José Luís Lopes Silva

CONSTRUÇÃO DE UM JOGO DE SEGURANÇA VOLTADO PARA ALUNOS DO MSI

1 2 9 0 UNIVERSIDADE D COIMBRA

1 2  9 0

UNIVERSIDADE D
COIMBRA

José Luís Lopes Silva

CONSTRUÇÃO DE UM JOGO DE SEGURANÇA VOLTADO PARA ALUNOS DO MSI

Dissertação no âmbito do Mestrado em Segurança Informática, especialização em Segurança Informática orientada pelo Professor Doutor Paulo Alexandre Ferreira Simões e pelo Professor Doutor Tiago José dos Santos Martins da Cruz à Faculdade de Ciências e Tecnologia / Departamento de Engenharia Informática.

Outubro de 2021

Faculdade de Ciências e Tecnologia

Departamento de Engenharia Informática

Construção de um jogo de segurança voltado para alunos do MSI

José Luís Lopes Silva

Dissertação no âmbito do Mestrado em Segurança Informática, especialização em Segurança Informática orientada pelo Professor Doutor Paulo Alexandre Ferreira Simões e pelo Professor Doutor Tiago José dos Santos Martins da Cruz à Faculdade de Ciências e Tecnologia / Departamento de Engenharia Informática.

Outubro de 2021

1 2 9 0

UNIVERSIDADE D
COIMBRA

Esta página foi intencionalmente deixada em branco.

Agradecimentos

Exmos. Hernâni, Rúben e Nunes - Excelentes amigos e profissionais, aos quais, desejo um enorme sucesso;

Exmo. Diogo Lourenço - Obrigado por todo o apoio, desejo-te um enorme sucesso;

Exma. Mia Rose - Um agradecimento especial por toda a atenção, desejo-te um enorme sucesso;

Exmo. Tiago Cruz - Obrigado por todo o apoio ao longo desde caminho, pela companhia, pelas dicas, pelas correções e por todo o carinho prestado. Sem dúvida, um excelente profissional e uma grande inspiração. Desejo um enorme sucesso;

Exmo. Paulo Simões - Obrigado pelo apoio prestado e pela dedicação, a qual, foi essencial para melhorar este projeto. Um dos melhores, de elevada perseverança, *brio* e generosidade. Desejo um enorme sucesso;

Docentes do Departamento de Engenharia Informática da Universidade de Coimbra - Muito obrigado pelo apoio prestado e pela oportunidade em realizar este projeto, aos quais, desejo um enorme sucesso;

Docentes da Escola Superior de Tecnologia de Castelo Branco - Um agradecimento especial, por todo o trabalho realizado e por colocarem ênfase na componente prática. Obrigado por serem um corpo docente ativo e por trazerem experiências inovadoras. Desejo um enorme sucesso;

À Família - os quais preservo no meu coração, desejo saúde e coragem;

Esta página foi intencionalmente deixada em branco.

Abstract

Globally, in the last decade universities are increasingly committed to offering courses and programs in computer security, due to the high demand for specialists in the current market. However, establishing efficient and effective courses in Cybersecurity Education is a huge challenge [239]. The new computer security courses end up addressing several areas, sometimes in an introductory way, either due to time limitations or inherent complexity. Therefore, several universities actively invest in teaching processes, with systematic and significant improvements in computer security courses, with practical activities, certifications, professional internships and Cybersecurity Training activities [124].

In 2017, the University of Coimbra initiated, for the first time, the Master in Computer Security (MSI), which introduced security concepts applied to the areas of software, cryptography, networks and computer infrastructure. The present project arises when considering the possibility of improving this course, with a strong practical component. Using the *Active Learning* [126] methodology with the Gamification concept [67], we aim to involve students and teachers from different areas of research, in learning computer security, through of Cybersecurity Training activities.

Thus, this project presents as a solution, a training platform in computer security skills, on the theme of *Capture the Flag* (CTF) competitions [170]. This solution aims to provide security exercises in an iterative environment (cyber range), based on real case studies, so that students can develop their offensive and defensive skills.

However, the necessary skills and substantial time required to create and maintain a CTF platform is a big barrier [239]. This project outlines the necessary measures to solve this problem, proposing a platform that incorporates automation and virtualization mechanisms. On the one hand, simplifying the management of CTF exercises, and on the other hand, enabling practical and motivating learning for future MSI course students.

Keywords

Cybersecurity Education, Cybersecurity Training, Capture the Flag, Active Learning

Esta página foi intencionalmente deixada em branco.

Resumo

Globalmente, na última década as universidades apostam cada vez mais na oferta de cursos e programas em segurança informática, devido à elevada procura por especialistas no mercado atual. Porém, estabelecer cursos eficientes e efetivos em *Cybersecurity Education* é um enorme desafio [239]. Os novos cursos em segurança informática, acabam por abordar diversas áreas, por vezes de forma introdutória, seja por limitação de tempo ou pela complexidade inerente. Pelo que, várias universidades investem ativamente nos processos de ensino, com melhorias sistemáticas e significativas nos cursos de segurança informática, com atividades práticas, certificações, estágios profissionais e atividades de *Cybersecurity Training* [124].

Em 2017, a Universidade de Coimbra iniciou, pela primeira vez, o Mestrado em Segurança Informática (MSI), o qual, introduziu conceitos de segurança aplicados às áreas de *software*, criptografia, redes e infraestruturas informáticas. O presente projeto surge ao considerar a possibilidade de melhorar este curso, com uma forte componente prática. Utilizando a metodologia *Active Learning* [126] com o conceito Gamification [67], pretendemos envolver os alunos e os professores, de diferentes áreas de investigação, na aprendizagem e gosto pela segurança informática, através de actividades de *Cybersecurity Training*, como os exercícios *Capture the Flag*.

Assim, este projeto apresenta como solução, uma plataforma de treino em competências de segurança informática, na temática das competições de *Capture the Flag* [170]. Esta solução visa disponibilizar exercícios de segurança num ambiente iterativo (*cyber range*), baseados em casos de estudo reais, para que os alunos desenvolvam as suas habilidades ofensivas e defensivas.

No entanto, as habilidades necessárias e o tempo substancial, necessárias para criar e manter uma plataforma de CTF, é uma grande barreira [239]. Neste projeto são traçadas as devidas medidas para solucionar o problema, propondo uma plataforma, que incorpore mecanismos de automação e virtualização. Por um lado, simplificando a gestão dos exercícios CTF, e por outro lado, possibilitando uma aprendizagem prática e motivante aos futuros alunos do curso MSI.

Palavras-Chave

Cybersecurity Education, *Cybersecurity Training*, *Capture the Flag*, *Active Learning*

Esta página foi intencionalmente deixada em branco.

Conteúdo

Acrónimos

API Application Programming Interface.

APT Advanced persistent threat.

CDR Challenge Design Requirements.

CLI Command Language Interpreter.

CSIS Center for Strategic and International Studies.

CSRF Cross-site Request Forgery.

CTF Capture the Flag.

FS File System.

ICS Industrial Control Systems.

IoT Internet of Things.

LFI Local File Inclusion.

LFS Large File Storage.

LMS Learning Management System.

MITM Man-in-The-Middle.

OOB Out-of-band.

PoC Prova de conceito.

PPC Professional Programming Challenges.

RCE Remote Code Execution.

RFI Remote File Inclusion.

SCADA Supervisory Control And Data Acquisition.

SI Segurança Informática.

SOAP Simple Object Access Protocol.

SQLi SQL Injection.

SSTI Server Side Template Injection.

VCN Virtual Clone Network.

WAF Web Application Firewall.

XSS Cross-Site Scripting.

Esta página foi intencionalmente deixada em branco.

Lista de Figuras

Lista de Tabelas

Esta página foi intencionalmente deixada em branco.

Capítulo 1

Introdução

Na presente dissertação intitulada "Construção de um jogo de segurança voltado para alunos do MSI"pretende-se conceber uma plataforma, a qual disponibilizará exercícios *Capture the Flag* aos alunos de segurança informática, com o objetivo de aplicar os conceitos teóricos disponibilizados nas aulas de mestrado, em cenários práticos.

Pretende-se também averiguar, através da formulação de hipóteses, o impacto da experiência adquirida com os exercícios *Capture the Flag* na aprendizagem dos alunos no mestrado de segurança informática, bem como, a sua relevância para a entrada destes alunos no mercado de trabalho da área da segurança informática.

Cybersecurity Capture the flag (CTF) designa uma competição que têm como objetivo a captura ou a defesa de uma *flag*, num dado ambiente. Em segurança, este ambiente envolve redes, servidores e serviços, sendo a *flag* representada virtualmente por um *token*, uma cadeia de caracteres com uma determinada estrutura, permitindo que seja facilmente reconhecida [171].

No presente trabalho iremos construir uma plataforma CTF para os alunos do MSI, na qual, iremos introduzir funcionalidades que consideramos pertinentes, concebendo cenários práticos, de modo, a tornar os processos em *Cybersecurity Education* mais eficientes e efetivos.

A plataforma a construir terá as seguintes funcionalidades: um sistema para armazenar e coleccionar os exercícios CTF, que irá incluir serviços de análise, verificação e classificação; um *website* com a documentação, os materiais de interesse e as soluções propostas pelos alunos do MSI; automatismos para controlar os exercícios CTF nas plataformas de virtualização; sistemas de monitorização com serviço de *chat*; e por último, o desenvolvimento de uma *Toolkit* para gerir e orquestrar os exercícios CTF.

Após a implementação deste sistema e a disponibilização do mesmo, iremos validar a nossa solução através da revisão de literatura e por meio de questionários, os quais, serão realizados com os alunos do Departamento de Engenharia Informática, com a finalidade de validar-mos as seguintes hipóteses de trabalho:

- Na hipótese 1, pretende-se testar se os conhecimentos prévios exigidos, e os conhecimentos adquiridos nos cursos de segurança informática, são adequados para a realização dos exercícios CTF;
- Na hipótese 2, pretende-se testar se a aprendizagem feita através dos exercícios de CTF é útil para o sucesso académico nos cursos superiores de segurança informática;

- Na hipótese 3, pretende-se averiguar se os exercícios de CTF estimularam a motivação e o interesse dos alunos para a área de segurança informática;
- Na hipótese 4, pretende-se aferir se a aprendizagem feita através dos exercícios de CTF se adequa ao mercado de trabalho atual na área da segurança informática.

A estrutura da presente dissertação divide-se nos seguintes capítulos:

- Capítulo 2 - no qual abordamos os conceitos introdutórios relacionados com o tema proposto e onde iremos definir os conceitos relevantes para a presente dissertação, bem como, as tecnologias utilizadas neste projeto;
- Capítulo 3 - Estado da arte, onde iremos apresentar o panorama atual das competições CTF, as suas categorias, os seus formatos e as plataformas utilizadas atualmente;
- Capítulo 4 - Planeamento e Metodologia, onde iremos apresentar a metodologia de trabalho e os procedimentos utilizados ao longo do projeto;
- Capítulo 5 - Arquitetura, onde iremos apresentar a arquitetura do sistema a desenvolver, definindo os requisitos funcionais e os não funcionais;
- Capítulo 6 - Implementação, onde iremos apresentar os detalhes de implementação da arquitetura proposta;
- Capítulo 7 - Resultados, onde iremos apresentar os resultados da revisão de literatura e os resultados obtidos nos questionários, realizados com os alunos do DEI;
- Capítulo 8 - Discussão dos resultados, onde iremos apresentar as considerações sobre os resultados obtidos;
- Capítulo 9 - Limitações e futuros desenvolvimentos, onde iremos descrever as limitações e os futuros desenvolvimentos deste projeto;
- Capítulo 10 - Conclusão, onde apresentamos as conclusões obtidas ao longo do presente projeto.

Capítulo 2

Conceitos Introdutórios e Tecnologias utilizadas

No presente capítulo apresentamos um conjunto de conceitos introdutórios, relevantes para a presente dissertação.

Este capítulo foi organizado em duas secções:

- na secção 2.1, definimos conceitos introdutórios, no âmbito do tema *Cybersecurity Education*;
- na secção 2.2, definimos as tecnologias e as ferramentas utilizadas no âmbito deste projeto;

2.1 Conceitos Introdutórios

Active Learning - *Active Learning Methodologies*

Active learning consiste em envolver o aluno diretamente no processo de aprendizagem e fornecer conteúdos utilizando atividades práticas [50].

Gamification

O processo de gamificação consiste em utilizar as técnicas e os elementos dos videojogos em outros contextos, que não sejam os videojogos [77].

Cybersecurity Education

Cybersecurity Education ou *Security Awareness Training*, área de ensino voltada para a segurança informática, que através de cursos, formações, certificações e outros programas de treino, ensina os aspetos e as técnicas de segurança informática, e consciencializa para as ameaças, para os riscos associados e para os ataques informáticos [27] [176].

Exercício de segurança - *Cyber Exercise*

Um exercício de segurança informática é um problema ou desafio de treino em segurança informática, que recorre a cenários de ataque e/ou defesa num ambiente virtual e/ou físico, com o objetivo de melhorar os conhecimentos e as habilidades de ataque e/ou defesa dos participantes [257]. Na literatura encontramos vários termos equivalentes, como *Cyber Exercise*, *Security Challenge* ou *Cybersecurity challenge*.

Challenge Design Requirements

Os Challenge Design Requirements (CDR) são os requisitos de um exercício de segurança, a título de exemplo, enumeramos os requisitos mais comuns (adaptados de [88]):

- Possui um meta de aprendizagem claramente definida?;
- Adapta-se ao *background* dos participantes?;
- Mecânicas definidas? Quais são as ferramentas necessárias e o que fazer?;
- Define um nível de dificuldade progressivo?;
- Promove discussões sobre as soluções entre os participantes? (existe uma solução mais simples para resolver o problema?);
- Adapta-se às habilidades dos participantes?;
- O exercício inclui ajudas (*"hints"*), que ajudam a chegar à solução?;
- Solução clara, padronizada e simples? (em vez de solução baseada em *obscure knowledge*)?;
- Existe uma duração planeada para cada exercício?;
- Se explica os problemas resultantes da interação de diferentes componentes?;
- Adapta-se às políticas e às práticas internas de *secure coding*?;

Ciclo de vida dos exercícios de segurança - Life Cycle of Cyber Exercise

O ciclo de um exercício de segurança é dividido em cinco fases [250], que são:

- *Preparation* - fase em que se define os objetivos do exercício, a história do cenário, o método de pontuação e o seu ambiente;
- *Dry run* (simulação) - testas o ambiente desenvolvido de acordo com os objetivos do exercício;
- *Execution* - o exercício entra em execução, e os participantes (do lado atacante e/ou do defensor) tentarão atingir os seus objetivos;
- *Evaluation* - o desempenho dos participantes é avaliado com base no método de pontuação e nos objetivos de aprendizagem;
- *Repetition* - o ambiente é restaurado e todo o processo é repetido para um novo exercício;

Cyber Warrior

Um *Cyber Warrior* é definido como um profissional de Segurança Informática (SI), que é responsável pela segurança e operações de uma infraestrutura, com foco no estabelecimento de mecanismos de defesa e prevenção de ataques informáticos [40].

Cyber Wargames - Serious games

Serious games recorre aos jogos e às simulações, para fins educacionais. Os *Cyber Wargames*, em segurança informática, simulam as experiências e os eventos de um ataque

informático real, sendo consideradas como *Serious games*. Os *wargames* ensinam os conceitos práticos de segurança informática, por vezes, recorre-se a processos de gamificação. No contexto empresarial, os *wargames* servem para verificar quais são as falhas da organização, como esta irá responder a um ataque simulado, como se adaptam os planos de *business continuity* e quais são os planos de contingência [245].

Existem vários variantes de *Cyber Wargames* e ambientes de aprendizagem (*Hands-on learning environments*), que passamos a descrever:

- *Security E-learning Platforms* - são plataformas de ensino à distância no âmbito da segurança informática, que permitem a interação entre os instrutores e os alunos [53];
- *Cyber ranges* - representam ambientes complexos e similares aos ambientes reais, que são destinados a atividades de treino (*cyberwarfare training*), ao desenvolvimento de tecnologias (*cybertechnology*) e a análises forense [252];
- *Security Testbeds* - ambiente ou plataforma, onde os fabricantes e operadores testam o *hardware* e o *software* num ambiente protegido e isolado (também referido como *"sandbox"*). A *testbed* é utilizada como ambiente de treino e formação em segurança informática, onde os seus componentes são testados [100];
- *Capture the Flag* - no contexto da segurança informática, Capture the Flag (CTF) é um tipo de *cyber wargame*, que pode ser jogado individualmente ou por equipas, e abrange um conjunto de competências técnicas no domínio da segurança informática [86].

Security E-learning Platforms

Os Learning Management System (LMS) são sistemas que ajudam a gerir as atividades de aprendizagem [2].

- *Avatao* - plataforma de *e-learning*, que fornece exercícios em segurança informática (*hands-on exercises*) [10];
- *Hacking-Lab* - plataforma *online*, onde os participantes tem de realizar várias tarefas em simultâneo (manter as aplicações em execução, encontrar as vulnerabilidades e corrigir as mesmas) [154];
- *iCTF framework* - apresentada na Fig.2.1 [247], permite às instituições de ensino e às organizações a execução e a customização de exercícios CTF, através de um *website*, de uma base de dados, um *scorebot* e das máquinas virtuais dos participantes;
- InCTF - corresponde a uma modificação da *iCTF framework*, que simplifica o *setup*, substituindo as máquinas virtuais por *containers Docker* [203];

Cyber ranges

As *Cyber Ranges* retratam incidentes reais, destacamos algumas projetos:

- *DETER, DeterLab* - consiste num laboratório virtual com o objetivo de incentivar a pesquisa e a aprendizagem em segurança informática [76];
- *National Cyber Range* [94] - este ambiente fornece um vasto conjunto de máquinas virtuais e hardware, com simulação de comunicações, que retratam ataques informáticos, como *malware*, *distributed denial of service attacks* ou *cross-site scripting*;

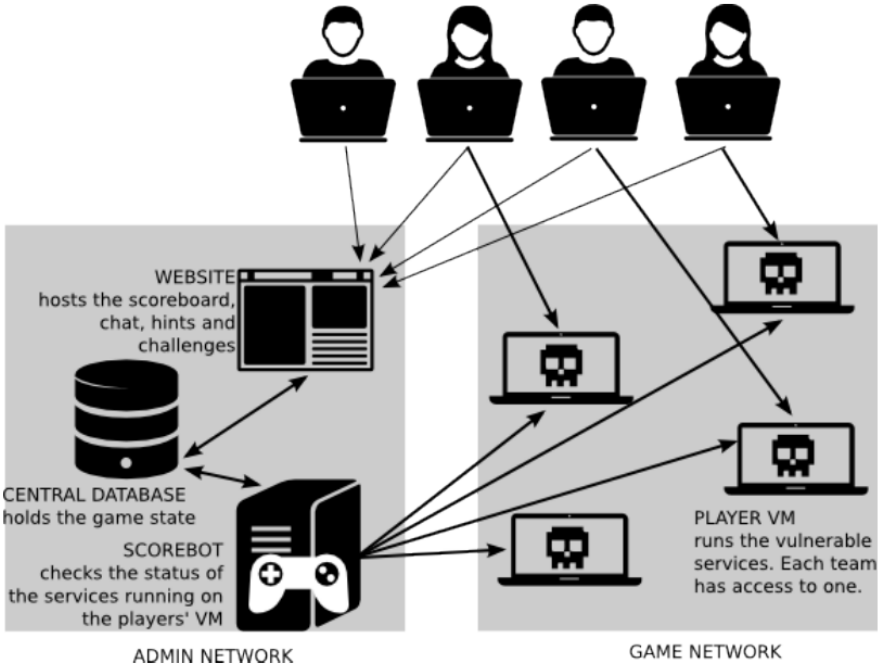

Figura 2.1: Arquitetura do sistema iCTF (Fonte: [247])

- *SimSpace Cyber Range* [214] - disponibiliza uma rede virtual, que inclui replicas de redes organizacionais, as Virtual Clone Network (VCN);
- *EDURange* [22] - foi desenhado para o ensino, para complementar as aulas teóricas. Os cenários foram desenhados especificamente para os alunos desenvolverem novas técnicas;
- *KYPO Cyber Range* [32] - *Cyber Exercise and Research Platform* que é focada na modelação e simulação de sistemas e redes computacionais, em ambientes virtuais isolados.

Security Testbeds

Uma das finalidades das *testbeds* consiste no ensino de competências durante os cursos e as formações de segurança informática. As *testbeds* de segurança envolvem vários componentes, virtuais e/ou físicos. Por exemplo, na Fig.2.2, apresenta uma cenário típico, composto por vários exercícios de segurança (*Cyber Security Challenges*), e por uma infraestrutura Industrial Control Systems (ICS), que pode ser virtual ou física (*ICT Infratructure*) [100].

Enumeramos várias *testbeds* de segurança, que foram utilizadas em cursos e formações, em particular:

- *Testbed* Internet of Things (IoT) (*AIT Austrian Institute of Technology*) - os investigadores criaram uma *testbed* na área do IoT, representada na Fig.2.3. A *testbed* é composta por uma rede, com vários dispositivos IoT, e pelo *Command and Control*, que envolve processos de análise, monitorização de operações e algoritmos de *machine learning* [213];
- *KYPO4INDUSTRY* (*Masaryk University Czech*) - é uma *testbed* para ensinar ICS *cybersecurity* na prática, através de um jogo educativo, onde os alunos são respon-

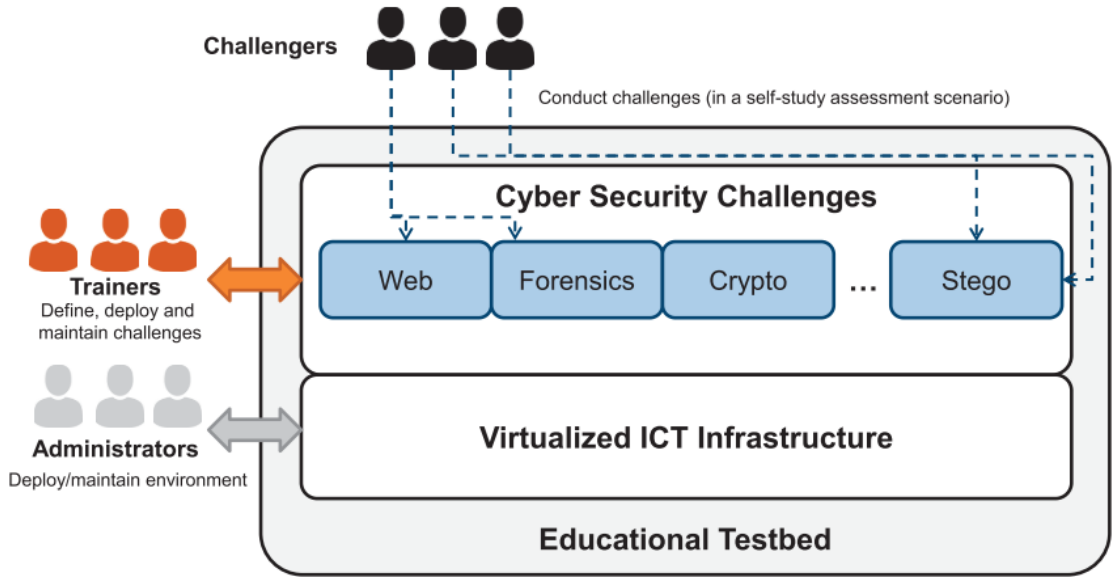

Figura 2.2: Exemplo Setup of a cyber security testbed (Fonte: [100])

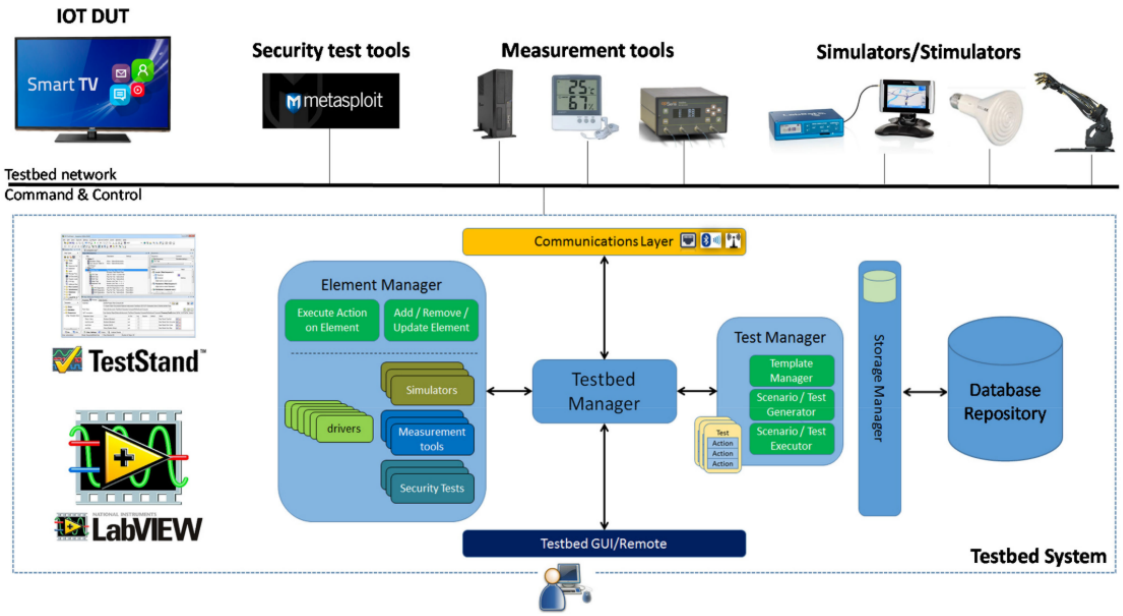

Figura 2.3: *Security Testbed* para dispositivos IoT (Fonte: [213])

sáveis pelas operações de uma organização, e tem de aplicar os conceitos das aulas teóricas para proteger o ambiente [32];

Capture the Flag Competitions

- DEF CON CTF - a ronda de qualificação é realizada *online* no formato *Jeopardy* e a final é presencial no formato *Attack-defence*. O tamanho da equipa é limitado a 8. Tópicos abordados: *SQL injection*, *Cross Site Scripting*, *buffer overflow*, *Timing attacks* , *Heat exploits*, *Malformed network constructs*;

- *Google Capture The Flag* - incluiu exercícios sobre *machine learning*, *artificial intelligence*, *hardware security* e IoT;
- UCSB iCTF - é realizada no formato Attack-defence. Não são impostos limites no número de participantes por equipa. Tópicos abordados: *Deobfuscation*, *binary*, *web*, *application security*, *network security*, *reverse engineering*, *buffer overflow*, *sandbox*, *dalvik bytecode*;
- Mozilla CTF - é realizada no formato *Jeopardy*. Não são impostos limites no número de participantes por equipa. Tópicos abordados: *exploitation*, *crypto*, *cracking*, *web security*;
- PHD CTF - a ronda de qualificação é realizada *online* no formato *Jeopardy* e a final é presencial no formato *Attack-defence*. O tamanho da equipa é limitado a 7. Tópicos abordados: *web security*, *crypto*, *forensics*, *scripting*, *reverse engineering*, *linux*;
- RuCTFe - é realizada no formato *Attack-defence*. Tópicos abordados: *SQL injection*, *binary exploitation*, *buffer overflow*, *reverse engineering*, *buffer overflow*;
- Hack.lu CTF - a ronda de qualificação é realizada no formato *Jeopardy* e a final é realizada no formato *Attack-defence*. Tópicos abordados: *Crypto*, *reverse engineering*, *forensics*, *web security*;
- SECUINSIDE CTF - formato *Jeopardy*. Equipas até 8 elementos. Tópicos abordados: *Crypto*, *reverse engineering*, *forensics*, *web security*;
- rwth CTF - é realizada no formato *Attack-defence*, não existe limite no número de participantes por equipa. Tópicos abordados: *Binary exploitation challenge*, *android cell phone challenge*, *cryptographic challenge*, *code war games*;
- CSAW CTF - a ronda de qualificação é realizada no formato *Jeopardy* e a final no formato *Attack-defence*. O tamanho da equipa não é limitado. Tópicos abordados: *Trivia* , *Recon*, *Web*, *reversing*, *exploitation*, *miscellaneous*, *crypto*;
- PICO CTF - formato *Jeopardy*. Equipas até 8 elementos. Tópicos abordados: *Crypto*, *reverse engineering*, *forensics*, *web security*;
- *IoT CTF by NULLCON* - foca-se na exploração de vulnerabilidades em dispositivos e protocolos IoT.

Capture the Flag Platforms

Apresentamos de seguida, as plataformas CTF existentes [58]:

- *CTFd* - plataforma para exercícios CTF, no formato *jeopardy* [60];
- *RootTheBox* - plataforma didáctica, com *CTF Scoreboard* e *Game Manager* [230];
- *Scorebot* - plataforma para exercícios CTF (criada pela equipa LegitBS - DEF CON) [211];
- *SecGen* - gerador de *"vulnerable virtual machines"*, *"lab environments"*, e *"hacking challenges"* [212];
- *echoCTF.RED* - plataforma para disponibilizar e gerir uma infraestrutura para exercícios CTF [215];

- *FBCTF* - plataforma CTF criada pelo *Facebook* [93];
- *Haaukins* - plataforma de virtualização (*Automated Virtualization Platform*) para exercícios CTF [119];
- *HackTheArch* - *scoreboard* para exercícios CTF [8];
- *Mellivora* - *lightweight plataform* para exercícios CTF [174];
- *NightShade* - *"simple security capture the flag framework"* [183];
- *LibreCTF* - plataforma CTF utilizada nas competições *EasyCTF* [161];
- *PicoCTF* - plataforma CTF utilizada nas competições *PicoCTF* [200];
- *mkctf* - *framework* para gerir e monitorizar exercícios CTF [178].

2.2 Tecnologias e ferramentas utilizadas

Recomendações para a dissertação

Para a redação da dissertação utilizei os seguintes materiais:

- *Template* em LaTeX do Departamento de Eng. Informática, Universidade de Coimbra (Nuno Lourenço) [74];
- *"O Pequeno Livro da Dissertação"* (Paulo Gil, Ana Relvas) - Livro de sugestões para realizar uma dissertação [104];
- *"Erros frequentes de escrita"* (Paulo Rupino) - Documento de boas práticas [74];
- *"Recomendações de Funcionamento"* (Edmundo Monteiro) - Objetivos e método de avaliação da dissertação [74];
- *"Planear, Preparar, Apresentar"* (Fernando Boavida) - Palestra com recomendações para comunicar ideias e/ou resultados do seu trabalho [20].

Overleaf

O Overleaf é um editor de LaTeX *online*, com colaboração em *"tempo real"* [194].

TeXstudio

TeXstudio é um editor de documentos LaTeX, com preenchimento automático e assistentes para manipular imagens, tabelas e fórmulas matemáticas [229].

Mendeley Desktop

Mendeley permite organizar as referências bibliográficas e gerar as mesmas automaticamente [175].

Let's Encrypt

Let's Encrypt é uma autoridade de certificação (CA), e também, um serviço prestado pela *Internet Security Research Group* (ISRG). *Let's Encrypt* permite criar os certificados digitais (SSL/TLS) gratuitamente e de forma automática [99].

Todas as páginas *web* e as Application Programming Interface (API) deste projeto utilizam *Let's Encrypt*.

Apache HTTP Server

O *Apache HTTP Server* é um servidor, de código aberto, eficiente e extensível que fornece serviços web [97].

Git

O *Git* é uma ferramenta de controlo de versões, que manipula os dados e os ficheiros como uma série de *snapshots* [33].

Um *commit* significa que estamos a guardar o estado desses dados. O *Git* grava uma imagem de como os ficheiros estão naquele momento e armazena uma referência para o *snapshot* criado.

Se o estado dos arquivos não mudarem entre os *commits*, então o *Git* usa *links* para os arquivos já armazenados. O *Git* opera como um fluxo de *snapshots*.

O *Git* mantêm todo o histórico do projeto e usa a função de *hash SHA-1*, para detetar as alterações no projeto.

Git Large File Storage

O módulo *Large File Storage* (LFS) permite que o Git suporte e manipule arquivos de qualquer tamanho [51].

GitLab

GitLab é uma plataforma *DevOps* focada no desenvolvimento, na segurança e nas operações em torno do ciclo de desenvolvimento de *software* [113].

Em [111] encontram-se descritas as funcionalidades do GitLab. As seguintes foram estudas e utilizadas no âmbito deste projeto:

- *Create* - permite gerir o projeto e realizar as mesmas operações *Git* a partir da interface *Web*, sendo que, também permite editar, rever e auditar o código através de uma *WebIDE*;
- *Verify* - permite executar *pipelines* que validam, testam e compilam o projeto;
- *Package* - suporta várias ferramentas de *packaging*, como o *Container Registry*;
- *Secure* - permite identificar vulnerabilidades no código, através de ferramentas de análise estáticas e dinâmicas, testes de *fuzzing* e análise de dependências;
- *Configure* - suporta várias ferramentas de configuração, como o *Kubernetes Management*. Esta ferramenta integra os *clusters Kubernetes* com o *GitLab*, a fim de optimizar os processos de desenvolvimento com *containers*;

GitLab Runners

GitLab Runner é uma aplicação, que executa os trabalhos da pipeline. Esta aplicação pode ser executada em qualquer *host* (máquina local ou *cloud*).

A Fig.2.4 apresenta o fluxo de execução do *GitLab Runner*, composto pela fase de registo e pela fase de trabalho.

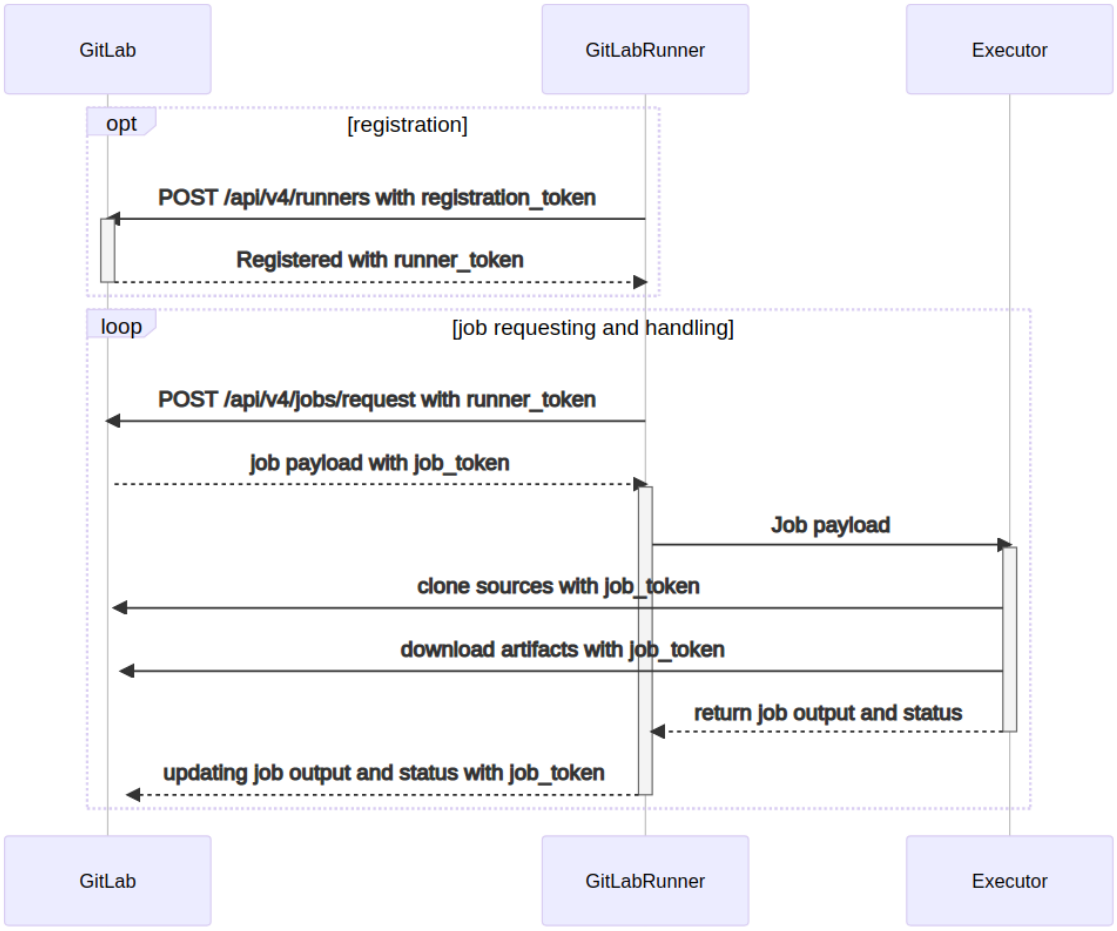

Figura 2.4: Fluxo de execução do GitLab Runner (Fonte: [114])

Na fase de registo, o *GitLab Runner* é registado na *dashboard* do *GitLab*, com um conjunto de permissões. O *GitLab* indica o estado do *GitLab Runner* (ativo, ocupado ou indisponível).

Na fase de trabalho (*job requesting and handling*), o *GitLab Runner* recebe o *payload* de trabalho, faz o *clone* do projeto *Git* e o *download* dos recursos necessários.

Os *logs* de execução da *pipeline* são apresentados na *dashboard* do *GitLab*. As *pipelines* para os projetos *Git*, na plataforma *GitLab*, são definidas através de ficheiros YAML, com a extensão *.gitlab-ci.yml*

Jenkins

O *Jenkins* é uma servidor de integração contínua desenvolvido em *Java*.

As *pipelines* do *Jenkins* automatizam a criação de *builds*, o *deploy* dos sistemas e a entrega dos projetos. Além disso, o *Jenkins* realiza tarefas e testes automatizados, executa análises de código e permite a gestão de *builds* em vários ambientes [139].

SonarQube

SonarQube é uma plataforma desenvolvida pela *SonarSource* que realiza análises estáticas ao código fonte. *SonarQube* tem como finalidade detetar vulnerabilidades, más práticas e *bugs*.

Os relatórios do *SonarQube* fornecem indicações sobre: vulnerabilidades, código duplicado, *coding standards*, *unit tests*, *code coverage* ou *code complexity* [216].

Rocket Chat

O *Rocket Chat* é um servidor de *chat*, que permite a colaboração em equipa através de vários canais. Suporta partilha de ficheiros, mensagens de texto, de *audio* ou de *video*, bem como, chamadas de áudio e vídeo com partilha de ecrã [39].

A REST API do *Rocket Chat* permite controlar e estender as suas funcionalidades. As tarefas de gestão podem ser realizadas pela API, nomeadamente a gestão de utilizadores, grupos, canais e o processamento de mensagens [38].

O *frontend* do *Rocket Chat* é personalizável e suporta múltiplas interfaces em simultâneo, com componentes diferentes. Em particular, a interface que pode ser uma *dashboard web*, um *widget* ou uma aplicação móvel.

Hubot

O *hubot* é o *bot* predefinido no *Rocket Chat*, que vem pré-programado com um conjunto de funcionalidades, definas em *scripts*. Ao desenvolver novos *scripts* personalizamos e capacitamos o robô, para desempenhar novas tarefas [129].

Docker

"Docker is essentially a toolkit that enables developers to build, deploy, run, update, and stop containers using simple commands and work-saving automation through a single API" [132].

Docker Registry

Docker Registry "is a stateless, highly scalable server side application that stores and lets you distribute Docker images" [82].

GitLab Container Registry

Com o *Container Registry* integrado no *GitLab*, cada projeto tem um espaço próprio para armazenar as imagens *Docker* [112].

Kubernetes

Kubernetes é uma plataforma de código aberto, que tem como objetivo gerir cargas de trabalho e serviços em *containers*.

Trata-se de uma plataforma de orquestração de *containers*, que permite a realização de operações de escala e automatizar diversos processos [83].

Kboom

O *kboom* [146] é uma ferramenta desenvolvida para o *Kubernetes* (semelhante ao *boom*), que permite gerar as seguintes cargas:

- *short-term load* - é um teste de escala, que permite identificar qual é a capacidade útil do *cluster*, quantos *pods* podem ser alojados e quanto tempo demora a carregar os mesmos;
- *long-term load for soak testing* - consiste em testar o sistema com um nível típico de carga, ao longo de um período de tempo, de modo, a verificar o comportamento do sistema em uso;

Tigera Calico

Tigera Calico é uma solução de rede e segurança para *containers*, máquinas virtuais e cargas de trabalho, que pode ser utilizada em várias plataformas (*Kubernetes*, *OpenShift*, *Docker EE*, *OpenStack* ou serviços em *bare metal* [29];

Cloud2 - DEI

O serviço Cloud2 é o sistema de virtualização do Departamento de Engenharia Informática, tratando-se da plataforma de virtualização *XCP-ng*.

Serviços disponibilizados e nomenclatura utilizada neste projeto

CTF@DEI é o nome atribuído à plataforma desenvolvida neste projeto.

O servidor "ctf"é responsável por disponibilizar os seguintes serviços:

- *Dashboard* e *scoreboard* da plataforma CTF@DEI, disponível em: ctf.dei.uc.pt;
- Sonar Qube, ferramenta que analisa as vulnerabilidades dos exercícios CTF, disponível em: ctfsonar.dei.uc.pt;
- Serviço de *chat*, que despoleta notificações automáticas, disponível em: ctfchat.dei.uc.pt;
- *Website* com a documentação e os materiais para os exercícios CTF, disponível em: ctfdocs.dei.uc.pt;

O serviço de *chat* disponibilizado, poderá ser utilizado através da aplicação móvel, disponível para *Android* e *iOS*, ou através da seguinte URL: ctfchat.dei.uc.pt.

O servidor "ctfgit"é responsável por disponibilizar a plataforma *GitLab*, a qual utilizamos para guardar a coleção de exercícios CTF. Encontra-se disponível em: ctfgit.dei.uc.pt.

O *GitLab Container Registry*, utilizado neste projeto, permite guardar os "objetos executáveis"(*Docker Images*). Encontra-se disponível através da seguinte URL: ctfgit.dei.uc.pt: 5050 (requer autenticação por *token*).

A URL ctfplay.dei.uc.pt é utilizada para expor os exercícios CTF. Por exemplo: ctfplay.dei.uc.pt:30050 e ctfplay.dei.uc.pt:30052 são exercícios CTF diferentes.

Os servidores "*kn0*", "*kn1*", "*kn2*"correspondem ao *cluster Kubernetes*.

CTF Toolkit é ferramenta em Command Language Interpreter (CLI), desenvolvida no âmbito deste projeto.

O servidor "*ctfkali*"corresponde à máquina CTFBOX, a qual é utilizada pelos alunos, e possui a *CTF Toolkit* e outras ferramentas para resolver os exercícios CTF.

Esta página foi intencionalmente deixada em branco.

Capítulo 3

Estado da arte

No presente capítulo abordamos o estado da arte sobre os exercícios CTF e as plataformas envolvidas. O objetivo deste capítulo consiste em compreender quais são as abordagens utilizadas nas competições CTF, os seus formatos e variantes, e como as podemos classificar e aplicar as mesmas.

Primeiro, apresentamos algumas informações básicas sobre as competições CTF. Em segundo, apresentamos um estudo compreensivo sobre as categorias dos exercícios CTF. Por último, analisamos as plataformas CTF existentes, apresentando as suas vantagens e desvantagens.

Este capítulo foi organizado em três secções:

- na secção 3.1, apresentamos o panorama geral das competições CTF;
- na secção 3.2, abordamos os formatos dos competições CTF, as suas aplicabilidades e os componentes utilizados;
- na secção 3.3, analisamos as categorias dos exercícios CTF;
- na secção 3.4, apresentamos algumas das plataformas CTF existentes;

3.1 O crescimento do Cybersecurity Capture the flag

Quando as ameaças eram compreendidas apenas por alguns especialistas, o desenvolvimento de código malicioso era um teste de conhecimento e habilidade técnica. *Creeper* foi o primeiro *Worm* da história, este exibia mensagens de provocação, algo considerado como recreativo e de baixo risco [90] [157].

Com o avanço e a disseminação da tecnologia, o risco das ameaças cresce, surgem novas oportunidades comerciais e conferências no âmbito da segurança informática [157]. Em 1993, embora *Jeff Moss* aos 18 anos, pretendesse criar um evento único, notavelmente, acabou por fundar uma das maiores conferências de segurança do mundo, a DEF CON [49].

A DEF CON, conferência em segurança informática, lançou várias competições na área de informática e eletrónica, com um grau de exigência elevado, o que atraiu organizações e especialistas talentosos [103]. A DEF CON supera-se nos anos seguintes, e pela primeira vez em 1996, o termo *Capture the Flag* é aplicado em competições de segurança.

Cybersecurity Capture the flag (CTF) designa uma competição que têm como objetivo a captura ou a defesa de uma bandeira, num dado ambiente. Em segurança, este ambiente envolve redes, servidores e serviços, e a bandeira é representada virtualmente por um *token*, uma cadeia de caracteres com uma estrutura, que permite que seja facilmente reconhecida [171].

Em 2012, com o objetivo de registar e divulgar as competições de CTF, surge uma plataforma, que se vulgarizou como *scoreboard* mundial, chamada *CTFTime* [62].

Com base nos dados públicos recolhidos a partir do *CTFTime*, entre 2012 e 2019, representamos na Fig.3.1, a evolução no número de competições anuais registadas na plataforma *CTFTime*. Os dados recolhidos mostram que as competições CTF crescem em popularidade, e em 2019, foram registadas cerca de 198 competições.

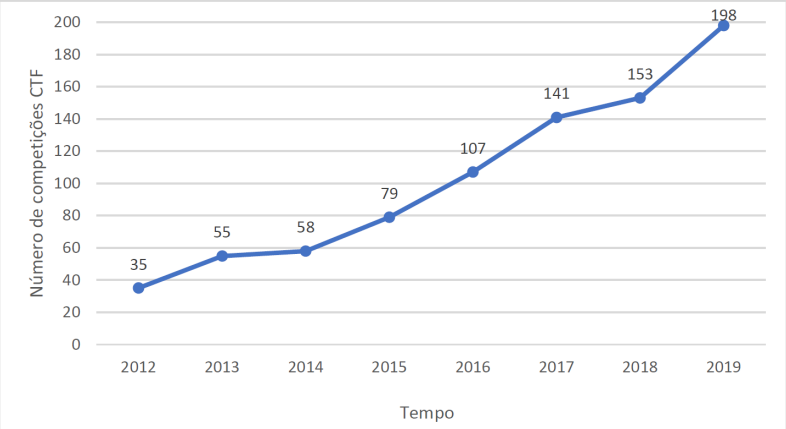

Figura 3.1: Eventos CTF registados na plataforma CTFTime entre 2012 e 2019

Além disso, a importância para as agências e organizações aumentou, uma vez que, fazem parte das competições CTF, a *European Union Agency for Cybersecurity* (ENISA) [242], a *Google*, o *Facebook*, a *Trend Micro*, a *Kaspersky*, a *Multicert*, a *Deloitte*, entre muitas outras organizações respeitadas.

Em 2021, as competições mais significativas para o *scoreboard* mundial do CTFTime são *DEF CON*, *PlaidCTF*, *MMA CTF*, *RuCTFE*, *Hack.lu CTF* e *0CTF*. Algumas competições envolvem prémios monetários, como a competição da Google em 2021, com um prémio de 13.337 US dólares para os vencedores do primeiro lugar [118].

Os fatores que contribuíram para o crescimento das competições estão relacionados com o crescimento das ameaças e com a necessidade de especialistas nestas áreas [103] . Em 2021, o panorama apresentado pela *F-secure*, indica um desenvolvimento e crescimento notável em *malware*, com o aparecimento repentino de novas técnicas de extorsão e espionagem. Entre as quais, o chamado *Ransomware 2.0*, com furto e encriptação dos dados confidenciais, paralisação total do negócio, com resgate em criptomoedas e pressão em ameaças [91].

As universidades começaram a reagir a esta necessidade [40], porém foram encontrados obstáculos no que toca à educação relacionada à segurança informática [135]. Ativamente, os professores reforçam a importância da auto-aprendizagem (*self-learners* e *Learning by doing*) [126], afirmando que a aprendizagem poderá ser motivada por enigmas, jogos e competições CTF (Processo de Gamificação).

Investigadores mencionam nos seus estudos, que os jogos de segurança e as *testbeds* podem

aproximar a aprendizagem com os sistemas reais, o que complementa as aulas teóricas. Conceitos como *Game Design*, *Gamificação* e *Active Learning* são aplicadas no ensino da segurança informática [13] [140] [17].

Apesar de as competições CTF terem-se destacado na segurança informática, existem diversos videojogos onde este conceito já foi aplicado. Em 1999, o videojogo *Unreal Tournament* (*First-Person Shooter*) tornou-se popular através dos seus modos *online* competitivos, incluindo o *Capture the Flag* [238]. Neste modo, os jogadores trabalham em equipa para capturar a bandeira adversária e devolvê-la à sua própria base [179].

A transposição de elementos ou técnicas para cenários de não-jogo é conhecida como *Gamificação* [77]. A partir de 2010, este termo vulgarizou-se e foi aplicado em processos empresariais, *marketing* e na educação, com o principal objetivo de motivar o diálogo, a autonomia e a criatividade dos participantes [219].

Em teoria, enquanto os jovens jogam videojogos, assumem responsabilidades, enfrentam desafios e concentram-se, com uma postura de resistência e coragem [48] [173]. Em 2010, *game designers* afirmam que os jogadores são motivados a solucionar problemas e a tentar de novo após o fracasso, aspetos que podem ser aplicados em outras áreas, incluindo a educação [172].

Em torno dos videojogos competitivos surgem comunidades activas de jogadores, que têm impacto na construção de competências sociais e no desenvolvimento dos jovens [1].

Entre 2007 e 2021, surgem vários estudos e publicações (ver apêndice 10), que abordam as experiências vivência-das no ensino da segurança informática, com recurso às competições CTF, *cyber ranges*, *wargames* e *cybersecurity testbeds*. Estas publicações, aproximam os conceitos de gamificação e *Learning-by-doing* à formação e ao ensino de competências técnicas - *Cyber Security Education* (Conjunto de palavras-chaves utilizadas na recolha dos artigos: *cybersecurity*, *competition*, *education*, *Capture the Flag*).

Na secção seguinte, abordamos as competições CTF e os seus formatos.

3.2 As competições CTF e os seus formatos

No contexto da presente dissertação, utilizaremos o conceito *Capture the Flag* no contexto da segurança informática, embora seja também aplicado a jogos de vídeo ou no desporto ao ar livre.

CTF é um exercício, cujo o objetivo é encontrar ou proteger as *flags* (bandeiras), que se encontram numa determinada rede de computadores, servidores e sistemas informáticos.

Como referido em 3.1, a primeira competição de CTF foi organizada durante a conferência DEF CON, em 1996. A competição DEF CON CTF é conhecida pelo seu grau de exigência e atrai um grande número de especialistas e organizações [103].

A DEF CON CTF é uma competição prestigiada (encontra-se no topo do *ranking* do *CTF-Time*), e destina-se às melhores equipas do mundo, o que justifica, a elevada dificuldade da ronda de qualificação (onde apenas um exercício CTF pode demorar muitas horas a ser resolvido).

As melhores equipas da DEF CON são selecionadas para organizar as próximas competições. A Tabela 3.1 apresenta os organizadores da DEF CON CTF ao longo dos anos.

Tabela 3.1: Organizadores da competição DEF CON CTF

Organizadores	Anos
Ghetto Hackers	2002 - 2004
Kenshoto	2005 - 2008
DDTek	2009 - 2012
Legitimate Business Syndicate	2013 - 2017
Order of the Overflow	Desde 2018

Tal como as competições de programação, as competições CTF tem como propósito, não só trazer questões complexas da área de segurança informática, mas também, promover a aprendizagem e capacitar os participantes para esta área.

