

ANO
2018



UNIVERSIDADE DO ESTADO DE SANTA CATARINA - UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT
PROGRAMA DE PÓS GRADUAÇÃO EM COMPUTAÇÃO
APLICADA

RICARDO DE LA ROCHA LADEIRA | TREASUREHUNT: GERAÇÃO
AUTOMÁTICA DE DESAFIOS APLICADOS NO ENSINO DE SEGURANÇA COMPUTACIONAL

Segurança Computacional é uma área cada vez mais importante. A necessidade de proteção das informações contrasta com a falta de profissionais e o pouco espaço dedicado à área em cursos de Tecnologia da Informação. Jogos e competições vêm sendo usados para motivar alunos de Computação a aprofundarem seus conhecimentos práticos sobre o tema e despertar o interesse por Segurança. Este trabalho propõe o uso de aleatorização para gerar problemas e competições inteiras de forma automatizada, obtendo instâncias exclusivas de problemas, segundo parâmetros definidos pelo organizador da competição.

Orientador: Rafael Rodrigues Obelheiro

Joinville, 2018

DISSERTAÇÃO DE MESTRADO

**TREASUREHUNT:
GERAÇÃO AUTOMÁTICA
DE DESAFIOS APLICADOS
NO ENSINO DE
SEGURANÇA
COMPUTACIONAL**

RICARDO DE LA ROCHA LADEIRA

JOINVILLE, 2018

UNIVERSIDADE DO ESTADO DE SANTA CATARINA - UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT
MESTRADO EM COMPUTAÇÃO APLICADA

RICARDO DE LA ROCHA LADEIRA

TREASUREHUNT: GERAÇÃO AUTOMÁTICA DE DESAFIOS
APLICADOS NO ENSINO DE SEGURANÇA COMPUTACIONAL

JOINVILLE

2018

RICARDO DE LA ROCHA LADEIRA

**TREASUREHUNT: GERAÇÃO AUTOMÁTICA DE DESAFIOS
APLICADOS NO ENSINO DE SEGURANÇA COMPUTACIONAL**

Dissertação submetida ao Programa de Pós-Graduação em Computação Aplicada do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, para a obtenção do grau de Mestre em Computação Aplicada.

Orientador: Prof. Dr. Rafael Rodrigues Obelheiro

JOINVILLE

2018

de la Rocha Ladeira, Ricardo
TreasureHunt: Geração Automática de Desafios
Aplicados no Ensino de Segurança Computacional /
Ricardo de la Rocha Ladeira. - Joinville , 2018.
119 p.

Orientador: Rafael Rodrigues Obelheiro
Dissertação (Mestrado) - Universidade do Estado de
Santa Catarina, Centro de Ciências Tecnológicas,
Programa de Pós-Graduação em Computação Aplicada,
Joinville, 2018.

1. Geração Automática de Problemas. 2. Segurança
Computacional. 3. Ensino. I. Rodrigues Obelheiro,
Rafael. II. Universidade do Estado de Santa
Catarina. Programa de Pós-Graduação. III. Título.

**TreasureHunt: Geração de Desafios Automáticos Aplicados no Ensino de
Segurança Computacional**

por

Ricardo de la Rocha Ladeira

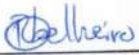
Esta dissertação foi julgada adequada para obtenção do título de

Mestre em Computação Aplicada

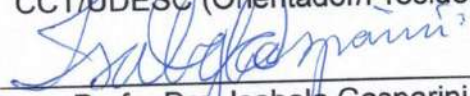
Área de concentração em "Ciência da Computação",
e aprovada em sua forma final pelo

**CURSO DE MESTRADO ACADÊMICO EM COMPUTAÇÃO APLICADA
DO CENTRO DE CIÊNCIAS TECNOLÓGICAS DA
UNIVERSIDADE DO ESTADO DE SANTA CATARINA.**

Banca Examinadora:



Prof. Dr. Rafael Rodrigues Obelheiro
CCT/UDESC (Orientador/Presidente)



Profa. Dra. Isabela Gasparini
CCT/UDESC



Profa. Dra. Michelle Silva Wingham
UNIVALI

Joinville, SC, 23 de abril de 2018.

AGRADECIMENTOS

Agradeço ao Instituto Federal Catarinense, por meio do Programa Institucional de Qualificação de servidores para o Instituto Federal Catarinense (PIQIFC), e à Universidade do Estado de Santa Catarina, que tornaram possível a realização deste trabalho.

RESUMO

Segurança Computacional é uma área cada vez mais importante, considerando o constante crescimento e a sofisticação das ameaças presentes no mundo digital. A necessidade de proteção das informações contrasta com a falta de profissionais e o pouco espaço dedicado à área em cursos de Tecnologia da Informação. Jogos e competições vêm sendo usados para motivar alunos de Computação a aprofundarem seus conhecimentos práticos sobre o tema e para despertar o interesse de potenciais estudantes e profissionais por Segurança. A elaboração desses jogos requer conhecimento especializado, muitas vezes escasso. Além disso, o ineditismo dos problemas geralmente é vital para atingir o nível desejado de dificuldade e assim garantir a competitividade. Este trabalho propõe o uso de aleatorização para gerar problemas e competições inteiras de forma automatizada, obtendo instâncias exclusivas de problemas, segundo parâmetros definidos pelo organizador da competição. Para avaliar a proposta, uma ferramenta para criar desafios foi implementada como prova de conceito, e foram promovidas competições com problemas gerados automaticamente, nas quais tomaram parte alunos de cursos de nível superior e de qualificação profissional em Computação em duas instituições. Foram analisados o desempenho nas competições e a percepção de satisfação por parte dos alunos envolvidos. Os resultados evidenciam que a geração automática de desafios é viável, e que a competição aplicada no ensino de Segurança Computacional foi motivadora para fins didáticos.

Palavras-chaves: Geração Automática de Problemas. Segurança Computacional. Ensino.

ABSTRACT

Computer Security is an increasingly important area, given the constant growth and sophistication of threats 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 degrees. Games and competitions have been used to motivate students of Computing 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 scarce specialized knowledge to develop new problems, since the novelty of these is usually vital to reach the desired level of difficulty and thus to ensure competitiveness. This work proposes the use of randomization to generate problems and entire competitions in an automated fashion, obtaining exclusive instances of problems according to parameters defined by the organizer of the competition. In order to evaluate the proposal, a tool for creating security challenges was implemented as proof of concept, and competitions with automatically generated problems were promoted, in which students from higher education and professional qualification courses in Computing took part in two institutions. The performance in the competitions and the perception of satisfaction by the students involved were analyzed. The results show that the automatic generation of challenges is feasible, and the competition applied to the teaching Computer Security was motivating for didactic purposes.

Keywords: Automatic Problem Generation. Computer Security. Teaching.

LISTA DE ILUSTRAÇÕES

Figura 1 – Incidentes de segurança reportados ao CERT.br.	21
Figura 2 – (a) Jogo de <i>videogame</i> CyberCIEGE. (b) Jogo de <i>videogame</i> Agent Surefire.	33
Figura 3 – (a) Carta Denial of Service de valor 6 do jogo EoP. (b) Carta Elevation of Privilege de valor 8 do jogo EoP. (c) Carta Denial of Service de valor 3 do jogo EoP.	35
Figura 4 – Control-Alt-Hack™. Jogo de cartas sobre Segurança Computacional.	36
Figura 5 – Duas instâncias do problema “Comentário em código-fonte de página HTML” e seus respectivos códigos-fonte.	52
Figura 6 – Solução alternativa de quatro instâncias do problema “Descompilar binário e obter fonte Java”.	53
Figura 7 – Primeira instância do problema composto (<i>Esteganografia em imagens</i> ◦ <i>(De)codificação de arquivo em base64</i>).	55
Figura 8 – Segunda instância do problema composto (<i>Esteganografia em imagens</i> ◦ <i>(De)codificação de arquivo em base64</i>).	55
Figura 9 – Diagrama de Atividades para Geração de Competição.	56
Figura 10 – Casos de Uso da aplicação <i>web</i>	58
Figura 11 – Placar individual detalhado do TreasureHunt.	58
Figura 12 – Execução do <i>script</i> <code>jogo.sh</code>	61
Figura 13 – Tela inicial do sistema <i>web</i>	62
Figura 14 – Tela de submissão de <i>flag</i> do sistema <i>web</i>	62
Figura 15 – Tela de ajuda (“Como Jogar?”) do sistema <i>web</i>	63
Figura 16 – Placar geral de uma competição no sistema <i>web</i>	64
Figura 17 – Perfil geral dos jogadores.	73
Figura 18 – <i>Boxplots</i> de número de acertos nas competições 1 e 2.	74
Figura 19 – Comparação de acertos entre os grupos C2.1 e C2.2 em todas as turmas.	75
Figura 20 – Taxa de submissões corretas (em %) das três turmas nas duas competições.	76
Figura 21 – Tempo médio (em s) para conclusão da atividade nas três turmas nas duas competições.	77

LISTA DE TABELAS

Tabela 1 – Comparativo entre os trabalhos relacionados e a ferramenta proposta.	44
Tabela 2 – Matriz de composições.	54
Tabela 3 – Consistência interna do questionário segundo o valor de alfa.	71
Tabela 4 – Técnicas estatísticas usadas na avaliação dos resultados.	71
Tabela 5 – Exercícios com mais acertos na C1.	78
Tabela 6 – Exercícios com mais acertos na C2.1.	78
Tabela 7 – Exercícios com mais acertos na C2.2.	79
Tabela 8 – Exercícios com menos acertos na C1.	79
Tabela 9 – Exercícios com menos acertos na C2.1.	80
Tabela 10 – Exercícios com menos acertos na C2.2.	81
Tabela 11 – Taxa de acertos por problemas simples.	81
Tabela 12 – Taxa de acertos por problemas compostos.	81
Tabela 13 – Resultados das questões sobre satisfação.	84
Tabela 14 – Questões analisadas na Tabela 13.	84
Tabela 15 – Resultados da questão 1.6.	85
Tabela 16 – Resultados da questão 2.1.	85
Tabela 17 – Resultados sobre o nível das questões da competição.	85
Tabela 18 – Resultados sobre motivação com competitividade e composição de problemas.	86
Tabela 19 – Resultados da questão 1.4.	86

LISTA DE ABREVIATURAS E SIGLAS

2D	<i>Duas Dimensões</i>
ACM	<i>Association for Computing Machinery</i>
APG	<i>Automatic Problem Generation</i>
ASCII	<i>American Standard Code for Information Interchange</i>
CCDC	<i>Collegiate Cyber Defense Competition</i>
CDX	<i>Cyber-Defense Exercise</i>
CEPSH/UDESC	<i>Comitê de Ética em Pesquisas Envolvendo Seres Humanos da Universidade do Estado de Santa Catarina</i>
CIO	<i>Chief Information Officer</i>
CTF	<i>Capture the flag</i>
DHS	<i>Department of Homeland Security</i>
DNS	<i>Domain Name System</i>
DoD	<i>Department of Defense</i>
EoP	<i>Elevation of Privilege</i>
EUA	<i>Estados Unidos da América</i>
FBI	<i>Federal Bureau of Investigation</i>
HTML	<i>HyperText Markup Language</i>
ID	<i>Identificador</i>
IEEE-CS	<i>Institute of Electrical and Electronics Engineers Computer Society</i>
IFC	<i>Instituto Federal Catarinense</i>
(ISC) ²	<i>International Information Systems Security Certifications Consortium</i>
JPEG	<i>Joint Photographic Experts Group</i>
MEC	<i>Ministério da Educação</i>
SBC	<i>Sociedade Brasileira de Computação</i>
SDL	<i>Security Development Lifecycle</i>
SGBD	<i>Sistema de Gerenciamento de Banco de Dados</i>
SHA256	<i>Secure Hash Algorithm 256 bits</i>

SQL	<i>Structured Query Language</i>
STRIDE	<i>Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service e Elevation of Privilege</i>
TCLE	<i>Termo de Consentimento Livre e Esclarecido</i>
TI	<i>Tecnologia da Informação</i>
UDESC	<i>Universidade do Estado de Santa Catarina</i>

LISTA DE SÍMBOLOS

$f \circ f$	Composição
%	Porcentagem

SUMÁRIO

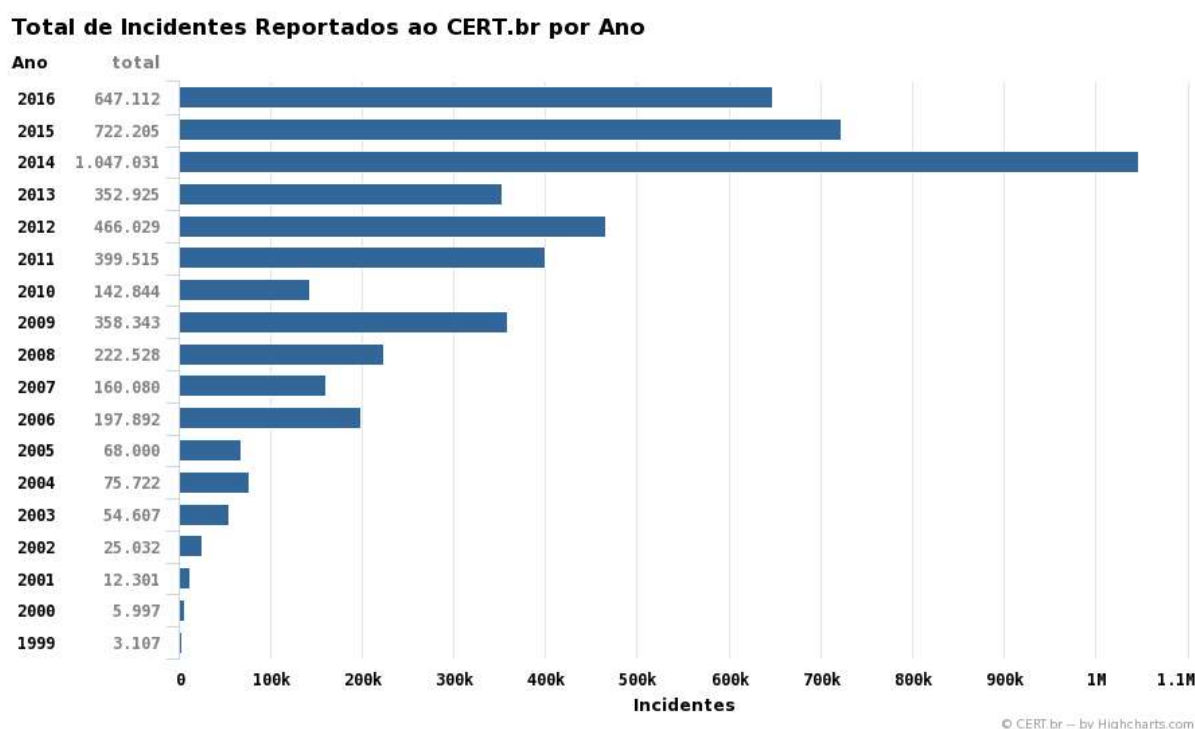
1	INTRODUÇÃO	21
1.1	Problema de Pesquisa	25
1.2	Solução Proposta	26
1.3	Objetivos	26
1.3.1	Objetivo Geral	26
1.3.2	Objetivos Específicos	26
1.4	Escopo	27
1.5	Método de Pesquisa	27
1.6	Organização do Texto	29
2	JOGOS PARA SEGURANÇA	31
2.1	Introdução ao Uso de Jogos no Ensino de Segurança	31
2.2	Tipos de Jogos para o Ensino de Segurança	32
2.2.1	<i>Videogames</i>	32
2.2.2	Jogos de Tabuleiro e Cartas	34
2.2.3	Caça ao Tesouro	36
2.2.4	Ataque e Defesa	38
2.3	Considerações sobre Jogos para Segurança	39
2.4	Trabalhos Relacionados	41
2.5	Considerações do Capítulo	45
3	GERAÇÃO AUTOMATIZADA DE DESAFIOS DE SEGURANÇA	47
3.1	Parâmetros Gerais do Desafio	47
3.2	Seleção de Técnicas	48
3.2.1	Análise de Desafios Existentes	48
3.2.2	Técnicas Selecionadas	50
3.3	Protótipo de Implementação	54
3.3.1	Visão Geral	54
3.3.2	Funcionamento do Gerador de Desafios	59
3.3.3	Funcionamento da Aplicação Web	61
3.4	Considerações do Capítulo	64
4	AVALIAÇÃO	65
4.1	Projeto de Experimento	65
4.2	Execução da Atividade	68

4.3	Análise de Resultados	69
4.3.1	Planejamento da Análise Estatística	70
4.3.2	Resultados do Questionário de Levantamento de Perfil	71
4.3.3	Resultados de Desempenho	72
4.3.3.1	Número de Acertos	72
4.3.3.2	Taxa de Submissões Corretas	74
4.3.3.3	Tempo Médio de Conclusão	76
4.3.3.4	Aproveitamento por Problema	77
4.3.3.5	Taxa de Acertos por Tipo de Problema (Simples/Composto)	80
4.3.3.6	Aproveitamento por Técnica	82
4.3.4	Resultados dos Questionários Pré e Pós-Teste	82
4.3.5	Observações	86
4.4	Discussão dos Resultados	88
4.5	Considerações do Capítulo	89
5	CONCLUSÃO	91
	REFERÊNCIAS	95
	APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLA- RECIDO	104
	APÊNDICE B – QUESTIONÁRIO DE LEVANTAMENTO DE PERFIL	107
	APÊNDICE C – QUESTIONÁRIO PRÉ-TESTE	109
	APÊNDICE D – QUESTIONÁRIO PÓS-TESTE	111
	APÊNDICE E – DICIONÁRIO DE DADOS	113
	APÊNDICE F – TABELA DE APROVEITAMENTO DE TÉCNICAS .	115
	APÊNDICE G – DEPENDÊNCIAS OPERACIONAIS	117

1 INTRODUÇÃO

Cibersegurança ou Segurança Computacional é uma área cada vez mais importante e necessária no mundo atual, considerando a onipresença tecnológica e a necessidade de proteção das informações. Esta proteção envolve não apenas tecnologias, mas também conhecimento e adoção de boas práticas de segurança por parte das pessoas que desenvolvem, gerenciam e usam os sistemas computacionais que manipulam as informações (FURNELL; CLARKE, 2012). As estatísticas do CERT.br (2017) sustentam esta necessidade no tocante aos incidentes reportados a este grupo, como pode ser observado na Figura 1. Constata-se que, nos últimos três anos nela indicados (entre 2014 e 2016), o número de incidentes somados representa pouco menos de 49% do total reportado em 18 anos, desde 1999, quando a coleta destes começou a ser realizada.

Figura 1 – Incidentes de segurança reportados ao CERT.br.



Fonte: CERT.br, 2017.

Embora dispositivos físicos e lógicos possam contribuir com a segurança de redes de computadores, a integração mútua entre ferramentas, processos e pessoas é necessária. Somente o uso de recursos tecnológicos não fornece garantia de segurança; um ambiente, para ser seguro, requer combinação de controles técnicos e humanos, bem como de outros fatores (ALHOGAIL, 2015). Para Solms (2006), isto

envolve a introdução de boas práticas e mudanças na cultura interna da organização. Astakhova (2014) complementa afirmando que a solução para problemas como uso indevido, roubos e extorsões no espaço de informação, que preocupam toda comunidade internacional, passam pela humanização da cultura de Segurança da Informação sob o ponto de vista prioritário de seus participantes, e não dos instrumentos técnicos. Para Dhillon, Syed e Pedron (2016), pesquisas têm mostrado que construir e sustentar uma boa cultura de segurança é importante em tempos de mudanças radicais. No entanto, muitas vezes a necessidade de difundir a Cibersegurança é negligenciada pelas instituições. Além disso, recursos humanos especializados em Segurança são necessários, mas o mercado carece deste tipo de profissional (CHEUNG et al., 2011).

Entende-se que as pessoas constituem um dos elementos importantes para atingir níveis aceitáveis de segurança e mitigar incidentes. Para que isto ocorra, é necessário iniciar e consolidar uma cultura de educação em Segurança, fornecendo base sobre o assunto na educação formal. Contudo, educar o usuário final é apenas uma das atividades; é importante atrair, capacitar e formar profissionais especializados, para que se tenha consciência da importância de proteger a informação, um dos principais ativos do século para as organizações (DHILLON; BACKHOUSE, 2000; DZAZALI; ZOLAIT, 2012; KOBERSY et al., 2015).

O Relatório da Segurança Digital no Brasil, referente ao terceiro trimestre de 2017, revelou que os ciberataques cresceram em 44% entre o segundo e o terceiro trimestre de 2017 (DFNDR LAB, 2017). De 115.000 dispositivos Cisco analisados na Internet, 92% executavam software com vulnerabilidades conhecidas. Em um estudo da Cisco (2016), os executivos de segurança disseram confiar menos em suas ferramentas e processos de segurança em relação ao ano anterior.

Em 2015, 59% das empresas afirmaram que sua infraestrutura de segurança estava “muito atualizada”. Em 2014, 64% disseram o mesmo. No entanto, suas preocupações crescentes com segurança os motivam a aprimorar suas formas de defesa (CISCO, 2016).

O INTERNET CRIME COMPLAINT CENTER (2016), um centro do *Federal Bureau of Investigation* – FBI voltado a queixas referentes a crimes digitais, recebeu um total de 298.728 denúncias com perdas reportadas superiores a 1,3 bilhão de dólares em 2016 (INTERNET CRIME COMPLAINT CENTER, 2016). A pesquisa da ROBERT HALF (2016), empresa de recrutamento e seleção, complementa expondo que mais de 90% dos executivos seniores de tecnologia de empresas brasileiras afirmam que enfrentarão mais ameaças nos próximos cinco anos por falta de profissionais com qualificação em Segurança da Informação. Segundo o mesmo estudo, mais de 40% dos CIOs (*Chief Information Officer*) afirmaram que expandirão as contratações

de profissionais de Segurança da Informação, mas 47% encaram com dificuldade o encontro de profissionais com o perfil desejado.

Para Roberto Portella, “não há profissionais especialistas em segurança da informação suficientes para atender à demanda atual e futura. A economia está tornando ainda mais complexa essa disciplina de TI, bem como aumentando a dependência das empresas por soluções que resguardecem os seus negócios” (ROBERT HALF, 2016).

Este desafio não é exclusivo do Brasil. No mundo inteiro estima-se que até 2020 a carência de profissionais de Segurança atinja o número de 1,5 milhão de profissionais (TODT et al., 2016). O governo norte-americano, por exemplo, sinalizou para o início de ações que promovam treinamentos em Cibersegurança através da criação de uma comissão para o reforço da Cibersegurança nacional (THE WHITE HOUSE, 2016). Esta comissão estabeleceu medidas como iniciar um programa nacional de aprendizado de Segurança Cibernética para treinar 50.000 novos profissionais e iniciar um programa nacional de força de trabalho para treinar 100.000 novos profissionais, ambos até 2020, incorporar a consciência em Segurança Cibernética em todos os níveis da educação, entre outras (TODT et al., 2016). Outra ação governamental ocorre no mês de outubro e é denominada *Mês de Conscientização em Cibersegurança*¹ (*Cyber Security Awareness Month*), promovido pelo Departamento de Segurança Interna (*Department of Homeland Security* – DHS) dos Estados Unidos da América – EUA e apoiado pela *Texas State University* e pelo Governo do Estado do Texas. Esta iniciativa destina-se a engajar e educar parceiros dos setores público e privado através de eventos e iniciativas para aumentar a conscientização sobre a importância da Segurança Cibernética, fornecendo ferramentas e recursos necessários para a manutenção da segurança *on-line* e o aumento da resiliência da nação no caso de um incidente cibernético (DEPARTMENT OF HOMELAND SECURITY, 2017).

A existência de poucos cursos formais sobre Segurança Cibernética colabora para a falta de profissionais do ramo. Muitos profissionais da área pouco estudam formalmente estes assuntos. O estudo global de Segurança da Informação (SUBY; DICKSON, 2015), da *International Information Systems Security Certifications Consortium* – (ISC)², respondido por quase 14 mil profissionais de segurança, concluiu que treinamento insuficiente em tecnologias de segurança foi uma lacuna identificada pelos entrevistados. Na mesma pesquisa, a iniciativa mais lembrada para retenção de profissionais de Segurança da Informação nas organizações foi “oferecer programas de treinamento”, com 61% dos apontamentos. Quando questionados se as organizações promovem recursos adequados para treinamentos e oportunidades de desenvolvimento profissional para suas forças-tarefa em Segurança da Informação, os profis-

¹ <<https://www.dhs.gov/national-cyber-security-awareness-month>>

sionais da área técnica responderam “não” em 45% dos casos, “sim” em 43% e “não sei” em 12%. Já os profissionais de gerência responderam “não” em 35% dos casos, “sim” em 59% e “não sei” em 6% (SUBY; DICKSON, 2015).

Existem poucas disciplinas que contêm Segurança Computacional em seus conteúdos, e, em geral, as ementas são elaboradas para atender uma das duas situações seguintes: abordar uma grande quantidade de assuntos de forma genérica e conceitual ou optar por englobar poucos tópicos, mas com aprofundamento razoável. Segundo Mirkovic e Peterson (2014), muitas aulas de Cibersegurança são ministradas de forma antiquada, usando livros didáticos e palestras, com foco em teoria e estudos de caso. Além disso, conteúdos relacionados à Segurança são vistos pela primeira vez, em geral, em cursos de graduação e pós-graduação. Acrescenta-se ainda a defasagem dos currículos, especialmente no contexto brasileiro. Os currículos de referência da SBC (Sociedade Brasileira de Computação), por exemplo, ficaram defasados por mais de dez anos. O MEC (Ministério da Educação), somente em 2016, instituiu diretrizes curriculares para os cursos de Computação, considerando Segurança um dos temas recorrentes e que devem ser considerados entre as habilidades e competências no provimento de formação profissional através da Resolução nº 5, de 16 de Novembro de 2016 (MEC, 2016). O relatório para diretrizes curriculares para cursos de TI (Tecnologia da Informação), elaborado em 2017 pela ACM (*Association for Computing Machinery*) e pela IEEE-CS (*Institute of Electrical and Electronics Engineers Computer Society*), sugere que a área de Cibersegurança deve emergir nos novos projetos curriculares (ACM, 2017).

Existe ainda outro aspecto em questão: o preconceito e a discussão ética que circundam o ensino de Cibersegurança (MIRKOVIC; PETERSON, 2014; BRATUS; SHUBINA; LOCASIO, 2010; VIGNA, 2003), já que os conhecimentos também podem ser usados de forma ilegal. Porém, para Mirkovic e Peterson (2014), os benefícios de ensinar estudantes a atacar sistemas superam as desvantagens trazidas, já que futuros profissionais de Segurança precisam estar acostumados com a prática de enfrentar ambientes adversários.

Para atrair alunos e futuros profissionais para a área de Segurança Computacional, é importante promover o ensino através de metodologias que ofereçam eficácia pedagógica² (PRASHAR, 2015; TERI et al., 2014) e incentivo ao discente.

O campo da segurança cibernética é contraditório – o verdadeiro desafio consiste em superar atacantes humanos motivados e bem informados. Infelizmente, esse aspecto está faltando nas classes atuais de cibersegurança, que muitas vezes são ensinadas através de palestras

² Clarke e Nelson (2012) definem eficácia pedagógica como uma ampla área de pesquisa que pode incluir as afecções pedagógicas e a eficácia do curso.

e ocasionalmente através de exercícios práticos “com seus pés molhados” (MIRKOVIC; PETERSON, 2014, tradução nossa).

1.1 PROBLEMA DE PESQUISA

Jogos representam uma ferramenta pedagógica motivadora para os estudantes (KAPP, 2012). Entre os elementos que contribuem para isso estão a pontuação e os formatos de recompensa, os sistemas de placar e os relacionamentos sociais envolvidos na atividade (SAILER et al., 2017). Além disso, trabalham não somente habilidades técnicas, mas também aspectos como trabalho em equipe, ambiente de pressão (VIGNA et al., 2014), liderança, e tomada de decisão. Para Vigna et al. (2014), devido à motivação extra proporcionada por um ambiente competitivo, as competições de segurança têm se tornado cada vez mais populares.

Um dos tipos mais comuns de jogos na área de Segurança Computacional é o desafio (também conhecido como *caça ao tesouro*), que consiste em encontrar palavras secretas (*flags*) ocultas em arquivos usando variadas técnicas e ferramentas computacionais. Em geral, desafios de segurança contêm vários problemas que exigem diferentes técnicas e ferramentas para resolvê-los. Esses problemas são tradicionalmente criados de forma manual, o que impõe algumas dificuldades. Primeiramente, a criação de problemas é uma tarefa trabalhosa, que exige conhecimento técnico especializado, o que limita a popularização desse tipo de jogo pela escassez de recursos humanos disponíveis. Em segundo lugar, é comum que, após a sua realização, desafios sejam publicados na Internet, não raro com as soluções para os problemas. Isso inviabiliza o reaproveitamento de questões anteriores, pois boa parte da dificuldade do jogo está em encontrar a estratégia certa para resolver cada problema. Em terceiro lugar, problemas criados manualmente costumam ter uma *flag* única, idêntica para todos os jogadores. Como as *flags* são usadas como comprovação da resolução dos problemas, essa unicidade permite que elas sejam copiadas e/ou compartilhadas entre jogadores, possibilitando que estes tenham acertos computados mesmo sem efetivamente resolver os problemas correspondentes. Por fim, é importante observar ainda que as dificuldades supracitadas reforçam-se mutuamente: à complexidade da criação de problemas somam-se a perda de valor dos problemas após sua publicação na Internet e a questão do compartilhamento de *flags*, gerando incentivos negativos para a realização de desafios de segurança.

