TreasureHunt: Geração Automática de Desafios Aplicados no Ensino de Segurança Computacional

Defesa de Dissertação

Universidade do Estado de Santa Catarina
Mestrado em Computação Aplicada

Ricardo de la Rocha Ladeira, Rafael Rodrigues Obelheiro (orientador)
{ricardo.ladeira@ifc.edu.br, rafael.obelheiro@udesc.br}

23 de abril de 2018

Sumário

Introdução

- Segurança Computacional é um tema **onipresente**.

Introdução

- Necessidade de uma **cultura em segurança**.
- Pouco conhecimento do público geral.
- Pouca mão de obra
  - Pesquisas indicam necessidade e falta de profissionais especializados
- Iniciativas para conscientização em Segurança existem (THE WHITE HOUSE, 2016; VIGNA 2003; WHITE, 2006).

- São necessárias ações **no contexto da educação formal**.
  - Oportunidade de pesquisa
  - Aumentar a aquisição de habilidades (práticas) em Segurança Computacional
  - Aumentar o interesse na área e atrair profissionais.

Introdução

- Problemas no ensino de cibersegurança
  - Poucos cursos abrangendo o assunto;
  - Currículos desatualizados;
  - Espaço pequeno: amplitude superficial *versus* profundidade exígua;
  - Eficácia das metodologias;
  - Preconceito e discussão ética (BRATUS 2010; VIGNA, 2003).
- **Jogos** têm sido usados para contornar esses problemas.

Jogos para Segurança

- **Jogos** são uma importante ferramenta pedagógica para a Segurança Computacional
  - Motivam
  - Ensinam

- Desafios (*treasure hunt*, forenses...), jogos de tabuleiro, *videogames*, ataque e defesa, entre outros.

- Cada um com recursos e público alvo diferentes.

Jogos para Segurança

- **Desafios**: problemas que precisam ser resolvidos com processos e ferramentas, tipicamente sem interação com outros jogadores.
  - Realizar engenharia reversa em um arquivo
  - Descobrir o conteúdo de uma mensagem criptografada
  - Encontrar um arquivo oculto
- Podem ser realizados em equipes ou individualmente.
- Podem ser lineares, não lineares ou mistos.
- Variam em complexidade e recursos necessários.

Jogos para Segurança

Considerações

- Trabalham não somente habilidades técnicas: liderança, *team-work*, tomada de decisão, ambiente de pressão etc.
- Indícios de incremento nas habilidades dos estudantes em (WEISS, 2013; RAMAN *et al.*, 2014; PETULLO *et al.*, 2016; CHEUNG *et al.*, 2011; CHEUNG *et al.*, 2012).
- Aprendizado ocorre nos **treinamentos** e através da **troca de experiências** em equipe.

Jogos para Segurança

Considerações

- Lições aprendidas nos trabalhos estudados direcionaram a pesquisa para a criação de um jogo do tipo **desafio**.
  - Flexibilidade na complexidade de problemas
  - Conhecimentos específicos exercitados
  - Podem ser aplicados a indivíduos ou grupos
  - Não exigem infraestruturas sofisticadas ou dedicadas
- Dificuldades
  - Criação de problemas requer conhecimento especializado (que é escasso)
    - Atividade normalmente **manual** e trabalhosa
  - Reaproveitamento de problemas
    - Perda do fator surpresa
    - Compartilhamento de respostas

Objetivos

- **Geral**: automatizar a geração de problemas para competições de Segurança, obtendo instâncias exclusivas de problemas, de forma parametrizável pelo organizador da competição.

- **Específicos**:
  - Selecionar técnicas;
  - Identificar composições possíveis;
  - Modelar uma competição de caça ao tesouro;
  - Aplicar conceitos através de uma ferramenta de geração de problemas automáticos e compostos;
  - Avaliar o efeito de uma competição no contexto acadêmico; e
  - Medir a percepção de satisfação, aprendizagem e interesse dos estudantes na atividade.

Geração Automática de Desafios

Parâmetros Gerais

- Proposta de uma competição não linear do tipo *desafio*.
- Encontrar a palavra secreta no formato `TreasureHunt{texto-aleatorio}`
- Individual.
- 1 ponto por acerto.
- Ranqueamento por pontos e, em caso de empate, por tempo.
- Ferramenta geradora de desafios com problemas compostos.

Geração Automática de Desafios

Desafios Existentes

- Análise de repositório de desafios mantido pela comunidade (CTF, 2017).
- Competições realizadas entre janeiro de 2016 e março de 2017.
- Levantamento de
  - Classes de problemas
  - Linearidade das competições
  - Problemas compostos
  - Técnicas frequentes

Geração Automática de Desafios

Desafios Existentes

- **Competições analisadas**: 84
- **Problemas analisados**: 1250
- **Problemas compostos**: 86 (6,9%)
- **Classes de problemas**:
  - Criptografia/Codificação
  - Engenharia Reversa
  - Forense
  - Miscelânea
  - *Web*

Geração Automática de Desafios

Desafios Existentes

- **Linearidade das competições**:
  - 46 Não lineares (95,8%)
  - 1 Linear (2,1%)
  - 1 Mista (2,1%)
  - Informação indisponível sobre 36 competições
- Cerca de **200 técnicas encontradas**.
  - Refinamentos para selecionar 8 técnicas com base
    - na frequência de uso;
    - no perfil do público-alvo (assunção sobre conhecimentos em Linux e Segurança Computacional);
    - no objetivo do trabalho; e
    - na familiaridade do autor com as técnicas (secundariamente).