Assim, as competições são diversificadas, sendo que, o nível de dificuldade varia consoante o público alvo (e a sua faixa etária) e os organizadores (equipas, organizações, universidades).

Para perceber esta diversidade, identificamos os formatos principais dos exercícios CTF (existem outras variantes, que resultam da combinação destes formatos principais). Existem dois formatos essenciais, que são o *Jeopardy* e o *Attack/Defence*, sendo que, também apresentamos uma variante destes dois, o formato *King of the Hill*.

Exercícios CTF no formato Jeopardy

Neste formato, os exercícios CTF são isolados, e como tal, estes podem ser resolvidos por qualquer ordem, sem prevalências (salvo raras excepções). As competições CTF, neste formato, envolvem uma lista de exercícios (ver Fig.3.2), a qual, tipicamente, é representada por uma grelha (componente denominado por *scoreboard*).

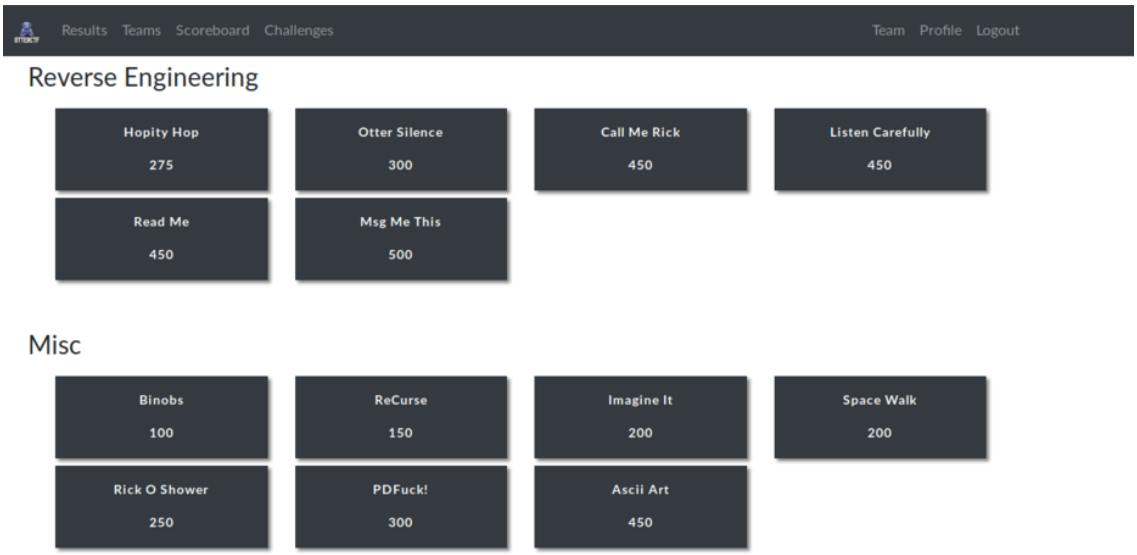

Figura 3.2: Competição CTF no formato Jeopardy (Fonte: [248])

Os exercícios CTF distinguem-se pela sua categoria (assunto abordado na secção 3.3) e pelo nível de dificuldade, este nível é representado através de pontos, ou seja, os exercícios CTF que valem mais pontos são também os mais difíceis.

O formato *Jeopardy* é realizado tanto individualmente como em equipas (existem competições para diversas faixas etárias: 12 anos, 16 anos, 25 anos, *seniores*/PhD). Este formato

poderá ser realizado de forma presencial, apenas *online* ou híbrida.

A ronda de qualificação da DEF CON é realizada no formato *Jeopardy* em modo *online*.

O objetivo dos participantes numa competição CTF, no formato *Jeopardy*, consiste em obter a maior quantidade de pontos durante a competição. No fim da competição, o(s) participantes(s) com mais pontos são os vencedores.

Para pontuar, os participantes tem de aceder ao *scoreboard* e solucionar os exercícios propostos. A maioria dos exercícios realiza-se, em torno de uma rede de computadores, serviços e aplicações vulneráveis. Outros exercícios são realizados em conjunto com as organizações e as entidades parceiras, que realizam atividades com os participantes (*quizzes*, enigmas e outros jogos didáticos).

No formato *Jeopardy*, os participantes não podem prejudicar os adversários, nem o servidor da competição (*scoreboard*). A determinada altura da competição, é recorrente existir cooperação entre as equipas, troca de ideias e soluções (esta cooperação é um comportamento tipicamente aceitável, mas depende das regras da competição).

Exercícios CTF no formato Attack/Defence

Em contraste com o *Jeopardy*, o formato *Attack/Defence* destina-se sobretudo à competição por equipas. Neste formato, as *flags* identificam as equipas e possuem uma data de expiração (*timestamp*).

A final da competição DEF CON é realizada no formato *Attack/Defence*, em modo presencial, em *Las Vegas* (*Nevada*).

Cada equipa é responsável por gerir uma infraestrutura independente, com várias vulnerabilidades. A qual é composta por *hardware* (SCADA, ICS, IoT, Cloud) e *software* (servidores, serviços, máquinas virtuais). O cenário de uma equipa pode ser igual ou diferente do cenário da outra equipa (estes cenários são denominados na literatura por *cyber ranges*, ver capítulo 2).

As equipas tem de reconhecer e compreender o ambiente informático, de forma autónoma (não existem enunciados ou guias). Para além das competências técnicas, as equipas necessitam de *soft skills*, que lhes permitam comunicar, cooperar, reorganizar e agir perante qualquer situação imprevisível.

No formato *Attack/Defence*, as competições são realizadas através de rodas, e existe um momento de validação (ou *tick*).

Durante as primeiras rondas (fase de *warmup*), as infraestruturas das equipas estão isoladas. Nesta fase, os participantes analisam a sua infraestrutura, a fim de, enumerar os serviços e identificar vulnerabilidades existentes na sua infraestrutura.

Neste formato, também é comum, existir uma *cyber range*, que é comum a todas as equipas, destinada a efeitos de testes. Na fase de *warmup*, os participantes realizam ataques livres (*scan*, DoS) nesta *cyber range* comum. Nas restantes rondas, a *cyber range* comum é utilizada para o desenvolvimento e teste de *exploits* (fora da zona de ataque).

Após a fase de *warmup*, as redes das equipas deixam de estar isoladas, uma nova rede une os participantes (zona de ataque). Sendo frequente que, durante a competição, os organizadores facultem a captura desta rede a todas as equipas (por exemplo, através de ficheiros *.pcap*).

As equipas tem vários objetivos a decorrer em paralelo, que são os seguintes:

- *Availability* - garantir a disponibilidade dos serviços;
- *Defence* - proteger as *flags* dos ataques adversários, que são realizados pelas equipas adversárias (e por vezes, pelos organizadores);
- *Attack* - atacar os sistemas dos adversários. Por um lado, capturar a *flag* do adversário e submete-la no *scoreboard*, por outro, sabotar o sistema adversário, substituindo a *flag* existente, por outra, que identifica a equipa que atacou o sistema.

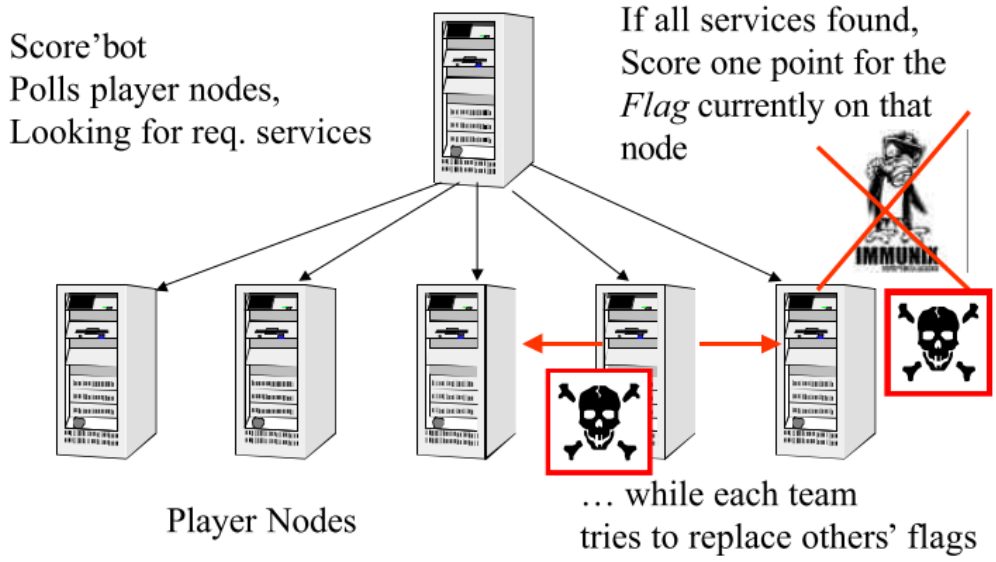

Figura 3.3: DEF CON Capture the Flag (2003) (Fonte: [54])

Em cada ronda, e de forma aleatória, ocorre o momento de validação (ver Fig.3.3), no qual, o *scoreserver* (ou *scorebot*) verificam o seguinte conjunto de aspetos:

- verifica-se, se os serviços estão disponíveis e a responder corretamente, sendo que, as equipas são penalizadas, se as suas funcionalidades não estiverem disponíveis no momento de validação da ronda;
- verifica-se, se as *flags* podem ser obtidas (do ponto de vista das equipas adversárias), se sim, significa que a equipa ainda não corrigiu as vulnerabilidades do seu sistema;
- verifica as *flags* existentes nos sistemas das equipas, como a *flag* identifica uma equipa, os pontos vão para a equipa determinada pela *flag* existente, no momento de validação da ronda.

No fim da competição *Attack/Defence*, a equipa com mais pontos é a vencedora.

Exercícios CTF no formato King of the Hill (modelo híbrido)

No formato *King of the Hill* existe um cenário, uma infraestrutura com um conjunto de servidores pré-configurados e sem nenhuma equipa associada.

Por exemplo, a Fig.3.4, uma rede simétrica, disputada por 4 equipas (representas por cores). Envolve tarefas, como corrigir vulnerabilidades, *Pivoting* (obter acesso a mais máquinas, através de uma máquina comprometida) ou *Implants* (introduzir *backdoors*).

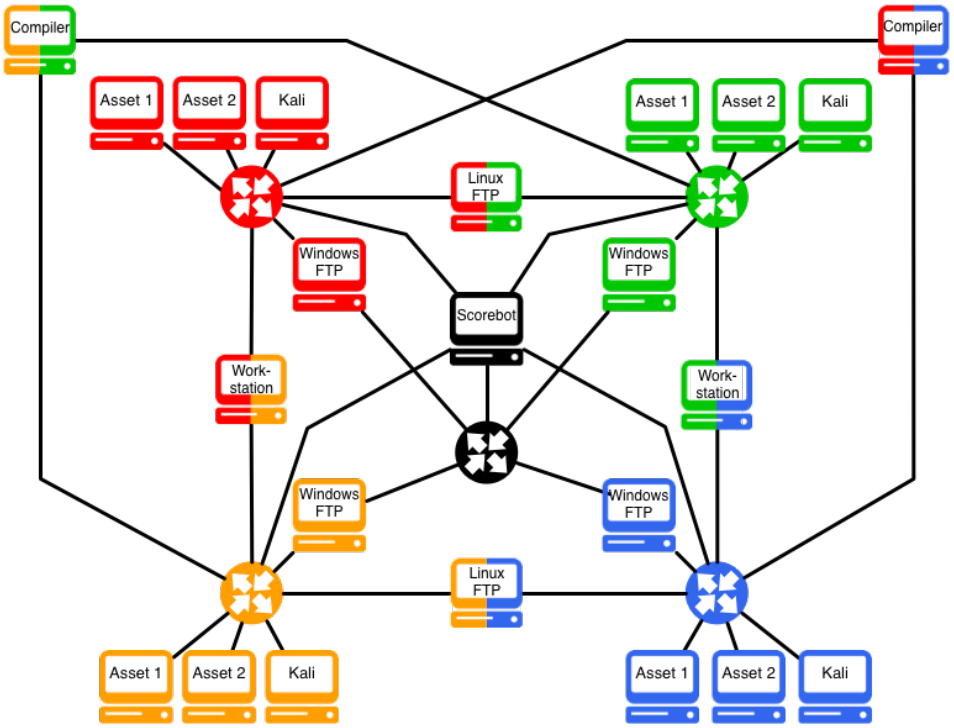

Figura 3.4: King of the Hill's Cyber Range (Fonte: [186])

Tal como no formato *Jeopardy*, os participantes tem de recuperar as *flags* e submete-las no *scorebord*. Porém, ao fazê-lo, os participantes obtém acesso ao servidor.

No formato *King of the Hill*, para além de capturar a *flag*, as equipas substituem-na por um nova, marcando a sua posição. Após, a captura do servidor, a equipa tem de desenvolver correções para os exercícios CTF, de modo, a implementar mecanismos para se defender dos adversários.

O cenário *King of the Hill* pode mudar ao longo do tempo, isto é, durante a competição podem surgir novos exercícios CTF no *scorebord*, o que cria novas oportunidades para recuperar os servidores. Os organizadores podem introduzir vulnerabilidades nos servidores existentes ou adicionar novos servidores à infraestrutura.

As equipas pontuam por manterem a posse de um servidor, ao longo das rondas. Em todas as rondas, de forma aleatória, o sistema irá verificar as *flags* presentes em cada serviço. Os pontos serão atribuídos em função destas *flags*.

Detalhes sobre as Flag

Como indicado, os exercícios CTF exigem a captura de uma *flag*. A *flag* é um sinalizador, que é usado como prova, de que os participantes atingiram ou protegeram o seu objetivo.

Em segurança informática, as *flags* são *strings*, e podem ser criadas de diversas maneiras, com os seguintes aspetos [70]:

- texto ou informações - quando composta por uma frase sobre o exercício ou quando formada por uma *string* longa e aleatória;
- *timestamp* - podem conter a data e a hora (validade da *flag*) ou conter o instante em que a *flag* foi gerada;

- *host* ou serviço - quando identificam um *host* ou um serviço;
- *team* - quando identificam uma equipa em particular;

As aplicabilidades dos exercícios CTF

De acordo com o formato utilizado e a escala do evento, existem várias aplicações para os exercícios CTF, nomeadamente:

1. em competições de segurança informática - as competições CTF, com o propósito de treinar os participantes, preparar uma equipa e desenvolver habilidades e conhecimentos técnicos [144];
2. em eventos CTF ligados às organizações - os exercícios CTF são utilizados pelas organizações, para vários fins. Por exemplo, o desenvolvimento das competências dos colaboradores ou para fins de recrutamento e seleção de profissionais;
3. em jogos didáticos de segurança informática (*hackathon*, *wargames*, *workshops*) - os exercícios CTF são utilizados em *workshops*, em palestras e também em demonstrações;

Os componentes dos exercícios CTF

Os exercícios CTF envolvem vários componentes, que variam de acordo com o âmbito e com o cenário do evento. Os componentes disponibilizados nas competições CTF são tipicamente os seguintes:

- *Scoreboard* e *Scoreserver* - servidor do exercício CTF, que tem a responsabilidade de atribuir os pontos aos participantes;
- *Game Database* - base de dados, onde são guardadas as informações do jogo (participantes, exercícios, submissões);
- Plataformas, redes, infraestruturas - corresponde ao cenário, que é concebido e entrega aos participantes;
- Sistemas físicos - corresponde a equipamentos físicos. Por exemplo, servidores ou outros equipamentos com vulnerabilidades, deliberadamente mal configurados;
- Sistemas virtuais - maquinas virtuais ou *containers*, que disponibilizam serviços ou aplicações vulneráveis;
- *Team Servers* - servidores que interligam os participantes, aos sistemas físicos e/ou virtuais;

3.3 Categorias dos Exercícios Capture the Flag

Na presente secção, propomos uma taxonomia com as categorias predominantes nos exercícios CTF.

Os exercícios CTF dedicados à segurança informática dividem-se em categorias, no formato *Jeopardy*, é frequente existir esta distinção, uma vez que, os exercícios são independentes

(ver Fig.3.2). Nas competições *Attack/Defence*, os exercícios CTF combinam estas mesmas categorias.

Independente do formato de uma competição CTF (*Jeopardy* ou *Attack/Defence*), os exercícios CTF distinguem-se entre si, na medida em que, envolvem conceitos, técnicas e ferramentas distintas.

Ao longo desta secção abordamos individualmente cada categoria de exercícios, os principais conceitos e ferramentas envolvidas (esta taxonomia poderá ser utilizada como referência, por exemplo, para classificar os exercícios CTF).

A taxonomia, que propomos para classificar os exercícios CTF, ramifica-se nas seguintes categorias:

- 3.3.1 - *Web Security*;
- 3.3.2 - *Criptografia*;
- 3.3.3 - *Binary Analysis and Reverse Engineering*;
- 3.3.4 - *Networking*;
- 3.3.5 - Computer and Mobile Forensics;
- 3.3.6 - Open Source Intelligence;
- 3.3.7 - *Professional Programming Challenges*;
- 3.3.8 - Miscellaneous;

De seguida, detalhamos cada uma destas categorias, a fim de determinar quais são os conceitos, as técnicas e as ferramentas, alvo de estudo, nos exercícios CTF.

3.3.1 Categoria Web Security - *Web Security* (Web)

A categoria *Web Security* explora as falhas na segurança das aplicações *web*, isto é, aplicações acedidas num *browser* e executadas num servidor *web*. Esta categoria envolve aspetos de diferentes tecnologias, como serviços *web*, sistemas de arquivos e base de dados. Neste domínio, os exercícios envolvem intrusões no sistema, furto de dados sensíveis e realização de operações não autorizadas.

As técnicas desta categoria relacionam-se com o problema principal, de que os utilizadores podem submeter *inputs* arbitrários, pelo que, são abordados mecanismos de segurança aplicados às aplicações *web*, tais como, a gestão das sessões dos utilizadores e o processamento dos pedidos e das respostas da aplicações.

Os exercícios CTF tipicamente envolvem uma componente de reconhecimento, que permite identificar qual é o problema com a aplicação, que superfície de ataque existe (informações que iram permitir traçar uma estratégia para resolver o exercício).

Para apresentar as subcategorias principais e as diferentes abordagens e metodologias aplicáveis a esta categoria de exercícios CTF, construímos um *Diagrama de Ishikawa* (ver Fig.3.5).

Reconhecemos que as subcategorias de estudo da categoria *web* são as seguintes:

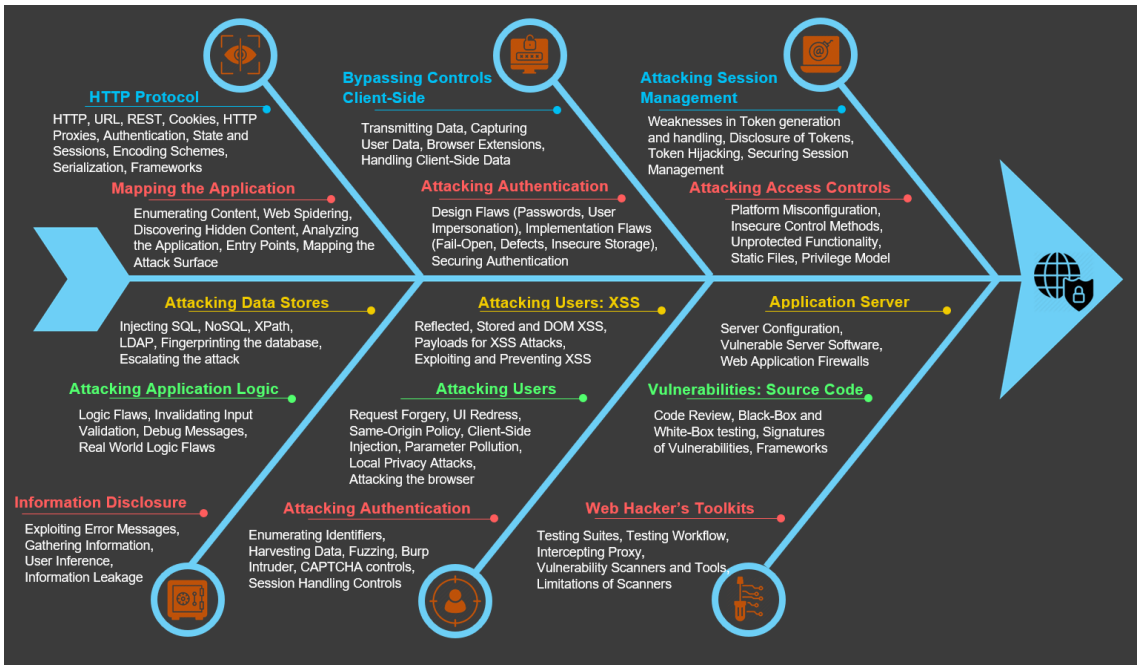

Figura 3.5: Categoria Web Security - Diagrama de Ishikawa (*Template* e ícones: [155] [95])

HTTP Protocol, Mapping the Application: Em *Websecurity*, o reconhecimento sobre um *website* é realizado através de uma análise ao seu conteúdo, enumeração dos métodos de requisição HTTP (dos diferentes *inputs* entre o cliente e o servidor), dos serviços e das tecnologias utilizadas. O escopo do exercício CTF, nesta categoria, foca-se num aspeto de segurança em particular: causas que comprometem a segurança da aplicação *web* (normalmente deriva de uma vulnerabilidade, uma falha no design ou na sua implementação).

Bypassing Controls Client-Side: Os controlos de segurança implementados do lado cliente dividem em duas abordagens principais. Por um lado, a aplicação realiza as suas verificações do lado do cliente, então este controlo falha, porque qualquer controlo do lado do cliente, pode ser contornado. Por outro lado, a aplicação transmite dados por meio do cliente, partindo do pressuposto de que não foram modificados, sendo que estes dados podem ser manipulados sobre diferentes protocolos e tecnologias (alguns exemplos são a injeção de *JavaScript* ou a manipulação de estruturas XML/JSON e outros objetos).

Attacking Authentication: Os controlos responsáveis por autenticar o utilizador não só envolvem o *login* dos utilizadores, como também outras funções periféricas relacionadas, como o registo, a recuperação da conta ou a alteração da password. Sobre a autenticação, existe uma grande variedade de vulnerabilidades, tanto no *design* da solução, como na própria implementação. Esta subcategoria de exercícios CTF envolve, problemas simples, relacionados com passwords fracas e *bruteforce*, e problemas mais complexos, que se relacionam com falhas em bibliotecas e serviços. Por exemplo, detalhes na lógica dos serviços de autenticação ou falhas na manipulação e transmissão de *tokens*.

Attacking Session Management: Os controlos responsáveis por gerir as sessões dos utilizadores fazem parte dos exercícios CTF. Em particular, os mecanismos *stateful*, que permitem que o servidor identifique cada utilizador inequivocamente, em pedidos diferentes. As falhas nestes controlos permitem mascarar ações entre utilizadores, sem saber as credências dos mesmos.

Attacking Data Stores: Os exercícios CTF focados nos sistemas de armazenamento,

envolvem um grande gama de técnicas e diferentes vulnerabilidades associadas. Este ramo também foca-se sobretudo na injeção de *inputs*, em canais Out-of-band (OOB), inferências e *time delays*. Os sistemas de armazenamento mais utilizados nos exercícios CTF são as bases de dados (*MySQL*, *MS-SQL*, *Oracle*), porém existem exercícios que envolvem outras tecnologias, como o *NoSQL*, o *XPath* ou mesmo o *LDAP*.

Attacking Application Logic: A lógica da aplicação carateriza outra subcategoria de problemas CTF, que abrange uma grande variedade de falhas individuais, que podem culminar em vulnerabilidades, como SQL Injection (SQLi) ou Cross-Site Scripting (XSS).

Application Server: Sobre os servidores de *backend*, os exercícios CTF envolvem vulnerabilidades relacionadas com os sistemas operativos, injeção de comandos *Linux*, injeção de *scripts* das linguagens *web* ou mesmo executáveis (*web backdoor*). Esta subcategoria pode envolver outros serviços e protocolos, como Simple Object Access Protocol (SOAP), injeções de XML/JSON, manipulação de pedidos HTTP no *backend* e diversos serviços, como email, *Kerberos* ou *Active Directory*.

Attacking Users: Alguns exercícios incidem na manipulação das ações dos utilizadores, utilizando a aplicação *web* para atingir esse fim. Em detalhe, existem vulnerabilidades como o redireccionamento do utilizador ou o Cross-site Request Forgery (CSRF), mas também diferentes técnicas e abordagens, que permitem manipular os pedidos do utilizador, fixar a sua sessão, obter dados do *browser*, explorar *bugs* e vulnerabilidades locais (*cliente-side SQL injection*).

A seguinte lista apresenta abordagens mais avançadas, que se aplicam e estão interligados aos exercícios desta categoria, nomeadamente:

- ***Information Disclosure***: Corresponde a um conjunto de métodos e práticas, que se focam em obter informações confidencias. Durante a monitorização de um ataque focam-se nas fontes de informação existentes. Os problemas da aplicação, como anomalias ou mensagens de erro, expõem o funcionamento interno da aplicação, o que possibilita um ataque para extrair as informações guardadas pela aplicação;

- **Automatização de ataques**: A maioria dos ataques são personalizados e de alguma forma, envolvem a lógica da aplicação. Executar as solicitações manualmente é um trabalho repetitivo, pelo que a criação de *scripts* permite automatizar parte deste trabalho. Em alternativa existem ferramentas, como o *Burp Suite* e o *OWASP ZAP*, que permitem monitorizar as respostas das aplicações e automatizar grande parte destes ataques;

- **Aplicações nativas e compiladas**: Existem diferentes tecnologias e linguagens, que se aplicam às aplicações *web*. Algumas aplicações são escritas e compiladas em C/C++, podendo envolver vulnerabilidades relacionadas com a linguagem de programação em si. Por exemplo, algumas aplicações *web* são vulneráveis a formatos de dados inválidos e a *Buffer Overflows*;

- **Arquitetura das aplicações**: Em aplicações que possuem diferentes componentes e camadas, a exploração de um defeito num dos componente pode afetar os outros componentes. Estas situações acabam por afetar a aplicação no seu todo, o ambiente de produção, e por sua vez, ameaçam as infraestruturas *cloud* e os os ambientes destes provedores.

- **Servidor Web**: As aplicações estão ligadas aos servidores *web*, como o *Apache* ou o *Nginx*. Inevitavelmente, a aplicação pode ser totalmente afetada, por causa

de defeitos na segurança do servidor *web*, por exemplo, falhas nas configurações ou problemas do próprio *software*.

- **Análise ao código**: Nalguns exercícios de CTF é possível rever manualmente o código e identificar as vulnerabilidades através de metodologias e ferramentas de analise de código. Principalmente nos exercícios CTF que incluem o código fonte da aplicação.

- ***Web Application Toolkits***: Existem várias ferramentas e recursos que auxiliam a identificar as vulnerabilidades, e que permitem explorar as mesmas de forma automatizada;

- ***Web Application Hacking Methodologies***: De forma estratégica, estas metodologias organizam as técnicas e os procedimentos em tarefas individuais, por meio de listas de verificação e árvores de ataque.

Esta categoria é destinada a formar competências na segurança de aplicações web, sendo útil para *developers*, administradores de sistemas e *websites*. Dentro deste espetro, existe uma grande variedade de exercícios CTF aplicados às vulnerabilidades *web*, detalhes nas linguagens de programação, bibliotecas, tecnologias, *frameworks*, sistemas de gestão de conteúdo (*WordPress*, *Joomla*, *Drupal*) e base de dados.

Existem algumas técnicas importantes durante a resolução de cenários CTF. Neste contexto, a seguinte lista de considerações permite sistematizar e avaliar uma aplicação *web*:

- Enumerar as páginas do *website* e procurar por áreas ocultas;

- Enumerar e reconhecer os *inputs* dos clientes/utilizadores;

- Testar os *inputs* e verificar o tempo de resposta do servidor;

- Analisar eventuais erros provenientes do servidor, base de dados, *frameworks* ou *pulgins* da aplicação;

- Reconhecer as vulnerabilidades *web* mais comuns (*OWASP Top 10*): *path traversal*, *session manipulation*, SQLi, Remote Code Execution (RCE), XSS, Local File Inclusion (LFI), Remote File Inclusion (RFI), CSRF, Server Side Template Injection (SSTI);

- Recorrer a ferramentas de deteção e exploração (*Burp Suite*, *Sqlmap*, *Nmap*);

- Identificar soluções de segurança (*Security controls*) como Web Application Firewall (WAF) ou *Honeypots*;

- Analisar o tráfego da rede (*Wireshark*, *Nmap*) e recorrer a *web proxies*;

- Conhecer os protocolos de comunicação e de segurança (HTTP, IPv6, OAuth);

- Conhecer as várias estratégias de *pentesting* (análise estática, análise dinâmica)

(Fontes de informação utilizadas para recolher os conceitos apresentados na presente secção 3.3.1: [218] [196] [201])

3.3.2 Categoria Criptografia

Na categoria de criptografia, chamada *cryptanalysis* ou simplesmente *crypto*, os exercícios CTF focam-se nos conceitos de criptografia, nas falhas de implementação, nas fraquezas dos algoritmos e no *design* dos protocolos. Os exercícios CTF disponibilizam serviços de criptografia, dados encriptados e funções criptográficas implementadas.

Dentro desta categoria, os problemas envolvem uma componente de criptografia, mas esta área de problemas expandiu para um campo multidisciplinar. A resolução de problemas nesta categoria, envolve também os aspetos relacionados com redes de computadores, eletrónica, arquitetura de computadores. Os exercícios CTF fazem uma conexão entre os modelos de criptografia e as implementações reais aplicadas em dispositivos físicos. Assim, esta categoria pode ser fragmentada em quatro subcategorias principais: fundamentos, criptografia simétrica, criptografia assimétrica e aplicações.

Na subcategoria dos fundamentos de criptografia, os exercícios CTF envolvem os algoritmos mais clássicos, como *Caesar* e *Vigenère*, assim como os conceitos de permutação, de substituição e o *one time pad*. A seguinte lista enumera os diferentes fundamentos presentes nesta subcategoria:

- ***Authenticated encryption*** (AE): Corresponde a um conjunto de funções de criptografia simétrica, para além de retornar o texto cifrado, retorna também uma *tag*, que garante a integridade da mensagem de texto;
- ***Format-preserving encryption*** (FPE): Corresponde às funções criptográficas que encriptam a mensagem, mas que mantém o mesmo formato de entrada. Por exemplo, uma FPE encripta um endereço IP e preserva a sua estrutura;
- ***Fully homomorphic encryption*** (FHE): Conjunto de funções que permitem substituir segmentos criptografados por outros segmentos igualmente criptografados, sem a necessidade de desencriptar todo o documento;
- ***Searchable Encryption***: Conjunto de funções que permitem realizar *queries*, em que os termos de pesquisa encontram-se encriptados, sobre dados em base de dados que estão também encriptadas. Tal como as funções FHE, ocultar os dados e as pesquisas permitirá melhorar a privacidade das aplicações;
- ***Tweakable encryption*** (TE): Conjunto de funções de encriptação, que aceita um parâmetro adicional, chamado de *tweak*. As funções TE possuem diversas aplicabilidades, por exemplo, nos processos de encriptação dos discos, onde a função TE usa um valor que depende da posição dos dados criptografados, normalmente, o número do sector ou o índice do bloco (o qual é definido como *tweak*).

Em Criptografia, as restantes subcategorias de problemas CTF focam-se sobretudo em problemas relacionados com os algoritmos de criptografia existentes. No caso da criptografia simétrica, os conceitos abordados são os seguintes:

- ***Block Ciphers***: Este tema engloba dois algoritmos principais baseados em blocos, o DES e o AES. Os exercícios CTF consideram vários detalhes, como as propriedades das chaves e dos blocos. Existem diversos modos de operar os blocos, em particular, o modo Electronic Codebook (ECB), o modo Chipher Block Chaining (CBC) e o modo Counter (CTR).

- ***Stream ciphers***: As *stream ciphers* consistem num fluxo de *bits* submetidos à operação XOR (ou outra operação), com as mensagens que vão ser encriptadas. As implementações em *hardware* dedicado são realizadas sobre *application specific integrated circuits* (ASIC), *Programmable Logic Devices* (PLD) e *Field-Programmable Gate Arrays* (FPGAs), como exemplo, existe a *stream cipher Grain-128a* e a A5/1. De forma similar, existem soluções em *software* implementadas em CPU's modernos, como o RC4 (que foi utilizado nas primeiras redes Wi-Fi) e o Salsa20 (*counter stream cipher*).
- **Funções de *Hash***: Nas funções de *hash* existem duas situações a explorar: a resistência à pré-imagem (garantia de que não é possível encontrar o *input* a partir de um dado *hash*); e as colisões (diferentes *inputs* que produzem o mesmo *hash*).
- ***Keyed Hashing***: *Keyed Hashing* resulta da combinação de funções de *hash* com uma *secret key*. Existem dois tipos de algoritmos *message authentication codes* (MAC), que autentica uma mensagem e protege a sua integridade, e as *pseudorandom functions* (PRF), que produzem conteúdo pseudo-aleatório.
- ***Authenticated Encryption***: Existem algoritmos que podem encriptar e autenticar uma mensagem, nomeadamente os MAC's e as *Authenticated ciphers* (AES-GCM).

Inerentemente à criptografia assimétrica, surgem os conceitos de chave pública. Esta subcategorias de problemas CTF foca-se sobretudo nos seguintes algoritmos e suas variantes:

- RSA: O sistema de *Rivest, Shamir e Adleman* que revolucionou a criptográfica em 1977, funciona através da criação de um objeto matemático chamado de *trapdoor permutation*. Uma função que transforma um número x em y no mesmo intervalo, onde calcular y de x é fácil, mas calcular x de y é praticamente impossível – a menos que seja conhecida a chave, chamada de *trapdoor*;
- Diffie–Hellman: Em 1976, *Diffie e Hellman* introduziram um sistema de *public-key distribution*, um protocolo que permite dois clientes estabelecerem uma chave secreta, em ambientes partilhados com terceiros;
- Elliptic curves: A introdução de *Elliptic curve cryptography* (ECC) em 1985 (método mais eficiente que o RSA e o Diffie-Hellman), e tal como o RSA, o ECC multiplica grandes números, mas de forma a combiná-los numa curva matemática, chamada elliptic curve.

Por último, importa referir também, aplicações criptográficas, estas baseadas nos conceitos referidos anteriormente. Por exemplo o protocolo *Transport Layer Security* (TLS), que protege as conexões entre servidores e clientes, ou a tecnologia *Blockchain*, que é considerada como um protocolo da confiança, ou os *Smart Contract*, que atuam como um acordo entre duas pessoas na forma de código.

(Fontes de informação utilizadas para recolher os conceitos apresentados na presente secção 3.3.2: [9] [217])

3.3.3 Categoria Binary Analysis and Reverse Engineering

A categoria de engenharia reversa (*Reverse Engineering*) é utilizada para analisar *malware*, detetar vulnerabilidades e *rootkits* e auxiliar no design de antivírus (e outros *Security controls*).

A analise de binários, de processos e de *memory dumps* fazem parte desta categoria de exercícios CTF, assim como a analise da execução de um programa, depuração interna, implementação de mecanismos de proteção em binários, análise de antivírus e análise forense de um *kernel*.

Enumeramos os principais conceitos utilizados nos exercícios CTF, no âmbito da categoria engenharia reversa (note que, focamo-nos essencialmente o sistema *Linux*).

Reconhecemos que as subcategorias de estudo da categoria engenharia reversa e *Binary Analysis* são as seguintes:

- ***ELF Binary Format***: *Executable and Linking Format* (ELF) é um formato *standard* utilizado em *Linux*, tanto em executáveis, como *shared libraries*, *core dumps* e *object files*. Estes permitem perceber como o programa é mapeado no disco e como é carregado em memória. O formato ELF é uma composição de componentes fortemente orquestrados, com diferentes especificações, em particular: *ELF file types*, *Program headers*, *Section headers*, *Symbols*, *Relocations*, *Dynamic linking* e *ELF parsers*.

- ***Linux Process Tracing*** (*ptrace system call*) – O *ptrace* é utilizado para ler ou escrever na memoria do processo e permite aceder a dados, *stack*, registos, memoria *heap* e ao código. O programa ELF é mapeado no espaço de endereçamento do processo, pelo que é possível analisar e modificar esses dados, de forma semelhante a um arquivo ELF no disco. O *ptrace* é utilizado para controlar o fluxo do programa, realizar tarefas de depuração, deteção de vírus, análise da memória e realizar *hotpatching*.

- ***ELF Virus Technology***: Fim da década de 80 e início dos anos 90, *Silvio Cesare* publicou vários *papers* e importantes técnicas sobre os vírus em formatos ELF. As técnicas são PLT/GOT *redirection*, infeções por *padding* usando dados e texto, *relocatable code injection*, *kmem patching* e *hijacking* de funções do *kernel*. Estes conceitos são utilizados para realizar *patch* sobre os binários, em diferentes âmbitos, segurança, engenharia reversa ou engenharia de *software*.

- ***ELF Binary protection***: Conjunto de técnicas que permitem ofuscar, encriptar binários ELF e proteger o código fonte (*binary protection e binary hardening techniques*). Por exemplo: o *DacryFile* (2001), o *Burneye* (2002) ou o *Shiva* (2003). Existem outros conceitos, por exemplo, os *dumb protectors* ou o uso dos mecanismos *stub*.

- ***ELF Binary Forensics***: Existem fins totalmente diferentes para alterar um binário. A analise *forense* permite determinar se um binário foi alterado e de que modo, focando-se na deteção de anomalias, como a pesquisa por vírus, *backdoors*, *bootnets* ou evidencias de código suspeito.

- ***Process Memory Forensics***: A análise à memória em tempo de execução é semelhante à de um binário regular, porém existem mais segmentos e ocorrem várias mudanças, envolvendo realocações, alinhamento de segmentos e expansões (esta análise é importe para entender como um processo aparece na memória).

- ***Extended Core File Snapshot Technology*** (ECFS): Desenhado para análise *forense*, o software ECFS cria *snapshots* dos processos e fornece um formato de arquivo, que descreve cada nuance dos processos.

- ***Linux /proc/kcore Analysis*** – O *kernel* do *Linux* é também um binário ELF, que é carregado na memória durante o arranque. Existem várias técnicas para infectar um *kernel* em tempo de execução, nomeadamente, *syscall table injection*, *interrupt handler patching*, *function trampolines*, *debug register rootkits*, *exception table infection* e *Kprobe instrumentation*.

(Fontes de informação utilizadas para recolher os conceitos apresentados na presente secção 3.3.3: [87], [188])

3.3.4 Categoria Networking

Uma grande parte dos exercícios CTF envolve conceitos em redes: enumeração de serviços, comunicações com os servidores, exploração de protocolos inseguros. Como tal, a seguinte lista desenvolve os conceitos, que consideramos essências no âmbito dos exercícios CTF.

Reconhecemos que as subcategorias de redes (*networking*) são as seguintes:

- **Protocolos e *standards***: Envolve os conceitos básicos em redes, como o modelo OSI, os endereçamentos IPv4 e IPv6, aspetos do *Data-Link Layer* (endereços MAC, o protocolo ARP), e os *sockets*;

- ***Host and Port Scanning***: Consistem em técnicas e algoritmos de reconhecimento, por exemplo: o *Stealth SYN Scan*; *scans* com determinadas *flags* (*TCP FIN, NULL, and Xmas Scans*) [184], *spoofing decoys* ou *idle scanning*;

- ***Network Sniffing***: Este tema relaciona-se com a captura de *network packets*, através de programas e bibliotecas, como o *tcpdump* e a biblioteca *libpcap sniffer* (*raw sockets*);

- ***Denial of service*** (DoS): Os ataques DoS comprometem a disponibilidade e podem parar um serviço, por exemplo, *Ping of Death*, *Ping Flooding*, SYN Flooding, Teardrop, Amplification Attacks, Distributed DoS Flooding (dependendo das regras da competição, estes ataques podem ser utilizados em cenários de *attack/defense*);

- ***TCP/IP Hijacking/MITM***: Em ataques Man-in-The-Middle (MITM), os atacantes intercetam as comunicações, assumem o controlo e falsificam os pacotes de uma ligação (*RST Hijacking*);

- ***System Security***: Identificar o comportamento de um *software malicioso*, a partir de evidências, como *logs* e capturas de rede (no âmbito das redes, como os *worms* e as *botnets*). Analisar as vulnerabilidades e implementar medidas de segurança (por exemplo, firewalls ou sistemas de deteção de intrusão (IDS/IPS));

- ***Network Applications***: Conjunto de tecnologias e aplicações aplicados à comunicação de dados. Alguns exemplos: TLS, *Kerberos*, *Wireless Network Security* (IEEE 802.11i), *Electronic Mail Security* (*Pretty Good Privacy*, S/MIME, *DomainKeys Identified Mail*) e IP *Security* (IP *Security Policy*, *Encapsulating Security Payload*, *Internet Key Exchange* e *Crypto Suites*).

(Fontes de informação utilizadas para recolher os conceitos apresentados na presente secção 3.3.4: [234] [206])

3.3.5 Categoria Computer and Mobile Forensics

Esta categoria de exercícios CTF, foca-se no estudo de evidências e no desenvolvimento de processos *forense*, envolvendo técnicas que auxiliam uma investigação *forense*. Por exemplo: a recuperação de dados, a monitorização de sistemas ou a análise de ameaças.

Reconhecemos que as subcategorias de estudo da categoria *Computer and Mobile Forensics* são as seguintes:

- ***Host Analysis***: As arquiteturas dos sistemas operativos, em particular os sistemas de ficheiros e formatos dos arquivos, em máquinas *Linux*, *Windows* e *Macintosh*;
- ***Storage Analysis***: Conjunto de técnicas que permitem analisar *datasets* e sistemas RAID, por exemplo, sistemas *Network Attached Storage* (NAS) ou *Storage Area Network* (SAN);
- ***Mail Analysis***: Procedimentos aplicados à análise de servidores e clientes de email, formatos dos ficheiros (PST, OST). Entre outras ferramentas dedicadas à análise de email, por exemplo, o *E-mail Examiner*, o *EnCase* ou o *FTK*;
- ***Tracking User Activity***: Conjunto de técnicas para reconstruir e rastrear as ações dos utilizadores, em particular, os documentos criados/modificados e o histórico do *browser*. Estas técnicas permitem reconstruir uma *timeline* de eventos, através de análises a aplicações comuns, como o *Microsoft Office*, e através da manipulação e reconstrução de atividades usando o perfil dos *browsers* (*histórico*, *cache* ou *cookies*);
- ***Mobile Devices***: Este ramo foca-se em particular na recuperação de dados e evidências dos dispositivos móveis e *wearables*. A ferramenta *Device Seizure* (DS) permite realizar imagens lógicas, *scans* de malware e análises aos *firmwares*, aos ficheiros e às aplicações;
- ***Steganography*** (também conhecida nos exercício CTF por *Stegano*): Consiste em esconder um arquivo dentro de outro arquivo. Esta área envolve os formatos das imagens, a sua representação digital e técnicas de esteganografia. Algumas técnicas e abordagens conhecidas são o LSB, o *KL divergence*, *model-preserving* e *mimicking natural processing*;

(Fontes de informação utilizadas para recolher os conceitos apresentados na presente secção 3.3.5: [66] [71] [101])

3.3.6 Categoria Open Source Intelligence

Open Source Intelligence (OSINT) corresponde aos dados coletados através de fontes públicas. Este processo de reconhecimento é parte integrante em exercícios de *penetration testing*. Esta categoria envolve várias fontes de informação e formas de tirar proveito das mesmas.

3.3.7 Categoria Professional Programming Challenges

Atualmente, existem várias competições dedicadas à programação: *Competitive Programming*. Incluímos esta categoria, uma vez que, alguns exercícios das competições CTF

também se classificam neste âmbito, sendo frequente, o desenvolvimento de ferramentas e *exploits* nas competições CTF.

A categoria Professional Programming Challenges (PPC) consiste na resolução de um determinado problema através da programação, estas competições relacionar-se com problemas reais e complexos, por exemplo, terem como objetivo o desenvolvimento de um novo algoritmo.

3.3.8 Categoria Miscellaneous

Miscellaneous (Misc): A categoria *Miscellaneous*, ou simplesmente outros, abrange desafios que não estão associados a nenhuma das categorias anteriores. Podem envolver enigmas ou tarefas relacionadas com o próprio evento CTF, por exemplo, desafios propostos pelas entidades parceiras das competições CTF.

3.4 Plataformas Capture the Flag

Na presente secção analisamos as plataformas CTF existentes. Durante o estudo das plataformas CTF existentes, verificamos que os exercícios distinguem-se também entre estáticos e dinâmicos.

Os exercícios CTF estáticos são desafios, que não envolvem nenhum sistema, isto é, são apenas compostos por materiais estáticos (por exemplo, um ficheiro *zip* encriptado). Por outro lado, os exercícios CTF dinâmicos envolvem redes de computadores e servidores (*cyber ranges*).

Em geral, existem várias funcionalidades a ter em conta para uma plataforma de CTF, as quais, agrupamos em três grupos:

- Antes do evento CTF: registar os participantes e as equipas; customizar as interfaces; publicar as regras, os prémios e definir um horário; registar os exercícios e apresentar o *scoreboard*;
- Durante o evento CTF: controlar a execução da plataforma; observar o número de submissões; disponibilizar o *scoreboard* atualizado; gerar as notificações e os alertas; inserir dicas nos desafios; monitorizar eventos e obter estatísticas da própria plataforma;
- Após o evento CTF: publicar o painel de pontuações (*scoreboard*) e obter um *backup* das estatísticas da competição.

3.4.1 Framework/Plataforma CTFd

Criada por *Kevin Chung* em 2017, CTFd é uma *framework* dedicada a atividades de *Cyber Security Training*, focada na personalização. A *framework* deu origem a uma plataforma *open source*, também com o mesmo nome, disponível em [60].

Atualmente, CTFd é uma das plataformas mais utilizada nas competições CTF, tendo sido recomendada por várias entidades, como *FireEye Inc* e *New York University*.

Desde 2014, que a *FireEye* realiza competições CTF todos os anos, no âmbito da categoria *reverse engineering*. No início a *FireEye* não utilizava nenhuma plataforma, pelo

que, os dados da competição eram limitados. Em 2016, a *FireEye* recorreu à *framework* CTFd, para conceber uma plataforma personaliza para as suas necessidades, o que permitiu melhorar a fase de registo dos participantes, a compreensão com os exercícios e oferecer *feedback* durante as suas competições.

A *New York University* recorreu à *framework* CTFd e construiu vários *plugins*, que personalizam automaticamente os exercícios CTF para cada aluno. Assim, dois alunos não recebem exatamente o mesmo exercício, e estas personalizações nos exercícios são criadas automaticamente. O CTFd avalia automaticamente as respostas dos alunos, se a solução estiver correta, então o seu número de pontos no *scoreboard* aumenta.

A *framework* CTFd suporta exercícios estáticos e dinâmicos (sendo que, os exercícios dinâmicos têm de ser executados à parte). Existem vários *pulgins* disponíveis, que adicionam novas funcionalidades à *framework*.

A interface do *scorebord* do CTFd, apresenta uma tabela com os pontos de cada participante e um gráfico com a evolução das 10 melhores equipas (ver Fig.3.6).

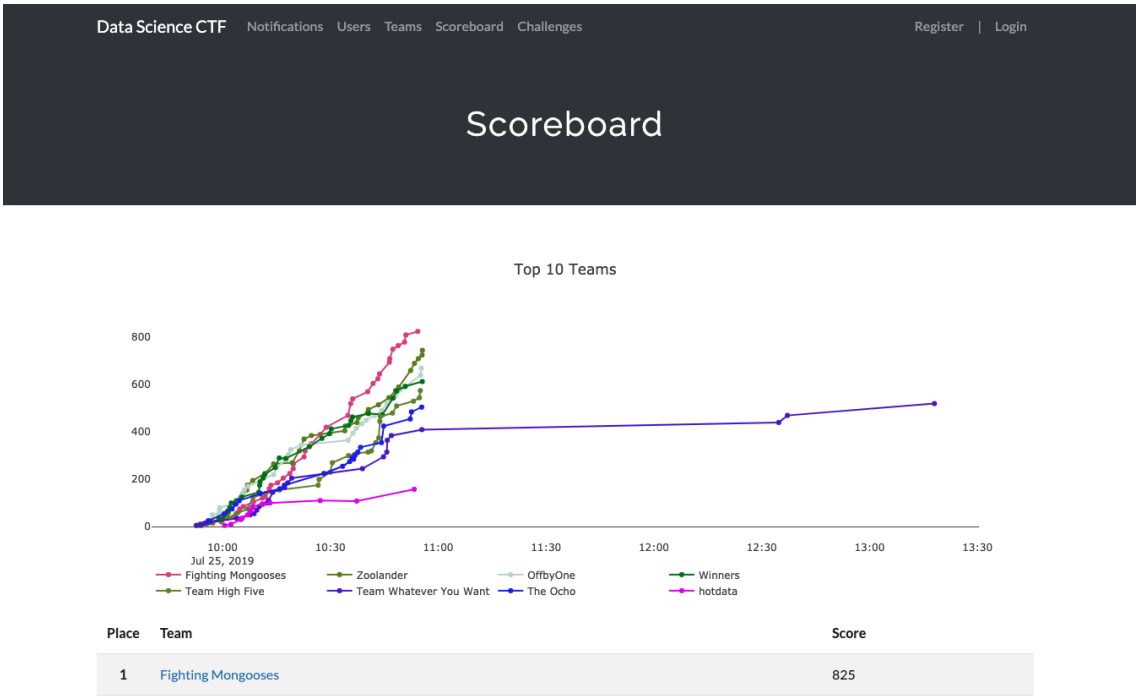

Figura 3.6: Plataforma CTFd - Interface com o Scoreboard (Fonte: [35])

As principais funcionalidades presentes na plataforma CTFd são as seguintes [60]:

- Gestão dos participantes e/ou das equipas;
- Gestão de exercícios CTF;
- Personalização das páginas *web* e estilos;
- *Feed* do scoreboard compatível com o *CTFTime*;
- Personalização das funcionalidades (arquitetura baseada em *plugins*);
- Exportar e importar os dados da plataforma (função de *backup*);

- Suporte de *e-mail* SMTP (Mailgun);
- Gestão de conteúdos em *Markdown*.

3.4.2 Root The Box

Em 2016, Joe D. (Moloch) e seus colaboradores, criaram a plataforma *Root The Box*. Ao contrário da plataforma CTFd (que fornece as operações CRUD básicas para uma competição), *Root The Box* simula um jogo, com objetivos, missões, *cutscenes* e diálogos.

Nesta plataforma, existem organizações fictícias que agregam um conjunto de *box's* (exercícios CTF), e para passar de nível, os participantes tem de completar determinados objetivos em cada organização. A Fig.3.7 representa a interface deste jogo, onde se observa um conjunto de perguntas na categoria de criptografia.