Portanto, o problema de pesquisa abordado nesta dissertação é *como mitigar as dificuldades envolvidas na realização de desafios de segurança por intermédio de uma solução computacional*. Mais especificamente, a dissertação explora formas de permitir o reaproveitamento de problemas de modo a inviabilizar o compartilhamento de *flags* e sem que esse reaproveitamento signifique a perda do fator surpresa.

1.2 SOLUÇÃO PROPOSTA

Visando a reduzir as dificuldades identificadas e de modo a facilitar o desenvolvimento e a promoção de desafios de Segurança, esta dissertação propõe a geração e composição automatizada de problemas para esse tipo de jogo. A ideia é automatizar a geração de problemas a partir de um rol de técnicas que podem ser aplicadas individualmente ou de forma composta, e gerar competições inteiras formadas por conjuntos equivalentes de problemas, mas com *flags* únicas para cada jogador, através de uma ferramenta implementada como prova de conceito.

A hipótese do trabalho, portanto, é de que a automatização de competições de Segurança é viável e a aleatorização de problemas é eficaz em produzir desafios sem trivializar soluções. Esta hipótese será empiricamente testada através de um experimento dividido em três momentos. Realizar-se-ão duas competições de desafio de Segurança Computacional, sendo que, na primeira, todos jogadores receberão os mesmos problemas, e na segunda, um grupo receberá os mesmos problemas e outro grupo receberá problemas diferentes.

O principal resultado esperado da pesquisa é que a geração e a composição automatizada de problemas sejam viáveis e oportunizem a disseminação de desafios de segurança. Pretende-se ainda verificar se os resultados de desempenho dos grupos divididos na segunda competição serão diferentes, visando a identificar se a aleatorização dos problemas é eficaz em produzir desafios distintos e se isso se reflete no desempenho dos jogadores. De forma menos direta, deseja-se que o aspecto lúdico desses jogos possa tornar a área de Segurança Computacional mais atrativa para um universo maior de pessoas.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

O objetivo geral do trabalho é automatizar a geração de problemas para competições de Segurança, obtendo instâncias exclusivas de problemas, de forma parametrizável pelo organizador da competição.

1.3.2 Objetivos Específicos

- Realizar um estudo de identificação e análise de competições e técnicas, bem como de suas composições;
- Desenvolver uma ferramenta de automatização de desafios através de atividades que perpassam etapas de concepção, elaboração, construção e transição; e

- Avaliar uma competição de desafio no que diz respeito ao desempenho e à percepção de satisfação dos estudantes participantes da atividade.

1.4 ESCOPO

O trabalho tem como objetivo automatizar uma competição de segurança no âmbito educacional através de uma ferramenta que gera problemas compostos. A partir de um conjunto definido de atividades, avaliar-se-á a satisfação dos estudantes e obter-se-ão resultados de desempenho.

Com relação à ferramenta desenvolvida, esta contém funcionalidades de geração de problemas simples e compostos, validações de erros, configuração automática do Sistema de Gerenciamento de Banco de Dados – SGBD e envio de dados para o servidor *web*, e está detalhada no Capítulo 3. Jogadores interagirão com o servidor por meio de uma aplicação *web*, que permite visualização de regras e placar, autenticação, *download* de exercícios e submissão de respostas. A aplicação estará disponível somente em rede local, pois deverá ser acessada apenas por estudantes presentes nas aulas. A funcionalidade de criação de usuários é feita de forma automática, não podendo o jogador decidir suas credenciais. O jogo também não contempla relatórios de erros e acertos ou retornos detalhados, informando apenas se o jogador acerta ou erra a questão. A ferramenta não prevê composições de três ou mais níveis, ou seja, é responsável por aplicar composições somente de duas técnicas.

Não faz parte deste trabalho avaliar a interação do professor com a ferramenta desenvolvida. Também não há intenção de avaliar qualquer aspecto além do desempenho e da percepção de satisfação dos estudantes com a atividade proposta. Portanto, questões de facilidade de uso, percepção de utilidade e análise de interface, por exemplo, estão fora do escopo deste trabalho.

1.5 MÉTODO DE PESQUISA

Considerando a atualidade do tema de pesquisa e a dependência de atividade prática para obtenção de resultados, esta pesquisa está classificada como exploratória, tornando possível a familiarização do leitor com o tema de pesquisa. Visando a avaliar os efeitos da atividade, comparando causas e consequências, a pesquisa pode ser considerada também explicativa, pois descreve o desempenho dos jogadores e as diferenças de percepção destes antes e depois da participação na atividade. O método adotado é o hipotético-dedutivo, pois partiu-se de hipóteses para a construção de deduções posteriormente validadas no experimento realizado. Pretendia-se mostrar a viabilidade da criação de uma competição gerada de forma automática e concluir se a aleatorização de problemas seria eficaz em produzir desafios distintos sem trivializar

a solução de problemas. Para tal ação, comparou-se os resultados de desempenho na atividade, agrupando os estudantes em dois grupos distintos.

O trabalho proposto partiu do estudo da aplicação de técnicas de Segurança Computacional e de suas possíveis composições, com base em um repositório de desafios mantido pela comunidade (CTF... , 2017), em pesquisas bibliográfica e experimental. Tais pesquisas possibilitaram a modelagem de uma competição e a criação da ferramenta geradora de problemas para sua posterior aplicação em cursos da educação formal. Os dados obtidos foram analisados quantitativamente.

Dentre um conjunto de problemas de competições recentes contidas no repositório, levantamento, classificação e seleção de técnicas frequentemente utilizadas foram realizados, bem como a análise de composição entre elas, para em seguida proceder à implementação do gerador e compositor de problemas.

A construção do protótipo da ferramenta foi realizada com base na seguinte ordem:

1. Implementação dos problemas isoladamente;
2. Implementação do gerador de problemas isolados;
3. Implementação do gerador de problemas com técnicas compostas; e
4. Implementação do sistema de submissão de *flags*.

A primeira etapa corresponde à implementação dos problemas sem composição de técnicas e em *scripts* avulsos. A segunda etapa consiste em reunir todas as rotinas de criação de problemas em um *script*. A terceira etapa acrescenta a possibilidade de compor duas técnicas em um mesmo problema. A quarta etapa representa a interface *web* através da qual o usuário irá interagir na competição.

Para obtenção de resultados, o desempenho dos estudantes foi analisado através dos dados armazenados no SGBD. Nesta situação, cada estudante foi identificado por um número e sua identidade foi preservada, conforme determinado no Termo de Consentimento Livre e Esclarecido – TCLE (Apêndice A) assinado por ele. A coleta do retorno dos participantes da atividade também foi feita por uma abordagem de pesquisa quantitativa através de questionários, com respostas predefinidas e caráter objetivo e estatístico, permitindo mensurar a percepção de satisfação dos jogadores com a atividade. Todo estudante participante da atividade, obrigatoriamente, registrou aceite assinando o TCLE. Por envolver seres humanos, o projeto no qual esta dissertação se insere passou por aprovação do Comitê de Ética em Pesquisas Envolvendo Seres Humanos da Universidade do Estado de Santa Catarina – CEPESH/UDESC.

O público-alvo de aplicação é formado por estudantes de instituições de ensino, especificamente em turmas de cursos de Tecnologia da Informação na Universidade do Estado de Santa Catarina – Campus Joinville e do Instituto Federal Catarinense – Campus Blumenau, no ano de 2017. Desta forma, os sujeitos envolvidos nestas atividades são alunos, que assumem o papel de *jogadores*, e professores, que assumem o papel de *organizadores* da competição.

1.6 ORGANIZAÇÃO DO TEXTO

Este trabalho está dividido em cinco capítulos, incluindo esta introdução. O Capítulo 2 traz uma revisão sobre jogos de Segurança Computacional e discute os principais trabalhos relacionados. O Capítulo 3 descreve a proposta do trabalho, contendo a modelagem da competição e o detalhamento do desenvolvimento da ferramenta proposta. O Capítulo 4 explica a etapa de avaliação, composta pelo projeto dos experimentos realizados, a execução da atividade e a apresentação da análise dos resultados obtidos através dos dados coletados pelo sistema, de respostas em questionários e de observações. O Capítulo 5 relata as conclusões e as considerações finais do trabalho, contendo também as ideias para trabalhos futuros.

O Apêndice A contém o TCLE. Os Apêndices B, C e D trazem os modelos de questionários de levantamento de perfil de jogador, pré-teste e pós-teste, respectivamente. O Apêndice E traz o dicionário de dados do Banco de Dados utilizado no sistema. O Apêndice F contém a tabela completa de aproveitamento de técnicas, e o Apêndice G mostra as dependências operacionais para uso da ferramenta.

2 JOGOS PARA SEGURANÇA

Este capítulo apresenta uma revisão de literatura sobre jogos no contexto de Segurança Computacional. A Seção 2.1 faz uma introdução sobre o tema. Os diversos tipos de jogos usados no ensino de Segurança são descritos na Seção 2.2. A Seção 2.3 discute as considerações relacionadas a esses jogos, o que inclui pontos fortes e fracos de cada tipo de jogo. A Seção 2.4 trata dos trabalhos relacionados e a Seção 2.5 traz as considerações sobre o capítulo.

2.1 INTRODUÇÃO AO USO DE JOGOS NO ENSINO DE SEGURANÇA

Embora ainda não sejam largamente difundidas nos currículos dos cursos de TI, a inserção da Cibersegurança e das práticas não-tradicionais no ensino, tais como o *e-learning*¹ (TULARAM, 2016), *m-learning*² (KAMBOURAKIS, 2013) e sala de aula invertida³ (O'LEARY, 2017) tem ganhado espaço. Os motivos para isso podem ter relação com a falta (YASINSAC et al., 2003) e a necessidade de uma abordagem consolidada de conscientização em Cibersegurança (KRITZINGER; BADA; NURSE, 2017), embora já existam propostas de trabalhos que busquem identificar as melhores abordagens de ensino (CHUNG et al., 2014) e que proponham preencher essa lacuna com a criação de um modelo de currículo no âmbito específico da Computação Forense (PALMER et al., 2015).

Para Mirkovic e Peterson (2014), os avanços da Cibersegurança costumam ocorrer por ciclos em que os pesquisadores criam mecanismos de defesa e os criminosos elaboram ataques em resposta. No entanto, este tipo de prática está ausente do ensino da Cibersegurança (MIRKOVIC; PETERSON, 2014).

Além disso, para que seja possível trabalhar na área, é necessário utilizar ferramentas tecnológicas sofisticadas (LEGG, 2015). Compreender ferramentas de preservação, extração e análise de evidências digitais de forma apropriada (PAN et al., 2012), tais como *Volatility* (ferramenta de análise de dados carregados na memória principal) e *Wireshark* (analisador de tráfego), por exemplo, é útil para trabalhos que envolvem problemas de Computação Forense.

Para agregar conhecimento prático em disciplinas de Segurança é necessário considerar o pouco espaço dedicado ao tópico na maioria dos cursos de Computação

¹ Modalidade de educação a distância com suporte na internet (ALMEIDA, 2003).

² Novo tipo de *e-learning*, baseado em dispositivos móveis (SHARPLES, 2000).

³ Método pedagógico que utiliza aulas assíncronas em vídeo e problemas práticos como lição de casa, e atividades de resolução de problemas ativas em grupo na sala de aula (BISHOP; VERLEGER, 2013).

(WEISS et al., 2015) e a necessidade de conhecer como diferentes abstrações computacionais (tais como arquiteturas em camadas, interfaces de dispositivos e construções de linguagens de programação) são efetivamente implementadas, de modo a compreender melhor a superfície de ataque de um sistema (BRATUS, 2007).

2.2 TIPOS DE JOGOS PARA O ENSINO DE SEGURANÇA

A inserção de jogos para aumentar a consciência em Cibersegurança é uma prática que vem ganhando espaço em ambientes militares, corporativos e acadêmicos por diferentes métodos (WHITE; DODGE, 2006; BOOPATHI; SREEJITH; BITHIN, 2015) explicados na sequência da Seção, tais como:

- *videogames* (GUIMARAES; SAID; AUSTIN, 2011; OLANO et al., 2014);
- jogos de tabuleiro e cartas (DENNING; KOHNO; SHOSTACK, 2012; GIBSON, 2013);
- caça ao tesouro (VIGNA, 2003); e
- competições de ataque e defesa (WHITE; DODGE, 2006; PETULLO et al., 2016; VIGNA et al., 2014).

É importante mencionar que essa classificação não é uniforme. Chothia e Novakovic (2015), por exemplo, classificam os jogos do tipo *caça ao tesouro* como *captura da bandeira* estilo *Jeopardy!* e *ataque e defesa* e *captura da bandeira* como *captura da bandeira* estilo *ataque e defesa*. Vigna et al. (2014) estabelecem ainda outra forma de classificação em que as competições podem assumir duas formas principais: *baseadas em desafios*, sendo estas as competições sem interação, ou *interativas*, sendo estas as que as equipes ou os jogadores interagem entre eles.

As classes de jogos listadas neste trabalho estão detalhadas nas seções a seguir.

2.2.1 Videogames

Videogames são jogos em que a interação ocorre por um dispositivo de entrada e é perceptível em um dispositivo de vídeo. Os jogos de *videogame* sobre Cibersegurança costumam ter um público-alvo heterogêneo e enfatizam aspectos básicos. O CyberCIEGE⁴ (exibido na Figura 2 (a)) é um jogo de *videogame* em que jogadores operam e defendem suas redes, observando consequências diferentes com base em suas escolhas enquanto estão sob ataque. O jogo cobre aspectos significativos de

⁴ <<http://my.nps.edu/web/cisr/cyberciege>>

defesa e Segurança de Redes de Computadores em um cenário onde é necessário proteger ativos da empresa e adquirir e configurar estações de trabalho, servidores, sistemas operacionais e as próprias redes. O jogador precisa manter o equilíbrio entre segurança, produtividade e satisfação dos funcionários, operando dentro das limitações orçamentárias da empresa (IRVINE; THOMPSON; ALLEN, 2005). O jogo é pago, mas possui uma versão de avaliação gratuita, com funcionalidades limitadas. Outros jogos de *videogame* conhecidos são o Agent Surefire (MAVI INTERACTION, 2016) (pago) e o SecurityEmpire (gratuito) (OLANO et al., 2014). Todos estes envolvem uma temática em que o conhecimento de boas práticas de Segurança da Informação pode fornecer vantagens ao jogador, assim como o desconhecimento pode demonstrar más consequências advindas das escolhas do jogador, promovendo a sua conscientização. A Figura 2 exibe os cenários de dois jogos de *videogame*, o CyberCIEGE e o Agent Surefire.

Figura 2 – (a) Jogo de *videogame* CyberCIEGE. (b) Jogo de *videogame* Agent Surefire.



Fonte: IRVINE, THOMPSON; ALLEN, 2005; MAVI INTERACTION, 2016.

O PBS Cybersecurity Lab⁵ é um jogo de *videogame* em 2D (duas dimensões), gratuito, que ensina os jogadores a manterem suas vidas digitais seguras. O jogador assume o papel de diretor de uma empresa que sofre ataques cibernéticos. O objetivo é cumprir tarefas solicitadas através de cenas narradas e manter a empresa defensivamente forte perante ataques. O jogo trabalha aspectos de fraudes digitais, conceitos básicos de codificação e defesa contra ataques cibernéticos (NOVA LABS, 2018).

O Game of Threats™(PWC, 2017) é um jogo de estratégia, pago, em que o participante assume o papel de defensor dos dados de uma organização ou de atacante de um grupo criminoso que tenta comprometer os ativos da organização. O jogo é voltado a qualquer tipo de tomador de decisão de uma organização e tem como obje-

⁵ <<http://www.pbs.org/wgbh/nova/labs/lab/cyber/>>

tivo conscientizar sobre as ameaças que as organizações enfrentam e demonstrar as consequências de suas decisões. Ao final, o jogo disponibiliza um resumo detalhado com todas as ações e resultados para ambos os lados do jogo (PWC, 2017).

2.2.2 Jogos de Tabuleiro e Cartas

Jogos de tabuleiro e cartas (ou *board and card games*, em inglês) também podem introduzir conceitos de Segurança da Informação e podem ser baseados em jogos já existentes, adaptando a temática para o contexto da Cibersegurança (DENNING; KOHNO; SHOSTACK, 2012). Entre os mais conhecidos estão o *Elevation of Privilege – EoP* (MICROSOFT, 2016), o [d0x3d!]⁶ ([d0x3d!], 2016), o *Control-Alt-Hack™* (DENNING; KOHNO; SHOSTACK, 2012), o *OWASP Cornucopia* (OWASP, 2018) e o *Security Cards* (DENNING; FRIEDMAN; KOHNO, 2018).

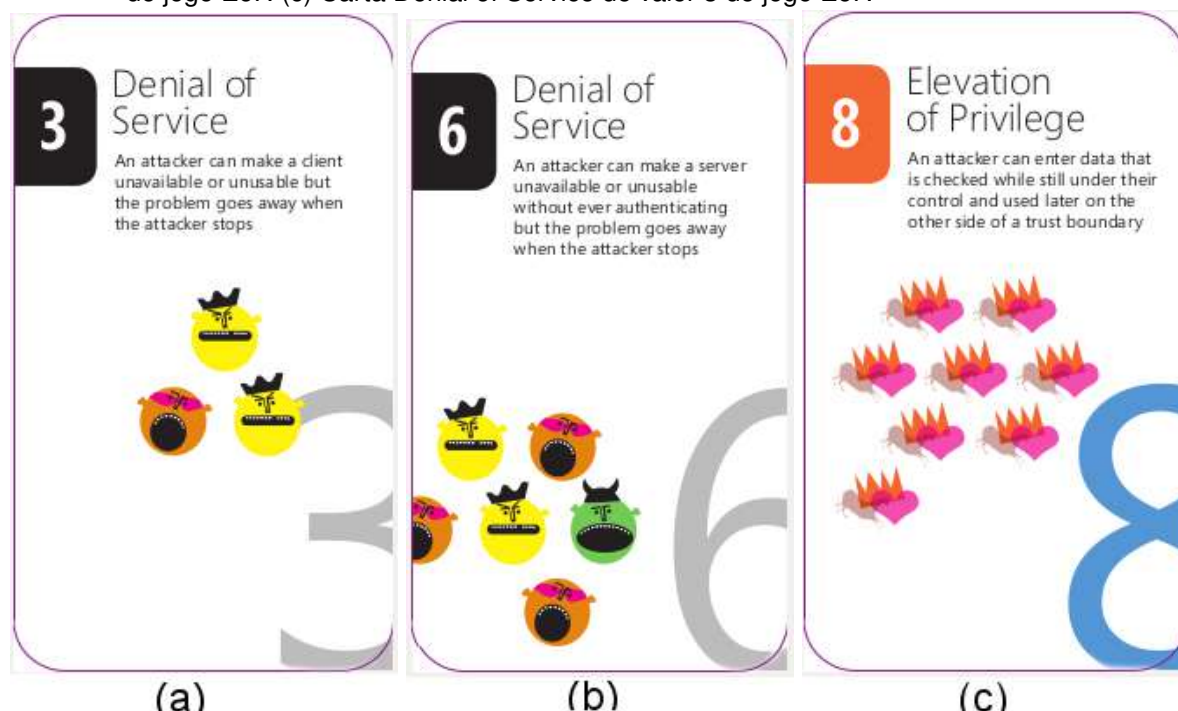
Elevation of Privilege (*Elevação de Privilégios*, em tradução livre) é um jogo de cartas criado e disponibilizado gratuitamente pela Microsoft. Ele é voltado à modelagem de ameaças (SHOSTACK, 2014), um dos componentes do processo de desenvolvimento criado pela Microsoft, denominado *Security Development Lifecycle – SDL* (MICROSOFT, 2016). O jogo foca nas ameaças STRIDE (*Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service e Elevation of Privilege*⁷). Cada carta contém um tipo de ameaça associada a um valor com base na complexidade desta ameaça. Por exemplo, uma carta *Denial of Service* de valor 6 é menos poderosa que uma carta *Elevation of Privilege* de valor 8 e mais poderosa que uma carta *Denial of Service* de valor 3. Em cada caso há uma explicação sobre o ataque, como pode ser visto na Figura 3. As cartas *Denial of Service* de pesos 3 e 6 se diferenciam na abrangência do ataque: enquanto a carta de valor 3 deixa um cliente indisponível ou inutilizável, a carta de valor 6 deixa um servidor indisponível ou inutilizável, o que representa um ataque mais poderoso. Segundo Jones (2010), a jogabilidade é simples: a cada rodada um jogador escolhe uma de suas cartas e todos os outros devem jogar uma carta que combina com ela (carta do mesmo tipo), descartar uma carta de tipo diferente ou jogar uma carta de trunfo. O vencedor será aquele que jogar o trunfo de maior valor ou, na ausência deste, a carta de mesmo tipo e de maior valor. O vencedor de cada rodada inicia a próxima, e o jogo termina quando todas as cartas forem usadas.

[d0x3d!] é um jogo de tabuleiro, gratuito, projetado para introduzir a terminologia de Segurança de Redes, os mecanismos de ataque e defesa e os conceitos básicos de Segurança Computacional para estudantes ([d0x3d!], 2016). Os jogado-

⁶ <<http://d0x3d.com/d0x3d/welcome.html>>

⁷ Em português, Falsificação de Identidade, Manipulação de dados, Repudição, Revelação de Informação, Negação de Serviço e Elevação de Privilégio.

Figura 3 – (a) Carta Denial of Service de valor 6 do jogo EoP. (b) Carta Elevation of Privilege de valor 8 do jogo EoP. (c) Carta Denial of Service de valor 3 do jogo EoP.



Fonte: MICROSOFT, 2016.

res assumem o papel de *hackers* infiltrados em uma rede. O objetivo do jogo é obter os quatro recursos digitais disponíveis, que são: credenciais de autenticação, dados financeiros, propriedade intelectual e informações pessoalmente identificáveis. Ao obter os recursos, os *hackers* precisam sair da rede sem serem descobertos ([d0x3d!], 2016). Neste jogo todos vencem ou todos perdem ([d0x3d!], 2016), não havendo competição entre os jogadores.

O Control-Alt-Hack™ é um jogo (pago) que tematiza os jogadores como *hackers de chapéu branco* (*white hat hackers*, ou *hackers éticos*) os quais trabalham para a Hackers Inc., uma empresa de Segurança de Computadores que atua na área de Auditoria de Segurança. O jogo é centrado em tarefas que exigem a aplicação de habilidades *hackers* para ter sucesso, e foi projetado para aumentar o entendimento sobre a importância da Segurança Computacional e os riscos envolvidos no armazenamento inseguro de informações (DENNING; KOHNO; SHOSTACK, 2012). A Figura 4 mostra o jogo Control-Alt-Hack™.

O OWASP Cornucopia é um jogo de cartas gratuito que serve para ajudar as equipes de desenvolvimento de software a identificar requisitos de segurança em processos de desenvolvimento ágeis, convencionais e formais. É baseado na estrutura do guia de referência da OWASP para práticas de programação segura (OWASP, 2018).

Security Cards é um jogo de cartas gratuito, criado na Universidade de Washing-

ton. O jogo é voltado para ameaças de Segurança Computacional e trabalha quatro dimensões: impacto humano, motivações de adversários, recursos de adversários e métodos de adversários, sendo um jogo focado em conceitos e recomendado para fins educacionais (DENNING; FRIEDMAN; KOHNO, 2018).

Figura 4 – Control-Alt-Hack™. Jogo de cartas sobre Segurança Computacional.



Fonte: DENNING; KOHNO; SHOSTACK, 2012.

2.2.3 Caça ao Tesouro

Competições do tipo *desafio*, também chamadas de *caça ao tesouro* (ou *treasure hunt*), consistem em conjuntos de problemas que precisam ser resolvidos com processos e ferramentas, sem interação com outros jogadores. Os desafios motivam os jogadores a encontrarem algum(ns) recurso(s) secreto(s) (*flags*) e, ao mesmo tempo, consistem em competições que podem ser facilmente reproduzidas, uma vez que exigem um número limitado de ferramentas que possibilitem sua solução. O *Wireshark*, por exemplo, é usado para resolver problemas de análise de mensagens escondidas em campos de soma de verificação (*checksum*) de cabeçalhos DNS (*Domain Name System*) (NORTHCUTT, 2016). Para problemas de esculpimento de arquivos (*file carving*), além do *Wireshark*, um editor hexadecimal, tal como o *010 Editor*, auxilia na decodificação das mensagens (MALWAREWOLF, 2015). Em desafios forenses, a descoberta de mensagens esteganografadas em imagens pode ser realizada utilizando ferramentas como o *f5* e o *outguess* (LADEIRA; OBELHEIRO, 2017).

Os problemas de competições do tipo caça ao tesouro não preveem a interação direta entre jogadores/equipes, mas promovem a competição com base em sistemas de pontuação. Por exemplo, jogadores/equipes que cumprirem primeiro uma tarefa recebem mais pontos do que aqueles que terminarem depois, seguindo uma escala decrescente (VIGNA et al., 2014), ou pelo tempo de realização da tarefa, fazendo

com que todos que cumprirem a tarefa até o tempo x recebam y pontos. Segundo Vigna (2003), estabelecer alvos fixos para todas as equipes promove a competição de uma maneira mais saudável do que ocorre em jogos nos quais equipes atacam umas às outras. Apesar disso, são flexíveis a ponto de possibilitarem também a prática individual.

Quando o desafio é formado por várias tarefas, em geral é necessário ter passado pela tarefa anterior para passar à seguinte, e assim o jogador (ou a equipe) avança até chegar à última etapa. A complexidade das etapas aumenta progressivamente, de forma que os problemas finais sejam mais difíceis de se resolver, se comparados aos iniciais (CAPUANO, 2017). Caso não exista ordem de resolução, os problemas podem ser classificados com pontuações diferenciadas, sendo aqueles mais difíceis os que recompensam com mais pontos (CHOTHIA; NOVAKOVIC, 2015).

Neste sentido, jogos podem ser classificados quanto à sua linearidade. O jogo linear é aquele em que jogador/equipe avança gradualmente à medida que resolve uma etapa do desafio, ao passo que o jogo não linear permite o avanço de etapa(s) mesmo que o jogador não a(s) solucione(m) (VYKOPAL; BARTÁK, 2016), com penalização na forma de recompensa do jogo (por exemplo, com perda de pontos). Existe ainda uma classificação intermediária com o agrupamento de problemas por fases. Neste caso, um conjunto de problemas forma uma fase do desafio. O jogador precisa resolver parte destes problemas para avançar de fase, mas não necessariamente todos, e não necessariamente em sequência (de forma não linear). Quando atinge o número mínimo de problemas para avançar de fase, o jogador pode escolher se permanece na fase e tenta resolver os problemas restantes ou avança para a fase seguinte (CHAPMAN; BURKET; BRUMLEY, 2014).

Uma variação de desafios é o formato *Jeopardy!*, que faz alusão a um homônimo programa de televisão norte-americano e reproduz o seu formato. *Jeopardy!* é um jogo de perguntas e respostas que formam um conjunto de etapas encadeadas com pontuação crescente. Segundo Eagle (2013), os organizadores publicam um número fixo de problemas com suas respectivas pontuações. O acesso aos problemas é sequencial, como ocorre no programa de televisão, ou chaveado por tempo, para impor uma progressão específica. Pode ser jogado individualmente, de forma que o ritmo seja ditado pelo jogador e sem que haja competição direta. Um jogo no estilo *Jeopardy!*, aplicado a estudantes de ensino médio e aspirantes aos cursos superiores de exatas, é o PicoCTF (CHAPMAN; BURKET; BRUMLEY, 2014).

Desafios vêm sendo aplicados há alguns anos fora do contexto de sala de aula. O Departamento de Defesa (*Department of Defense* – DoD) dos EUA promoveu entre 2006 e 2013 o DC3 *Digital Forensics Challenge* (LACEY; PETERSON; MILLS, 2009), um desafio aberto a participantes de todo o mundo. Experiências no Brasil in-

cluem a Campus Party⁸, desde 2011, o Cryptorace⁹, desde 2015, e o Hackaflag¹⁰, da Roadsec, e o Workshop de Forense Computacional do SBSeg, em 2015 e 2016. O público-alvo desses desafios, porém, não é bem definido ou controlado, incluindo alunos de graduação, de pós-graduação e até mesmo profissionais. Com isso, o nível das atividades pode ficar inadequado, equilibrando jogadores com experiências diferentes ou desmotivando os jogadores iniciantes.