Geração Automática de Desafios

Técnicas Selecionadas

- Comentário em código-fonte de página HTML
- Comentário em arquivo `robots.txt`
- (De)codificação em arquivo `base64`
- (Des)criptografia de Cifra de César
- (De)codificação de caractere ASCII para inteiro
- Descompilar binário e obter fonte Java
- Descompilar binário e obter fonte Python
- Esteganografia em imagens

- Técnicas implementadas individualmente e compostas quando possível.

Geração Automática de Desafios

Gerador de Desafios

- Implementação via *script*, com informações parametrizáveis.
- Organizador da competição interage com a ferramenta.
- Geração dos problemas da competição e dos conjuntos de arquivos de cada jogador.
- Configuração automática do SGBD.
  - Criação de tabelas e usuários
  - Armazenamento de todas as submissões no SGBD
- Envio dos problemas em arquivo ZIP para o servidor *web*.
- Código e dependências operacionais disponíveis em repositório *online*.

Geração Automática de Desafios

Gerador de Desafios

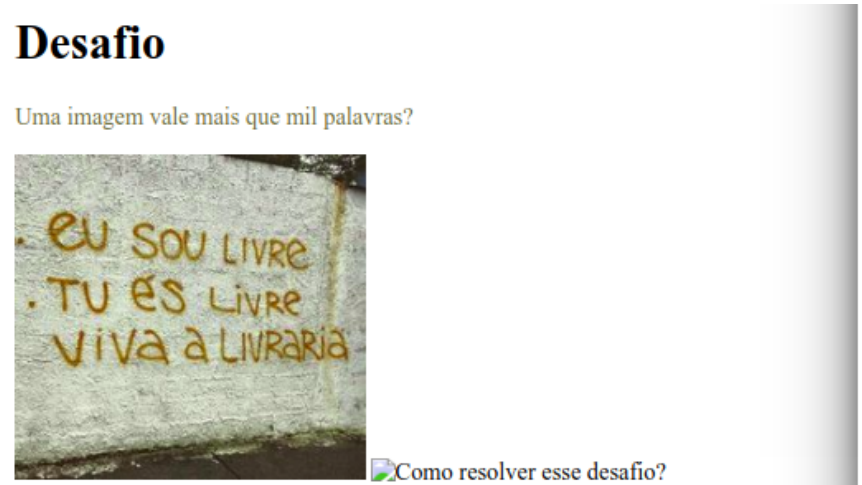
Figura: execução do script gerador de desafios.

```
----------
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:
```

Geração Automática de Desafios

Gerador de Desafios

Figura: problemas simples.



