UNIVERSIDADE
LUSÓFONA

Monitorizador

Trabalho Final de curso

Relatório Final

Miguel Lourenço, 22202315, LEI
Vasco Pereira, 22202735, LEI

Orientador: Rui Ribeiro
Entidade Externa: CyberS3c

Departamento de Engenharia Informática da Universidade Lusófona
Centro Universitário de Lisboa
27 de junho de 2025
www.ulusofona.pt

Direitos de cópia

Agradecimentos

Gostaríamos de expressar os nossos sinceros agradecimentos à Universidade Lusófona por nos ter proporcionado os recursos e o enquadramento necessários para o desenvolvimento deste Trabalho Final de Curso. Esta etapa representou um marco importante no nosso percurso académico, permitindo-nos consolidar conhecimentos e adquirir experiência prática em contexto real.

Dirigimos um agradecimento especial ao nosso orientador, Professor Rui Ribeiro, pelo acompanhamento atento, pelas sugestões pertinentes e pela disponibilidade demonstrada ao longo de todo o projeto. O seu contributo foi essencial para a concretização deste trabalho.

Agradecemos igualmente à empresa parceira CyberS3c pela colaboração, apoio técnico e confiança depositada em nós. A oportunidade de trabalhar com uma entidade especializada no setor permitiu-nos desenvolver competências valiosas na área da cibersegurança, contribuindo significativamente para o sucesso do projeto.

Por fim, agradecemos às nossas famílias e amigos, pelo constante incentivo, paciência e apoio durante todo este percurso académico.

Resumo

Este projeto propõe a criação e o desenvolvimento de uma API de Threat Intelligence para integração numa plataforma de gestão de ativos, análise de vulnerabilidades e prevenção de ciberataques, em colaboração com a [CyberS3c], que fornecerá os recursos e o apoio técnico necessários para a sua implementação.
O desenvolvimento desta API, com um enfoque mais direcionado ao contexto empresarial, visa fornecer um panorama externo da superfície de exposição da rede da organização. Através desta ferramenta, será possível obter informações detalhadas sobre domínios e endereços IPs, realizar enumeração de subdomínios, identificar portas abertas e serviços expostos, detetar certificados SSL/TLS válidos, expirados ou mal configurados, bem como recolher dados sobre as tecnologias utilizadas nos ativos webs. Adicionalmente, permitirá consultar vazamentos de dados (leaks) associados a e-mails e domínios da organização, recorrendo a fontes como o Leak-Lookup, e integrar com múltiplos feeds de Threat Intelligence, como o OpenCTI, o MISP e o feed da [SegurançaInformática], entre outros. Para além destas, a API contará ainda com muitas outras funcionalidades orientadas para a deteção, monitorização e mitigação de ameaças digitais, proporcionando uma cobertura ampla e evolutiva das necessidades de segurança das organizações.
Estas funcionalidades permitem às organizações obter uma visão completa e atualizada da sua exposição digital externa, facilitando uma resposta mais eficaz a potenciais ameaças.
A colaboração com a [CyberS3c] viabiliza o desenvolvimento de uma solução robusta, que será integrada numa plataforma destinada a reforçar a resiliência organizacional e a proteção contra ataques maliciosos, unindo o conhecimento teórico ao know-how prático no setor da cibersegurança. Além disso, a plataforma será desenvolvida em conformidade com as principais regulações europeias, como a [NIS2] e o [DORA], assegurando o alinhamento com os requisitos legais e normativos em vigor.

Palavras-chave:

- Gestão de Ativos
- Análise de Vulnerabilidades
- Prevenção de Ataques
- Cibersegurança
- Rede Externa/ Interna
- Integridade dos Ativos
- CyberS3c

Abstract

This project proposes the creation and development of a Threat Intelligence API to be integrated into a platform for asset management, vulnerability analysis, and attack prevention, in collaboration with [CyberS3c], which will provide the necessary resources and technical support for its implementation.
The development of this API, with a strong focus on the business context, aims to provide an external overview of the organization's exposure surface. Through this tool, it will be possible to gather detailed information on domains and IP addresses, perform subdomain enumeration, identify open ports and exposed services, detect valid, expired or misconfigured SSL/TLS certificates, and collect data on the technologies used in web assets. Additionally, it will enable the consultation of data leaks associated with corporate emails and domains using sources such as Leak-Lookup, as well as integration with multiple Threat Intelligence feeds, including OpenCTI, MISP, and the feed from [SegurançaInformática], among others. Beyond these, the API will include many other features focused on the detection, monitoring, and mitigation of digital threats, offering a broad and continuously evolving range of cybersecurity capabilities tailored to organizational needs.
These features allow organizations to obtain a complete and up-to-date view of their external digital exposure, facilitating a more effective response to potential threats.
The collaboration with [CyberS3c] enables the development of a robust solution that will be integrated into a platform designed to enhance organizational resilience and protection against malicious attacks, bridging theoretical knowledge with practical expertise in the cybersecurity sector. Furthermore, this solution will be developed in accordance with major European regulations such as [NIS2] and [DORA], ensuring full alignment with current legal and regulatory requirements.