2.2.4 Ataque e Defesa

Os jogos de ataque e defesa são competições elaboradas e que demandam um conhecimento técnico mais abrangente que as anteriores. Geralmente são realizadas entre equipes e exigem que estas defendam e/ou ataquem servidores e aplicações baseados em rede, ao longo de dias ou semanas. As ações de ataque e defesa envolvem instalação, atualização, configuração e manutenção de serviços, bem como rotinas de automatização e monitoramento de tarefas e serviços, o que representa um conjunto complexo de atividades. Essas competições podem ser divididas em três categorias: competições de defesa, *red versus blue* e *capture the flag*.

Competições de defesa são realizadas da seguinte maneira: experientes analistas de segurança atacam e assim testam as habilidades das equipes em proteger servidores simulando situações de risco às quais as organizações estão frequentemente expostas. Neste caso, as equipes jogadoras devem implementar ações de defesa e não podem realizar ataques. O tráfego gerado na rede poderá ser hostil ou legítimo, cabendo às equipes identificar os hostis e mitigar os efeitos advindos destes ataques. As equipes pontuam quando conseguem se defender dos ataques, ao passo que perdem pontos quando os ataques conseguem comprometer algum(ns) serviço(s). Nestas competições, as equipes recebem uma máquina virtual com diversas falhas de segurança e precisam corrigi-las, seguir boas práticas e manter os serviços disponíveis sob condições de ataque (WHITE; DODGE, 2006; PETULLO et al., 2016). As principais competições de defesa são o CDX (*Cyber-Defense Exercise*) (PETULLO et al., 2016) e a CCDC (*Collegiate Cyber Defense Competition*) (WHITE; DODGE, 2006). Estas competições costumam ter graduados em cursos de TI e estudantes de graduação e de pós-graduação como público-alvo. Questionários respondidos pelos jogadores antes e depois da participação nestas competições evidenciaram ganho nas habilidades em Cibersegurança (PETULLO et al., 2016).

As competições *red versus blue* são aquelas em que equipes podem atacar (*red team*) ou defender (*blue team*), mas não podem realizar as duas ações. As equi-

⁸ <<http://brasil.campus-party.org/>>

⁹ <<http://roadsec.com.br/cryptorace/>>

¹⁰ <<https://roadsec.com.br/hackaflag/>>

pes atacantes ganham pontos ao conseguirem explorar vulnerabilidades, ao passo que equipes de defesa ganham pontos enquanto mantêm os serviços ativos. Estas ações são monitoradas por um robô que verifica em determinados intervalos de tempo se algum serviço foi comprometido. Para Mirkovic e Peterson (2014) e Vigna (2003), as ações de ataque costumam ser mais motivadoras em relação às de defesa em jogos de Segurança.

As competições de captura da bandeira (*capture the flag* – CTF) reúnem características de defesa e ataque, ou seja, toda equipe pode (e deve) realizar as duas ações. Em geral, estes jogos recebem um público experiente em Cibersegurança (CHAPMAN; BURKET; BRUMLEY, 2014), tanto de membros de universidades quanto de empresas. O objetivo é obter sequências de símbolos escondidas em arquivos localizados nas máquinas das equipes adversárias. A estas sequências de símbolos dá-se o nome de *flags* ou bandeiras. Para capturar uma *flag* é necessário explorar vulnerabilidades existentes em serviços dos adversários, tais como servidores *web*. Ao mesmo tempo, além das ações de ataque, a equipe deve proteger os serviços que executa para impedir que os adversários obtenham suas *flags*, identificando e corrigindo falhas sem parar os serviços em execução (VIGNA et al., 2014). Nestes exercícios a pontuação é dada tanto por comprometer serviços adversários quanto por proteger os próprios serviços. Exemplos de competições de CTF tradicionais são o MIT CTF¹¹, o DEFCON CTF (COWAN et al., 2003) e o iCTF (VIGNA et al., 2014).

É importante destacar que não há consenso quanto ao uso da terminologia *capture the flag*. Existem autores que tratam competições do tipo caça ao tesouro como sinônimo de competições de captura da bandeira, ou como um subtipo de CTF, dito CTF em *Challenge Mode* (TAYLOR et al., 2017).

2.3 CONSIDERAÇÕES SOBRE JOGOS PARA SEGURANÇA

Relatos sobre competições indicam a exigência de recursos como laboratórios, computadores, *firewalls*, máquinas virtuais etc, e pessoas, necessárias para administrar as competições (TAYLOR et al., 2017). Estas exigências podem ser uma dificuldade à medida que a competição cresce em número de participantes (VIGNA et al., 2014). Schreuders et al. (2017) complementa afirmando que um dos principais obstáculos para integrar o ensino de Segurança Computacional no currículo é a quantidade de trabalho necessário para criar novos exercícios práticos. Entre outros fatores, o instrumento deve ser de fácil uso, o que inclui acesso, criação (não requerendo configuração de máquinas virtuais manualmente), modificação e compartilhamento (WEISS et al., 2015). Propostas como o DETER (DETERLAB, 2017), o ISEAGE (RURSCH; JACOBSON, 2013), o SEED LABS (DU, 2018) e o ICS *testbed*

¹¹ <<http://ctf.mit.edu/>>

(CANDELL; ZIMMERMAN; STOUFFER, 2015) possibilitam abstrair parte desta complexidade, fornecendo ambientes para execução de experimentos em Cibersegurança. O DeterLab, por exemplo, permite realizar experimentos sobre análise comportamental e tecnologias defensivas, incluindo ataques de negação de serviço, *malware*, criptografia, detecção de padrões e protocolos de armazenamento tolerantes a intrusões, e atualmente conta com 691 computadores disponíveis para uso nos testes (DETERLAB, 2017).

Para exercícios práticos e competições são necessários instrutores e técnicos capacitados em configurar serviços e redes de computadores, bem como isolar os tráfegos das competições, pois tipicamente estas são realizadas em universidades, onde há tráfego legítimo de informações a todo momento (VIGNA, 2003). Ainda segundo Vigna (2003), é frequente também a necessidade de ter acesso de administrador aos sistemas operacionais para alterar configurações de ferramentas, reconfigurar serviços e instalar *patches*.

No que diz respeito à aprendizagem, Vigna et al. (2014) afirmam que não é nas competições que a maioria das habilidades são adquiridas, mas sim no período de preparação que antecede a competição; a aprendizagem é promovida por meio de treinamentos, estudos e troca de experiências. Há indícios de incremento nas habilidades técnicas dos estudantes que participaram de jogos descritos em diversos trabalhos (WEISS; MACHE; NILSEN, 2013; RAMAN; LAL; ACHUTHAN, 2014; PETULLO et al., 2016; CHEUNG et al., 2011; CHEUNG et al., 2012), mas a eficácia pedagógica destas atividades perpassa ainda outros aspectos, tais como liderança, trabalho em equipe (para jogos não individuais), tomada de decisão e controle de tempo (CONKLIN, 2006).

Com relação aos tipos de jogos apresentados na Seção 2.2, os *videogames* e jogos de tabuleiro e cartas estão geralmente mais preocupados em conscientizar os jogadores por meio de conceitos e estratégias do que com as tecnologias correntes. Exemplos disso são o CyberCIEGE e o Control-Alt-Hack™. Os jogos de ataque e defesa exigem conhecimentos práticos mais aprofundados, tais como desenvolver código para explorar vulnerabilidades e corrigir serviços vulneráveis (VIGNA et al., 2014). Os desafios exigem conhecimentos práticos, mas com menor profundidade do que ataque e defesa, e são, de certo modo, mais flexíveis na medida em que podem envolver problemas com os mais variados níveis de complexidade (de comentário no código-fonte de uma página HTML (*HyperText Markup Language*) à análise forense de *dumps* de memória e engenharia reversa de código binário). Desta forma, competições do tipo caça ao tesouro podem ter problemas com variados graus de dificuldade, propiciando competitividade e desafio aos jogadores mais preparados, ao mesmo tempo em que possibilitam que os iniciantes façam algum progresso e assim

mantenham-se motivados.

O ineditismo é um fator vantajoso e motivador presente nestes jogos. Os passos necessários para concluir a atividade não são conhecidos, seja ela de um jogo de *videogame* ou de um desafio forense, e esta surpresa pode tornar o jogo divertido. De certa forma, isto traz uma dificuldade a mais aos criadores, já que muitas vezes as soluções podem ser encontradas após a aplicação dos jogos, tornando a atividade praticamente descartável por perder o "fator surpresa". Este problema é chamado de *compartilhamento de desafios* (TAYLOR et al., 2017).

Um bom desafio de CTF de alto valor costuma apresentar algo único: uma vulnerabilidade rara, uma combinação surpreendente de vulnerabilidades que precisam ser exploradas em sequência ou uma nova reviravolta em um problema clássico. Em muitos desses casos, o desafio é descobrir a localização dos pontos fracos no sistema e as etapas necessárias para explorar essas deficiências (BURKET et al., 2015, tradução nossa).

Além disso, outra situação indesejável pode acontecer durante a aplicação do jogo: a cópia (autorizada ou não) de *flags* encontradas por outras equipes, um problema conhecido como *compartilhamento de flags*. Esse problema ocorre porque nem todos os participantes de um jogo de caça ao tesouro têm interesse em ganhar a competição, mas, por vezes, apenas em avançar etapas. Quando a pontuação atribuída às soluções decresce à medida que o tempo passa, por exemplo, equipes mais adiantadas podem não se importar em compartilhar suas *flags* com equipes que estão mais atrasadas, especialmente se forem da mesma escola (BURKET et al., 2015).

2.4 TRABALHOS RELACIONADOS

A Seção 2.2 apresentou diversos jogos, de tipos variados, voltados para o desenvolvimento de habilidades em Segurança Computacional. Esta seção examina em mais detalhes outras propostas do uso de geração automática de problemas (*Automatic Problem Generation* – APG) em desafios de Segurança, relacionando-as com o presente trabalho. O PicoCTF¹² é um jogo de desafios desenvolvido no Carnegie Mellon University, destinado a estudantes. Mantido e aplicado anualmente, apresenta problemas na forma de um jogo baseado na *web*, com duas formas de visualização de desafios: *baseada em texto*, ideal para competidores sérios, e *Toaster Wars*, um formato interativo com efeitos de sons e cenários (CHAPMAN; BURKET; BRUMLEY, 2014). Os trabalhos que discorrem sobre o PicoCTF (CHAPMAN; BURKET; BRUMLEY, 2014; BURKET et al., 2015) não descrevem as ferramentas necessárias para criar e resolver os problemas propostos. Na edição de 2014, o PicoCTF começou a

¹² <<https://picoctf.com/>>

utilizar APG para mitigar a ocorrência de cópias de *flags*, tendo sido identificado como o trabalho pioneiro nessa área.

O MetaCTF é um jogo de desafio, elaborado para estudantes, que automatizou a geração de problemas em sua versão de 2015. A competição foi voltada ao ensino de Engenharia Reversa de código e análise de *malware*, exigindo conhecimentos em ferramentas como `objdump`, `readelf`, `ltrace`, `strace`, `ptrace` e `gdb` (FENG, 2015). O trabalho de Feng (2015) avaliou a qualidade e a utilidade de tarefas extra-classe (*homework*) que antecederam a competição. O resultado obtido foi um aumento significativo no desempenho dos alunos nessas tarefas quando o jogo foi aplicado.

No MetaCTF de 2015 os problemas foram projetados para mitigar as oportunidades de trapaça (FENG, 2015). Não há publicações recentes sobre o MetaCTF, mas este continua utilizando diferentes instâncias de problemas com APG (FENG, 2017) e é agora chamado de PSU CTF¹³. O PicoCTF, em suas edições mais recentes, também segue utilizando APG com *flags* distintas em cada instância de problemas (CARLISLE, 2017).

Apesar do compartilhamento de *flags* ser dificultado por meio da geração de instâncias distintas de problemas, as equipes podem compartilhar as orientações sobre como resolvê-los. No entanto, as equipes que copiam a ideia de solução ainda devem aplicar os conceitos do passo a passo no seu desafio específico para solucioná-lo. Para Feng (2015) e Mansurov (2016), os jogadores automaticamente desenvolvem as habilidades e os conhecimentos que o desafio exige ao fazerem isso, embora nem sempre isto seja verdade. Em situações em que a execução de um mesmo *script* elaborado por um jogador ou uma equipe produz a *flag* esperada, não há, necessariamente, desenvolvimento de habilidades e conhecimentos.

O trabalho de Schreuders et al. (2017) disserta sobre o desenvolvimento do SecGen, uma ferramenta capaz de gerar desafios randômicos em conjuntos de máquinas virtuais. Os autores chamam esses conjuntos de máquinas virtuais de *cenários ricos*. O trabalho é rico em diversidade de problemas, tais como serviços de redes, esteganografia, vulnerabilidades em sistemas e jogo estilo CTF. A ferramenta proposta (SecGen) envolveu uma equipe de mais de dez pessoas. Ao criar uma competição, cada jogador recebe uma máquina virtual com um Sistema Operacional diferente e com conjuntos distintos de desafios, o que minimiza o compartilhamento de técnicas e *flags*. As análises do trabalho indicam satisfação na interação com a ferramenta e adequação no nível de dificuldade dos problemas. No entanto, o fato dos problemas não serem uniformes pode gerar um desequilíbrio entre os competidores, fazendo com que certos jogadores sejam favorecidos ou prejudicados em função de seus conhecimentos e dos problemas sorteados em seus respectivos desafios. Por exemplo, um

¹³ <<https://cs201.oregonctf.org/>>

jogador com um conjunto amplo de conhecimentos pode ser prejudicado caso o desafio gerado concentre-se em uma sub-área com a qual ele não está familiarizado; de forma análoga, um jogador com conhecimentos limitados pode ser beneficiado caso receba um desafio com problemas concentrados em uma área com a qual está familiarizado.

A geração automática de problemas distintos traz duas vantagens: a primeira é propiciar a reaplicação da atividade sem que soluções existentes interfiram diretamente nas soluções das novas instâncias geradas (BURKET et al., 2015); a segunda é identificar quando uma equipe copia respostas da outra (BURKET et al., 2015; FENG, 2015). Fornecer instâncias distintas de problemas pode fazer com que as equipes elaborem sequências de passos, utilizem ferramentas e/ou apliquem parâmetros próprios e distintos dos utilizados por outras equipes nas soluções dos seus problemas. Apesar das vantagens, uma dificuldade deste formato é garantir que diferentes instâncias de problemas tenham níveis de dificuldade semelhantes.

Embora o PicoCTF tenha utilizado criação automática de problemas de Segurança Computacional em sua execução, seus problemas abordavam sempre uma categoria, ou seja, os problemas não compunham mais de uma técnica diferente (BURKET et al., 2015). O MetaCTF gerou exercícios automaticamente, mas em uma abordagem restrita a uma classe de problemas. Além disso, o trabalho de Feng (2015) relata o uso de codificação `base64` em um de seus problemas, mas não trabalha o conceito de composição de técnicas, ficando restrito a uma técnica e, eventualmente, a uma forma de codificação adicionada ao problema. O SecGen permite aninhar técnicas, mas somente no sentido de compor um problema com alguma forma de codificação, sendo esta aplicada diretamente na *flag* (SCHREUDERS et al., 2017).

A Tabela 1 apresenta um comparativo entre os trabalhos relacionados e o TreasureHunt, que é a proposta deste trabalho. A tabela mostra as diferenças e semelhanças dos principais trabalhos no que diz respeito à geração automática, composição de problemas, uniformidade de problemas e classes de problemas abordadas. A geração automática de problemas é vista em todos os trabalhos, com a ressalva que o PicoCTF gera apenas problemas isolados, não competições inteiras. Com relação à geração de problemas usando composição de técnicas, o PicoCTF não possui essa característica, enquanto que MetaCTF e SecGen limitam esta composição à codificação da *flag* após gerar um problema utilizando uma técnica (razão pela qual estão representados com o símbolo \pm na Tabela 1). O TreasureHunt aplica a composição de forma mais ampla, evitando apenas combinações inviáveis de técnicas, como será mostrado na sequência deste trabalho.

A uniformidade de problemas diz respeito à aplicação de problemas de complexidade e técnicas semelhantes a todos os jogadores. No SecGen, jogadores po-

Tabela 1 – Comparativo entre os trabalhos relacionados e a ferramenta proposta.

Trabalho	Geração automática	Composição de problemas	Uniformidade de problemas	Classes de problemas abordadas
PicoCTF	problemas	×	✓	<ul style="list-style-type: none"> • Engenharia Reversa • Web • Miscelânea • Codificação/ Criptografia
MetaCTF	competição	±	×	<ul style="list-style-type: none"> • Engenharia Reversa
SecGen	competição	±	×	<ul style="list-style-type: none"> • Web • Forense • Miscelânea • Codificação/ Criptografia
TreasureHunt	competição	✓	✓	<ul style="list-style-type: none"> • Engenharia Reversa • Forense • Miscelânea • Codificação/ Criptografia

Fonte: elaborado pelo autor, 2018.

dem receber máquinas virtuais totalmente diferentes, tanto em Sistema Operacional quanto em exercícios e em classes de problemas. Isso significa que a familiaridade ou a dificuldade com determinada técnica podem influenciar no placar. O MetaCTF gera problemas com a mesma técnica, pois só trabalha com Engenharia Reversa, mas eles são não uniformes no que diz respeito à complexidade. Desta forma, jogadores recebem instâncias mais ou menos complexas que os outros, também influenciando no placar e na quantidade de esforço necessário para resolver o exercício.

A lista de classes de problemas abordadas em cada trabalho leva em consideração apenas as classes efetivamente implementadas, conforme descrito nas referências, e não todas as classes que poderiam ser implementadas. Observa-se que o MetaCTF é restrito a problemas de Engenharia Reversa, enquanto que os demais trabalhos abordam um conjunto mais extenso de classes de problemas.

Em comum, todos os trabalhos fornecem o jogo com sistema de placar, aplicam as atividades de forma não linear e possibilitam a criação de instâncias exclusivas, com as ressalvas já feitas.

2.5 CONSIDERAÇÕES DO CAPÍTULO

Este capítulo apresentou o tema Jogos para Segurança e a inserção destes na educação. Foram apresentados os principais tipos e exemplos de jogos da área, bem como foram discutidos aspectos pedagógicos e operacionais sobre eles, incluindo questões sobre o ganho de habilidades e particularidades no objetivo dos diferentes jogos.

O capítulo também apresentou os principais trabalhos relacionados, abordando a geração automática de problemas em cada um deles e as diferenças e semelhanças com a solução desenvolvida nesta pesquisa, detalhada na sequência deste trabalho.

Foram discutidas dificuldades recorrentes na organização de jogos, tais como o compartilhamento de respostas e a perda do fator surpresa quando os problemas se repetem. O próximo capítulo apresenta uma proposta, baseada na geração automática de problemas, para contornar tais dificuldades no contexto de jogos do tipo *desafio* ou *caça ao tesouro*.

3 GERAÇÃO AUTOMATIZADA DE DESAFIOS DE SEGURANÇA

Conforme discutido no Capítulo 2, jogos e competições são um meio eficaz de desenvolver habilidades em Segurança Computacional. Ao incentivar os jogadores a ampliar seus conhecimentos práticos sobre técnicas e ferramentas de Segurança, essas atividades podem desempenhar um papel importante para complementar a formação teórica de alunos de cursos de Computação e despertar o interesse por Segurança, atraindo assim novos talentos para a área. Nesse contexto, jogos do tipo caça ao tesouro ou desafio são particularmente interessantes, pois apresentam grande flexibilidade em termos de complexidade de problemas e conhecimentos específicos exercitados, e não exigem infraestruturas computacionais sofisticadas ou dedicadas.

A organização desse tipo de competição envolve a elaboração de problemas, o que exige conhecimento técnico especializado, que costuma ser escasso, além de geralmente ser algo manual e trabalhoso. Uma alternativa poderia ser reaproveitar problemas existentes, mas isso é desvantajoso em dois aspectos: a perda do fator surpresa e a possibilidade do compartilhamento de respostas. Para contornar essas dificuldades, propõe-se automatizar a geração dos problemas que compõem um desafio, usando de aleatorização para fazer com que os problemas gerados sejam únicos. Desta forma, a reaplicação da atividade possibilita o reaproveitamento de problemas, pois as instâncias geradas serão distintas.

Neste capítulo, apresenta-se a proposta de uma competição com geração automatizada de desafios de Segurança. Na Seção 3.1 estão os parâmetros gerais do desafio. A Seção 3.2 discute a seleção de técnicas do gerador de desafios. A Seção 3.3 detalha o protótipo da implementação, e a Seção 3.4 apresenta as considerações sobre o capítulo.

3.1 PARÂMETROS GERAIS DO DESAFIO

Em geral, competições de Segurança costumam ser atividades que apenas atestam o conhecimento dos jogadores, trazendo as ideias de concorrência e superação. Em outras palavras, o público-alvo costuma ser formado por indivíduos com conhecimentos prévios no assunto e que apenas buscam vencer a competição com base nos critérios da instituição organizadora (EAGLE, 2013). No entanto, no âmbito educacional, no qual este trabalho se insere, competir deixa de ser o objetivo principal para ser um meio de influenciar positivamente os processos de ensino e aprendizagem, considerando que o papel de jogador será desempenhado por estudantes. Tão importante quanto reconhecer os jogadores mais capacitados é estimular os demais

jogadores a aprofundarem seus conhecimentos em Segurança e a seguir carreira na área.

As competições podem ser individuais ou por equipes. Embora muitas competições de Segurança sejam realizadas em equipes (CTF..., 2017), a competição individual traz, no âmbito educacional, a vantagem de permitir uma mensuração mais clara do desempenho e da percepção de satisfação na atividade, que são objeto de avaliação neste trabalho. Além disso, turmas de Computação não costumam ser numerosas.

Os jogos do tipo caça ao tesouro são compostos por um conjunto de problemas que podem ser independentes (não lineares), sequenciais (lineares) ou mistos. A competição proposta neste trabalho disponibiliza todos os desafios de forma não linear, permitindo que o jogador os resolva na ordem que preferir.

Cada problema costuma ser associado a uma pontuação de acordo com a sua complexidade. A versão proposta considera que todos os problemas têm peso 1, assim, conta-se apenas o número de acertos e elimina-se a necessidade de contagem de pontos. A atribuição de pontos distintos entre os problemas poderia provocar diferentes reações nos jogadores. Primeiramente, seria necessário garantir que os problemas mais complexos recompensassem os jogadores com mais pontos, mas nem sempre é fácil estabelecer esta ordem antes da aplicação da atividade. Além disso, a pontuação diferente poderia influenciar na ordem em que os problemas seriam resolvidos, podendo interferir no fator competição e no tempo despendido em cada atividade. A variável tempo é considerada somente em casos de desempate, o que significa dizer que quando há empate no número de acertos, o horário da última submissão correta desempata em ordem decrescente, ficando à frente aquele que submeteu a última resposta correta primeiro.

3.2 SELEÇÃO DE TÉCNICAS

3.2.1 Análise de Desafios Existentes

O desenvolvimento da ferramenta foi iniciado seguindo o planejamento proposto na Seção 1.5 (Método de Pesquisa). A partir de um repositório de competições do tipo desafio (CTF..., 2017), mantido pela comunidade, analisou-se na íntegra o conjunto de competições ocorridas entre janeiro de 2016 e março de 2017. Este período foi considerado para restringir a pesquisa às competições mais recentes, realizando o levantamento sobre as técnicas e os formatos mais atuais de jogos do tipo desafio. Neste período, 84 competições foram inseridas no repositório e foram objeto de análise nesta pesquisa. Com base no enunciado de cada exercício e em soluções divulgadas, classes genéricas, às quais as técnicas envolvidas na solução estariam

contidas, foram definidas pelo autor. Assim, após análise das competições para as quais todo problema foi contabilizado, emergiram as cinco classes de problemas listadas a seguir:

- Criptografia/Codificação
- Engenharia Reversa
- Forense
- *Web*
- Miscelânea

Os problemas de Criptografia/Codificação somaram 306, enquanto 235 problemas envolviam Engenharia Reversa, 385 problemas aplicavam técnicas da área Forense, 264 de *Web* e 153, que não se enquadravam nas classes anteriores, foram classificados como miscelâneos.

Ao todo, 1250 problemas foram analisados. As técnicas envolvidas nos problemas foram contabilizadas e documentadas, de forma a tornar possível a rastreabilidade destas, identificando em quais competições elas foram usadas. Cabe ressaltar que, dentre os 1250 problemas, 86 (6,9%) eram compostos, isto é, formados pela junção de duas ou mais técnicas.

A característica de linearidade foi analisada para cada competição. Em 48 competições foi possível encontrar a informação, sendo a forma não linear vista em 46 destas (95,8% do total). A forma linear e o formato misto foram encontrados em uma competição cada um.

No fim de 2017 (após a realização deste levantamento, portanto), foi publicado um trabalho com pesquisa semelhante, mapeando os desafios constantes no mesmo repositório analisado nesta pesquisa, nos últimos três anos (BURNS et al., 2017). Burns et al. (2017) categorizaram os problemas do repositório em seis áreas: criptografia, *web*, engenharia reversa, forense, *pwn* (exploração de vulnerabilidades) e miscelânea. A proporção de problemas apresentadas no trabalho de Burns et al. (2017) ficou próxima à deste trabalho, exceto pelo fato de que as técnicas com o objetivo de explorar vulnerabilidades e ganhar privilégios foram classificadas como *pwn*, e não engenharia reversa, embora sejam utilizadas técnicas de engenharia reversa na resolução deste tipo de problema.

3.2.2 Técnicas Seleccionadas

A partir de um rol de cerca de 200 técnicas distintas encontradas nas competições constantes no repositório, as 40 técnicas mais frequentes foram separadas e, destas, oito foram escolhidas para a construção do protótipo da ferramenta geradora de desafios automáticos e compostos.

Os critérios principais de seleção de técnicas foram o perfil do público-alvo e o objetivo do trabalho e, de forma secundária, a familiaridade do autor com as técnicas. Para definir o perfil do público-alvo antes de selecionar as técnicas, partiu-se da premissa de que os jogadores seriam estudantes sem conhecimentos avançados em Linux e Segurança Computacional, o que corresponde ao perfil médio dos discentes que eventualmente viriam a participar da avaliação do trabalho. Esta ação foi necessária porque a definição das técnicas ocorreu antes do período de matrículas, não havendo turmas efetivamente formadas.

As técnicas seleccionadas para criação do protótipo da ferramenta compositora de problemas foram:

- **Comentário em código-fonte de página HTML:** problema em que a *flag* é inserida arbitrariamente em alguma posição de um arquivo HTML válido. A página carrega estilos e imagens diferentes em cada instância, bem como contém códigos criptografados para confundir os jogadores. Este exercício faz parte da classe Miscelânea e realiza análise de código-fonte. Pode ser resolvido, entre outras formas, visualmente ou por meio de comandos (inclusive programados).
- **Comentário no arquivo `robots.txt`:** problema que contém um conjunto de arquivos de um *website* válido e, entre eles, um arquivo `robots.txt`¹ que contém uma quantidade parametrizável de comentários. Todos os comentários são sequências aleatórias de caracteres, exceto um, que corresponde à *flag*. Este exercício faz parte da classe Miscelânea e realiza análise de código-fonte. Pode ser resolvido, entre outras formas, visualmente ou por meio de comandos (inclusive programados).
- **(De)codificação de arquivo em `base64`:** problema em que a *flag* é inserida em um arquivo de texto sorteado arbitrariamente em um diretório parametrizável. Após isso o arquivo é codificado em formato `base64`². Este exercício faz parte

¹ `robots.txt` é um arquivo de configuração de indexação de conteúdo na *web*. Ele é responsável por identificar diretórios e arquivos cujo acesso é permitido ou proibido aos robôs de buscadores (MONDAL et al., 2012)

² `base64` é um esquema de codificação comumente usado que representa dados binários em um formato de *string* ASCII (LIU et al., 2011)

da classe Criptografia/Codificação e pode ser resolvido, entre outras formas, com o uso de um decodificador de `base64`.

- **(Des)criptografia de Cifra de César:** problema em que a *flag* é inserida em um arquivo de texto sorteado arbitrariamente em um diretório parametrizável. Após isso, o arquivo é criptografado com a Cifra de César³, com chave aleatoriamente escolhida entre 1 e 25, garantindo que o texto cifrado não será igual ao original. Este exercício faz parte da classe Criptografia/Codificação e pode ser resolvido, entre outras formas, manualmente ou com uma ferramenta decodificadora da Cifra de César.
- **(De)codificação de caractere ASCII para inteiro:** problema em que a *flag* é inserida em um arquivo de texto sorteado arbitrariamente em um diretório parametrizável. Após isso o arquivo é codificado em valores inteiros que representam os caracteres originais com base na tabela ASCII (*American Standard Code for Information Interchange*), separados por espaço em branco. Este exercício faz parte da classe Criptografia/Codificação e pode ser resolvido, entre outras formas, manualmente ou com uma ferramenta de conversão de valor inteiro para caractere.
- **Descompilar binário e obter fonte Java:** problema em que um código Java, que produz um texto arbitrário, é criado. Neste código é inserida uma *flag* em uma variável e o código é compilado para *bytecode*, gerando um arquivo `.class`. Este exercício faz parte da classe Engenharia Reversa, e pode ser resolvido, entre outras formas, por meio de comandos de impressão de caracteres ou ferramentas de descompilação de *bytecode* Java.
- **Descompilar binário e obter fonte Python:** semelhante ao problema anterior, este problema contém um código em Python que imprime um texto arbitrário. Neste código é inserida uma *flag* em uma variável e o código é compilado para *bytecode*, gerando um arquivo `.pyc`. Este exercício faz parte da classe Engenharia Reversa, e pode ser resolvido, entre outras formas, por meio de comandos de impressão de caracteres ou ferramentas de descompilação de *bytecode* Python.
- **Esteganografia em imagens:** problema em que a *flag* é esteganografada⁴ em uma imagem escolhida arbitrariamente em um diretório parametrizável. Este exercício faz parte da classe Forense e pode ser resolvido por meio da ferramenta padrão de esteganografia utilizada no gerador de problemas (`outguess`).