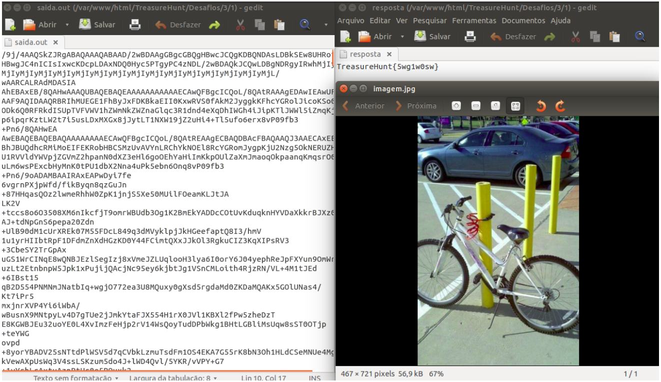

```
22          <img
    src="http://img.ibxk.com.br/2016/03/31/3120433266
    w=1040" alt="Imagem do desafio">
23          <img alt="Como resolver esse desafio?">
24          <ul> <!-- TreasureHunt{ago0tl7n2nbo} -->
25              <li>Nao caia em pegadinhas</li>
26              <li>Fique atento</li>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

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

Geração Automática de Desafios

Gerador de Desafios

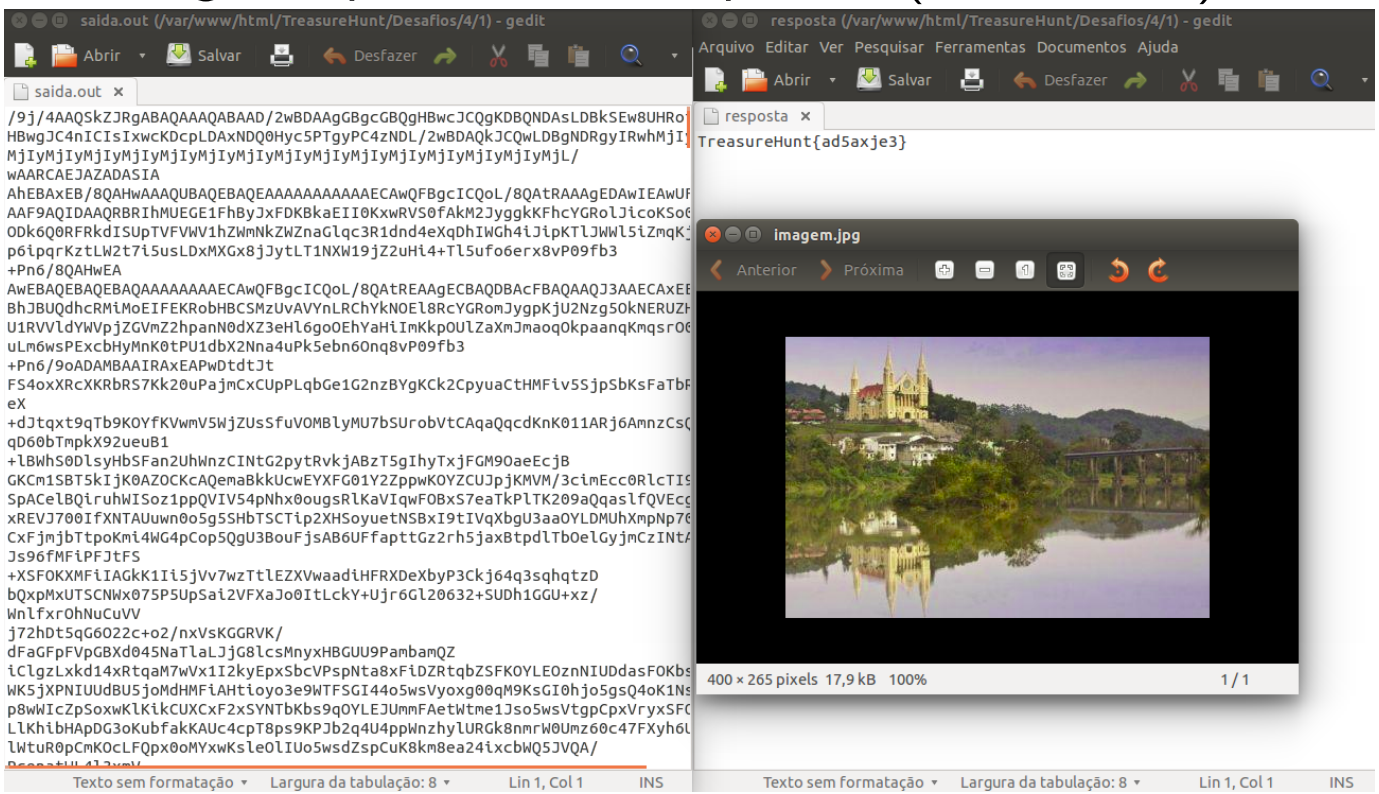
Figura: problemas compostos (instância 1).



Geração Automática de Desafios

Gerador de Desafios

Figura: problemas compostos (instância 2).



Geração Automática de Desafios

Sistema de Submissão

- Implementação via *web*.
- Jogadores interagem com a ferramenta.
  - Obtêm o arquivo ZIP com o conjunto de problemas
  - Submetem respostas aos problemas
  - Visualizam placares individual detalhado e geral
- Código e dependências operacionais disponíveis em repositório *online*.

Geração Automática de Desafios

Sistema de Submissão

Figura: interface de submissão de resposta e placar individual detalhado.

Geração Automática de Desafios

Sistema de Submissão

Figura: placar geral.

Geração Automática de Desafios

Trabalhos Relacionados

- Três trabalhos envolvendo APG (*Automatic Problem Generation*, ou Geração Automática de Problemas) em Segurança Computacional fortemente relacionados ao trabalho proposto:
  - PicoCTF (BURKET *et al.*, 2015)
  - MetaCTF (FENG, 2015)
  - SecGen (SCHREUDERS et al., 2017)

Geração Automática de Desafios

Trabalhos Relacionados

Tabela: comparativo entre os trabalhos relacionados e a ferramenta proposta.

Trabalho	**Geração automática**	**Composição de problemas**	**Uniformidade de problemas**	**Classes de problemas abordadas**
PicoCTF	problemas	×	✓	▶ Engenharia Reversa ▶ *Web* ▶ Miscelânea ▶ Codificação/ Criptografia
MetaCTF	competição	±	×	▶ Engenharia Reversa