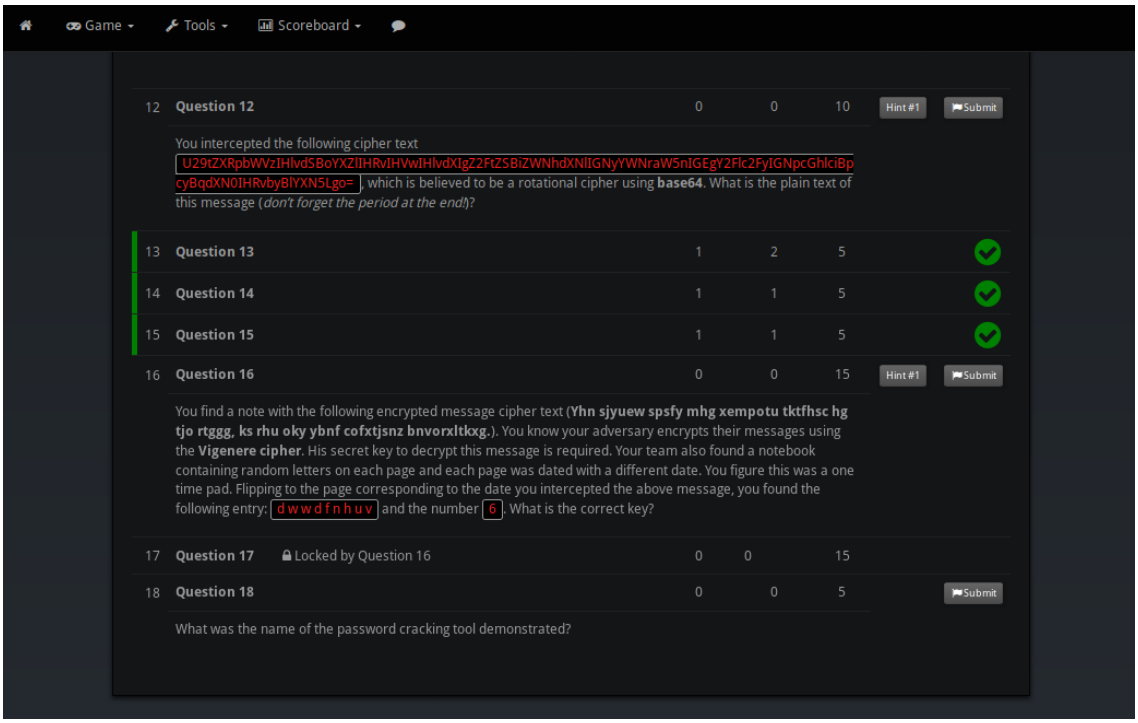

Figura 3.7: Plataforma Root the box - interface com os exercícios CTF (Fonte: [108])

As principais funcionalidades presentes na plataforma *Root The Box* são as seguintes:

- Gestão dos participantes e/ou das equipas;
- Gestão de exercícios CTF;
- Personalização das páginas *web* e estilos;
- *Scoreboard*, gráficos e atualizações dinâmicas;
- *Feed* do *scoreboard* compatível com o *CTFTime*;
- *CyberChef* integrado no menu de ferramentas;
- Suporta o modo de história, com caixas de diálogo de introdução ou secções com gráficos *cutscenes*;

- Recursos opcionais avançados: permite a execução de *scorebots*, disponibiliza um painel *Wall of Sheep*;
- suporta integração com o *Rocket Chat* (servidor de *chat*).

3.4.3 Plataforma PicoCTF

Desde 2018, que a *Carnegie Mellon University* disponibiliza vários conteúdos de interesse no âmbito dos exercícios CTF. A plataforma *picoCTF* é *open source* e foi utilizada na competição *picoCTF* em 2019 (encontra-se disponível em [105]).

O *website* do projeto picoCTF [240] é composto por:

- *picoPrimer* - uma *wiki* (página de documentação) no âmbito da segurança informática;
- *picoGym* - plataforma construída e disponibilizada pela universidade, com vários exercícios CTF (ver Fig.3.8);
- *Carnegie Mellon University CyLab* - laboratório de suporte aos programas de graduação em segurança informática;
- competições *picoCTF* - a *Carnegie Mellon University* organiza nas suas instalações competições CTF.

Atualmente, a *Carnegie Mellon University* foca-se no desenvolvimento da plataforma *picoGym*, plataforma *online* de registo gratuito, que possui vários exercícios CTF, e que tem como público alvo, alunos das escolas secundárias e das universidades (esta plataforma é utilizada em mais de 40 instituições de ensino [240]).

As competições picoCTF realizadas desde 2018, possuem várias entidades parceiras, como *Cisco*, *National Security Agency USA*, *Boeing*, *General Motors*, *Siemens* e *Trane Technologies*.

Os exercícios CTF encontram organizados por categorias, a interface *web* possui uma *Webshell*, que permite interagir com uma máquina *Linux* (ver Fig.3.8).

3.4.4 Plataforma FBCTF

O *Facebook* organiza anualmente competições CTF (*Facebook Capture The Flag events*), onde os participantes resolvem problemas complexos (*"security puzzles"*), com o objetivo de promover uma cultura de segurança e aprofundar a formação dos profissionais de segurança [92].

A plataforma FBCTF, disponibilizada pelo *Facebook* em 2016, tem como objetivo disponibilizar competições no formato *Jeopardy* e *"King of the Hill"*.

Os exercícios CTF são organizados por países e disponibilizados em cima de um planisfério. Ao resolver os exercícios CTF, os participantes conquistam os países (ver Fig.3.9).

As principais funcionalidades presentes na plataforma FBCTF são as seguintes:

- Gestão dos participantes e/ou das equipas;

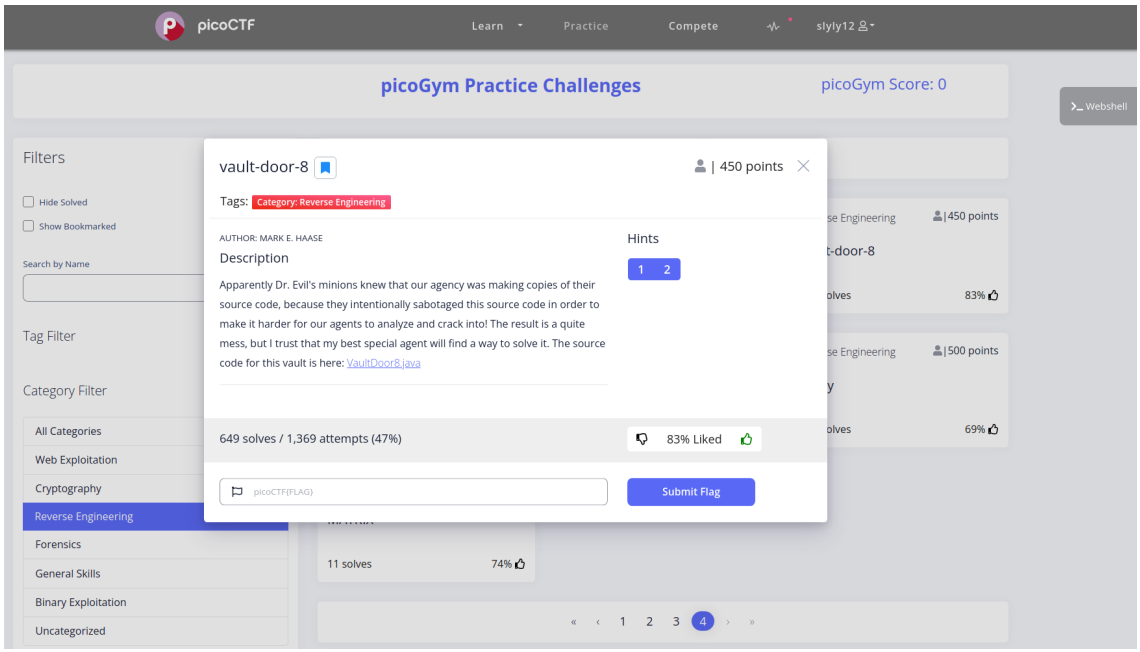

Figura 3.8: Plataforma PicoCTF - interface com os exercícios CTF (Fonte: [241])

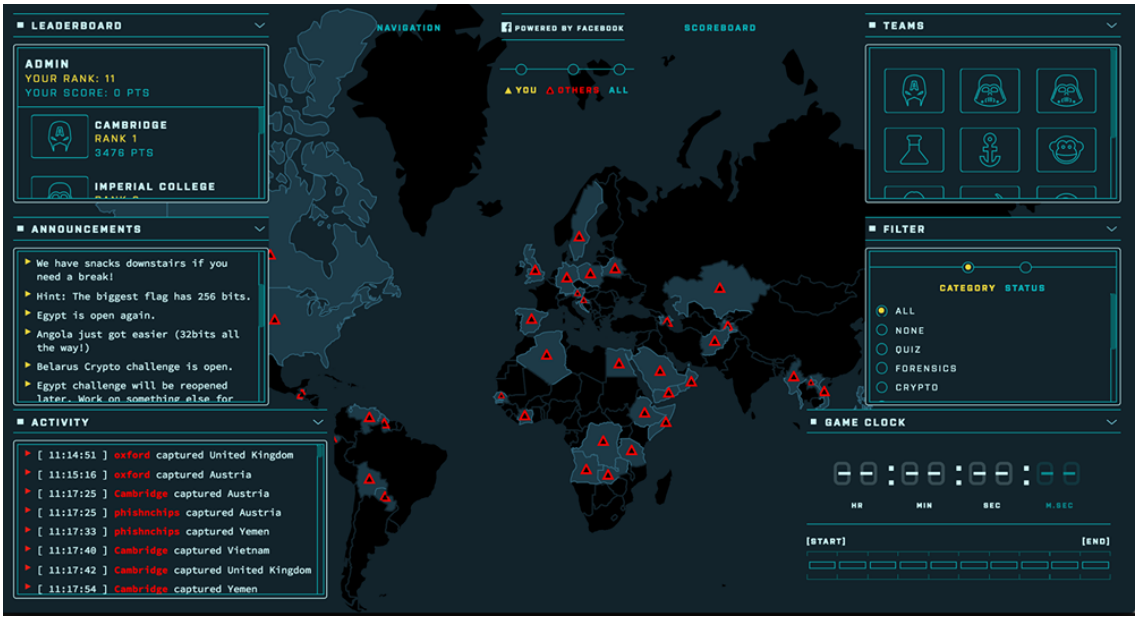

Figura 3.9: Plataforma FBCTF - interface com os exercícios CTF (Fonte: [220])

- Gestão de exercícios CTF;
- *Scoreboard*, gráficos e atualizações dinâmicas;
- Possui um serviço *chat* integrado na própria *dashboard*.

3.5 Síntese

No presente capítulo apresentamos o estado da estado da arte, onde abordamos as competições CTF, os seus formatos e aplicabilidades.

Na secção 3.3, apresentamos as categorias dos exercícios CTF: *Web Security*, *Cryptography*, *Binary Analysis and Reverse Engineering*, *Networking*, *Computer and Mobile Forensics*, *Open Source Intelligence*, *Professional Programming Challenges* e *Miscellaneous*.

Por último, apresentamos algumas plataformas, que são utilizadas nas competições CTF e nos cursos de segurança informática, as quais, iremos ter em conta no desenho da plataforma CTF@DEI (capítulo 5).

Esta página foi intencionalmente deixada em branco.

Capítulo 4

Planeamento e Metodologia

No presente capítulo, temos como objetivo definir a metodologia de trabalho, isto é, os mecanismos que nos permitem garantir o cumprimento do projeto proposto.

Neste capítulo, apresentamos a metodologia de trabalho e os procedimentos utilizados ao longo deste projeto.

Indicamos as tarefas de trabalho do projeto, onde as classificamos por nível de importância e esforço estimado, e apresentamos o cronograma de atividades.

Por último, através de uma avaliação de risco, ponderemos os riscos envolvidos nas tarefas, e em função disso, indicamos os respetivos planos de mitigação.

4.1 Planeamento

No Mestrado em Segurança Informática, o estágio corresponde a um esforço anual de 54 ECTS, divididos entre a meta intermédia e a final.

De acordo com a Proposta de Estágio que prepusemos, definimos os Work Packages (WP) necessárias em dois grupos (ver Tabela 4.1), para a meta intermédia e final.

Para a primeira fase do projeto, estudo da solução e planeamento, formulamos o seguinte plano de trabalhos:

- WP1 - Elaboração do Estado da Arte sobre os exercícios Capture the flag;
- WP2 - Definição da Arquitetura da plataforma *CTF@DEI*;
- WP3 - Elaboração do Relatório Intermédio;

Para a segunda fase do projeto, desenvolvimento e conclusão, formulamos o seguinte plano de trabalhos:

- WP4 - Redefinição da Arquitetura da plataforma *CTF@DEI*;
- WP5 - Desenvolvimento da plataforma *CTF@DEI*;
- WP6 - Análise de resultados, através da revisão da literatura;

- WP7 - Validação da plataforma, através da realização de questionários;
- WP8 - Elaboração do Relatório Final;

Tabela 4.1: Plano geral de atividades

Título	Tipo	Início	Fim
1ª Fase - estudo da solução e planeamento			
WP1 - Elaboração do Estado da Arte sobre os exercícios CTF	WP	10/2019	11/2019
WP2 - Definição da Arquitetura da plataforma CTF@DEI	WP	11/2019	01/2020
WP3 - Elaboração do Relatório Intermédio	WP	10/2019	01/2020
Entrega e defesa do Relatório Intermédio	Meta		02/2020
2ª Fase - desenvolvimento			
WP4 - Redefinição da Arquitetura da plataforma CTF@DEI	WP	02/2020	03/2020
WP5 - Desenvolvimento da plataforma CTF@DEI	WP	03/2020	09/2020
WP6 - Análise de resultados	WP	08/2021	10/2021
WP7 - Elaboração do Relatório Final	WP	08/2021	10/2021
Entrega e defesa do Relatório Final	Meta		11/2021

Na subsecções 4.1.1 e 4.1.2, decompomos o plano geral de atividades 4.1, em tarefas.

4.1.1 1ª Fase do projeto - estudo da solução e planeamento

As tarefas da 1ª fase, estudo da solução e planeamento, são as seguintes.

- WP1 - Elaboração do Estado da Arte sobre os exercícios Capture the flag:
  - WP1.T1 - Análise das competições CTF - conceitos, formatos e categorias;
  - WP1.T2 - Estudo das plataforma CTF existentes - Funcionalidades;
  - WP1.T3 - Casos de uso e experiências com os exercícios CTF;
- WP2 - Definição da Arquitetura da plataforma *CTF@DEI*:
  - WP2.T1 - Requisitos funcionais e não funcionais;
  - WP2.T2 - Arquitetura Lógica e Física;
  - WP2.T3 - Desenho dos artefactos (atores, casos de uso, *mockups*);
- WP3 - Elaboração do Relatório Intermédio.

Na Fig.4.1 apresentamos o diagrama de *Gantt*, da 1ª Fase do projeto - estudo da solução e planeamento, relativamente ao esforço real.

4.1.2 2ª Fase do projeto - desenvolvimento e conclusão

As tarefas da 2ª fase, desenvolvimento e conclusão, são as seguintes:

- WP4 - Redefinição da Arquitetura da plataforma *CTF@DEI*;

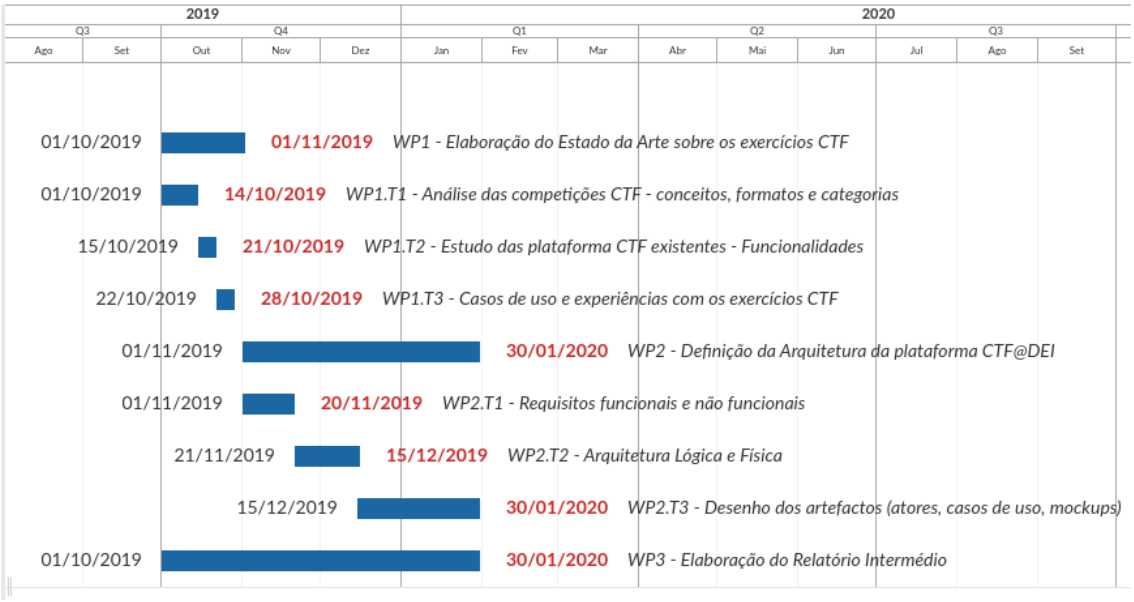

Figura 4.1: Cronograma da 1ª Fase do projeto

  - WP4.T1 - Requisitos funcionais e não funcionais;
  - WP4.T2 - Arquitetura Lógica e Física;
  - WP4.T3 - Seleção das tecnologias;

- WP5 - Desenvolvimento da plataforma *CTF@DEI*;
  - WP5.T01 - Implementação do servidor de Armazenamento;
  - WP5.T02 - Criação da Coleção CTF;
  - WP5.T03 - Recolha de exercícios CTF;
  - WP5.T04 - Adaptação dos exercícios CTF;
  - WP5.T05 - Implementação da plataforma CTFd;
  - WP5.T06 - Configuração da base de dados e Redis;
  - WP5.T07 - Implementação da API e do cliente CLI;
  - WP5.T08 - Implementação do website de documentação;
  - WP5.T09 - Implementação do serviço de chat e bots;
  - WP5.T10 - Implementação do "Analisador de vulnerabilidades";
  - WP5.T11 - Configuração do Proxy, DNS e certificados HTTPS;
  - WP5.T12 - Desenvolvimento dos programas de automação;
  - WP5.T13 - Testes com a pipeline CI do Gitlab;
  - WP5.T14 - Configuração dos Git Runners, com os programas de automação;
  - WP5.T15 - Instalação do Kubernetes;
  - WP5.T16 - Reestruturação do Kubernetes, com o Calico;
  - WP5.T17 - Reestruturação do Kubernetes, com o DNS do DEI;
  - WP5.T18 - Configuração do Kubectl;
  - WP5.T19 - Instalação do Virtualbox e configuração do VBoxManage;
  - WP5.T20 - Implementação do Docker Registry e LFS no GitLab;

  - WP5.T21 - Desenvolvimento da CTF Toolkit;

- WP6 - Análise de resultados;
  - WP6.T1 - Revisão da literatura;
  - WP6.T2 - Elaboração dos questionários;
  - WP6.T3 - Validação da plataforma;
- WP7 - Elaboração do Relatório Final.

Nas Fig.4.2, Fig.4.3 e Fig.4.4 apresentamos o diagrama de *Gantt*, da 2ª Fase do projeto - estudo da solução e planeamento, relativamente ao esforço real.

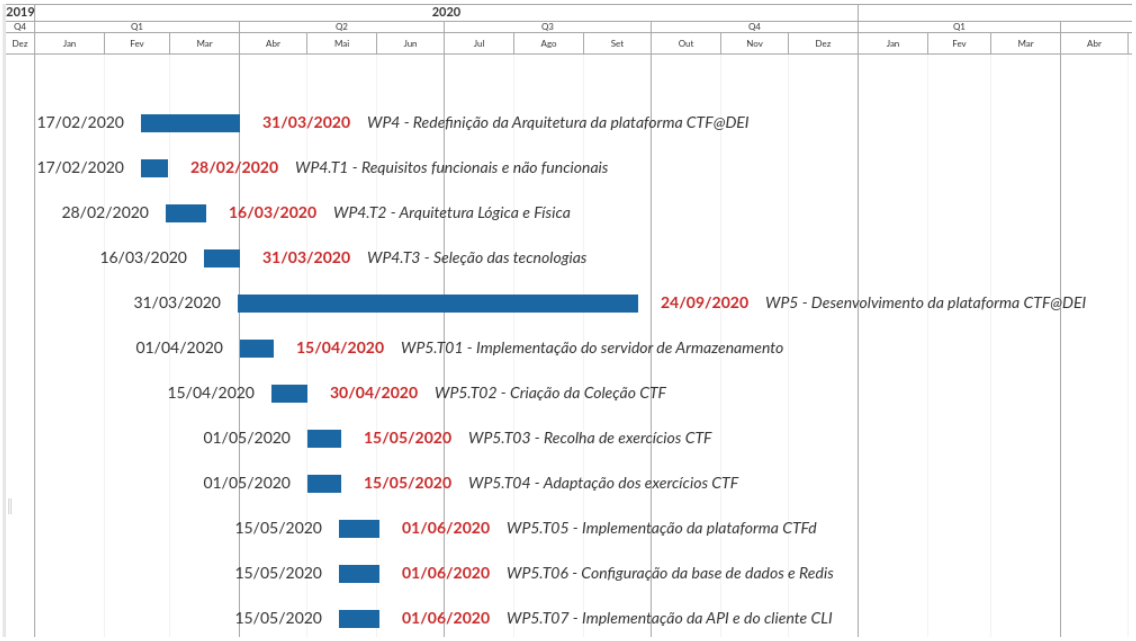

Figura 4.2: Cronograma da 2ª Fase do projeto - Parte 1

4.1.3 Desvios ao Planeamento

Os diagramas de *Gantt* anteriores mencionam o esforço real para cada uma das tarefas. Face ao progresso apresentado, foi possível identificar as seguintes anomalias:

- Atraso no WP2 (Definição da Arquitetura da plataforma CTF@DEI):

  - *causa* – A complexidade das plataformas CTF existentes, requereu uma investigação exaustiva, tendo esta sido também afetada, pela inexperiência com as plataformas e pela necessidade de experimentar as plataformas existentes (R01, R02, R03 da Tabela.4.2;

  - *impacto* – Afetou a qualidade de WP3 - Elaboração do Relatório Intermédio;

  - *mitigação* - Atualizamos a arquitetura desenvolvida e adoptamos a Método MoSCoW (PM01 e PM04 da Tabela.4.3).

- Atraso no WP3 - Elaboração do Relatório Intermédio:

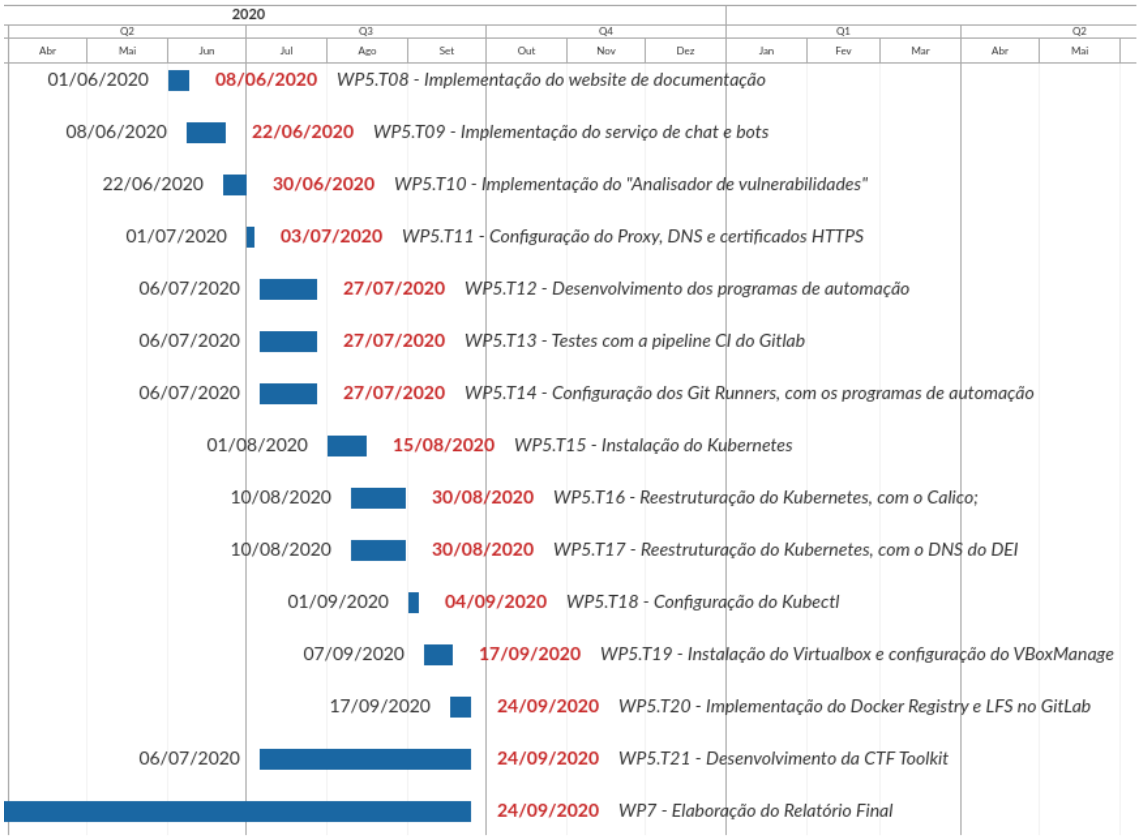

Figura 4.3: Cronograma da 2ª Fase do projeto - Parte 2

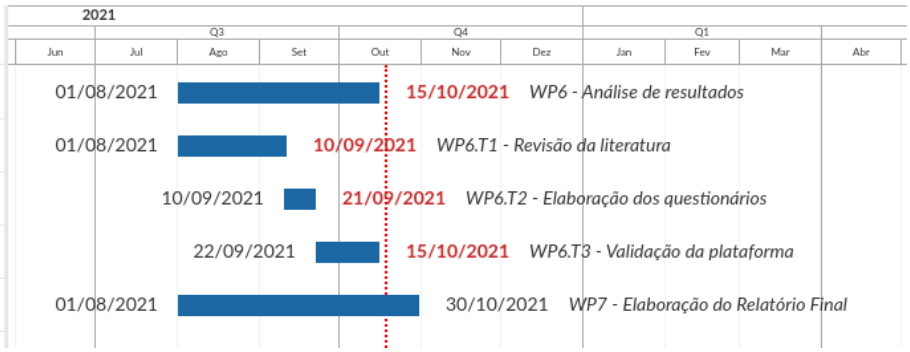

Figura 4.4: Cronograma da 2ª Fase do projeto - Parte 3

- *causa* – Consequência do atraso sofrido na tarefa anterior (R05 da Tabela.4.2;

- *impacto* – Pouco tempo de revisão da versão intermédia do relatório;

- *mitigação* – Foram exercidas cargas horárias extra para garantir a execução da tarefa (PM02 Tabela.4.3).

- Necessidade de introduzir o WP4 - Redefinição da Arquitetura da plataforma CTF@DEI:

  - *causa* – em WP2, a complexidade da arquitetura não é implementável no tempo disponível (R06 da Tabela.4.2;

  - *impacto* – Criação de um novo WP4 - para redefinir a arquitetura apresentada;

  - *mitigação* – Foram feitos ajustes na arquitetura desenvolvida, tendo em conta a defesa intermédia; foram exercidas cargas horárias extra para garantir a execução das tarefas; e as tarefas foram classificadas novamente com o método MoSCoW; (PM01, PM02 e PM04 Tabela.4.3).

- Atraso no WP5 - Desenvolvimento da plataforma CTF@DEI:

  - *causa* – desconhecimento das tecnologias necessárias e atraso no WP5 provado pela pandemia Covid-19 (R04 e R05 da Tabela.4.2;

  - *impacto* – atraso nas tarefas seguintes;

  - *mitigação* – foram realizados cursos *online* sobre *Docker* e *Kubernetes* (PM03 da Tabela.4.3); o projeto teve de ser interrompido, tendo o cronograma sido reajustado em função disso;

- Início tardio no WP6 - Análise de resultados:

  - *causa* – início tardio no WP6 provado pela pandemia Covid-19 (R05 da Tabela.4.2;

  - *impacto* – atraso na realização dos questionários e nas tarefas seguintes;

  - *mitigação* – apesar do início tardio, foram exercidas cargas horárias extra para garantir a execução da tarefa (PM02 Tabela.4.3).

- Atraso no WP7 - Elaboração do Relatório Final:

  - *causa* – Consequência do atraso sofrido na tarefas anteriores (R05 da Tabela.4.2; (R05 da Tabela.4.2;

  - *impacto* – Pouco tempo de revisão da versão final do relatório;

  - *mitigação* – Foram exercidas cargas horárias extra para garantir a execução da tarefa (PM02 Tabela.4.3).

4.1.4 Metodologia e Ferramentas de Gestão de Projetos

Face às questões adversas da pandemia *Covid-19*, a metodologia aplicada entre o orientado e respetivos orientadores foi o *Waterfall*. Esta metodologia, assegurou o refinamento dos requisitos na arquitetura, a sua avaliação, e facilitou a execução das tarefas planeadas. Optamos pela metodologia *Waterfall* pelas seguintes razões [46]:

- por nos permitir detetar eventuais erros, durante a fase de análise e desenho da arquitetura, permitindo-nos redefinir os requisitos, evitando implementações desnecessárias;
- após o estabelecimento dos requisitos, podemos estimar com exatidão os custos (em termos de tempo), e classificar as tarefas por importância;

Também, consideramos as desvantagens da metodologia *Waterfall*, em particular, as seguintes:

- os alunos não são envolvidos nas fases de desenho e implementação;
- se existirem atrasos, então todas as tarefas seguintes serão adiadas.

Em complemento com a metodologia *Waterfall*, e face a restrições de tempo, classificamos e priorizamos as tarefas através do método *MoSCoW* [25] - deixando assim parte dos requisitos para futuros desenvolvimentos.

Recorremos ao método *MoSCoW* na 2ª fase do projeto, uma vez que, detectamos dificuldades durante a fase intermédia. Assim, priorizamos as tarefas, segundo o grau de relevância das funcionalidades para os alunos do MSI.

Para a gestão do projeto e alocação de tempo das tarefas, adoptamos a ferramenta:

- OpenProject - plataforma *open source* e *on-premise*, suporta metodologias tradicionais e ágeis, e foi utilizada para organizar as tarefas, estimar o esforço e gerar os diagramas de *Gantt* [189];

4.1.5 Análise de Riscos

Na Tabela.4.2 apresentamos a análise dos possíveis riscos associados ao projeto. Tanto, a probabilidade como o impacto, foram definidos numa escala de 1 a 5, sendo que 5 significa maior probabilidade / impacto no projeto.

Tabela 4.2: Análise de riscos do projeto

ID	Risco	Probabilidade	Impacto
R01	Indefinição do trabalho expectável	3	5
R02	Adição de requisitos ou complexidade	2	4
R03	Remoção de requisitos ou complexidade	2	4
R04	Desconhecimento das tecnologias em utilização	3	4
R05	Atrasos na realização de tarefas	4	5
R06	Incapacidade de implementar a arquitetura desenhada	2	4

Na Tabela.4.3, de forma a minimizar os possíveis impactos, apresentamos os planos de mitigação previstos para os riscos referidos na Tabela.4.2.

Tabela 4.3: Planos de mitigação face aos riscos do projeto

ID	Plano de Mitigação	Riscos
PM01	Atualizar a arquitetura desenvolvida e o projeto desenvolvido até ao momento	R01 R02 R03 R06
PM02	Exercer cargas horárias extra para garantir a execução das tarefas	R05 R06
PM03	Realizar cursos online e formações adicionais	R04
PM04	Classificação das tarefas por importância	R02 R06

Capítulo 5

Arquitetura

Para atingir o objetivo proposto neste relatório, necessitamos de elencar os requisitos funcionais, os não funcionais e a arquitetura do sistema.

Recorremos à técnica de MoSCoW, para definir a prioridade dos requisitos em quatro níveis: *Must have*, *Should have*, *Could have*, *Wouldn't have* [25].

5.1 Requisitos funcionais

Para implementar a arquitetura são necessários os seguintes requisitos funcionais.

FR01 - Providenciar um serviço que permita armazenar e gerir uma coleção de exercícios CTF (#1 *Must have*).

FR02 - Providenciar uma plataforma Web que enumere os exercícios e apresente os seus enunciados (#1 *Must have*).

FR03 – A gestão dos participantes será realizada através do *website* (#2 *Should have*).

FR04 – A gestão dos exercícios será realizada através do *website* (#2 *Should have*).

FR05 – Os participantes submetem as *flags* no *website* (#1 *Must have*).

FR06 - A plataforma valida as submissões e atribui os devidos pontos (#1 *Must have*).

FR07 – O *website* terá um quadro de pontuações, que reflete os pontos obtidos pelos participantes (#3 *Could have*).

FR08 – Os exercícios apresentados no *website* poderão ser inseridos, atualizados ou removidos, através do terminal (#3 *Could have*).

FR09 – Os exercícios inseridos na coleção serão sincronizados automaticamente com o *website* (#4 *Wouldn't have*).

FR10 – Os participantes poderão submeter as *flags* através do terminal (#3 *Could have*).

FR11 – A plataforma analisa estaticamente o código fonte do exercício e enumera as vulnerabilidades (#4 *Wouldn't have*).

FR12 – A plataforma envia uma notificação quando um novo exercício é adicionado (#4 *Wouldn't have*).

FR13 – Os exercícios poderão ser instalados em plataformas de virtualização (#3 *Could have*).

FR14 – Os participantes podem iniciar um exercício e controlar o mesmo através do terminal (#4 *Wouldn't have*).

FR15 – As alterações de estado dos exercícios em execução são apresentadas e notificadas aos participantes (#4 *Wouldn't have*).

FR16 – Os *writeups* dos exercícios são apresentados no *website* (#4 *Wouldn't have*). A par dos requisitos funcionais, na subseção seguinte especificamos os atributos não funcionais.

5.2 Requisitos não funcionais

Para implementar a arquitetura são necessários os seguintes requisitos não funcionais (atributos de qualidade).

- FNR01 – Utilizar soluções *on-premise* para armazenar e processar os exercícios (#1 *Must have*): O código dos exercícios deverá ser protegido e gerido de forma *on-premise*. Os exercícios não podem ser utilizados em outros contextos ou confundidos com aplicações reais.
- FNR02 – Autenticação e ligações seguras (#1 *Must have*): A iteração com os serviços da plataforma deverá requerer autenticação. As sessões deverão ser realizadas em canais seguros, com HTTPS ou SSH.
- FNR03 – Integridade dos exercícios CTF (#1 *Must have*): A integridade dos exercícios deverá ser preservada, através do controlo das suas versões.
- FNR04 – Portabilidade dos exercícios CTF (#2 *Should have*): O formato dos dados exportados deverá ser compatível com o formato de outros sistemas, como o formato do *CTFtime*.

Estes requisitos não funcionais são os atributos mínimos, que consideramos para esta arquitetura. Indicamos, a título de exemplo, outras possibilidades mais ambiciosas:

- Definir requisitos para tornar as operações do sistema atómicas, consistentes e isoladas, principalmente, nas transações referentes à gestão dos exercícios e à submissão das *flags*.
- Em relação ao *website*, e como estratégia de conceber uma plataforma mais adequada para o ensino, ponderamos definir requisitos de funcionalidade, usabilidade, confiabilidade e desempenho.

A seguinte subseção apresenta a arquitetura proposta para os requisitos definidos.

5.3 Arquitetura Lógica

O diagrama da arquitetura lógica encontra-se na Fig.5.1, e da qual descrevemos os seus componentes e o propósito de cada um deles.

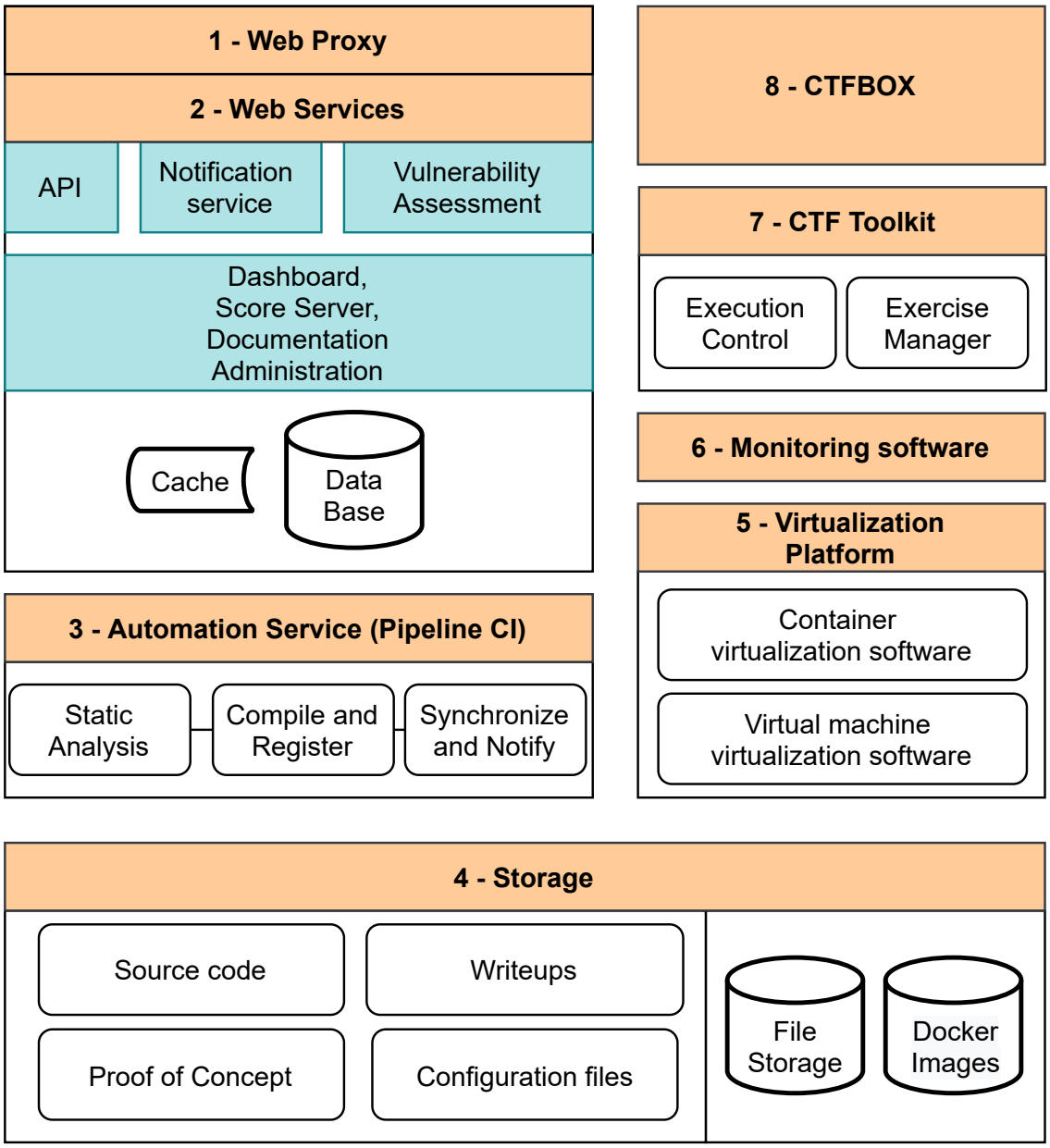

Figura 5.1: Diagrama da arquitetura lógica da Plataforma CTF@DEI

1. *Proxy* – componente que irá expor as API e as interfaces dos serviços *Web*;

2. *Web Services* – Este componente é composto pelos seguintes elementos:

   - API – componente que irá permitir interagir com a plataforma, através do terminal (FR08-10);
   - Serviço de chat - componente que será utilizado para notificar os participantes (FR12, FR15);
   - Analisador de vulnerabilidades – Componente que realiza uma análise estática ao código dos exercícios, enumera as vulnerabilidades e atribui uma classificação (FR11);
   - Serviço web – componente que disponibiliza as seguintes interfaces gráficas:

     - *Dashboard* – interface que permite consultar os exercícios e submeter as *flags* (FR02, FR05-06);

     - *Scoreboard* – interface que apresenta os pontos de cada participante (FR07);

     - Documentação – interface que apresenta os *writeups* e outros conteúdos / materiais de suporte (FR16);

     - Administração – interface que permite gerir os exercícios e os utilizadores (FR03-04);
   - Base de dados – local onde iremos armazenar os dados dos exercícios e dos participantes;
   - Serviço de cache – componente que servirá de cache para os conteúdos da base de dados;

3. Serviço de automação - Este componente é responsável por gerir os exercícios automaticamente;

   - Função que classifica o exercício – submete o exercício a uma análise estática, através dos serviços do analisador de vulnerabilidades;
   - Função que gera o objeto executável – transforma o código fonte do exercício num executável, guarda esse objeto e cria uma URL que o identifica (FR13);
   - Função que atualiza o *website* – processa o ficheiro de configuração do exercício, atualiza o *website* com essas informações e gera uma notificação (FR09, FR12);

4. Armazenamento - Este componente é responsável por guardar os exercícios e os objetos executáveis (FR01, FNR01);

   - Armazenamento de exercícios – Corresponde a uma estrutura responsável por armazenar e catalogar os exercícios, que será utilizada para criar a coleção CTF (FNR04). Cada exercício é composto por:

     - Código-fonte – Trata-se do código fonte do exercício (FNR03);

     - Writeup – Consiste num pequeno relatório escrito em *markdown*, que será exposto na página de documentação do *website* da plataforma;

     - PoC – Trata-se de um programa que, quando executado, demonstra a vulnerabilidade e captura a *flag* do exercício, solucionando o exercício CTF.

     - Ficheiro de configuração – Componente que discrimina o enunciado, os anexos, as flags, a categoria e o roadmap do exercício. Incluí as regras de instalação do exercício, isto é, os recursos computacionais a alocar e as regras de rede e de firewall;

- Armazenamento de ficheiros - Componente que irá armazenar arquivos "grandes", neste caso, irá armazenar as máquinas virtuais dos exercícios (FR13);
- Armazenamento de imagens *Docker* – Componente que irá guardar imagens *Docker*, que serão utilizadas para instanciar os *containers* (FR13);

5. Plataformas de virtualização – Conjunto de plataformas que serão responsáveis por executar os exercícios (FR13);
   - Software de virtualização de *containers* – Componente, controlado pela *CTF Toolkit*, e que irá executar os *containers*;
   - Software de virtualização de máquinas virtuais – Componente, controlado pela *CTF Toolkit*, e que irá executar as máquinas virtuais;
6. Software de monitorização – Componente que vigia o estado dos exercícios em execução e, caso existam alterações, gera notificações / eventos (FR15);
7. *CTF Toolkit* – Programa, em linha de comandos (CLI), que funcionará em terminais Bash, e irá permitir: a gestão dos exercícios nas plataformas de virtualização (FR14, FNR02); a manipulação dos exercícios no *website* (FR08); e o envio das *flags* (FR10);
8. *CTFBOX* – Componente formado por um sistema operativo *Linux* (que será gravado em *pendrives* e discos *SSD*), e que incluirá *software* pré-instalado (*CTF Toolkit* e chaves de autenticação) (FNR02). Através do *CTFBOX*, será possível gerir a plataforma *CTF@DEI* como administrador, sem ser necessário instalar ou configurar programas nos computadores dos participantes;

De acordo com os componentes apresentados, na subseção seguinte apresentamos o diagrama da arquitetura física.

5.4 Arquitetura Física

O diagrama da arquitetura física encontra-se na Fig.5.2, e da qual descrevemos os seus elementos e o propósito de cada um deles.

A arquitetura física é composta por vários servidores e computadores.

- Elemento 1 – Servidor responsável pelos serviços *web* e de automação da plataforma CTF;
- Elemento 2 – Servidor responsável por armazenar os exercícios (coleção CTF) e os objetos executáveis (imagens *Docker* e máquinas virtuais);
- Elemento 3 - Grupo de computadores que executa o sistema *CTFBOX*, cada computador (ou máquina virtual) representa um utilizador diferente, registado na plataforma;
- Elementos 4 - Os dois grupos de servidores representados correspondem às plataformas de virtualização: *Kubernetes* e *Virtualbox*. Estes servidores serão geridos pelos utilizadores registados.

A subseção seguinte define a lista de tecnologias para cada um dos componentes da arquitetura.

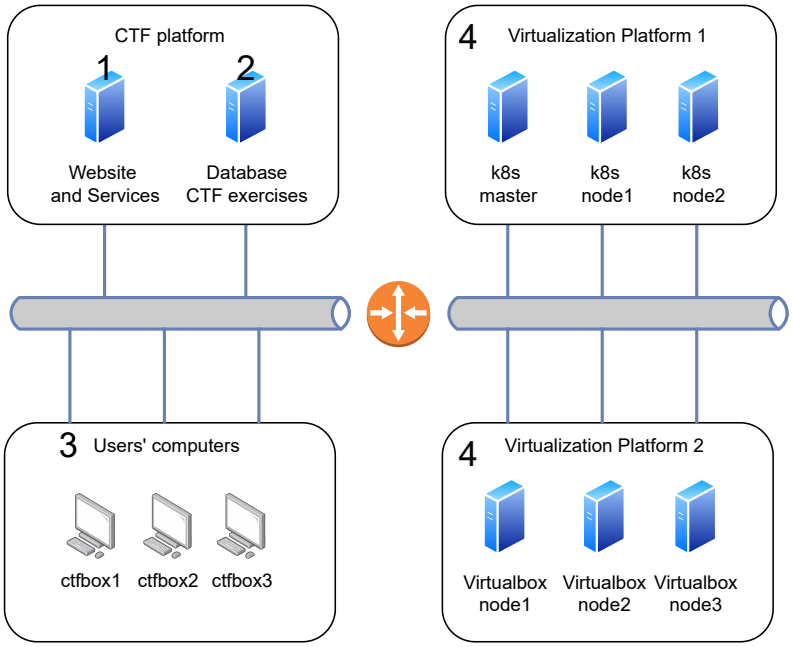

Figura 5.2: Diagrama da arquitetura física da Plataforma CTF@DEI

5.5 Seleção das tecnologias

As tecnologias que iremos utilizar para o *front-end* e para o *back-end* do *website* dependem da plataforma *Capture the Flag* que iremos utilizar.

A plataforma *Root the box* [230] foi desenvolvida em *Python*, com recurso da framework *Tornado* [237].

A plataforma *CTFd* foi desenvolvida em *Python*, com recurso de *Flask*, *Jinja2* e *SQLAlchemy* (suporta diferentes bases de dados) [60].

Optamos pela plataforma *CTFd* devido aos seguintes motivos:

- A arquitetura do *CTFd* permite desenvolver novas funcionalidades através de *plugins*;
- A plataforma *CTFd* possui uma API, que permitirá automatizar operações relevantes (sincronizar os exercícios, registar os utilizadores e submeter as *flags*);
- *CTFd* é uma plataforma de referência para os organizadores das competições CTF, compatível com o *scoreboard* do *CTFtime* [63].

Para guardar os dados do *website* recorremos à base de dados relacional *MySQL*, por ser um sistema popular, confiável e de alta performance [69]. Para efeitos de *cache* e performance iremos utilizar *Redis* [205].

Escolhemos o *Apache2*, que será utilizado como *proxy reservo* para os serviços web. Os certificados para estes serviços serão gerados através dos serviços da *Let's Encrypt*, utilizando o *certbot* para automatizar esta tarefa [98].

Para notificar os participantes, iremos utilizar o *Rocket Chat* [39], por ser uma aplicação flexível, que permitir criar diferentes canais, onde iremos representar os exercícios e gerar as notificações.

Para armazenar os exercícios, optamos por utilizar a plataforma *GitLab* pelas razões, que passamos a descrever de seguida.

- O *GitLab* é utilizado pela comunidade do DEI, com autenticação *Single Sign-On* através do serviço LDAP do DEI;
- Iremos utilizar os mecanismos já existentes da plataforma *GitLab*, para criar e parametrizar a visibilidade do grupo CTF. Pretendemos que este grupo *GitLab* seja privado e restrito aos utilizadores da plataforma CTF@DEI (FNR01).

Por simplicidade, optamos por guardar as máquinas virtuais no próprio servidor *Git*, através da extensão Large File Storage (LFS) (suporte ao armazenamento de ficheiros de grandes dimensões).

Para gerar as imagens dos *containers* iremos utilizar *Docker* [81], e para testar os exercícios iremos utilizar *Docker Compose* [78].

Para guardar as imagens *Docker*, consideramos as seguintes hipóteses:

- *Azure Registry*, que faz parte do panorama da *Cloud Native* [47], e permite guardar os objetos e enviá-los para as várias plataformas de virtualização, incluindo o *Kubernetes*;
- *Vagrand Cloud* [244] e *Docker Hub* [128], plataformas públicas, que permitem guardar *boxes vagrand* e imagens *Docker*, respetivamente;
- *Docker Registry*, trata-se de um serviço que permite distribuir as imagens *Docker*, e pode ser instalado de forma *on-premise*, com a ferramenta *Docker Compose*.

Optamos por utilizar o *Docker Registry* integrado no próprio *GitLab*, através dos pacotes *Omnibus Gitlab*.

Para automatizar a gestão dos exercícios, optamos por utilizar as *pipelines* da plataforma *GitLab CI/CD*. Para atender os requisitos apresentados, (criar o objeto executável, atualizar o *website* e submeter o exercício para análise), optamos por utilizar o serviço *GitLab Runner*, com a funcionalidade de *shell* executor, que nos permite executar *scripts* em *Bash* [115].

Para analisar o código dos exercícios optamos por utilizar *SonarQube*, porque suporta várias linguagens de programação [216].

As plataformas de virtualização irão expor uma interface (CLI ou API), que nos permita orquestrar o ciclo de vida dos exercícios, definir diferentes propriedades / recursos computacionais, e configurar as redes virtuais (*Infrastructure as Code*). As plataformas de virtualização que iremos utilizar neste projeto são o *Kubernetes* [150] e o *VirtualBox* [190].

- O *Kubernetes* permitirá executar as imagens *docker* e orquestrar múltiplos *deployments* através do terminal. Para interagir com o *Kubernetes* iremos utilizar o *Kubectl* [151];
- O *Virtualbox* permitirá executar máquinas virtuais. Optamos por esta ferramenta, por simplicidade, sendo que os alunos do DEI já estão familiarizados com o uso da mesma. Para interagir com o *VirtualBox* iremos utilizar *VBoxManage* [191].

Os programas referidos nos componentes da arquitetura, *3-Automation Service* e *7-CTF Toolkit*, serão desenvolvidos em *Bash, Python e Java* 11.

Para criar a máquina *CTFBOX*, iremos utilizar a distribuição *Kali* [141]. Neste sistema iremos adicionar as seguintes ferramentas: *CTF Toolkit*, *Docker*, *Docker Compose*, *VirtualBox*, *VBoxManage* e *Kubectl*. O *Clonezilla* [45] será utilizado para criar as cópias do *CTFBOX*.

5.6 Síntese

Neste capítulo, apresentamos os requisitos funcionais e os não funcionais, através dos quais definimos uma arquitetura possível.

Na seção 5.3 Arquitetura Lógica apresentamos os componentes, e na subseção 5.4 Arquitetura Física, referimos como serão organizados fisicamente.

O capítulo seguinte é destinado aos detalhes de implementação desta arquitetura.

Capítulo 6

Implementação

O presente capítulo contempla os detalhes de implementação da arquitetura proposta no capítulo 5 na Fig.5.1, tendo sido organizado nas seguintes secções:

- a secção 6.1 apresenta os detalhes de implementação do *Web Proxy*;
- a secção 6.2 descreve os serviços web implementados;
- na secção 6.3 detalhamos o processo de automação;
- a secção 6.4 refere-se ao servidor de armazenamento;
- na secção 6.5 focamos a implementação das plataformas de virtualização;
- na secção 6.6 apresentamos os detalhes da *CTF Toolkit*;
- na secção 6.7 apresentamos o sistema CTFBox;
- por último, na secção 6.8 referimos os aspetos necessários para realizar uma atividade com os alunos do MSI.