³ Cifra de César (também conhecida como *Caesar Cipher*) é uma cifra de substituição monoalfabética baseada na troca de cada caractere do texto original por outro n posições à frente no alfabeto, de forma circular, sendo n a chave (GOYAL; KINGER, 2013).

⁴ Técnica que consiste em esconder mensagens utilizando um meio de cobertura (HUSSAIN et al., 2015).

Todas as técnicas citadas foram implementadas neste trabalho, inicialmente, em versões individuais, em *scripts* separados (etapa 1 descrita na Seção 1.5 – Método de Pesquisa), para depois comporem o código do gerador de desafios (etapa 2 descrita na Seção 1.5 – Método de Pesquisa). Este gerador, explicado na sequência do capítulo, é um *script* que carrega os parâmetros do usuário e realiza a criação das instâncias de cada problema.

A Figura 5 apresenta duas instâncias distintas do problema “Comentário em código-fonte de página HTML”, obtidas através da versão inicial do gerador de desafios, e seus respectivos códigos-fonte. Nota-se que a *flag* aparece em linhas distintas (linha 40 na imagem da esquerda e linha 36 na imagem da direita), bem como não é composta pela mesma sequência de símbolos. Apesar disso, optou-se por *flags* de tamanho igual. As imagens exibidas e o estilo das páginas são aleatórios, de forma que cada usuário receba um arquivo com configurações diferentes.

Figura 5 – Duas instâncias do problema “Comentário em código-fonte de página HTML” e seus respectivos códigos-fonte.



Fonte: elaborado pelo autor, 2017.

O problema *Descompilar binário e obter fonte Java* permite que o usuário chegue à solução de formas distintas. Algumas formas de se chegar à solução envolvem o uso de ferramentas de descompilação e análise do código, bem como integração de uma ferramenta destas com comandos de filtragem de linhas. Outra forma é através do comando `strings`. A Figura 6 exibe a solução para o problema utilizando a ferramenta `strings`.

Esta solução deixa claro que, por mais que as palavras secretas sejam distintas, os mesmos comandos podem ser utilizados para se chegar à resposta. Portanto, compartilhar a resposta (*flag* ou palavra secreta) não surte efeito para os jogadores,

Figura 6 – Solução alternativa de quatro instâncias do problema “Descompilar binário e obter fonte Java”.

The image displays four terminal windows arranged in a 2x2 grid, each showing the execution of the 'strings' command on a specific challenge file. The user is 'ricardo@leicester' and the directory is '/var/www/html/TreasureHunt/Desafios\$'. Each window shows the output of the 'strings' command, which includes the path to the 'TreasureHunt' class file and the class name itself.

```

ricardo@leicester: /var/www/html/TreasureHunt/Desafios$ strings 1/1/Desafio.class | grep TreasureHunt
TreasureHunt{jjpFtgbw}
ricardo@leicester: /var/www/html/TreasureHunt/Desafios$

ricardo@leicester: /var/www/html/TreasureHunt/Desafios$ strings 3/1/Desafio.class | grep TreasureHunt
TreasureHunt{zqlyIfvu}
ricardo@leicester: /var/www/html/TreasureHunt/Desafios$

ricardo@leicester: /var/www/html/TreasureHunt/Desafios$ strings 2/1/Desafio.class | grep TreasureHunt
TreasureHunt{I226a5wk}
ricardo@leicester: /var/www/html/TreasureHunt/Desafios$

ricardo@leicester: /var/www/html/TreasureHunt/Desafios$ strings 4/1/Desafio.class | grep TreasureHunt
TreasureHunt{tfU12xfI}
ricardo@leicester: /var/www/html/TreasureHunt/Desafios$

```

Fonte: elaborado pelo autor, 2018.

mas os comandos necessários para a solução, quando compartilhados, poderão fazer com que os jogadores obtenham as respostas e assimilem o funcionamento das ferramentas ora em uso.

A etapa de implementação do gerador de desafios com técnicas compostas demonstrou ser possível compor técnicas gerando instâncias distintas para cada jogador. A composição de técnicas se configura em aplicar uma técnica e depois fazer com que a *flag*, o arquivo de saída ou o conjunto de arquivos de saída seja(m) submetida(s) à aplicação de uma outra técnica – que pode ser a mesma novamente – para, então, ser gerado um novo arquivo ou conjunto de arquivos de saída. Por exemplo, um problema com a composição entre *base64* e César irá aplicar a codificação *base64* em um arquivo, gerando um arquivo de saída intermediário e temporário. A este arquivo de saída será aplicada a Cifra de César, gerando um novo arquivo de saída que compõe as duas técnicas em um problema, sendo este disponibilizado ao jogador. Assim, para solucionar o exercício, o jogador precisa encontrar a chave utilizada na Cifra de César, obter o arquivo codificado em *base64*, decodificá-lo e procurar pela *flag*.

Há, porém, técnicas que não podem ser compostas com outras. A Tabela 2 mostra as técnicas que podem ser compostas com a marcação de um “X” que, quando presente na célula_{ij}, onde *i* é a linha e *j* a coluna, significa possibilidade de composição. Nela é possível ver, por exemplo, que as composições (*Esteganografia em Imagens* ◦ *Esteganografia em Imagens*) e (*Codificação em base64* ◦ *Cifra de César*) são possíveis, ao passo que a composição (*Cifra de César* ◦ *Cifra de César*) não é, já que a aplicação da cifra mais de uma vez terá sempre uma chave equivalente à aplicação da cifra uma vez.

É importante citar que a ordem de aplicação das técnicas influencia na instância gerada. Por exemplo, aplicar a codificação *base64* antes de um problema de descompilação de código implica gerar uma *flag*, codificá-la e depois inseri-la no código a ser compilado. Caso a ordem inversa seja aplicada, uma *flag* é inserida no

Tabela 2 – Matriz de composições.

	HTML	Robots	Base64	Cesar	A2I	Java	Python	Esteg
HTML			X	X	X			X
Robots			X	X	X			X
Base64	X	X	X	X	X	X	X	X
César	X	X	X			X	X	X
A2I	X	X	X		X	X	X	X
Java			X	X	X			X
Python			X	X	X			X
Esteg			X	X	X			X

Legenda:

- **HTML:** Comentário em código-fonte de página HTML
- **Robots:** Comentário no arquivo `robots.txt`
- **Base64:** (De)codificação de arquivo em `base64`
- **César:** (Des)criptografia de Cifra de César
- **A2I:** (De)codificação de caractere ASCII para inteiro
- **Java:** Descompilar binário e obter fonte Java
- **Python:** Descompilar binário e obter fonte Python
- **Esteg:** Esteganografia em imagens

Fonte: elaborado pelo autor, 2017.

código fonte e, após a compilação, o arquivo de *bytecode* é codificado em `base64`.

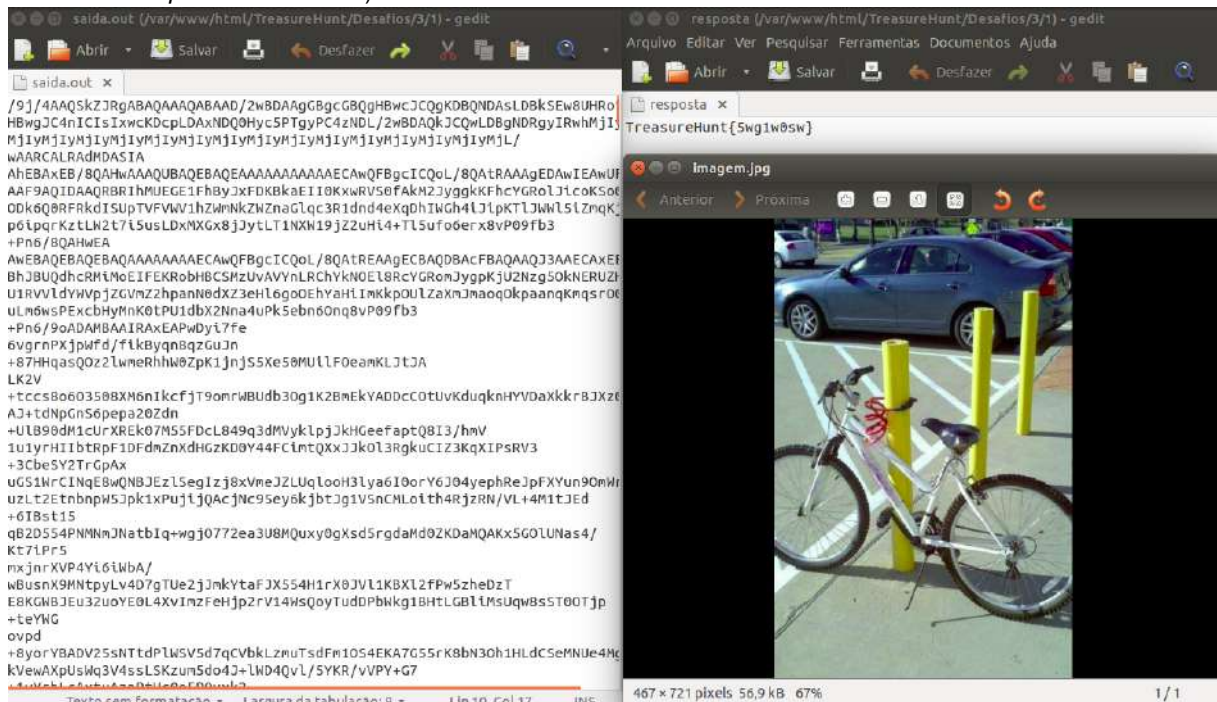
As Figuras 7 e 8 apresentam instâncias do problema que compõe *Esteganografia em imagens* e *(De)codificação de arquivo em base64*. Neste problema, o jogador recebe um arquivo com nome parametrizável, mas definido como `saida.out` por padrão. O arquivo está codificado em `base64` e, portanto, deve ser decodificado. Ao proceder à decodificação, o arquivo resultante, exibido como `imagem.jpg`, representa uma imagem JPEG (*Joint Photographic Experts Group*) que contém um texto esteganografado com a ferramenta `outguess`. Utilizando o `outguess` é possível extrair o conteúdo da imagem e assim obter a *flag*. As imagens selecionadas para cada instância são diferentes, bem como as *flags* geradas, produzindo arquivos diferentes quando codificados em `base64`.

3.3 PROTÓTIPO DE IMPLEMENTAÇÃO

3.3.1 Visão Geral

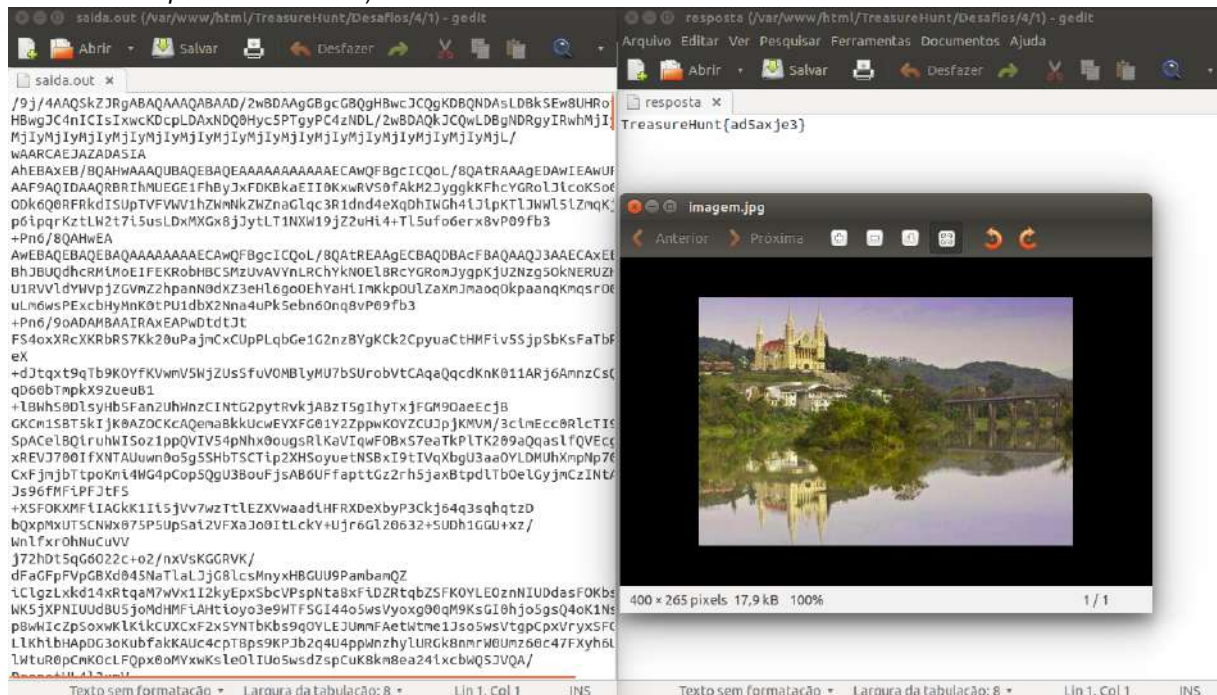
A ferramenta implementada é composta por duas partes: o *script* gerador de problemas e a aplicação web. O gerador de desafios, implementado em Shell script, se utiliza do SGBD MySQL e do servidor web Apache. A aplicação web foi criada utilizando a linguagem PHP. O fluxo das atividades envolvidas na geração de competições pode ser visto no Diagrama de Atividades da Figura 9.

Figura 7 – Primeira instância do problema composto (*Esteganografia em imagens* ◦ (*De*)codificação de arquivo em base64).



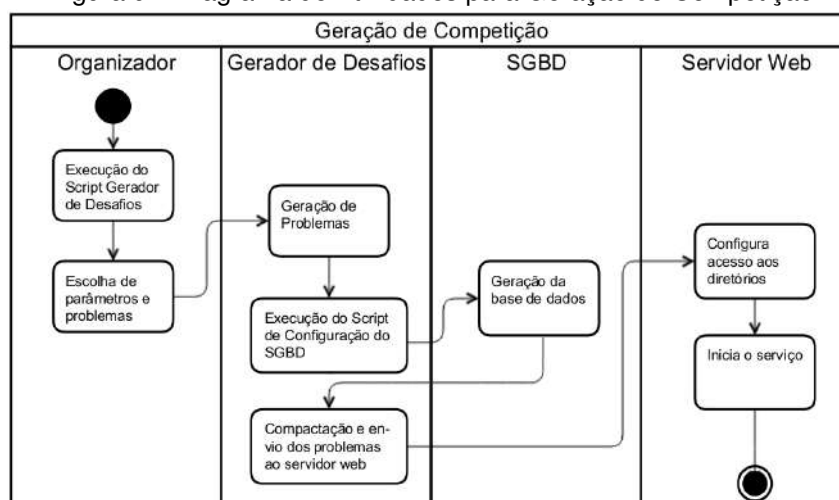
Fonte: elaborado pelo autor, 2018.

Figura 8 – Segunda instância do problema composto (*Esteganografia em imagens* ◦ (*De*)codificação de arquivo em base64).



Fonte: elaborado pelo autor, 2018.

Figura 9 – Diagrama de Atividades para Geração de Competição.



Fonte: elaborado pelo autor, 2018.

Os requisitos funcionais do gerador de problemas são:

- RF1. O sistema deve ser capaz de gerar e compor problemas pseudoaleatórios;
- RF2. O sistema deve ser capaz de cadastrar as respostas dos problemas no SGBD;
- RF3. O sistema deve anonimizar os jogadores por meio de identificadores;
- RF4. O sistema deve ser capaz de criar credenciais automaticamente;
- RF5. O sistema deve ser capaz de comprimir os problemas de cada jogador; e
- RF6. O sistema deve ser capaz de salvar cópia dos problemas para o organizador.

Os requisitos funcionais da aplicação *web* são:

- RF1. O sistema deve ser capaz de autenticar os usuários;
- RF2. O sistema deve ser capaz de permitir o envio de respostas;
- RF3. O sistema deve ser capaz de analisar as respostas enviadas;
- RF4. O sistema deve ser capaz de fornecer o placar da competição;
- RF5. O sistema deve ser capaz de fornecer resultados de problemas de forma individualizada; e
- RF6. O sistema deve ser capaz de salvar os dados das competições.

Os requisitos não funcionais gerais são:

- RNF1. O sistema deve utilizar o SGBD MySQL;
- RNF2. O sistema deve utilizar o servidor *web* Apache;
- RNF3. O sistema deve funcionar em Sistema Operacional *Unix-like*.

Os requisitos não funcionais do gerador de problemas são:

- RNF1. O sistema deve gerar cada instância de problema simples em menos de 30 segundos;
- RNF2. O sistema deve gerar cada instância de problema composto em menos de 60 segundos; e
- RNF3. O *script* deve ser implementado em linguagem Shell.

Os requisitos não funcionais da aplicação *web* são:

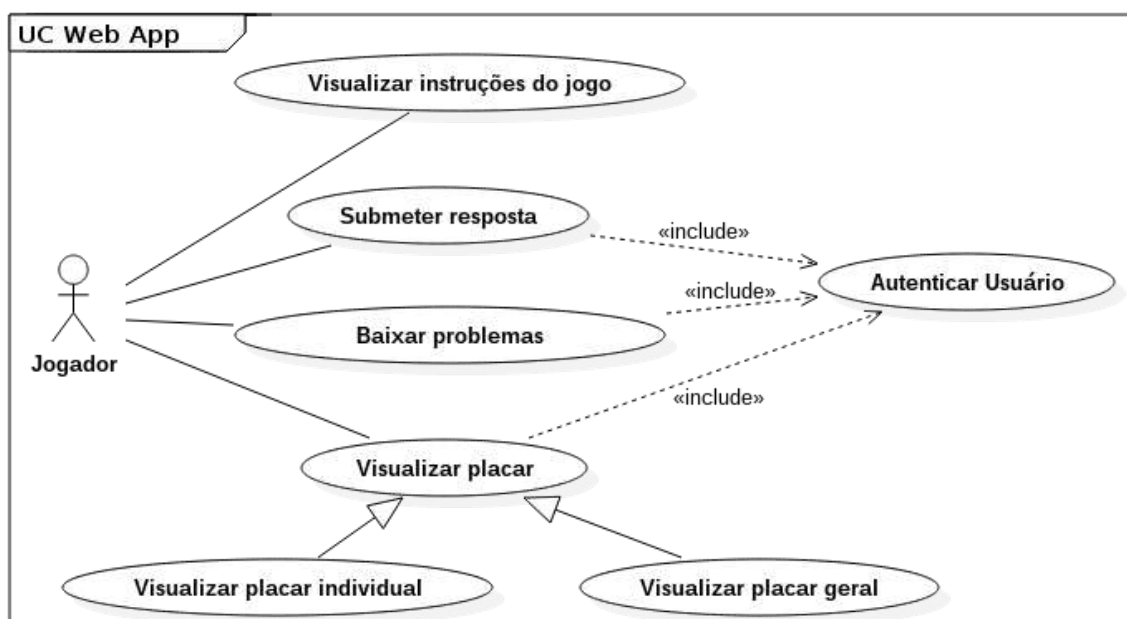
- RNF1. A aplicação deve ser implementada em PHP; e
- RNF2. O sistema deve fornecer o resultado de análise de resposta em menos de 5 segundos.

O organizador utiliza o gerador de desafios para informar parâmetros e definir a quantidade de jogadores e de problemas. O *script* do gerador é responsável por realizar a comunicação com o SGBD e o servidor *web*, deixando a competição pronta após o encerramento de sua execução.

Para que cada usuário receba uma instância distinta dos problemas, será fornecido acesso à aplicação *web* por meio de um identificador individual. Ao realizar a autenticação, o usuário tem acesso a um arquivo compactado contendo todas as suas instâncias de problemas. Todos os jogadores receberão instâncias dos mesmos problemas, seguindo a mesma ordem, com *flags* geradas aleatoriamente, que são únicas para cada jogador. As ferramentas necessárias para a resolução dos problemas não são enviadas no arquivo compactado, sendo desejável que já estejam instaladas nas máquinas dos jogadores, embora esta decisão fique a critério do organizador da competição. A Figura 10 apresenta os casos de uso referentes à interação do jogador com a aplicação *web*.

Quando o sistema é liberado para uso, cada jogador recebe seu identificador e uma senha, que são pessoais e intransferíveis. Para realizar o acesso ao sistema e efetuar a autenticação é necessário utilizar um computador conectado na mesma rede

Figura 10 – Casos de Uso da aplicação web.



Fonte: elaborado pelo autor, 2018.

em que o servidor do jogo está executando, acessando-o através de um navegador *web*.

Após realizar a autenticação, é possível baixar o arquivo compactado com os problemas. A partir deste momento, o jogador deve focar na resolução dos problemas, e pode interagir com o sistema *web* através da submissão da *flag*. Há ainda a opção de observar o placar da competição, que só mostra os identificadores dos jogadores (e assim o jogador não sabe quem são os outros colegas, mas sabe sua classificação), o número de acertos e o horário da última submissão correta, e um placar individual detalhado, que mostra todos os identificadores de problemas seguidos por seu *status* (“Resolvido” ou “Não Resolvido”) e o número de tentativas, como mostra a Figura 11.

Figura 11 – Placar individual detalhado do TreasureHunt.

Problema	Status	Nº de Tentativas
1	Resolvido	1
2	Resolvido	1
3	Não Resolvido	3
4	Não Resolvido	0
5	Não Resolvido	2

Fonte: elaborado pelo autor, 2018.

3.3.2 Funcionamento do Gerador de Desafios

O gerador de desafios é composto por um conjunto de *scripts* e diretórios com imagens e textos predefinidos. Nestes *scripts* estão os parâmetros, com diretórios, nomes de arquivos, quantidade de *flags* e outros, as funções de validação de entrada, de exibição de menu, de composição de técnicas, de geração de problemas individuais e de preparação da base de dados. Os diretórios de imagens e textos são utilizados por problemas específicos em que o gerador sorteia os arquivos que farão parte da instância de cada jogador. Há um conjunto de arquivos padrão enviado junto com o gerador, mas o organizador pode manipular estes diretórios, inserindo ou removendo arquivos, a seu critério.

O *script* principal (`jogo.sh`) solicita as informações necessárias ao organizador, que são quantidade de problemas e jogadores. A partir disso, o *script* solicita ao organizador que informe os códigos dos problemas de acordo com numeração fixa definida (Figura 12) e gera os problemas. A validação impede que problemas inexistentes sejam solicitados.

Para gerar as instâncias aleatórias, utilizou-se a ferramenta `shuf` e o arquivo `/dev/urandom`. A ferramenta `shuf` gera permutações aleatórias e foi utilizada para o sorteio de arquivos que fariam parte das instâncias de cada problema. O arquivo `/dev/urandom` cria combinações pseudoaleatórias e foi utilizado para gerar o texto contido nas palavras secretas.

O gerador de problemas proposto neste trabalho considera o número de pontos e o horário da última submissão correta ao ranquear os jogadores. Na versão atual da ferramenta, cada problema vale um ponto. Futuramente, a proposta do trabalho é permitir a parametrização destes valores, de forma que o organizador – professor responsável pela elaboração e aplicação da atividade – os arbitre, assim sendo possível, portanto, manter todos os problemas com pesos progressivos ou até mesmo nulos, caso este fator não seja considerado importante. A variável tempo é considerada somente em casos de desempate, o que significa dizer que quando há empate no número de acertos, o horário da última submissão correta desempata em ordem decrescente, ficando à frente aquele que submeteu primeiro. Mesmo que o tempo não esteja sendo utilizado para fins de desempate, a última submissão correta sempre aparecerá no placar, como apresentado na Figura 16, na Subseção 3.3.3.

Como já citado, a competição foi prevista para ser realizada individualmente, embora seja possível adaptá-la para ser jogada em equipes, a critério do organizador. Neste caso, o ID (identificador) do jogador torna-se o ID da equipe, e é possível que todos os jogadores tenham acesso simultâneo ao sistema.

É importante mencionar que a criação dos problemas individuais e compostos

segue um fluxo diferente. No problema individual, o *script* de geração de instâncias da técnica selecionada é executado e cria os arquivos de saída de cada jogador. Uma vez escolhidas as técnicas presentes em um problema composto, o *script* de geração de instâncias da primeira técnica é executado e gera a palavra secreta no(s) arquivo(s). Depois disso, o(s) arquivo(s) resultantes é/são utilizado(s) como parâmetro de entrada para a aplicação da segunda técnica, gerando um arquivo ou um conjunto de arquivos finais. Os arquivos intermediários são excluídos após a conclusão da criação das instâncias do problema. Considerando que cada técnica tem suas especificidades (há técnicas aplicáveis somente a imagens ou a textos, por exemplo), um valor de controle (numérico) também é usado para identificar a composição, atendendo as peculiaridades de cada técnica no momento de gerar as instâncias. Assim, a inserção e a composição de uma nova técnica ao gerador, quando esta for implementada, não são diretas e automáticas. As composições com uma nova técnica não exigem, de uma maneira geral, que novas implementações sejam desenvolvidas, mas apenas ajustadas com as particularidades da técnica.

Em uma competição de desafios, é comum a palavra secreta seguir um padrão, sendo formada por uma sequência de símbolos que não deixe dúvidas de que é a resposta. Seguindo essa prática, o desafio proposto tem, em todos os seus problemas, a *flag* no formato `TreasureHunt{texto_arbitrario}`. Este formato faz com que qualquer outro texto encontrado, decodificado ou produzido na tentativa de resolução do desafio seja descartado por se saber de antemão que não está no padrão de *flag*.

A Figura 12 mostra o funcionamento do *script* principal. Ao escolher um problema, o usuário deve informar o identificador numérico deste (exemplo: 2 para o problema *(Des)criptografia de Cifra de César*). Caso a opção seja por um problema composto, deve informar os dois dígitos dos problemas na ordem em que quer realizar a composição. Por exemplo, para a composição *(Codificação em base64 o Cifra de César)*, o código informado deve ser “12”.

O programa ainda gera um arquivo com as respostas e permite que o organizador mantenha cópias das instâncias dos usuários. Por padrão, ao concluir a geração de desafios, o conjunto de problemas de cada usuário é compactado em um arquivo ZIP e enviado para o diretório do servidor *web*. Após esta etapa o jogo efetua chamada ao *script* `ConfiguraBD.sh` para proceder à criação da base de dados `TreasureHunt` (nome padrão), das tabelas e inserção de registros de usuário e respostas dos problemas no SGBD. As tabelas estão especificadas no Apêndice E através do dicionário de dados.

Após a configuração do Banco, os exercícios estão prontos. Caso o organizador deseje, pode aplicá-los também sem o uso do sistema de submissão *web*.

Figura 12 – Execução do *script* `jogo.sh`.

```

Treasure Hunt!
-----
Informe a quantidade de DESAFIOS: 6
Informe a quantidade de JOGADORES: 10
-----
Vamos criar os desafios!
-----
Lista de problemas disponíveis:
1: (De)codificação de arquivo em base64
2: (Des)criptografia de Cifra de César
3: Comentário em código-fonte de página HTML
4: Comentário no arquivo robots.txt
5: (De)codificação de caractere ASCII para inteiro
6: Descompilar binário e obter fonte Java
7: Descompilar binário e obter fonte Python
8: Esteganografia em imagens
Obs.: escolha 1 ou 2 problemas. Exibiremos uma mensagem de erro se a composição
não existir.
-----
Informe o(s) problema(s) do desafio 1: █

```

Fonte: elaborado pelo autor, 2018.

O conteúdo sensível do banco de dados inclui senhas e respostas, e foi armazenado utilizando *hash* e *salt* aleatório para as senhas e *hash* para as respostas. O algoritmo de *hash* escolhido foi o SHA256 (*Secure Hash Algorithm 256 bits*).

3.3.3 Funcionamento da Aplicação Web

O sistema *web* é a parte através da qual os jogadores interagem com o jogo. Embora o jogo seja resolvido de maneira *offline*, o sistema *web* é responsável por manter os arquivos de cada jogador, exibir o placar individual detalhado, exibir o placar geral, apresentar um texto explicativo sobre o jogo e apresentar o contato dos desenvolvedores.

Somente as regras do jogo e o contato dos desenvolvedores podem ser visualizados antes de realizar autenticação, assim mantendo os arquivos de cada jogador e as informações sobre seus exercícios resolvidos e não resolvidos de forma privada. A Figura 13 apresenta a tela inicial do sistema *web*.

Práticas de segurança foram adotadas na configuração do servidor *web*. Foram definidos o máximo de clientes permitidos e os redirecionamentos para páginas de erro e para a página principal. Esta configuração deve ser realizada manualmente pelo organizador, razão pela qual um modelo de arquivo de configuração do servidor *web* está disponível no repositório⁵ que contém os códigos do jogo.