SecGen	competição	±	×	▶ *Web* ▶ Forense ▶ Miscelânea ▶ Codificação/ Criptografia
TreasureHunt	competição	✓	✓	▶ Engenharia Reversa ▶ Forense ▶ Miscelânea ▶ Codificação/ Criptografia

Avaliação

Projeto de Experimento

- Aplicação da competição em laboratório de informática.
- Aceitação mediante assinatura do TCLE.
- Três momentos:
  - Aula preparatória: apresentação de ferramentas e exercícios.
  - Competição 1: 6 exercícios equivalentes a todos.
  - Competição 2: dividida em dois grupos
    - C2.1 recebe os mesmos problemas da C1
    - C2.2 recebe problemas com composições diferentes

Avaliação

Projeto de Experimento

- Questionários:
  - levantamento de perfil
  - pré-competição
  - pós-competição
- Os dois primeiros aplicados na aula preparatória e o último após a segunda competição.
- Questionários pré e pós-competição verificavam a percepção de satisfação, interesse e aprendizagem dos jogadores.

Avaliação

Projeto de Experimento

- Problemas selecionados para a C1 e para o grupo C2.1:
  - (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

Avaliação

Projeto de Experimento

- Problemas selecionados para o grupo C2.2:
  - (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

Avaliação

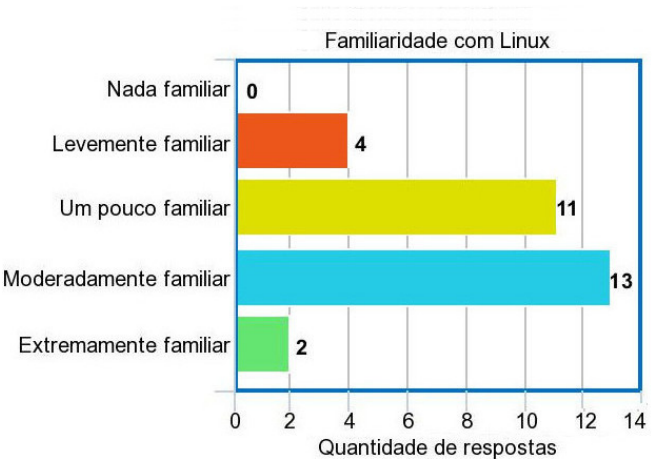
Execução

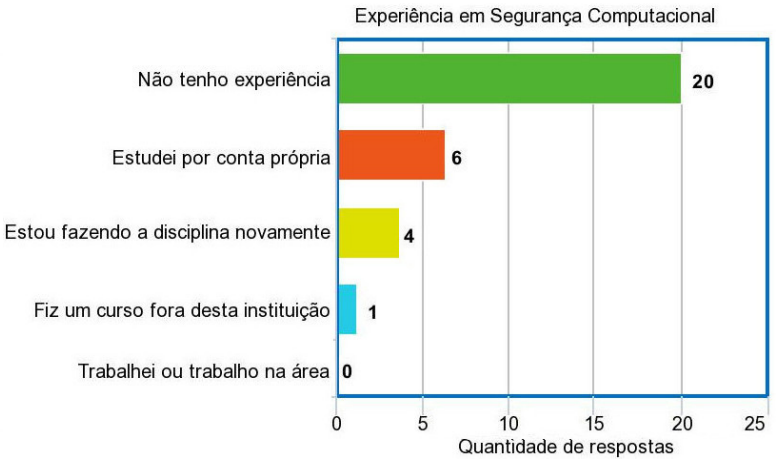
- Aplicação da competição em 3 turmas:
  - BCC – UDESC
  - Curso FIC – IFC
  - TADS – IFC
- Laboratórios equipados com as ferramentas.
- SO Unix-like.
- Total de 30 alunos.
- Aplicado em novembro de 2017.

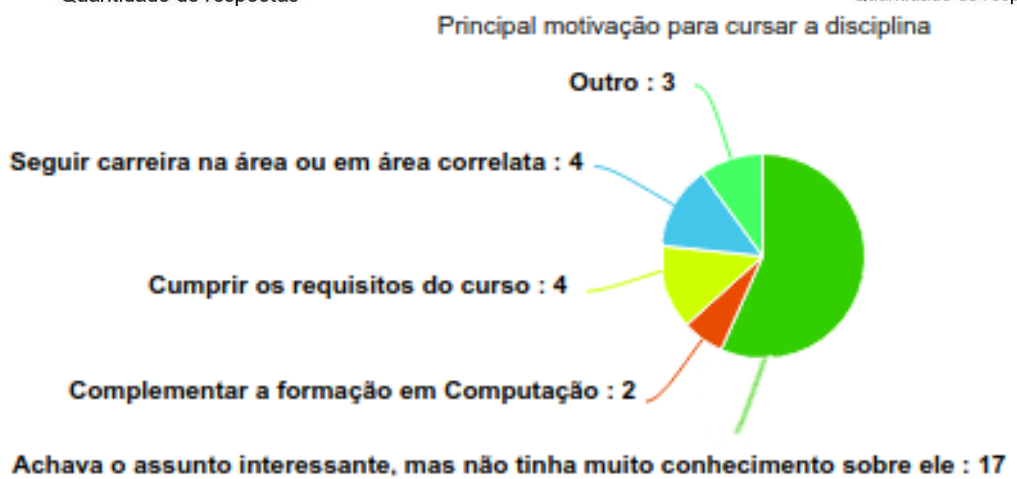
Avaliação

Perfil dos Jogadores

Figura: levantamento de perfil de jogador.







Avaliação

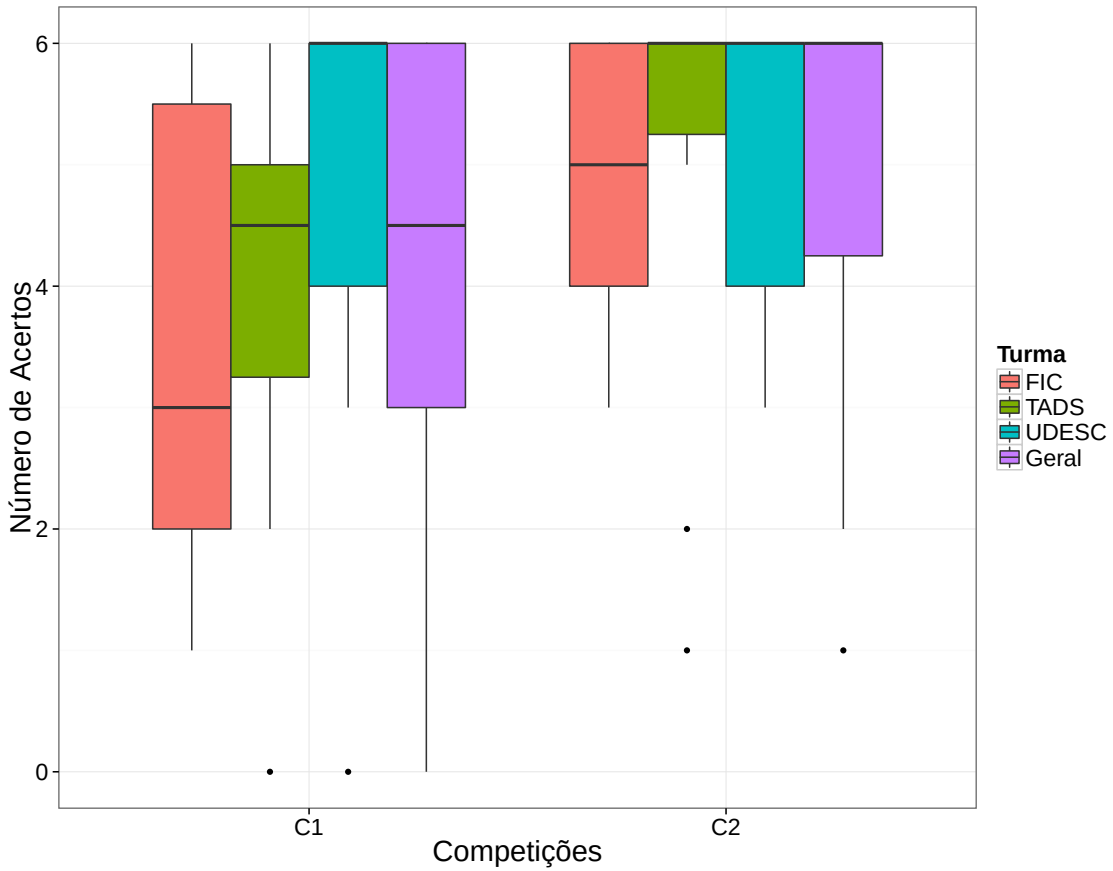
Resultados de Desempenho

- Número de acertos
- Taxa de submissões corretas
- Tempo médio de conclusão
- Aproveitamento por técnica
- Taxa de acertos por tipo de problema (simples/composto)

- Resultados comparados com Teste de Wilcoxon para amostras pareadas entre