A plataforma CTF@DEI foi idealizada para entusiastas por segurança informática, sendo que, esta é extensível (suporta várias plataformas de virtualização ou *vulnerability assessment*). Assim, propomos na Fig.6.1, uma *framework*, que pode ser utilizada como referência. Definimos as diretrizes da *framework* do seguinte modo:

- A azul, enumeramos os serviços que interagem com os utilizadores (detalhes de implementação nas seções: 6.1 *"Web Proxy"*, 6.2 "Serviços Web", e em 6.5 "Plataformas de virtualização");
- *Automation Service*, componente que sincroniza os exercícios CTF com os outros serviços (detalhes de implementação em 6.3);
- *Launch Service*, que é responsável pela orquestração dos exercícios CTF nas várias plataformas de virtualização previamente registadas (detalhes de implementação em 6.6);
- *Storage*, onde armazenamos os vários *assets* dos exercícios CTF: as máquinas virtuais (*File Storage*), as imagens *Docker* (Docker Images) e a base de dados com os dados dos utilizadores (*Database* e *Cache*);

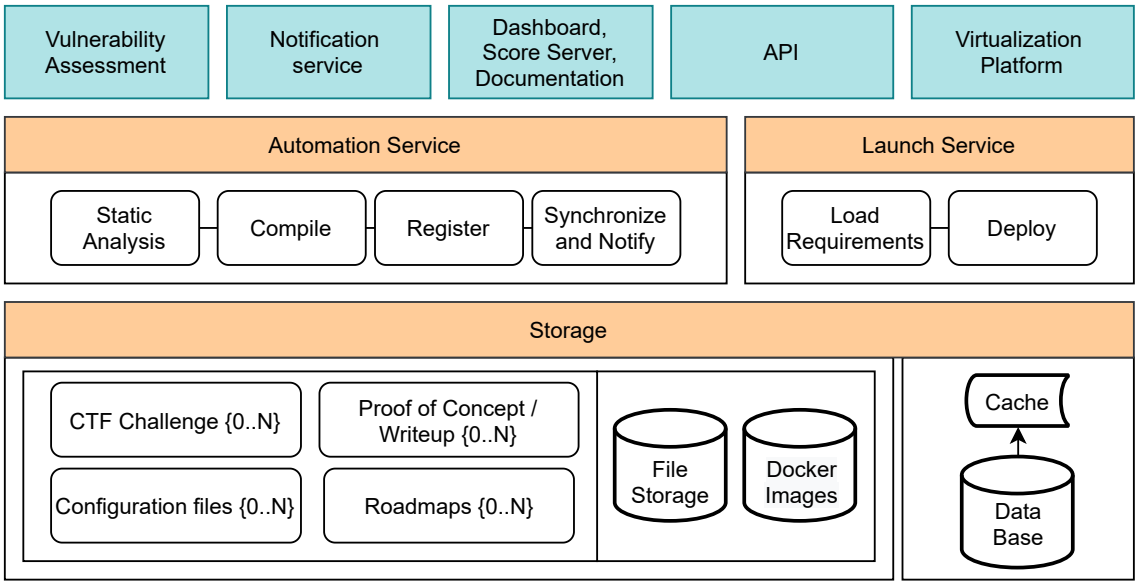

Figura 6.1: Framework proposta para a plataforma CTF@DEI

6.1 Web Proxy

O servidor *web* foi implementado numa máquina virtual, na plataforma Cloud2 do DEI (plataforma de virtualização *XCP-ng*).

A máquina virtual foi criada com uma imagem do sistema operativo *Ubuntu 20.04 Server*. Durante a instalação do sistema operativo, criamos o administrador, a que chamamos "ctf", e instalamos o serviço *OpenSSH*.

No servidor *web*, criamos um novo utilizador, o qual utilizamos para aceder ao mesmo. A autenticação no serviço SSH é realizada através de chaves RSA.

No servidor foram instaladas várias ferramentas, necessárias para o nosso *setup*, nomeadamente: *vim-nox* (inclui *syntax highlighting*) [75], o *Git*, o *Docker* e o *Docker-compose* [78].

No servidor foi instalada e configurar a *firewall iptables*, com o *iptables-persistent*. Nesta *firewall* definimos um conjunto de regras, as quais expõem apenas os serviços *ssh* e *web*, bloqueando as restantes ligações.

Terminada a configuração base, iniciamos a instalação e a configuração do *web proxy*, tendo sido instalado o *Apache2* e *Certbot*.

O *Apache2* é utilizado como *reverse proxy*, uma vez que, os serviços / aplicações serão instalados em *containers*. Optamos, por configurar um *web proxy*, com o objetivo de suportar múltiplos serviços *https*, e para prevenir que pedidos mal-formados atinjam as aplicações.

No *web proxy* configuramos os diferentes *virtualhosts* com o protocolo *Transport Layer Security (TLS)*, tornando as ligações encriptadas. Os certificados foram gerados através do *Certbot* [185].

Para criar os certificados foi necessário registar os domínios. Para tal, recorremos aos serviços do DEI, no qual, registamos cinco nomes para o mesmo servidor *web*.

O nome *ctf.dei.uc.pt* é utilizado na plataforma *web* e nas ligações *SSH*, o nome *ctfsonar.dei.uc.pt* corresponde ao software que analisa o código dos exercícios, o nome *ctfchat.dei.uc.pt* corresponde ao serviço de notificações, o nome *ctfdocs.dei.uc.pt* corresponde à *wiki* / documentação, e por último, o *ctfmonitor.dei.uc.pt*, que corresponde ao serviço de monitorização.

Como os *virtualhosts* são semelhantes, e uma vez que, este processo repete-se, implementamos um *script*. Este *script*, cria um novo *virtualhost* através de um *template*, garantindo que são aplicadas as devidas configurações.

Por último, como existem especificidades com cada serviço / aplicação, então as configurações de cada *virtualhost* foram editadas, de modo, a configurar devidamente cada *reverse proxy*. Estes detalhes particulares sobre os *virtualhosts* encontram-se ilustrados no Apêndice 10.

Na secção seguinte, abordamos individualmente, os detalhes na implementação dos serviços / aplicações, deste servidor *web*.

6.2 Serviços Web

Na presente seção, mencionamos os detalhes de implementação dos serviços *web*, que foram instalados e expostos através do *Web Proxy*, previamente apresentado na secção 6.1. Para facilitar a pesquisa, a seção foi divida nas seguintes subsecções:

- Seleção da *framework* CTF - onde justificamos a seleção da *framework* CTF utilizada;
- CTFd - detalhes de implementação da plataforma CTFd;
- Página *Wiki* - detalhes de implementação da página de documentação;
- Serviço de *Vulnerability assessment* - detalhes de implementação do *Sonar Qube*;
- Serviço de *chat* - detalhes de implementação do *Rocket Chat*;

6.2.1 Seleção da *framework* CTF

Ao contrário dos outros serviços (*Wiki*, *Vulnerability assessment* e *chat*), que podem ser facilmente substituídos, a *framework* CTF é um dos componentes mais relevantes, porque será responsável por:

- gestão dos utilizadores e dos exercícios (operações create, read, update e delete);
- interfaces *web* para apresentar os *roadmaps*, os exercícios e permitir a submissão de *flags*;
- *scoreboard* e gráficos, que apresentam as atividades realizadas e as estatísticas dos alunos.

Durante a fase de seleção da *framework* CTF, experimentamos na prática, várias possibilidades *open source*.

Identificamos que a maioria das plataformas existentes é destinada às competições CTF, porém, entendemos que para uma plataforma de treino e preparação para as competições,

podemos criar outras funcionalidades, pelo que, começamos por desenhar um *website*, com funcionalidades para os alunos e professores cooperarem. No apêndice 10, incluímos o desenho deste *website* e respetivos *mockups* elaborados.

Com base nesta ideia, procuramos selecionar uma *framework*, que fosse robusta e segura, mas também com uma arquitetura suficientemente modular, a fim de serem introduzidas novas funcionalidades, por exemplo, definir *roadmaps* didáticos.

Tendo em conta a lista das várias plataformas CTF, apresentada na secção 2.1, optamos por analisar e testar plataformas do tipo *jeopardy*, que suportem a gestão dos exercícios, das *flags*, bem como, definir grupos de exercícios para representar os cenários dos *roadmaps*. Sendo que, o *Root the Box* e o *CTFd* preenchem estes requisitos.

A framework *Root the Box* é *open source*, representa um jogo interativo de segurança informática [230], onde podemos criar organizações fictícias, que podem ser encadeadas por níveis. Os exercícios são organizados e distribuídos nestas organizações (as quais idealizamos como *roadmaps*).

No *Root the Box* exploramos a gestão dos utilizadores, das organizações fictícias e os gráficos de progresso que são apresentados aos participantes.

A framework *Root the Box* foi pensada para gerir "boxes", isto é, máquinas virtuais vulneráveis, estas representam as organizações. Este jogo funciona através de missões, e por cada organização conquistada, os participantes recebe pontos. Os pontos permitem progredir no jogo, adquirir ajudas e obter acesso ao código fonte dos exercícios.

Através do *Github*, colocamos uma questão aos criadores do *Root the Box*, a fim de saber, se já tinham criado uma coleção de exercícios CTF para o *Root the Box* (*issue* [24]). Os criadores do *Root the Box* responderam que ainda não existe uma coleção, embora tenham compatibilidade com o *OWASP Juice Shop* [195] (projeto com vários exercícios CTF sobre *web*), mencionam também, que criar um repositório de exercícios era algo que também já tinham idealizado fazer.

CTFd é uma *framework*, frequentemente utilizada para desenvolver plataformas e exercícios CTF, de fácil utilização, e que possam ser customizados [59]. Esta *framework* define uma arquitetura baseada em *plugins*, que permite estender as funcionalidades originais do *CTFd* [107].

No *CTFd*, os exercícios são geridos no modo *Jeopardy* e podem ser organizados por categorias. Os dados de progresso dos participantes podem ser exportados / importados. O quadro de pontuações menciona os pontos de cada participante, e ao longo do tempo, o *scoreboard* apresenta o progresso dos participantes.

Exploramos as várias interfaces *Web* e a API existentes para o *CTFd*, do ponto de vista dos administradores e dos participantes. Sendo que, no caso da API, encontramos um bug, que foi notificado na *issue* [106]. O componente "dynamic_challenges"utiliza uma fórmula matemática, que define o valor de pontos de um exercício. Esta função é exposta pela API e permite que se execute zero a dividir por zero, o que gera uma exceção, que não é capturada no código.

6.2.2 Framework CTFd

Como indicado na arquitetura, optamos por utilizar a *framework CTFd*, dado que permite gerir os exercícios e os utilizadores, é extensível através de *plugins*, e compatível com os serviços do *CTFTime* [61].

A instalação de uma aplicação baseada na *framework CTFd*, pode ser realizada de várias maneiras diferentes, com sistemas de base de dados também diferentes. Neste projeto, optamos pela implementação descrita na Fig.6.2, que é segmentada nos seguintes componentes:

- O primeiro *container Docker* corresponde à base de dados *MySQL*, que permite guardar os dados do *website*;
- O segundo *container Docker* é o serviço *Redis*, que guarda os dados na memória, para diminuir o atraso no processamento dos mesmos;
- O terceiro *container* corresponde ao servidor *Flask*, que disponibiliza o *website CTFd* e a sua API.

Os três *containers* encontram-se interligados através de uma *Bridge Network*, que permite que os *containers* conectados à mesma rede comuniquem, enquanto fornecem isolamento [79].

Através de *Bind mounts*, montamos os diretórios do próprio servidor *web* (File System (FS)), dentro dos *containers*. Optamos por esta abordagem, porque permite gerir as configurações e analisar os *logs* facilmente [80].

O *container CTFd* foi exposto com mapeamento de portos, o que faz com que a aplicação responda no *localhost*, no porto 8000. O Apache expõe este serviço na interface *eth0*, através de um *proxy reverso*, em *ctf.dei.uc.pt*.

O *script* que implementa este *setup* encontra-se disponível no repositório "ctf-setup"(ver anexo 10).

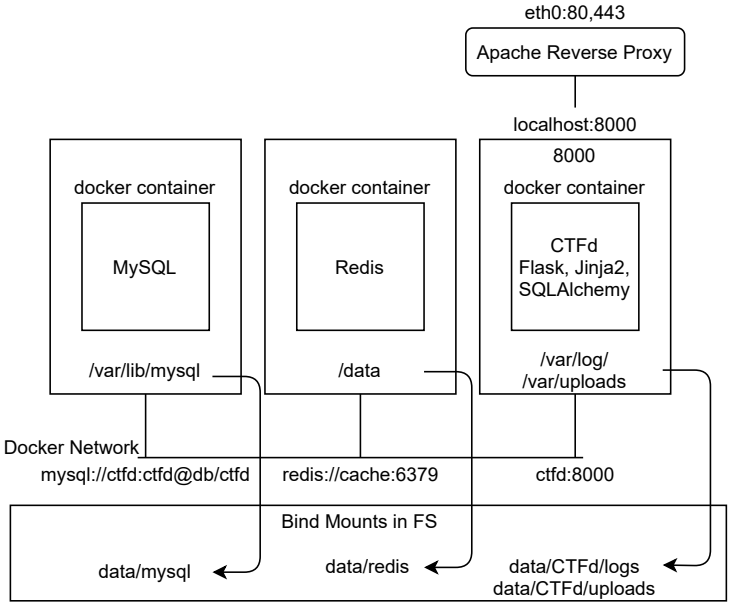

Figura 6.2: Implementação Docker da plataforma CTFd

6.2.3 Página Wiki

Em relação à página de documentação, instalamos a ferramenta *Jekyll* [138], através de um *container Docker*, e configuramos um *Bind mount*, para que o *Jekyll* tenha acesso aos

ficheiros *markdown*.

Quando os ficheiros *markdown* são alterados, este serviço, *Jekyll*, gera o código HTML e atualiza a página *web*. O *virtualhost* (*Apache2*) foi configurado para mostrar apenas o conteúdo da pasta *webroot* (ficheiros *HTML*).

A título de exemplo, a Fig.6.3, apresenta um dos *writeups* de um exercício CTF, a página foi construída automaticamente através do respetivo ficheiro *markdown*.

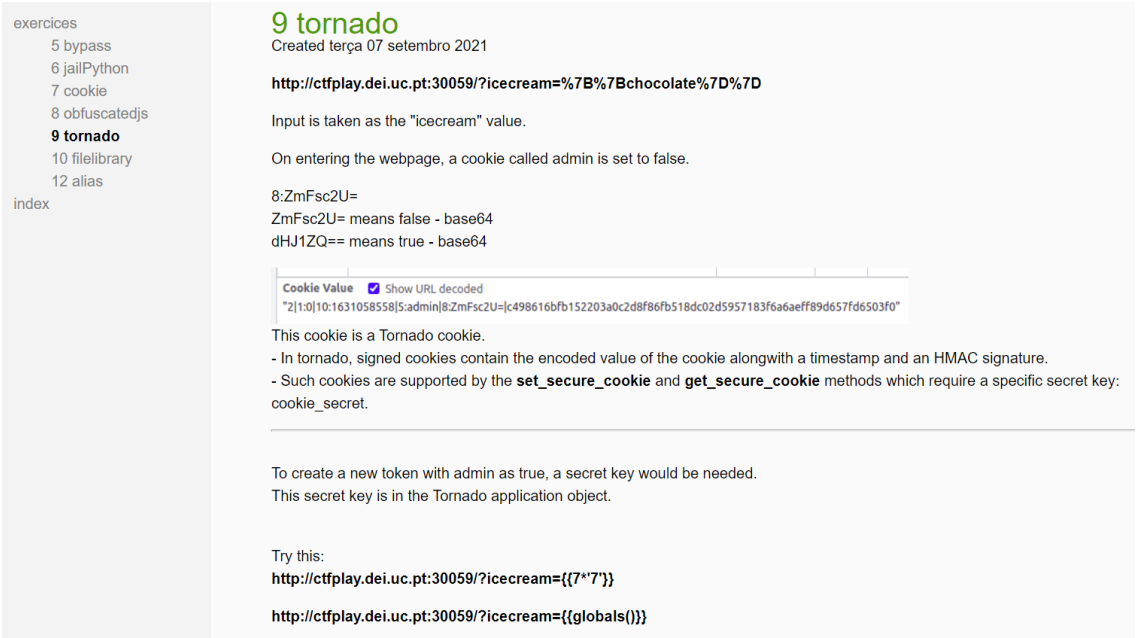

Figura 6.3: Website de documentação (ctfdocs)

O *script* desenvolvido, que implementa este *setup* encontra-se disponível no repositório "ctf-setup"(ver anexo 10).

6.2.4 Serviço de Vulnerability assessment

Para o serviço de *vulnerability assessment*, optamos por recorrer ao *Sonar Qube*, por suportar várias linguagens de programação.

A instalação do serviço *Sonar Qube* em produção encontra-se descrita na Fig. 6.4, e foi realizada através dos seguintes componentes:

- O primeiro *container Docker* corresponde à base de dados *PostgreSQL*, que persiste os dados da aplicação *Sonar Qube*. Foram criados dois volumes *Docker*, um para os dados do *PostgreSQL* e outro para as configurações;
- O segundo *container* corresponde ao servidor *Sonar Qube*. Foram criados quatro volumes *Docker*, para guardar, respetivamente, os dados, as configurações, as extensões e os *plugins*.

Os dois *containers* encontram-se interligados através de uma *Bridge Network* interna, que permite que os dois *containers* comuniquem entre si.

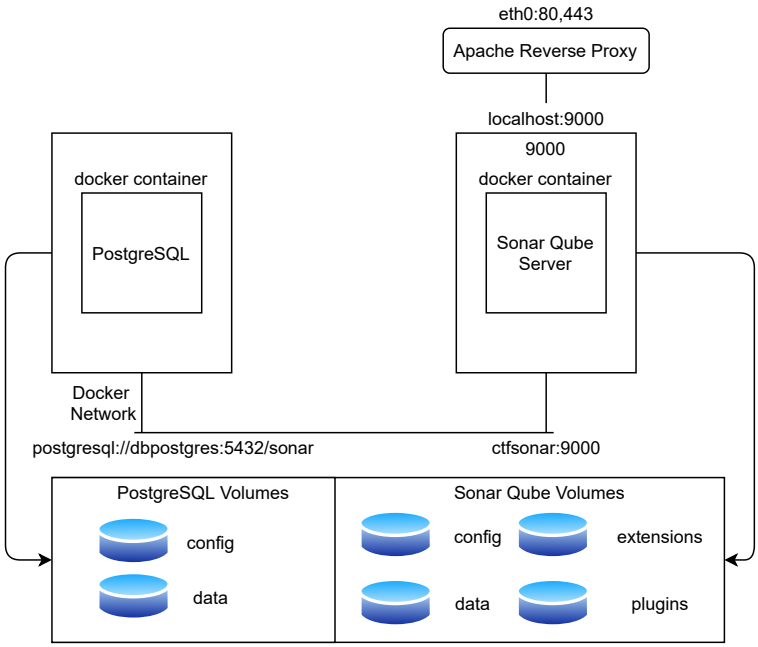

Figura 6.4: Implementação Docker do serviço Sonar Qube

Para expor o *container Sonar Qube*, mapeamos o porto 9000 no *localhost*, permitindo a configuração do *proxy reverso ctfsonar.dei.uc.pt*.

O valor padrão no *Ubuntu* para o número de áreas de memória é 65536, e nos experimentos realizados com o *Sonar Qube*, este valor mostrou ser insuficiente, causando instabilidade. Para que o *Sonar Qube* funcione corretamente, alteramos a diretiva *vm.max_map_count* e multiplicamos o valor padrão por quatro.

Apresentamos na Fig.6.5, a interface do *Sonar Qube*, que apresenta os resultados da análise estática realizada para cada um dos exercícios CTF.

6.2.5 Serviço de chat

Para implementar o serviços de *chat*, recorremos ao *Rocket Chat*, que é uma aplicação utilizada em várias organizações, e que pode ser extensível através da sua API.

A instalação do serviço de *chat* em produção encontra-se descrita na Fig. 6.6, e foi realizada através dos seguintes componentes:

- O primeiro *container Docker* corresponde ao *MongoDB*. Optamos por este sistema porque o *Rocket Chat* utiliza bases de dados NoSQL [37];
- O segundo *container* corresponde ao servidor *Rocket Chat* e à respetiva API;

Na Fig.6.7 descreve como interligamos o *Rocket Chat* com os restantes componentes da plataforma CTF@DEI, tendo sido implementadas as seguintes ligações:

- *WebHooks* - os serviços de automação (*Gitlab Executors*) enviam atualizações para o *chat*, através de *WebHooks*;

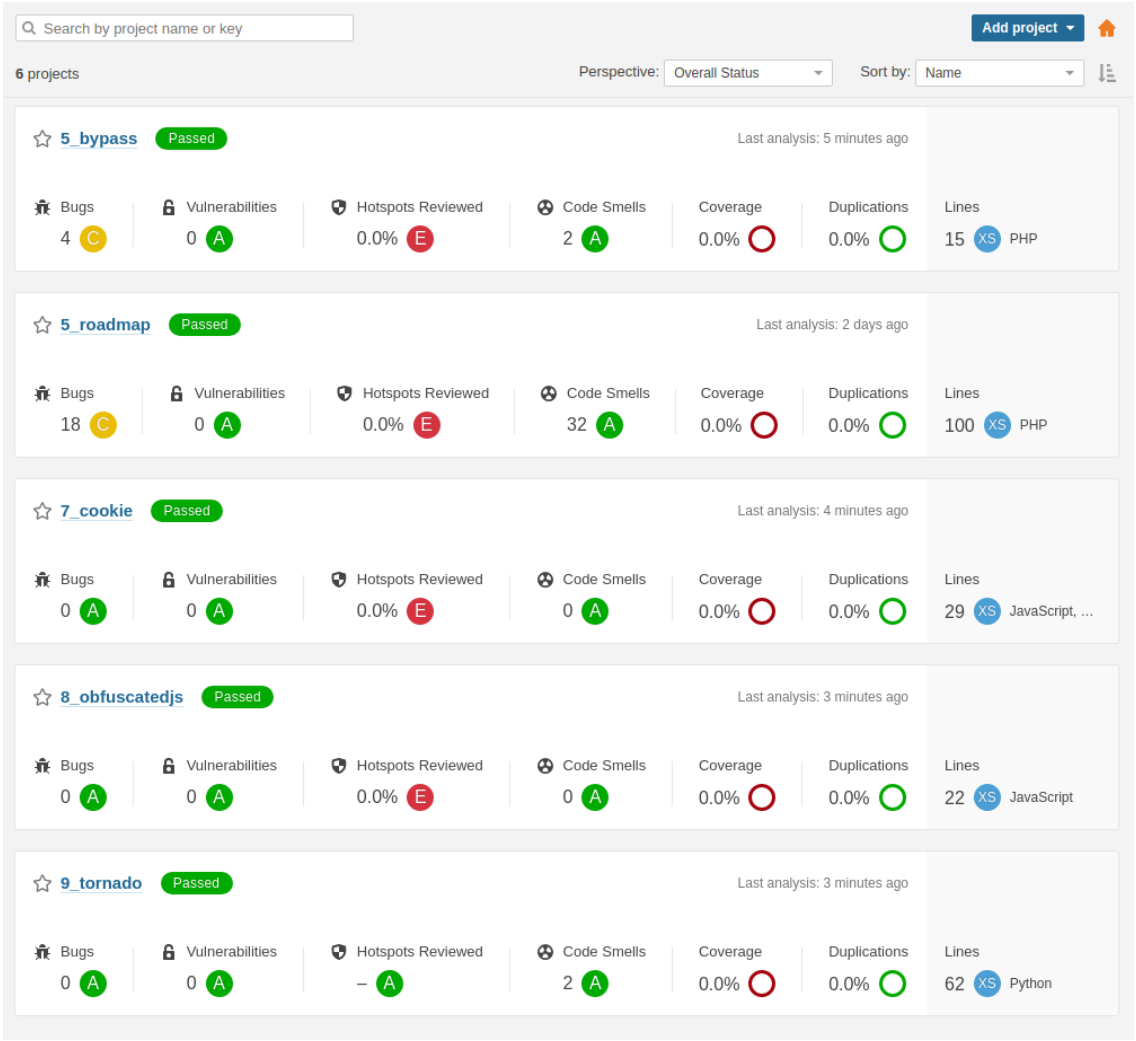

Figura 6.5: Interface do serviço Sonar Qube (ctfsonar)

- WebApp Interfaces - disponibiliza interfaces para serem incluídas nos restantes *frontends* de outros sistema, por exemplo, através de *iFrames*. Estas interfaces são personalizáveis nas definições do servidor. Para a interface do *website* e para a aplicação móvel, optamos por utilizar as interfaces padrão;
- API - utilizamos a API do *Rocket Chat* na *CTF Toolkit*, para criar / arquivar os canais dos exercícios, e para enviar mensagens;
- *SSL Connections* - as ligações com o servidor LDAP são realizadas através de túneis *SSL*.

Na secção seguinte, abordamos os detalhes na implementação do serviço de automação.

6.3 Serviços de automação

Na arquitetura apresentada no capítulo 5, dividimos o serviço de automação em três funções: uma que classifica o código do exercício; outra que gera o objeto executável a partir do código fonte; e a terceira que sincroniza os dados dos exercícios e notifica os utilizadores.

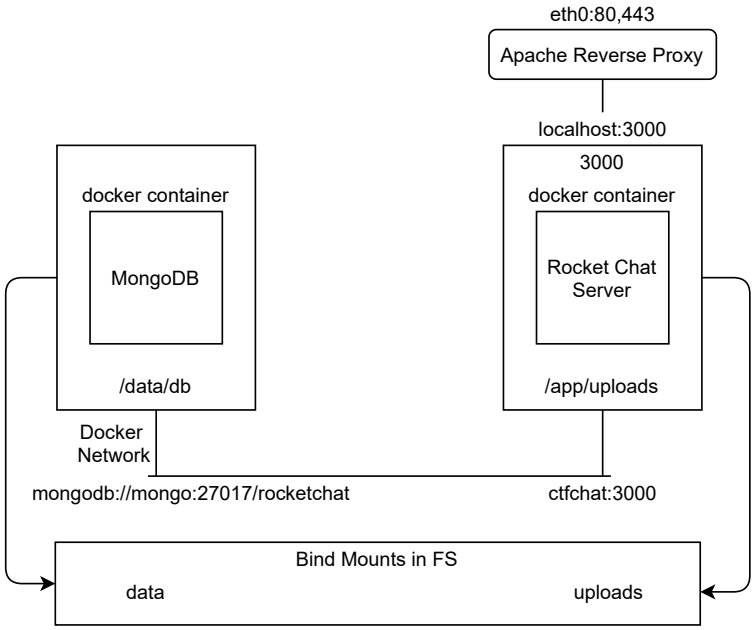

Figura 6.6: Implementação Docker do serviço Rocket Chat

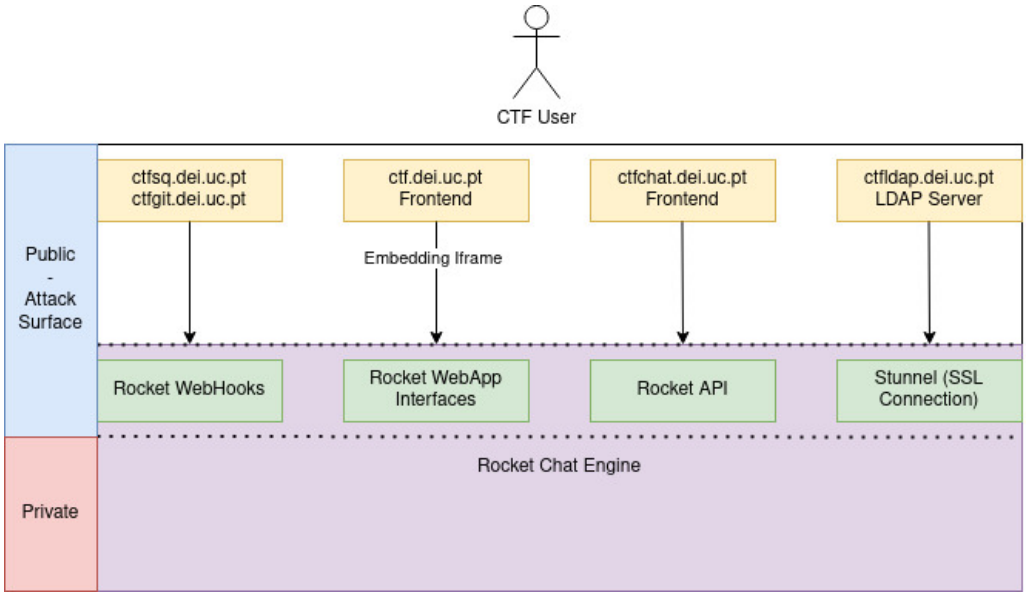

Figura 6.7: Conexões entre o serviço de chat e os restantes componentes da plataforma CTF@DEI

Como referido, este componente tem como objetivo gerir os exercícios automaticamente, isto é, o serviço acede à coleção CTF e sincroniza as informações com os restantes componentes, mantendo a plataforma coesa.

Pretendemos ainda que, sempre que ocorram novas alterações à coleção CTF, sejam executadas as funções referidas. Como a coleção de CTF será armazenada num repositório do GitLab (ver secção 6.4), optamos por utilizar *GitLab Runners*. Os *GitLab Runners* são responsáveis por executar as *pipelines* nos repositórios *Git* da plataforma *GitLab*.

Segundo a documentação [109], os *GitLab Runners SSH executors* são susceptíveis a ata-

ques MITM (*man-in-the-middle*), uma vez que, não possuem a opção *StrictHostKeyChecking*. Esta opção, verifica a autenticidade dos *hosts*, através de uma base de dados local, de forma a prevenir ataques de *spoofing* ou *man-in-the-middle*.

Por questões de segurança, optamos por não transferir parâmetros para o *GitLab Runner*, apenas enviamos um sinal sempre que sejam realizados *commits* ao projeto *git*. Assim, o *GitLab Runner*, ao receber este sinal, irá realizar *pull* ao projeto git, o que atualiza a coleção CTF local, e posteriormente, irá realizar as devidas funções que projetamos.

Estas funções encontram-se implementadas no executável *CTF Toolkit* (detalhes da sua implementação na secção 6.6). Este executável foi copiado para o *GitLab Runner*, para que este, possa gerir a plataforma automaticamente.

Por questões de simplicidade, a instalação do *GitLab Runner* foi realizada de forma manual, segundo a documentação em [110], tendo sido tomadas as seguintes medidas durante a implementação deste serviço:

- No repositório da coleção CTF registamos o *GitLab Runner*;
- Na raiz do repositório definimos uma *pipeline*, através do ficheiro *.gitlab-ci.yml*;
- Criamos, um novo utilizador, no sistema *Linux* chamado *gitlab-runner*, e na sua *home*, configuramos o programa *CTF Toolkit* e vários *tokens*, que concedem acesso aos outros componentes (*Sonar Qube*, *Rocket Chat* e *CTFd*);
- Quando existirem novos *commits* na *branch master*, a pipeline aciona o *GitLab Runner* através do protocolo SSH (por segurança, não transferimos nenhum parâmetro).

A estratégia implementada, que agrega a *CTF Toolkit* e o serviço de automação em conjunto (representado na Fig.6.8), realiza o seguinte procedimento:

- os alunos, através da rede do DEI, realizam *commits* no *Git Server* (são geradas notificações para o servidor de chat, através de *WebHooks*);
- o *GitLab Runner*, registado no projeto *git*, é acionado quando são realizados novos *commits*. O *GitLab Runner* executa as seguintes ações de automação:

  - envia os exercícios para o(s) sistema(s) de *vulnerability assessment*;

  - os exercícios são compilados (as imagens *docker* são registadas no *Docker Registry*);

  - a *dashboard* e o *scoreserver* são atualizados, com as informações dos exercícios e dos roadmaps;

- por último, os exercícios encontram-se disponíveis para execução, os quais, podem ser geridos através da *CTF Toolkit* (iniciar, pausar, terminar).

Na secção seguinte, abordamos os detalhes na implementação do servidor de *storage*.

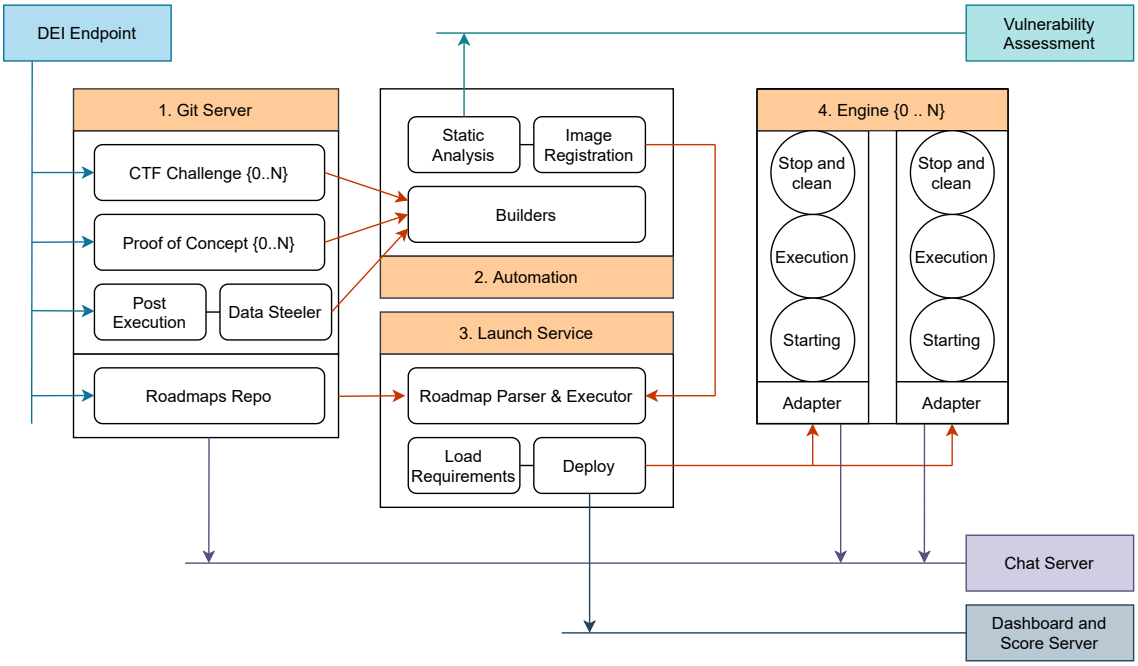

Figura 6.8: Estratégia de automação implementada na plataforma CTF@DEI

6.4 Storage

Na presente secção, mencionamos os detalhes de implementação da coleção de exercícios CTF e do servidor de *storage*.

Começamos por perceber, se já existem repositórios com vários exercícios CTF, e identificamos o repositório *CSAW-CTF* [192], referente a uma competição CTF. Os exercícios são organizados, segundo uma estrutura de directórios, que é a seguinte: nome da competição > categoria > pontos/nome do exercício.

Ao seja, parte das informações que precisamos estão nesta estrutura, outras informações, como os autores de cada exercício, não se encontram facilmente.

Entendemos que, esta estrutura possa ser melhorada em vários sentidos, uma vez que, não define uma ordem, nem nos fornece meta-dados relevantes (como o autor ou os requisitos de virtualização).

Surge então, a necessidade, de criamos um novo *standard* e definir um ficheiro de propriedades/configuração, que represente o exercício CTF.

Este ficheiro de propriedades tem como objetivo declarar todas as propriedades necessárias, para que o exercício seja importado para o *website* e executado na plataforma de virtualização, de forma automática.

Durante o desenho do ficheiro de propriedades/configuração, tivemos em conta as seguintes considerações:

- Os ficheiros de configuração iram tratar os exercícios CTF de forma abstracta, isto é, apenas indicam onde se encontram os objetos executáveis (por exemplo, uma máquina virtual ou uma imagem *Docker*). Desta forma, conseguimos lidar facilmente com a diversidade dos exercícios e suas tecnologias;

- Como referido, não existe um padrão para indicar os autores dos exercícios CTF (por vezes, encontram-se no código, outras vezes, encontram-se no *website* da competição). Este ficheiro contém todas as informações necessárias para representar o exercício CTF na interface *web*;
- O mesmo exercício poderá ser executado de várias maneiras, com diferentes recursos computacionais ou interfaces de rede. Pelo que, no ficheiro incluímos vários perfis de execução, onde definimos um perfil com os requisitos mínimos, outro com os requisitos recomendados, entre outros perfis com características especificas;
- A previsibilidade do exercício é um aspeto importante. Se o exercício for executado com um conjunto de regras bem definidas, então o seu comportamento passa a ser sistemático. Se optarmos por não definir estas condições, perdemos o controlo, e o comportamento dos exercícios poderá ser instável (por exemplo, alguns exercícios só funcionam corretamente no *Virtualbox*).

Para inserir o exercício no *website* é necessário indicar as seguintes propriedades específicas: o nome do exercício, a categoria, os pontos, o enunciado, as referências bibliográficas e os anexos. Assim, através de um ficheiro de configuração, definimos as seguintes propriedades:

1. Nome - O nome do exercício identifica o exercício na interface e tem de ser único, para permitir a identificação do *deployment* no contexto do *Kubernetes*;
2. *URL*'s - se aplicável, incluímos um *link* para identificar o repositório original, o *website* dos autores ou a competição CTF;
3. Autores - Lista dos autores ou das entidades que conceberam o exercício CTF;
4. Categoria/Roadmap - Atributo que identifica a categoria ou o *roadmap* do exercício. Na interface *web*, os exercícios são agrupados por grupos, estes grupos são formados através desta propriedade;
5. Palavras-chave (ou *tags*) - Lista que indentifica os conceitos utilizados no exercício, a fim de facilitar a pesquisa no *website*;
6. Enunciado - Corresponde à descrição do exercício CTF, que poderá ser escrito em HTML, o qual será posteriormente, integrado na interface *web* da plataforma CTF;
7. Pontos - Atributo que corresponde ao número de pontos do exercício, sendo este diretamente proporcional à sua dificuldade;
8. Visibilidade - Atributo que define se o exercício está ativo ou inativo. Ao definir como ativo, o exercício fica disponível para todos os utilizadores no *website*, caso contrário, permanecerá oculto;
9. Lista de Anexos - Lista que específica os ficheiros a considerar como anexos;
10. Lista de *Flags* - Lista que específica, quais são as *flags* do exercício (pode existir mais do que uma *flag* no mesmo exercício);
11. Lista de perfis de configuração - lista de perfis, que especifica os comandos que permitem executar o exercício CTF. Definimos os perfis de execução de acordo com a plataforma de virtualização do exercício, no caso do *Kubernetes*, utilizamos as seguintes propriedades: URL da imagem *Docker*, as características da rede, o mapeamento dos portos, os recursos a alocar e restrições de segurança (*namespace*)

Para armazenar a coleção de exercícios CTF utilizamos a ferramenta Git, tendo sido criado o repositório *"ctf-collection"*, de acordo com a Fig.6.9.

Cada exercício da coleção possui um diretório com um ID único, que contém: o sistema vulnerável (que poderá ser integrado através de um *submodule*); o ficheiro com as propriedades; e um diretório com os vários *writeups* e Prova de conceito (PoC) dos alunos.

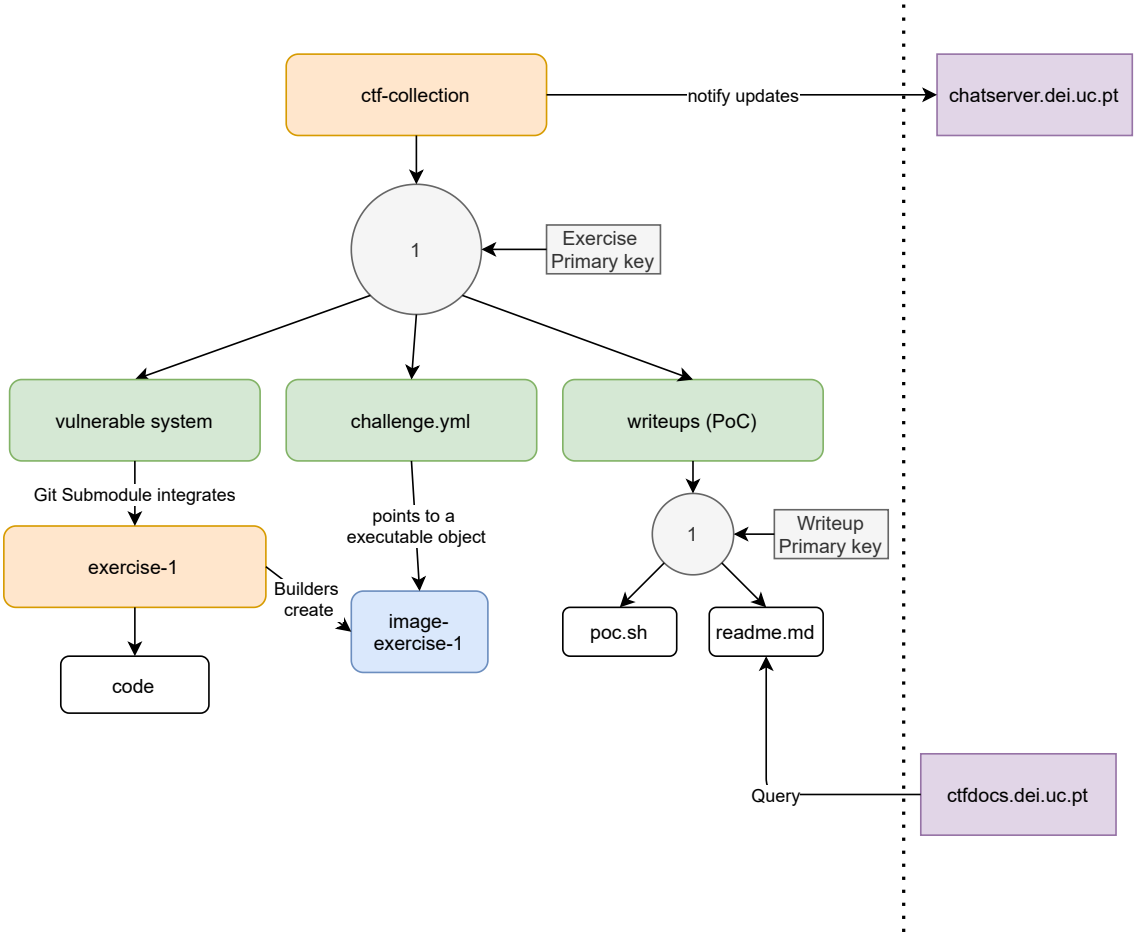

Figura 6.9: Estrutura do repositório da coleção CTF

Para armazenar a coleção de exercícios CTF utilizamos a ferramenta Git, em conjunto com a plataforma *GitLab*.

Optamos por utilizar a plataforma *GitLab*, uma vez que, o DEI já disponibiliza esta plataforma aos alunos, com autenticação *Single Sign-on*, através das contas do departamento.

Neste projeto não utilizamos a plataforma já existente no DEI, optamos por criar uma nova instância com a última versão da plataforma, a fim de realizar os experimentos de automação e não prejudicar as atividades dos outros projetos (disponível em *ctfgit.dei.uc.pt*).

Assim, outra máquina virtual, *Ubuntu 20.04* na plataforma Cloud2 do DEI, onde foi instalado o *GitLab*, de acordo, com o *script* presente no repositório *ctf-setup* (apêndice 10).

Para restringir os exercícios CTF aos membros registados e para isolar os exercícios dos sistemas que são legítimos, criamos um grupo fechado na plataforma *GitLab*, chamado "ctf". Ao utilizar *Git* é recomendado que os repositórios permaneçam abaixo de *1Gb* [4]. Para contornar esta situação e na medida que os exercícios de CTF vão incluir ficheiros de grandes dimensões, então o *Git* LFS foi activado para para o grupo "ctf".

Por último, integramos o *Docker Registry* no próprio Gitlab, através do pacotes *gitlab omnibus*. Definimos regras apropriadas na infraestrutura do DEI, para ser possível, aceder ao *Docker Registry*. Na Fig.6.10, apresentamos a interface do *Docker Registry*, com os exercícios CTF.

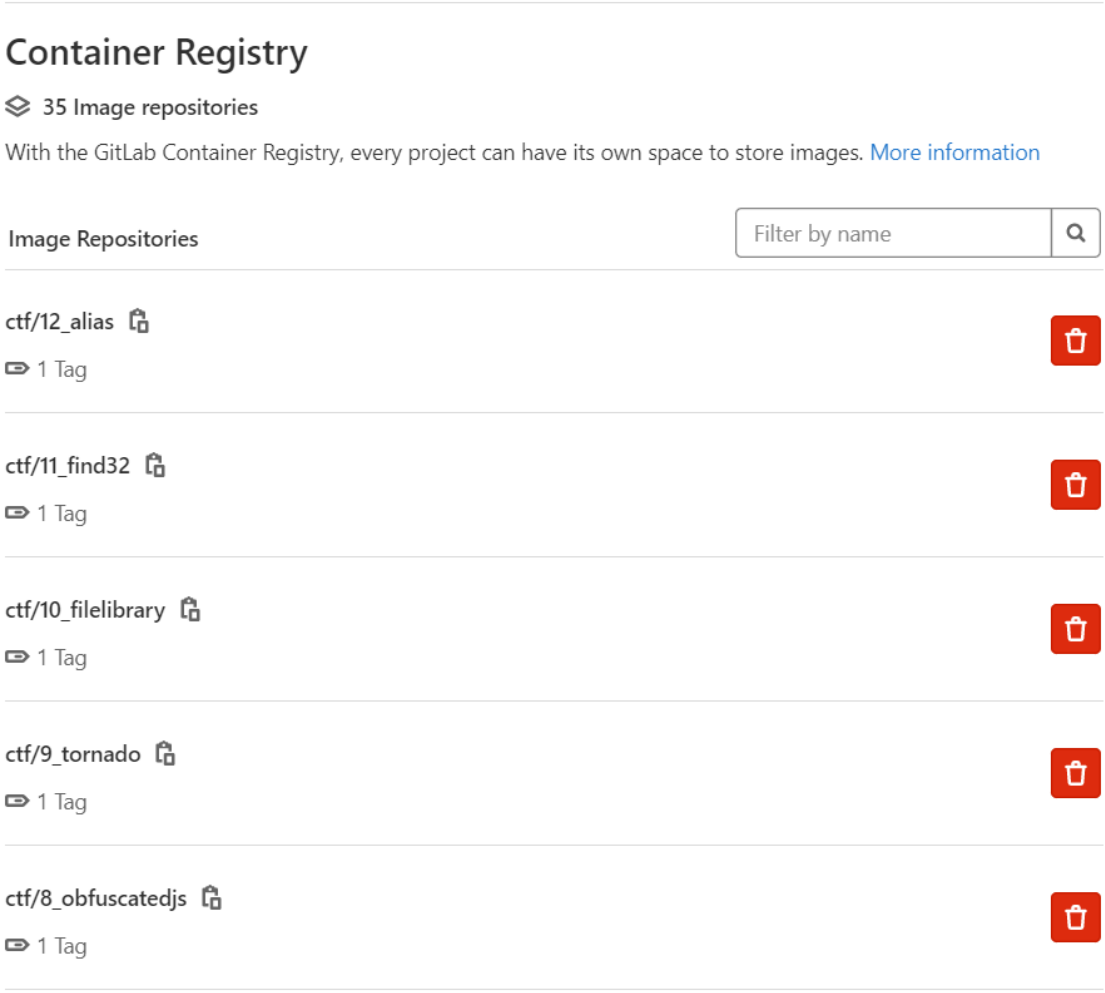

Figura 6.10: Interface do Docker Registry integrado no Gitlab

Na secção seguinte, abordamos os detalhes na implementação do serviço de armazenamento (*storage*).

6.5 Plataformas de Virtualização

Nas presente subseção, indicamos os detalhes de implementação das plataformas de virtualização, neste caso, a instalação e a configuração do *Kuberntes* e do *Virtualbox*.

6.5.1 Kubernetes

O plano que definimos para a plataforma *Kubernetes* encontra-se definido na Fig.6.11, onde temos como objetivo, criar um *Cluster*, com três máquinas virtuais, denominadas por *kn0*, *kn1* e *kn2*.

Figura 6.11: Cluster Kubernetes para a plataforma CTF@DEI

O *control plane* do *Kubernetes* (componente principal do *cluster*), foi instalado na máquina virtual *kn0*. As restantes máquinas virtuais, *kn1* e *kn2*, são utilizadas como *worker nodes*, logo, executarão a carga de trabalho útil, isto é, neste caso, os exercícios CTF.

No cenário (ver Fig.6.11), definimos duas redes, a de desenvolvimento e a de produção (Rede 1 e Rede 2). A rede de desenvolvimento é utilizada para testes, com *deploys* automáticos a partir das *pipelines* definidas no *GitLab*. A rede de produção contém os exercícios de CTF disponibilizados no *website*.

A *vlan* das máquinas virtuais, e que permite a comunicação com as mesmas, tem como *subnet*, a gama 10.3.0.0/24 (DEI Network). A rede do *Kubernetes*, e que permite a comunicação entre os *pods*, foi definida na gama 192.168.0.0/16.

Durante a configuração do *Kubernetes* na infraestrutura do DEI, deparamos-nos com várias situações, nomeadamente as seguintes: a reconfiguração do coredns; a definição de um *Service* por exercício; e a reconfiguração do *network layer* do *Kubernetes*.

1ª situação: DNS *queries* na rede do *Kubernetes* falham.

A primeira situação, levou-nos a alterar o *coredns* do *Kubernetes*, no *control plane*. O serviço DNS *default* do *Kubernetes* vem configurado com o DNS da *Google* (8.8.8.8). Embora esta configuração possa ser relevante para a maioria dos casos, no nosso cenário, tal configuração não é a ideal, porque o DEI disponibiliza um DNS interno, a fim de possibilitar as comunicações entre as aplicações e serviços internos existentes no DEI.

Para resolver esta situação, tivemos de reconfigurar o serviço DNS do *Kubernetes*, em particular o *pod coredns* - componente que faz parte do *control plane* e que resolve os nomes na rede interna, neste caso, a rede 192.168.0.0/16).

O *patch*, indicado na Fig.6.12, altera os *Upstream Name Servers* do *coredns*. Esta propriedade define quem são os servidores DNS, por defeito, nas redes do *Kubernetes*.

```
admin@ctfk8s-n0:~/setup/k8s-in-dei$ cat patch-cluster-dns.yml
apiVersion: v1
metadata:
  name: coredns-custom
  namespace: kube-system
data:
  upstreamNameservers: |
    ["193.136.212.1",]
kind: ConfigMap
admin@ctfk8s-n0:~/setup/k8s-in-dei$ cat 4-patch-cluster-dns.sh
#!/bin/bash

kubectl apply -f patch-cluster-dns.yml
kubectl -n kube-system rollout restart deployment coredns
#kubectl logs
```

Figura 6.12: Patch ao serviço DNS do DEI

2ª situação: Os *pods* estão isolados e não comunicam com o exterior.

Segundo a documentação do *Kubernetes*, para disponibilizar um serviço para os utilizadores, podemos optar por umas das seguintes abordagens: *CLusterIP*, *NodePort* e *LoadBalancer* [149].

Para os exercícios CTF, optamos pela abordagem *LoadBalancer*, o que significa que as comunicações são geridas por um *load balancer* externo diretamente conetado com os pods em 192.168.0.0/16, logo, o *Kubernetes* deixa de ser responsável pelas regras de balanceamento de carga.

