, the *Java decompilation* exercise was the most successful with 100.0% of hits. The justifications for the ease with the Java and Python problems may be related to the amount of possible solutions with the set of tools presented in the preparatory class (for example, with *strings* and *cat*). The high success in solving the *Caesar Cipher encryption/decryption ◦ HTML comment* problem may have occurred because students know the structure of an HTML page, so that words between the "<! - -" and "- ->" signs would represent comments between the tags. The comment contains the symbols "{" and "}", which allows the player to infer that it is the flag encrypted with the Caesar Cipher, which only substitutes letters and not other symbols.

Analogously, the problems for which players have achieved the least success in terms of correctness have also been identified. Tables 5, 6, and 7 show the lowest achievement exercises in Competition 1, Competition 2 for group C2.1 and for group C2.2, respectively.

Table 2: Exercises with more success in C1.

Class	Problem(s)	Hits	%
TADS	*Caesar Cipher encryption/decryption ◦ HTML comment* *Python decompilation*	9	90.0
FIC	*Python decompilation*	7	100.0
BCC	*Caesar Cipher encryption/decryption ◦ HTML comment* *Python decompilation*	12	92.3
Overall	*Python decompilation*	28	93.3

Table 3: Exercises with more success in C2.1.

Class	Problem(s)	Hits	%
TADS	*Caesar Cipher encryption/decryption ◦ HTML comment* *Python decompilation*	5	100.0
FIC	*ASCII-to-decimal encoding/decoding* *Base64 encoding/decoding ◦ Java decompilation* *Caesar Cipher encryption/decryption ◦ HTML comment* *Python decompilation* *Comment in the robots.txt file ◦ Image steganography*	3	100.0
BCC	*Caesar Cipher encryption/decryption ◦ HTML comment* *Python decompilation*	6	100.0
Overall	*Caesar Cipher encryption/decryption ◦ HTML comment* *Python decompilation*	14	100.0

Table 4: Exercises with more success in C2.2.

Class	Problem(s)	Hits	%
TADS	*Java decompilation*	5	100.0
FIC	*Java decompilation*	4	100.0
BCC	*Image steganography* *Java decompilation*	7	100.0
Overall	*Java decompilation*	16	100.0

Table 5 shows that *Comment in the robots.txt file ◦ Image steganography* was the least proficient exercise, obtaining only 12 hits in the three classes, among the 30 participants. Exercises with the problem *Comment in the robots.txt file* composed with *Image Steganography* are complex because the player receives a set of files, not a single file, as in all other problems implemented in the prototype. Before looking for the flag itself, the student must guess the file it is in and identify that the robots.txt file is an image. Therefore, it is necessary to find the file and still use the *outguess* tool to extract the flag from the image.

In Competition 2, the exercises for which group C2.1 obtained fewer hits are presented in Table 6. In C2, group C2.1 had more difficulty in solving the *Image steganography ◦ Image steganography* problem, which obtained 9 hits from the total of 14 participants. The difficulty with the problem composite *Image Steganography* can be related to the various possible tools to find texts in images (for example, *cat* and *strings*), with attempts to use the *outguess* with a password, or the lack of perception that the extracted file is an image. Therefore, there is still an intermediate step after performing the first extraction, which consists in checking the type of the recovered file. By not doing this or assuming (mistakenly) that it is a text file, the player may find it difficult to solve the exercise.

Table 7 contains the exercises for which group C2.2 obtained the least hits. Group C2.2, which received challenges different from those delivered in C1, obtained fewer hits (10 of 16) in the composite exercise *Caesar Cipher encryption/decryption ◦ Comment in the robots.txt file*. It is believed that the greatest difficulty to solve this exercise was to identify, among the set of files received, that the flag would be in the robots.txt file, for the aforementioned reasons that make the exercises with this technique more challenging.

Table 5: Exercises with less success in C1.

Class	Problem(s)	Hits	%
TADS	*Comment in the robots.txt file ◦ Image steganography*	2	20.0
FIC	*Image steganography ◦ Image steganography*	2	28.6
BCC	*Image steganography ◦ Image steganography* *Comment in the robots.txt file ◦ Image steganography*	7	53.9
Overall	*Comment in the robots.txt file ◦ Image steganography*	12	40.0

Table 6: Exercises with less success in C2.1.