Após autenticação, o usuário já pode realizar a submissão de *flag* (Figura 14). Para isso, basta informar o ID do problema e a palavra secreta descoberta. O ID do problema é o identificador numérico de cada problema. Este identificador é o número que corresponde ao nome do diretório em que o problema está. O arquivo ZIP contém

⁵ <<https://github.com/TreasureHuntGame/TreasureHunt>>

Figura 13 – Tela inicial do sistema web.



Fonte: elaborado pelo autor, 2018.

diretórios cujo nome é um número, e cada diretório contém um arquivo ou um conjunto de arquivos que corresponde a um problema. A Figura 14 mostra ainda que cada jogador é identificado por um ID.

Figura 14 – Tela de submissão de *flag* do sistema web.

Fonte: elaborado pelo autor, 2018.

Em caso de dúvidas, ao passar o cursor sobre o item de menu “Como Jogar?” o jogador pode ler algumas instruções sobre o jogo, conforme mostra a Figura 15.

Ao realizar a submissão, o usuário é imediatamente informado sobre o resultado do problema. Existem as seguintes opções de resposta:

Figura 15 – Tela de ajuda (“Como Jogar?”) do sistema *web*.

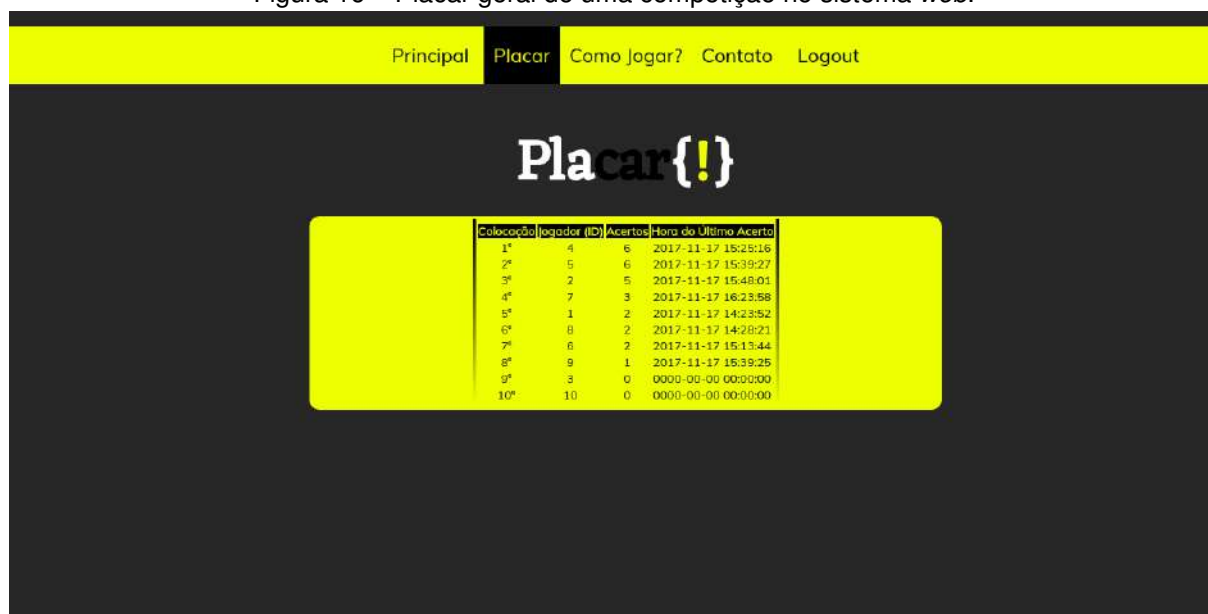


Fonte: elaborado pelo autor, 2018.

1. Problema com ID inválido!
2. Errou!
3. Errou! Considere submeter a flag no seguinte formato: `TreasureHunt{texto-aleatorio}`
4. Você já acertou a questão `ID_problema!`
5. Acertou! `n/m`

O caso 1 ocorre quando o usuário informa um identificador de problema inválido, tal como um número negativo ou superior ao número do último diretório contido no seu arquivo compactado. O caso 2 ocorre quando o usuário insere uma resposta no formato correto, mas contendo o texto aleatório incorreto. O caso 3 ocorre quando o usuário informa uma resposta em formato diferente do padrão. O caso 4 ocorre quando o usuário tenta ressubmeter uma resposta já acertada, o que evita submissões e inserção de registros desnecessárias. O caso 5 ocorre quando o usuário acerta a questão e informa quantos acertos ele já obteve em relação ao total de exercícios.

Todo jogador pode passar o cursor sobre a aba “Placar” para ver sua colocação durante a competição. A Figura 16 mostra que o Placar Geral exibe a colocação, o ID do jogador, o número de acertos e o horário da última submissão correta, que pode ser usado para desempatar a colocação em caso de empate no número de acertos.

Figura 16 – Placar geral de uma competição no sistema *web*.


Colocação	Jogador (ID)	Acertos	Hora do Último Acerto
1ª	4	6	2017-11-17 15:25:16
2ª	5	6	2017-11-17 15:39:27
3ª	2	5	2017-11-17 15:48:01
4ª	7	3	2017-11-17 16:23:58
5ª	1	2	2017-11-17 14:23:52
6ª	8	2	2017-11-17 14:28:21
7ª	6	2	2017-11-17 15:13:44
8ª	9	1	2017-11-17 15:39:25
9ª	3	0	0000-00-00 00:00:00
10ª	10	0	0000-00-00 00:00:00

Fonte: elaborado pelo autor, 2017.

O sistema foi desenvolvido sob licença Creative Commons Attribution-NonCommercial 4.0 International⁶ (CC BY-NC 4.0⁷). As dependências operacionais, necessárias para o funcionamento da ferramenta, estão disponíveis no Apêndice G.

3.4 CONSIDERAÇÕES DO CAPÍTULO

Este capítulo expôs os parâmetros gerais do desafio proposto e apresentou uma análise de competições, embasando a escolha de técnicas para o protótipo da ferramenta desenvolvida. Mostrou-se o conceito de composição de técnicas e foram apresentadas as composições possíveis entre as oito técnicas selecionadas para o protótipo. O funcionamento do protótipo foi explicado dividindo-o em gerador de problemas, *script* encarregado de gerar os desafios e configurar o SGBD, e aplicação *web*, responsável por promover a competição e a interação com o jogador.

Para validar o uso do TreasureHunt, foi projetado um experimento envolvendo competições no âmbito educacional, a partir do qual se obtiveram dados de percepção dos jogadores e de desempenho. Estes assuntos são abordados no próximo capítulo, que versa sobre a avaliação da pesquisa.

⁶ Licença que garante a liberdade para compartilhamento e adaptação do produto e impede seu uso para fins comerciais.

⁷ <<https://creativecommons.org/licenses/by-nc/4.0/>>

4 AVALIAÇÃO

Para investigar a aplicabilidade da geração automática de desafios proposta no Capítulo 3, foi realizado um estudo no qual alunos da Universidade do Estado de Santa Catarina (UDESC) e do Instituto Federal Catarinense (IFC) participaram de um desafio de Segurança com problemas gerados com o TreasureHunt. Este capítulo apresenta esse estudo. A Seção 4.1 descreve o projeto do experimento. A Seção 4.2 discorre sobre a execução das atividades. A Seção 4.3 apresenta a análise dos resultados obtidos e as observações efetuadas sobre o estudo. A Seção 4.4 apresenta a discussão dos resultados e a Seção 4.5 traz as considerações do capítulo.

4.1 PROJETO DE EXPERIMENTO

A proposta do experimento foi realizar a competição em ambientes acadêmicos e medir tanto a eficácia da ferramenta desenvolvida em criar problemas distintos sem trivializar suas soluções quanto a percepção de satisfação dos estudantes com a atividade, com métricas posteriormente detalhadas na Seção 4.3.1.

Nesta atividade, o professor, também chamado de organizador, deve utilizar a ferramenta desenvolvida para gerar os desafios e possibilitar acesso individual aos jogadores por meio da aplicação *web*. Para tanto, planejou-se o experimento de forma a ser realizado em três aulas consecutivas, assim definidas:

1. **Aula preparatória**¹: momento em que um conjunto básico de ferramentas é apresentado aos alunos. O objetivo é familiarizá-los com ferramentas potencialmente úteis para solucionar problemas de Segurança. Definições, exemplos de uso e exercícios fazem parte desta aula. Esse conjunto é suficiente para resolver todos os problemas, a despeito da possibilidade de resolvê-los com outras ferramentas não apresentadas neste momento.
2. **Competição 1**: primeiro contato dos jogadores com o TreasureHunt. Atividade individual, sem a intervenção do organizador e realizada em laboratórios de Informática, com as ferramentas indicadas na aula preparatória previamente instaladas. Consulta ao material da aula preparatória e à Internet são liberadas. O tempo previsto para a aplicação da atividade é de cerca de 1 hora e 30 minutos. Nesta competição, todos os jogadores recebem instâncias de problemas com as mesmas técnicas. A escolha dos problemas se deu a partir da matriz de compo-

¹ Os arquivos da aula estão no repositório do trabalho, disponíveis em <https://github.com/TreasureHuntGame/TreasureHunt>.

sições, disponível na Tabela 2 (página 54), de forma a utilizar todas as técnicas para as quais houve implementação no gerador de desafios. Por decisão do autor, a repetição de técnicas ficou limitada a dois problemas por competição, ou seja, uma técnica não estaria presente em mais do que dois exercícios para não trivializá-los com a repetição de comandos. Seis problemas foram elaborados, sendo dois individuais e quatro compostos. A decisão de mesclar exercícios individuais e compostos se deu para poder analisar a diferença de desempenho entre estes tipos de exercícios. O tempo previsto para a atividade foi determinante para a decisão sobre a quantidade de exercícios, assim cada exercício poderia ser resolvido, em média, em 15 minutos. Os exercícios desta competição foram:

- (Des)criptografia de Cifra de César ◦ Comentário em código-fonte de página HTML
- Descompilar binário e obter fonte Python
- (De)codificação de caractere ASCII para inteiro
- Comentário no arquivo `robots.txt` ◦ Esteganografia em Imagens
- Esteganografia em Imagens ◦ Esteganografia em Imagens
- (De)codificação de arquivo em `base64` ◦ Descompilar binário e obter fonte Java

3. **Competição 2:** atividade semelhante à Competição 1, porém, metade da turma resolve os mesmos exercícios (Grupo C2.1) constantes na competição anterior, com a ordem diferente, e a outra metade (Grupo C2.2) da turma resolve exercícios com as mesmas técnicas, mas com composições diferentes. A turma não deve ser avisada sobre a diferença dos exercícios, e consulta ao material da aula preparatória e à Internet são liberadas. O critério de divisão dos dois grupos é o desempenho na Competição 1, de forma a mantê-los equilibrados. Assim, tomando como base a colocação dos jogadores na Competição 1 e o número de acertos, propõe-se a criação de dois grupos em que o somatório de acertos de seus jogadores na Competição 1 seja igual ou o mais próximo possível disso. Em caso de quantidade ímpar de jogadores, o grupo C2.1 recebe um jogador a mais, em todas as turmas. O novo conjunto de exercícios foi escolhido pelos organizadores manualmente, utilizando como critério a complexidade dos problemas, procurando deixá-la semelhante a dos problemas da Competição 1, contendo também dois exercícios individuais e quatro compostos. Os exercícios desta competição para o Grupo C2.2 foram:

- (De)codificação de caractere ASCII para inteiro ◦ Comentário em código-fonte de página HTML

- (Des)criptografia de Cifra de César ◦ Comentário no arquivo `robots.txt`
- Descompilar binário e obter fonte Java
- (De)codificação de arquivo em `base64` ◦ Descompilar binário e obter fonte Python
- Esteganografia em Imagens
- (De)codificação de arquivo em `base64` ◦ Esteganografia em Imagens

Ao finalizar as competições, o servidor *web* é parado e os dados contidos no SGBD são coletados pela organização através de *database dump*², sendo restaurados em um computador central para análise de resultados.

A fim de obter os resultados qualitativos de percepção de satisfação dos jogadores, foram elaborados três questionários:

1. **Questionário de Levantamento de Perfil**³: identificando o perfil dos jogadores e seus conhecimentos prévios na área (disponível no Apêndice B);
2. **Questionário Pré-teste**⁴: identificando as impressões dos jogadores antes da competição (disponível no Apêndice C); e
3. **Questionário Pós-teste**⁵: identificando as impressões dos jogadores após a participação nas competições (disponível no Apêndice D).

Os questionários 1 e 2 foram preparados para serem disponibilizados no início da aula preparatória. O questionário 3 foi preparado para ser respondido ao final da segunda competição. Todos seguem uma escala de Likert (LIKERT, 1932) de cinco pontos.

Os questionários 2 e 3 serão comparados com base na escala de Likert, visando a identificar se há diferença na percepção dos estudantes sobre satisfação com a competição. A ideia de comparar as impressões dos jogadores antes e depois do jogo foi vista em trabalhos que avaliaram aspectos relacionados a jogos de Segurança (CHEUNG et al., 2012; OLANO et al., 2014; MIRKOVIC et al., 2015a; PETULLO et al., 2016). Algumas questões presentes nos três questionários elaborados foram inspiradas em trabalhos correlatos através dos quais também avaliou-se a impressão dos

² *Database dump* (despejo de banco de dados) consiste no armazenamento de tabelas e seus registros em arquivos para posterior restauração em outras máquinas. Estes arquivos são compostos por comandos SQL (*Structured Query Language*), que quando executados reproduzem uma base de dados equivalente à original.

³ Disponível em: <<http://bit.ly/2AdEWMW>>

⁴ Disponível em: <<http://bit.ly/2AoVkew>>

⁵ Disponível em: <<http://bit.ly/2lYORDL>>

jogadores para com jogos de Segurança (ANDRADE, 2012; CHEUNG et al., 2012; WEISS; MACHE; NILSEN, 2013; CHAPMAN; BURKET; BRUMLEY, 2014; MIRKOVIC et al., 2015b; VYKOPAL; BARTÁK, 2016; WEE; BASHIR; MEMON, 2016).

A avaliação foi inspirada no Modelo de Kirkpatrick (KIRKPATRICK, 1996). Este modelo de avaliação, utilizado em programas de treinamento e educacionais, é dividido em quatro passos (KIRKPATRICK, 1996):

1. **Reação:** diz respeito ao que os participantes sentem durante a atividade;
2. **Aprendizado:** refere-se a princípios, fatos e técnicas absorvidos pelos participantes;
3. **Comportamento:** avalia se houve mudança no comportamento dos participantes; e
4. **Resultados:** avalia o efeito do treinamento.

Dentre os quatro níveis definidos, o primeiro (reação) está situado na avaliação deste trabalho, pois os questionários foram aplicados visando a mensurar o sentimento de contentamento dos jogadores com a atividade, não obstante questões relativas à aprendizagem estejam relacionadas, mas não foram objeto de avaliação.

Para analisar a eficácia na aleatorização de problemas, os resultados de desempenho, armazenados no SGBD, foram analisados do ponto de vista estatístico (conforme será descrito na Seção 4.3.1). As métricas utilizadas foram: número de acertos, taxa de submissões corretas, tempo médio de conclusão, exercícios com maior e menor aproveitamento, técnicas com maior e menor aproveitamento e aproveitamento de problemas individuais e compostos.

4.2 EXECUÇÃO DA ATIVIDADE

As atividades propostas no projeto de experimento foram realizadas em três turmas:

- Turma formada majoritariamente por estudantes dos últimos semestres do curso de Bacharelado em Ciência da Computação da UDESC, *Campus Joinville* (aqui chamada de UDESC), na disciplina de Segurança de Redes de Computadores, contendo 13 estudantes participantes;
- Turma do sexto semestre do curso de Tecnologia em Análise e Desenvolvimento de Sistemas do IFC, *Campus Blumenau* (aqui chamada de TADS), contendo 10 estudantes participantes; e

- Turma de Segurança Computacional do Curso de Qualificação Profissional em Configuração de Servidores Cisco (aqui chamada de FIC) do IFC *Campus* Blumenau, contendo 7 estudantes participantes.

Para todas as turmas, o conjunto de exercícios das duas competições foi o mesmo. As duas instituições dispõem de laboratórios de Informática equipados com as ferramentas necessárias e sistemas *Unix-like*. Na UDESC, o sistema instalado é o Ubuntu, e no IFC o sistema instalado é o Arch Linux.

As atividades foram realizadas pelo autor e pelo orientador deste trabalho, na condição de organizadores, durante o mês de novembro de 2017.

No início da aula preparatória, os estudantes foram convidados a ler e, em caso de aceitação, assinar o TCLE. Todos os estudantes, em todas as turmas nas quais o experimento foi realizado, aceitaram participar da atividade. As etapas seguintes consistiram na resposta aos questionários de levantamento de perfil do jogador e de pré-teste, para então iniciar a aula preparatória.

A aula preparatória foi realizada conforme planejado, com apresentação da definição de um conjunto de ferramentas, seguida por exemplos práticos de uso, exercícios e correção destes. O material da aula foi disponibilizado para consulta. n A aula seguinte consistiu na primeira competição. No início desta aula foi explicado o funcionamento da competição, foram distribuídos os IDs dos jogadores e o endereço de acesso à aplicação *web*, para então ser realizada a competição. À medida em que um jogador concluía a atividade, este era dispensado da aula e podia sair do recinto.

A terceira e última aula da atividade consistiu na segunda competição. Novamente a competição foi explicada, sendo distribuídos IDs dos jogadores e o endereço de acesso à aplicação *web*. Nesta atividade os IDs dos jogadores eram diferentes daqueles da primeira competição. Mais uma vez, o jogador podia se ausentar do laboratório após concluir a atividade.

4.3 ANÁLISE DE RESULTADOS

Os resultados foram analisados a partir de duas fontes: questionários respondidos pelos jogadores e submissões realizadas através da aplicação *web* e armazenadas no SGBD. A Seção 4.3.1 apresenta o planejamento da análise estatística. A Seção 4.3.2 mostra os resultados do questionário de levantamento de perfil dos jogadores. A Seção 4.3.3 discute os resultados de desempenho dos jogadores. A Seção 4.3.4 apresenta os resultados dos questionários pré e pós-teste. Ao final da seção, na Subseção 4.3.5 (Observações), constam relatos dos professores observadores (organizadores) da atividade.

4.3.1 Planejamento da Análise Estatística

A análise foi baseada em medidas de desempenho extraídas das respostas submetidas nas competições e nas respostas dos questionários. A análise do desempenho mensurou acertos, tempo de resolução e a dificuldade dos problemas propostos e das técnicas utilizadas. Para comparar o desempenho entre competições foram usados dois testes estatísticos não-paramétricos, o teste da soma dos postos de Wilcoxon (*Wilcoxon rank sum test*) e o teste de postos sinalizados de Wilcoxon (*Wilcoxon signed rank test*), que é equivalente ao teste *U* de Mann-Whitney (FIELD; MILES; FIELD, 2012). O primeiro é um teste para amostras não pareadas, que compara grupos, e foi usado para comparar os dados de C2.1 e C2.2, uma vez que não há interseção entre esses grupos. O segundo é um teste para amostras pareadas, que compara a evolução de indivíduos, e foi usado para comparar as competições C1 e C2.

A análise dos questionários mensurou a receptividade dos alunos às competições (aspectos de satisfação). Também foi avaliada a existência de diferenças estatisticamente significativas entre as respostas dos questionários pré e pós-competição, o que evidenciaria uma mudança de percepção induzida pela participação nos desafios. Para minimizar o viés nas respostas, optou-se por deixar os questionários anônimos (isto é, sem a identificação dos respondentes), o que inviabiliza a análise da evolução dos indivíduos. Assim, os grupos de respostas foram considerados como amostras independentes, e para essa análise foi usado o teste da soma dos postos de Wilcoxon (para amostras não pareadas), que é mais conservador que o teste para amostras pareadas: para dados idênticos, o teste não pareado produz um valor-*p* maior do que o teste pareado, o que diminui a probabilidade de identificar erroneamente uma diferença inexistente (erro tipo I) (FIELD; MILES; FIELD, 2012).

A escolha por testes não paramétricos se justifica pelo fato dos dados não atenderem a uma premissa do teste *t* de Student, que exige que as variáveis sejam normalmente distribuídas. A premissa foi verificada usando o teste de Shapiro-Wilk (FIELD; MILES; FIELD, 2012). Em todos os testes estatísticos foi adotado como nível de significância $\alpha = 0,05$.

Nos questionários, para medir a consistência interna das questões em mensurar cada um desses aspectos, foi efetuada uma análise de confiabilidade, usando o coeficiente α de Cronbach (FIELD; MILES; FIELD, 2012). Os valores obtidos para o α de Cronbach foram interpretados com base na Tabela 3.

A Tabela 4 resume as técnicas estatísticas usadas na avaliação dos resultados. O pacote estatístico R (R CORE TEAM, 2018) foi usado na análise dos dados e na condução dos testes estatísticos.

Tabela 3 – Consistência interna do questionário segundo o valor de alfa.

Valor de alfa	Consistência interna
Maior do que 0,80	Quase perfeito
De 0,80 a 0,61	Substancial
De 0,60 a 0,41	Moderado
De 0,40 a 0,21	Razoável
Menor do que 0,21	Pequeno

Fonte: (LANDIS; KOCH, 1977).

Tabela 4 – Técnicas estatísticas usadas na avaliação dos resultados.

Seção	Objeto da análise	Técnica estatística
4.3.3.1	acertos em C1 x C2	Wilcoxon, amostras pareadas
	acertos em C2.1 x C2.2	Wilcoxon, amostras não pareadas
4.3.3.3	tempo de conclusão em C1 x C2	Wilcoxon, amostras pareadas
	tempo de conclusão em C2.1 x C2.2	Wilcoxon, amostras não pareadas
4.3.4	evolução das respostas (pré-pós) confiabilidade (Tabelas 3 e 7)	Wilcoxon, amostras não pareadas coeficiente alfa de Cronbach

Fonte: elaborado pelo autor, 2018.

4.3.2 Resultados do Questionário de Levantamento de Perfil

O questionário de levantamento de perfil foi aplicado nas três turmas antes da aula preparatória, com o objetivo de identificar o perfil dos jogadores. Trinta estudantes responderam ao questionário, sendo 10 da turma TADS, 7 da turma FIC e 13 da turma UDESC. Os resultados, descritos a seguir, estão resumidos na Figura 17.

Considerando todas as turmas, a média de idade dos participantes foi de 22,97 anos e o desvio padrão de 5,61 anos. Na turma TADS a média de idade foi de 25,11 anos e o desvio padrão 5,23. Na turma UDESC a média foi 22,31 anos e o desvio padrão 2,84 anos. Na turma FIC, 21,43 anos e desvio padrão de 9,14 anos. Embora constem 30 respostas, este cálculo desconsidera uma resposta em que um estudante respondeu ter 99 anos de idade. A maioria dos estudantes (28 de 30) é do sexo masculino. Nas turmas FIC e UDESC, 100% dos jogadores eram do sexo masculino, ao passo que, na turma TADS, 20% dos participantes eram do sexo feminino, o que representa duas alunas. 76,67% dos estudantes cursaram os ensinos médio e fundamental em escola pública, e 36,67% em escola privada. Os números excedem 100% porque o estudante pode ter frequentado escolas públicas e privadas durante sua vida acadêmica, portanto, podia assinalar as duas respostas. Dentre os estudantes, três (10% do total) possuem formação em curso superior, sendo dois da turma FIC, representando 28,57% do total de alunos desta turma, e um da turma TADS, representando 10% do total de alunos desta turma.

Sobre a experiência com o sistema operacional Linux, 43,33% dos estudantes afirmaram ter familiaridade moderada, com razoável conforto na linha de comando e

criação de *scripts* simples. 36,67% responderam ser um pouco familiar, 13,33% levemente familiar e 6,67% extremamente familiar. Ninguém respondeu ser nada familiar. Esta pergunta serviu como indício de validação da adequação dos exercícios em relação aos conhecimentos prévios dos estudantes sobre Linux, ratificando a ideia de que os jogadores não possuíam conhecimentos avançados no sistema, mas teriam condições de resolver as tarefas propostas. Além disso, todas as ferramentas apresentadas para resolver desafios de Segurança na aula de preparação são ferramentas de linha de comando para sistemas Linux.

Vinte estudantes, o que representa 66,67%, afirmaram não ter experiência com Segurança Computacional. Seis estudantes (20%) afirmaram ter estudado por conta própria, quatro (13,33%) afirmaram estar fazendo a disciplina novamente e um (3,33%) fez um curso fora da instituição. Com base nas respostas, a premissa de que o público-alvo não era experiente em Segurança foi confirmada e também serviu para validar a adequação dos exercícios.

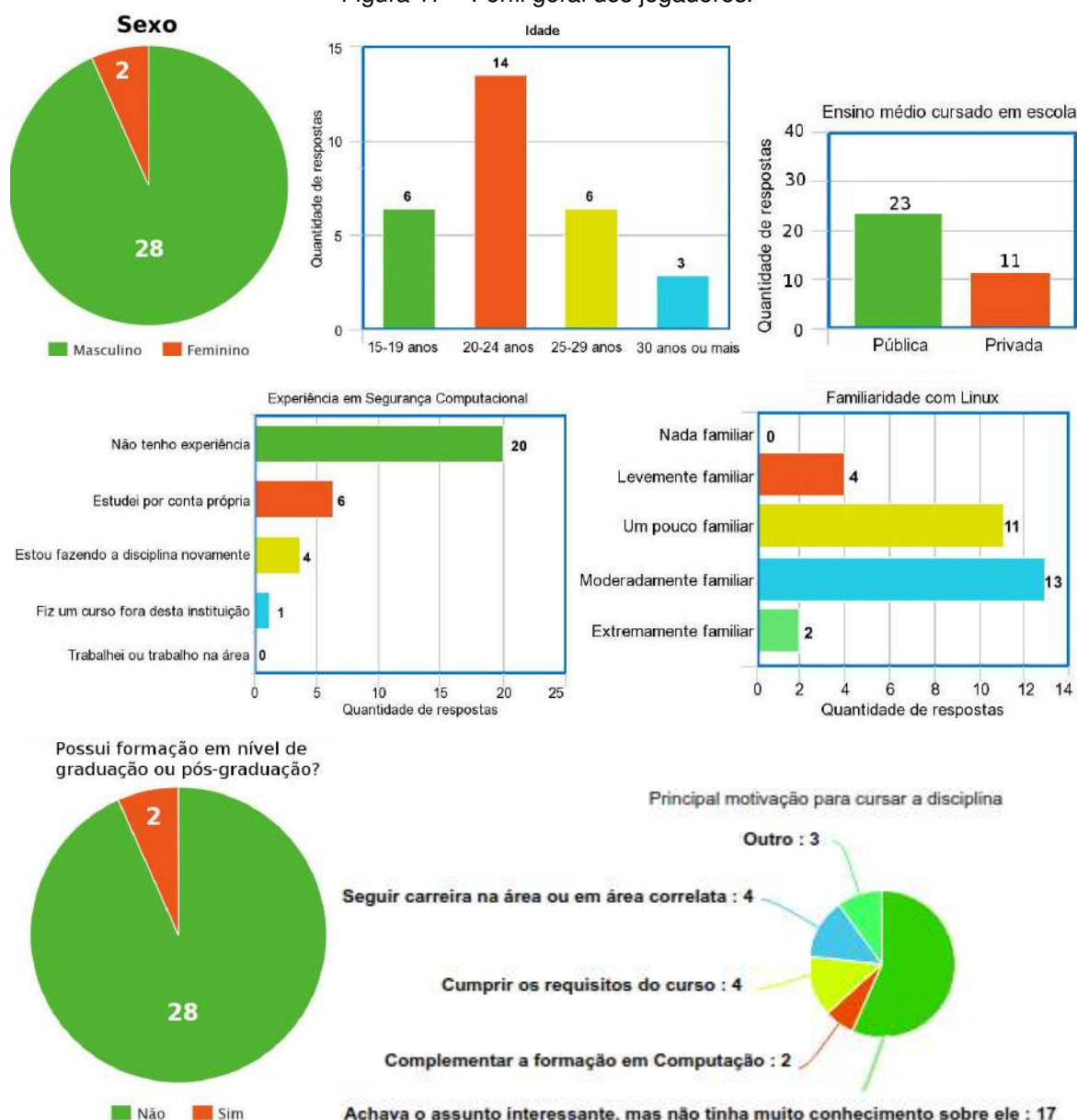
No que diz respeito à motivação para cursar uma disciplina de Segurança, 56,67% responderam que achavam o assunto interessante, mas não tinham muito conhecimento, 6,67% pretendiam complementar a formação em Computação, 13,33% faziam prioritariamente para cumprir os requisitos do curso, 13,33% pretendiam seguir carreira em Segurança ou em uma área correlata e 10% forneceram outras respostas. Esta resposta acompanha a anterior no que diz respeito ao conhecimento limitado em Segurança, e mostra que ao menos 70% dos alunos envolvidos tinham interesse na área, seja para ampliar seus conhecimentos ou pensando em uma carreira em Segurança. É possível que os demais participantes também se interessem pela área, mas de forma secundária ao complemento da formação e aos requisitos do curso.