Para cada exercício CTF é criado um *Deployment* e um *Service*. No *Deployment* indicamos o número de réplicas e o link para o *Docker Registry*, onde se encontra a imagem do exercício CTF. O *Service*, do tipo *LoadBalancer*, expõe então estes pods, no IP e porto especificado, neste caso, nos endereços: 10.3.2.215:30034 e 10.3.2.216:30034 (ver Fig.6.13).

```
apiVersion: v1
kind: Service
metadata:
  name: <tag_ctf_name>
  labels:
    category: <tag_roadmap_name>
    challenge: <tag_ctf_name>
spec:
  type: LoadBalancer
  externalIPs:
  - 10.3.2.215
  - 10.3.2.216
  selector:
    category: <tag_roadmap_name>
    challenge: <tag_ctf_name>
  ports:
    - port: 80
      name: port-80
      targetPort: 80
      nodePort: 30034
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: <tag_ctf_name>
  labels:
    category: <tag_roadmap_name>
    challenge: <tag_ctf_name>
spec:
  replicas: 3
  selector:
    matchLabels:
      category: <tag_roadmap_name>
      challenge: <tag_ctf_name>
  template:
    spec:
      imagePullSecrets:
      - name: ctfgit
      containers:
      - name: <tag_roadmap_name>
        image: ctfgit:5050/ctf/<ref>
        ports:
        - containerPort: 80
          name: port-80
```

Figura 6.13: Definições genéricas para criar os exercícios CTF no K8s

3ª situação: As comunicações entre *pods* de *worker nodes* diferentes falham.

As comunicações realizadas entre *pods* de *worker nodes* diferentes são bloqueadas, porque a rede exterior desconhece a rede interna do *Cluster Kubernetes*, neste caso, a rede 192.168.0.0/16. Verificamos inconsistências intermitentes nos exercícios CTF, uma vez que, não obtemos resposta a todos os *requests* enviados.

Quando testamos a conetividade, verificamos que a comunicação entre *pods* no mesmo *worker node* funciona, mas entre *pods* de diferentes *worker nodes* não existe conetividade (*time out*).

As comunicações realizadas entre *pods* de *worker nodes* diferentes falham, porque a rede exterior não faz as operações de *routing* da rede 192.168.0.0/16 e descarta os *packets*.

Assim, para que o *Cluster Kubernetes* opere na infraestrutura da *Cloud2* do DEI optamos por reconfigurar e estender o *network layer*, através de *Network Plugins*. Sem interferir com as configurações da própria infraestrutura, testamos várias abordagens, em particular, o modo *IP-in-IP* e o modo *VXLAN*.

Para comparar estes dois modos, configuramos ambos, e analisamos as suas diferenças, quer nas tabelas de *routing*, quer através da ferramenta *tcpdump* (ver Fig.6.14).

No modo *IP-in-IP*, os pacotes com destino a um *pod*, cujo o IP encontra-se compreendido no intervalo entre 192.168.1.193 e 192.168.1.254 (máscara 255.255.255.192 ou /26), são entregues ao gateway 10.3.2.215, através da interface *tunl0*.

Observamos ainda que, no modo *IP-in-IP*, os pacotes enviados para a rede 10.3.0.0/24 foram encapsulados, na medida que, possuem dois cabeçalhos IP. O primeiro cabeçalho permite a comunicação na rede do DEI, neste caso, é enviado do *worker node* 10.3.2.215, com destino ao *worker node* 10.3.2.216. E o segundo cabeçalho, permite que a comunicação entre os *pods* ocorra, uma vez que, no processo de desencapsulamento, sabemos que a comunicação foi originada no *pod* 192.168.1.194 e destina-se ao *pod* 192.168.2.196.

Ao passo que, o modo *IP-in-IP* permite-nos a criação de redes virtuais em *layer 3*, o modo VXLAN distingue-se, na medida que, permite-nos criar redes virtuais em *layer 2*, isto é, as *frames Ethernet* são encapsuladas em pacotes UDP.

Com o modo VXLAN implementado, verificamos que os *worker nodes* trocam pacotes UDP, com as informações do protocolo VXLAN (*flags, vxlan network identifier*). O cabeçalho *Ethernet* e o cabeçalho IP possuem, respetivamente, os *mac-address* e os endereços IP das interfaces *eth0* dos *worker nodes kn1* e *kn2*.

No modo VXLAN, o pacote UDP contém um outro pacote, com uma novo cabeçalho (*frame Ethernet* e IP), por sua vez, este cabeçalho interior *frame Ethernet* contém os *mac-address* das interfaces virtuais *vxlan.calico* e os endereços IP dos *pods*.

Consideramos que, em situações que o *Cloud provider* não suporta o IP-in-IP ou a implementação é realizada em redes das quais não temos controlo, o VXLAN é uma abordagem possível, embora, este modelo exija mais processamento, para encapsular / desencapsular os pacotes.

No nosso cenário, optamos por utilizar o modo VXLAN (overlay networks), o qual, foi implementado através do *Calico CNI*. Os ficheiros de instalação e configuração do *Calico CNI* encontra-se disponíveis no repositório do projeto (ver Apêndice 10).

```
DIRECT:
        # route table
        Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
        192.168.1.192  10.3.2.215       255.255.255.192 UG    0      0        0 eth0
        192.168.2.192  10.3.2.216       255.255.255.192 UG    0      0        0 eth0
        # tcpdump - default
        Ethernet II, Src: 9a:7b:5d:5e:f9:14, Dst: 16:25:ff:ed:dc:b6
        Internet Protocol Version 4, Src: 192.168.1.194, Dst: 192.168.2.196
IP-in-IP:
        # route table
        Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
        192.168.1.192  10.3.2.215       255.255.255.192 UG    0      0        0 tunl0
        192.168.2.192  10.3.2.216       255.255.255.192 UG    0      0        0 tunl0
        # tcpdump - IP in IP
        Ethernet II, Src: 9a:7b:5d:5e:f9:14, Dst: 16:25:ff:ed:dc:b6
        Internet Protocol Version 4, Src: 10.3.2.215, Dst: 10.3.2.216
        Internet Protocol Version 4, Src: 192.168.1.194, Dst: 192.168.2.196
VXLAN
        # route table
        Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
        0.0.0.0         10.3.0.254      0.0.0.0         UG    100    0        0 eth0
        10.3.0.0        0.0.0.0         255.255.0.0     U     0      0        0 eth0
        10.3.0.254      0.0.0.0         255.255.255.255 UH    100    0        0 eth0
        192.168.11.241  0.0.0.0         255.255.255.255 UH    0      0        0 caliccd5757287b
        192.168.11.242  0.0.0.0         255.255.255.255 UH    0      0        0 cali06e83864c85
        192.168.11.243  0.0.0.0         255.255.255.255 UH    0      0        0 cali5131286e758
        192.168.221.192 192.168.221.192 255.255.255.192 UG    0      0        0 vxlan.calico
        # tcpdump - VXLAN
        Ethernet II, Src: 9a:7b:5d:5e:f9:14, Dst: 16:25:ff:ed:dc:b6 (worker-nodes)
        Internet Protocol Version 4, Src: 10.3.2.215, Dst: 10.3.2.216
        User Datagram Protocol
        Ethernet II, Src: 66:52:49:17:3e:c1, Dst: 66:0a:05:85:f8:00 (vxlan)
        Internet Protocol Version 4, Src: 192.168.10.194, Dst: 192.168.11.242
        Transmission Control Protocol
```

Figura 6.14: Modos de rede no K8s: Direct, IP-in-IP e MxVLAN

6.5.2 Virtualbox

Durante a recolha de exercícios CTF (ver lista de *websites* no Apêndice 10), verificamos que parte dos exercícios CTF são criados e exportados através do *Virtualbox*.

Em vários exercícios do *Vulnhub*, verificamos que os autores do exercícios alertam, para o facto, de que a máquina virtual foi exportada no *Virtualbox*, e portanto, irá funcionar melhor no *Virtualbox*, do que no *VMware*. Assim, não querendo excluir este nenhum formato, optamos por preparar um computador com o *Virtualbox*.

Preparamos um computador, com o sistema *Ubuntu 20.04 Desktop*, e posteriormente, instalamos o *Virtualbox*. O processo de instalação foi realizado através da *CTF Toolkit*, que neste caso, é uma *wrapper function*, que encapsula os comandos em *bash* (ver Fig.6.15).

Registar os processo em funções (mesmo que seja uma tarefa simples, como esta), garantenos que, os alunos que utilizam a *CTF Toolkit*, estão a instalar a mesma versão do *Virtualbox*, exatamente nas mesmas condições. Logo, a experiência que terão com o exercício CTF estará, à partida, assegurada.

6.6 CTF Toolkit

Na seção anterior, descrevemos os detalhes de implementação das plataformas de virtualização, estas responsáveis pela execução dos exercícios CTF.

A *CTF Toolkit* implementa a lógica, entre o repositório e as plataformas de virtualização,

```
ctf.virtualbox.install(){
    README="
# Install the virtualbox 6.1: ctf.virtualbox.install do
#
# Usage: ctf.virtualbox.install do
"
if [ "$#" -eq 0 ]
then
    echo "$README"
else
    wget -q https://www.virtualbox.org/download/oracle_vbox_2016.asc -O- | sudo apt-key add -
    wget -q https://www.virtualbox.org/download/oracle_vbox.asc -O- | sudo apt-key add -

    echo "deb [arch=amd64] http://download.virtualbox.org/virtualbox/debian $(lsb_release -cs) contrib" |
    sudo tee -a /etc/apt/sources.list.d/virtualbox.list

    sudo apt update
    sudo apt install virtualbox-6.1

    wget https://download.virtualbox.org/virtualbox/6.1.8/Oracle_VM_VirtualBox_Extension_Pack-6.1.8.vbox-extpack
    sudo VBoxManage extpack install Oracle_VM_VirtualBox_Extension_Pack-6.1.8.vbox-extpack
fi
}
```

Figura 6.15: Wrapper function da CTF Toolkit - virtualbox

de modo, a que seja possível realizar as seguintes operações:

1. *Exercise Manager* - manipula os exercícios nos vários componentes da plataforma;
2. *Execution Control* - permite a gestão dos exercícios nas plataformas de virtualização, isto é, instala, inicia, pausa e pára um exercício CTF;
3. *Static Analysis* - programa que submete o exercício a uma análise estática (utilizado pelo serviço de automação);
4. *Compile and Register* - programa que transforma o código fonte do exercício executável, e gera, uma URL que o identifica no servidor de armazenamento (utilizado pelo serviço de automação);
5. *Synchonize and Notify* - processa o ficheiro de configuração do exercício, atualiza o *website* com essas informações e gera uma notificação (utilizado pelo serviço de automação).

A *CTF Toolkit* é composta por dois ficheiros: um ficheiro *jar*, chamado *exercise manager*; e um ficheiro bash, o qual, adicionamos à lista de alias da *sheel linux*.

O ficheiro *exercise manager* foi desenvolvido em *Java 11* e compilado através do *Maven*. Este componente foi implementado através de uma *framework Java* já existente, chamada *Picocli*. Esta framework permite-nos conceber aplicações para a *sheel* do *linux*.

O código desenvolvido para o *exercise manager* divide-se em quatro partes:

- aplicação - que recebe os parâmetros e executa a ação correspondente;
- acções - conjunto de *callbacks*, que são chamadas, quando é executado um comando;
- gestor de ficheiros - conjunto de classes, que permite carregar e guardar os dados na base de dados local;
- *util* - estabelece as ligações com os componentes da infraestrutura;

As acções correspondem às classes presentes no diagrama da Fig.6.16. Estas são responsáveis por enviar a informação dos exercícios para os vários componentes da plataforma. Todas as acções implementam a interface *Callable*, sendo que, o método *call* é chamado quando executamos o respectivo comando no terminal.

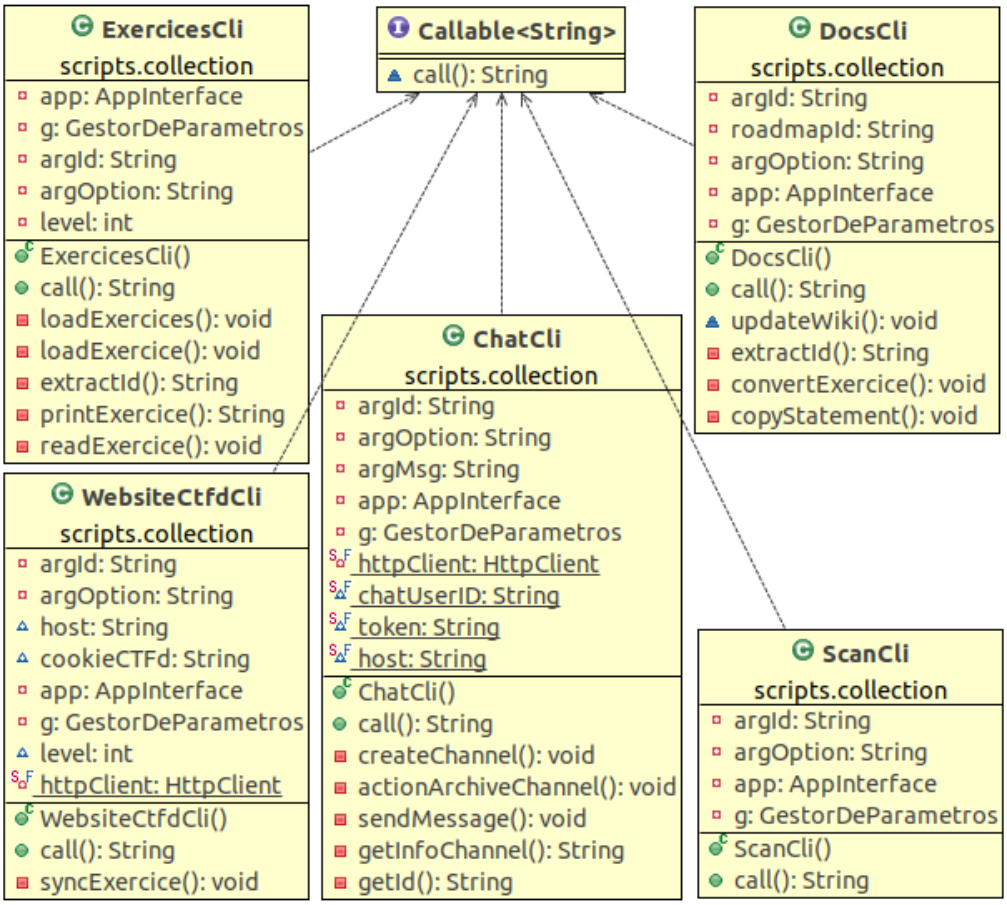

Figura 6.16: CTF Toolkit - implementação das callbacks da CTFToolkit

As ligações com os componentes da plataforma são realizadas através de túneis SSH, ligações SFTP e *requests* às API's. Para que o binário final, seja portátil e não dependa de dependências externas, então as ligações SSH e SFTP foram implementadas através da biblioteca *JSch* [137].

Para suportar a implementação referida no diagrama Fig.6.16 tivemos em consideração as comunicações que são estabelecidas, em particular, o SSH e o FTP. O *package*, chamado *Util*, permite-nos realizar as ligações SSH e FTP, tendo sido implementado, através da biblioteca *jschSSHChannel*.

Ainda na Fig.6.16, representamos o *package* chamado *models*. Este *package* permite guardar e carregar as informações dos *hosts*, e as respetivas *passwords* e *tokens*.

Em relação ao ficheiro *Execution Control*, este foi desenvolvido em *bash*, e corresponde a uma série de funções para a *shell* do Linux, as quais deverão ser adicionadas como alias.

Assim, no ficheiro *bash* desenvolvemos para cada plataforma de virtualização, um conjunto de funções, para realizar as operações de instalar, iniciar, pausar e parar um exercício CTF. Estas funções foram implementadas, para o *Kubernetes* e para o *Virtualbox*, através das ferramentas *kubectl* e *vboxmanage*.

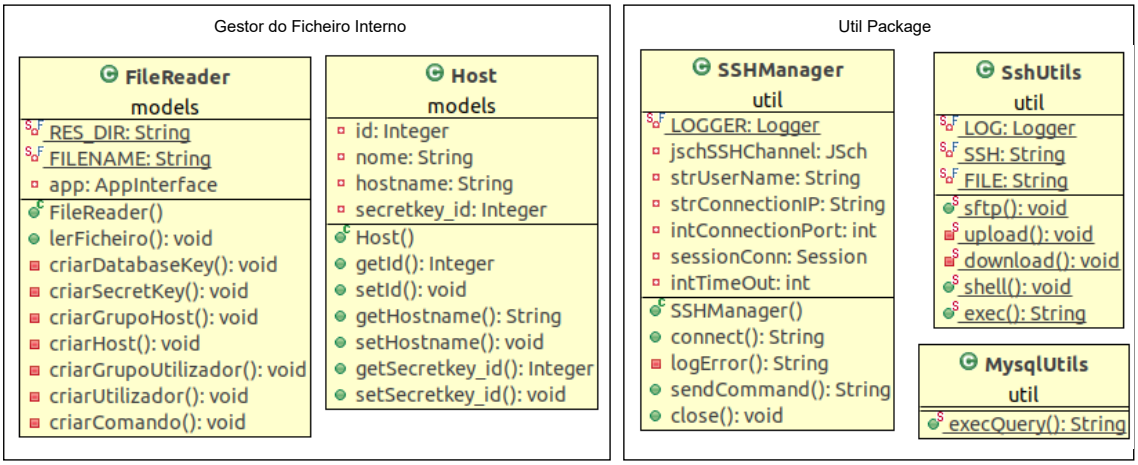

Figura 6.17: CTF Toolkit - *package Util* e *Models*

As restantes operações (*Static Analysis*, *Compile and Register* e *Synchonize and Notify*) são também funções implementas em *bash*. Porém, estas funções foram implementadas através da ferramenta que construímos em *Java*, o *exercice manager*.

6.7 CTFBOX

Na presente seção, descremos a implementação do sistema CTFBOX, que permitirá aos alunos gerirem a plataforma *CTF@DEI*.

Como já indicado na secção 5.3, o CTFBOX é na prática um sistema operativo, no qual acrescentamos novas ferramentas, que nos permitem aceder à plataforma CTF, como administrador. De acordo com a Fig.6.18, o CTFBOX foi dividido em quatro partes:

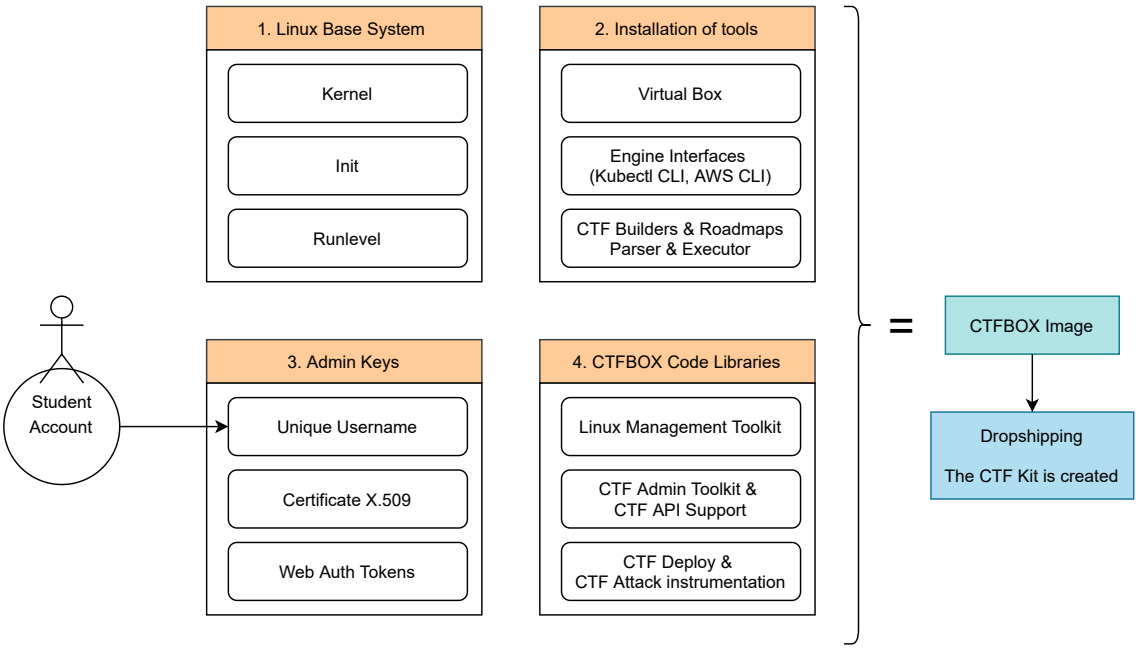

Figura 6.18: Diagrama de implementação do CTFBOX

- *Linux Base System* - optamos por utilizar *Kali* (distribuição *Linux* utilizada para em *Penetration Testing*, *Ethical Hacking* e *network security assessments*);
- *Installation of tools* - neste sistema instalamos as ferramentas necessárias (*Virtual Box* e Kubectl). Estas ferramentas são necessárias, para controlar as plataformas de virtualização;
- *Admin Keys* - para cada utilizador geramos um certificado *X.509* e vários *tokens*. Este certificado permite que o utilizador se autentique no servidor *web* e nas plataformas de virtualização. Os *tokens*, únicos por utilizador, permitem o acesso à API do *scoreboard*;
- *CTFBOX Code Libraries* - instalamos a *CTF Toolkit*, de forma, a que os alunos consigam gerir os exercícios na plataforma (pretendemos que, no futuro, os alunos desenvolvam outras ferramentas para as competições CTF).

A CTFBOX é simplesmente uma máquina virtual, mas permite que os alunos sejam introduzidos aos exercícios CTF, e tenham à sua disposição as ferramentas para o fazer. Esta máquina virtual poderá ser utilizada pelos alunos, durante as sessões de treino, os *workshops* e também durante as competições CTF.

Na secção seguinte, apresentamos os detalhes na preparação do primeiro *roadmap*, para os alunos do MSI.

6.8 Preparação de um roadmap

Apresentamos na presente secção, o procedimento realizado para preparar uma lista de exercícios CTF (*roadmap*) aos alunos do DEI, a fim de validar-mos a plataforma.

Para instalar os exercícios CTF, será necessário aceder à máquina CTFBOX, e executar os comandos indicados na Fig.6.19.

A lista de funções, poderá ser obtida ao escrever "ctf."e pressionar a tecla tab. Existem duas formas de instalar um exercício, através da função *ctf.install*, ou de forma equivalente, através das funções de cada componente da plataforma CTF@DEI. Cada um dos comandos retorna a sua documentação, se for executado sem parâmetros.

Selecionamos 5 exercícios da coleção CTF, da categoria *web*, estes foram instalados através dos comandos indicados anteriormente. Optamos pela categoria *web*, uma vez que, durante o primeiro semestre, os alunos focam-se em unidades curriculares ligadas ao desenvolvimento de aplicações. Os exercícios de demonstração, que selecionamos, foram os seguintes:

Tabela 6.1: Exercícios CTF de demonstração

name	description	solution
bypass	a simple html form, check the source code	add a new attribute to http request
cookie	with the new cookies, the system is safe	manipulation of cookie values
obfuscatedjs	This is a super secure portal with a really unusual HTML file	de-obfuscate javascript code

tornado	I learned to use templates with tornado. Are you going for an ice-cream?	template injection attack, tornado secure cookies
filelibrary	Welcome to my safe File Library! getFile?../	path traversal attack

Após a execução dos comandos indicados na Fig.6.19, as interfaces *ctf*, *ctfchat* (ver Fig.6.20), *ctfsonar* (ver Fig.6.5) e *ctfdocs* (ver Fig.6.3) foram atualizadas com os exercícios selecionados.

6.9 Síntese

Neste capítulo, apresentamos os detalhes de implementação da plataforma CTF@DEI.

De acordo com a arquitetura apresentada no capítulo 5, todos os componentes foram implementados e testados, com excepção do sistema de monitorização.

Por último, a plataforma encontra-se disponível na infraestrutura do DEI, na qual, disponibilizamos 5 exercícios, a fim de validar a plataforma com os alunos do DEI.

No capítulo seguinte, apresentamos os resultados obtidos.

```
guest@ctfbox:~$ ctf.
ctf.chat.add               ctf.vagrant.install
ctf.chat.remove            ctf.vagrant.reload
ctf.chat.send              ctf.vagrant.start
ctf.compile                ctf.vagrant.stop
ctf.deploy                 ctf.virtualbox.config
ctf.docs.buildHtml         ctf.virtualbox.import
ctf.docs.publish           ctf.virtualbox.install
ctf.docs.updateExercices   ctf.virtualbox.links
ctf.exercices.list         ctf.virtualbox.list
ctf.install                ctf.virtualbox.outros
ctf.install.disable        ctf.virtualbox.pause
ctf.install.enable         ctf.virtualbox.poweroff
ctf.reload                 ctf.virtualbox.remove
ctf.run                    ctf.virtualbox.restart
ctf.scan.analyse           ctf.virtualbox.resume
ctf.scorebord.install      ctf.virtualbox.savestate
ctf.vagrant.clean          ctf.virtualbox.start
ctf.vagrant.create         ctf.virtualbox.startVRDP

# opção 1: instalar o exercício com o ID 1
guest@ctfbox:~$ ctf.install 1

# opção 2: instalar o exercício com o ID 1
guest@ctfbox:~$ ctf.scorebord.install 1
guest@ctfbox:~$ ctf.chat.add 1
guest@ctfbox:~$ ctf.scan.analyse 1
guest@ctfbox:~$ ctf.docs.buildHtml && ctf.docs.publish
guest@ctfbox:~$ ctf.virtualbox.install 1 && ctf.virtualbox.start 1

guest@ctfbox:~$ ctf.chat.add

# Adiciona o exercício ao ctfchat
#
# Usage: ctf.chat.add <id>
#
# Example 1: ctf.chat.add 3
#
```

Figura 6.19: CTF Toolkit - Interface Bash

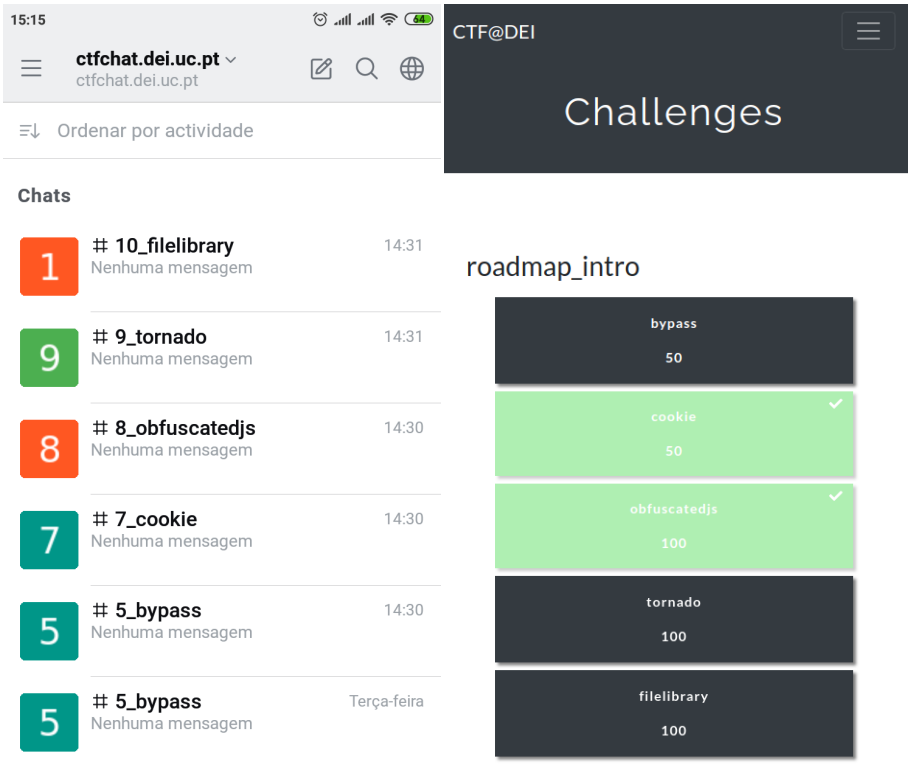

Figura 6.20: Interface ctfchat e ctf

Esta página foi intencionalmente deixada em branco.

Capítulo 7

Resultados

No capítulo anterior, descrevemos os detalhes da implementação da plataforma *CTF@DEI*. A plataforma foi validada, com uma amostra de 5 exercícios (ver 6.8), os quais foram resolvidos por 6 alunos do DEI.

Após a resolução dos exercícios, os alunos preencheram um questionário (ver apêndice 10).

Devido às restrições impostas pela *Covid-19*, a amostra foi reduzida, pelo que, no sentido de comparar os resultados práticos obtidos, com aqueles que são referidos na revisão de literatura, formulamos 4 hipóteses de estudo. As 4 hipóteses que quisemos testar foram:

- Na hipótese 1, pretende-se testar se os conhecimentos prévios exigidos, e os conhecimentos adquiridos nos cursos de segurança informática são adequados para a realização dos exercícios CTF;
- Na hipótese 2, pretende-se testar se a aprendizagem feita através dos exercícios de CTF é útil para o sucesso académico nos cursos superiores de segurança informática;
- Na hipótese 3, pretende-se averiguar se os exercícios de CTF estimularam a motivação e o interesse dos alunos para a área de segurança informática;
- Na hipótese 4, pretende-se aferir se a aprendizagem feita através dos exercícios de CTF se adequa ao mercado de trabalho atual na área da segurança informática.

O principal objetivo deste capítulo foi, mediante os resultados enunciados na revisão de literatura, e os resultados práticos obtidos pelos questionários e pela resolução dos exercícios CTF, verificar se as hipóteses formuladas seriam validadas, ou pelo contrário denegadas.

O segundo objetivo deste capitulo "Resultados"foi também, o de saber se os resultados práticos obtidos, iriam de encontro aos resultados apresentados na revisão de literatura, ou se estes últimos seriam completamente díspares dos primeiros.

Por último, incluímos os resultados da performance e da usabilidade da plataforma de exercícios CTF, desenhada e implementada nos capítulos anteriores, e validada pelos alunos do DEI.

Seguidamente, apresentamos os resultados para cada uma das hipóteses formuladas.

7.1 Hipótese 1 - Adequação do Know-how prévio e adquirido para a resolução dos exercícios CTF

Na e os conhecimentos prévios exigidos, e os conhecimen-tos adquiridos nos cursos de segurança informática são adequados para a realização dosexercícios CTF, pretende-se testar se os conhecimentos prévios exigidos, e os conhecimentos adquiridos nos cursos de segurança informática são adequados para a realização dos exercícios CTF.

Para testar esta hipótese, recorremos a várias artigos na área de *Cybersecurity Education* e *Cybersecurity challenges*. Consideramos várias universidades, nomeadamente: *Howard University*, *University of Massachusetts Boston*, *Amrita University*, *University of South Australia*, *Japan Advanced Institute of Science and Technology*, *Bournemouth University*, *Norwegian University of Science and Technology*, *Massachusetts Institute of Technology* e *Business and Technology University*.

Concluímos, com a análise dos 11 artigos, que 9 deles validam positivamente a **Hipótese 1**, e 2 deles invalidam a hipótese formulada.

Relativamente aos artigos que validam a hipótese 1:

- de acordo com [28] e [65], predomina nos cursos de segurança a área de conhecimento *Data Security*. Em [222] concluíram que nos exercícios CTF, também predomina a mesma área de conhecimento;
- em [127], consideram que os exercícios CTF são relevantes e complementares aos programas dos cursos em SI, e através do *Cyber Lab*, os exercícios CTF desempenham um papel ativo na educação e na investigação;
- em [41], concluí-se que podemos aproximar os alunos dos conhecimentos exigidos nas competições CTF, através da aplicação de metodologias *hands-on learning* aos cursos de SI;
- segundo [12], o uso da gamificação no curso de segurança contribuiu para atingir os pré-requisitos das competições CTF e o interesse dos alunos pelo segurança informática aumentou em 62%;
- em [204], os autores concluíram que o nível de dificuldade das principais competições CTF não é adequado para principiantes novatos;
- no artigo [169], os autores concluíram que foi possível alinhar com sucesso o plano curricular, para integrar os exercícios de segurança e lançar os alunos nas competições CTF;
- em [16], os investigadores abordaram os processos em *Cybersecurity Training* e concluíram que os requisitos necessários para os exercícios CTF, podem ser adquiridos através de um ambiente de treino dedicado;
- em [225], os investigadores concluíram que os exercícios CTF conjugados com o curso, permitiram aos alunos o desenvolvimento de competências e técnicas em segurança informática;
- em [257], os investigadores mencionaram que o número de *cyber ranges* e *testbeds* aumentou nos cursos de segurança, em vários domínios, como ferramenta de aprendizagem para atingir determinadas competências práticas;

Relativamente aos artigos que invalidam a hipótese 1:

- em [254], os exercícios CTF requereram um vasto conjunto de técnicas, e não foram projetados para equipas com pouca experiência prática. Os cursos de segurança ajudaram no sentido de obter estes requisitos, mas os autores tiveram de recorrer a sessões de treino e a *workshops* para preparar os alunos para a competição CTF;

- em [42], os autores mencionam que as principais competições CTF são destinadas a especialistas de SI e este nível de competências é demasiado alto para os alunos novatos dos cursos de SI;

De seguida, desenvolvemos os artigos que validam a hipótese 1.

No artigo [28], através de uma avaliação aos programas de SI nas melhores universidades, os investigadores perceberam que os cursos envolvem sobretudo competências práticas e estão alinhados com determinados *currículos*, em particular, o *IEEE-CS/ACM computing curricula* e o currículo *CSEC2017*. Concluíram que, os conhecimentos adquiridos com os cursos estão enquadrados em seis áreas principais, que são: *Data Security*, *Software Security*, *System Security*, *Human Security*, *Organizational Security* e *Societal Security*.

Segundo o relatório [65], cerca de 96 instituições de ensino basearam-se no currículo *CSEC2017*, sendo que *Data Security* é a área de conhecimento mais predominante nos cursos.

Com as mesmas áreas de conhecimentos definidas em *CSEC2017*, no artigo [224], os autores analisaram 71 artigos em *cybersecurity education* e concluíram que as áreas de conhecimento que prevalecem são: *Data Security*, *Software Security* e *System security*.

No artigo [222], os autores analisaram cerca de 15 mil *writeups* de exercícios CTF entre 2012 e 2020, e concluíram que as áreas de conhecimento que prevalecem são *Data Security* e *Connection security*, de acordo com a Fig.7.1.

Figura 7.1: Distribuição das áreas de conhecimento nos writeups CTF (Fonte: [222])

Na *Howard University*, o grupo *Cyber Security and Research Center* disponibiliza atividades extra-curriculares para ajudar os alunos a desenvolver as técnicas necessárias para as competições CTF. Este programa é formado por um grupo de alunos, e inclui um laboratório, que realiza várias competições CTF, atividades *hands-on*, *bootcamps* e palestras [127]. O grupo disponibiliza vários materiais de estudo, que servem como pré-requisitos para as competições CTF, disponíveis em [131].

O grupo indicou vários tópicos e conceitos para integrar novos participantes, em particular: noções com o sistema *Kali* e *Linux command line*, *encoding systems*, *log analysis*, *crytography*, *web exploitation*, *password cracking*, *scanning*, *enumeration*, *traffic analysis*, *reverse engineering* e *wireless security*. Estes conceitos, considerados relevantes para os exercícios CTF, foram apresentados e discutidos nas aulas teóricas dos cursos de segurança informática. Nem todas as ferramentas descritas pelo grupo (como *Hashcat*, *Kismet*, *Aircrack-ng* ou *Burp Suite*), são abordadas nos cursos de segurança, embora estas sejam importantes para os exercícios CTF. O grupo concluiu, que os exercícios CTF complementam as aulas teóricas dos cursos de segurança [130].

Em 2012, na *University of Massachusetts Boston* [41], os autores concluíram que a eficácia das competições CTF depende do interesse dos alunos. Porém, para os alunos que pretendem começar nas competições CTF, existe uma barreira de conhecimentos alta, que acaba por diminuir o interesse dos mesmos.

Face a esta adversidade, os investigadores recorrem a novas metodologias, baseadas em *workshops* e *hands-on learning*, que segundo os autores, tem um papel fundamental na motivação dos alunos. Concluem os autores, que é possível atingir os pré-requisitos das competições CTF e aproximar os alunos dos conhecimentos exigidos.

Em relação à pedagogia de gamificação, em 2014, *Banfield* e *Wilkerson* [12] realizaram um estudo qualitativo sobre a utilidade da mesma. Os autores comprovam que com a presente pedagogia, é possível aumentar a motivação intrínseca e a auto-eficácia dos alunos em exercícios relacionados com ataques informáticos.

A auto-eficácia corresponde à confiança pessoal do indivíduo (diretamente conectado com a motivação intrínseca), ao executar uma estratégia para atingir os objetivos designados.

Banfield e *Wilkerson* concluíram, que a auto-eficácia dos alunos aumentou, quando recorreram à pedagogia de gamificação. Na turma tradicional, 28% dos alunos mostraram interesse em comprometer os servidores, ao passo que, na turma do curso com gamificação, o mesmo interesse foi de 90%. Para atingir requisitos exigentes, os investigadores defendem que a auto-eficácia é essencial, e a gamificação aumenta ainda mais a auto-eficácia dos alunos [12].

(Na *Ted Talk "The puzzle of motivation"* [228], é mencionado que a motivação está conectada com a motivação intrínseca, em três aspetos importantes: autonomia, domínio e propósito. Autonomia, que corresponde ao impulso de dirigir as nossas vidas; Domínio, como o desejo de ficar cada vez melhor em algo que importa; e propósito, como o deseja de alcançar algo maior do que nós próprios.)

Em 2014, na *Amrita University* [204], os investigadores criaram uma *framework* no intuito de avaliarem e classificarem as competições CTF, a fim de identificarem fatores que dificultam a entrada de novos alunos nas competições CTF.

A *framework* mencionada permitiu a criação de um *ranking*, que ordena e compara as diferentes competições CTF. Esta classificação também permite que os professores e os alunos, identifiquem quais são as competições CTF, que se enquadram no curso de segurança.

A *framework* apresentada, que avalia as competições CTF, é composta por 10 requisitos, que são os seguintes:

1. Os desafios da competição são divididos por categorias e são mapeados no tipo de vulnerabilidade correspondente;
2. As tarefas são analisadas quanto à existência de solução ou não;

3. Nível de dificuldade do exercício;
4. A periodicidade do evento CTF (anual, semestral);
5. O treino prévio que é dado antes das competições CTF;
6. Os incentivos oferecidos, os prémios e os reconhecimentos;
7. O número total de equipas por evento;
8. As técnicas de segurança utilizadas (*scanning*, *patching*);
9. A localidade do evento;
10. O alcance do evento;

Segundo o terceiro requisito da *framework*, os investigadores avaliaram o grau de dificuldade das competições CTF mais conhecidas. Consideraram que, o nível de dificuldade dos exercícios das competições *DEFCON CTF*, *SECUINSIDE CTF*, *CSAW CTF* e *PHD CTF* é de 80%. Segundo os autores, este nível de dificuldade não é adequado para principiantes novatos, como consequência, tal facto pode gerar frustração, desinteresse pelos alunos e até desistência.

Por outro lado, existem diversas competições CTF, que se enquadram e são desenhadas em conformidade com os planos curriculares dos cursos de segurança informática, como é o caso do *PICO CTF* ou do *Google CTF Beginners Quest*, que são vocacionadas para um público mais inexperiente.

Em 2014, na *University of South Australia* (UniSA), os autores aplicaram novas abordagens, para melhorar os cursos universitários, com a particularidade, de integrarem os exercícios CTF no âmbito dos cursos. Esta abordagem consistiu fundamentalmente, em recorrer à *framework* NICE e à teoria denominada por *"Situational Crime Prevention"* [169].

A universidade procurou melhorar o curso de segurança sistematicamente. A cada iteração, selecionavam competências indicadas no documento da *framework* NICE. Deste processo, resultou uma lista de áreas de conhecimento, capazes de envolver os alunos nos exercícios CTF.

De acordo com a Fig.7.2, as áreas selecionadas encontram-se listadas à esquerda (coluna *NICE Competencies*), e nas colunas da direita encontram-se os aspetos da teoria de *"Situational Crime Prevention"*, que foram considerados como importantes para o curso.

Observando a Fig.7.2, os investigadores focaram-se em *"Increase the perceived effort"*, referindo a importância de capacitar os alunos para identificarem, reagirem e abordarem corretamente (e sobretudo) as medidas preventivas relacionadas com a segurança informática.

De acordo com as áreas e as competências, selecionadas pela universidade para o curso de segurança (*Information Systems/Network Security*, *Operating Systems*, *Vulnerabilities Assessment*, *Incident Management*, *Infrastructure Design* e *Encryption/Cryptography*), os autores identificaram as categorias dos exercícios CTF, que demonstraram os conceitos e tópicos abordados nos cursos de segurança.

Os exercícios CTF selecionados para este curso de segurança envolveram as seguintes categorias: *misconfiguration*, *privilege escalation*, *default configuration*, *cryptography in SSH*, *levels of access* e *vulnerabilities assessment.*

NICE Competencies (Number of KSAs addressed in the courses)	Situational Crime Prevention Theory				
	Increase the perceived effort	Increase the perceived risks	Reduce the rewards	Remove excuses	Reduce provocations
Information Systems/Network Security (7)	Yes				Yes
Assessment (1)	Yes				
Infrastructure Design (3)	Yes		Yes		
Operating Systems (7)	Yes		Yes		Yes
Encryption (1)	Yes		Yes		
Cryptography (2)	Yes		Yes		
Identity Management (1)	Yes				
Incident Management (4)	Yes	Yes			
Computer Languages (1)					
Configuration Management (2)	Yes				
Computer Network Defense (10)	Yes				Yes
Computer Forensics (1)		Yes			
Information Assurance (3)	Yes	Yes			
Vulnerabilities Assessment (5)	Yes				Yes
Knowledge Management (1)					
Criminal Law (1)				Yes	

Figura 7.2: Tópicos abordados no curso da Segurança da Universidade UniSA (segundo a Framework NICE e a prevenção dos crimes informáticos) (Fonte: [169])

Durante o curso de segurança, os alunos resolveram uma série de exercícios CTF a par com as aulas teóricas, e deste modo, os investigadores obtiveram as seguintes conclusões:

- os exercícios proporcionaram um ambiente didático, para praticar as técnicas aprendidas nos tutoriais e nas aulas teóricas;
- com os exercícios CTF, a diversidade de competências num grupo de alunos com *backgrounds* diferentes, enriqueceu a discussão e a partilha de ideias, o que fez com que os alunos investigassem novas estratégias ofensivas e defensivas por conta própria;
- os exercícios CTF foram uma excelente ferramenta para aprender os riscos de segurança envolvidos ao disponibilizar um serviço publicamente;
- em relação aos ataques realizados e às ocorrências inesperadas, os alunos estudaram novas vulnerabilidades e aprenderam novas abordagens para explorar os pontos fracos de um sistema.

Em 2016, em *Japan Advanced Institute of Science and Technology* [16], a fim de aumentar o nível de habilidades dos profissionais de segurança no Japão, os investigadores implementaram uma nova *framework*, que permitiu aos alunos de segurança informática treinarem as suas competências através dos exercícios CTF.

No âmbito deste tema (*Cybersecurity Education*), os investigadores argumentam que o processo de *cybersecurity training* pode ser divididas em três classes de requisitos principais, que são:

- *Individual skills* - são as técnicas isoladas, por exemplo: realizar um captura da rede ou uma análise de vulnerabilidades. Requer um estudo dos métodos e das ferramentas existentes;

- *Team skills* - são as habilidades necessárias, para que a equipa de segurança seja eficaz no seu todo. Requer aperfeiçoamento da cooperação, da comunicação e das competências sociais;
- *Computer Security Incident Response Team skills* - são as habilidades avançadas, que as equipas de segurança necessitam de desenvolver, para lidar com ambientes austeros e com situações imprevistas. Requer o desenvolvimento de capacidades de liderança, que permitem observar, organizar e decidir em grupo;

Os investigadores consideram que é de extrema importância que exista educação em segurança informática. Não apenas para os futuros profissionais da área, mas também para as pessoas comuns, que representam a maioria dos utilizadores na *Internet* (conceito denominado por *Cybersecurity Literacy Education* ou *Cybersecurity Awareness Training*).

Os autores prevêem que a gamificação irá desempenhar um papel importante na educação da segurança informática, sendo que irá contribuir para melhorar a aprendizagem e a motivação dos alunos.

Assim, e com o objetivo de os futuros engenheiros desenvolverem as técnicas necessárias, os investigadores implementaram a *framework* representada na Fig.7.3, que introduz os exercícios CTF nos cursos de segurança. A *framework* é composta pelos seguintes elementos:

- *Training Specification* - Através dos utilizadores e dados em histórico (cenários de treino, incidentes e vulnerabilidades catalogadas), são definidas as atividades de treino e os exercícios;
- *Content Definition* - é gerado o conteúdo de treino para o formato apropriado, por exemplo, para ser utilizado num LMS, como o *Moodle*;
- *Cyber Range Instantiation* - é gerado o *setup* chamado de *Cyber Range*, com as configurações de rede e os respetivos serviços.

Concluem os investigadores, através da framework implementada, que os requisitos necessários para a resolução dos exercícios CTF, só podem ser adquiridos através de um meio / ambiente que exponha os alunos às questões reais de segurança informática, e estas experiências podem ser tratadas em ambientes de treino controlados (*Cyber Ranges*, exercícios CTF, *testbeds*).

Em 2020, na *Bournemouth University*, os autores [225] mencionam que os exercícios CTF não foram desenhados para principiantes. Segundo os autores, existe uma falta de profissionais qualificados no sector da segurança informática, e os exercícios CTF podem melhorar as técnicas dos alunos e o interesse por esta área. Segundo os autores, as competições CTF são uma possível abordagem para melhorar este quadro e mitigar a falta de profissionais qualificados nesta área.

O estudo apresentado pelos investigadores pretende colmatar a discrepância e a falta de profissionais na área, através da identificação dos requisitos básicos para um aluno focado nas competições CTF.

Através de questionários, os autores concluíram que os exercícios CTF contribuem positivamente para os cursos de segurança informática, dado que, os alunos desenvolveram as suas competências e técnicas ao longo das competições CTF.

Para preparar os alunos para as competições CTF, o estudo realizado sugere que:

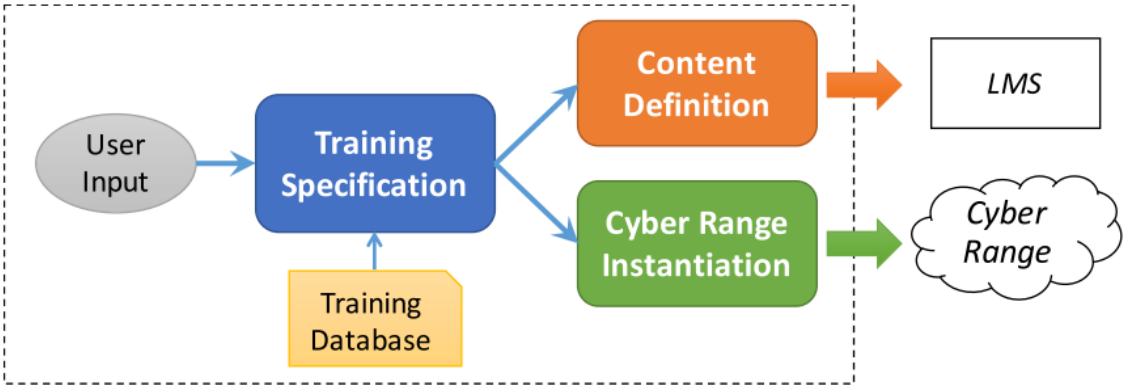

Figura 7.3: Cybersecurity training support framework (Fonte: [16])

- a variedade de exercícios CTF é um aspeto relevante, porque os alunos possuem interesses e *backgrounds* diferentes;
- o nível de dificuldade deverá ser incremental, para abranger o maior número de alunos;
- o suporte e a orientação aos alunos são aspetos importantes, principalmente para os alunos novatos [225].

Em 2020, na *Norwegian University of Science and Technology* [257], através da revisão de literatura apresentada, os autores observaram que o interesse pelos tópicos *cyber range* e *testbed* cresceu nos últimos anos.

Entre 2010 e 2017, o número de ambientes propostos especificamente para atividades educativas aumentou, estes enquadram-se nos pré-requisitos dos exercícios CTF, com experiências práticas em cenários de ataque e defesa, análises de vulnerabilidades e *Reverse engineering*. Na revisão da literatura, os autores mencionaram que os domínios das *cyber ranges* e das *testbeds* foram *Hybrid Network and Applications*, *Social Engineering*, IoT, *Cloud*, *Autonomous System*, *Network*, *Critical Infrastructure* e Supervisory Control And Data Acquisition (SCADA).

Desenvolvemos, agora, os artigos que invalidam a hipótese 1.

Em 2011, no *Massachusetts Institute of Technology* [254], os autores realizaram sessões de treino, *workshops* e uma competição CTF. Os investigadores mencionaram uma série de problemas pedagógicos com os exercícios CTF. Estes problemas são gerados, porque existe uma grande barreira para quem se inicia nas competições CTF. Os exercícios CTF requerem esforço, empenho e abordam um vasto conjunto de técnicas e conhecimentos em várias áreas (sistemas operativos, bases de dados, administração de sistemas, redes), que necessitam de conhecimentos prévios.

Segundo os investigadores, os exercícios CTF permitem que os alunos aprendam sobre uma ampla gama de tópicos e considerações nos sistemas informáticos, ao possibilitar novas experiências práticas sobre ataques e defesas, e também, ao possibilitar estudos e projetos de investigação, fora das competições.

Segundo os autores, os pré-requisitos podem ser atingidos através de sessões de treino e *workshops*, porém as competições CTF não foram projectadas para proteger as equipas e os alunos com pouca ou nenhuma experiência prática em segurança informática.

Em 2020, em *Business and Technology University* [42], os investigadores mencionaram que segurança informática é uma área difícil, porque requer competências em vários domínios (programação, base de dados, redes, arquitetura de sistemas e administração de sistemas). Nas competições mais exigentes, os alunos irão enfrentar especialistas com este nível de competências. Competências relevantes nestas áreas também permitirão aos alunos, fornecer soluções eficazes para as infraestruturas e para os sistemas de informação.

Para atingir estes pré-requisitos exigentes, os investigadores referem que é importante atingir determinados objetivos nos cursos de segurança informática, em particular:

- criar um currículo apropriado (composto por aspetos gerais e por técnicas práticas em SI);
- utilizar abordagens proativas no setor da educação;
- recorrer a um laboratório prático, que aplique os conceitos teóricos na prática (um laboratório em que os alunos se sintam motivados a progredir e a estudar para as competições CTF).

De acordo com os artigos analisados, concluímos que existe um esforço nas universidades, em criar mecanismos que aproximem os alunos aos pré-requisitos exigidos. Embora, os exercícios CTF sejam difíceis, por conjugarem várias unidades curriculares, a aprendizagem pode certamente ser simples, se utilizarmos as ferramentas e as metodologias apropriadas.

Questionamos seis alunos do DEI sobre esta hipótese, logo depois de terem experimentado a plataforma *CTF@DEI*, e obtivemos os resultados da Fig.7.4 e Fig.7.5.

Sobre os conhecimentos prévios exigidos para a frequência do MSI obtivemos uma média de 2.8. Já em relação, aos conhecimentos adquiridos durante o curso do MSI obtivemos uma média de 3.3.

Os resultados obtidos indicam que os alunos não tem conhecimentos suficientes pré-adquiridos para a resolução de exercícios. Em relação aos conhecimentos obtidos no curso, os resultados indicam que estes são razoavelmente suficientes.

A revisão da literatura apresentada referiu que os pré-requisitos necessários e os conhecimentos adquiridos no curso de segurança são importantes para os exercícios CTF. Os resultados obtidos dos questionários mencionam que os conhecimentos prévios são insuficientes e os adquiridos durante o curso são razoavelmente suficientes, para resolver os exercícios CTF. Esses resultados demonstram na prática, o que obtivemos da revisão da literatura, isto é, a importância dos pré-requisitos e dos conhecimentos adquiridos no curso para a resolução dos exercícios CTF.