Class	Problem(s)	Hits	%
TADS	*ASCII-to-decimal encoding/decoding* *Base64 encoding/decoding ◦ Java decompilation* *Image steganography ◦ Image steganography* *Comment in the robots.txt file ◦ Image steganography*	4	80.0
FIC	*Image steganography ◦ Image steganography*	1	25.0
BCC	*Image steganography ◦ Image steganography* *Comment in the robots.txt file ◦ Image steganography*	4	66.7
Overall	*Image steganography ◦ Image steganography*	9	64.3

Table 7: Exercises with less success in C2.2.

Class	Problem(s)	Hits	%
TADS	*Cipher encryption/decryption ◦ Comment in the robots.txt file*	3	60.0
FIC	*Cipher encryption/decryption ◦ Comment in the robots.txt file*	2	50.0
BCC	*ASCII-to-decimal encoding/decoding ◦ HTML comment* *Cipher encryption/decryption ◦ Comment in the robots.txt file*	5	71.4
Overall	*Cipher encryption/decryption ◦ Comment in the robots.txt file*	10	62.5

5.3.4 Success Ratio per Technique

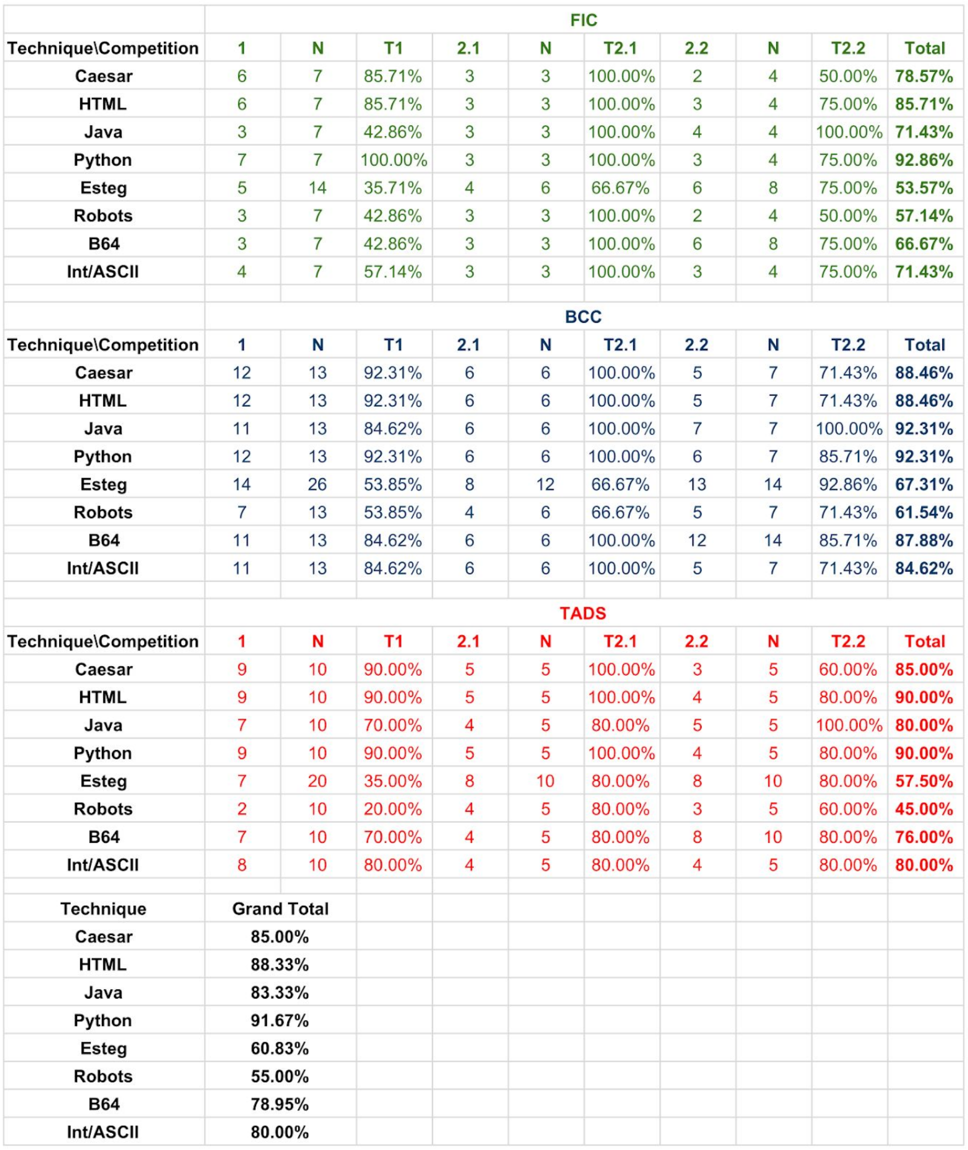
In addition to evaluating the difficulty level of the problems, isolating the techniques it is possible to identify those that imposed more and less difficulty to the players. Figure 6, taken from a spreadsheet, shows this results.