Key-words:

- Asset Management
- Vulnerability Analysis
- Attack Prevention
- Cybersecurity
- External/Internal Network
- Asset Integrity
- CyberS3c

Índice

Lista de Figuras

Lista de Tabelas

Lista de Siglas

ODS	Objetivos de Desenvolvimento Sustentável
API	Interface de Programação de Aplicações
LEI	Licenciatura em Engenharia Informática
TFC	Trabalho Final de Curso

1 Introdução

1.1 Enquadramento

A cibersegurança é, atualmente, uma das maiores preocupações para as empresas. A frequência e a sofisticação dos ciberataques aumentaram exponencialmente nos últimos anos, e um único ataque bem-sucedido pode causar interrupções em larga escala, afetando infraestruturas críticas como energia, saúde, finanças e transportes. Estima-se que o dano causado por estes ataques ultrapassou os cinco mil milhões de euros para as empresas em 2021 e, de acordo com algumas projeções, este valor aumentará a um ritmo de 15% ao ano. A recente mudança de paradigma laboral, impulsionada principalmente pela pandemia da COVID-19, obrigou as empresas a acelerar a sua digitalização, tornando-as mais dependentes de dispositivos digitais e da internet como principal canal de trabalho. O trabalho remoto é hoje uma realidade incontornável, assim como as arquiteturas com serviços cloud.
Esta nova realidade aumenta a exposição a ciberataques, pois acarreta novos riscos e vulnerabilidades, forçando as organizações a reforçarem constantemente as suas defesas.
Paralelamente, surge uma pressão regulatória significativa com diretivas como a [NIS 2] , que exige resiliência e conformidade das infraestruturas críticas. Esta diretiva de segurança foi criada em resposta a um ambiente digital cada vez mais complexo e vulnerável. A plataforma VIRIATUS, idealizada e desenvolvida pela [CyberS3c], oferece uma solução inovadora para os desafios de cibersegurança das empresas. Utilizando inteligência artificial, a plataforma monitoriza, deteta e mitiga riscos em tempo real, além de garantir a conformidade com as normas, como a [NIS 2] . O VIRIATUS foi reconhecido com a nomeação para finalista da 9.ª edição dos [Portugal Digital Awards] , destacando-se como um dos projetos que estão a transformar o paradigma digital em Portugal. Segundo a análise “A visão das empresas portuguesas sobre os riscos”, que auscultou mais de 130 líderes nacionais, os ataques cibernéticos (46%) são uma das principais preocupações dos gestores, superando a instabilidade política e a inflação. O VIRIATUS está em constante evolução para oferecer uma plataforma de cibersegurança cada vez mais robusta e adaptada aos desafios modernos.

1.2 Motivação e Identificação do Problema

No cenário atual da cibersegurança, empresas e organizações enfrentam uma ameaça cada vez mais sofisticada. Ciberataques avançados, como ransomware e ataques de *zero-day*, multiplicam-se em complexidade e frequência, obrigando as organizações a priorizar a segurança dos seus sistemas. Simultaneamente, a pressão regulatória aumenta, com diretivas como a [NIS 2] , que impõe requisitos rigorosos de resiliência e conformidade às infraestruturas consideradas críticas.
Cumprir a diretiva [NIS 2] é fundamental para as empresas e organizações, especialmente para aquelas que operam em setores críticos. Além de reforçar a proteção das infraestruturas, mitigar riscos operacionais e financeiros, e fortalecer a confiança dos *stakeholders*, o não cumprimento da [NIS 2] pode levar a sanções

financeiras severas e até à suspensão das atividades da empresa. A adesão a estas diretrizes demonstra compromisso com as normas legais e evita penalizações que podem prejudicar o bom funcionamento dos negócios.
Este contexto obriga as empresas a integrar eficazmente a monitorização e a resposta a possíveis ameaças, mantendo a conformidade com as diretivas e sem comprometer a agilidade operacional. Adicionalmente, as empresas dependem cada vez mais de infraestruturas digitais externas para o desempenho das suas atividades. Por isso, uma gestão eficiente dos ativos e uma identificação rápida das vulnerabilidades tornaram-se fatores cruciais para garantir a continuidade do negócio e proteger informações sensíveis.
Neste contexto, este projeto propõe o desenvolvimento de uma API de Threat Intelligence. Esta API permitirá às organizações obter uma visão completa e atualizada da sua exposição digital externa, facilitando uma resposta mais eficaz a potenciais ameaças e fornecendo um panorama geral da sua superfície de ataque. Combinando fontes públicas e APIs externas, este projeto visa não apenas criar uma ferramenta eficiente, mas também auxiliar as organizações a dar resposta aos desafios impostos pelas crescentes exigências do mercado e pelas regulamentações em vigor.

1.3 Objetivos

O objetivo principal deste projeto é o desenvolvimento de uma API de Threat Intelligence para futura integração na plataforma de cibersegurança VIRIATUS. O foco específico reside no varrimento externo de ativos digitais, permitindo aos utilizadores obter uma visão abrangente da superfície de exposição online da sua organização.

Através da análise automatizada de domínios e IPs, a API oferece funcionalidades cruciais, como:

- Mapeamento de subdomínios a partir de um domínio principal;
- Recolha e análise de certificados TLS/SSL;
- Verificação de domínios e IPs em *blacklists*;
- Identificação das tecnologias utilizadas nos serviços expostos;
- Listagem de CVEs (Common Vulnerabilities and Exposures) associadas a serviços e versões detetadas;
- Deteção de portas abertas;
- Análise de cabeçalhos HTTP e possíveis configurações inseguras.