  - C1 e C2
  - C2.1 e C2.2

Avaliação

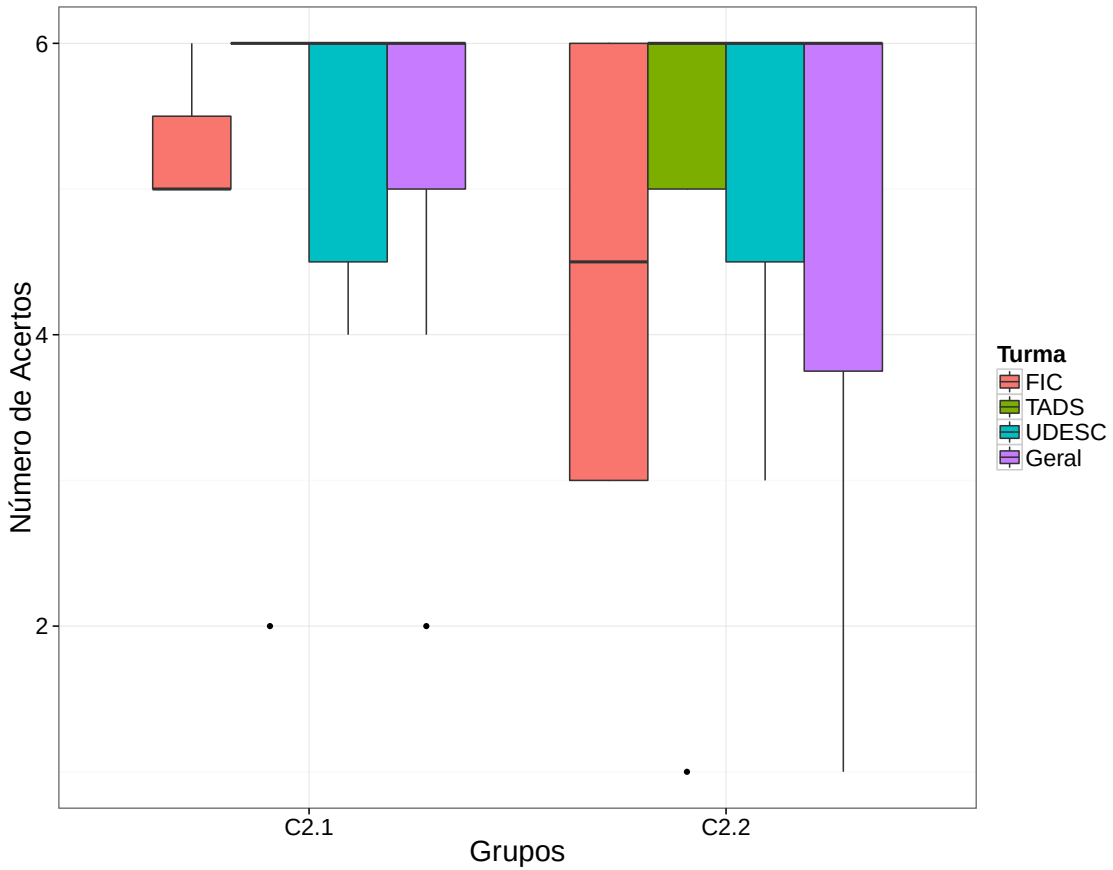
Figura: boxplot de acertos em C1 e C2.



- Desempenho com diferença estatisticamente significativa.

Avaliação

Figura: boxplot de acertos em C2.1 e C2.2.

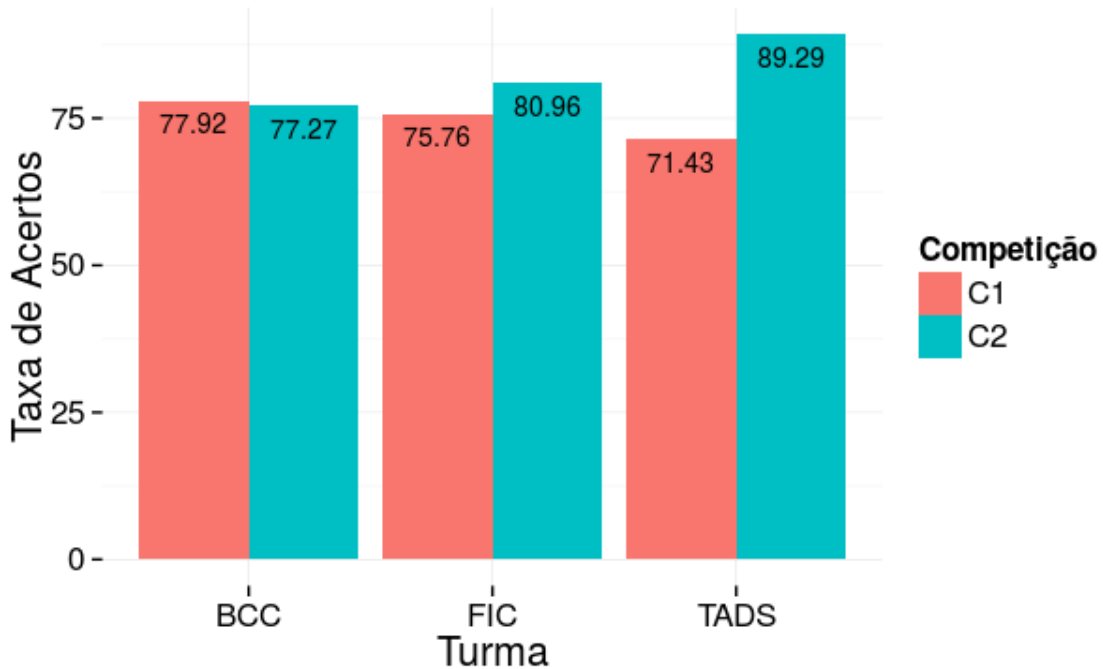


- Desempenho sem diferença estatisticamente significativa.

Avaliação

Resultados de Desempenho

Figura: taxa de submissões corretas (em %) por turma.

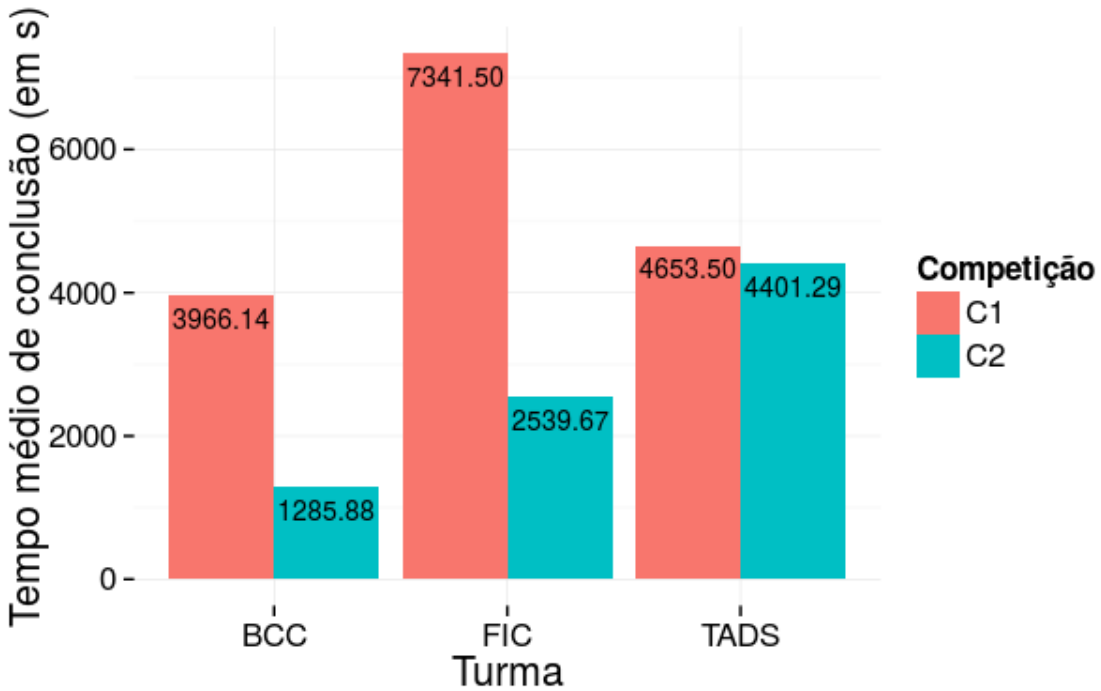


Não houve ocorrências de compartilhamento de respostas.

Avaliação

Resultados de Desempenho

Figura: tempo médio (em s) para conclusão da atividade.



Estatisticamente, não foi possível afirmar que C2.1 e C2.2 são diferentes.

Avaliação

Resultados de Desempenho

- Técnica mais fácil no geral: Python (91,67%).
  - Múltiplas soluções com as ferramentas conhecidas.
- Técnica mais difícil no geral: Robots (55,00%).
  - Encontrar o arquivo correto.
  - Desenvolver *script* de automatização de solução.
  - Composição exige uso de mais ferramentas.

- Em todas as competições, problemas simples foram mais fáceis de resolver se comparados aos problemas compostos.

Avaliação

Resultados dos Questionários

- Pré-teste: 30 respostas.
- Pós-teste: 29 respostas.
- Questões em escala de Likert
  - Discordo totalmente . . . Concordo totalmente
  - Muito baixa . . . Muito alta
  - Muito desmotivador . . . Muito motivador
- Consistência avaliada pelo coeficiente α de Cronbach.
- Diferença estatística verificada pelo Teste de Wilcoxon para amostras não pareadas.

Avaliação

Resultados dos Questionários

Tabela: resultado das questões de 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 ($p = 0,012$)

Tabela: enunciados das questões.

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.

Avaliação

Resultados dos Questionários

- Jogadores satisfeitos com a competição.
- Valores acima de 4,00 pontos na coluna *Geral*.