7.2 Hipótese 2 - Aprendizagem através dos exercícios CTF

Hipótese 2 - A aprendizagem feita através dos exercícios de CTF é útil para o sucesso académico nos cursos superiores de segurança informática?

Para analisar esta questão recorremos novamente a várias artigos na área de *Cybersecurity Education* e *Cybersecurity Challenges*, a fim de analisamos os estudos realizados e os resultados já obtidos.

Consideramos várias universidades, como a universidade de *Birmingham*, *Appalachian State University*, a universidade da Catalunha, a universidade de *East Tennessee State*,

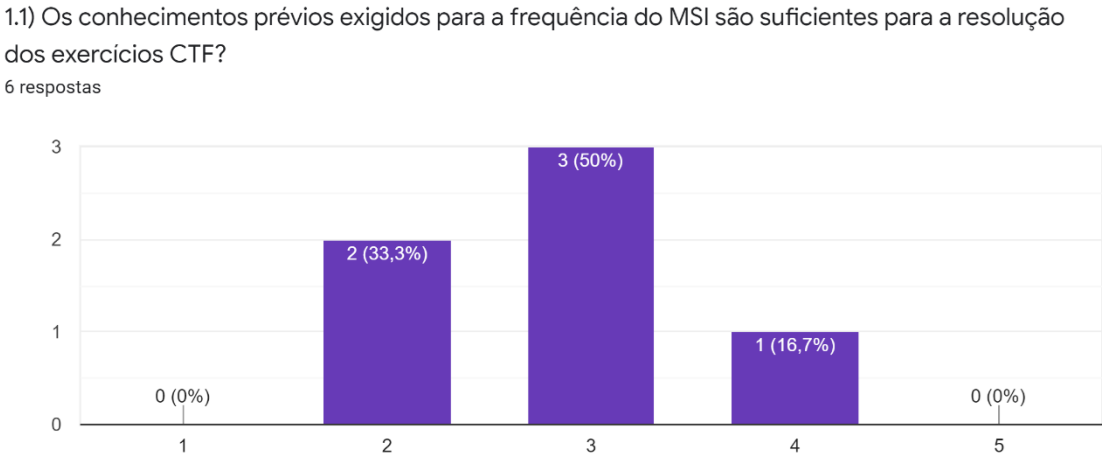

Figura 7.4: Questionário - Os conhecimentos prévios exigidos para a frequência do MSI são suficientes para a resolução dos exercícios CTF?

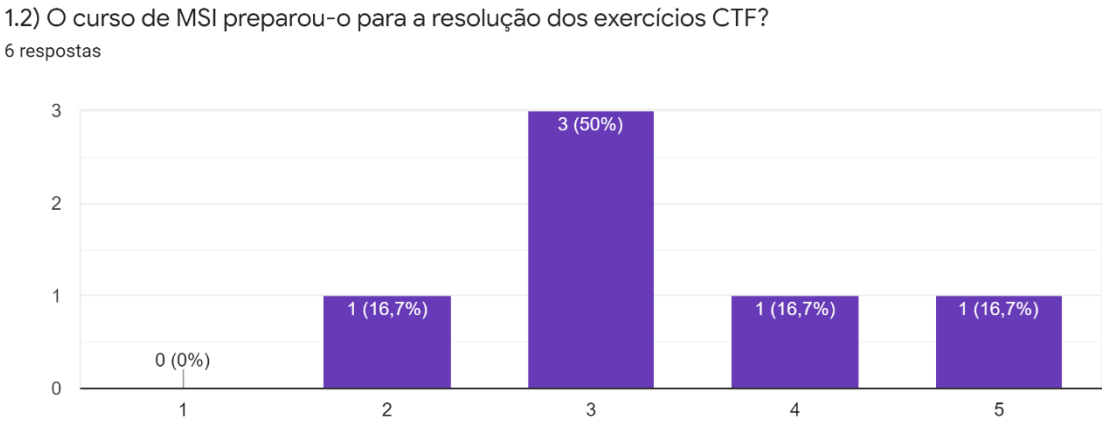

Figura 7.5: Questionário - O curso de MSI preparou-o para a resolução dos exercícios CTF?

a universidade de *Masaryk*, o *Institute of Information Security* em *Yokohama* e a *Spanish National Distance University*.

Concluímos, com a análise dos 11 artigos analisados, que 6 validam positivamente a **Hipótese 2**, 1 artigo invalida a hipótese, e que 4 artigos são inconclusivos (mencionam aspetos positivos e negativos).

Relativamente aos artigos que validam a hipótese 2:

- em [43], valida-se que é possível utilizar os exercícios CTF nos cursos;
- em [122], concluíram que a metodologia *"ethical hacking"* melhora os modelos de aprendizagem dos cursos;

- em [16], os investigadores apresentaram um conjunto de requisitos para integrar os exercícios CTF nos cursos de SI;
- em [102], é proposta uma nova plataforma para transpor aspetos e competências dos CTF para os cursos universitários;
- [6] e [124] validam os exercícios CTF como uma ferramenta eficaz, e o conceito de gamificação poderá ser benéfico no treino de competências;
- em [159], concluíram que os exercícios CTF são benéficos na aprendizagem dos alunos (os autores enumeram várias recomendações na gestão dos ambientes didáticos);
- em [251], conclui-se que os exercícios CTF são benéficos e possibilitam uma aprendizagem interativa com os alunos;
- em [235], concluíram que os exercícios CTF são uma ferramenta potencial e promissora em várias áreas (segurança, programação);
- em [182], os exercícios foram modificados dinamicamente, para elaborar provas de avaliação para os alunos (plataforma CyExec);

Relativamente aos artigos que invalidam a hipótese 2:

- em [162], os alunos do curso de segurança informática, tiveram dificuldades em completar os exercícios CTF;
- em [43], concluíram os autores, que os alunos consideram os exercícios CTF demasiado difíceis, e a utilização de *flags* não é adequada, para avaliar os conhecimentos dos alunos ou transmitir *feedback*;
- em [159], referem que o código de conduta das universidades, pode restringir algumas das atividades realizadas nos exercícios CTF;
- em [251], concluem os autores, que existem aspetos negativos a ter em consideração, resultantes da aplicação dos exercícios CTF aos cursos de segurança informática;
- em [182], concluem que os custos, a manutenção e o desenvolvimento de novos cenários, acabam por reduzir a eficácia dos exercícios CTF na aprendizagem;

De seguida, desenvolvemos os artigos que validam a hipótese 2.

Em 2015, na universidade de *Birmingham*, o trabalho de *Chothia* validou o uso dos desafios e das competições CTF nos cursos de segurança; e o seu uso na avaliação dos conteúdos de um curso superior [43]. Este estudo permitiu concluir que na/o:

- Avaliação dos alunos - a capacidade de um aluno obter as *flags* está diretamente ligada com as notas do aluno;
- Avaliação das habilidades básicas - a resolução dos exercícios através de *flags* avalia efetivamente as habilidades básicas e a compreensão dos tópicos de segurança;
- Plágio - a utilização de *flags* estáticas possibilita um plágio mais generalizado e pode não atender às espectativas;

- Compreensão dos conceitos - a aquisição de *flags* está pouco correlacionada com a compreensão profunda dos conceitos de segurança pelo aluno (recomenda-se o uso de *flags* conjugado com respostas detalhadas);
- *Feedback* - com os exercícios CTF, a satisfação com o curso melhorou, com feedback positivo dado pelos alunos;

Em 2015, na *Appalachian State University*, concluíram que os alunos precisavam de um conjunto de habilidades para os exercícios CTF, para conseguirem defender adequadamente os servidores dos ataques realizados por especialistas [122].

Face a esta adversidade, os investigadores concluíram que a aplicação da metodologia *"ethical hacking"* melhorou os modelos de aprendizagem dos cursos de segurança informática.

Em 2016, no artigo [16], os investigadores definiram um conjunto de requisitos, que as atividades de treino e os exercícios CTF devem atender, a fim de serem eficazes e úteis nos cursos de segurança. O conjunto de requisitos apresentado pelos autores, resume-se aos seguintes tópicos:

- os exercícios tem de ser adequados para o público-alvo, de acordo com o nível de conhecimento e as competências dos participantes;
- os exercícios tem de estar de acordo com as técnicas, que o curso pretende desenvolver;
- os exercícios deverão utilizar ferramentas e procedimentos práticos, para garantir, que os alunos podem lidar com um ataque informático real;
- os exercícios deverão ser abrangentes e diversificados, para atingir o maior número de candidatos;
- os exercícios deverão ter um custo e desempenho sustentável.

Na Universidade da Catalunha, em 2016, os autores do artigo [102] criaram uma plataforma de CTF, a ICT-FLAG, que oferece um novo design e funcionalidades inovadoras (ferramentas de *eLearning*, automatismos nos cenários CTF, *scoreboard*, sistemas de avaliação e métricas aplicadas à aprendizagem dos alunos). Esta plataforma foi desenhada para melhorar a aprendizagem dos alunos, e permite uma experiência baseada nas competições CTF e no conceito gamificação.

Segundo [6] e [124], em 2016, as experiências obtidas com os jogos CTF são uma ferramenta eficaz para aprender e treinar competências nos cursos de segurança informática. Os investigadores concluem que o conceito de gamificação aplicado à segurança, poderá ser uma opção benéfica para a consciencialização em segurança informática, e eficaz no treino de competências.

Na universidade de *East Tennessee State*, *Lehrfeld* e *Guest* [159], em 2016, interligaram os exercícios CTF com os cursos de segurança. Concluem os investigadores, que esta abordagem foi benéfica para a aprendizagem dos alunos, nos seguintes aspetos:

- Vetores de ataque - na identificação da superfície de ataque (vetores de ataque e suas vulnerabilidades) e respetivos métodos de ataque;
- *Exploiting* - na realização de atividades de *pentesting*, com oportunidades de executar ataques em aplicações e sistemas reais;

- Avaliação - na ponderação de estratégias, de como um determinado ataque pode ser evitado e mitigado;

Em relação aos componentes e outros aspetos dos exercícios CTF, os investigadores *Lehrfeld* e *Guest* afirmaram que:

- As plataformas e os exercícios CTF contribuíram para avaliar o progresso dos seus alunos e validar se os objetivos de aprendizagem estão a ser atingidos;
- O estabelecimento de uma base de habilidades é um aspeto importante para um curso de segurança informática, que pode ser atingindo com os exercícios CTF;
- Os novos exercícios e as suas modificações foram implementadas com base nas experiências anteriores e nos conhecimentos técnicos dos participantes;
- Os exercícios CTF são ajustáveis para vários perfis e níveis de dificuldade, por exemplo, são direcionados à segurança das aplicações (para programadores) ou direcionados aos aspetos técnicos dos sistemas (para administradores de sistemas).

Em 2020, na universidade de *Masaryk*, os investigadores *Vykopal* e *Švábenský* [251] concluem que os exercícios CTF são benéficos para os cursos de segurança, quando aplicados aos seguintes aspetos:

- Trabalhos de casa - identificaram que a maioria dos alunos prefere substituir os trabalhos de casa tradicionais, por exercícios CTF;
- Duração do CTF - os exercícios CTF com duração de várias semanas, no contexto das aulas, são menos stressantes do que as competições CTF, e fornecem mais oportunidades aos alunos (experimentam diferentes ataques, resoluções e estratégias de ataques e defesa);
- Aprendizagem interativa - os professores ensinam habilidades de uma forma interativa, através de demonstrações práticas e discussão de ideias.

Em 2020, na *Spanish National Distance University*, os investigadores integraram os exercícios de CTF numa disciplina de segurança informática. Concluíram que, as experiências com estes exercícios são promissoras. Os exercícios CTF também encorajaram os professores a expandir os exercícios para outras disciplinas, e afirmam que esta abordagem pode ser exportada para diferentes áreas relacionadas, para além da segurança informática [235].

Em 2021, no *Institute of Information Security* em *Yokohama*, *Nakata* e *Otsuka* apresentam o *CyExec* [182]. Os investigadores concluem que as plataformas serão cada vez mais eficazes, e que irão reproduzir fielmente os ambientes reais e incidentes de segurança.

Nakata e *Otsuka* mencionam que através do *CyExec* é possível alterar ligeiramente os exercícios, mantendo o objetivo de aprendizagem, o que possibilita a reutilização dos mesmos cenários em contextos de avaliação.

Desenvolvemos, agora, os artigos que invalidam e apresentam limitações à hipótese 2.

Em 2010, no projeto *Blunderdome*, os investigadores e professores recorreram aos exercícios CTF, para demonstrar a importância do conceito *active learning*. Os autores concluíram que os alunos tiveram dificuldades em completar os exercícios, pelo que foi necessário uma maior intervenção por parte dos professores [162].

Em 2015, na universidade de *Birmingham* [43], em relação aos exercícios selecionados pela universidade de *Birmingham*, os alunos mencionaram que estes são difíceis. O nível de dificuldade dos exercícios CTF é uma limitação, que exclui muitos alunos das competições CTF.

Ao contrário das respostas escritas e dos métodos tradicionais de avaliação, os investigadores analisaram a abordagem do uso de *flags* no contexto das aulas. A submissão de *flags* indica ao aluno que conseguiu chegar à solução, mas não fornece informação suficiente, sobre o trabalho realizado ou o nível de conhecimentos do aluno.

Assim, os autores concluem que a avaliação através de *flags* evitará que os alunos recebam um *feedback* personalizado sobre o trabalho realizado. Os questionários de fim de curso, realizados pelos alunos, também revelam que o *feedback* é altamente valorizado nos cursos.

Os investigadores identificaram que o uso de *flags* é facilmente partilhável, pelo que, os autores recomendam que as *flags* sejam acompanhadas com respostas escritas, onde os alunos possam demonstrar se compreenderam os conceitos e quais foram os métodos de trabalho utilizados.

Em 2016, os investigadores *Lehrfeld* e *Guest*, referem que expor os alunos a atividades de *"hacking"* pode ser uma experiência negativa para os cursos de segurança. Algumas instituições ensinam a executar ataques informáticos, outras recusam a prática destas atividades, tal facto, depende do código de conduta de cada instituição [159].

Em 2020, os investigadores *Vykopal* e *Švábenský* [251] mencionam alguns aspetos que limitam a implementação dos exercícios CTF, em particular:

- Avaliação dos cursos - os professores devem considerar cuidadosamente a plataforma, o formato do CTF, a sua duração e o método de pontuação (distribuição dos desafios e seus pontos), para poderem integrarem os exercícios CTF na avaliação dos cursos de segurança;
- Partilha de *flags* - um CTF com duração de várias semanas é mais vulnerável à partilha de respostas pelos alunos;
- *Feedback* - o formato dos exercícios CTF, em que a validação das respostas é um procedimento instantâneo, não fornece um *feedback* exato, nem permite identificar os alunos em risco. Para um maior acompanhamento, as respostas podem ser acompanhadas com os *writeups*, e a plataforma pode facultar análises avançadas sobre o progresso dos alunos;
- *Scoreboard* – o *scoreboard* pode não ser suficiente, porque existe uma grande probabilidade de a maioria dos alunos terminar quase todas as tarefas propostas;
- Dicas nas plataformas – as plataformas atuais funcionam como estruturas estáticas, que não se adequam ao desempenho e à experiência dos alunos. Por exemplo, as dicas oferecidas são genéricas e não se ajustam ao perfil do aluno;

Nakata e *Otsuka* mencionam, em 2021, que o custo de implementação, a manutenção da plataforma e o desenvolvimento de novos cenários, ou mesmo a partilha de conteúdos entre os alunos (partilha de respostas), acabam por reduzir a eficácia dos exercícios CTF na aprendizagem [182].

Em suma, verificamos que várias universidades optaram por integrar os exercícios CTF, tendo os investigadores mencionado as suas experiências e as lições aprendidas.

Identificamos como limitações, a grande dificuldade dos exercícios, as normas das universidades (que podem não permitir o uso de ataques), os custos com as plataformas CTF e com os laboratórios de segurança, o uso de *flags* (que foram consideradas inadequadas para avaliar e dar feedback aos alunos, além de possibilitar situações de plágio).

Questionamos seis alunos do DEI sobre a hipótese formulada, logo depois de terem experimentado a plataforma *CTF@DEI*, e obtivemos os resultados da Fig.7.6 e Fig.7.7.

Obtivemos ponderações altas (em média, acima de 4), sobre o facto de os exercícios CTF enriquecerem o curso do MSI e facilitarem a aprendizagem dos alunos nesta área.

Os resultados obtidos na prática, validam a hipótese formulada, o que vai de encontro à revisão de literatura, com a ressalva, de que os autores identificaram outros aspetos, para além dos exercícios CTF, que contribuiram para a aprendizagem.

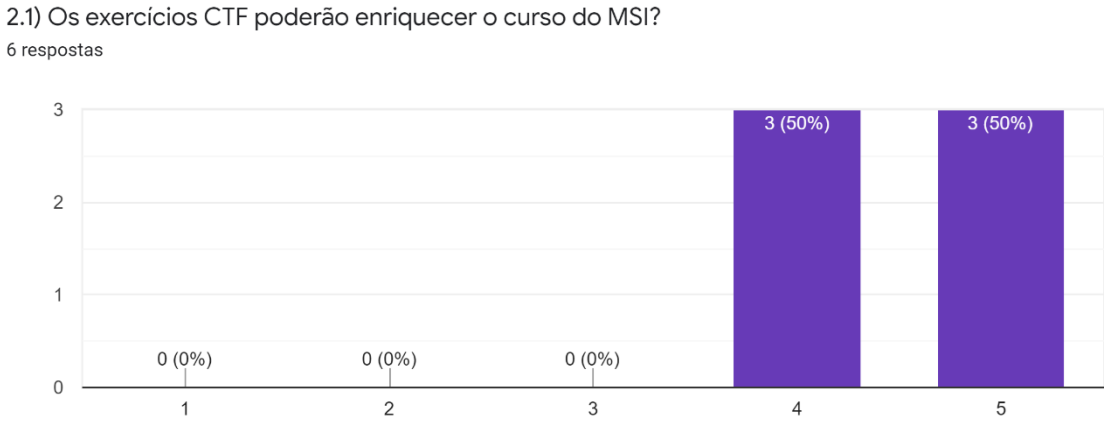

Figura 7.6: Questionário - Os exercícios CTF poderão enriquecer o curso do MSI?

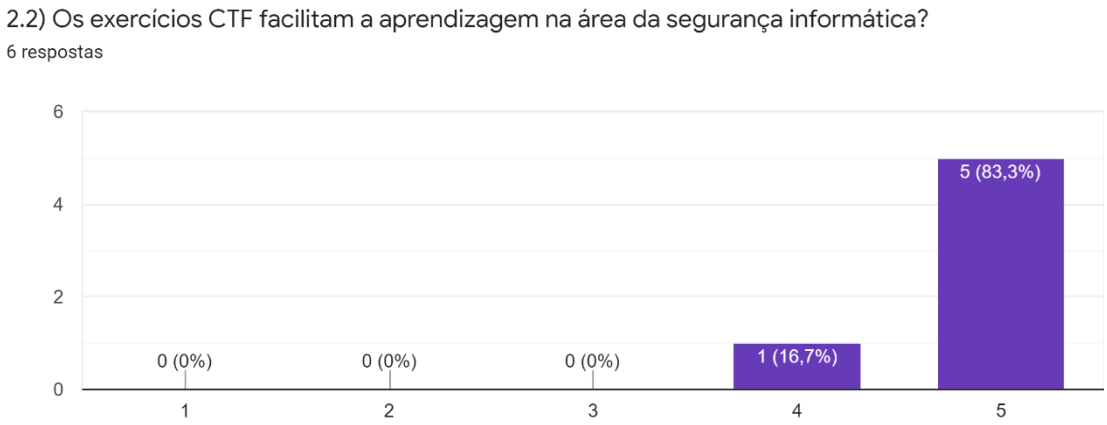

Figura 7.7: Questionário - Os exercícios CTF facilitam a aprendizagem na área da segurança informática?

7.3 Hipótese 3 - Motivação e interesse pela Segurança Informática

Hipótese 3 - Os exercícios de CTF estimularam a motivação e o interesse dos alunos para a área de segurança informática?

A fim de analisar esta questão, analisamos os resultados de vários artigos relacionados com o tema CTF, onde tivemos em conta as experiências realizadas e as conclusões reportadas com a realização dos exercícios e competições CTF.

Consideramos várias universidades, como a universidade de *Rhode Island*, *Eastern Michigan*, *Vienna*, *Leeds Beckett*, *Altai State*, *Towson*, *Leeds Beckett*, *Ionian*, *Tenaga Nasional*, *Georgia*, *West of England* e *Carnegie Mellon*.

Concluímos, com a análise de 12 artigos, que 11 validam positivamente a **Hipótese 3**, e 1 invalida a hipótese.

Relativamente aos artigos que validam a hipótese 3:

- em [253], os exercícios CTF são uma forma de introduzir experiências práticas motivantes para os alunos dos cursos de segurança;
- em [12], o autor conclui métodos centrados nos alunos, aumentam a autoeficácia e consequentemente a sua motivação;
- em [85], os exercícios CTF são considerados como uma oportunidade para envolver e motivar os alunos pela área da segurança informática;
- em [67], os alunos valorizaram positivamente e estavam motivados, tanto com os exercícios CTF, como com o laboratório de segurança;
- em [208], os professores recorreram a métodos de gamificação, para envolver e motivar os alunos nos cursos de segurança;
- em [166], a *CTF-based educational framework* criou um ambiente cooperativo e motivante entre os professores e os alunos, com um feedback positivo;
- em [187], recorreram aos exercícios CTF e a experiências de ataque e defesa, para encorajar o trabalho em equipa e motivar os alunos;
- em [209], o simulador, baseado nas competições CTF, encorajou e motivou os alunos a cooperar, para protegerem os servidores do laboratório;
- em [143], utilizaram vários aspetos dos eventos CTF, *escape rooms* e *puzzles*, para motivar os alunos nos cenários de segurança informática;
- em [19], os investigadores concluíram que, depois da competição CTF realizada, que o interesse e a motivação pela segurança informática aumentou em 15.6%;
- em [158], através de uma ambiente remoto, os exercícios CTF foram utilizados para proporcionar experiências didácticas práticas, o que motivou os alunos;

Relativamente aos artigos que invalidam a hipótese 3:

- em [31], os investigadores identificaram que a pressão exercida pelas equipas profissionais desmotiva os alunos dos cursos de segurança informática;

De seguida, desenvolvemos os artigos que validam a hipótese 3.

Em 2013, na universidade de *Rhode Island*, a plataforma concebida e os cenários práticos baseados nos exercícios CTF, permitiram ao autor introduzir abordagens práticas nos cursos de segurança. Concluiu o autor em [253], que é uma forma eficaz de motivar os alunos, para questões relacionadas com a defesa de redes e sistemas informáticos.

Em 2014, na universidade de *Eastern Michigan*, no artigo [12], concluiu-se que a criação de objetivos de aprendizagem combinados com o conceito de gamificação, aumenta a motivação intrínseca e a autoeficácia dos alunos. Os investigadores consideraram que as aulas deveriam ser centradas nos alunos, e nas metas para atingir a motivação intrínseca (satisfação em realizar as tarefas), em contraste com a motivação extrínseca (motivados por uma avaliação ou por um emprego).

Nos testes de [12], o mesmo questionário foi aplicado a duas turmas diferentes, no total de 96 alunos. Na primeira turma, os professores utilizaram os métodos tradicionais, baseados em palestras. Na segunda turma, os professores aplicaram os conceitos *Experiential learning theory* (processo centrado no aluno, que é baseado na experiência adquirida na resolução dos exercícios e no processo de aprendizagem prática), *Motivation and self-efficacy* e Gamificação.

Utilizando os métodos tradicionais, os autores obtiveram, nos questionários de motivação intrínseca e extrínseca, valores muitos baixos, aproximadamente 30%, em oposição, aos 92% obtidos utilizando os métodos *Experiential learning theory*.

Em 2014, no artigo [85], o autor menciona que é cada vez mais difícil chegar aos alunos mais jovens, e motivar os mesmos para continuarem a estudar, em particular, na área da segurança informática. Face a este problema, o autor concluí que os exercícios CTF são uma forma de gerar envolvimento e interesse nos alunos.

Em 2015, na universidade de *Vienna*, os autores [67] disponibilizaram exercícios CTF durante o curso de segurança informática. Através de questionários realizados aos alunos concluíram que:

- a competição contínua, a pontuação e os conselhos dos professores motivaram os alunos a resolverem os exercícios CTF;
- quando a experiência do curso se torna semelhante a um jogo competitivo, existe mais esforço e dedicação por parte dos alunos;
- a grande maioria dos alunos reconheceu a utilidade dos exercícios CTF e recomenda o curso a outros alunos;
- em alguns casos, os alunos não conseguiram investir mais tempo nos exercícios CTF, apesar de quererem tê-lo feito;
- os alunos que participam regularmente em competições, valorizam a importância dos exercícios CTF disponibilizados no laboratório do curso de segurança;
- o sistema é incapaz de avaliar exercícios parcialmente concluídos, mesmo que a abordagem do aluno funcione, o que pode ser um problema;

Em 2016, na universidade de *Leeds Beckett*, os autores em [208] recorreram à gamificação para ensinar os tópicos de segurança. Através de um esquema de pontos de experiência ("pontos XP"), a abordagem permitiu marcar os exercícios importantes, dar feedback

aos alunos e realizar uma *"avaliação gamificada"* (semelhante aos jogos competitivos). Concluíram os investigadores, que a estratégia é eficaz e que envolve os alunos nos cursos de segurança.

Em 2016, na universidade de *Altai State* (Barnaul, Russia), no artigo de [166], o autor apresenta uma *framework* baseada nos exercícios CTF para o ensino. Isto permitirá aos alunos adquirir o conhecimento, as técnicas e experiência em segurança informática. A *CTF-based educational framework* testada entre 2014 e 2016 teve sucesso, com um feedback positivo e vários benefícios pedagógicos, tendo o autor concluído que:

- a grande vantagem da *framework* proposta é ser centrada nos alunos e permitir que estes adquiram os conhecimentos e as técnicas necessárias;
- o interesse e o número de participantes ativos nas sessões de treino CTF aumentou. Os alunos mais experientes prestaram assistência e *coaching* aos outros alunos;
- a maioria dos alunos mencionou que o aspeto mais importante foi a oportunidade de aprender novas técnicas e conhecimentos;

Em 2017, na universidade de *Towson* [187], o projeto apresentado foi desenhado para incluir exercícios de ataque e defesa (baseados nas competições CTF), aplicáveis aos cursos de segurança informática. Para melhorar a motivação e o interesse dos alunos, os investigadores consideraram as seguintes técnicas:

- os alunos foram encorajados a trabalhar em equipa e a integrarem aspetos teóricos, com exercícios práticos;
- os ambientes disponibilizados aproximam-se dos sistemas reais e foram testados por profissionais da área;
- os alunos foram encorajados a desenvolver aspetos de comunicação, nos relatórios e na apresentação dos seus resultados;

Em 2018, na universidade de *Leeds Beckett* [209], os alunos dos cursos de segurança interagiram com o programa *Hackerbot*, um sistema que simula um atacante malicioso e fornece vários exercícios de segurança, baseados nas competições CTF. Os resultados apresentados foram encorajadores e motivaram os alunos. Com este sistema, o curso tornou-se apelativo e prático, e os alunos cooperaram em equipas para protegerem os servidores dos ataques realizados pelo sistema. As experiências realizadas neste simulador, aproximaram os alunos das certificações e do mercado de trabalho.

Em 2020, na universidade de *Ionian*, o estudo consistiu na implementação de uma nova abordagem no ensino de competências de segurança e privacidade. A abordagem envolveu aspetos das plataformas CTF, *escape rooms*, *puzzles/interactive books* e *Alternate Reality Games* (ARGs). Os investigadores concluíram que estas abordagens reforçam a aprendizagem dos alunos e motivam-nos a estudar os conceitos ligados à segurança informática [143].

Em 2020, na universidade de *Tenaga Nasional* [19], os investigadores recorreram a questionários para validar a importância das competições CTF nas instituições de ensino. O evento *Cyberhunt* 2019, envolveu 32 alunos, dos 13 aos 17 anos. Ao analisarem os questionários, antes e depois da competição realizada, os investigadores concluíram que o interesse e a motivação pela segurança informática aumentaram em 15.6%. Os resultados apresentados pelos investigadores foram os seguintes (escala de *Likert* com 5 pontos):

- os alunos avaliaram as sessões de treino em 4,25 (aulas práticas de preparação);
- os alunos avaliaram a competição CTF *Cyberhunt* 2019 em 4,19;
- em relação ao conhecimento adquirido durante a competição CTF, os alunos deram uma avaliação de 4,25.

Em 2021, na universidade de *West of England* [158], os investigadores combinaram o conceito de CTF, com os tópicos IoT e *Industrial Control Systems*. Face às implicações da pandemia *Covid-19*, o cenário prático foi disponibilizado em ambiente remoto, e os alunos puderam observar os dispositivos, através da plataforma *Microsoft Teams*.

O ambiente prático serviu como oportunidade de divulgação em outras escolas, onde foram realizados *workshops* com os alunos. Os exercícios CTF aumentaram a motivação dos alunos, tendo sido obtido um feedback positivo [158].

Desenvolvemos os artigos que invalidam a hipótese 3.

Em 2015, na universidade de *Carnegie Mellon*, no artigo [31], os autores concluem que os exercícios CTF do tipo ataque e defesa são desmotivantes para os alunos, porque estes sentem-se pressionados pelas equipas profissionais de segurança, que adotam ataques complexos.

Em suma, verificamos através das experiências recolhidas, que existem vários casos de sucesso, onde efetivamente a abordagem de recorrer aos exercícios CTF permitiu envolver e motivar os alunos, aumentando o interesse da comunidade para o estudo da segurança informática.

Questionamos seis alunos do DEI sobre a hipótese formulada, logo depois de terem experimentado a plataforma *CTF@DEI*, e obtivemos os resultados da Fig.7.8 e Fig.7.9.

Na primeira questão, sobre se os exercícios CTF aumentaram a motivação para a área de SI obtivemos uma média de 4.33. Na segunda questão, acerca dos exercícios CTF poderem contribuir para a melhoria do desempenho académico, a média obtida foi 4.

Os resultados obtidos referem que os exercícios CTF aumentaram consideravelmente a motivação dos alunos, e que pode melhorar o desempenho académico dos alunos. Estes resultados vão ao encontro da revisão de literatura apresentada, o que consequentemente valida, mais uma vez, a hipótese formulada.

7.4 Hipótese 4 - Mercado de trabalho

Hipótese 4 - A aprendizagem feita através dos exercícios de CTF adequa-se ao mercado de trabalho atual na área da segurança informática?

Para analisar esta questão, recolhemos a vários artigos que relacionam os exercícios CTF com a indústria na área da segurança.

Consideramos várias universidades, como a universidade de *Memphis* e a universidade de *Tampa*, e publicações, da ISACA e da Center for Strategic and International Studies (CSIS).

Concluímos, com a análise dos 14 artigos, que 12 validam positivamente a hipótese e 2 invalidam a hipótese.

Relativamente aos artigos que validam a hipótese 4:

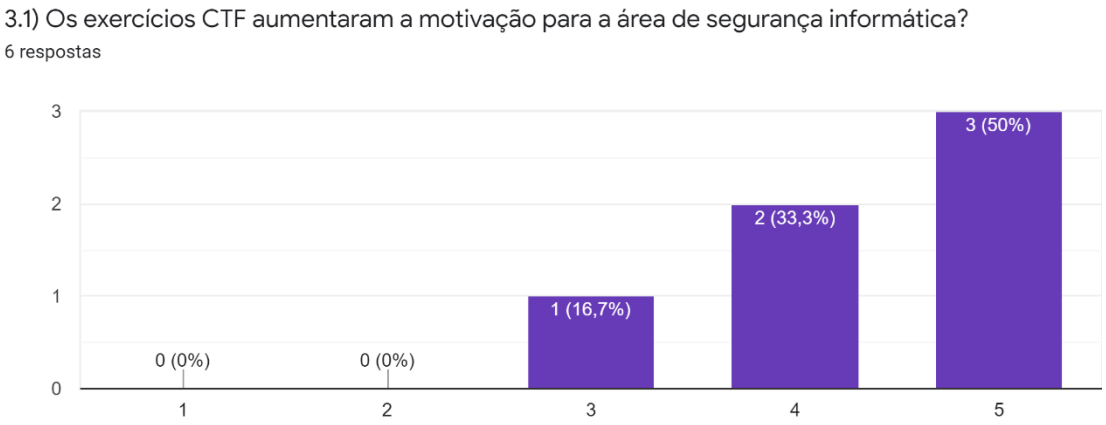

Figura 7.8: Questionário - Os exercícios CTF aumentaram a motivação para a área de segurança informática?

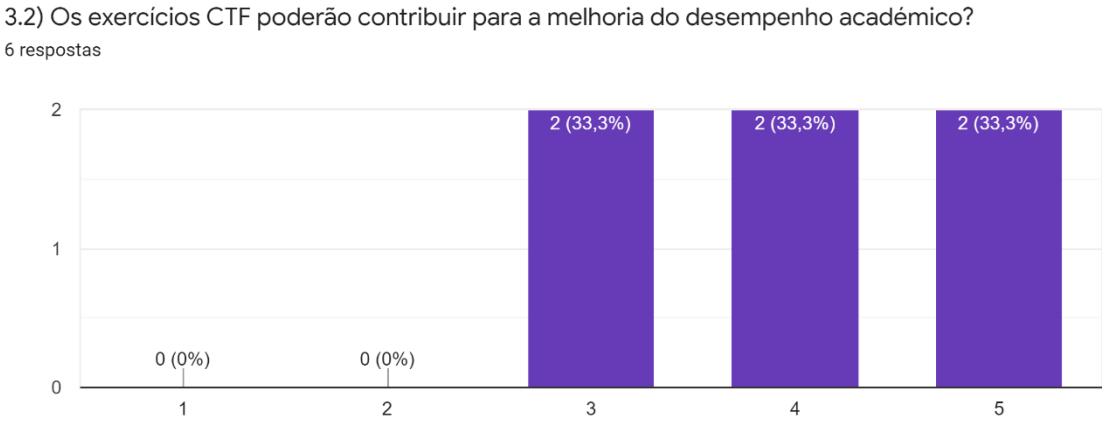

Figura 7.9: Questionário - Os exercícios CTF poderão contribuir para a melhoria do desempenho académico?

- em [163] (2010) e [57] (2021), os autores mencionam a importância dos exercícios CTF para as organizações, com oportunidades de recrutamento e avaliação dos candidatos aos empregos de SI;
- em [89], foi mencionado que 77% dos empregadores em tecnologias de informação acreditam que os programas de educação não preparam os alunos para a indústria de segurança informática;
- em [68], foram adoptadas novas abordagens para incentivar os alunos a resolverem problemas de segurança, baseados em situações reportadas pelas organizações;
- em [120], mencionam que as organizações consistentemente valorizam as experiências práticas e as certificações;

- em [85], as próprias organizações realizam formações e competições CTF, para captar novos candidatos e promover o ensino na SI;
- em [133], a *IBM Security* recorreu a ambientes didáticos para oferecer formação e preparar melhor os alunos para os cargos de SI;

  em [6], os autores recorreram a jogos CTF para envolver os alunos em novas experiências práticas motivantes;
- em [147], os autores estudaram as certificações profissionais, requeridas no mercado de trabalho de segurança informática, e melhoraram o laboratório didático de segurança da universidade;
- em [88], os autores propõem vários requisitos relevantes para desenhar exercícios CTF relevantes para a indústria de SI;
- em [136], o estudo reforça a necessidade de especialistas e identificou os fatores mais importantes no recrutamento de profissionais de SI;

  em [73], o autor refere que a maioria das universidades não fornecem garantias, de que os seus diplomados sejam qualificados para a indústria de SI;
- em [42], menciona que a procura e o interesse por programas de segurança informática aumentou, e os alunos procuram realizar certificações e estágios na área;

Relativamente aos artigos que invalidam a hipótese 4:

- em [44], enumerou vários fatores, que limitam a relação de eventos CTF nas organizações;
- em [225], entrevistaram vários participantes das competições CTF e apuraram as principais causas, que limitam o uso de exercícios CTF nas organizações;

De seguida, desenvolvemos os artigos que validam a hipótese 4.

Em 2010, em [163], *Lyne* mencionou que os exercícios CTF ajudaram as organizações a captar talentos e a desenvolver o interesse pela segurança informática, e que o Reino Unido procurará definir a próxima geração de profissionais em segurança, através das competições CTF. Do mesmo modo em 2021, em [57], *Seymour*, aluno de doutoramento na universidade de *Oxford*, mencionou que os exercícios de CTF servem como oportunidades de recrutamento para as organizações.

Em 2010, *Evans* e *Reeder* em [89], no âmbito do CSIS, apresentaram o panorama da segurança informática à época e definiram novas estratégias e soluções.

Os autores mencionaram que não existiam profissionais suficientes, para operar os sistemas em produção. Para atingir a meta de profissionais qualificados necessários para o mercado de trabalho, os autores propuseram quatro elementos principais:

- promover e financiar o desenvolvimento de programas adequados nas universidades;
- suportar o desenvolvimento e a adoção de técnicas rigorosas em certificações profissionais;
- combinar os processo de contratação, com recurso a exercícios de treino, para aumentar o nível de competências técnicas;

- assegurar um plano de carreira, como acontece na engenharia civil ou na medicina. Recompensar e reter aqueles que procuram qualificações de alto nível.

Os autores mencionaram que estavam a seguir uma estratégia nacional, semelhante ao esforço realizado na educação nos anos 1950, para promover a ciência e matemática.

Em 2013, na universidade de *Memphis*, os autores implementaram laboratórios didáticos para fornecer experiências desafiadoras, semelhantes aos desafios de segurança, que as empresas enfrentam atualmente [68].

Os autores introduziram uma nova abordagem, chamada *"Puzzle-based Learnning"* no ensino da segurança informática. Esta abordagem consiste em utilizar vários elementos como diagramas, grafos e jogos para desenvolver a capacidade de resolver problemas (reflectir, imitar e experimentar).

A abordagem permitiu que os alunos estudassem as várias classes de sistemas propostos, com diferentes componentes de *software* e *hardware*, ataques complexos, respetivas medidas de segurança e, casos mais avançados, como as Advanced persistent threat (APT).

Concluíram os autores que, com bases nas experiências descritas, que o treino nesta área é essencial para preparar adequadamente os alunos para o mercado de trabalho.

Em 2013, foi realizado pelo CSIS um relatório sobre a segurança informática a nível mundial [120]. A primeira oferta de profissionais em segurança informática vem dos métodos tradicionais de ensino. Porém, os autores mencionaram que existiam formas mais eficientes para preparar os profissionais.

A Fig.7.10 foi criada com as seguintes métricas: o investimento realizado em programas do ensino superior nas áreas de ciência, tecnologia, engenharia e matemática (STEM); os currículos técnicos em segurança informática no ensino superior; e a performance e o reconhecimento internacional nos exercícios CTF.

Os resultados obtidos mencionam que os melhores resultados ocorreram nos Estados Unidos e no Reino Unido. Estes países apresentaram bons indicadores, tanto na qualidade de ensino em segurança informática, como nos programas de treino e nos exercícios CTF ocorridos.

Os autores tecem várias recomendações, como por exemplo: redefinir os perfis para os empregos em SI; começar a aceitar candidatos provenientes do ensino não académico (isto é, não é obrigatório ter um curso universitário). Os autores mencionam também, que o ensino de competências em tecnologias da informação e SI, deverá começar nas camadas mais jovens, antes da entrada destes nas universidades, através de experiências e atividades práticas (*hands-on experiences and training*, *labs* ou *classroom exercises*).

Em 2014, no artigo [85], o autor menciona que os exercícios CTF são utilizados nas organizações, para captar novos profissionais e para melhorar os currículos dos especialistas em segurança informática.

Nos eventos de CTF predomina alguma emoção em torno dos desafios, este ambiente é propício à apresentação de novos candidatos às empresas e é uma forma de os profissionais de segurança mostrarem as suas habilidades.

Em 2016, em [133], *IBM Security* lança duas *cyber ranges* (*o X-Force Command Cyber Range* e o *X-Force Command Cyber Tactical Operations Center*), que são dois simuladores dedicados aos alunos de segurança informática.

Estes ambientes ensinam determinados aspetos técnicos aos participantes, em particular,

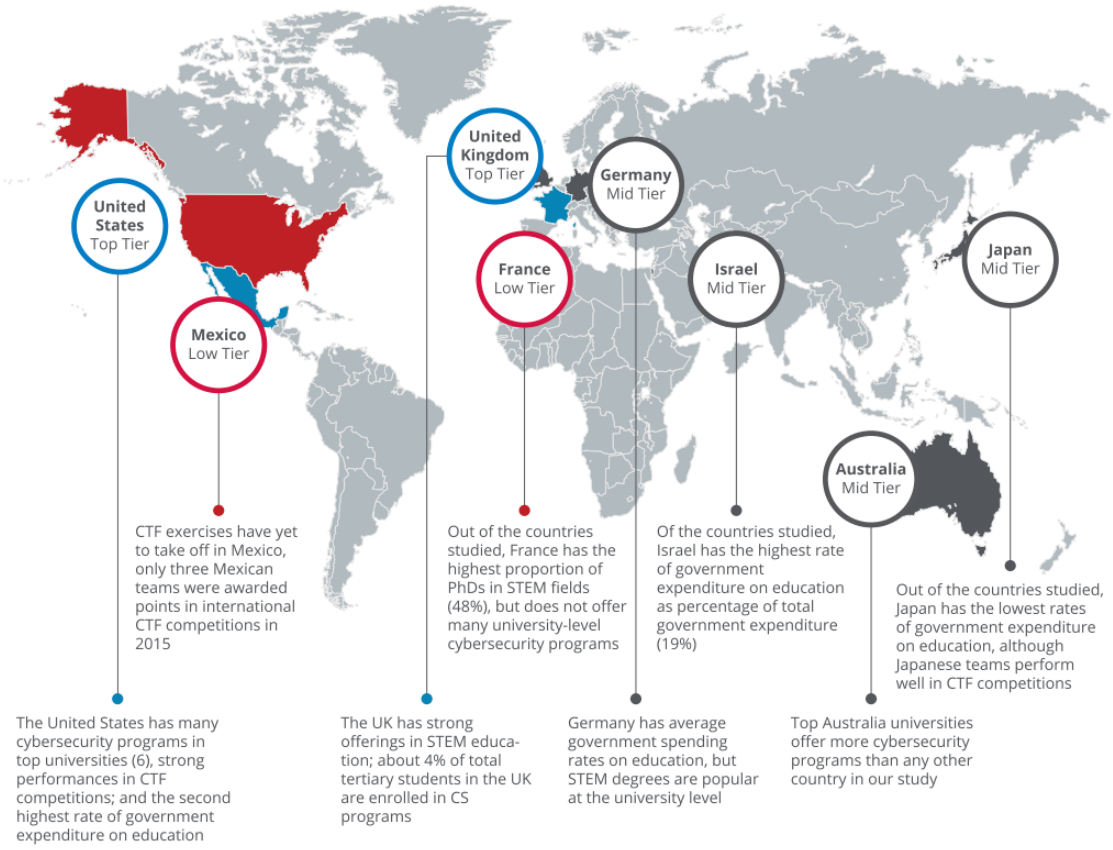

Figura 7.10: Classificação da educação em segurança informática por país (2013) (Fonte: [120])

nas áreas de administração de sistemas e controlo de operações. Através de simulações, os instrutores exemplificam diversos ataques informáticos e ensinam a projectar mecanismos de defesa apropriados.

Embora estes dois ambientes não incluam exercícios CTF, as *Cyber Range* promovidas pela *IBM Security*, oferecem formação e experiências práticas relevantes, e como tal, preparam melhor os alunos para os cargos de SI. Estes casos práticos ajudam a preparar os alunos, tanto para as competições CTF, como para o mercado de trabalho.

Em 2016, segundo [6], as experiências obtidas com os jogos CTF são uma ferramenta eficaz para aprender e treinar competências nos cursos de segurança informática.

Os investigadores concluíram que o conceito de gamificação, aplicado à segurança informática, foi uma opção eficaz no treino de competências em SI. O mesmo conceito foi também benéfico para a consciencialização dos alunos em segurança informática, aspeto indicado como relevantes para as organizações.

Em 2017, na Universidade de *Tampa* (EUA), os autores apresentam uma abordagem diferente para manter os requisitos dos programas de segurança atualizados. O método aplicado pelos autores, consiste na análise de diversas certificações profissionais prestigiadas na área da segurança informática, com a finalidade de extrair os tópicos relevantes para os cursos de segurança das universidades [147].

As aulas do curso foram planeadas para preparar os alunos para as certificações profis-

sionais, através de um laboratório de exercícios (certificado pelo norma ISO 27001), que oferece aos alunos várias experiências práticas, semelhantes às competições CTF [243].

A Fig.7.11 menciona cinco fatores, que os cursos de segurança podem herdar das certificações profissionais, nomeadamente:

- ***Threat Landscape*** - corresponde aos vetores, às ferramentas e às técnicas de ataque. Este fator relaciona-se particularmente com as competições CTF, porque existem diversas competições focadas na exploração de sistemas e em atividades ofensivas;
- ***Changing Technology*** - carateriza a evolução da tecnologia em si. Por exemplo, IoT e *Cloud Computing* são dois paradigmas recentes, que tem grande impacto no ramo da segurança informática;
- ***Workforce Needs*** - consiste nas pesquisas regulares para compreender as tendências e as mudanças que afetam a área de segurança. O *feedback* dos investigadores e as avaliações das associações, como ISACA e CompTIA, são aspetos decisivos, que justificam determinadas certificações, cursos e competições CTF;
- ***Industry Standards*** - correspondem às diretrizes e aos padrões da indústria, como *ISO standards*, NIST *security frameworks* ou PCI-DSS;
- ***Government, Regulation*** - corresponde às leis e regulamentos, que causam impacto significativo na segurança e privacidade. Por exemplo, HIPAA ou SOX.

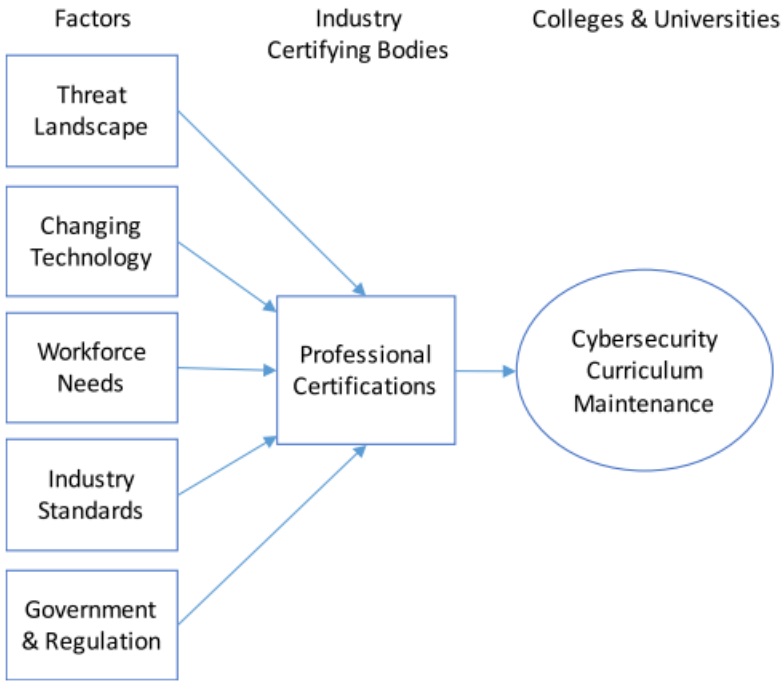

Figura 7.11: Fatores que causam impacto nas certificações e nos cursos (Fonte: [147])

Nas competições CTF estão presentes os fatores *Threat Landscape*, *Changing Technology* e *Workforce Needs*, porém os tópicos *Industry Standards* e *Government/Regulation* não são tipicamente abordados nestas competições. Questões centradas nos *standards* da indústria, nas leis ou nos regulamentos em vigor, deverão fazer parte dos programas de segurança.

Embora a indústria esteja a fazer progressos para lidar com os desafios da segurança informática, os autores concluíram que existe o risco de os cursos de segurança não se adaptarem às mudanças da indústria. Para lidar com esta adaptação, os autores mencionam que é necessário preparar os alunos para uma área altamente competitiva, as certificações e as competições CTF são úteis para incorporar as mudanças nos currículos dos programas dos cursos.

Em 2019, no artigo [88] (*Siemens AG*), os investigadores mencionaram que, não só as empresas precisam de integrar aspetos de segurança nas suas estruturas, como também precisam de treinar os engenheiros a conceberem produtos seguros. Os autores determinaram vários CDR relevantes, que foram aplicados em exercícios CTF, a par com a indústria de SI.

Estes requisitos tem com finalidade melhorar os artefactos dos exercícios CTF, para que seja possível providenciar um treino de consciencialização de qualidade (*awareness training*) no contexto das organizações e no desenvolvimento de *software*.

Referem os autores, que novas metodologias, como os exercícios CTF, providenciam uma maior retenção da informação, proporcionam atividades satisfatórias, as quais, estão em conformidade com as diretrizes / políticas de segurança atuais. Assim, uma metodologia eficaz, com exercícios CTF corretamente desenhados, será eficaz no treino de competências para as organizações.

Em 2020, no relatório da *ISACA* sobre os esforços aplicados realizados no setor da SI, concluiu-se que os profissionais nas empresas são escassos e que sobraram vagas por preencher. A Fig.7.12 menciona a importância de cada um dos factores, para determinar se um candidato à área de SI é qualificado.

A experiência prática é o tópico mais importante, para determinar que um candidato é qualificado. Porém, os resultados da ISACA mencionam que a maior lacuna para as organizações entrevistas reside em *soft skills*, seguindo-se lacunas de conhecimento e técnicas (particularmente em redes, infraestruturas e operações).

Em 2020, segundo [73], o autor identificou vários desafios: as organizações apresentam falta de profissionais; conseguir que mais alunos sigam a área da segurança; e readaptar os currículos dos cursos para que considerem, também, técnicas ofensivas.

Assim, o autor refere que é necessário criar uma nova *framework*, baseada na *NICE Cybersecurity Workforce Framework*, cujo foco seja o ensino de cenários ofensivos - para preencher a lacuna identificada no artigo.

Em 2020, na universidade da *Georgia*, a procura por disciplinas relacionadas com a segurança informática aumentou. Os alunos da universidade procuram por programas de segurança com certificação (como a *Offensive Security Certified Professional*), e por estágios na área. Sendo que, vários alunos trabalham em cargos de segurança, paralelamente com os estudos universitários [42].

De seguida, desenvolvemos os artigos que invalidam a hipótese 4.

Em 2014, no *Polytechnic School of Engineering NYU*, *Chung* e *Cohen* indicaram que é difícil, às universidades e empresas, organizar e competir nas condições dos eventos CTF [44].