Overall, the most difficult technique was *Comment in the robots.txt file*, with 55.0% of exercises solved. In the TADS and BCC classes it was also the *Comment in the robots.txt file*, with 45.0% and 61.5% of the exercises solved. In the FIC class the most difficult technique was *Image Steganography*, with 53.6% of the exercises solved. Exercises with the *Comment in the robots.txt file* are complex because of players receive a set of files rather than a single file, as in other implemented problems. So, before looking for the flag, the player must find the file in which it is or implement a solution via script that looks for the flag pattern in all the files in the directory. The problem may be compounded, which would require looking for the flag pattern in other formats (e.g., encoded in Base64). The difficulty with the *Image Steganography* problem may again be related to the various possible tools for finding texts in images (for example, *cat* and *strings*) and attempts to use *outguess* with a password (*k* parameter).

The easiest technique overall was *Python decompilation*, with 91.7% hit rate of problems. In all classes this technique was also the easiest, with 90.0% of hits in the TADS class, 92.9% in the FIC class and 92.3% in the BCC class. In the BCC class, *Java decompilation* also reached 92.3% hits. The reason for this can be related to the number of possible solutions with the set of tools presented in the preparatory class (for example, with *strings* and *cat*) and to the individual application of the problem *Python decompilation* in the competitions C1 and C2.1, which required the application of fewer tools to obtain the flag. The original spreadsheet with all results by technique can be seen in <http://bit.ly/2E0smqW>.

5.3.5 Average Time to Complete

Figure 7 shows the average time to complete the activity among the students who solved every problem. The starting time of the competition and the time of the last correct submission for each player were considered for this. In the TADS and FIC classes, two students achieved the maximum score in Competition 1; in the BCC class, seven. Of the students with maximum scores in Competition 2, seven were in the TADS class, three in the FIC class, and eight in the BCC class.



					FIC					
Technique\Competition	**1**	**N**	**T1**	**2.1**	**N**	**T2.1**	**2.2**	**N**	**T2.2**	**Total**
Caesar	6	7	85.71%	3	3	100.00%	2	4	50.00%	**78.57%**
HTML	6	7	85.71%	3	3	100.00%	3	4	75.00%	**85.71%**
Java	3	7	42.86%	3	3	100.00%	4	4	100.00%	**71.43%**
Python	7	7	100.00%	3	3	100.00%	3	4	75.00%	**92.86%**
Esteg	5	14	35.71%	4	6	66.67%	6	8	75.00%	**53.57%**
Robots	3	7	42.86%	3	3	100.00%	2	4	50.00%	**57.14%**
B64	3	7	42.86%	3	3	100.00%	6	8	75.00%	**66.67%**
Int/ASCII	4	7	57.14%	3	3	100.00%	3	4	75.00%	**71.43%**
					BCC					
Technique\Competition	**1**	**N**	**T1**	**2.1**	**N**	**T2.1**	**2.2**	**N**	**T2.2**	**Total**
Caesar	12	13	92.31%	6	6	100.00%	5	7	71.43%	**88.46%**
HTML	12	13	92.31%	6	6	100.00%	5	7	71.43%	**88.46%**
Java	11	13	84.62%	6	6	100.00%	7	7	100.00%	**92.31%**
Python	12	13	92.31%	6	6	100.00%	6	7	85.71%	**92.31%**
Esteg	14	26	53.85%	8	12	66.67%	13	14	92.86%	**67.31%**
Robots	7	13	53.85%	4	6	66.67%	5	7	71.43%	**61.54%**
B64	11	13	84.62%	6	6	100.00%	12	14	85.71%	**87.88%**
Int/ASCII	11	13	84.62%	6	6	100.00%	5	7	71.43%	**84.62%**
					TADS					
Technique\Competition	**1**	**N**	**T1**	**2.1**	**N**	**T2.1**	**2.2**	**N**	**T2.2**	**Total**
Caesar	9	10	90.00%	5	5	100.00%	3	5	60.00%	**85.00%**
HTML	9	10	90.00%	5	5	100.00%	4	5	80.00%	**90.00%**
Java	7	10	70.00%	4	5	80.00%	5	5	100.00%	**80.00%**
Python	9	10	90.00%	5	5	100.00%	4	5	80.00%	**90.00%**
Esteg	7	20	35.00%	8	10	80.00%	8	10	80.00%	**57.50%**
Robots	2	10	20.00%	4	5	80.00%	3	5	60.00%	**45.00%**
B64	7	10	70.00%	4	5	80.00%	8	10	80.00%	**76.00%**
Int/ASCII	8	10	80.00%	4	5	80.00%	4	5	80.00%	**80.00%**
Technique	**Grand Total**									
Caesar	**85.00%**									
HTML	**88.33%**									
Java	**83.33%**									
Python	**91.67%**									
Esteg	**60.83%**									
Robots	**55.00%**									
B64	**78.95%**									
Int/ASCII	**80.00%**									