- Evolução em termos absolutos em todas as questões.
- Duas questões com diferença estatisticamente significativa.
- Confiabilidade com alfa de Cronbach de 0,78 (substancial, próximo de quase perfeita).

Avaliação

Resultados dos Questionários

- Questão 1.6: "*Sinto-me suficientemente preparado para (começar a) participar de competições de Segurança Computacional*".
- Interesse e percepção do conhecimento

Tabela : 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 ($p = 0,018$)

- Percepção de insuficiência, mas com evolução estatisticamente significativa.

Avaliação

Resultados dos Questionários

- Questão 2.1: "*A probabilidade de eu tentar seguir carreira na área de Segurança Computacional é*".
- Interesse em uma carreira em segurança (interesse e perspectiva profissional).

Tabela : 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 ($p = 0,75$)

- Predisposição nem favorável, nem contrária à carreira na área.
- Participação no desafio praticamente não alterou essa perspectiva.

Avaliação

Resultados dos Questionários

- Questões 3.1 e 3.2 indagaram sobre a adequação da atividade sobre dificuldade (3.1) e tempo gasto (3.2) na atividade.
- Confiabilidade substancial (alfa de Cronbach = 0,72).

Tabela : resultados das questões 3.1 e 3.2.

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

- Resultado contrastante com o desempenho.

Avaliação

Resultados dos Questionários

- Questões 4.1 e 4.2 indagaram sobre a motivação com a competitividade (4.1) e com a composição de problemas (4.2).

Tabela : resultados das questões 4.1 e 4.2.

Questão	**UDESC**	**TADS**	**FIC**	**Geral**
4.1	3,92	4,00	4,14	4,00
4.2	4,33	4,50	3,71	4,24

- Atividade motivadora, com dois resultados entre “neutro” e “motivador”.

Avaliação

Discussão dos Resultados

- Desempenho melhorou da `C1` para a `C2`
  - Em tempo
  - Em número de acertos
- `C2.1` e `C2.2` sem diferença estatisticamente significativa.