4.3.3 Resultados de Desempenho

4.3.3.1 Número de Acertos

Por meio do SGBD foi possível obter o desempenho dos jogadores em termos de número de acertos, em cada uma das turmas, em todas as competições. A Figura 18 apresenta o número de acertos das três turmas nas duas competições, por meio de *boxplots*. A Figura 18 mostra que a mediana na turma FIC ficou em 3,00 pontos, da turma TADS ficou em 4,50 pontos e da turma UDESC ficou em 6,00 pontos. Em todas as turmas houve registros de estudantes obtendo a pontuação máxima, ao passo que houve registro de um jogador obtendo zero acertos em cada uma das turmas TADS e UDESC. O resultado geral mostra que 75% dos alunos acertaram metade ou mais dos problemas, e que 25% acertaram todos. Os dados da Figura 18 permitem concluir que a complexidade dos exercícios foi adequada para as turmas FIC e TADS. Na turma da UDESC a complexidade foi menos adequada, a julgar pela quantidade

Figura 17 – Perfil geral dos jogadores.



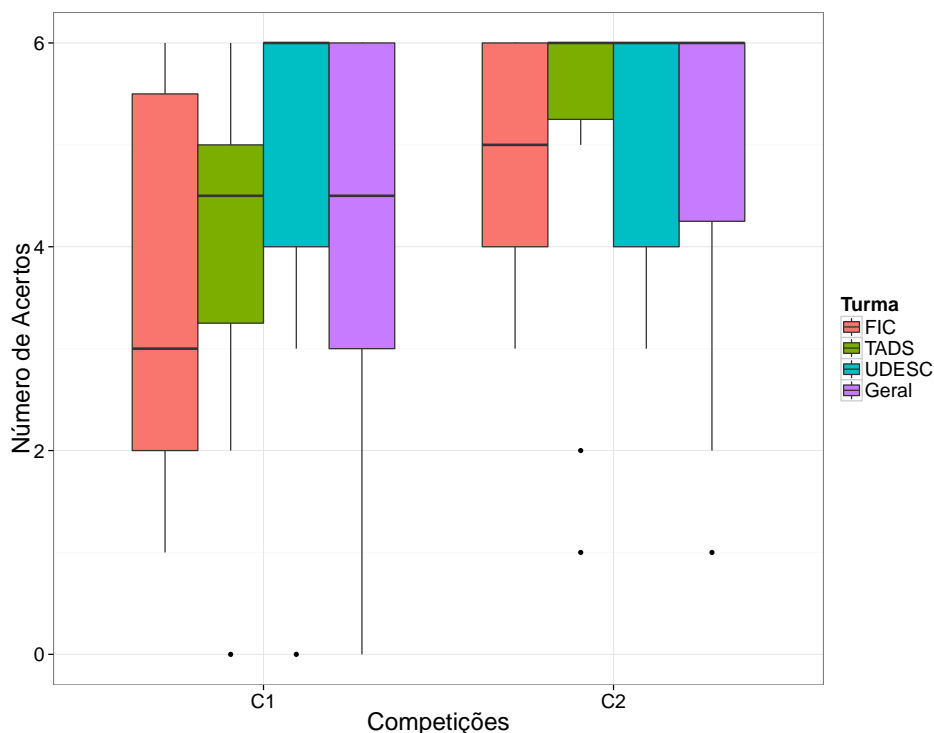
Fonte: elaborado pelo autor, 2018.

de jogadores (sete) que obtiveram o máximo de acertos já na primeira competição. A Figura 18 também torna possível perceber que, da C1 para a C2, a mediana da turma FIC aumentou de 3,00 pontos para 5,00 pontos e da turma TADS aumentou de 4,50 para 6,00 pontos, enquanto a da turma UDESC manteve-se em 6,00 pontos. Com base nos resultados gerais, percebe-se que a mediana ficou em 6,00 pontos. Ao todo, 60% (18 de 30) dos estudantes atingiram a pontuação máxima. O primeiro quartil ficou em 4,00 pontos e o valor mínimo, nesta competição, foi de 1,00 ponto. A Figura 18 mostra que o desempenho dos estudantes melhorou e, para a maioria, foi fácil resolver os exercícios.

O teste de Wilcoxon para amostras pareadas identificou diferença estatística-

mente significativa entre a quantidade de acertos nas Competições 1 e 2 ($V = 4,5$, valor- $p = 0,00093 < 0,05$). Tem-se, portanto, que o desempenho na C2 foi diferente, e superior, se comparado ao desempenho dos jogadores na C1.

Figura 18 – *Boxplots* de número de acertos nas competições 1 e 2.



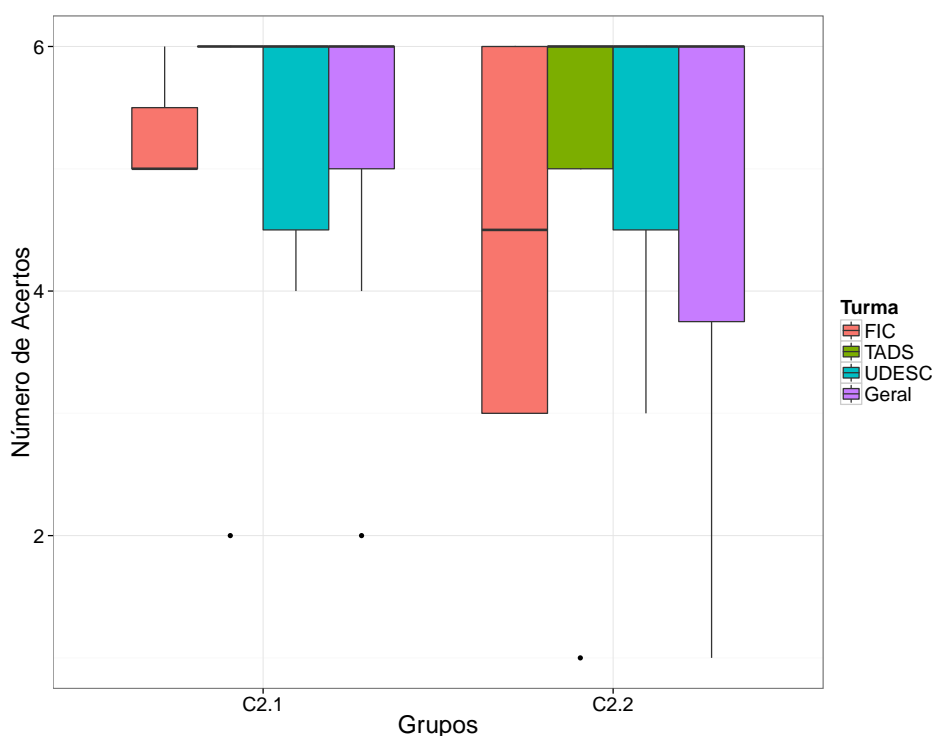
Fonte: elaborado pelo autor, 2018.

Os resultados da Figura 19 mostram a comparação de acertos dos dois grupos da Competição 2 (C2.1 e C2.2), em todas as turmas. Nota-se que tanto no grupo C2.1 quanto no C2.2 a maioria dos jogadores obteve a quantidade máxima de acertos, pois a mediana em todos os casos, com exceção da turma FIC, ficou em 6,00 pontos. O grupo C2.2 apresentou maior variabilidade no número de acertos e uma quantidade maior de escores baixos, mas, de modo geral, o desempenho de todas as turmas foi considerado alto. Aplicando o teste de Wilcoxon para amostras não pareadas em C2.1 e C2.2, o resultado obtido não indica diferença estatisticamente significativa ($W = 125,5$, valor- $p = 0,54 > 0,05$). Portanto, considerando que o desempenho dos alunos na Competição 2 foi melhor do que na Competição 1, a dificuldade diminuiu no segundo campeonato independentemente do grupo (C2.1 ou C2.2), e não é possível afirmar que o desempenho entre os grupos da Competição 2 foi diferente.

4.3.3.2 Taxa de Submissões Corretas

A taxa de submissões corretas por turma foi observada através de uma tabela do SGBD que armazenou todas as submissões realizadas no sistema, e foi calculada dividindo o total de acertos pelo total de submissões realizadas. Enquanto na Sub-

Figura 19 – Comparação de acertos entre os grupos C2.1 e C2.2 em todas as turmas.



Fonte: elaborado pelo autor, 2018.

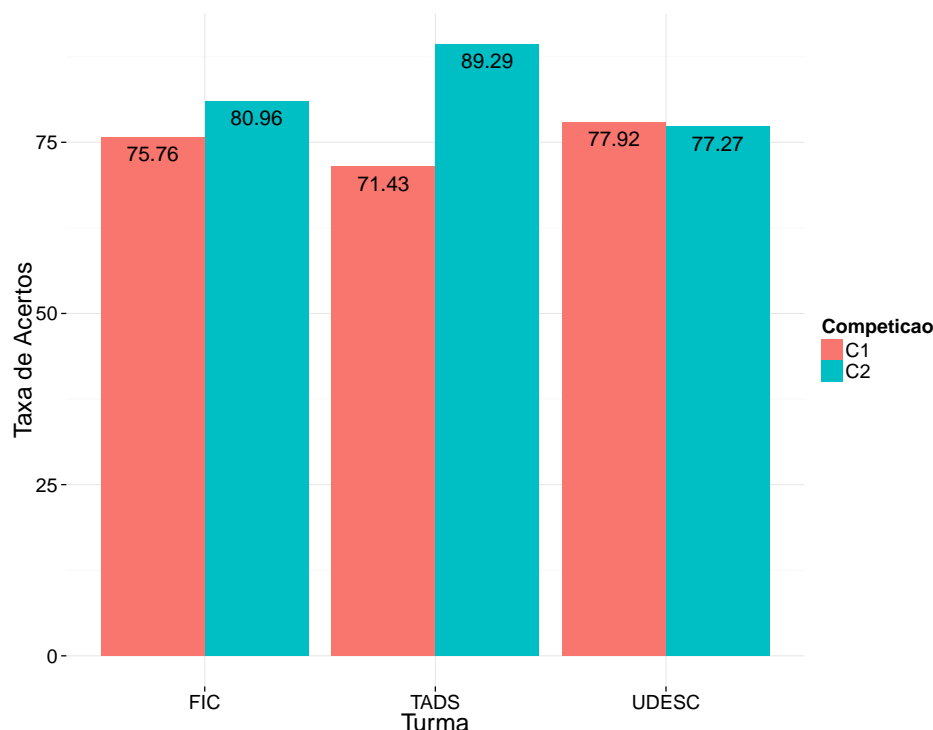
seção 4.3.3.1 examinou-se o índice de acertos obtido pelos alunos, nesta seção é analisado o índice de submissões corretas. Esses índices podem ser diferentes porque os jogadores podem efetuar várias tentativas até chegar à resposta correta, o que contribui na revelação de problemas de comunicação, na transmissão das regras do jogo, problemas no sistema, compartilhamento de *flags* e tentativas de força bruta. A Figura 20 mostra a taxa de submissões corretas (em %) em todas as turmas, nas duas competições.

A média geral na Competição 1 foi de 75,25%, e na Competição 2 foi de 82,14%. Da tabela de submissões no SGBD e da Figura 20 é possível perceber que, com exceção da turma UDESC, a taxa de submissões corretas foi maior na segunda competição. Ainda analisando a tabela, percebeu-se que a maior parte dos erros ocorreu por inserção de espaços em branco nas *flags* ou por informar o identificador errado do problema, e foi possível perceber que nenhum estudante enviou *flags* já enviadas por outros colegas, o que significa que não houve compartilhamento de *flags*. Com relação à queda da taxa na turma UDESC, esta ocorreu em função de um estudante enviar diversas respostas aleatórias na tentativa de adivinhar a palavra secreta de um problema.

Os resultados indicam que a comunicação aos estudantes quanto às regras do jogo e ao formato da palavra secreta surtiram efeito. Maior ênfase em cuidados ao copiar ou digitar espaços em branco entre a *flag* informada e atenção ao digitar

corretamente o número identificador dos problemas são ações que podem resultar em taxas maiores em competições futuras.

Figura 20 – Taxa de submissões corretas (em %) das três turmas nas duas competições.



Fonte: elaborado pelo autor, 2018.

4.3.3.3 Tempo Médio de Conclusão

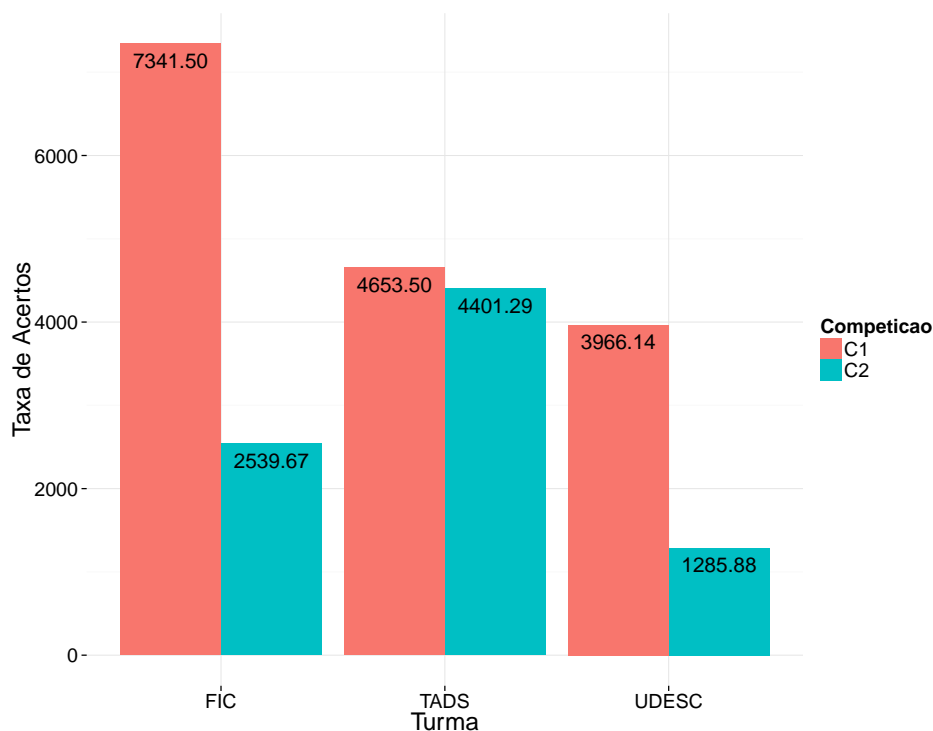
A Figura 21 mostra o tempo médio para conclusão da atividade entre os alunos que conseguiram obter 100% dos acertos. Para isso, foram considerados o horário de início da competição e o horário da última submissão correta de cada jogador. Nas turmas TADS e FIC, dois alunos gabaritaram a Competição 1; na turma UDESC, sete. Na Competição 2, sete alunos gabaritaram na turma TADS, três na turma FIC e oito na UDESC.

Da Figura 21 é possível perceber que o tempo médio para concluir a atividade baixou em termos absolutos, e a quantidade de alunos que conseguiu acertar 100% dos exercícios aumentou nas três turmas. No entanto, é necessário ponderar que o tempo é afetado por vários fatores, tais como atraso dos estudantes e eventuais erros em ferramentas ou no servidor. O tempo médio geral da Competição 1 ficou em 3383,35 segundos (56,4 minutos). O tempo médio geral da Competição 2 ficou em 2706,39 segundos (45,1 minutos), uma redução de 11,3 minutos em média.

O teste de Wilcoxon para amostras pareadas mostra que houve diferença significativa entre os tempos de conclusão das Competições 1 e 2 ($V = 381$, valor- $p = 0,0016 < 0,05$). Por outro lado, o teste de Wilcoxon para amostras não pareadas

mostra que a diferença entre os grupos C2.1 e C2.2 não é estatisticamente significativa ($W = 108$, valor- $p = 0,89 > 0,05$). Este é mais um indício de que os exercícios ficaram mais fáceis após a aplicação da primeira competição, mas inconclusivos sobre a diferença de tempo para resolução de C2.1 e C2.2.

Figura 21 – Tempo médio (em s) para conclusão da atividade nas três turmas nas duas competições.



Fonte: elaborado pelo autor, 2018.

4.3.3.4 Aproveitamento por Problema

Avaliou-se, do conjunto de seis problemas nas duas competições, aqueles para os quais se obteve mais e menos acertos. A seguir estão listados, nas Tabelas 5, 6 e 7, os exercícios de menor aproveitamento na Competição 1, na Competição 2 para o grupo C2.1 e para o grupo C2.2, respectivamente, utilizando as mesmas abreviações constantes na Tabela 2 (página 54).

Para a Competição 2, em qualquer caso, sempre foi mais fácil lidar com problemas não compostos com base nos resultados. Os problemas individuais de descompilação de código Java e Python estiveram entre os de maior aproveitamento em todos os jogos. O exercício de maior aproveitamento na C1 foi *Descompilar binário e obter fonte Python*, com 93,33% de acertos dos problemas. Este exercício também foi o de maior aproveitamento na C2.1, junto com *César o HTML*, com 100,00% de acertos. Na C2.2, o exercício Java foi o de maior aproveitamento, com 100,00% de acertos. As justificativas para a facilidade com os problemas Java e Python podem estar relacionadas à quantidade de soluções possíveis com o conjunto de ferramentas apresentado

Tabela 5 – Exercícios com mais acertos na C1.

Turma	Problema(s)	Acertos	%
TADS	<ul style="list-style-type: none"> • César ○ HTML • Python 	9	90,00
FIC	<ul style="list-style-type: none"> • Python 	7	100,00
UDESC	<ul style="list-style-type: none"> • César ○ HTML • Python 	12	92,31
Geral	<ul style="list-style-type: none"> • Python 	28	93,33

Fonte: elaborado pelo autor, 2017.

Tabela 6 – Exercícios com mais acertos na C2.1.

Turma	Problema(s)	Acertos	%
TADS	<ul style="list-style-type: none"> • César ○ HTML • Python 	5	100,00
FIC	<ul style="list-style-type: none"> • A2I • Base64 ○ Java • César ○ HTML • Python • Robots ○ Esteg 	3	100,00
UDESC	<ul style="list-style-type: none"> • César ○ HTML • Python 	6	100,00
Geral	<ul style="list-style-type: none"> • César ○ HTML • Python 	14	100,00

Fonte: elaborado pelo autor, 2017.

na aula preparatória (por exemplo, com `strings` e `cat`). O alto aproveitamento na resolução do problema *César ○ HTML* pode ter ocorrido porque os estudantes conhecem a estrutura de uma página HTML, de forma a perceber que palavras entre os sinais de “<! – –” e “– –>” representariam comentários entre as *tags*. O comentário contém os símbolos “{“ e “}”, o que permite ao jogador inferir que se trata da *flag* alterada com Cifra de César.

Tabela 7 – Exercícios com mais acertos na C2.2.

Turma	Problema(s)	Acertos	%
TADS	• Java	5	90,00
FIC	• Java	4	100,00
UDESC	• Esteg • Java	7	100,00
Geral	• Java	16	100,00

Fonte: elaborado pelo autor, 2017.

De forma análoga, os problemas para os quais os jogadores obtiveram menor aproveitamento em termos de acertos também foram identificados. A seguir estão listados, nas Tabelas 8, 9 e 10, os exercícios de menor aproveitamento na Competição 1, na Competição 2 para o grupo C2.1 e para o grupo C2.2, respectivamente, utilizando as mesmas abreviações constantes na Tabela 2 (página 54).

Tabela 8 – Exercícios com menos acertos na C1.

Turma	Problema(s)	Acertos	%
TADS	• Robots ◦ Esteg	2	20,00
FIC	• Esteg ◦ Esteg	2	28,57
UDESC	• Esteg ◦ Esteg • Robots ◦ Esteg	7	53,85
Geral	• Robots ◦ Esteg	12	40,00

Fonte: elaborado pelo autor, 2017.

A Tabela 8 mostra que *Robots ◦ Esteg* foi o exercício composto de menor aproveitamento, obtendo apenas 12 acertos nas três turmas, dentre os 30 participantes. Exercícios com o problema *Comentário no arquivo robots.txt* composto com *Esteganografia em Imagens* são complexos porque o jogador recebe um conjunto de arquivos, e não um arquivo, como ocorre em todos os outros problemas implementados no protótipo. Assim, antes de procurar a *flag* propriamente, deve desconfiar do arquivo em que ela está e identificar que o arquivo *robots.txt* se trata de uma imagem. Portanto, é necessário encontrar o arquivo e ainda utilizar o *outguess* para extrair a *flag* da imagem.

Na Competição 2, os exercícios para os quais o grupo C2.1 obteve menos

acertos estão apresentados na Tabela 9. Na $C2$, o grupo $C2.1$ teve mais dificuldade em resolver o problema *Esteg* \circ *Esteg*, que obteve 9 acertos do total de 14 participantes. A dificuldade com o problema *Esteganografia em Imagens* composto com ele mesmo pode estar relacionada com as várias ferramentas possíveis para encontrar textos em imagens (por exemplo, `cat` e `strings`), com tentativas de uso do `outguess` utilizando o parâmetro k , para quando a imagem é esteganografada com senha ou a falta de percepção de que o arquivo extraído trata-se de uma imagem. Portanto, há ainda uma etapa intermediária após realizar a primeira extração, que consiste em verificar o tipo do arquivo esteganografado. Ao não realizar isso ou assumir (equivocadamente) que se trata de um arquivo de texto, o jogador pode encontrar dificuldades para solucionar o exercício.

Tabela 9 – Exercícios com menos acertos na $C2.1$.

Turma	Problema(s)	Acertos	%
TADS	<ul style="list-style-type: none"> • A2I • Base64 \circ Java • Esteg \circ Esteg • Robots \circ Esteg 	4	80,00
FIC	<ul style="list-style-type: none"> • Esteg \circ Esteg 	1	25,00
UDESC	<ul style="list-style-type: none"> • Esteg \circ Esteg • Robots \circ Esteg 	4	66,67
Geral	<ul style="list-style-type: none"> • Esteg \circ Esteg 	9	64,29

Fonte: elaborado pelo autor, 2017.

A Tabela 10 contém os exercícios para os quais o grupo $C2.2$ obteve menos acertos. O Grupo $C2.2$, que recebeu desafios distintos daqueles entregues na $C1$, obteve menos acertos (10 de 16) no exercício composto *César* \circ *Robots*. Acredita-se que a maior dificuldade para resolver este exercício esteve em identificar, dentre o conjunto de arquivos recebidos, que a *flag* estaria no arquivo `robots.txt`, pelos já citados motivos que tornam os exercícios com esta técnica mais desafiadores.

4.3.3.5 Taxa de Acertos por Tipo de Problema (Simples/Composto)

A taxa de acertos dos problemas simples, nas duas competições e nas três turmas, está apresentada na Tabela 11. Nela constam o número de acertos obtidos, o total de questões e o valor percentual de acertos. Observando a Tabela 11, pode-se

Tabela 10 – Exercícios com menos acertos na C2.2.

Turma	Problema(s)	Acertos	%
TADS	• César ○ Robots	3	60,00
FIC	• César ○ Robots	2	50,00
UDESC	• A2I ○ HTML • César ○ Robots	5	71,43
Geral	• César ○ Robots	10	62,50

Fonte: elaborado pelo autor, 2017.

afirmar que na Competição 2 houve aumento na taxa de acertos de exercícios simples, tanto para C2.1 quanto para C2.2.

Tabela 11 – Taxa de acertos por problemas simples.

C1				C2.1				C2.2			
TADS	FIC	UDESC	Total	TADS	FIC	UDESC	Total	TADS	FIC	UDESC	Total
17/20 (85%)	11/14 (78,57%)	23/26 (88,46%)	51/60 (85%)	9/10 (90%)	6/6 (100%)	12/12 (100%)	27/28 (96,43%)	9/10 (90%)	7/8 (87,5%)	14/14 (100%)	30/32 (93,75%)

Fonte: elaborado pelo autor, 2017.

A taxa de acertos dos problemas compostos, nas duas competições e nas três turmas, está apresentada na Tabela 12. Nela também constam o número de acertos obtidos, o total de questões e o valor percentual de acertos. A Tabela 12 mostra que, novamente, na Competição 2 houve aumento na taxa de acertos de exercícios, tanto para C2.1 quanto para C2.2. Em C2.1, o aumento foi de 8,93% em relação a C2.2. Além disso, comparando as Tabelas 9 e 10 é possível perceber que o percentual de acertos foi maior nos exercícios simples em relação aos compostos em todas as turmas e competições, o que já era esperado. Exercícios simples exigem, em geral, o uso de apenas uma ferramenta para obter a solução, e são menos complexos que os exercícios compostos. Embora neste trabalho os exercícios sejam compostos com apenas dois níveis, a aplicação de uma técnica a mais traz maior complexidade e pelo menos uma ferramenta a mais é necessária para se obter a solução de um problema,.

Tabela 12 – Taxa de acertos por problemas compostos.

C1				C2.1				C2.2			
TADS	FIC	UDESC	Total	TADS	FIC	UDESC	Total	TADS	FIC	UDESC	Total
23/40 (57,5%)	14/28 (50%)	37/52 (71,15%)	74/120 (61,67%)	17/20 (85%)	10/12 (83,33%)	20/24 (83,33%)	47/56 (83,93%)	15/20 (75%)	11/16 (68,75%)	22/28 (78,57%)	48/64 (75%)

Fonte: elaborado pelo autor, 2017.

4.3.3.6 Aproveitamento por Técnica

Além de avaliar os problemas mais fáceis e difíceis, isolando as técnicas é possível identificar aquelas que impuseram maior e menor dificuldade aos jogadores.

No geral, a técnica mais difícil foi *Comentário no arquivo robots.txt*, com 55% dos exercícios resolvidos. Nas turmas TADS e UDESC também foi a técnica *Comentário no arquivo robots.txt*, com 45% e 61,54% dos exercícios resolvidos. Na turma FIC a técnica mais difícil foi *Esteganografia em Imagens*, com 53,57% dos exercícios resolvidos. Como já explicado na Seção 4.3.3.4, exercícios com o problema *Comentário no arquivo robots.txt* são complexos em virtude do recebimento de um conjunto de arquivos, e não de um único arquivo, como ocorre com os outros problemas implementados. Assim, antes de procurar a *flag*, o jogador deve encontrar o arquivo em que ela está ou implementar uma solução via *script* que procure pelo padrão da *flag* em todos os arquivos do diretório. Ainda assim, o problema pode ser composto, o que exigiria procurar pelo padrão da *flag* em outros formatos (por exemplo, codificada em `base64`). A dificuldade com o problema *Esteganografia em Imagens* novamente pode estar relacionada com as várias ferramentas possíveis para encontrar textos em imagens (por exemplo, `cat` e `strings`) e com tentativas de uso do `outguess` utilizando senha (parâmetro *k*).

A técnica mais fácil no geral foi *Descompilar binário e obter fonte Python*, com 91,67% de taxa de acertos dos problemas. Em todas as turmas esta técnica também foi a mais fácil, com 90% de acertos na turma TADS, 92,86% na turma FIC e 92,31% na turma UDESC. Na turma UDESC, *Descompilar binário e obter fonte Java* também atingiu 92,31% de acertos. As justificativas para isso podem estar relacionadas à quantidade de soluções possíveis com o conjunto de ferramentas apresentado na aula preparatória (por exemplo, com `strings` e `cat`) e à aplicação individual do problema *Descompilar binário e obter fonte Python* nas competições C1 e C2.1, que exigia a aplicação de menos ferramentas para obter a *flag*.

A tabela completa contendo o aproveitamento por técnica está disponível no Apêndice F.

4.3.4 Resultados dos Questionários Pré e Pós-Teste

Os questionários pré e pós-teste, aplicados antes e depois da competição, respectivamente, ajudam a avaliar e mensurar a percepção dos jogadores acerca de satisfação com a atividade. Ambos foram aplicados nas três turmas.

O questionário pré-teste foi respondido por 30 estudantes, sendo 10 da turma TADS, 7 da turma FIC e 13 da turma UDESC. O questionário pós-teste foi respondido por 29 estudantes, sendo 10 da turma TADS, 7 da turma FIC e 12 da turma UDESC.

Visando a identificar a satisfação dos jogadores com a atividade, foram observadas, em especial, as questões 1.1, 1.2, 1.3, 1.5 e 1.7 do pré e do pós-teste. Essas questões usam uma escala de Likert de cinco pontos, com respostas indo de *Discordo fortemente* (1) a *Concordo fortemente* (5). Valores superiores a 3 indicam satisfação positiva, com a pontuação máxima sendo igual a 5.

A comparação entre os questionários pré e pós-teste pode ser usada para analisar se houve mudança positiva nos resultados, o que seria um indicativo de que a competição teve êxito em tornar a ideia de competição pedagógica mais clara para o aluno, bem como aumentou sua motivação. A Tabela 13 exhibe os resultados das questões mencionadas e a Tabela 14 apresenta as questões e seus respectivos identificadores.