Figure 6: Results per technique.

It can be seen from Figure 7 that the average time to complete the activity decreased from C1 to C2, and that the number of students who managed to solve all exercises increased in the three classes. However, it is necessary to consider that time is affected by several factors, such as student tardiness and possible errors in tools or server. The overall average time for Competition 1 was 3383.4 s (56.4 min), and for Competition 2 it was 2706.4 s (45.1 min), a reduction of 11.3 min.

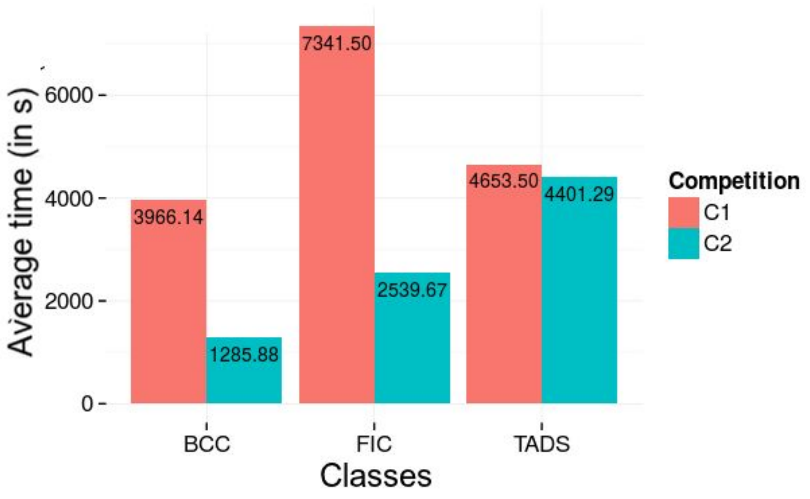


Figure 7: Average time to complete the activity in the two competitions.

The Wilcoxon test for paired samples shows that there is a significant difference between the completion times of Competitions 1 and 2 ($V = 381$, $p = 0.0016 < 0.05$). However, the Wilcoxon test for unpaired samples shows that the difference between groups C2.1 and C2.2 is not statistically significant ($W = 108$, $p = 0.89 > 0.05$). This is further indication that the exercises were easier after the application of Competition 1, but inconclusive about the time difference for the completion of C2.1 and C2.2.

5.4 Pre- and Post-Test Questionnaires Results

Pre- and post-test questionnaires helped assess the effect of the activity and measure the players' perception of satisfaction, learning, and interest in Computer Security. The pre-test questionnaire was answered by 30 students and the post-test was answered by 29.

The analysis of the questionnaires measured the students' receptivity to the competitions and how they contribute to learning and influence their professional career perspectives. We also evaluated the existence of statistically significant differences between the answers in the pre- and post-test questionnaires, which would show a change of perception based on participation in the challenges. In order to minimize the bias in the responses, we decided to leave the questionnaires anonymous (ie, without the identification of the respondents), which makes it impossible to analyze the evolution of the individuals. Thus, the response groups were considered as independent samples. For this analysis the Wilcoxon rank-sum test (for unpaired samples) was used with significance level $\alpha = 0.05$.