- Resultados de percepção de satisfação, aprendizagem e interesse foram positivos.

Avaliação

Observação

- Curiosidade de estudantes não matriculados;
- Fator competição presente na UDESC (Questão 4.1);
- Descontração;
- Erro na ferramenta `outguess`.

Conclusão

- Criação do protótipo de uma ferramenta de geração automática de competição do tipo caça ao tesouro.
  - Problemas equivalentes
  - Instâncias distintas
  - Diferentes classes de problemas
  - Composição de técnicas
- Competição aplicada em três turmas, utilizando três aulas.

Conclusão

- Não foi encontrada diferença estatística entre os grupos C2.1 e C2.2.
  - Eficácia da aleatorização precisa ser estudada profundamente.
- Resultados de satisfação, aprendizagem e interesse indicam que a atividade foi bem recebida.
  - Amostra foi considerada pequena.

Conclusão

Perspectivas Futuras

- Aprimorar o TreasureHunt.
- Acrescentar ID à competição.
- Gerenciamento autônomo de usuários.
- Expandir
  - a quantidade de técnicas;
  - a quantidade de ferramentas por técnica;
  - a quantidade de níveis de composição;
  - o público alvo.
- Há mais funcionalidades em desenvolvimento.

Referências

- BRATUS, S.; SHUBINA, A.; LOCASTO, M.E. Teaching the principles of the hacker curriculum to undergraduates. SIGCSE 2010. 122-126. 2010.
- BURKET, J.; CHAPMAN, P.; BECKER, T.; GANAS, C.; BRUMLEY, D.; Automatic Problem Generation for Capture-the-Flag Competitions. 2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15). ACM, Washington DC, USA. 2015.
- CHEUNG, R.S.; COHEN, J.P.; LO, H.Z.; ELIA, F. Challenge Based Learning in Cybersecurity Education. In International Conference on Security and Management (2011).
- CHEUNG, R.S.; COHEN, J.P.; LO, H.Z.; ELIA, F.; CARRILLO-MARQUEZ, V. Effectiveness of Cybersecurity Competitions. In International Conference on Security and Management (2012).
- CTF write-ups repository. Disponível em: <https://github.com/ctfs>. Acesso em: 27 fev. 2017.

Referências