Segundo os autores, os benefícios dos exercícios CTF podem ser limitados face aos problemas que originam, em particular os seguintes:

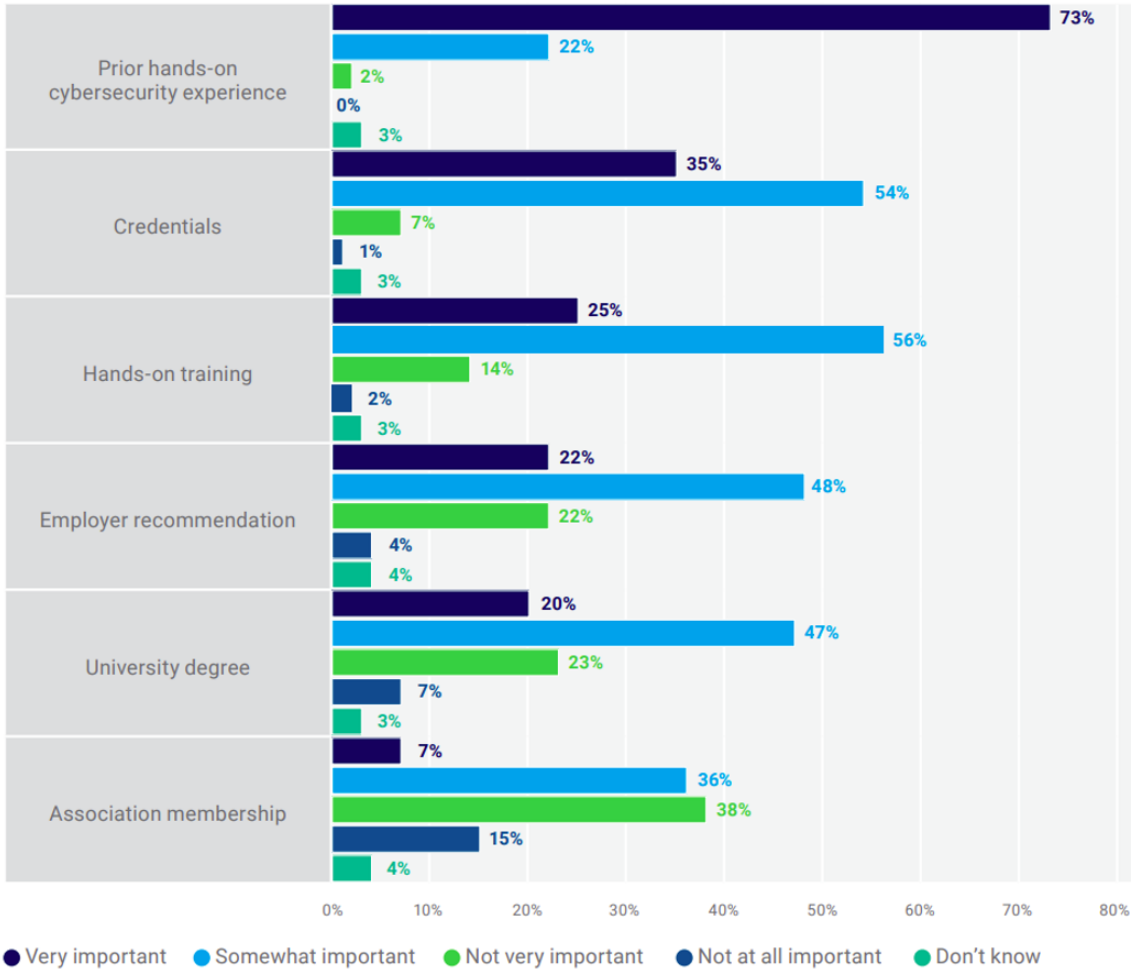

Figura 7.12: Factores que determinam a qualificação de um candidato (2020) (Fonte: [136])

- a natureza dos exercícios CTF não permite que novos alunos se integrem rapidamente nas competições. Os participantes que não conseguem resolver os exercícios, sentem-se desencorajados e irão parar de competir;
- as qualidades que definem, se os exercícios CTF são solucionáveis ou não, não estão presentes na maioria das competições (por exemplo, enunciados pouco descritivos, pré-requisitos elevados, comunicação pouco neutra);
- Os exercícios CTF são projetados para colocar os participantes fora da zona de conforto, com temas avançados (*file format reversing*, *byte code decompilation*, *machine code disassemblers*);
- o sucesso dos exercícios CTF está relacionado com o seu *design*. Os exercícios CTF mais difíceis podem ser relevantes para os participantes, mas aumentam o risco de ocorrerem desistências;
- a competição de CTF deverá garantir um certo nível de qualidade (quality assurance), onde os organizadores investigam os exercícios CTF e testam as várias soluções (no capítulo 2, apresentamos o ciclo de desenvolvimento de um exercício CTF [250], onde mencionamos a fase de simulação);
- a plataforma utilizada tem impacto nos exercícios CTF. Alguns problemas frequentes são: tentativas de *brute-force* às respostas dos exercícios (*flags*); o *website* fica

indisponível ou instável durante a competição;

- os participantes procuram aceder às contas das outras equipas, a informações privadas ou tentam ganhar pontos arbitrariamente.

Em 2020, na universidade de *Bournemouth* [225] foi realizada uma competição CTF e, posteriormente, foram realizadas entrevistas aos alunos.

Segundo os autores têm sido realizadas contribuições relevantes em *cyber security education*, porém existem lacunas mencionadas pelos autores, que podem limitar a presença as empresas das competições CTF.

Através das entrevistas realizadas aos alunos, os investigadores apuraram os seguintes fatores relevantes:

- os alunos mencionaram que os exercícios CTF exigem um trabalho árduo de preparação;
- os alunos consideraram que possuem os conhecimentos necessários para os exercícios CTF, mas apenas a nível teórico, e mencionaram que não sabem como os pôr em prática;
- os materiais disponibilizados ajudaram os alunos e permitiram integrar os iniciantes nas competições;

Em suma, recolhemos vários artigos que comprovam a hipótese formulada, embora existem certamente limitações, que devem ser consideradas no design dos exercícios CTF.

Questionamos seis alunos do DEI sobre a hipótese formulada, logo depois de terem experimentado a plataforma *CTF@DEI*, e obtivemos os resultados da Fig.7.13 e Fig.7.14.

Na primeira questão: a aprendizagem através dos exercícios CTF facilitará a entrada no mercado de trabalho na área da segurança informática, obtivemos uma média de 3,83. Na segunda questão: a aprendizagem obtida com os exercícios CTF melhorará o desempenho profissional na área da segurança informática, a média obtida foi 4,16.

Segundo os resultados obtidos, os exercícios CTF facilitam, consideravelmente, a entrada no mercado de trabalho e o desempenho profissional. Obtivemos resultados em concordância com a revisão de literatura apresentada, concluindo-se que, a hipótese formulada foi validada.

7.5 Performance e usabilidade da plataforma

Na presente seção temos como objetivo realizar os testes de performance e validar a usabilidade da plataforma CTF@DEI.

Na subseção 7.5.1 apresentamos os resultados de performance obtidos através da ferramenta *Kboom*(apresentada no capítulo 2), e através dos questionários realizados com os alunos do DEI.

Na subseção 7.5.2 apresentamos os resultados obtidos, relativamente à usabilidade da plataforma *CTF@DEI*, através dos questionários realizados com os alunos do DEI.

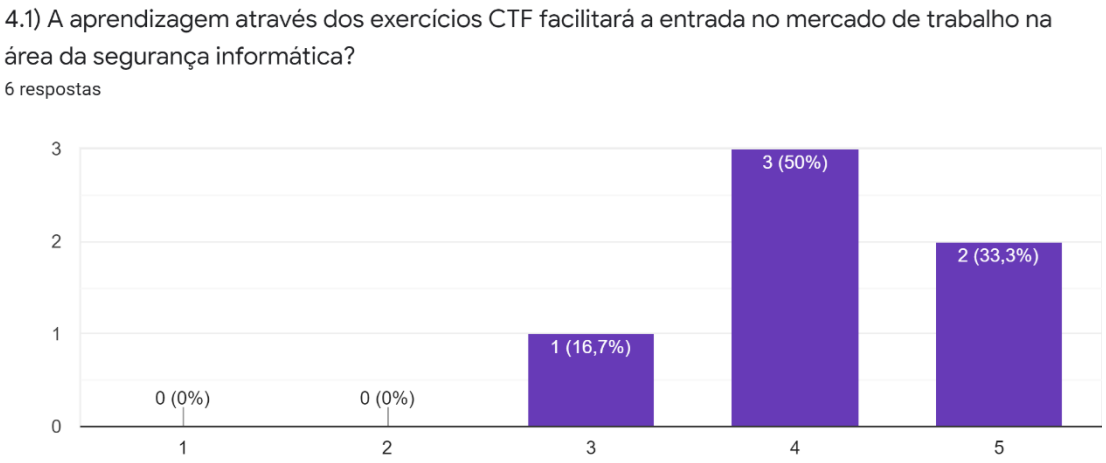

Figura 7.13: Questionário - A aprendizagem através dos exercícios CTF facilitará a entrada no mercado de trabalho na área da segurança informática?

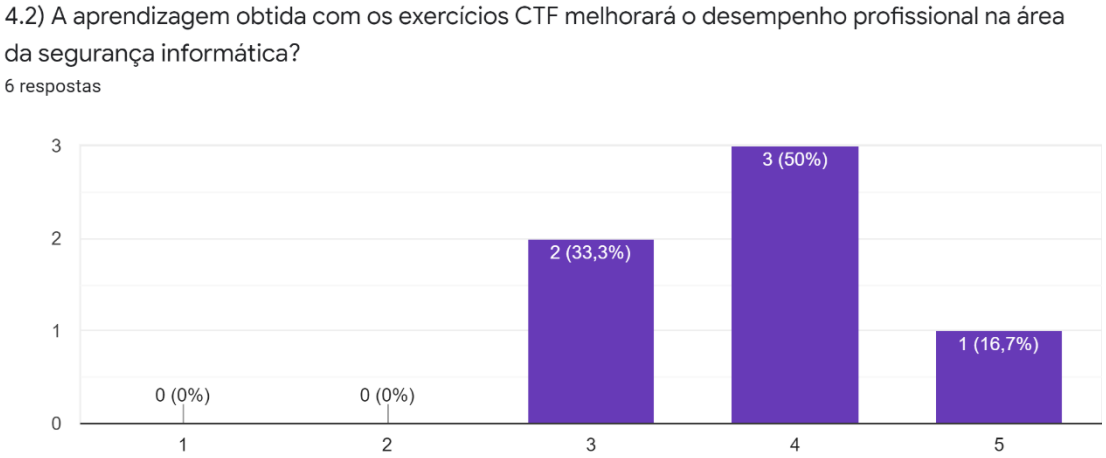

Figura 7.14: Questionário - A aprendizagem obtida com os exercícios CTF melhorará o desempenho profissional na área da segurança informática?

7.5.1 Performance

A ferramenta *Kboom* foi utilizada neste projeto, para realizar testes de escala ao *Kubernetes*, a fim de determinar o limite do *cluster* e quanto tempo demora a atingir essa capacidade.

A Fig.7.15, apresenta o código que foi executado para realizar o teste de escala, de acordo com as seguintes condições:

- o teste instância um novo *pod*, de 14 em 14 segundos;

- os *pods* criados são *BusyBox* (*"software suite that provides several Unix utilities in a single executable file"* [26]);
- se o *pod* não estiver em execução em 14 segundos, então não será contabilizado no *Overall* (falha);

```
# Script - scale test using kboom

curl https://raw.githubusercontent.com/mhausenblas/kboom/master/kboom -o kubectl-kboom \
    && chmod +x ./kubectl-kboom \
    && sudo mv ./kubectl-kboom /usr/local/bin


kubectl create ns kboom

curl -LO https://raw.githubusercontent.com/mhausenblas/kboom/master/permissions.yaml \
    && kubectl apply -f permissions.yaml

# irá criar um total de 1000 pods
# Considera-se falha, se o pod não estiver ativo após 14 segundos (timeout)
# one-by-one, o programa só cria um novo pod, depois de instanciar o anterior
kubectl kboom generate --mode=scale:14 --load=pods:1000

# kubectl kboom results
watch kubectl kboom results
```

Figura 7.15: Kboom - Bash script para testar o Kubernetes

Optamos por definir dois testes, com configurações diferentes. O 1º teste foi realizado, com duas máquinas virtuais, com as seguintes especificações:

- 1x máquina virtual, como *control plane*, com *Ubuntu 20.04.1 LTS*, com 4vCPU's (4 sockets with 1 core per socket), 6Gb de RAM e 50Gb de disco (SSD *flash*) - na plataforma de virtualização XCP-ng;
- 1x máquina virtual, como *workers nodes*, com *Ubuntu 20.04.1 LTS*, com 8vCPU's (8 sockets with 1 core per socket), 12Gb de RAM e 50Gb de disco (SSD *flash*) - na plataforma de virtualização XCP-ng;

O 2º teste foi realizado, com três máquinas virtuais, com as seguintes especificações:

- 1x máquina virtual, como *control plane*, com *Ubuntu 20.04.1 LTS*, com 4vCPU's (4 sockets with 1 core per socket), 6Gb de RAM e 50Gb de disco (SSD *flash*) - na plataforma de virtualização XCP-ng;
- 2x máquina virtual, como *workers nodes*, com *Ubuntu 20.04.1 LTS*, com 8vCPU's (8 sockets with 1 core per socket), 12Gb de RAM e 50Gb de disco (SSD *flash*) - na plataforma de virtualização XCP-ng;

Durante a execução do teste, garantimos que o *cluster* está no estado inicial, isto é, momento após a instalação, e portanto, não possui nenhum exercício CTF em execução. Os testes definidos foram executados duas vezes.

No 1º teste obtivemos um resultado de, aproximadamente, 100 *pods* criados em noventa minutos. No 2º teste obtivemos um resultado de, aproximadamente, 200 *pods* criados em noventa minutos. Os resultados obtidos encontram-se descritos na Tabela 7.1.

Tabela 7.1: Resultados obtidos com o script do Kboom

Test	Overall [0 : 1000]	Runtime	Fastest pod	Slowest pod	p50 pods	p95 pods
1º Test	103	1h32m	1h22m	1h31m	1h26m	1h31m
1º Test repeat	103	1h33m	1h22m	1h32m	1h27m	1h31m
2º Test	200	1h29m	1h11m	1h29m	1h20m	1h28m
2º Test repeat	210	1h30m	1h12m	1h29m	1h21m	1h28m

Notamos que, para o 1º teste, a partir de um certo limite, neste caso, 103 pods, os pedidos de criação de *pods* permanecem no estado *pending*. O *cluster* garantiu o funcionamento destes 103 *pods*, e constatamos através da Fig.7.16, que durante o 1º teste, o *worker node* alocou gradualmente a memória RAM, até atingir um certo limite.

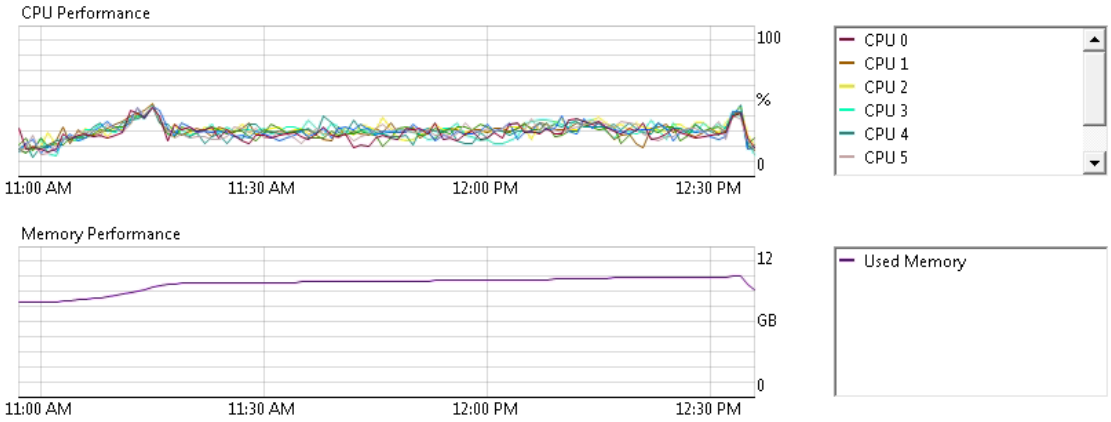

Figura 7.16: Gráficos de performance no teste 2 - CPU e Memória RAM

Questionamos seis alunos do DEI sobre o desempenho, logo depois de terem experimentado a plataforma *CTF@DEI*, e obtivemos uma média de 3,83, de acordo com os resultados da Fig.7.17.

7.5.2 Usabilidade

Para testar a usabilidade do sistema, recorremos a questionários com os alunos do DEI.

Questionamos seis alunos do DEI sobre a usabilidade do sistema, após terem experimentado a plataforma *CTF@DEI*, e obtivemos também uma média de 3,83, de acordo com os resultados da Fig.7.18.

7.6 Síntese

Neste capítulo, enunciamos os resultados da revisão de literatura e os resultados obtidos pelos alunos do DEI, através dos exercícios CTF e dos questionários disponibilizados.

Comparamos os resultados práticos com a revisão da literatura.

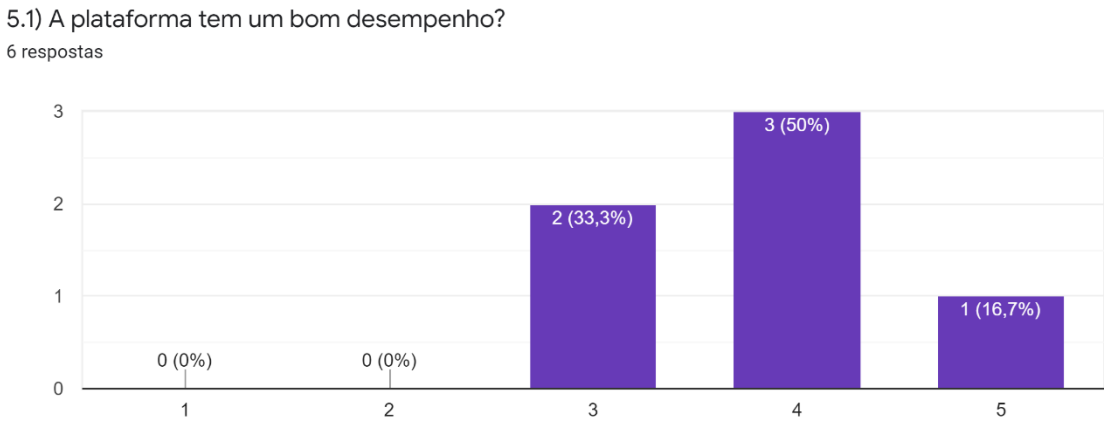

Figura 7.17: Questionário - A plataforma tem um bom desempenho?

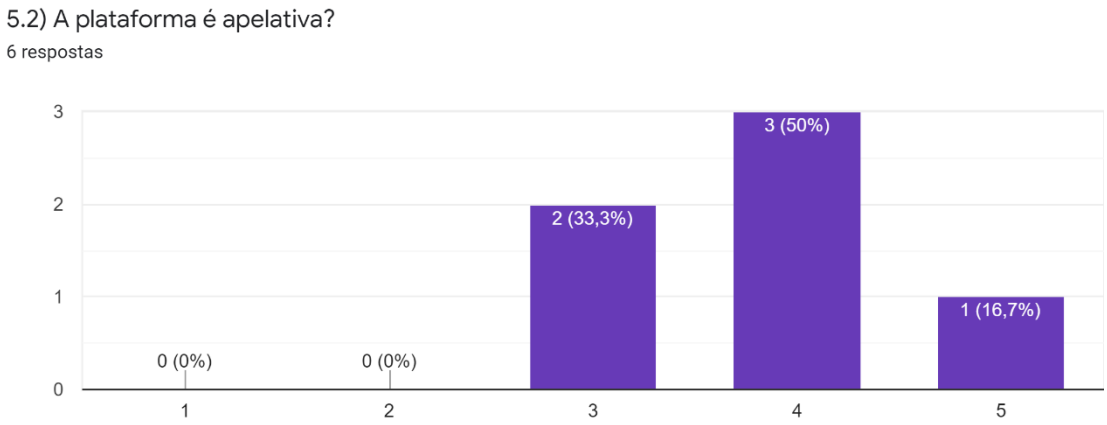

Figura 7.18: Questionário - A plataforma é apelativa?

Validamos todas as hipóteses formuladas, quer através dos resultados apresentados na literatura referida, quer através dos resultados obtidos pelos alunos do DEI.

Por último, foi avaliada positivamente a performance e a usabilidade da plataforma *CTF@DEI* implementada.

Seguidamente, passamos à discussão dos resultados obtidos.

Esta página foi intencionalmente deixada em branco.

Capítulo 8

Discussão dos resultados

No capítulo anterior, enunciamos os resultados obtidos para as hipóteses formuladas, e constatamos que todas as hipóteses foram validadas.

No presente capítulo "Discussão dos resultados"temos como objetivo tecer considerações sobre os resultados obtidos, tendo em atenção as duas perspectivas consideradas, a revisão de literatura e os resultados práticos obtidos pelos questionários.

Os resultados obtidos permitem afirmar que os conhecimentos prévios que os alunos do 1º ano de um mestrado em segurança informática são determinantes para a resolução de exercícios CTF.

Os conhecimentos adquiridos durante esse mesmo mestrado, segundo os resultados, são outro fator indispensável ao sucesso nos exercícios CTF.

Constatamos também, que a incorporação de aulas práticas com exercícios CTF, no desenho de um mestrado em segurança informática, para além de, motivar e estimular o interesse dos alunos pela área de segurança, permite também, apetrecha-los para o mercado de trabalho cada vez mais exigente e complexo nesta área.

No sentido da resolução desta problemática: preparar alunos para os cursos de mestrado em segurança informática, e posteriormente para o mercado de trabalho nessa mesma área, apresentamos, de seguida possíveis soluções.

A preocupação, por parte das instituições universitárias, em considerar nos programas dos cursos de Licenciatura na área da Informática, a componente segurança informática, sobretudo na sua vertente prática (utilizando metodologias de *active learning*), permitirá dotar os futuros alunos de mestrado em segurança informática com os conhecimentos necessários para o seu sucesso escolar.

Destacamos, o *Mossé Cyber Security Institute* [180] na Austrália, que disponibiliza cursos e certificações, em conjunto com uma plataforma de treino em torno dos exercícios CTF. Existem dois cursos principais, o *offensive security*, com as disciplinas *Red Teaming*, *Penetration Testing Tools*, *Social Engineering* e *Windows Kernel Rootkits*, e o curso *Defensive Security*, com as disciplinas, *Incident Responder*, *Threat Hunting*, *Network Forensics* e *Applied Reverse Engineering*. A plataforma da instituição desenhada para o treino de competências CTF possui mais de 200 horas de treino de alto-nível.

Os *curricula* dos cursos de mestrado na área da segurança informática deverão ser concebidos incluindo aulas práticas e laboratórios, estimulando os alunos a participarem em competições CTF.

Atualmente, nas universidades portuguesas, como é o caso do Instituto Superior Técnico, a Universidade do Porto, o Politécnico de Leiria e a Universidade de Aveiro, tanto os alunos das licenciaturas como os alunos dos mestrados, participam ativamente, em equipa, nas competições CTF. No ano 2017, na competição *SecThon* da *Multicert* [181], na qual participamos, constatamos *in-loco*, que a equipa vencedora, a *STT* do Instituto Superior Técnico, destacou-se claramente das restantes equipas. Essa clara vantagem deve-se ao facto de, o Instituto Superior Técnico, incorporar, há vários anos, exercícios CTF nos seus cursos académicos. Nas disciplinas dos cursos, os alunos são incentivados a resolver os exercícios CTF e a integrarem equipas com o objetivo de participarem nas competições CTF.

O desenvolvimento de *curricula* adequado na área da segurança informática, quer em termos teóricos, quer em termos práticos, nos cursos superiores de informática, aumentará a procura destes cursos, por futuros alunos. As instituições universitárias que tenham esta preocupação na concepção dos seus cursos irá obter uma vantagem competitiva promovendo uma oferta de maior qualidade. Por outro lado, cursos com qualidade reconhecida pelos alunos permitirá aumentar a motivação, melhorar a aprendizagem e o sucesso escolar.

Por exemplo, a Universidade de Tampa que disponibiliza uma laboratório de segurança, onde os alunos tem acesso às mais recentes tecnologias, motiva os alunos para praticaram competências na área de SI e "ethical hacking"em ambientes seguros para o efeito. No vídeo em [226], os alunos dão o seu testemunho acerca das várias parcerias que a Universidade de Tampa estabeleceu com as organizações. Estes referem que a universidade é um excelente local, quer ao nível de equipamentos e materiais disponibilizados no laboratório, quer através da participação das empresas e dos estágios profissionais, para começar uma carreira profissional, e para desenvolver competências técnicas na área da SI.

Por último, a elevada procura por especialistas em segurança informática, por parte das organizações, irá colocar um grande desafio às instituições de ensino superior em adaptar os conteúdos programáticos das suas unidades curriculares ao mundo empresarial. Seria importante um diálogo recíproco entre instituição de ensino superior e empresas na área da segurança informática, permitindo desenvolver competições CTF em contexto real e estágios na área da segurança informática com a componente CTF.

A *Multicert*, empresa de Cibersegurança e Certificação Digital, líder no mercado nacional, da qual já falamos, patrocina e organiza competições CTF para recrutar os melhores alunos de ensino superior para integrarem os seus quadros de trabalhadores.

Presentemente, o DEI estabeleceu uma parceria com a *Feedzai* [3] no sentido de criação de novos projetos pedagógicos na área da programação. Uma sugestão que deixamos neste trabalho é a de, o DEI, estabelecer parcerias com a mesma empresa e outras empresas da área tecnológica no sentido do desenvolvimento de projetos na área SI. Sugerimos, também, ao DEI, instituição onde somos alunos de Mestrado que incentive os alunos dos cursos da área de informática, sobretudos dos cursos de mestrado, a criarem uma equipa, seguindo o exemplo do Instituto Superior Técnico, para participar em competições CTF, ou mesmo organizar, no DEI, onde temos Recursos Humanos (Professores altamente especializados em Segurança Informática) e recursos logísticos (salas bem apetrechadas de meios técnicos) de elevada qualidade, competições CTF inter-escolas universitárias.

Em conclusão, na presente discussão dos resultados, analisamos os mesmos, tendo em conta as hipóteses formuladas, e tentamos apresentar soluções no sentido de preencher as lacunas existentes nos cursos de segurança informática e na falta de profissionais da área, nomeadamente através da metodologia *active learning* e processos de gamificação, utilizando exercícios CTF.

Capítulo 9

Limitações e futuros desenvolvimentos

A limitação mais significativa, que condicionou a implementação da plataforma CTF@DEI, e a sua disponibilização a uma amostra considerável de alunos do mestrado em segurança informática, foi provocada pela *Covid*. Os dois períodos prolongados de confinamento, impuseram aulas *online* em substituição das aulas presenciais.

Este facto, imprevisível, dificultou o objetivo da validação da plataforma e a realização de exercícios CTF, que foi pensado para ser em modo presencial.

O desconfinamento tardio em Outubro de 2021, apenas permitiu a realização dos exercícios CTF, com uma amostra mais reduzida de alunos.

Devido à *Covid*, e às consequências descritas, no objetivo principal do nosso trabalho, surgiu a oportunidade de introduzirmos novos objetivos. Para além, do objetivo principal do nosso trabalho - construção de uma plataforma de exercícios CTF - conseguimos testar e validar, quer do ponto de vista teórico, quer do ponto de vista prático (com as limitações expostas acima relativamente ao tamanho da amostra), quatro hipóteses de trabalho, relativas à importância dos exercícios CTF, nos *curricula* de cursos superiores e no mercado de trabalho da área da segurança informática.

Em suma, as limitações a este trabalho foram simultâneamente uma ameaça e uma oportunidade.

Relativamente aos desenvolvimentos futuros deste projeto, indicamos os seguintes:

- Elaborar novos conteúdos para serem adicionados à plataforma e enriquecer a coleção de exercícios (exercícios CTF, *writeups*, materiais de estudo e PoC);
- Desenvolver novas interfaces *web*, atendendo aos *mockups* apresentados no apêndice 10;
- Integrar vários sistemas de monitorização, que verifiquem os exercícios CTF em execução, e comparar os resultados obtidos entre estes sistemas;
- Implementar outras ferramentas de *static analysis*, que analisem o código fonte, as imagens *Docker* e os ficheiros das máquinas virtuais. Desenvolver critérios de comparação e métricas para estas ferramentas;
- Instalar ferramentas de *dynamic analysis*, para analisar os exercícios CTF durante a

sua execução, e desenvolver critérios de comparação e métricas para estas ferramentas;

- Integrar outras plataformas de virtualização, como *VMware*, *Amazon AWS* ou *Google Cloud*;
- Automatizar a execução de PoC na *CTF Toolkit*, de forma, a que seja possível demonstrar na prática, a resolução dos exercícios CTF;
- Desenvolver um *bot* para o serviço de *chat*, que interaja com os utilizadores, e que permita executar os comandos da *CTF Toolkit* a partir do *chat*;
- Conceber o mecanismo de pausa nos exercício CTF, de forma a que seja possível criar um *snapshot* e analisar o mesmo, através de ferramentas apropriadas.

Por último, apresentar a plataforma CTF@DEI aos alunos do mestrado em segurança informática, no início de cada ano lectivo, no sentido de os motivar e melhorar a sua aprendizagem.

Capítulo 10

Conclusão

Na presente dissertação intitulada "Construção de um jogo de segurança voltado para alunos do MSI"atingimos com sucesso o objetivo proposto, isto é, desenhamos, implementamos, disponibilizamos e validamos a plataforma CTF@DEI, com as funcionalidades propostas, na infraestrutura do DEI, para os alunos do MSI.

Consideramos a presente plataforma, como uma mais valia para os alunos do MSI, uma vez que, através dos resultados apresentados, constatamos que as experiências vivências com os exercícios CTF tem impacto na aprendizagem dos alunos. Verificamos também, que estes cenários práticos possuem relevância na entrada destes alunos, no mercado de trabalho da área da segurança informática.

A plataforma CTF@DEI, de utilização colaborativa, disponibilizada aos alunos do DEI, inclui os seguintes serviços: um sistema para armazenar e coleccionar os exercícios CTF (que, através de uma *pipeline*, analisa e classifica os exercícios); um *website* com a documentação e com os materiais para os exercícios CTF; duas plataformas de virtualização, em particular, o *Kubernetes* e o *VirtualBox*; automatismos para controlar os exercícios CTF nas respetivas plataformas de virtualização; um serviço de *chat*, para onde são enviadas as notificações destes serviços; e por último, uma *Toolkit*, a qual utilizamos para orquestrar os exercícios CTF (iniciar, pausar e terminar os mesmos, nas plataformas de virtualização).

Uma vez que, definimos os procedimentos para gerir os exercícios, através de um ferramenta, chamada *CTF Toolkit*, então os futuros alunos do MSI encontrarão formas de reproduzir devidamente os exercícios existentes na coleção. Bem como, terão a oportunidade de adicionar novos conteúdos à coleção, para os alunos seguintes.

A plataforma desenvolvida foi também pensada para suportar múltiplos serviços, de análise, classificação, monitorização e virtualização dos exercícios CTF. Pelo que, no futuro pretendemos agregar os resultados obtidos através de diferentes serviços e ferramentas.

O facto de termos criado ficheiros de configuração permitiu-nos reproduzir o mesmo exercício CTF, sempre nas mesmas condições. Algo que, nos permitiu reduzir a complexidade em executar os exercícios CTF, como também, nos permitiu fazê-lo de forma automática.

Em suma, salientamos que todas as hipóteses formuladas foram validadas, e portanto, consideramos que as actividades de *Cybersecurity Training*, nomeadamente os exercícios CTF, contribuem para a aprendizagem em segurança informática. A metodologia *Active Learning* e os processos de gamificação, inerentes às competições CTF, viabilizam uma aprendizagem prática e também motivante.

Esta página foi intencionalmente deixada em branco.

Referências

[1] *TED - Gaming can make a better world - Jane McGonigal*, 2020 (acedido em Março, 2020), disponível em: https://www.youtube.com/watch?v=dE1DuBesGYM.

[2] *Learning Management System*, 2020 (acedido em Novembro, 2020), disponível em: https://elearning.iefp.pt/pluginfile.php/46766/mod_scorm/content/0/equ05/06equ05.htm.

[3] *Feedzai*, 2021 (acedido em Agosto, 2021). Disponível em: https://feedzai.com/.

[4] *Git Large File Storage (LFS)*, 2021 (acedido em Fevereiro, 2021), disponível em: https://ctfgit.dei.uc.pt/help/topics/git/lfs/index.

[5] Hussain Aldawood and Geoffrey Skinner. Reviewing cyber security social engineering training and awareness programs-pitfalls and ongoing issues. *Future Internet*, 11(3), 2019.

[6] F. Alotaibi, S. Furnell, I. Stengel, and M. Papadaki. A Review of Using Gaming Technology for Cyber-Security Awareness University of Applied Sciences Karlsruhe , Germany. *Infonomics-Society.Org*, 6(2):660–666, 2016.

[7] F. Alotaibi, S. Furnell, I. Stengel, and M. Papadaki. A Review of Using Gaming Technology for Cyber-Security Awareness University of Applied Sciences Karlsruhe , Germany. *Infonomics-Society.Org*, 6(2):660–666, 2016.

[8] Military Cyber Professionals Association. *Scoring server*, 2020 (acedido em Novembro, 2020), disponível em: https://github.com/mcpa-stlouis/hackthearch.

[9] Jean-philippe Aumasson. *Serious Cryptography*. No Starch Press, Inc., 2018.

[10] Avatao. *Secure coding training Platform*, 2020 (acedido em Novembro, 2020), disponível em: https://avatao.com.

[11] Thomas A Babbitt. The Importance of Using Hacker Contests and Mindset in Teaching Networks and Information Assurance. 2011.

[12] James Banfield and Brad Wilkerson. Increasing Student Intrinsic Motivation And Self-Efficacy Through Gamification Pedagogy. *Contemporary Issues in Education Research (CIER)*, 7(4):291–298, 2014.

[13] Tiffany Bao. Autonomous Computer Security Game : Techniques , Strategy and Investigation. 2018.

[14] Florian Barth and Matthias Luft. Towards a practical approach for teaching IT-security. *ICSIT 2012 - 3rd International Conference on Society and Information Technologies, Proceedings*, pages 300–305, 2012.

[15] Yan Bei, Robert Kesterson, Kyle Gwinnup, and Carol Taylor. Cyber defense competition: a tale of two teams. *Journal of Computing Sciences in Colleges*, 27(1):171–177, 2011.

[16] Razvan Beuran, Ken-ichi Chinen, Yasuo Tan, and Yoichi Shinoda. Towards Effective Cybersecurity Education and Training. *Research report (School of Information Science, Graduate School of Advanced Science and Technology, Japan Advanced Institute of Science and Technology)*, IS-RR-2016:1–16, 2016.

[17] Razvan Beuran, Takuya Inoue, Yasuo Tan, and Yoichi Shinoda. Realistic Cybersecurity Training via Scenario Progression Management. *Proceedings - 4th IEEE European Symposium on Security and Privacy Workshops, EUROS and PW 2019*, pages 67–76, 2019.

[18] Razvan Beuran, Cuong Pham, Dat Tang, Ken Ichi Chinen, Yasuo Tan, and Yoichi Shinoda. Cybersecurity education and training support system: CyRIS. *IEICE Transactions on Information and Systems*, E101D(3):740–749, 2018.

[19] Ahmad Dahaqin Bin Ibrahim, Ahmad Haziq Ashrofie Hanafi, Haikal Rokman, Md Nabil Ahmad Zawawi, Zul Azri Ibrahim, and Fiza Abdul Rahim. Comparative Analysis on Student's Interest in Cyber Security among Secondary School Students using CTF Platform. *2020 8th International Conference on Information Technology and Multimedia, ICIMU 2020*, pages 73–77, 2020.

[20] Fernando Boavida. *Palestra Planear Preparar Apresentar*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.youtube.com/watch?v=mnZ3yWxdW8I.

[21] Kevin Bock, George Hughey, and Dave Levin. King of the Hill: A Novel Cybersecurity Competition for Teaching Penetration Testing. *USENIX Workshop on Advances in Security Education*, pages 1–9, 2018.

[22] Stefan Boesen, Richard Weiss, James Sullivan, Michael E. Locasto, Jens Mache, and Erik Nilsen. EDURange: Meeting the pedagogical challenges of student participation in cybertraining environments. *7th Workshop on Cyber Security Experimentation and Test, CSET 2014*, pages 2–3, 2014.

[23] K. Boopathi, S. Sreejith, and A. Bithin. Learning cyber security through gamification. *Indian Journal of Science and Technology*, 8(7):642–649, 2015.

[24] Root The Box. *Github - RootTheBox issue 356*, 2020 (acedido em Agosto, 2021). Disponível em: https://github.com/moloch–/RootTheBox/issues/356.

[25] Agile Business. *MoSCoW Prioritisation*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.agilebusiness.org/page/ProjectFramework_10_MoSCoWPrioritisation.

[26] busybox. *BusyBox*, 2021 (acedido em Outubro, 2021). Disponível em: https://busybox.net/.

[27] Krzysztof Cabaj, Dulce Domingos, Zbigniew Kotulski, and Ana Respício. Cybersecurity education: Evolution of the discipline and analysis of master programs. *Computers and Security*, 75:24–35, 2018.

[28] Krzysztof Cabaj, Dulce Domingos, Zbigniew Kotulski, and Ana Respício. Cybersecurity education: Evolution of the discipline and analysis of master programs. *Computers and Security*, 75:24–35, 2018.

[29] Tigera Calico. *Get Calico up and running in your Kubernetes cluster*, 2021 (acedido em Agosto, 2021). Disponível em: https://docs.projectcalico.org/getting-started/kubernetes/.

[30] Nicholas Capalbo, Theodore Reed, and Michael Arpaia. RTFn : Enabling Cybersecurity Education through a Mobile Capture the Flag Client. *Proceedings of SAM'11*, pages 500–506, 2011.

[31] Martin Carlisle, Michael Chiaramonte, and David Caswell. Using CTFs for an Undergraduate Cyber Education. (Figure 1).

[32] Pavel Čeleda, Jan Vykopal, and Karel Slavíček. KYPO4INDUSTRY : A Testbed for Teaching Cybersecurity of Industrial Control Systems. pages 1026–1032, 2020.

[33] Scott Chacon. *Pro Git book*, 2020 (acedido em Novembro, 2020), disponível em: https://git-scm.com/book/en/v2.

[34] We chall. *Wargames*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.wechall.net/.

[35] Ernest Chan. *Gamifying Data Science Education*, 2020 (acedido em Agosto, 2021). Disponível em: https://duo.com/labs/research/gamifying-data-science-education.

[36] Peter Chapman, Jonathan Burket, and David Brumley. PicoCTF: A game-based computer security competition for high school students. *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education, 3GSE 2014*, (May), 2014.

[37] Rocket Chat. *Rocket Chat - Installing and Updating*, 2020 (acedido em Agosto, 2021). Disponível em: https://docs.rocket.chat/quick-start/installing-and-updating.

[38] Rocket Chat. *Rocket.Chat Developer - API*, 2020 (acedido em Novembro, 2020), disponível em: https://developer.rocket.chat/reference/api.

[39] Rocket Chat. *Real-time conversations with your colleagues, other companies or customers*, 2020 (acedido em Novembro, 2020), disponível em: https://rocket.chat/.

[40] Ronald S. Cheung, Joseph Paul Cohen, Henry Z. Lo, and Fabio Elia. Challenge Based Learning in Cybersecurity Education. 1, 2011.

[41] Ronald S. Cheung, Joseph Paul Cohen, Henry Z. Lo, Fabio Elia, and Veronica Carrillo-Marquez. Effectiveness of Cybersecurity Competitions. *Proceedings of the International Conference on Security and Management (SAM)*, 1:1, 2012.

[42] Girshel Chokhonelidze, Giorgi Basilaia, Mikheil Kantaria, and Marine Dgebuadze. Teaching the Cybersecurity Courses at the University in Georgia. *International Journal of Innovative Science and Research Technology*, 5(4), 2020.

[43] Tom Chothia and Chris Novakovic. 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 2015*, 2015.

[44] Kevin Chung and Julian Cohen. Learning Obstacles in the Capture The Flag Model. *Usenix*, 2014.

[45] Clonezilla. *The Free and Open Source Software for Disk Imaging and Cloning*, 2021 (acedido em Agosto, 2021). Disponível em: https://clonezilla.org.

[46] Adobe Experience Cloud. *Waterfall Methodology*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.workfront.com/project-management/methodologies/waterfall.

[47] CNCF. *Cloud Native Interactive Landscape*, 2021 (acedido em Agosto, 2021). Disponível em: https://landscape.cncf.io.

[48] Roberto Cohen. *Gamification em Help Desk e Service Desk*, 2020 (acedido em Maio, 2020), disponível em: https://books.apple.com/cy/book/gamification-em-help-desk-e-service-desk/id1246375785.

[49] DEF CON. *The DEF CON Story*, 2021 (acedido em Agosto, 2021). Disponível em: https://defcon.org/html/links/dc-about.html.

[50] Art Conklin. Cyber defense competitions and information security education: An active learning solution for a capstone course. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 9(C):1–6, 2006.

[51] Software Freedom Conservancy. *Git Large File Storage*, 2020 (acedido em Novembro, 2020), disponível em: https://git-lfs.github.com/.

[52] Daniel Conte de Leon, Christopher E. Goes, Michael A. Haney, and Axel W. Krings. ADLES: Specifying, deploying, and sharing hands-on cyber-exercises. *Computers and Security*, 74:12–40, 2018.

[53] Defta Costinela-Luminita. Information security in E-learning Platforms. *Procedia - Social and Behavioral Sciences*, 15:2689–2693, 2011.

[54] C. Cowan, S. Arnold, S. Beattie, C. Wright, and J. Viega. Defcon Capture the Flag: Defending vulnerable code from intense attack. *Proceedings - DARPA Information Survivability Conference and Exposition, DISCEX 2003*, 1(May 2003):120–129, 2003.

[55] Tiago Cruz and Paulo Simões. Fostering cybersecurity awareness among computing science undergraduate students: motivating by example. *European Conference on Information Warfare and Security, ECCWS*, 2020-June(June):72–81, 2020.

[56] Tiago Cruz and Paulo Simões. Down the Rabbit Hole: Fostering Active Learning through Guided Exploration of a SCADA Cyber Range. *Applied Sciences*, 11(20):9509, 2021.

[57] CTF. *CTF - Challenge and opportunity*, 2021 (acedido em Outubro, 2021). Disponível em: https://www.bcs.org/articles-opinion-and-research/ctf-challenge-and-opportunity/.

[58] Awesome CTF. *A curated list of Capture The Flag frameworks, libraries, resources, softwares and tutorials*, 2020 (acedido em Novembro, 2020), disponível em: https://github.com/apsdehal/awesome-ctf.

[59] ctfd. *Plataforma CTFd - About*, 2020 (acedido em Agosto, 2021). Disponível em: https://ctfd.io/about.

[60] CTFd. *CTFd is a Capture The Flag framework*, 2021 (acedido em Agosto, 2021). Disponível em: https://github.com/CTFd/CTFd.

[61] ctftime. *CTFTime - oauth*, 2020 (acedido em Agosto, 2021). Disponível em: https://ctftime.org/for-organizers/oauth/.

[62] CTFTime. *CTFTime About*, 2021 (acedido em Agosto, 2021). Disponível em: https://ctftime.org/about.

[63] CTFTime. *OAuth2 configuration*, 2021 (acedido em Agosto, 2021). Disponível em: https://ctftime.org/for-organizers/oauth/.

[64] ctftime. *CTFtime - Writeups*, 2021 (acedido em Agosto, 2021). Disponível em: https://ctftime.org/writeups.

[65] cybersec4europe. *Deliverable 6.2 of the Cyber Security for Europe project*, 2020 (acedido em Agosto, 2021). Disponível em: https://cybersec4europe.eu/addressing-the-shortage-of-cybersecurity-skills-in-europe/.

[66] European Union Agency For Cybersecurity. *ENISA Cyber Security Training - Introduction to Network Forensics*. Number Final Version 1.1 August 2019. 2019.

[67] Adrian Dabrowski, Markus Kammerstetter, Eduard Thamm, Edgar Weippl, and Wolfgang Kastner. Leveraging Competitive Gamification for Sustainable Fun and Profit in Security Education. *USENIX Summit on Gaming, Games, and Gamification in Security Education*, pages 1–8, 2015.

[68] Dipankar Dasgupta, Denise M. Ferebee, and Zbigniew Michalewicz. Applying Puzzle-based Learning to cyber-security education. *Proceedings of the 2013 Information Security Curriculum Development Conference, InfoSec CD 2013*, pages 20–26, 2013.

[69] Datamation. *Advantages of Using MySQL*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.datamation.com/storage/8-major-advantages-of-using-mysql/.

[70] Andy Davis, Tim Leek, Michael Zhivich, Kyle Gwinnup, and William Leonard. The Fun and Future of CTF *. *Usenix*, 2014.

[71] Chris Davis, Aaron Philipp, and David Cowen. *Hacking Exposed Computer Forensics*. McGraw-Hill, Inc., USA, 1 edition, 2004.

[72] Jon Davis and Shane Magrath. A Survey of Cyber Ranges and Testbeds. page 29, 2013.

[73] Maurice Dawson. National Cybersecurity Education: Bridging Defense to Offense. *Land Forces Academy Review*, 25(1):68–75, 2020.

[74] Departamento de Engenharia Informática Coimbra. *Plataforma Estágios*, 2021 (acedido em Agosto, 2021). Disponível em: https://estagios.dei.uc.pt.

[75] Debian. *Vi IMproved with scripting languages support*, 2020 (acedido em Agosto, 2021). Disponível em: https://packages.debian.org/buster/vim-nox.

[76] DETER. *Research project and operator of DETERLab*, 2020 (acedido em Novembro, 2020), disponível em: https://deter-project.org/.

[77] Sebastian Deterding, Dan Dixon, Rilla Khaled, and Lennart Nacke. From game design elements to gamefulness - Defining gamification. *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, MindTrek 2011*, (March 2014):9–15, 2011.

[78] Docker. *Docker Compose*, 2020 (acedido em Agosto, 2021). Disponível em: https://docs.docker.com/compose.

[79] Docker. *Docker Network Bridge*, 2020 (acedido em Agosto, 2021). Disponível em: https://docs.docker.com/network/bridge.

[80] Docker. *Docker Storage Bind-mounts*, 2020 (acedido em Agosto, 2021). Disponível em: https://docs.docker.com/storage/bind-mounts.

[81] Docker. *Empowering App Development*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.docker.com.

[82] Docker Docs. *Docker Registry*, 2020 (acedido em Novembro, 2020), disponível em: https://docs.docker.com/registry/.

[83] Kubernetes Documentation. *What is Kubernetes?*, 2021 (acedido em Agosto, 2021). Disponível em: https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/.

[84] Chris Eagle. Computer security competitions: Expanding educational outcomes. *IEEE Security and Privacy*, 11(4):69–71, 2013.

[85] Christopher Eagle. Using capture the flag events as training opportunities. *Hitachi Review*, 63(5):301–303, 2014.

[86] ECSO. Understanding Cyber Ranges: From Hype to Reality. (March), 2020.

[87] Jon Erickson. *Hacking: The Art of Exploitation, 2nd Edition.* No Starch Press, USA, second edition, 2008.

[88] Tiago Espinha Gasiba, Kristian Beckers, Santiago Suppan, and Filip Rezabek. On the requirements for serious games geared towards software developers in the industry. *Proceedings of the IEEE International Conference on Requirements Engineering*, 2019-Septe:286–296, 2019.

[89] Karen Evans and Franklin Reeder. *A Human Capital Crisis in Cybersecurity.* Number July. 2010.

[90] exabeam. *Creeper: The World's First Computer Virus*, 2019 (acedido em Agosto, 2021). Disponível em: https://www.exabeam.com/information-security/creeper-computer-virus.

[91] F-Secure. *Attack landscape update - Ransomware, automated recon, and supply chain attacks*, 2021 (acedido em Agosto, 2021). Disponível em: https://blog.f-secure.com/attack-landscape-update-h1-2021.

[92] Facebook. *Official CTF - Facebook CTF*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.facebook.com/officialctf/.

[93] FBCTF. *The Facebook CTF is a platform to host Jeopardy and King of the Hill style Capture the Flag competitions*, 2020 (acedido em Novembro, 2020), disponível em: https://github.com/facebookarchive/fbctf.

[94] Bernard Ferguson, Anne Tall, and Denise Olsen. National cyber range overview. *Proceedings - IEEE Military Communications Conference MILCOM*, pages 123–128, 2014.

[95] Flaticon. *Cyber security Icons*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.flaticon.com/free-icons/cyber-security.

[96] Vitaly Ford, Ambareen Siraj, Ada Haynes, and Eric Brown. Capture the flag unplugged: An offline cyber competition. *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*, (July 2018):225–230, 2017.

[97] Apache Software Foundation. *The Apache HTTP Server Project*, 2021 (acedido em Agosto, 2021). Disponível em: https://httpd.apache.org/.

[98] Electronic Frontier Foundation. *Certbot - automate the HTTPS process*, 2021 (acedido em Agosto, 2021). Disponível em: https://certbot.eff.org.

[99] Linux Foundation. *Let's Encrypt*, 2021 (acedido em Agosto, 2021). Disponível em: https://letsencrypt.org/about/.

[100] Maximilian Frank, Maria Leitner, and Timea Pahi. Design Considerations for Cyber Security Testbeds: A Case Study on a Cyber Security Testbed for Education. *Proceedings - 2017 IEEE 15th International Conference on Dependable, Autonomic and Secure Computing, 2017 IEEE 15th International Conference on Pervasive Intelligence and Computing, 2017 IEEE 3rd International Conference on Big Data Intelligence and Compu*, 2018-Janua:38–46, 2018.

[101] Jessica Fridrich. Steganography in Digital Media. *Steganography in Digital Media*, 2009.

[102] David Ganan, Santi Caballe, Robert Clariso, and Jordi Conesa. Analysis and Design of an eLearning Platform Featuring Learning Analytics and Gamification. *Proceedings - 2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2016*, (November 2017):87–94, 2016.

[103] Efstratios Gavas, Nasir Memon, and Douglas Britton. Winning cybersecurity one challenge at a time. *IEEE Security and Privacy*, 10(4):75–79, 2012.

[104] Paulo J S Gil and Ana F C Relvas. O Pequeno Livro da Dissertação Recursos para realizar uma tese sem dor. 2015.

[105] Github. *picoCTF - The platform used to run picoCTF 2019*, 2019 (acedido em Agosto, 2021). Disponível em: https://github.com/picoCTF/picoCTF.

[106] Github. *Github - CTFd issue 1642*, 2020 (acedido em Agosto, 2021). Disponível em: https://github.com/CTFd/CTFd/issues/1642.

[107] Github. *Github Plugins CTFd*, 2020 (acedido em Agosto, 2021). Disponível em: https://github.com/CTFd/plugins.

[108] Github. *Root The Box - Exercices*, 2021 (acedido em Agosto, 2021). Disponível em: github.com/moloch–/RootTheBox/wiki/Screenshots.

[109] Gitlab. *Gitlab Runner Executor*, 2020 (acedido em Agosto, 2021). Disponível em: https://docs.gitlab.com/runner/executors/.

[110] Gitlab. *Gitlab Runner installation*, 2020 (acedido em Agosto, 2021). Disponível em: https://docs.gitlab.com/runner/install/linux-manually.html.

[111] GitLab. *GitLab Workflow*, 2020 (acedido em Novembro, 2020), disponível em: https://about.gitlab.com/topics/version-control/what-is-gitlab-workflow/.

[112] GitLab. *Container Registry*, 2020 (acedido em Novembro, 2020), disponível em: https://ctfgit.dei.uc.pt/help/user/packages/container_registry/index.

[113] GitLab. *Documentation*, 2020 (acedido em Novembro, 2020), disponível em: https://docs.gitlab.com/ee/README.html.

[114] GitLab. *Executors Runner*, 2020 (acedido em Novembro, 2020), disponível em: https://docs.gitlab.com/runner/.

[115] Gitlab. *Gitlab Docs - The Shell executor*, 2021 (acedido em Agosto, 2021). Disponível em: https://docs.gitlab.com/runner/executors/shell.html.