The aspects of satisfaction and perception of difficulty of the problems were the object of multiple questions. To measure the internal consistency of the questions in measuring each of these aspects a reliability analysis was performed using Cronbach's alpha [29].

The comparison between the questionnaires served to analyze possible changes in the results and could indicate if there was success in clarifying the idea of pedagogic competition for the student as well as increased their motivation. Table 8 shows the questionnaire results and Table 9 shows the questions and their respective identifiers.

Table 8: Questionnaires results.

Question	Attribute	BCC		TADS		FIC		Overall		Evolution	Significant?
		pre	post	pre	post	pre	post	pre	post		
1.1	Satisfaction	4.00	4.50	4.00	4.60	4.14	4.43	4.03	4.52	+0.49	Yes ($p = 0.0076$)
1.2	Satisfaction	4.23	4.64	4.20	4.40	3.86	4.29	4.13	4.48	+0.35	No ($p = 0.12$)
1.3	Satisfaction	4.54	4.58	4.60	4.50	4.14	4.57	4.47	4.55	+0.08	No ($p = 0.51$)
1.4	Learning	3.08	3.17	3.10	2.90	3.43	2.57	3.17	2.93	−0.24	No ($p = 0.43$)
1.5	Satisfaction	4.85	4.83	4.80	4.80	4.57	4.71	4.77	4.79	+0.02	No ($p = 0.81$)
1.6	Interest	2.15	3.05	2.20	2.40	2.00	2.86	2.13	2.79	+0.66	Yes ($p = 0.018$)
1.7	Satisfaction	4.08	4.50	4.00	4.60	4.00	4.14	4.03	4.45	+0.42	No ($p = 0.012$)

2.1	Interest	2.77	2.84	3.20	3.00	2.86	3.29	2.93	3.00	+0.07	No (p = 0.75)
3.1	Difficulty	–	2.85	–	3.10	–	2.71	–	2.90	–	–
3.2	Difficulty	–	3.32	–	2.90	–	3.14	–	3.14	–	–
4.1	Motivation	–	3.92	–	4.00	–	4.14	–	4.00	–	–
4.2	Motivation	–	4.33	–	4.50	–	3.71	–	4.24	–	–

Table 9: Questions shown in Table 8.

Question ID	Question
1.1	Games and competitions make me more motivated to learn than lectures.
1.2	I would like games and competitions to be explored in other courses.
1.3	I am interested in practical activities involving Computer Security.
1.4	I have difficulty in practical activities involving Computer Security.
1.5	Practical Computer Security exercises increase understanding about this area.
1.6	I feel sufficiently prepared to (begin to) participate in Computer Security competitions.
1.7	I understand that Computer Security competitions increase the appeal of this area to the general public.
2.1	The probability that I will try to pursue a career in the area of Computer Security is...
3.1	The problems in the competition were difficult to solve.
3.2	I spent a lot of time solving the problems.
4.1	Motivation from game competitiveness.
4.2	Motivation from problem composition.

In order to identify the players' satisfaction with the activity, we observed in particular questions 1.1, 1.2, 1.3, 1.5, and 1.7, both from pre- and post-test. These questions use a five-point Likert scale, with responses ranging from strongly disagree (1) to fully agree (5). Values greater than 3 indicate positive satisfaction.

Table 8 shows that the satisfaction results were above 4.00 points in the *Overall* column, indicating that the students were satisfied with the competition, on average. The *Evolution* column contains the difference between the overall means of the post- and the pre-test, and the column *Significant?* indicates whether this difference is statistically significant ($p \leq 0.05$).

We notice that all post-test satisfaction results were higher than the pre-test results. There was an evolution in absolute numbers considering the results by class, except for question 1.3 in the TADS class, for which there was a decrease of 0.10 points, and for question 1.5 in the BCC class, for which there was a decrease of 0.02 points. One must also consider that in this group there was one less response in the post-test questionnaire in relation to the pre-test.

The Wilcoxon test for non-paired samples was applied to measure the statistical significance of the differences. The differences found for questions 1.2, 1.3, and 1.5 were not statistically significant ($p > 0.05$), while the differences for questions 1.1 and 1.7 were significant. It is important to note that the means for satisfaction questions were already high (greater than 4) in the pre-test questionnaire, which limits the possibility of evolution, since the maximum possible score is 5. However, the results indicate that participation in competitions has increased the motivation associated with games as a learning tool and the perception that games can attract public attention to Computer Security.