Adicionalmente, a API integra Indicadores de Compromisso (IoCs) provenientes da plataforma OpenCTI , MISP e do *feed* da Segurança Informática. Esta integração enriquece a análise com IoCs validados e contextualizados, reforçando a capacidade de deteção de ameaças emergentes.

Esta API alinha-se com a visão estratégica da plataforma VIRIATUS, concebida para oferecer visibilidade total da superfície externa e interna das organizações. O VIRIATUS permite realizar análises avançadas e proativas, identificar automaticamente ativos externos, detetar vulnerabilidades e riscos de exposição, e correlacionar ameaças em

tempo real, com o objetivo de antecipar ações maliciosas antes que estas se concretizem.

A API é modular e extensível, integrando múltiplas APIs externas especializadas em ciberinteligência. Este *design* permite a recolha de dados provenientes de fontes diversas e complementares, que são automaticamente tratados, organizados e enriquecidos dentro da própria API. Isso garante coerência, relevância e contexto às informações apresentadas ao utilizador, permitindo uma análise rápida, eficaz e acionável sobre a postura de segurança externa da organização.

É importante salientar que, embora esta API de Threat Intelligence tenha sido desenvolvida para uma futura integração na plataforma VIRIATUS, essa integração não faz parte dos requisitos deste projeto. A responsabilidade pela integração técnica com a VIRIATUS será inteiramente assumida pela CyberS3c, de acordo com o seu plano interno e os requisitos específicos da referida plataforma. Este projeto tem como foco exclusivo o desenvolvimento da API como um componente independente, garantindo elevados padrões de eficiência, segurança e preparação para integração futura.

Apesar de esta fase do projeto estar centrada no desenvolvimento da API de varrimento externo, a plataforma VIRIATUS contará igualmente com um componente de varrimento interno. Este componente está atualmente a ser desenvolvido pela empresa CyberS3c, o que permitirá oferecer uma cobertura de segurança completa, tanto do ponto de vista externo como interno. Assim, reforça-se ainda mais a capacidade de deteção, mitigação e resposta a ciberameaças. O VIRIATUS posiciona-se, portanto, como uma ferramenta essencial para reforçar a resiliência cibernética das organizações, garantindo uma visibilidade total do seu ciberespaço e a proteção contínua das suas infraestruturas críticas.

1.4 Estrutura do Documento

O presente documento estrutura-se da seguinte forma:

- Na 1º Secção é apresentada a Introdução e enquadramento do projeto
- Na 2º Secção é apresentada a análise da viabilidade e pertinência do projeto
- Na 3º Secção é apresentada a Especificação e Modelação do projeto
- Na 4º Secção é apresentada a Solução Desenvolvida pelo projeto
- Na 5º Secção é apresentada o Método e Planeamento do projeto
- Na 6º Secção é apresentada a validação e Testes
- Na 7º Secção é apresentado o Resultados
- Na 8 º Secção é apresentado a Conclusão
- No Anexo A é apresentado os Testes

2 Pertinência e Viabilidade

2.1 Pertinência

Este projeto foi desenvolvido para responder à crescente necessidade de proteger as infraestruturas externas das empresas contra ciberataques. O foco é especial em setores críticos como saúde e transportes, onde os custos de recuperação são significativamente mais elevados e as diretivas regulatórias são mais rigorosas. Estudos recentes, como os realizados pela CyberS3c, demonstram que dois terços das organizações foram alvo de ataques nos últimos 12 meses. Este dado não só sublinha a frequência dos ataques, mas também a lacuna existente na capacidade de contrarresposta e prevenção das empresas, conforme ilustrado na **Figura 1** – *Estudo se empresas foram alvos de ataques*.

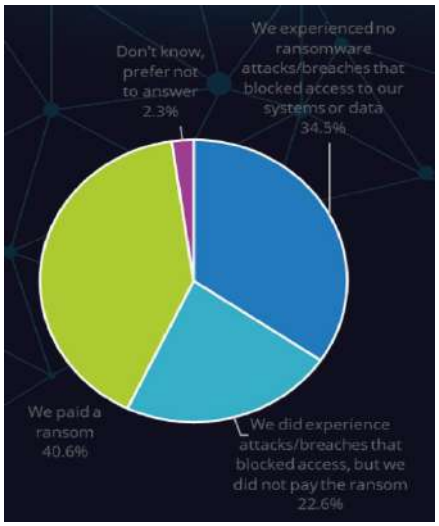


Figura 1-Estudo se empresas foram alvos de ataques

Neste contexto, a plataforma VIRIATUS apresenta-se como uma resposta inovadora aos desafios atuais de cibersegurança, disponibilizando uma solução integrada que vai além das abordagens tradicionais. O elemento central do presente projeto é a criação de uma API dedicada à análise da infraestrutura externa, concebida para ser o motor de deteção e gestão de riscos digitais.

Esta API permitirá realizar varrimentos automatizados para descobrir ativos expostos, identificar vulnerabilidades conhecidas (CVEs) em serviços acessíveis publicamente, monitorizar certificados digitais e analisar a reputação de domínios e endereços IP associados à organização. Esta componente será essencial para o processamento contínuo de *feeds* de informação externos, constituindo o núcleo da capacidade de análise preditiva e reativa da solução.