[116] Hugo Gonzalez, Rafael Llamas, and Omar Monta. Using a CTF Tournament for Reinforcing Learned Skills in Cybersecurity Course Using CTF Tournament for Reinforcing Learned Skills in Cybersecurity Course. (March 2020), 2019.

[117] Hugo Gonzalez, Rafael Llamas, and Francisco Ordaz. Cybersecurity Teaching through Gamification: Aligning Training Resources to our Syllabus. *Research in Computing Science*, 146:35–43, 2017.

[118] Google. *capturetheflag*, 2021 (acedido em Agosto, 2021). Disponível em: https://capturetheflag.withgoogle.com.

[119] Haaukins. *Highly accessible and automated virtualization platform for security education*, 2020 (acedido em Novembro, 2020), disponível em: https://github.com/aau-network-security/haaukins.

[120] Stephen Harris. Hacking the skills shortage. *Engineer (Online Edition)*, page 1, 2013.

[121] Stephen Hart, Andrea Margheri, Federica Paci, and Vladimiro Sassone. Riskio: A Serious Game for Cyber Security Awareness and Education. *Computers and Security*, 95:1–18, 2020.

[122] Regina D Hartley. Ethical Hacking Pedagogy: An Analysis and Overview of Teaching Students to Hack. *Journal of International Technology and Information Management*, 24(4):94–104, 2015.

[123] Regina D Hartley. Ethical Hacking Pedagogy: An Analysis and Overview of Teaching Students to Hack. *Journal of International Technology and Information Management*, 24(4):94–104, 2015.

[124] Maurice Hendrix, Ali Al-Sherbaz, and Victoria Bloom. Game Based Cyber Security Training: are Serious Games suitable for cyber security training? *International Journal of Serious Games*, 3(1), 2016.

[125] J.W. Ho, N. Mallesh, and M. Wright. The Design and Lessons of the ASCENT Security Teaching Lab. *Proceedings of the 13th Colloquium for Information Systems Security Education*, (0621280):124–132, 2009.

[126] V. Ho. Learning by Doing. *Encyclopedia of Health Economics*, (July):141–145, 2014.

[127] howard. *Cybersecurity Center Programs (Extra-Curricular)*, 2021 (acedido em Agosto, 2021). Disponível em: https://business.howard.edu/academic-centers/cybersecurity-education-and-research-center-cerc/cybersecurity-center-programs.

[128] Docker HUB. *Build and Ship any Application Anywhere*, 2021 (acedido em Agosto, 2021). Disponível em: https://hub.docker.com.

[129] Hubot. *Hubot is your friendly robot sidekick*, 2020 (acedido em Novembro, 2020), disponível em: https://hubot.github.com/.

[130] hucerc. *Howard University Cyber Security and Research Center*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.hucerc.com/.

[131] hucerc. *Skills needed for Capture The Flag (CTF)*, 2021 (acedido em Agosto, 2021). Disponível em: http://www.hucerc.com/basic-tools-and-skills-needed-for-capture-the-flag/.

[132] IBM. *What is docker?*, 2020 (acedido em Novembro, 2020), disponível em: https://www.ibm.com/in-en/cloud/learn/docker.

[133] infosecinstitute. *Penetration Testing: Job Knowledge and Professional Development*, 2016 (acedido em Agosto, 2021). Disponível em: https://resources.infosecinstitute.com/topic/penetration-testing-job-knowledge-professional-development.

[134] Infosec Institute. *Security awareness training and phishing simulations*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.infosecinstitute.com/iq/.

[135] Cynthia Irvine. The value of capture-the-flag exercises in education: An interview with chris eagle. *IEEE Security and Privacy*, 9(6):58–60, 2011.

[136] ISACA. STATE OF CYBERSECURITY 2020: GLOBAL UPDATE ON WORKFORCE EFFORTS AND RESOURCES State of Cybersecurity 2020 Part 1: Global Update on Workforce Efforts and Resources Personal Copy of Raul Rivera (ISACA ID: 395792). pages 1–23, 2020.

[137] JCraft. *JSch is a pure Java implementation of SSH2*, 2021 (acedido em Outubro, 2021). Disponível em: http://www.jcraft.com/jsch/.

[138] Jekyll. *Jekyll - Transform your plain text into static websites and blogs*, 2020 (acedido em Agosto, 2021). Disponível em: https://jekyllrb.com.

[139] Jenkins. *Build, test, and deploy software*, 2020 (acedido em Novembro, 2020), disponível em: https://www.jenkins.io/doc/tutorials/.

[140] Craig Jordan, Matt Knapp, Dan Mitchell, Mark Claypool, and Kathi Fisler. CounterMeasures: A game for teaching computer security. *Annual Workshop on Network and Systems Support for Games*, pages 1–6, 2011.

[141] Kali. *The most advanced Penetration Testing Distribution*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.kali.org/.

[142] Stylianos Karagiannis, Christoforos Ntantogian, Emmanouil Magkos, Luís L. Ribeiro, and Luís Campos. PocketCTF: A fully featured approach for hosting portable attack and defense cybersecurity exercises. *Information (Switzerland)*, 12(8):1–13, 2021.

[143] Stylianos Karagiannis, Thanos Papaioannou, Emmanouil Magkos, and Aggeliki Tsohou. Game-Based Information Security/Privacy Education and Awareness: Theory and Practice. *Lecture Notes in Business Information Processing*, 402(November):509–525, 2020.

[144] Menelaos Katsantonis, Panayotis Fouliras, and Ioannis Mavridis. Conceptual analysis of cyber security education based on live competitions. *IEEE Global Engineering Education Conference, EDUCON*, (April):771–779, 2017.

[145] Menelaos Katsantonis, Panayotis Fouliras, and Ioannis Mavridis. Conceptual analysis of cyber security education based on live competitions. *IEEE Global Engineering Education Conference, EDUCON*, (April):771–779, 2017.

[146] kboom. *kboom - short-term load for scale testing and long-term load for soak testing*, 2021 (acedido em Outubro, 2021). Disponível em: https://github.com/mhausenblas/kboom.

[147] Kenneth J. Knapp, Christopher Maurer, and Miloslava Plachkinova. Maintaining a cybersecurity curriculum: Professional certifications as valuable guidance. *Journal of Information Systems Education*, 28(2):101–114, 2017.

[148] Kontra. *Kontra - Application Security Training*, 2021 (acedido em Agosto, 2021). Disponível em: https://application.security/free/owasp-top-10.

[149] Kubernetes. *Kubernetes Docs - services networking*, 2020 (acedido em Agosto, 2021). Disponível em: https://kubernetes.io/docs/concepts/services-networking/service/.

[150] Kubernetes. *Automated container deployment, scaling, and management*, 2021 (acedido em Agosto, 2021). Disponível em: https://kubernetes.io.

[151] Kubernetes. *Kubernetes - Install Tools*, 2021 (acedido em Agosto, 2021). Disponível em: https://kubernetes.io/docs/tasks/tools/.

[152] Cheng Chung Kuo, Kai Chain, and Chu Sing Yang. Cyber attack and defense training: Using emulab as a platform. *International Journal of Innovative Computing, Information and Control*, 14(6):2245–2258, 2018.

[153] Pentester Lab. *We make learning web hacking easier*, 2021 (acedido em Agosto, 2021). Disponível em: https://pentesterlab.com/.

[154] Technical Hacking Lab. *Hacking Lab Infrastructure*, 2020 (acedido em Novembro, 2020), disponível em: http: //www.hacking-lab-ctf.com/technical.html.

[155] Template LAB. *Template LAB: 25 Great Fishbone Diagram Templates*, 2021 (acedido em Agosto, 2021). Disponível em: https://templatelab.com/fishbone-diagram-templates/.

[156] Practical Pentest Labs. *Take your Hacking skills to the next level*, 2021 (acedido em Agosto, 2021). Disponível em: https://practicalpentestlabs.com.

[157] lastline. *A Brief History of Malware — Its Evolution and Impact*, 2018 (acedido em Agosto, 2021). Disponível em: https://www.lastline.com/blog/history-of-malware-its-evolution-and-impact.

[158] Phil Legg, Thomas Higgs, Pennie Spruhan, Jonathan White, and Ian Johnson. "Hacking an IoT Home": New opportunities for cyber security education combining remote learning with cyber-physical systems. pages 1–4, 2021.

[159] Michael Lehrfeld and Phillip Guest. Building an ethical hacking site for learning and student engagement. *Conference Proceedings - IEEE SOUTHEASTCON*, 2016-July, 2016.

[160] Chengcheng Li and Rucha Kulkarni. Survey of cybersecurity education through gamification. *ASEE Annual Conference and Exposition, Conference Proceedings*, 2016-June, 2016.

[161] LibreCTF. *Open-source CTF platform from EasyCTF*, 2020 (acedido em Novembro, 2020), disponível em: https://github.com/easyctf/openctf.

[162] George Louthan, Warren Roberts, Matthew Butler, and John Hale. The Blunderdome: An Offensive Exercise for Building Network, Systems, and Web Security Awareness. *Cset*, 2010.

[163] James Lyne. Cybersecurity recruitment challenge. *Infosecurity*, 7(5):37, 2010.

[164] Leandros Maglaras. Cyber Ranges and TestBeds for Education, Training, and Research. (February), 2021.

[165] Daniel Manson and Anna Carlin. A league of our own :The future of cyber defense competitions. *Communications of the IIMA*, 11(2):1–11, 2011.

[166] Alexander Mansurov. A CTF-Based Approach in Information Security Education: An Extracurricular Activity in Teaching Students at Altai State University, Russia. *Modern Applied Science*, 10(11):159, 2016.

[167] William R Marchand, Edwin Vega, and José Santillan. Capture the Flag for Computer Security Learning. *IX Congreso Iberoamericano de Seguridad Informática y IV Taller Educativo TIBETS*, (September 2019):300–304, 2017.

[168] G Markowsky, D Johnson, A Moody, R Soucy, and W Stackpole. The 2013 NECCDC - Lessons Learned. (May), 2013.

[169] Ben Martini and Kim Kwang Raymond Choo. Building the next generation of cyber security professionals. *ECIS 2014 Proceedings - 22nd European Conference on Information Systems*, pages 0–13, 2014.

[170] Lucas McDaniel, Erik Talvi, and Brian Hay. Capture the flag as cyber security introduction. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2016-March:5479–5486, 2016.

[171] Lucas McDaniel, Erik Talvi, and Brian Hay. Capture the flag as cyber security introduction. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2016-March:5479–5486, 2016.

[172] Jane McGonigal. *TED Talk - Gaming can make a better world*, 2010 (acedido em Maio, 2020), disponível em: https://www.youtube.com/watch?v=dE1DuBesGYM.

[173] Jane McGonigal. *Reality Is Broken: Why Games Make Us Better and How They Can Change the World*, 2011 (acedido em Maio, 2020), disponível em: https://www.amazon.com/Reality-Is-Broken-Better-Change/dp/0143120611.

[174] Mellivora. *CTF engine*, 2020 (acedido em Novembro, 2020), disponível em: https://github.com/Nakiami/mellivora.

[175] Mendeley. *Reference Manager For Desktop*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.mendeley.com/download-desktop/.

[176] MetaCompliance. *Why Is Security Awareness Training Important?*, 2020 (acedido em Novembro, 2020), disponível em: https://www.metacompliance.com/blog/why-is-security-awareness-training-important/.

[177] Martin Mink and Felix C. Freiling. Is attack better than defense?: Teaching information security the right way. *Proceedings of the 2006 Information Security Curriculum Development Conference, InfoSecCD '06*, pages 44–48, 2007.

[178] mkCTF. *A CTF framework to create, build, deploy and monitor challenges*, 2020 (acedido em Novembro, 2020), disponível em: https://github.com/koromodako/mkctf.

[179] Thomas Morrow. Intelligent Virtual Agents in Health Care. *Specialty Pharmacy Times*, 3661(April):2–7, 2013.

[180] mosseinstitute. *Mosse Institute*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.mosse-institute.com/company/history.html.

[181] Multicert. *Secthon Multicert*, 2021 (acedido em Agosto, 2021). Disponível em: https://secthon.multicert.com/.

[182] Ryotaro Nakata and Akira Otsuka. CyExec*: A High-Performance Container-Based Cyber Range With Scenario Randomization. *IEEE Access*, 9:109095–109114, 2021.

[183] NightShade. *Simple security capture the flag framework*, 2020 (acedido em Novembro, 2020), disponível em: https://github.com/UnrealAkama/NightShade.

[184] NMAP. *Nmap Network Scanning: Port Scanning Techniques and Algorithms*, 2021 (acedido em Agosto, 2021). Disponível em: https://nmap.org/book/scan-methods.html.

[185] Digital Ocean. *How to secure Apache with let's encrypt on Ubuntu*, 2020 (acedido em Agosto, 2021). Disponível em: https://www.digitalocean.com/community/tutorials/how-to-secure-apache-with-let-s-encrypt-on-ubuntu-18-04-pt.

[186] University of Maryland. *King of the Hill: A Novel Cybersecurity Competition for Teaching Penetration Testing*, 2021 (acedido em Agosto, 2021). Disponível em: https://koth.cs.umd.edu/.

[187] Mike O'Leary. Innovative pedagogical approaches to a capstone laboratory course in cyber operations. *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*, pages 429–434, 2017.

[188] Ryan Elfmaster O'Neill. *Learning Linux Binary Analysis*. Packt Publishing, 2016.

[189] open Project. *Open source project management software*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.openproject.org.

[190] Oracle. *Oracle VM VirtualBox - a powerful x86 and AMD64/Intel64 virtualization*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.virtualbox.org.

[191] Oracle. *VBoxManage*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.virtualbox.org/manual/ch08.html.

[192] osirislab. *A repository of challenges from various CTF competitions*, 2021 (acedido em Março, 2021), disponível em: https://github.com/osirislab/CTF-Challenges.

[193] Radek Ošlejšek, Jan Vykopal, Karolína Burská, and Vít Rusˇ. Evaluation of Cyber Defense Exercises Using Visual Analytics Process. 2018.

[194] Overleaf. *Editor LaTeX Online*, 2021 (acedido em Agosto, 2021). Disponível em: https://pt.overleaf.com/.

[195] OWASP. *OWASP - Juice Shop*, 2020 (acedido em Agosto, 2021). Disponível em: https://owasp.org/www-project-juice-shop/.

[196] OWASP. *OWASP - vulnerabilities*, 2021 (acedido em Agosto, 2021). Disponível em: https://owasp.org/www-community/vulnerabilities/.

[197] OWASP. *Projeto Juice Shop*, 2021 (acedido em Agosto, 2021). Disponível em: https://owasp.org/www-project-juice-shop/.

[198] OWASP. *Projeto Webgoat*, 2021 (acedido em Agosto, 2021). Disponível em: https://owasp.org/www-project-webgoat/.

[199] Kentrell Owens, Alexander Fulton, Luke Jones, and Martin Carlisle. pico-Boo!: How to avoid scaring students away in a CTF competition. *Journal of The Colloquium for Information System Security Education*, 2019.

[200] PicoCTF. *The platform used to run picoCTF 2019*, 2020 (acedido em Novembro, 2020), disponível em: https://github.com/picoCTF/picoCTF.

[201] Portswigger. *Portswigger - web security*, 2021 (acedido em Agosto, 2021). Disponível em: https://portswigger.net/web-security.

[202] portswigger. *PortSwigger - Web Security Academy*, 2021 (acedido em Agosto, 2021). Disponível em: https://portswigger.net/web-security.

[203] Arvind S Raj, Bithin Alangot, Seshagiri Prabhu, and Krishnashree Achuthan. Scalable and lightweight CTF infrastructures using application containers. *2016 USENIX Workshop on Advances in Security Education (ASE 16)*, 2016.

[204] Raghu Raman, Sherin Sunny, Vipin Pavithran, and Krishnasree Achuthan. Framework for evaluating Capture the Flag (CTF) security competitions. *2014 International Conference for Convergence of Technology, I2CT 2014*, pages 1–5, 2014.

[205] Redis. *in-memory data structure store*, 2021 (acedido em Agosto, 2021). Disponível em: https://redis.io.

[206] Chris Sanders and Jason Smith. *Applied Network Security Monitoring: Collection, Detection, and Analysis*. 2014.

[207] Sam Scholefield and Lynsay A. Shepherd. Gamification Techniques for Raising Cyber Security Awareness. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11594 LNCS:191–203, 2019.

[208] Z Schreuders and E Butterfield. Gamification for teaching and learning computer security in higher education. *USENIX Workshop on Advances in Security Education*, page 4, 2016.

[209] Z Cliffe Schreuders, Thomas Shaw, Mac Muireadhaigh, and Paul Staniforth. Hackerbot: Attacker Chatbots for Randomised and Interactive Security Labs, Using SecGen and oVirt. *USENIX Workshop on Advances in Security Education*, 2018.

[210] Z. Cliffe Schreuders, Thomas Shaw, Mohammad Shan-A-Khuda, Gajendra Ravichandran, Jason Keighley, and Mihai Ordean. Security Scenario Generator (SecGen): A Framework for Generating Randomly Vulnerable Rich-scenario VMs for Learning Computer Security and Hosting CTF Events. *Ase'17*, 2017.

[211] Scorebot. *Repositório Scorebot*, 2020 (acedido em Novembro, 2020), disponível em: https://github.com/legitbs/scorebot.

[212] SecGen. *Create randomly insecure VMs*, 2020 (acedido em Novembro, 2020), disponível em: https://github.com/cliffe/SecGen.

[213] Shachar Siboni, Vinay Sachidananda, Yair Meidan, Michael Bohadana, Yael Mathov, Suhas Bhairav, Asaf Shabtai, and Yuval Elovici. Security Testbed for Internet-of-Things Devices. *IEEE Transactions on Reliability*, 68(1):23–44, 2019.

[214] SimSpace. *Cyber Range*, 2020 (acedido em Novembro, 2020), disponível em: https://simspace.com/.

[215] Echothrust Solutions. *echoCTF is a pioneer computer security framework*, 2020 (acedido em Novembro, 2020), disponível em: https://github.com/echoCTF/echoCTF.RED.

[216] SonarQube. *Code Quality and Code Security*, 2020 (acedido em Novembro, 2020), disponível em: https://docs.sonarqube.org/latest/.

[217] William Stallings. *Network Security Essentials 4th Edition*. 2015.

[218] Dafydd Stuttard and Marcus Pinto. *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*, volume 7. 2011.

[219] Hector Suarez. SSETGami : Secure Software Education Through Gamification. 2017.

[220] Sudonull. *Facebook uploaded its CTF platform on Github*, 2021 (acedido em Agosto, 2021). Disponível em: https://sudonull.com/post/88474-Facebook-uploaded-its-CTF-platform-on-Github-Facebook-CTF.

[221] Valdemar Sv and Jan Vykopal. Gathering Insights from Teenagers ' Hacking Experience with Authentic Cybersecurity Tools. pages 12–15, 2018.

[222] Valdemar Švábenský, Pavel Čeleda, Jan Vykopal, and Silvia Brišáková. Cybersecurity Knowledge and Skills Taught in Capture the Flag Challenges arXiv : 2101 . 01421v1 [cs . CR] 5 Jan 2021. pages 1–32, 2020.

[223] Valdemar Švábenský and Jan Vykopal. Challenges Arising from Prerequisite Testing in Cybersecurity Games. pages 56–61, 2018.

[224] Valdemar Švábenský, Jan Vykopal, and Pavel Čeleda. What Are Cybersecurity Education Papers About? A Systematic Literature Review of SIGCSE and ITiCSE Conferences. pages 2–8, 2020.

[225] David Szedlak and Andrew M'Manga. Eliciting Requirements for a Student-focussed Capture the Flag. *Proceedings of 2020 7th IEEE International Conference on Behavioural and Social Computing, BESC 2020*, 2020.

[226] Tampa. *The University of Tampa - Cybersecurity Degree*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.ut.edu/academics/sykes-college-of-business/information-and-technology-management-(itm)-degrees/cybersecurity-degree.

[227] Clark Taylor, Pablo Arias, Jim Klopchic, Celeste Matarazzo, and Evi Dube. CTF: State-of-the-Art and Building the Next Generation. *2017 USENIX Workshop on Advances in Security Education (ASE 17)*, 2017.

[228] TED. *TED - The puzzle of motivation*, 2016 (acedido em Agosto, 2021). Disponível em: https://www.ted.com/talks/dan_pink_the_puzzle_of_motivation.

[229] TeXstudio. *A LaTeX editor*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.texstudio.org/.

[230] Root the Box. *A Game of Hackers (CTF Scoreboard, Game Manager)*, 2020 (acedido em Novembro, 2020), disponível em: https://github.com/moloch–/RootTheBox.

[231] Smash the Stack. *Smash the stack wargaming network*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.smashthestack.org/.

[232] Over the Wire. *Wargames*, 2021 (acedido em Agosto, 2021). Disponível em: https://overthewire.org/wargames/.

[233] Hack this site. *safe and legal training ground for hackers to test and expand their ethical hacking skills with challenges and CTFs*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.hackthissite.org/.

[234] Ed Tittel, Mike Chapple, and Michael James Stewart. *Certified Information Systems Security Professional (CISSP): Study Guide*. Wiley, 2003.

[235] Llanos Tobarra, Antonio Perez Trapero, Rafael Pastor, Antonio Robles-Gomez, Roberto Hernandez, Andres Duque, and Jesus Cano. Game-based learning approach to cybersecurity. *IEEE Global Engineering Education Conference, EDUCON*, 2020-April:1125–1132, 2020.

[236] CTF ToolKit. *A CTF Resource*, 2021 (acedido em Agosto, 2021). Disponível em: https://ctftoolkit.com/.

[237] Tornado. *Python web framework and asynchronous networking library*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.tornadoweb.org/en/stable.

[238] Unreal Tournament. *Unreal Wiki, Unreal Tournament*, 2020 (acedido em Maio, 2020), disponível em: https://unreal.fandom.com/wiki/Unreal_Tournament.

[239] Erik Trickel, Francesco Disperati, Eric Gustafson, Faezeh Kalantari, Mike Mabey, Naveen Tiwari, Yeganeh Safaei, Adam Doupé, and Giovanni Vigna. Shell We Play A Game? CTF-as-a-service for Security Education. *2017 USENIX Workshop on Advances in Security Education (ASE 17)*, 2017.

[240] Carnegie Mellon University. *picoCTF - Let learning happen through exploration*, 2021 (acedido em Agosto, 2021). Disponível em: https://picoctf.org/.

[241] Carnegie Mellon University. *picoGym - Practice Challenges*, 2021 (acedido em Agosto, 2021). Disponível em: https://play.picoctf.org/.

[242] uscybergames. *Home of the US Cyber Team*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.uscybergames.com.

[243] ut. *Master of Science in Cybersecurity - The University of Tampa*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.ut.edu/graduate-degrees/graduate-business/master-of-science-in-cybersecurity.

[244] Vagrantup. *Discovering Vagrant Boxes*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.vagrantup.com/vagrant-cloud/boxes/catalog.

[245] Ashok Vaseashta and Philip Susmann. Cyber Security – Threat Scenarios , Policy Framework and Cyber Wargames Cyber Security – Threat Scenarios , Policy Framework and Cyber Wargames. (October), 2014.

[246] Ashok Vaseashta and Philip Susmann. Cyber Security – Threat Scenarios , Policy Framework and Cyber Wargames Cyber Security – Threat Scenarios , Policy Framework and Cyber Wargames. (October), 2014.

[247] Giovanni Vigna, Kevin Borgolte, Jacopo Corbetta, Adam Doupe, Yanick Fratantonio, Luca Invernizzi, Dhilung Kirat, and Yan Shoshitaishvili. Ten Years of iCTF: The Good, The Bad, and The Ugly. *USENIX Summit on Gaming, Games, and Gamification in Security Education*, 2014.

[248] Vijeta. *What is CTF ?*, 2021 (acedido em Agosto, 2021). Disponível em: https://medium.com/@Blackpear1/what-is-ctf-9c05a45e5bd3.

[249] VulnHub. *VulnHub - Vulnerable By Design*, 2021 (acedido em Agosto, 2021). Disponível em: https://www.vulnhub.com.

[250] Jan Vykopal, Radek Ošlejšek, Pavel Celeda, Martin Vizváry, and Daniel Tovarˇ. KYPO Cyber Range : Design and Use Cases.

[251] Jan Vykopal, Valdemar Svabensky, and Ee Chien Chang. Benefits and Pitfalls of Using Capture The Flag Games in University Courses. *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*, (February):752–758, 2020.

[252] Jan Vykopal, Martin Vizvary, Radek Oslejsek, Pavel Celeda, and Daniel Tovarnak. Lessons learned from complex hands-on defence exercises in a cyber range. *Proceedings - Frontiers in Education Conference, FIE*, 2017-Octob:1–8, 2017.

[253] Richard H Wagner. Designing a network defense scenario using the open cyber challenge platform. *ProQuest Dissertations and Theses*, page 76, 2013.

[254] Joseph Werther, Michael Zhivich, Tim Leek, and Nickolai Zeldovich. Experiences in cyber security education: The MIT Lincoln laboratory capture-the-flag exercise. *4th Workshop on Cyber Security Experimentation and Test, CSET 2011*, 2011.

[255] Joseph Werther, Michael Zhivich, Tim Leek, and Nickolai Zeldovich. Experiences in cyber security education: The MIT Lincoln laboratory capture-the-flag exercise. *4th Workshop on Cyber Security Experimentation and Test, CSET 2011*, 2011.

[256] Xiaojun Wu and Shuzhen Tian. Student-centered learning in cybersecurity in summer semester. *Proceedings - Frontiers in Education Conference, FIE*, 2015:1–4, 2015.

[257] Muhammad Mudassar Yamin, Basel Katt, and Vasileios Gkioulos. Cyber ranges and security testbeds: Scenarios, functions, tools and architecture. *Computers and Security*, 88:101636, 2020.

Apêndices

Esta página foi intencionalmente deixada em branco.

Apêndice A

A Tabela 1 apresenta vários estudos e publicações sobre as experiências vivência-das no ensino da segurança informática, com recurso às competições CTF,

Tabela 1: Publicações no âmbito do tema Cyber Security Education

2007 - Is Attack Better Than Defense? Teaching Information Security the Right Way [177]
2009 - The Design and Lessons of the ASCENT Security Teaching Lab [125]
2010 - The Blunderdome: An Offensive Exercise for Building Network, Systems, and Web Security Awareness [162]
2011 - Experiences In Cyber Security Education: The MIT Lincoln Laboratory Capture-the-Flag Exercise [255]
2011 - The Importance of Using Hacker Contests and Mindset in Teaching Networks and Information Assurance [11]
2011 – Cyber Defense Competition: A table of two teams [15]
2011 - Challenge Based Learning in Cybersecurity Education [40]
2011 - RTFn: Enabling Cybersecurity Education through a Mobile Capture the Flag Client [30]
2011 - A League of Our Own: The Future of Cyber Defense Competitions [165]
2012 - Towards a Practical Approach for Teaching IT-Security [14]
2012 - Effectiveness of Cybersecurity Competitions [41]
2013 - The 2013 NECCDC - Lessons Learned [168]
2013 – Computer Security Competitions: Expanding Educational Outcomes [84]
2013 - A Survey of Cyber Ranges and Testbeds [72]
2014 – Cyber Security - Threat Scenarios, Policy Framework and Cyber Wargames [246]
2014 - Ten Years of iCTF: The Good, The Bad, and The Ugly [247]
2014 - PicoCTF: A Game-Based Computer Security Competition for High School Students [36]
2014 - Increasing Student Intrinsic Motivation And Self-Efficacy Through Gamification Pedagogy [12]
2014 - Learning Obstacles in the Capture The Flag Model [44]
2015 - Learning Cyber Security Through Gamification [23]
2015 - Leveraging Competitive Gamification for Sustainable Fun and Profit in Security Education [67]
2015 - An Offline Capture The Flag-Style Virtual Machine and an Assessment of its Value for Cybersecurity Education [43]
2015 - Ethical Hacking Pedagogy: An Analysis and Overview of Teaching Students to Hack [123]
2015 - Student-centered Learning in Cybersecurity in Summer Semester [256]

Tabela 1 Publicações no âmbito do tema Cyber Security Education

2016 - Cybersecurity Education through Gamification - the CTF Approach [160]
2016 - Towards Effective Cybersecurity Education and Training [16]
2016 - Game Based Cyber Security Training: are Serious Games suitable for cyber security training? [124]
2016 - A Review of Using Gaming Technology for Cyber-Security Awareness [7]
2016 - Analysis and Design of an eLearning Platform Featuring Learning Analytics and Gamification [102]
2016 - Gamification for teaching and learning computer security in higher education [208]
2017 - Capture the Flag for Computer Security Learning [167]
2017 - Capture the Flag Unplugged: An Offline Cyber Competition [96]
2017 - Innovative Pedagogical Approaches to a Capstone Laboratory Course in Cyber Operations [187]
2017 - Conceptual Analysis of Cyber Security Education based on Live Competitions [145]
2017 - Shell We Play A Game? CTF-as-a-service for Security Education [239]
2017 - Cybersecurity Teaching through Gamification: Aligning Training Resources to our Syllabus [117]
2017 - CTF: State-of-the-Art and Building the Next Generation [227]
2017 - SSETGami: Secure Software Education Through Gamification [219]
2017 - Security Scenario Generator (SecGen): A Framework for Generating Randomly Vulnerable Rich-scenario VMs for Learning Computer Security and Hosting CTF Events [210]
2018 – Cyber Attack and defense training: using emulab as a platform [152]
2018 - Design Considerations for Cyber Security Testbeds: A Case Study on a Cyber Security Testbed for Education [100]
2018 - ADLES: Specifying, deploying, and sharing hands-on cyber-exercises [52]
2018 - Challenges Arising from Prerequisite Testing in Cybersecurity Games [223]
2018 – Evaluation of Cyber Defense Exercises Using Visual Analytics Process [193]
2018 – Cybersecurity Education and Training Support System: CyRIS [18]
2018 - Gathering Insights from Teenagers Hacking Experience with Authentic Cybersecurity Tools [221]
2018 - King of the Hill: A Novel Cybersecurity Competition for Teaching Penetration Testing [21]

Tabela 1 Publicações no âmbito do tema Cyber Security Education

2018 - Hackerbot: Attacker Chatbots for Randomised and Interactive Security Labs, Using SecGen and oVirt [209]
2019 - Gamification Techniques for Raising Cyber Security Awareness [207]
2019 - pico-Boo!: How to avoid scaring students away in a CTF competition [199]
2019 - Using CTF Tournament for Reinforcing Learned Skills in Cybersecurity Course [116]
2019 - Reviewing Cyber Security Social Engineering Training and Awareness Programs Pitfalls and Ongoing Issues [5]
2020 - Benefits and Pitfalls of Using Capture the Flag Games in University Courses [251]
2020 - Fostering cybersecurity awareness among computing science undergraduate students: motivating by exemple [55]
2020 - What Are Cybersecurity Education Papers About? A Systematic Literature Review of SIGCSE and ITiCSE Conferences [247]
2020 - Riskio: A Serious Game for Cyber Security Awareness and Education [121]
2021 - Cyber Ranges and TestBeds for Education, Training, and Research [164]
2021 - Down the Rabbit Hole: Fostering Active Learning through Guided Exploration of a SCADA Cyber Range [56]

Esta página foi intencionalmente deixada em branco.

Apêndice B

No presente apêndice enumerados os vários *websites*, utilizados como referência, na construção e na readaptação dos exercícios CTF, para a plataforma construída.

Materiais de estudo e exercícios CTF disponíveis em:

- *CTFtime* - coleção de *writeups* [64];
- *VulnHub* - coleção de máquinas virtuais vulneráveis [249];
- *Over The Wire* - disponibiliza vários cenários e desafios de segurança informática, para um público novato [232];
- *Smash the Stack* - disponibiliza exercícios de segurança informática, que são baseados em vulnerabilidades e incidentes reais [231];
- *We Chall* - disponibiliza vários exercícios e desafios de segurança informática, programação, matemática e ciência [34];
- *Infosec Institute* - disponibiliza cursos práticos com certificação e desafios em áreas como *Ethical Hacking* e *Threat Hunting* [134];
- *Hack this site* - disponibiliza vários exercícios práticos dedicados, partilha de conteúdos e técnicas de segurança informática [233];
- *Kontra OWASP Top 10 for Web* - disponibiliza um conjunto de exercícios para ajudar os programadores a identificarem e a mitigarem as vulnerabilidades de segurança presentes no *OWASP Top 10* [148];
- *PortSwigger Web Security Academy* - trata-se da academia da *PortSwigger*, que disponibiliza materiais didácticos para explicar cada uma das vulnerabilidades [202];
- *Open Web Application Security Project (OWASP)* - *OWASP* disponibiliza vários projetos no âmbito da educação e treino de competências em segurança, como o *Juice Shop* [197] ou o *WebGoat* [198];
- *PocketCTF* - máquina virtual que inclui vários exercícios em *docker*, que podem ser instanciados facilmente em *containers* [142];
- *CTF Toolkit* - descreve vários procedimentos e comandos relevantes para os exercícios CTF [236];
- Practical Pentest Labs - disponibiliza cursos *hands-on* em várias categorias, como *Web* [156];
- Pentester Lab - cursos *hands-on* e certificações de segurança informática, na categoria *Web* [153].

Esta página foi intencionalmente deixada em branco.

Apêndice C

No presente apêndice apresentamos as configurações aplicadas aos *virtualhosts* no servidor *web*.

A Fig.1 apresenta o *virtualhost ctf.dei.uc.pt*, utilizado para expor os serviços da plataforma *ctfd*.

A Fig.2 apresenta o *virtualhost ctfdocs.dei.uc.pt*.

A Fig.3 apresenta o *virtualhost ctfchat.dei.uc.pt*, utilizado para expor os serviços do *Rocket Chat*.

A Fig.4 apresenta o *virtualhost ctfsonar.dei.uc.pt*, utilizado para expor os serviços do *Sonar Qube*.

```
# 1 | ctf | 2021-03-21
<VirtualHost 193.137.203.57:80>
    ServerName ctf.dei.uc.pt
    ServerAlias ctf
    DocumentRoot /home/ctf/ctf.dei.uc.pt/public_html

    ErrorLog /home/ctf/ctf.dei.uc.pt/logs/error_log
    CustomLog /home/ctf/ctf.dei.uc.pt/logs/access_log combined

    <Directory /home/ctf/ctf.dei.uc.pt/public_html>
        Options +FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

     RewriteEngine on
     RewriteCond %{SERVER_NAME} =ctf.dei.uc.pt [OR]
     RewriteCond %{SERVER_NAME} =ctf
     RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=permanent]

</VirtualHost>

# 1 | ctf | 2021-03-21
<VirtualHost 193.137.203.57:443>

    ServerName ctf.dei.uc.pt
    ServerAlias ctf
    DocumentRoot /home/ctf/ctf.dei.uc.pt/public_html

    ErrorLog /home/ctf/ctf.dei.uc.pt/logs/error_log
    CustomLog /home/ctf/ctf.dei.uc.pt/logs/access_log combined

    <Directory /home/ctf/ctf.dei.uc.pt/public_html>
        Options +FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    # Usage:
    # ProxyPass / http://[IP Addr.]:[port]/
    # ProxyPassReverse / http://[IP Addr.]:[port]/
    ProxyPass / http://0.0.0.0:8000/
    ProxyPassReverse / http://0.0.0.0:8000/

    ProxyPass /error/ !
    ErrorDocument 500 /home/ctf/ctf.dei.uc.pt/public_html/500.html

    SSLCertificateFile /etc/letsencrypt/live/ctf.dei.uc.pt/fullchain.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/ctf.dei.uc.pt/privkey.pem
    Include /etc/letsencrypt/options-ssl-apache.conf

</VirtualHost>
~
~
```

Figura 1: Virtualhost do ctf.dei.uc.pt

```
# 2 | ctfdocs | 2021-08-01
<VirtualHost 193.137.203.57:80>
    ServerName ctfdocs.dei.uc.pt
    ServerAlias ctfdocs
    DocumentRoot /home/ctf/ctfdocs.dei.uc.pt/ctfwiki-webroot

    ErrorLog /home/ctf/ctfdocs.dei.uc.pt/logs/error_log
    CustomLog /home/ctf/ctfdocs.dei.uc.pt/logs/access_log combined

    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}

</VirtualHost>

# 2 | ctfdocs | 2021-08-01
<VirtualHost 193.137.203.57:443>

    SSLCertificateFile /etc/letsencrypt/live/ctfdocs.dei.uc.pt/fullchain.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/ctfdocs.dei.uc.pt/privkey.pem
    Include /etc/letsencrypt/options-ssl-apache.conf

    SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown downgrade-
    ServerName ctfdocs.dei.uc.pt
    ServerAlias ctfdocs
    DocumentRoot /home/ctf/ctfdocs.dei.uc.pt/ctfwiki-webroot

    ErrorLog /home/ctf/ctfdocs.dei.uc.pt/logs/error_log
    CustomLog /home/ctf/ctfdocs.dei.uc.pt/logs/access_log combined


# Password Basic Auth
# Font: https://cwiki.apache.org/confluence/display/HTTPD/PasswordBasicAuth

# Option 1: static files
# ~/ctfdocs.dei.uc.pt/public_html/ctfwiki-webroot
    <Directory /home/ctf/ctfdocs.dei.uc.pt/ctfwiki-webroot>
        AllowOverride None
        AuthType basic
        AuthName "private"
        AuthUserFile /home/ctf/.htpasswd
        Require valid-user
    </Directory>

# Option 2: using docker
#    ProxyPass / http://0.0.0.0:4000/
#    ProxyPassReverse / http://ctfdocs.dei.uc.pt/
#    <Location />
#        AllowOverride None
#        AuthType basic
#        AuthName "private"
#        AuthUserFile /home/ctf/.htpasswd
#        Require valid-user
#    </Location>

</VirtualHost>
~
```

Figura 2: Virtualhost do ctfdocs.dei.uc.pt

```
# 3 | ctfchat | 2021-03-22
<VirtualHost 193.137.203.57:80>
    ServerName ctfchat.dei.uc.pt
    ServerAlias ctfchat
    DocumentRoot /home/ctf/ctfchat.dei.uc.pt/public_html

    ErrorLog /home/ctf/ctfchat.dei.uc.pt/logs/error_log
    CustomLog /home/ctf/ctfchat.dei.uc.pt/logs/access_log combined

    <Directory /home/ctf/ctfchat.dei.uc.pt/public_html>
        Options +FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    RewriteEngine on
    RewriteCond %{SERVER_NAME} =ctfchat.dei.uc.pt [OR]
    RewriteCond %{SERVER_NAME} =ctfchat
    RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=permanent]
</VirtualHost>

# 3 | ctf | 2021-03-21
<VirtualHost 193.137.203.57:443>

    ServerName ctfchat.dei.uc.pt
    ServerAlias ctfchat
    DocumentRoot /home/ctf/ctfchat.dei.uc.pt/public_html

    ErrorLog /home/ctf/ctfchat.dei.uc.pt/logs/error_log
    CustomLog /home/ctf/ctfchat.dei.uc.pt/logs/access_log combined

    <Directory /home/ctf/ctfchat.dei.uc.pt/public_html>
        Options +FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    # Certificado
    SSLCertificateFile /etc/letsencrypt/live/ctfchat.dei.uc.pt/fullchain.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/ctfchat.dei.uc.pt/privkey.pem
    Include /etc/letsencrypt/options-ssl-apache.conf

    # 15-06-2021 adicionei estas linhas para habilitar os websockets para a apk android
    # https://docs.rocket.chat/installing-and-updating/manual-installation/configuring-
    RewriteEngine On
    RewriteCond %{HTTP:CONNECTION} Upgrade [NC]
    RewriteCond %{HTTP:Upgrade} =websocket [NC]
    RewriteRule /(.*)           ws://localhost:3000/$1 [P,L]
    RewriteCond %{HTTP:Upgrade} !=websocket [NC]
    RewriteRule /(.*)           http://localhost:3000/$1 [P,L]
    # proxy reverso:
    ProxyPassReverse /          http://localhost:3000/

    # Atenção - no Apache é necessário ativar alguns módulos extra,
    # para que os websockets funcioname correctamente:
       # apt-get update && apt-get install apache2
       # a2enmod proxy_http && a2enmod proxy
       # a2enmod ssl && a2enmod proxy_wstunnel && a2enmod rewrite
</VirtualHost>
~
```

Figura 3: Virtualhost do ctfchat.dei.uc.pt

```
# ******************************************
# Nota importante o proxy reserso para o sonarqube
# só funciona com o seguinte comando:

# The Ubuntu20 operating system rises out of memory exceptions
# sudo sysctl -w vm.max_map_count=262144
# echo "vm.max_map_count=262144" >> /etc/sysctl.conf

# 4 | ctfsonar | 2021-03-21
<VirtualHost 193.137.203.57:443>
    ServerName ctfsonar.dei.uc.pt
    ServerAlias ctfsonar
    DocumentRoot /home/ctf/ctfsonar.dei.uc.pt/public_html

    ErrorLog /home/ctf/ctfsonar.dei.uc.pt/logs/error_log
    CustomLog /home/ctf/ctfsonar.dei.uc.pt/logs/access_log combined

    <Directory /home/ctf/ctfsonar.dei.uc.pt/public_html>
        Options +FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    Include /etc/letsencrypt/options-ssl-apache.conf

    # Usage:
    ProxyPass / http://0.0.0.0:8001/
    ProxyPassReverse / http://ctfsonar.dei.uc.pt/
    #Timeout 2400
    #ProxyTimeout 2400
    #ProxyBadHeader Ignore

    SSLCertificateFile /etc/letsencrypt/live/ctfsonar.dei.uc.pt/fullchain.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/ctfsonar.dei.uc.pt/privkey.pem

</VirtualHost>
~
```

Figura 4: Virtualhost do ctfsonar.dei.uc.pt

Esta página foi intencionalmente deixada em branco.

Apêndice D

No presente apêndice apresentamos alguns dos *mockups* desenvolvidos para a interface *web* da plataforma CTF@DEI, nomeadamente:

- na Fig.5 apresentamos a interface da *Dashboard*, proposta para a plataforma CTF@DEI;
- na Fig.6 apresentamos a interface para a *Coleção de exercícios CTF@DEI*, proposta para a plataforma CTF@DEI;

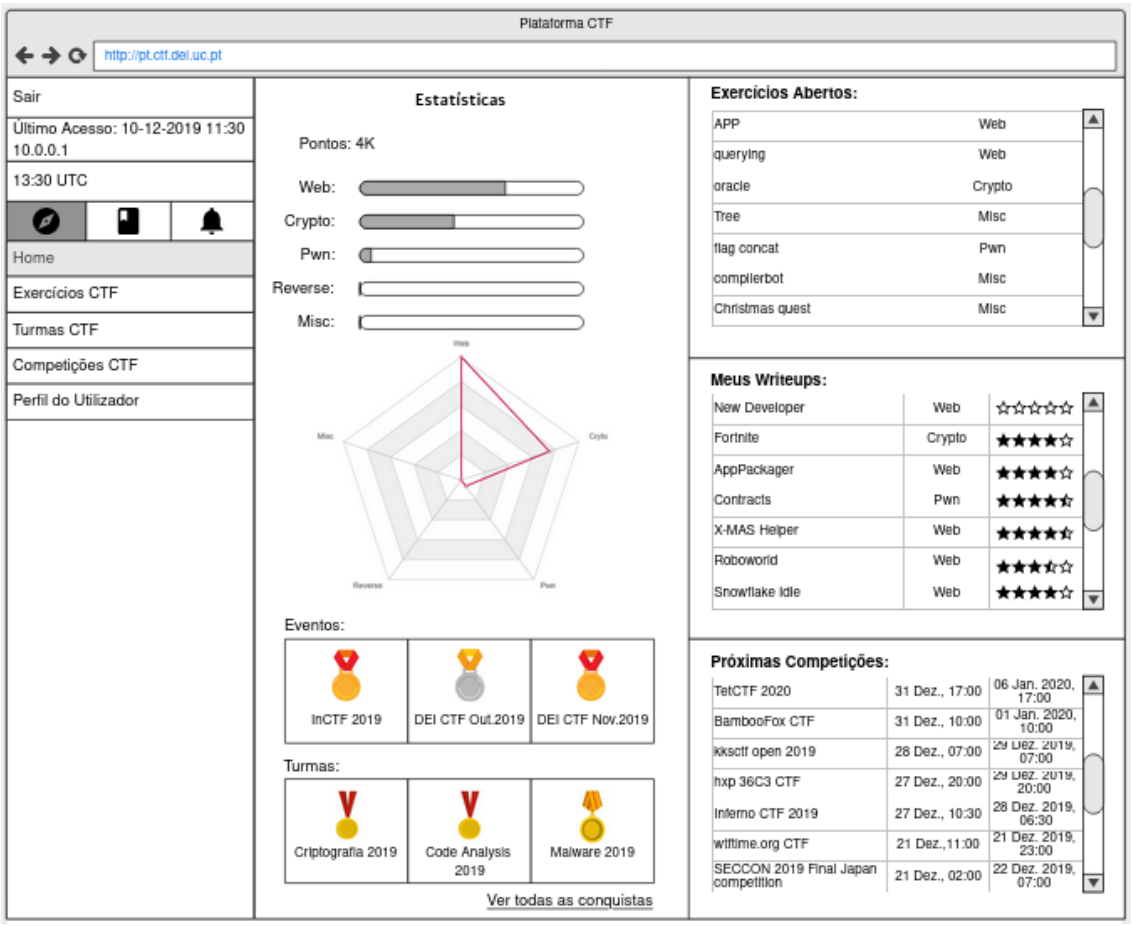

Figura 5: Mockup - Dashboard da plataforma CTF@DEI

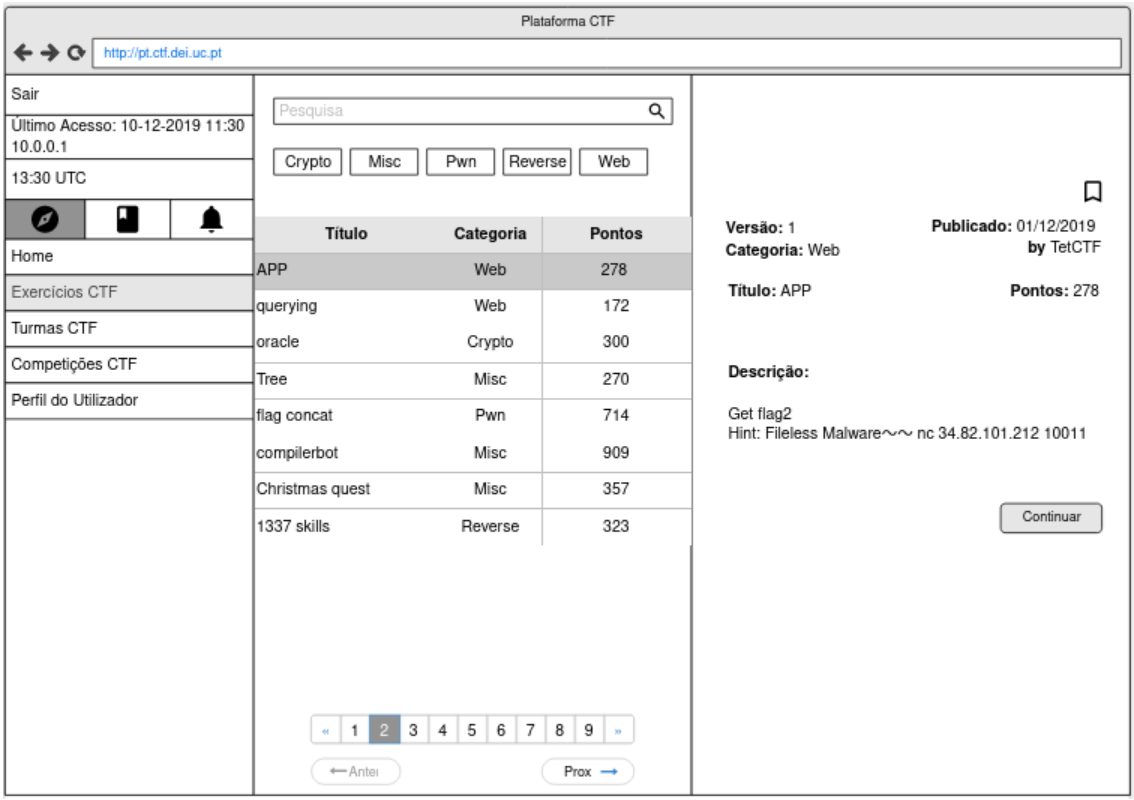

Figura 6: Mockup - Coleção de exercícios CTF@DEI

Apêndice E

No presente apêndice apresentamos o questionário, que foi disponibilizado a seis alunos do curso de mestrado em segurança informática. O objetivo deste questionário foi o de validar/invalidar as hipóteses de trabalho formuladas, e comparar com os resultados recolhidos na revisão de literatura efetuada.

As respostas do presente questionário são formuladas com uma escala de *Likert*, com cinco níveis, em que o nível 1 representa a discordância total, e o nível 5 representa a concordância plena.

Este questionário permite aferir:

- se os conhecimentos prévios e os conhecimentos adquiridos dos cursos de segurança permitem um bom desempenho nos exercícios Capture The Flag (CTF);
- o impato dos exercícios de CTF na motivação dos alunos, na qualidade dos cursos e no desempenho profissional na área da segurança informática;
- a qualidade da plataforma CTF implementada para a dissertação de José Silva, do curso de Mestrado em Segurança Informática (MSI) do Departamento de Engenharia Informática (DEI) da Universidade de Coimbra (UC).

Este questionário foi aplicado a seis alunos do MSI.

As respostas são valoradas de 1-5, sendo que 1 é Discordo totalmente e 5 é Concordo plenamente.

Tópico 1 - Importância dos conhecimentos prévios para a realização dos exercícios CTF

1.1) Os conhecimentos prévios exigidos para a frequência do MSI são suficientes para a resolução dos exercícios CTF?

1.2) O curso de MSI preparou-o para a resolução dos exercícios CTF?

Tópico 2 - A importância dos exercícios CTF para a qualidade do mestrado em segurança informática

2.1) Os exercícios CTF poderão enriquecer o curso do MSI?

2.2) Os exercícios CTF facilitam a aprendizagem na área da segurança informática?

Tópico 3 - A importância dos exercícios CTF na motivação dos alunos do MSI

3.1) Os exercícios CTF aumentaram a motivação para a área de segurança informática?

3.2) Os exercícios CTF poderão contribuir para a melhoria do desempenho académico?

Tópico 4 - A aprendizagem obtida através dos exercícios CTF, e sua utilidade no mercado de trabalho atual na área da segurança informática

4.1) A aprendizagem através dos exercícios CTF facilitará a entrada no mercado de trabalho na área da segurança informática?

4.2) A aprendizagem obtida com os exercícios CTF melhorará o desempenho profissional na área da segurança informática?

Tópico 5 - Acerca da plataforma criada para a dissertação do curso de MSI, para a dissertação de José Silva

5.1) A plataforma tem um bom desempenho?

5.2) A plataforma é apelativa?

Nome:

Número de aluno:

Agradecemos a sua colaboração.

Apêndice F

No servidor *ctfgit.dei.uc.pt/ctf* encontram-se vários repositórios, destacamos os seguintes:

- ctf-setup - Repositório que instala o servidor *web* (composto por *scripts* em *bash*, que reinstalam o *CTFd*, o *Rocket Chat*, o *Sonar Qube* e a *Wiki*), nas mesmas condições);
- ctf-dei - Repositório que instala as plataformas de virtualização *Kubernetes* na Cloud2 do DEI, realizando as modificações apropriadas;
- ctfbox-machine - Repositório que prepara a máquina CTFBOX;
- ctf-collection - Repositório com a Coleção de exercício CTF;
- ctfwiki-webroot - Repositório com as páginas *html* do *website* de documentação (este repositório atualiza a página *ctfdocs.dei.uc.pt*).