- FENG, Wu-chang. A scaffolded, metamorphic ctf for reverse engineering. In: 2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15). USENIX Association. 2015.
- NORTHCUTT, S. What the PCAP contest actually tells us. (2016). Disponível em: <`https://www.linkedin.com/pulse/what-pcap-contest-actually-tells-us-`
- PETULLO, W.M. et al. "The Use of Cyber-Defense Exercises in Undergraduate Computing Education."2016 USENIX Workshop on Advances in Security Education (ASE 16). (2016).
- RAMAN, R.; LAL, A.; ACHUTHAN, K. "Serious games based approach to cyber security concept learning: Indian context,"Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on, Coimbatore (2014), pp. 1-5.

Referências

- THE WHITE HOUSE. Commission on Enhancing National Cybersecurity, Presidential Documents. Executive Order 13718 of February 9, 2016. Disponível em: <https://www.gpo.gov/fdsys/pkg/FR-2016-02-12/pdf/2016-03038.pdf>. Acesso em: 16 dez. 2016.
- 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.
- VIGNA, G. "Teaching network security through live exercises."Security education and critical infrastructures. Springer US p. 3-18. 2003.
- VIGNA, G. et al. "Ten years of ictf: The good, the bad, and the ugly."2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14). 2014.
- WEISS, R.; MACHE, J.; NILSEN, E. Top 10 hands-on cybersecurity exercises. J. Comput. Sci. Coll. 29, 1. 140-147. 2013.
- WHITE, G.B.; DODGE JR, R.C. The national collegiate cyber defense competition. In: Proceedings of the Tenth Colloquium for Information Systems Security Education. University of Maryland, University College Adelphi, MD. June 5-8. 2006.

TreasureHunt: Geração Automática de Desafios Aplicados no Ensino de Segurança Computacional

Defesa de Dissertação

Universidade do Estado de Santa Catarina
Mestrado em Computação Aplicada

Ricardo de la Rocha Ladeira, Rafael Rodrigues Obelheiro (orientador)
{ricardo.ladeira@ifc.edu.br, rafael.obelheiro@udesc.br}

23 de abril de 2018