A pertinência desta abordagem torna-se evidente face à crescente complexidade das infraestruturas digitais. A migração para a *cloud*, o aumento do trabalho remoto e a interligação com parceiros externos ampliam significativamente a superfície de ataque. Configurações incorretas, falhas em serviços *web* e APIs públicas expõem as organizações a riscos que exigem monitorização constante. A análise pontual e centrada no perímetro interno já não é suficiente.

Perante este cenário, a API destaca-se como uma ferramenta essencial para permitir às organizações detetar fragilidades antes que sejam exploradas, reforçar a sua postura de

segurança e cumprir requisitos regulatórios, nomeadamente nos setores críticos. A sua integração futura com a plataforma VIRIATUS garante uma visão unificada e acionável do risco externo, promovendo uma cibersegurança mais robusta e preventiva.
Este projeto ganha ainda mais relevância ao considerar que cerca de 60% das PME encerram atividade até seis meses após um ataque grave. Uma ferramenta como esta pode representar a diferença entre a resiliência e o colapso.
Este impacto é reforçado pela colaboração com a CyberS3c, cuja experiência na área de cibersegurança ajudará a garantir que a API desenvolvida adote as melhores práticas do setor e responda com eficácia às exigências do mercado e à modernização dos ataques.

2.2 Viabilidade

A viabilidade do projeto é assegurada por uma análise abrangente que contempla fatores técnicos, económicos, sociais e ambientais, garantindo a sua implementação e sustentabilidade enquanto projeto académico.
O projeto alinha-se diretamente com os Objetivos de Desenvolvimento Sustentável (ODS), nomeadamente com o [ODS9] (Indústria, Inovação e Infraestruturas). Ao propor uma solução inovadora que reforça a robustez das infraestruturas das empresas, o projeto assume um papel fundamental no funcionamento seguro e eficiente das organizações no mundo digital atual. Adicionalmente, há um alinhamento com o [ODS16] (Paz, Justiça e Instituições Eficazes), uma vez que o projeto melhora a proteção de infraestruturas e previne ciberataques que podem comprometer a estabilidade operacional, especialmente em organizações que atuam em áreas críticas. Consequentemente, contribui para a melhoria da proteção de dados e da segurança digital.
No plano técnico, a viabilidade do projeto é sustentada pela vasta experiência da CyberS3c na implementação de soluções similares, o que comprova a adequação das ferramentas e tecnologias a utilizar. Este conhecimento, aliado ao suporte técnico especializado e aos recursos disponibilizados pela CyberS3c, garante uma base sólida para o desenvolvimento e a evolução contínua do projeto.
Economicamente, a solução desenvolvida apresenta-se como acessível e sustentável. Baseada num plano de subscrição já definido e precificado, garante a sua sustentabilidade e o potencial de faturação para a empresa. Este modelo de negócios, aliado à crescente procura por soluções de cibersegurança, proporciona uma alternativa de custo-benefício significativa para as empresas, uma vez que permite a redução de despesas associadas a ciberataques e a prevenção de vulnerabilidades.
De acordo com a Internacional Data Corporation (IDC), o investimento em cibersegurança em Portugal deverá atingir 250 milhões de euros até ao final de 2024, representando um crescimento de 15% em relação ao ano anterior. Esta tendência é ilustrada na Figura 2-Previsão do valores de cibersegurança. Esse aumento reflete a crescente preocupação das empresas com a proteção de dados e a conformidade com regulamentos como a diretiva NIS2, o que demonstra um mercado em expansão para soluções de segurança como a da CyberS3c. Além disso, estudos da [IBM] indicam que o custo médio de uma violação de dados pode ser superior a 4 milhões de dólares, tornando a implementação de plataformas como esta uma prioridade económica para as empresas.