Observando a Tabela 13, todos os resultados da coluna *Geral* ficaram acima de 4,00, o que indica que, na média, os jogadores estiveram satisfeitos com a competição aplicada. A coluna *Evolução* contém a diferença entre as médias gerais do pós e do pré teste (pós - pré), e a coluna *Significativa?* é determinada pelo teste de Wilcoxon para amostras não pareadas e interpreta se a diferença entre os resultados do pré e do pós-teste é estatisticamente significativa (valor- $p \leq 0,05$).

É possível perceber, comparando os resultados da Tabela 13, que todos os resultados do pós-teste superaram os resultados das mesmas questões do pré-teste. Houve evolução, em números absolutos, considerando os resultados separadamente por turma, com exceção da questão 1.3 (*Tenho interesse em atividades práticas envolvendo Segurança Computacional*) na turma TADS, para a qual houve queda de 0,10 ponto, e da questão 1.5 (*Exercícios práticos de Segurança Computacional aumentam o entendimento sobre esta área*) na turma UDESC, para a qual houve queda de 0,02 ponto. Há que se considerar também que nesta turma houve uma resposta a menos no questionário pós-teste em relação ao pré-teste.

Para mensurar a significância estatística das diferenças, aplicou-se o teste de Wilcoxon para amostras não pareadas, e constatou-se que para as questões 1.2, 1.3 e 1.5 a diferença encontrada não foi estatisticamente significativa (valor- p superior a 0,05), mas foi para as questões 1.1 e 1.7. É importante ressaltar que as médias para todas as questões já eram altas (superiores a 4) no questionário pré-teste, o que limita a possibilidade de evolução (uma vez que a pontuação máxima de cada questão é 5). Ainda assim, os resultados indicam que a participação nas competições aumentou a motivação associada a jogos como instrumento de aprendizagem e a percepção de que jogos podem despertar a atenção do público em geral para Segurança Computacional.

A análise de confiabilidade referente às questões 1.1, 1.2, 1.3, 1.5 e 1.7 teve

um alfa de Cronbach de 0,78. Segundo a Tabela 3 (página 71), isso indica confiabilidade substancial, próxima do limiar para confiabilidade quase perfeita (0,8). Nessa análise foram consideradas apenas as respostas do questionário pós-teste.

Tabela 13 – Resultados das questões sobre satisfação.

Questão	UDESC		TADS		FIC		Geral		Evolução	Significativa?
	pré	pós	pré	pós	pré	pós	pré	pós		
1.1	4,00	4,50	4,00	4,60	4,14	4,43	4,03	4,52	0,49	Sim ($p = 0,0076$)
1.2	4,23	4,64	4,20	4,40	3,86	4,29	4,13	4,48	0,35	Não ($p = 0,12$)
1.3	4,54	4,58	4,60	4,50	4,14	4,57	4,47	4,55	0,08	Não ($p = 0,51$)
1.5	4,85	4,83	4,80	4,80	4,57	4,71	4,77	4,79	0,02	Não ($p = 0,81$)
1.7	4,08	4,50	4,00	4,60	4,00	4,14	4,03	4,45	0,42	Sim ($p = 0,012$)

Fonte: elaborado pelo autor, 2018.

Tabela 14 – Questões analisadas na Tabela 13.

Identificador	Questão
1.1	Jogos e competições me deixam mais motivado a aprender do que aulas expositivas.
1.2	Eu gostaria que jogos e competições fossem explorados em outras disciplinas
1.3	Tenho interesse em atividades práticas envolvendo Segurança Computacional
1.5	Exercícios práticos de Segurança Computacional aumentam o entendimento sobre esta área.
1.7	Entendo que competições de Segurança Computacional aumentam o apelo desta área para o público geral.

Fonte: elaborado pelo autor, 2018.

A questão 1.6 verificou se os alunos se sentem preparados para participar de desafios, medindo o interesse e a percepção do próprio conhecimento, sob o seguinte enunciado: *Sinto-me suficientemente preparado para (começar a) participar de competições de Segurança Computacional*. As respostas estavam na mesma escala de Likert de cinco pontos (*Discordo totalmente ... Concordo totalmente*) das questões sobre satisfação. A Tabela 15 traz os resultados referentes a essa questão. No geral, os alunos se sentem pouco preparados para participar dessas competições: a média no pré-teste foi de 2,13, próxima da opção *Discordo parcialmente* na escala. A média no pós-teste aumentou para 2,79, mais próxima da opção *Neutro* na escala. Embora a percepção ainda seja mais negativa que positiva, houve uma evolução de 0,66 pontos em relação ao pré-teste, uma diferença estatisticamente significativa. Este resultado indica melhoria na percepção sobre o preparo com problemas de Segurança. Observando os resultados por turma, todas também apresentaram evolução.

Tabela 15 – Resultados da questão 1.6.

UDESC		TADS		FIC		Geral		Evolução	Significativa?
pré	pós	pré	pós	pré	pós	pré	pós		
2,15	3,05	2,20	2,40	2,00	2,86	2,13	2,79	0,66	Sim ($p = 0,018$)

Fonte: elaborado pelo autor, 2018.

A questão 2.1 investiga se os alunos têm interesse em uma carreira em Segurança, o que abrange interesse e perspectiva profissional. O enunciado desta questão era: *A probabilidade de eu tentar seguir carreira na área de Segurança Computacional é*, e as respostas estavam em uma escala de Likert de cinco pontos, indo de *Muito baixa* (1) a *Muito alta* (5). A Tabela 16 traz os resultados referentes a essa questão. O resultado do pós-teste foi 3,00 pontos (neutro), 0,07 acima da média do pré-teste, não representando diferença estatisticamente significativa. Isso significa, em primeiro lugar, que os alunos não demonstram uma predisposição favorável ou contrária a uma carreira profissional na área de Segurança, e, em segundo lugar, que a participação no desafio praticamente não alterou essa perspectiva.

Tabela 16 – Resultados da questão 2.1.

UDESC		TADS		FIC		Geral		Evolução	Significativa?
pré	pós	pré	pós	pré	pós	pré	pós		
2,77	2,84	3,20	3,00	2,86	3,29	2,93	3,00	0,07	Não ($p = 0,75$)

Fonte: elaborado pelo autor, 2018.

Para identificar se o nível das questões foi adequado, foram utilizadas como base as questões 3.1 (*Os problemas do jogo aplicado foram difíceis de se resolver*) e 3.2 (*Gastei muito tempo para resolver exercícios do jogo*) do pós-teste. Os resultados, exibidos na Tabela 17, mostram que as médias para ambas as questões ficaram próximas de 3,00 pontos, indicando proximidade com a neutralidade. Essa neutralidade significa que os respondentes não acharam os problemas particularmente fáceis ou difíceis.

Tabela 17 – Resultados sobre o nível das questões da competição.

Questão	UDESC	TADS	FIC	Geral
3.1	2,85	3,10	2,71	2,90
3.2	3,32	2,90	3,14	3,14

Fonte: elaborado pelo autor, 2018.

A análise de confiabilidade referente às questões 3.1 e 3.2 resultou em um alfa de Cronbach de 0,72. Segundo a Tabela 3 (Página 71), isso indica confiabilidade substancial, ou seja, as respostas para as duas perguntas apresentaram concordância elevada.

Os estudantes foram perguntados através do pós-teste, nas questões 4.1 e 4.2, sobre a motivação com a competitividade do jogo e com a composição de pro-

blemas, respectivamente. As respostas estavam em uma escala de Likert de cinco pontos, de *Muito desmotivador* (1) a *Muito motivador* (5). A Tabela 18 apresenta os resultados sobre esta avaliação, os quais mostram que a competitividade foi considerada um fator motivador na média geral. Na turma UDESC a média ficou ligeiramente abaixo de 4,00 pontos. A média de 4,24 pontos na questão 4.2 indica que os jogadores consideraram que este fator ficou com resultado entre *motivador* e *muito motivador*. Na turma FIC este resultado ficou entre *motivador* e *neutro*, com 3,71 pontos. Os resultados inferiores a 4,00 pontos (UDESC, questão 4.1, e FIC, questão 4.2) podem estar relacionados com a discordância de parte dos jogadores sobre a dificuldade dos problemas e o tempo despendido nas soluções.

Tabela 18 – Resultados sobre motivação com competitividade e composição de problemas.

Questão	UDESC	TADS	FIC	Geral
4.1 (competitividade)	3,92	4,00	4,14	4,00
4.2 (composição)	4,33	4,50	3,71	4,24

Fonte: elaborado pelo autor, 2018.

A questão 1.4 (*Tenho dificuldade em atividades práticas envolvendo Segurança Computacional*) contribui para a percepção de aprendizagem dos estudantes. As respostas estavam em uma escala de Likert de 5 pontos, de *Discordo fortemente* (1) a *Concordo fortemente* (5). Ao contrário das questões anteriores que usavam a mesma escala, nesta questão as respostas positivas têm pontuações inferiores a 3 (indicando que o aluno não tem dificuldade com as atividades).

Os resultados para a questão 1.4 são apresentados na Tabela 19, por meio da qual se percebe que a média geral do pós-teste obteve 2,93 pontos, enquanto o resultado geral do pré-teste para a mesma questão obteve 3,17, representando queda de 0,24 ponto. Ainda que a diferença não seja estatisticamente significativa, essa redução sugere que as competições tiveram um êxito moderado em diminuir a percepção da dificuldade com atividades práticas. O resultado é condizente também com os relatos sobre a discordância com a dificuldade dos problemas.

Tabela 19 – Resultados da questão 1.4.

UDESC		TADS		FIC		Geral		Evolução	Significativa?
pré	pós	pré	pós	pré	pós	pré	pós		
3,08	3,17	3,10	2,90	3,43	2,57	3,17	2,93	-0,24	Não (p=0,43)

Fonte: elaborado pelo autor, 2018.

4.3.5 Observações

A satisfação do jogador especificamente sobre a ferramenta desenvolvida não foi objeto de avaliação, conforme especificado no escopo do trabalho. Apesar disso,

fatores como usabilidade e percepção de utilidade podem estar agregados nos resultados indiretamente, e serão investigados em trabalhos futuros. O jogador não interage com o compositor de técnicas; apenas recebe suas instâncias de problemas em um arquivo compactado obtido através da aplicação *web*. Esta aplicação é responsável pela interação com o jogador, disponibilizando as funções de autenticação, fornecimento do arquivo com os problemas, exibição do placar, submissão de palavra secreta e resultado.

Com relação aos questionários, ressalta-se que mais de 66% dos estudantes afirmaram não ter experiência anterior em Segurança, o que sugere que os resultados de uma parcela das perguntas sobre jogos foi baseado na interação com esta ferramenta.

Durante as atividades, algumas questões puderam ser observadas: na turma da UDESC, na primeira competição, um estudante não matriculado na disciplina compareceu para acompanhar a atividade; na segunda competição, outros dois estudantes não matriculados compareceram. Ainda nesta turma, também foi possível verificar que o fator competição teve sua importância, com estudantes conversando sobre o que poderiam ter feito para obter resultados melhores ao final da primeira competição. Nesta turma houve relatos de problemas na ferramenta esteganográfica *outguess* na segunda competição.

Nas turmas TADS e FIC não houve menção ao placar e à competição. Os jogadores da turma do TADS resolveram os problemas de maneiras diversificadas. Por exemplo, o problema de conversão de decimal para ASCII foi resolvido em serviços *on-line*, em substituição manual e por meio de codificação em JavaScript no console do navegador *web* Google Chrome. O problema de encontrar a palavra secreta em um código Python compilado (envolvendo a técnica denominada *Descompilar binário e obter fonte Python*) foi resolvido com ferramentas de linha de comando Unix, tais como *strings* e *cat*, e por meio de ferramenta de descompilação *on-line*. Houve perguntas sobre programação de *scripts* ao organizador e, ainda, relatos de descontração durante as competições, tais como risadas ao concluir exercícios com imagens ou ao decifrar mensagens de pura distração, como no problema *Comentário em código-fonte de página HTML*.

Na turma FIC houve a desistência de dois estudantes, o que diminuiu o número de jogadores. Surgiram relatos de dificuldades com a extração do arquivo compactado que continha os problemas. Houve intervenção do organizador nesta ação.

Os benefícios da competição puderam ser corroborados com as avaliações aplicadas nas turmas FIC e UDESC. Nestas turmas houve prova teórica, individual e sem consulta. Os estudantes da turma FIC obtiveram 100% de acertos nas questões

relativas a técnicas vistas na competição, e os estudantes da UDESC obtiveram, na média, 88,5% de acertos.

4.4 DISCUSSÃO DOS RESULTADOS

De uma maneira geral, percebeu-se que os estudantes das três turmas mostraram interesse na área de Segurança Computacional. A solução de mais exercícios na segunda competição e em menos tempo, bem como a evolução da percepção dos estudantes acerca do preparo para resolver problemas de Segurança e da complexidade destes foram indicativos de aprendizagem por parte dos alunos. Ainda, a percepção deles sobre motivação para uso de jogos em aulas de Segurança foi positiva, ratificada pelas respostas nos questionários e por meio de observação.

É importante ressaltar que os resultados obtidos, tanto os de desempenho quanto os dos questionários, indicam que os exercícios tornaram-se mais fáceis após a aplicação da primeira competição. Um dos fatores que pode ter contribuído para isto é a quantidade limitada de técnicas selecionadas (oito). Além disso, a aplicação de desafios da segunda competição considerou o uso das mesmas oito técnicas da Competição 1, mas com composições distintas para o grupo C2.2.

A quantidade de jogadores com muitos acertos fez com que as medidas de tendência central (média e mediana) apresentassem valores altos e as diferenças de desempenho não fossem estatisticamente significativas. Com isso, alguns resultados, tais como a diferença entre o desempenho dos estudantes dos grupos C2.1 e C2.2, foram inconclusivos.

Observando o desempenho dos jogadores e a análise da Tabela 17 (página 85), nota-se que há um contraste: enquanto a maior parte dos estudantes obteve desempenho de 100% de rendimento no segundo campeonato, as respostas sobre a dificuldade dos problemas mostraram que, no geral, os jogadores não acharam os problemas fáceis ou difíceis, com resultado próximo do neutro neste item.

A familiaridade com as ferramentas utilizadas na C1 e com as técnicas vistas pode ter facilitado a atividade, tornando assim mais difícil a tarefa de avaliar a eficácia da aleatorização de problemas em produzir desafios distintos. Além disso, todas as ferramentas necessárias para solucionar os problemas foram apresentadas na aula preparatória, o que pode ter restringido e induzido o foco dos jogadores àquele conjunto de ferramentas. Embora isso não tenha sido observado diretamente, é concebível que alguns jogadores tenham encontrado respostas para os últimos problemas resolvidos em um desafio descartando as técnicas usadas nas soluções anteriores e tentando ver como as ferramentas remanescentes poderiam encaixar-se nos problemas restantes.

É necessário ponderar, ainda, que há ameaças à validade do experimento. Entre as possíveis ameaças identificadas estão a velocidade de transmissão de dados, as configurações dos computadores, as ferramentas utilizadas, eventuais atrasos dos estudantes nas aulas, o compartilhamento de *scripts* de solução, as diferentes abordagens dos organizadores/professores na aula de preparação e na explicação das regras do jogo. No caso dos questionários pré e pós-competição, a própria participação no questionário pode levar à mudança de comportamento do respondente (YU; OHLUND, 2010).

A avaliação geral da geração automatizada de desafios de Segurança é positiva. Acredita-se que a eficácia de aleatorização de problemas pode ser melhor quantificada com desafios contendo alguns problemas mais difíceis, e com menor probabilidade de repetição de técnicas. As medidas que podem ser tomadas nesse sentido incluem implementar novas técnicas e aumentar a quantidade de composições possíveis em cada problema.

4.5 CONSIDERAÇÕES DO CAPÍTULO

Este capítulo abordou a avaliação da pesquisa, partindo do projeto do experimento até execução e análise de resultados advindos de questionários, do SGBD e de observações realizadas durante a atividade. Os resultados obtidos foram interpretados e balizaram a discussão sobre as diferenças de desempenho entre as competições, bem como deverão nortear o seguimento do trabalho em atividades futuras.

O próximo capítulo traz as conclusões do trabalho e aborda as possibilidades de trabalhos futuros.

5 CONCLUSÃO

O contexto atual mostra que ensinar Segurança Computacional ainda é um desafio. Apesar do caráter contemporâneo e da importância relatada, o dinamismo inerente à área traz a necessidade quase instantânea de atualização. Acrescenta-se a isso o fato de que são necessárias novas práticas pedagógicas visando a atingir melhores resultados para os estudantes. Uma das práticas que contribui nesse sentido é a adoção de jogos e competições em sala de aula.

Foram apresentados os principais tipos de jogos de Segurança Computacional, entre os quais estão os jogos de caça ao tesouro, também chamados de desafio ou de CTF *Jeopardy!*. Estes jogos promovem a competição sem interação direta entre os jogadores, e são considerados flexíveis e fáceis de serem reproduzidos.

Considerando este cenário, o presente trabalho descreveu a criação do protótipo de um jogo do tipo caça ao tesouro para o ensino de Segurança Computacional através de uma competição. Técnicas frequentemente aplicadas em competições foram identificadas e compostas em instâncias distintas, trazendo como contribuição a formação de conjuntos de problemas exclusivos para cada jogador.

A ferramenta desenvolvida foi a principal contribuição do trabalho, por abordar os conceitos de geração automática de problemas e competições, abordar a possibilidade de problemas compostos, envolver técnicas de classes distintas, promover o equilíbrio nas competições através do fornecimento de problemas de mesma complexidade e abordar quatro classes de problemas. O trabalho ainda fornece informações sobre o mapeamento de dados em recentes competições do tipo desafio.

Os principais trabalhos relacionados propõem a geração automática da infraestrutura subjacente ou de cenários distintos, tais como máquinas virtuais com Sistemas Operacionais e vulnerabilidades diferentes (SecGen), ou a geração de problemas automáticos, com desequilíbrio na complexidade dos exercícios (MetaCTF) ou somente com alteração da *flag* para impedir o seu compartilhamento (PicoCTF). O trabalho proposto não aborda o cenário, mas cria automaticamente a competição com instâncias equivalentes em complexidade e distintas não somente na cadeia de caracteres que representa a palavra secreta a ser encontrada por cada jogador. Assim, a ferramenta permite o reaproveitamento de problemas e inviabiliza o compartilhamento de respostas.

A competição foi realizada em duas instituições (IFC e UDESC) e em três turmas, sendo duas de graduação e uma de curso de qualificação profissional. Houve uma aula preparatória, na qual foram apresentadas ferramentas para uso em proble-

mas de Segurança e exercícios, e duas aulas de competição. Na primeira, todos os jogadores receberam problemas que compunham as mesmas técnicas, mas cada um recebia uma instância distinta do problema. Na segunda competição, todas as turmas foram divididas em dois grupos, sendo que os jogadores de um grupo receberam instâncias de problemas compostos com as mesmas técnicas do primeiro desafio e os jogadores do outro receberam instâncias de problemas com composições diferentes.

O efeito da competição foi avaliado pelos jogadores através de questionários e do desempenho na atividade. A eficácia da ferramenta produzida em gerar desafios distintos foi medida comparando grupos de alunos que receberam problemas com composições diferentes, mas não foi encontrada diferença estatisticamente significativa entre os grupos. Assim, embora os resultados até o momento sejam encorajadores, a eficácia da aleatorização ainda precisa ser explorada mais a fundo. Em experimentos futuros, pretende-se gerar competições com novas técnicas e composições com maior complexidade, pois os resultados mostraram que os estudantes participantes da atividade obtiveram alto índice de aproveitamento, impedindo a diferenciação entre os grupos. Com relação à percepção de satisfação sobre a atividade desenvolvida, avaliada através de questionários pré e pós-teste, os resultados forneceram indícios de que a atividade foi bem recebida pelos discentes.

Apesar da citada boa recepção, somando todas as turmas, somente 30 alunos se dispuseram a participar da atividade. O tamanho da amostra foi considerado pequeno; neste sentido, deseja-se ampliar os estudos para obter resultados mais fidedignos. Tal ampliação pode ser feita no tamanho da amostra, mas também no nível de ensino dos jogadores (Ensino Médio ou Fundamental, por exemplo).

A pesquisa será continuada em três linhas: avaliação da ferramenta por parte do organizador, geração dinâmica de problemas com base no desempenho dos jogadores e avaliação geral da ferramenta. A primeira diz respeito ao sentimento do organizador quanto ao êxito em utilizar o jogo desenvolvido como ferramenta pedagógica no ensino de Segurança Computacional. A segunda visa a gerar os problemas de forma dinâmica com base no desempenho anterior do aluno em função do tempo despendido para solucionar o problema. Assim, quanto mais rápido o estudante concluir a atividade, mais complexo será o próximo problema gerado pela ferramenta. A avaliação geral da ferramenta permitirá obter o retorno tanto de jogadores quanto de organizadores, devendo, para isso, ser aplicada também em outras instituições e com o apoio de outros docentes. Nesta proposta de continuidade, fatores como usabilidade e facilidade de uso serão objeto de avaliação.

Para tornar possível gerar dinamicamente problemas com base no desempenho dos jogadores, faz-se necessário continuar trabalhando no desenvolvimento do protótipo do gerador de desafios. De forma imediata, a inclusão de novas técnicas,

a possibilidade de parametrização de níveis de composição, permitindo que mais de duas técnicas estejam presentes em um problema, e o gerenciamento autônomo de usuários, permitindo que os jogadores efetuem seu cadastro diretamente, são funcionalidades que devem ser implementadas.

É necessário alterar a ferramenta padrão de desafios de esteganografia, já que houve relatos de problemas em uma competição, e possibilitar a parametrização da ferramenta (por exemplo, criar problemas de esteganografia com as ferramentas `steghide` e outros com `f5.jar`). Na sequência, pretende-se incluir identificadores às competições, o que permitirá observar a evolução de jogadores com o passar do tempo e isolar os jogadores de turmas diferentes com mais facilidade para fins de análise de dados. Outra nova funcionalidade da ferramenta deve ser a parametrização do tempo de início e da duração da competição, não necessitando da ação do organizador para efetuar o encerramento manualmente e permitindo uma mensuração menos ruidosa deste fator.

REFERÊNCIAS

- 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. Version 0.98 Report. Disponível em: <<http://www.acm.org/binaries/content/assets/education/it2017.pdf>>. Acesso em: 05 jan. 2018.
- ALHOGAIL, A. Design and validation of information security culture framework. **Computers in human behavior**, Elsevier, v. 49, p. 567–575, 2015.
- ALMEIDA, M. E. B. de. Educação a distância na internet: abordagens e contribuições dos ambientes digitais de aprendizagem. **Educação e pesquisa**, SciELO Brasil, v. 29, n. 2, p. 327–340, 2003.
- ANDRADE, A. Recurso a simuladores na aprendizagem de fatores de segurança na exploração de tecnologias da informação. **Aprender na Era Digital-Jogos e Mobile Learning**. Santo Tirso: De Facto Editores, p. 65–82, 2012.
- ASTAKHOVA, L. The concept of the information-security culture. **Scientific and Technical Information Processing**, Springer Science & Business Media, v. 41, n. 1, p. 22, 2014.
- BISHOP, J. L.; VERLEGER, M. A. The flipped classroom: A survey of the research. In: **ASEE National Conference Proceedings, Atlanta, GA**. [S.l.: s.n.], 2013. v. 30, n. 9, p. 1–18.
- BOOPATHI, K.; SREEJITH, S.; BITHIN, A. Learning cyber security through gamification. **Indian Journal of Science and Technology**, v. 8, n. 7, p. 642–649, 2015.
- BRATUS, S. What hackers learn that the rest of us don't: notes on hacker curriculum. **IEEE Security & Privacy**, IEEE, v. 5, n. 4, 2007.
- BRATUS, S.; SHUBINA, A.; LOCASIO, M. E. Teaching the principles of the hacker curriculum to undergraduates. In: ACM. **Proceedings of the 41st ACM technical symposium on Computer science education**. [S.l.], 2010. p. 122–126.
- BURKET, J. et al. Automatic problem generation for capture-the-flag competitions. In: USENIX ASSOCIATION. **2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)**. [S.l.], 2015.
- BURNS, T. J. et al. Analysis and exercises for engaging beginners in online ctf competitions for security education. In: USENIX ASSOCIATION. **2017 {USENIX} Workshop on Advances in Security Education ({ASE} 17)**. [S.l.], 2017.
- CANDELL, R.; ZIMMERMAN, T.; STOUFFER, K. An industrial control system cybersecurity performance testbed. **National Institute of Standards and Technology. NISTIR**, v. 8089, 2015.

CAPUANO, E. **#JOLT Hackathon 2017**. 2017. Disponível em: <<https://blog.ecapuano.com/jolthackathon-2017/>>. Acesso em: 07 jan. 2018.

CARLISLE, M. **Doubt about PicoCTF and APG**. 2017. [mensagem pessoal]. Mensagem recebida por <ricardo.ladeira@ifc.edu.br> em 20 ago. 2017.

CERT.BR. **Estatísticas dos Incidentes Reportados ao CERT.br**. 2017. Disponível em: <<http://www.cert.br/stats/incidentes/>>. Acesso em: 08 ago. 2017.

CHAPMAN, P.; BURKET, J.; BRUMLEY, D. Picoctf: A game-based computer security competition for high school students. In: **3GSE**. [S.l.: s.n.], 2014.

CHEUNG, R. S. et al. Challenge based learning in cybersecurity education. In: **Proceedings of the 2011 International Conference on Security & Management**. [S.l.: s.n.], 2011. v. 1.

CHEUNG, R. S. et al. Effectiveness of cybersecurity competitions. In: **Proceedings of the International Conference on Security and Management (SAM)**. [S.l.: s.n.], 2012. p. 1.

CHOTHIA, T.; NOVAKOVIC, C. An offline capture the flag-style virtual machine and an assessment of its value for cybersecurity education. **2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)**, USENIX Association, 2015.

CHUNG, S. et al. What approaches work best for teaching secure coding practices. In: **Proceedings of the 2014 HUIC Education and STEM Conference**. [S.l.: s.n.], 2014.

CISCO. **Relatório anual de segurança da Cisco de 2016**. 2016. Disponível em: <https://www.cisco.com/c/dam/r/pt/br/internet-of-everything-ioe/assets/pdfs/cisco_2016_asr_pt-br.pdf>. Acesso em: 04 jan. 2018.

CLARKE, T. B.; NELSON, C. L. Classroom community, pedagogical effectiveness, and learning outcomes associated with twitter use in undergraduate marketing courses. **Journal for Advancement of Marketing Education**, v. 20, n. 2, 2012.

CONKLIN, A. Cyber defense competitions and information security education: An active learning solution for a capstone course. In: IEEE. **System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on**. [S.l.], 2006. v. 9, p. 220b–220b.

COWAN, C. et al. Defcon capture the flag: Defending vulnerable code from intense attack. In: IEEE. **DARPA Information Survivability Conference and Exposition, 2003. Proceedings**. [S.l.], 2003. v. 1, p. 120–129.

CTF write-ups repository. 2017. Disponível em: <<https://github.com/ctfs>>. Acesso em: 27 fev. 2017.

[d0x3d]. **[d0x3d!] – a network security game**. 2016. Disponível em: <<http://d0x3d.com/d0x3d/welcome.html>>. Acesso em: 25 nov. 2016.

DENNING, T.; FRIEDMAN, B.; KOHNO, T. **The Security Cards**. 2018. Disponível em: <<http://securitycards.cs.washington.edu/activities.html>>. Acesso em: 23 abr. 2018.

DENNING, T.; KOHNO, T.; SHOSTACK, A. Control-alt-hack tm: A card game for computer security outreach, education, and fun. 2012.

DEPARTMENT OF HOMELAND SECURITY. **National Cyber Security Awareness Month**. 2017. Disponível em: <<https://www.dhs.gov/national-cyber-security-awareness-month>>. Acesso em: 27 fev. 2017.

DETERLAB. **DeterLab: Cyber-Defense Technology Experimental Research Laboratory**. 2017. Disponível em: <<https://www.isi.deterlab.net/index.php3>>. Acesso em: 20 ago. 2017.

DFNDR LAB. **Relatório da Segurança Digital no Brasil. Terceiro trimestre**. 2017. Disponível em: <<https://lab.dfndrsecurity.com/wp-content/uploads/2017/10/DFNDR-Lab-Relato%CC%81rio-da-Seguranc%CC%A7a-Digital-no-Brasil-3%C2%BA-trimestre-2017.pdf>>. Acesso em: 03 jan. 2018.

DHILLON, G.; BACKHOUSE, J. Technical opinion: Information system security management in the new millennium. **Communications of the ACM**, ACM, v. 43, n. 7, p. 125–128, 2000.

DHILLON, G.; SYED, R.; PEDRON, C. Interpreting information security culture: an organizational transformation case study. **Computers & Security**, Elsevier, v. 56, p. 63–69, 2016.

DU, W. K. **SEED Project**. 2018. Disponível em: <<http://www.cis.syr.edu/~wedu/seed/>>. Acesso em: 23 abr. 2018.

DZAZALI, S.; ZOLAIT, A. H. Assessment of information security maturity: an exploration study of malaysian public service organizations. **Journal of Systems and Information Technology**, Emerald Group Publishing Limited, v. 14, n. 1, p. 23–57, 2012.