The reliability analysis regarding satisfaction issues had a Cronbach's alpha of 0.78, which indicates substantial reliability, close to the threshold for near-perfect reliability (0.8) [30]. Only the post-challenge questionnaire responses were considered in this analysis.

Question 1.6 verified whether the student felt ready to participate in challenges, measuring interest and perception of one's own knowledge. The answers were on a five-point Likert scale (I totally disagree ... I totally agree). Overall, the students felt unprepared to participate in these competitions. The mean in the pre-test was 2.13, close to the partially disagree option on the scale. The post-test mean increased to 2.79, closer to the neutral on the scale. Although the perception is still more negative than positive, there was a statistically significant change of 0.66 points in relation to the pre-test, indicating an improvement in the perception about the preparation with Cybersecurity problems. All the results by class showed evolution as well.

Question 2.1 investigated the interest of students in a career in Computer Security, encompassing interest and professional perspective. The responses were on a five-point Likert scale ranging from Very low (1) to Very high (5). The post-test mean was 3.00 points (neutral), 0.07 above the pre-test mean, and did not represent a statistically significant difference. This means that the students do not demonstrate a predisposition favorable or against a professional career in Cybersecurity, and that participation in the challenge practically did not change that perspective.

To identify whether the level of questions was adequate, questions 3.1 and 3.2 of the post-test were used as a basis. The results, shown in Table 8, indicates that the means for these questions were close to 3 points, ie, close to neutrality. This neutrality means that respondents did not find problems particularly easy or difficult. The reliability analysis for questions 3.1 and 3.2 resulted in a Cronbach's alpha of 0.72, which indicates substantial reliability [30], that is, the answers to the two questions showed high agreement.

Questions 4.1 and 4.2 of the post-test dealt with students' motivation from game competitiveness and problem composition, respectively. The responses were on a five-point Likert scale, from Very demotivating (1) to Very motivating (5). The results show that competitiveness was considered a motivating factor in general. In the BCC class the mean was 3.92 points. The overall mean of 4.24 points for question 4.2 indicates that students considered this factor to be between motivating and very motivating. In the FIC class this result was 3.71 points. The results below 4 points may be related to the disagreement on the part of the players about the difficulty of the problems and the time spent on the solutions.

Question 1.4 has contributed to students' perception of learning. The answers were on a five-point Likert scale, from Totally Disagree (1) to Totally Agree (5). Contrary to the other questions, in this question the positive answers have scores lower than 3, indicating that the student does not have difficulty with the activities. The overall post-test mean was 2.93 points, 0.24 points below the overall pre-test mean (3.17 points). The difference is not statistically significant but it suggests that the competitions have had a moderate success in reducing the perception of difficulty with practical activities. The result is also consistent with reports of disagreement regarding problem difficulty.

5.5 Observations

During the activities with the BCC class, it was observed that students not enrolled in the course attended to follow the activity. In this class, it was also possible to verify that the competition factor was important. Students talking about what they could have done to obtain better results at the end of the first competition evidence this.

In the TADS and FIC classes there was no mention of the score and the competition. Students in the TADS class solved the problems in different ways. For instance, the *ASCII-to-decimal encoding/decoding* problem was solved using online tools, by hand, and using JavaScript code written on the spot. The *Python decompilation* problem was solved with Unix command line tools (*strings* and *cat*) and using an online decompilation tool. There were also some instances of relaxation during the competitions, such as audible laughter after finding a flag or solving a problem that turned out to be a decoy (i.e., did not lead to the correct flag) in the *HTML comment* problem.

The learning benefits of the activity could be confirmed in written tests that included questions about techniques seen in the competition, and were applied in the FIC and BCC classes as part of their regular assessment procedures. All students in the FIC class answered correctly, while 88.5% of the students in the BCC class got correct answers.

5.6 Discussion of Results

In general, it was noticed that the students were interested in the area of Computer Security. More exercises solved in Competition 2 and in a shorter time, as well as the evolution of students' perception about the preparation to solve Cybersecurity problems and their complexity, were indicative of student learning. Based on the results, their perception about motivation to use games in class was positive.