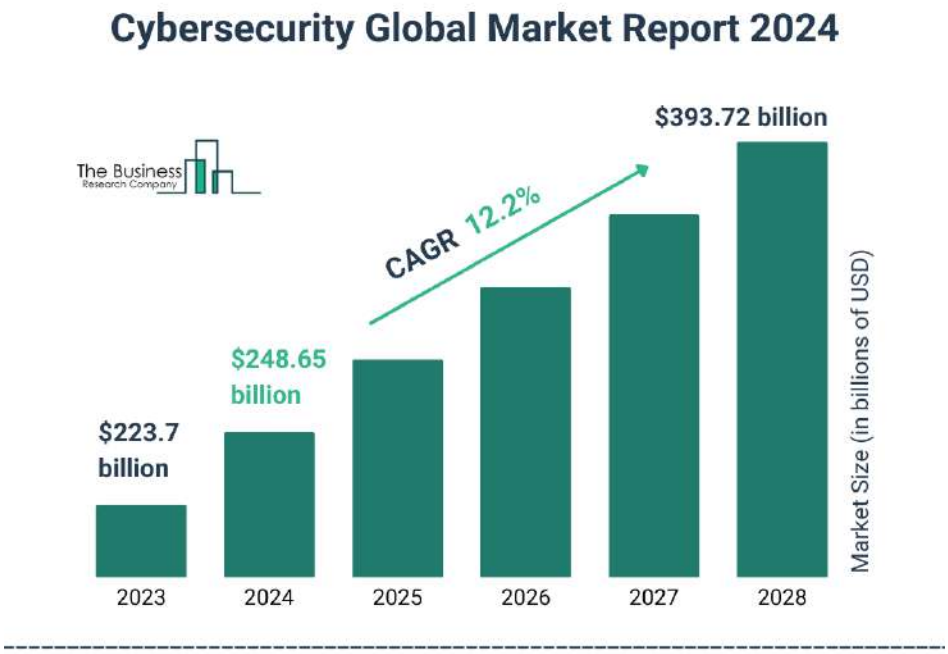


Figura 2-Previsão do valores de cibersegurança

A nível global, o mercado de tecnologia de cibersegurança registou um [crescimento anual] de 11,6% no segundo trimestre de 2023, atingindo os 19 mil milhões de dólares. Este dado reitera que o investimento em cibersegurança se tornou uma prioridade máxima para as organizações, conforme evidenciado na Figura 3-Mercado de tecnologia de cibersegurança

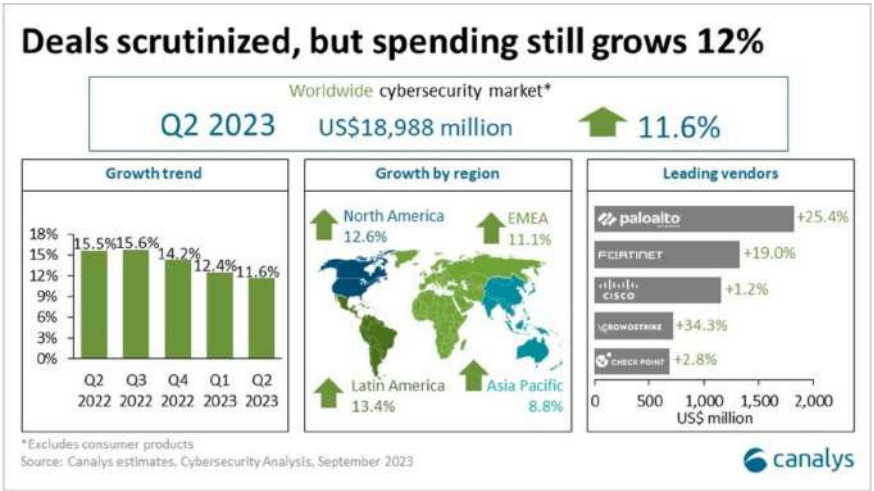


Figura 3-Mercado de tecnologia de cibersegurança

Já a nível nacional, dois terços das Pequenas e Médias Empresas (PME) estão a investir ativamente em medidas de cibersegurança. Segundo o jornal [Expresso] , 37% das empresas preveem gastar até 30 mil euros em soluções de proteção digital nos próximos

12 meses, sendo que este número tende a aumentar com a crescente ameaça. Segundo a [Check Point] , uma empresa portuguesa é atacada, em média, 565 vezes por semana um número significativamente superior à média europeia. Estes dados refletem que a plataforma está, portanto, estrategicamente posicionada para atender à crescente pressão das empresas nacionais para adotarem soluções de cibersegurança eficazes e preventivas, o que fortalece a sua viabilidade económica.

A nível social, o projeto visa o contexto empresarial, focando-se na gestão de ativos e na prevenção de ciberataques, alinhando-se com as exigências da Diretiva [NIS 2] e do Regulamento [DORA], que são obrigatórias para todas as empresas de setores críticos.

Desta forma, o projeto não só se posiciona como uma solução técnica, social e economicamente viável, mas também como uma ferramenta com potencial para gerar um impacto positivo significativo no setor da cibersegurança empresarial.

2.3 Análise Comparativa com Soluções Existentes

2.3.1 Soluções existentes

Neste capítulo, apresentamos uma análise comparativa de três soluções existentes no mercado que possuem funcionalidades semelhantes ou complementares à nossa proposta. O objetivo é destacar as suas principais diferenças e semelhanças, contextualizando a inovação e o posicionamento do nosso projeto. As soluções analisadas são:

- **Web Check** O Web Check é uma ferramenta OSINT (*Open Source Intelligence*) desenvolvida com o objetivo de fornecer uma visão aprofundada sobre o funcionamento interno de um *website*. Disponível em formato API, esta solução apresenta uma variedade de informações técnicas relevantes, como dados de IP, cadeia de certificados SSL, registos DNS, *cookies*, *headers*, detalhes do domínio, regras de rastreamento (*crawl rules*), localização do servidor, portas abertas, *traceroute*, desempenho do *website* e até a sua pegada de carbono digital. Para além da API, o Web Check disponibiliza uma interface *web* bastante interativa, que facilita a visualização e análise das informações recolhidas. Esta ferramenta é totalmente *open source* e encontra-se disponível no GitHub.

- **Shodan** O Shodan é um motor de busca, disponível no formato de API, que possibilita uma monitorização mais aprofundada da exposição da rede e investigações detalhadas. O Shodan permite visualizar portas abertas, serviços expostos, vulnerabilidades associadas e *banners* de serviços. É uma ferramenta bastante útil para a análise de superfícies de ataque e para varreduras de infraestrutura exposta.

- **SecurityTrails** O SecurityTrails oferece uma API OSINT poderosa e robusta, focada em fornecer uma visão abrangente da infraestrutura digital de qualquer domínio ou endereço IP. Ao integrar várias fontes de dados públicas e privadas,

o SecurityTrails disponibiliza funcionalidades como histórico DNS e WHOIS, enumeração de subdomínios, contexto geográfico e de propriedade de IPs, certificados SSL e registos MX.