EAGLE, C. Computer security competitions: Expanding educational outcomes. **IEEE Security & Privacy**, IEEE, v. 11, n. 4, p. 69–71, 2013.

FENG, W. A scaffolded, metamorphic ctf for reverse engineering. In: USENIX ASSOCIATION. **2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)**. [S.l.], 2015.

FENG, W. **Doubt about your Reverse Engineering CTF**. 2017. [mensagem pessoal]. Mensagem recebida por <ricardo.ladeira@ifc.edu.br> em 21 ago. 2017.

FIELD, A.; MILES, J.; FIELD, Z. **Discovering Statistics Using R**. [S.l.]: SAGE Publications, 2012.

FURNELL, S.; CLARKE, N. Power to the people? the evolving recognition of human aspects of security. **Computers & Security**, Elsevier, v. 31, n. 8, p. 983–988, 2012.

GIBSON, B. I. Educational games for teaching computer science. University of Canterbury. Computer Science and Software Engineering, 2013.

GOYAL, K.; KINGER, S. Modified caesar cipher for better security enhancement. **International Journal of Computer Applications**, Foundation of Computer Science, v. 73, n. 3, 2013.

GUIMARAES, M.; SAID, H.; AUSTIN, R. Using video games to teach security. In: ACM. **Proceedings of the 16th annual joint conference on Innovation and technology in computer science education**. [S.l.], 2011. p. 346–346.

HUSSAIN, M. et al. Pixel value differencing steganography techniques: Analysis and open challenge. In: IEEE. **Consumer Electronics-Taiwan (ICCE-TW), 2015 IEEE International Conference on**. [S.l.], 2015. p. 21–22.

INTERNET CRIME COMPLAINT CENTER. **Internet Crime Report**. 2016. Disponível em: <https://pdf.ic3.gov/2016_IC3Report.pdf>. Acesso em: 04 jan. 2017.

IRVINE, C. E.; THOMPSON, M. F.; ALLEN, K. Cyberciege: gaming for information assurance. **IEEE Security & Privacy**, IEEE, v. 3, n. 3, p. 61–64, 2005.

JONES, A. **Elevation of Privilege - The Game**. 2010. Disponível em: <<https://social.technet.microsoft.com/wiki/contents/articles/285-elevation-of-privilege-the-game.aspx>>. Acesso em: 11 fev. 2018.

KAMBOURAKIS, G. Security and privacy in m-learning and beyond: Challenges and state of the art. **International Journal of u-and e-Service, Science and Technology**, Citeseer, v. 6, n. 3, p. 67–84, 2013.

KAPP, K. M. **The gamification of learning and instruction: game-based methods and strategies for training and education**. [S.l.]: John Wiley & Sons, 2012.

KIRKPATRICK, D. Revisiting kirkpatrick's four-level model. **Training and development**, v. 50, n. 1, p. 54–59, 1996.

KOBERSY, I. S. et al. The system of the methodological principles of management of enterprise development. **Mediterranean Journal of Social Sciences**, v. 6, n. 3 S4, p. 25, 2015.

KRITZINGER, E.; BADA, M.; NURSE, J. R. A study into the cybersecurity awareness initiatives for school learners in south africa and the uk. In: SPRINGER. **IFIP World Conference on Information Security Education**. [S.l.], 2017. p. 110–120.

LACEY, T. H.; PETERSON, G. L.; MILLS, R. F. The enhancement of graduate digital forensics education via the dc3 digital forensics challenge. In: IEEE. **System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on**. [S.l.], 2009. p. 1–9.

LADEIRA, R. R.; OBELHEIRO, R. R. Práticas educacionais no ensino da computação forense: um relato de experiência. **Revista de Empreendedorismo, Inovação e Tecnologia**, v. 4, n. 1, p. 110–120, 2017.

LANDIS, J. R.; KOCH, G. G. The measurement of observer agreement for categorical data. **biometrics**, JSTOR, p. 159–174, 1977.

LEGG, P. A. Visualizing the insider threat: challenges and tools for identifying malicious user activity. In: IEEE. **Visualization for Cyber Security (VizSec), 2015 IEEE Symposium on**. [S.l.], 2015. p. 1–7.

LIKERT, R. A technique for the measurement of attitudes. **Archives of psychology**, 1932.

LIU, Z. et al. Base62x: An alternative approach to base64 for non-alphanumeric characters. In: IEEE. **Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on**. [S.l.], 2011. v. 4, p. 2667–2670.

MALWAREWOLF. **Network Forensics – Round 1: Ann’s Bad AIM**. 2015. Disponível em: <<https://malwerewolf.com/2015/03/network-forensics-round-1-anns-bad-aim/>>. Acesso em: 26 nov. 2016.

MANSUROV, A. A ctf-based approach in information security education: An extracurricular activity in teaching students at altai state university, russia. **Modern Applied Science**, v. 10, n. 11, p. 159, 2016.

MAVI INTERACTION. **Agent Surefire**. 2016. Disponível em: <<https://agentsurefire.com/insiderthreat/>>. Acesso em: 02 nov. 2016.

MEC. **Resolução nº 5, de 16 de Novembro de 2016**. 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. Disponível em: <http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=52101-rces005-16-pdf&category_slug=novembro-2016-pdf&Itemid=30192>. Acesso em: 04 jan. 2018.

MICROSOFT. **The Elevation of Privilege (EoP) Card Game**. 2016. Disponível em: <<https://www.microsoft.com/en-us/SDL/adopt/eop.aspx>>. Acesso em: 26 nov. 2016.

MIRKOVIC, J. et al. Evaluating cybersecurity education interventions: Three case studies. **IEEE Security & Privacy**, IEEE, v. 13, n. 3, p. 63–69, 2015b.

MIRKOVIC, J.; PETERSON, P. Class capture-the-flag exercises. In: **3GSE**. [S.l.: s.n.], 2014.

MIRKOVIC, J. et al. Engaging novices in cybersecurity competitions: A vision and lessons learned at acm tapia 2015. In: **2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)**. [S.l.: s.n.], 2015a.

MONDAL, M. et al. Defending against large-scale crawls in online social networks. In: ACM. **Proceedings of the 8th international conference on Emerging networking experiments and technologies**. [S.l.], 2012. p. 325–336.

NORTHCUTT, S. **What the PCAP contest actually tells us**. 2016. Disponível em: <<https://www.linkedin.com/pulse/what-pcap-contest-actually-tells-us-stephen-northcutt>>. Acesso em: 26 abr. 2016.

NOVA LABS. **Cybersecurity Lab**. 2018. Disponível em: <<http://www.pbs.org/wgbh/nova/labs/lab/cyber/>>. Acesso em: 05 jan. 2018.

OLANO, M. et al. Securityempire: Development and evaluation of a digital game to promote cybersecurity education. In: **3GSE**. [S.l.: s.n.], 2014.

O'LEARY, M. Innovative pedagogical approaches to a capstone laboratory course in cyber operations. In: ACM. **Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education**. [S.l.], 2017. p. 429–434.

OWASP. **OWASP Cornucopia**. 2018. Disponível em: <https://www.owasp.org/index.php/OWASP_Cornucopia#tab=Main>. Acesso em: 23 abr. 2018.

PALMER, I. et al. Digital forensics education: a multidisciplinary curriculum model. In: SPRINGER. **International Conference on Digital Forensics and Cyber Crime**. [S.l.], 2015. p. 3–15.

PAN, Y. et al. Game-based forensics course for first year students. In: ACM. **Proceedings of the 13th annual conference on Information technology education**. [S.l.], 2012. p. 13–18.

PETULLO, W. M. et al. The use of cyber-defense exercises in undergraduate computing education. In: **ASE@ USENIX Security Symposium**. [S.l.: s.n.], 2016.

PRASHAR, A. Assessing the flipped classroom in operations management: A pilot study. **Journal of Education for Business**, Taylor & Francis, v. 90, n. 3, p. 126–138, 2015.

PWC. **Game of Threats™- A cyber threat simulation**. 2017. Disponível em: <<https://www.pwc.com/us/en/financial-services/cybersecurity-privacy/game-of-threats.html>>. Acesso em: 28 dez. 2017.

R CORE TEAM. **R: A Language and Environment for Statistical Computing**. Vienna, Austria, 2018. ISBN 3-900051-07-0. Disponível em: <<http://www.R-project.org/>>.

RAMAN, R.; LAL, A.; ACHUTHAN, K. Serious games based approach to cyber security concept learning: Indian context. In: IEEE. **Green Computing Communication and Electrical Engineering (ICGCEE), 2014 International Conference on**. [S.l.], 2014. p. 1–5.

ROBERT HALF. **Segurança da Informação: defendendo o seu futuro**. 2016. Disponível em: <https://www.roberthalf.com.br/sites/roberthalf.com.br/files/legacy-pdfs/robert_half_it_security.pdf>. Acesso em: 16 dez. 2016.

RURSCH, J. A.; JACOBSON, D. When a testbed does more than testing: The internet-scale event attack and generation environment (iseage)-providing learning and synthesizing experiences for cyber security students. In: IEEE. **Frontiers in Education Conference, 2013 IEEE**. [S.l.], 2013. p. 1267–1272.

SAILER, M. et al. How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction. **Computers in Human Behavior**, Elsevier, v. 69, p. 371–380, 2017.

SCHREUDERS, Z. C. et al. Security scenario generator (secgen): A framework for generating randomly vulnerable rich-scenario vms for learning computer security and hosting ctf events. In: USENIX ASSOCIATION. **USENIX**. [S.l.], 2017.

SHARPLES, M. The design of personal mobile technologies for lifelong learning. **Computers & Education**, Elsevier, v. 34, n. 3-4, p. 177–193, 2000.

SHOSTACK, A. Elevation of privilege: Drawing developers into threat modeling. In: **3GSE**. [S.l.: s.n.], 2014.

SOLMS, B. V. Information security—the fourth wave. **Computers & Security**, Elsevier, v. 25, n. 3, p. 165–168, 2006.

SUBY, M.; DICKSON, F. The 2015 (isc) 2 global information security workforce study. **Frost & Sullivan in partnership with Booz Allen Hamilton for ISC2**, 2015.

TAYLOR, C. et al. Ctf: State-of-the-art and building the next generation. In: **USENIX ASSOCIATION. 2017 {USENIX} Workshop on Advances in Security Education ({ASE} 17)**. [S.l.], 2017.

TERI, S. et al. Student use and pedagogical impact of a mobile learning application. **Biochemistry and Molecular Biology Education**, Wiley Online Library, v. 42, n. 2, p. 121–135, 2014.

THE WHITE HOUSE. **Commission on Enhancing National Cybersecurity, Presidential Documents. Executive Order 13718 of February 9, 2016**. 2016. Disponível em: <<https://www.gpo.gov/fdsys/pkg/FR-2016-02-12/pdf/2016-03038.pdf>>. Acesso em: 16 dez. 2016.

TODT, K. et al. **Report on Securing and Growing the Digital Economy**. [S.l.: s.n.], 2016.

TULARAM, G. A. Traditional vs non-traditional teaching and learning strategies – the case of e-learning! In: **Proceedings of International Conference on Engineering Education and Research – ICEER 2016**. [s.n.], 2016. Disponível em: <https://www.westernsydney.edu.au/__data/assets/pdf_file/0005/1176746/iCEER2016_Conference_Proceedings_official.pdf>. Acesso em: 27 dez. 2017.

VIGNA, G. Teaching network security through live exercises. In: **Security education and critical infrastructures**. [S.l.]: Springer, 2003. p. 3–18.

VIGNA, G. et al. Ten years of ictf: The good, the bad, and the ugly. In: **3GSE**. [S.l.: s.n.], 2014.

VYKOPAL, J.; BARTÁK, M. On the design of security games: From frustrating to engaging learning. In: **ASE@ USENIX Security Symposium**. [S.l.: s.n.], 2016.

WEE, J. M. C.; BASHIR, M.; MEMON, N. D. Self-efficacy in cybersecurity tasks and its relationship with cybersecurity competition and work-related outcomes. In: **ASE@ USENIX Security Symposium**. [S.l.: s.n.], 2016.

WEISS, R.; MACHE, J.; NILSEN, E. Top 10 hands-on cybersecurity exercises. **Journal of Computing Sciences in Colleges**, Consortium for Computing Sciences in Colleges, v. 29, n. 1, p. 140–147, 2013.

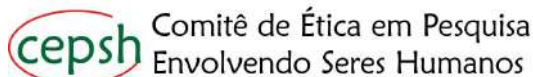
WEISS, R. S. et al. Teaching cybersecurity analysis skills in the cloud. In: ACM. **Proceedings of the 46th ACM Technical Symposium on Computer Science Education**. [S.l.], 2015. p. 332–337.

WHITE, G. B.; DODGE, R. The national collegiate cyber defense competition. In: **Proceedings of the Tenth Colloquium for Information Systems Security Education**. [S.l.: s.n.], 2006.

YASINSAC, A. et al. Computer forensics education. **IEEE Security & Privacy**, IEEE, v. 99, n. 4, p. 15–23, 2003.

YU, C.-h.; OHLUND, B. Threats to validity of research design. **Retrieved January**, v. 12, p. 2012, 2010.

APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO



GABINETE DO REITOR

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

O(a) senhor(a) está sendo convidado a participar de uma pesquisa de mestrado intitulada “Automatizando a Geração de Desafios Compostos Aplicados no Ensino de Segurança Computacional”, que fará observação e avaliação do desempenho dos estudantes na resolução de desafios de segurança. A pesquisa tem como objetivo avaliar a eficácia da geração de problemas de uma ferramenta e a satisfação, o interesse e o comprometimento do usuário na participação da atividade. Serão previamente marcados a data e o horário para a realização das atividades, utilizando questionário e observação. Estas medidas serão realizadas na Universidade do Estado de Santa Catarina. Também serão realizados exercícios. Não é obrigatório participar da atividade nem responder a todas as perguntas.

O(a) senhor(a) não terá despesas e nem será remunerado(a) pela participação na pesquisa.

Os riscos destes procedimentos serão mínimos, por envolverem resolução de exercícios práticos referentes a conteúdo ministrado em sala de aula com o uso do computador, e questões de opinião dos envolvidos. Caso você sinta algum desconforto ou cansaço durante a realização das atividades propostas, você poderá realizar uma pausa ou encerrar sua participação sem qualquer tipo de constrangimento.

A sua identidade será preservada, pois cada indivíduo será identificado por um número. Apenas os pesquisadores responsáveis (mestrando e orientador) terão acesso aos dados brutos com a identificação dos participantes.

Os benefícios e vantagens em participar deste estudo serão relativos ao aprendizado de tópicos de Segurança Computacional usando uma estratégia didático-pedagógica diferenciada em relação à abordagem tradicional de sala de aula. A longo prazo, espera-se que a geração automática de desafios se mostre uma ferramenta motivadora e eficaz para o aprendizado de variados assuntos em Segurança Computacional.

As pessoas que estarão acompanhando os procedimentos serão os pesquisadores Ricardo de la Rocha Ladeira (mestrando) e Rafael Rodrigues Obelheiro (orientador).

O(a) senhor(a) poderá se retirar do estudo a qualquer momento, sem qualquer tipo de constrangimento.

Solicitamos a sua autorização para o uso de seus dados para a produção de artigos técnicos e científicos. A sua privacidade será mantida através da não-identificação do seu nome.

Este termo de consentimento livre e esclarecido é feito em duas vias, sendo que uma delas ficará em poder do pesquisador e outra com o sujeito participante da pesquisa.

NOME DO PESQUISADOR RESPONSÁVEL PARA CONTATO: Rafael Rodrigues Obelheiro

NÚMERO DO TELEFONE: +55 47 3481-7892

ENDEREÇO: UDESC/CCT - Departamento de Ciência da Computação - Rua Paulo Malschitzki, 200 - Campus Universitário Prof. Avelino Marcante - Bairro Zona Industrial Norte - Joinville, SC

ASSINATURA DO PESQUISADOR:

Comitê de Ética em Pesquisa Envolvendo Seres Humanos – CEPESH/UDESC

Av. Madre Benvenuta, 2007 – Itacorubi – Florianópolis – SC -88035-901

Fone/Fax: (48) 3664-8084 / (48) 3664-7881 - E-mail: cepsh.reitoria@udesc.br / cepsh.udesc@gmail.com

CONEP- Comissão Nacional de Ética em Pesquisa

SEPN 510, Norte, Bloco A, 3º andar, Ed. Ex-INAN, Unidade II – Brasília – DF- CEP: 70750-521

Fone: (61) 3315-5878/ 5879 – E-mail: conep@saude.gov.br

TERMO DE CONSENTIMENTO

Declaro que fui informado sobre todos os procedimentos da pesquisa e, que recebi de forma clara e objetiva todas as explicações pertinentes ao projeto e, que todos os dados a meu respeito serão sigilosos. Eu compreendo que neste estudo, as medições dos experimentos/procedimentos de tratamento serão feitas em mim, e que fui informado que posso me retirar do estudo a qualquer momento.

Nome por extenso: _____

Assinatura: _____

Local: _____ Data: ____/____/____

APÊNDICE B – QUESTIONÁRIO DE LEVANTAMENTO DE PERFIL

- Queremos saber um pouco sobre você. Elaboramos 8 (oito) questões e pedimos que você as responda
- Sua privacidade será garantida e as informações só serão acessadas pelos organizadores da intervenção, conforme consta no Termo de Consentimento que você assinou.
- Tempo de resposta estimado: entre 2 e 5 minutos.

1. E-mail:

2. Quantos anos você tem?

3. Qual é o seu sexo?

- ☐ Masculino
- ☐ Feminino

4. Você cursou o Ensino Médio e o Ensino Fundamental em escola

- ☐ Pública
- ☐ Privada

5. Além do seu curso atual, você tem alguma outra formação em nível de graduação ou pós-graduação? Qual?

6. Marque a opção que melhor caracteriza sua familiaridade com o sistema operacional Linux:

- ☐ Nada familiar. Só sei reinicializar o Sistema.
- ☐ Levemente familiar. Consigo executar algumas tarefas básicas usando interface gráfica e ambientes de desenvolvimento (IDEs).
- ☐ Um pouco familiar. Uso principalmente a interface gráfica, mas sei usar alguns comandos de terminal para manipular arquivos e/ou compilar e executar programas.
- ☐ Moderadamente familiar. Sinto-me razoavelmente confortável na linha de comando, já escrevi alguns scripts simples.
- ☐ Extremamente familiar. Prefiro usar interface de linha de comando, já escrevi scripts razoavelmente complexos.

7. Resuma sua experiência em Segurança Computacional (marque uma ou mais opções).

- ☐ Não tenho experiência em Segurança Computacional.
- ☐ Estudei por conta própria.
- ☐ Estou fazendo a disciplina novamente.
- ☐ Fiz um curso fora desta instituição.
- ☐ Trabalhei ou trabalho na área.

8. Qual é a sua principal motivação para cursar a disciplina de Segurança?

- ☐ Achava o assunto interessante, mas não tinha muito conhecimento sobre ele.
- ☐ Complementar minha formação em Computação.
- ☐ Cumprir os requisitos do curso.
- ☐ Pretendo seguir carreira em Segurança ou em uma área correlata.
- ☐ Outro (especifique).

APÊNDICE C – QUESTIONÁRIO PRÉ-TESTE

- Elaboramos 8 (oito) questões e pedimos que você as responda.
- Sua privacidade será garantida e as informações só serão acessadas pelos organizadores da intervenção, conforme consta no Termo de Consentimento que você assinou.
- Tempo de resposta estimado: entre 2 e 5 minutos.

1. Para cada uma das afirmativas da primeira coluna, marque a opção que melhor caracteriza seu grau de concordância:

	Discordo totalmente	Discordo parcialmente	Neutro	Concordo parcialmente	Concordo totalmente
1.1. Jogos e competições me deixam mais motivado a aprender do que aulas expositivas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.2. Eu gostaria que jogos e competições fossem explorados em outras disciplinas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.3. Tenho interesse em atividades práticas envolvendo Segurança Computacional.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.4. Tenho dificuldade em atividades práticas envolvendo Segurança Computacional.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.5. Exercícios práticos de Segurança Computacional aumentam o entendimento sobre esta área.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.6. Sinto-me suficientemente preparado para (começar a) participar de competições de Segurança Computacional.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.7. Entendo que competições de Segurança Computacional aumentam o apelo desta área para o público geral.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2. Marque a opção que melhor descreve suas perspectivas profissionais envolvendo Segurança Computacional:

2.1. A probabilidade de eu tentar seguir carreira na área de Segurança Computacional é

☐ Muito
baixa

☐ Baixa

☐ Neutro

☐ Alta

☐ Muito
alta

APÊNDICE D – QUESTIONÁRIO PÓS-TESTE

- Elaboramos 13 (treze) questões e pedimos que você as responda.
- Sua privacidade será garantida e as informações só serão acessadas pelos organizadores da intervenção, conforme consta no Termo de Consentimento que você assinou.
- Tempo de resposta estimado: entre 2 e 5 minutos.

1. Para cada uma das afirmativas da primeira coluna, marque a opção que melhor caracteriza seu grau de concordância:

	Discordo totalmente	Discordo parcialmente	Neutro	Concordo parcialmente	Concordo totalmente
1.1. Jogos e competições me deixam mais motivado a aprender do que aulas expositivas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.2. Eu gostaria que jogos e competições fossem explorados em outras disciplinas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.3. Tenho interesse em atividades práticas envolvendo Segurança Computacional.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.4. Tenho dificuldade em atividades práticas envolvendo Segurança Computacional.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.5. Exercícios práticos de Segurança Computacional aumentam o entendimento sobre esta área.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.6. Sinto-me suficientemente preparado para (começar a) participar de competições de Segurança Computacional.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
1.7. Entendo que competições de Segurança Computacional aumentam o apelo desta área para o público geral.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2. Marque a opção que melhor descreve suas perspectivas profissionais envolvendo Segurança Computacional:

2.1. A probabilidade de eu tentar seguir carreira na área de Segurança Computacional é

- ☐ Muito baixa
 ☐ Baixa
 ☐ Neutro
 ☐ Alta
 ☐ Muito alta

3. Para cada uma das afirmativas da primeira coluna, marque a opção que melhor caracteriza seu grau de concordância:

	Discordo totalmente	Discordo parcialmente	Neutro	Concordo parcialmente	Concordo totalmente
3.1. Os problemas do jogo aplicado foram difíceis de se resolver.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.2. Gastei muito tempo para resolver exercícios do jogo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

4. Classifique o grau de motivação que você atribui a cada um dos aspectos do jogo descritos na primeira coluna:

	Muito desmotivador	Desmotivador	Neutro	Motivador	Muito motivador
4.1. A competitividade.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.2. A aplicação de exercícios compostos (aqueles em que mais de uma técnica estavam envolvidas).	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

5. (Opcional) Utilize este espaço para registrar suas impressões sobre a competição (críticas, sugestões ou outros comentários que julgar pertinentes).

APÊNDICE E – DICIONÁRIO DE DADOS

Este apêndice apresenta a descrição das tabelas do Banco de Dados presentes no sistema *web*. Os tipos de dados utilizados nos atributos são:

- **BOOL**: armazena um valor lógico (**TRUE** ou **FALSE**) utilizando 1 *bit*;
- **INT**: armazena numéricos inteiros utilizando 32 *bits*;
- **VARCHAR**: armazena caracteres alfanuméricos utilizando até 255 caracteres;
- **TIMESTAMP**: armazena a concatenação entre data e hora utilizando 32 *bits*;

Tabela: Usuário

Descrição: armazena dados de credenciais para acesso ao sistema.

Atributo	Tipo	Descrição
id	INT	Campo que deve armazenar o identificador do usuário. Este campo é uma chave primária com autoincremento (cláusulas PRIMARY KEY e AUTO_INCREMENT).
saltpass	VARCHAR(10)	Campo que deve armazenar o <i>salt</i> que será concatenado com a senha do usuário. Este campo não pode ser nulo (cláusula NOT NULL).
pass	VARCHAR(64)	Campo que deve armazenar o <i>hash</i> da senha concatenada com o <i>salt</i> . Este campo não pode ser nulo (cláusula NOT NULL).

Tabela: Resposta

Descrição: armazena dados que identificam as respostas dos exercícios.

Atributo	Tipo	Descrição
idUsuario	INT	Campo que deve armazenar o identificador do usuário. Este campo é uma chave estrangeira (referencia a tabela <code>Usuario</code>) e é parte da chave primária (cláusula <code>PRIMARY KEY</code>).
idProblema	INT	Campo que deve armazenar o identificador do problema/exercício. Este campo é parte da chave primária (cláusula <code>PRIMARY KEY</code>).
resposta	VARCHAR (64)	Campo que deve armazenar o <i>hash</i> da resposta. Este campo não pode ser nulo (cláusula <code>NOT NULL</code>).
tentativas	INT	Campo que deve armazenar a quantidade de tentativas de resposta do usuário pra o exercício. Este campo não pode ser nulo (cláusula <code>NOT NULL</code>).
acertou	BOOL	Campo que deve armazenar o valor verdadeiro (<code>TRUE</code>) quando o usuário já acertou a questão e falso (<code>FALSE</code>) quando a resposta da questão ainda não foi submetida corretamente. Este campo não pode ser nulo (cláusula <code>NOT NULL</code>).
hora	TIMESTAMP	Campo que deve armazenar o <i>timestamp</i> de uma submissão correta. Este campo possui valor padrão "0" (cláusula <code>DEFAULT</code> com valor 0).

Tabela: Submissao

Descrição: armazena dados de todas as submissões realizadas pelos usuários.

Atributo	Tipo	Descrição
idUsuario	INT	Campo que deve armazenar o identificador do usuário. Este campo é uma chave estrangeira (referencia a tabela <code>Usuario</code>) e é parte da chave primária (cláusula <code>PRIMARY KEY</code>).
idProblema	INT	Campo que deve armazenar o identificador do problema/exercício. Este campo é parte da chave primária (cláusula <code>PRIMARY KEY</code>).
respostaInformada	VARCHAR (64)	Campo que deve armazenar a resposta submetida pelo jogador. Este campo não pode ser nulo (cláusula <code>NOT NULL</code>).
hora	TIMESTAMP	Campo que deve armazenar o <i>timestamp</i> da submissão. Este campo é parte da chave primária (cláusula <code>PRIMARY KEY</code>).

APÊNDICE F – TABELA DE APROVEITAMENTO DE TÉCNICAS

Este apêndice apresenta a tabela completa de aproveitamento de técnicas nas competições C1 e C2. O documento original está disponível em <<http://bit.ly/2E0smqW>>.

	FIC									
Técnica/Jogo	1	N	T1	2.1	N	T2.1	2.2	N	T2.2	Total
César	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%
Desc. Java	3	7	42.86%	3	3	100.00%	4	4	100.00%	71.43%
Desc. 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%
	UDESC									
Técnica/Jogo	1	N	T1	2.1	N	T2.1	2.2	N	T2.2	Total
César	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%
Desc. Java	11	13	84.62%	6	6	100.00%	7	7	100.00%	92.31%
Desc. 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									
Técnica/Jogo	1	N	T1	2.1	N	T2.1	2.2	N	T2.2	Total
César	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%
Desc. Java	7	10	70.00%	4	5	80.00%	5	5	100.00%	80.00%
Desc. 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%
Técnica/Jogo	Total Geral									
César	85.00%									
HTML	88.33%									
Desc. Java	83.33%									
Desc. Python	91.67%									
Esteg	60.83%									
Robots	55.00%									
B64	78.95%									
Int/ASCII	80.00%									

APÊNDICE G – DEPENDÊNCIAS OPERACIONAIS

A versão atual do gerador de desafios foi criada em Sistema Operacional *Unix-like* e exige as seguintes ferramentas e pacotes para a geração dos problemas:

- `apache2`
- `awk`
- `base64`
- `bash`
- `bsdgames` (pacote que contém a ferramenta `caesar`)
- `cat`
- `cp`
- `cut`
- `default-jre`
- `default-jdk`
- `echo`
- `expr`
- `grep`
- `head`
- `libapache2-mod-php`
- `libapache2-mod-php7.0`
- `ls`
- `mkdir`
- `mv`
- `mysql-server-5.5`
- `mysql-client-5.5`
- `oracle-java8-installer`
- `outguess`

- php
- php-common
- php-mysql
- php7.0
- php7.0-cli
- php7.0-common
- php7.0-fpm
- php7.0-json
- php7.0-mysql
- php7.0-opcache
- php7.0-readline
- printf
- pyc
- python
- read
- rev
- rm
- sed
- seq
- sh (dash)
- shuf
- sleep
- sort
- strings
- tail
- tr
- wc
- xxd

- `zip`

Para solucionar os desafios, sugere-se, no mínimo, o uso das seguintes ferramentas:

- `base64`
- `caesar`
- `outguess`
- `sed`
- `sh`
- `strings`
- Editor de texto
- Navegador *web*

Cabe ressaltar que é possível solucionar problemas propostos com serviços *on-line* e outras ferramentas além das citadas acima. Um exemplo disso seria um problema hipotético de encontrar uma *flag* em um arquivo em texto claro. Neste caso, o uso de um editor de texto seria suficiente, mas uma solução alternativa combinando os comandos `cat` e `grep`, e possivelmente automatizando esta rotina com algum compilador ou interpretador de comandos como o `bash`, poderia alcançar a mesma resposta.