Performance and questionnaires results indicate that the exercises became easier after the application of Competition 1. The number of selected techniques (eight) is a factor that may have contributed to this. In addition, the same techniques of Competition 1 were used in Competition 2, but with different compositions for group C2.2.

Scores increased from C1 to C2 for both simple and compound problems, with statistically significant differences. In addition, the success ratio for simple problems was greater than for compound problems, which was expected. Simple problems usually require the use of a single tool to obtain the solution, and are less complex than compound ones. Although compositions were limited to two techniques, the application of one more technique brings greater complexity to solve a problem.

The number of players with high scores inflated the measures of central tendency, and resulted in non-significant differences in performance. Thus, some results were inconclusive, such as the difference between the performance of students in groups C2.1 and C2.2.

There is an apparent contradiction between students' performance and their answers to questions 3.1 and 3.2. While most students achieved maximum scores in Competition 2 (which suggests the problems were reasonably easy), the answers about the difficulty of the problems showed that the students did not found problems easy or difficult in general, with result close to the neutral in this aspect.

The automated generation of Cybersecurity challenges was positively evaluated. However, it is necessary to consider that the evaluated classes were small, limiting the sample size and affecting the results, and it is not possible to identify differences in some cases. We believe that the effectiveness of problem randomization can be better quantified with challenges containing more problems and with less repetition of techniques. Measures that may be taken in this regard include increasing the number of techniques and/or the number of options for each technique and increasing the number of possible compositions for each problem. In addition, applying competition to larger classes can yield more consistent results.

6 Conclusion

Teaching Cybersecurity is still a challenge. Despite the importance of the subject and its contemporary character, its inherent dynamism requires constant updating of course contents in order to achieve continuing relevance. It is also a fact that new pedagogical practices are needed to ensure successful outcomes for the students. The adoption of games and competitions are practices that contribute to this.

This work advocates the use of treasure hunting competitions to teach Computer Security, and proposes the use of randomization to automatically generate diverse instances of this type of competition, thereby circumventing the main difficulties associated with it. Regarding the state of the art, the main contributions of the work are the possibility of automatically generating entire competitions, not just individual problems, and the more extensive use of the concept of composition of techniques.

Competitions produced with a prototype of the problem generator were carried out in two institutions and in three groups. The effect of competition was assessed through questionnaires and performance in the activity. The effectiveness of the tool in generating distinct challenges was measured by comparing groups of students who received problems with different compositions, but no statistically significant difference was found between groups. Although the results so far are encouraging, the effectiveness of randomization still needs to be explored further. Regarding the perception of satisfaction, learning and interest, evaluated through questionnaires, the results provided indications that the activity was well received by the students.

In further work, we intend to hold competitions with larger groups of students, to add new techniques to the problem generator, and to allow for the composition of more than two techniques. We also intend to add identifiers to the competitions, which will allow to observe the evolution of players over time and isolate players from different classes more easily for data analysis purposes.

Acknowledgements

The authors would like to thank the Catarinense Instituto Federal and the State University of Santa Catarina, which made possible to carry out this work.

References

[1] S. Furnell and N. Clarke, “Power to the people? the evolving recognition of human aspects of security,” Computers & Security, Elsevier, v. 31, n. 8, p. 983–988, 2012.

[2] G. Dhillon, R. Syed, and C. Pedron, “Interpreting information security culture: an organizational transformation case study,” Computers & Security, Elsevier, v. 56, p. 63–69, 2016.

[3] R. S. Cheung et al., “Challenge based learning in cybersecurity education,” in Proceedings of the 2011 International Conference on Security & Management, vol. 1, 2011.

[4] M. Suby and F. Dickson, “The 2015 (ISC) 2 global information security workforce study,” Frost & Sullivan in partnership with Booz Allen Hamilton for ISC2, 2015.

[5] MEC, “Resolução CNE/CES 5/2016,” Institui as Diretrizes Curriculares Nacionais para os cursos de graduação na área da Computação, abrangendo os cursos de bacharelado em Ciência da Computação, em Sistemas de Informação, em Engenharia de Computação, em Engenharia de Software e de licenciatura em Computação, e dá outras providências. Diário Oficial da União, Brasília, 17/11/2016, Seção 1, p. 22–24, 2016.