2.3.2 Análise de benchmarking

Tabela 1-Benchmark

Características	Web Check	Shodan	SecurityTrails API	Viriatus (API externa)
API disponível	X	X	X	X
Interface Web	X			
Open Source	X			
Dados de IP	X	X	X	X
Certificados SSL	X		X	X
Registos DNS	X	X	X	X
Histórico DNS			X	X
Enumeração de subdomínios			X	X
Informação sobre subdomínios				X
Localização do servidor	X	X	X	
Verificação de portas abertas	X	X		X
Análise de serviços expostos		X		X
Banners de serviços		X		X
Identificação de vulnerabilidades por CVEs		X		X
Pesquisa de CVEs por software específico				X
Verificação de dataleaks associados a IP/domínio				X
Resolução inversa de DNS		X		X
Análise de ASN		X	X	X
Registos WHOIS			X	
Registos MX			X	
Fingerprinting de serviços/sistemas operativos		X		X

Headers HTTP analisados	X			X
Cookies analisados	X			X (detalhado)
Verificação de headers de segurança				X
Verificação de má configuração				X
Deteção de páginas administrativas expostas				X
Tecnologias Web / CMS identificadas				X
Deteção de software desatualizado / obsoleto				X
Presença de Web Application Firewall (WAF)				X
Redireccionamentos suspeitos				X
Segurança de iframes				X
Formulários HTTP inseguros				X
robots.txt / Crawl rules	X			
Traceroute	X			X
Pegada de carbono digital do website	X			
Visualizações gráficas	X			
Exportação de dados JSON	X	X	X	X
Autenticação necessária	X	X	X	
Limites de requisições	Médio	Alto	Médio	Médio
Versão gratuita disponível	X	X	Limitada	
Facilidade de integração	Média	Alta	Alta	Alta
Foco na superfície de ataque externa	X	X	X	X
Integração do Opencti				X

100% Portuguesa				X

Em suma, a API destaca-se como a solução mais completa e inovadora para a monitorização da superfície de ataque externa, apresentando uma maior cobertura de funcionalidades.
A API diferencia-se, ainda, pela verificação de *data leaks*, pela pesquisa de CVEs por software específico e pela pesquisa de CVEs genéricas.
A sua elevada facilidade de integração, aliada ao facto de ser uma plataforma 100% portuguesa, representa uma vantagem estratégica significativa para entidades nacionais. Esta vantagem manifesta-se quer em termos de suporte técnico, quer na adaptação à realidade regulatória local, nomeadamente na conformidade com a diretiva NIS2.

2.4 Proposta de inovação e mais-valias

A solução apresenta-se como uma proposta inovadora na área da Cibersegurança, sendo o desenvolvimento da API o elemento central e diferenciador de todo o projeto. A API estrutura resposta a ameaças, através da recolha, integração e análise de dados críticos, permitindo uma atuação proativa face aos riscos da rede externa de uma empresa Esta API permite a integração e a coleta automatizada de dados provenientes de diversas fontes, como feeds de ameaças, vulnerabilidades conhecidas (CVEs), informações sobre domínios, subdomínios e endereços IP. Estes dados são enviados diretamente para a plataforma, permitindo uma análise mais robusta e uma visão alargada e integrada do panorama de ameaças, sendo a base para uma resposta mais eficaz. A API facilita a interoperabilidade entre diferentes ferramentas de segurança já existentes na infraestrutura do cliente, aumentando a eficiência e reduzindo redundâncias permitindo uma maior agilidade na tomada de decisões e maior proteção contra vulnerabilidades e ameaças emergentes.

Entre as funcionalidades específicas da API, destacam-se:

- **Varrimento de Domínios e Subdomínios**: Realiza a varredura completa de domínios e dos seus respetivos subdomínios.
- **Varrimento de Endereços IP Externos**: Efetua a varredura de endereços IP externos pertencentes às organizações.
- **Pesquisa e Análise Centralizada de CVEs**: Permite pesquisar e analisar vulnerabilidades e exposições comuns (CVEs) de forma centralizada.
- **Recolha Automatizada e Integração de *Feeds* de *Threat Intelligence***: Garante a recolha automática e a integração de dados de inteligência de ameaças.
- **Busca por Ciberataques a um País Específico num Determinado Período**: Permite procurar ciberataques direcionados a um país num período de tempo definido.
- **Pesquisa sobre *Data Leaks***: Realiza buscas por informações relacionadas com vazamentos de dados.

Em termos de impacto social, tanto a plataforma Viriatus como a API contribuem de forma significativa para uma sociedade digital mais segura. Ao reforçar a segurança das infraestruturas empresariais e garantir a proteção de dados sensíveis, tornam-se peças-chave no combate à ciberataques.

No plano da sustentabilidade, a API assume também um papel relevante, ao permitir uma operação mais segura e eficiente das infraestruturas externas das empresas, reduzindo o risco de ciberataques e os custos associados. O seu custo-benefício é altamente favorável, face ao impacto potencial dos ataques que ajuda a prevenir.

No contexto da parceria, a colaboração com a CyberS3c traz diversas mais-valias. Como empresa especialista em cibersegurança, a CyberS3c fortalece a implementação da solução, fornecendo apoio técnico. Para o parceiro, a solução melhora a oferta de produtos e serviços, mas também contribui para reforçar a imagem da empresa na área de cibersegurança, em particular ao oferecer uma solução em conformidade com as últimas exigências regulatórias. Vai possibilitar também uma entrada num mercado em crescimento, devido à elevada demanda por soluções que resolvam o problema de cibersegurança do setor empresarial e infraestruturas de risco.

2.5 Identificação de oportunidade de negócio

O projeto apresenta uma oportunidade de negócio significativa, dado que, segundo a [Cybersecurity Ventures], os ciberataques, se fossem uma economia, representariam a terceira maior do mundo. Este facto evidencia a sua natureza crescente e contínua. No cenário atual, as empresas estão cada vez mais focadas em reforçar os seus sistemas de cibersegurança, especialmente face às exigências legais impostas a todas as empresas de infraestruturas críticas na União Europeia, com destaque para a aplicação da Diretiva NIS2.Conforme ilustrado na Figura 4-Economia anual mundial**.**

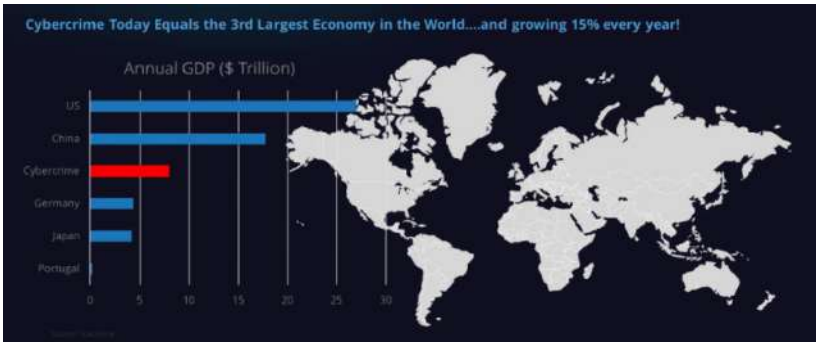


Figura 4-Economia anual mundial

Neste contexto, a exploração comercial deste projeto pode ser realizada de diferentes formas, tirando partido da necessidade crescente do mercado por soluções eficazes, acessíveis e adaptáveis. É importante ainda referir que o mercado europeu de cibersegurança está em expansão, com previsões que apontam para um crescimento anual médio de 10,3%, estimando-se que atinja cerca de 65 mil milhões de euros até 2028 ([Statista Research Department]). Estes dados confirmam não só o aumento da procura, mas também o espaço disponível para novas soluções tecnológicas e inovadoras.

Uma das principais oportunidades reside em oferecer soluções a pequenas e médias empresas (PMEs). Estas empresas frequentemente enfrentam desafios em termos de recursos para implementar as suas próprias medidas de segurança cibernética. A nossa plataforma visa combater esta lacuna, uma vez que oferece uma solução acessível e escalável, compatível com qualquer equipamento e fabricante, o que reduzirá também os custos de implementação. A Figura 5-Causas do Não Investimento em Cibersegurança contextualiza a necessidade.

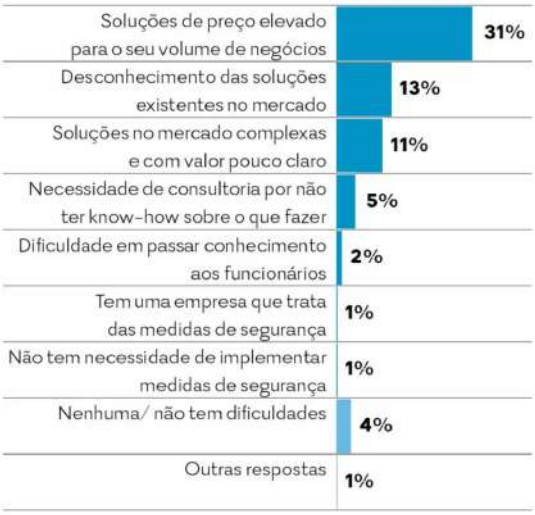


Figura 5-Causas do Não Investimento em Cibersegurança

Além disso, a disponibilização do VIRIATUS, onde a API de monitorização de fontes externas estará inserida, apresentará um modelo de subscrição. Este modelo permite às empresas aceder à plataforma e aderir ao plano que vá ao encontro das suas necessidades de utilização, permitindo uma receita recorrente e uma atuação contínua da plataforma. O modelo de precificação do VIRIATUS é apresentado na Figura 6-Precificação do Viriatus

Figura 6-Precificação do Viriatus

Em termos de expansão, a plataforma tem um grande potencial para ser expandida para mercados a nível internacional, especialmente dentro da União Europeia. Aqui, as exigências de conformidade com a regulamentação de segurança são cada vez mais críticas, impulsionando uma procura crescente por soluções deste tipo.

3 Especificação e Modelação

3.1 Análise de Requisitos

Este Capítulo tem como propósito a análise dos requisitos identificados pelo grupo para o sucesso do projeto. Os requisitos apresentados pretendem resumir as funcionalidades e as necessidades da aplicação. Para o projeto, os requisitos foram levantados ao longo de várias reuniões com o Professor Rui Ribeiro e com a CyberS3c que nos forneceram o material necessário para o desenvolvimento os mesmos.