[6] ACM, "Information technology curricula 2017," 2017, Curriculum Guidelines for Undergraduate Degree Programs in Information Technology - A Report in the Computing Curricula Series. Task Group on Information Technology Curricula. Association for Computing Machinery (ACM). IEEE Computer Society (IEEE-CS), 2017 July 27.

[7] J. Mirkovic and P. Peterson, "Class capture-the-flag exercises," in 3GSE, 2014.

[8] G. Vigna et al., "Ten years of iCTF: The good, the bad, and the ugly", in 3GSE, 2014.

[9] J. Burket et al., "Automatic problem generation for capture-the-flag competitions," in 3GSE, 2015.

[10] W. Feng, "A scaffolded, metamorphic CTF for reverse engineering", in 3GSE, 2015.

[11] G. B. White and R. Dodge, "The national collegiate cyber defense competition," in Proceedings of the Tenth Colloquium for Information Systems Security Education, 2006.

[12] G. Vigna, "Teaching network security through live exercises," in Security education and critical infrastructures. Springer, 2003. [Online]. Available: https://doi.org/10.1007/978-0-387-35694-5_2.

[13] R. Weiss, J. Mache, and E. Nilsen, "Top 10 hands-on cybersecurity exercises," Journal of Computing Sciences in Colleges, vol. 29, 2013.

[14] W. Petullo et al., "The use of cyber-defense exercises in undergraduate computing education." in ASE'16, 2016.

[15] R. S. Cheung et al., "Effectiveness of cybersecurity competitions", in Proceedings of the International Conference on Security and Management (SAM), 2012.

[16] A. Conklin, "Cyber defense competitions and information security education: An active learning solution for a capstone course," in HICSS'06. IEEE, 2006. [Online]. Available: https://doi.org/10.1109/HICSS.2006.110.

[17] M. Guimaraes, H. Said, and R. Austin, "Using video games to teach security," in Proceedings of the 16th annual joint conference on Innovation and technology in computer science education (ITiCSE), ACM, p. 346–346, 2011. [Online]. Available: https://doi.org/10.1145/1999747.1999860.

[18] M. Olano et al., "SecurityEmpire: Development and evaluation of a digital game to promote cybersecurity education." in 3GSE, 2014.

[19] T. Denning, A. Lerner, A. Shostack, and T. Kohno, "Control-Alt-Hack: The design and evaluation of a card game for computer security awareness and education," in ACM CCS, 2013. [Online]. Available: https://doi.org/10.1145/2508859.2516753.

[20] B. I. Gibson, "Educational games for teaching computer science", Master's thesis, Department of Computer Science and Software Engineering, University of Canterbury, New Zealand, 2013.

[21] E. Capuano, "#jolt hackathon 2017," 2017. [Online]. Available: https://blog.ecapuano.com/jolthackathon-2017/.

[22] T. Chothia and C. Novakovic, "An offline capture the flag-style virtual machine and an assessment of its value for cybersecurity education," 3GSE, 2015.

[23] P. Chapman, J. Burket, and D. Brumley, "PicoCTF: A game-based computer security competition for high school students." in 3GSE, 2014.

[24] T. H. Lacey, G. L. Peterson, and R. F. Mills, "The enhancement of graduate digital forensics education via the dc3 digital forensics challenge," in HICSS'09. IEEE, 2009. [Online]. Available: https://doi.org/10.1109/HICSS.2009.433.

[25] C. Taylor et al., "CTF: State-of-the-art and building the next generation," in ASE'17, 2017.

[26] Z. Schreuders et al., "Security scenario generator (SecGen): A framework for generating randomly vulnerable rich-scenario VMs for learning computer security and hosting CTF events," in ASE'17, 2017.

[27] C. Eagle, "Computer security competitions: Expanding educational outcomes," IEEE Security & Privacy, vol. 11, no. 4, pp. 69–71, 2013. [Online]. Available: http://doi.org/10.1109/MSP.2013.83.

[28] “CTF write-ups,” 2017. [Online]. Available: https://github.com/ctfs.

[29] A. Field, J. Miles, and Z. Field, Discovering Statistics Using R. SAGE Publications, 2012.

[30] J. R. Landis and G. G. Koch, “The measurement of observer agreement for categorical data,” Biometrics, vol. 33, no. 1, p. 159–174, 1977. [Online]. Available: http://doi.org/10.2307/2529310.