Os requisitos são definidos em dois grupos:

- **Requisitos Funcionais –** Descrição das funções que oferecem valor aos utilizadores
- **Requisitos Não Funcionais**– Definem restrições sobre o projeto ou a execução, tais como requisitos de desempenho, segurança.

3.1.1 Enumeração de Requisitos

Tabela 2-Requisitos Funcionais

ID do Requisito	Descrição do Requisito	Prioridade/Impacto	Critérios de Aceitação
REQ-01 Instalação do MISP numa Máquina Virtual de Testes	Instalar e configurar a plataforma MISP (Malware Information Sharing Platfotm) numa Máquina Virtual de Testes de modo a configurar um sistema de alertas de threat intelligence.	Médio/Médio	1 – A máquina virtual deve estar funcional com o MISP instalado. 2 - Deve ser possível aceder ao MISP via browser local 3 – Deve ser possível criar uma conta de admin e visualizar as informações dos feeds já instalados com o MISP
REQ-02 Instalação do OpenCTI numa Máquina Virtual de Testes	Instalar e configurar o OpenCTI (Open Cyber Threat Intelligence) numa Máquina Virtual de Testes para configurar uma plataforma de threat intelligence para ingestão e análise de dados.	Médio/Médio	1 – A máquina virtual deve estar funcional com o OpenCTI instalado. 2 – Deve ser possível aceder ao OpenCTI via browser local

REQ-03 Integração e Expansão de fontes publicas no OpenCTI	Integrar fontes públicas de Threat Intelligence no OpenCTI, incluindo fontes novas como AlientVault, MalwareBazzar, AbuseIPDB e ThreatMiner.	Médio/Médio	1 – As fontes públicas devem estar integradas com o OpenCTI via conectores 2 – Os dados destas fontes devem ser visíveis na timeline da interface web do OpenCTI, para consulta e análise. 3 – Os dados devem ser possíveis de atualizar por execução manual ou por cron jobs. 4 – Os logs de ingestão devem confirmar a entrada de fontes
REQ-04 Integração do feed CIRCL OSINT na plataforma MISP	Integrar o feed CIRCL OSINT na plataforma MISP para ingestão automática de indicadores. Deve ainda ser criada uma cron job para atualização automática da plataforma a cada 24 horas	Baixa/Baixo	1 – O CIRCL OSINT deve estar integrado com o OpenCTI via conector. 2 – Os dados do CIRCL OSINT devem ser visíveis na timeline da interface web do OpenCTI, para consulta e análise. 3 – Os dados devem ser possíveis de atualizar através da cron job criada. 4 – A cada 24 horas deve ser visível a atualização dos dados provenientes deste feed 5 – Os logs de ingestão devem confirmar a entrada do feed e da atualização do cron job.
REQ-05 Integração do MISP no OpenCTI	Estabelecer a integração bidirecional entre o MISP e OpenCTI para troca contínua de indicadores, o conector deve funcionar automaticamente e os dados de eventos criados no MISP devem aparecer no OpenCTI	Médio/Médio	1 – O MISP deve estar integrado com o OpenCTI via conectores 2 – Os eventos criados no MISP devem aparecer automaticamente no OpenCTI. 3 – Os logs de ingestão devem confirmar a entrada de dados provenientes do MISP.

REQ-06 Limpeza e Operacionalização do script Monitorizador fornecido pela Cybers3c	Rever, limpar, otimizar e colocar o script do Monitorizador da Cybers3c em funcionamento, corrigindo dependências, bugs, bibliotecas desatualizadas e organizar o código de forma modular.	Alta/Alto	1 - As funcionalidades previamente existentes, como a listagem de subdomínios, análise de headers, entre outras, devem estar operacionais. 2 - O script não deve estar conectado a nenhuma base de dados. 3 - O código não deve conter bugs, nem dependências desatualizadas. Todas as dependências utilizadas devem estar listadas de forma clara no ficheiro requirements.txt. 4 - O script deve implementar tratamento de exceções adequado para erros comuns (ex: falhas de rede, timeouts, inputs inválidos). 5 - O código deve estar modularizado, com funções separadas por responsabilidade, e organizado de forma limpa e legível. 6 - Problemas de desempenho que causavam timeouts devem estar corrigidos, garantindo a execução completa do script em tempo útil.
REQ-07 Integração de APIs de Infraestrutura, Vulnerabilidades e Leaks	Integrar no script Monitorizador fontes de inteligência que fornecem dados sobre infraestrutura exposta, urls de phishing, leaks de credenciais, exploits conhecidos e consulta de ip´s em blacklists. Algumas destas APIs são: Shodan,	Alta/Alto	1 – As APIs externas devem estar corretamente integradas, deixando as outras funcionalidades do monitorizador livres de erros 2 - O script deve retornar dados limpos e válidos de cada uma das fontes num formato estruturado (JSON) 3 – Erros de comunicação com as APIs devem ser corretamente tratados

	Leak-Lookup, ExploitDB e BlackListChecker		
REQ-08 Integração de APIs de Análise de Domínios e Segurança Web	Integrar no script Monitorizador serviços que permitam avaliar a reputação, segurança e estados de configuração de domínios através de pedidos HTTP. Algumas destas fontes incluem: URLscan.io, Mozzila HTTP Observatory, VirusTotal, Web-Check e Security-Trails	Alta/Alto	1 – As APIs externas devem ser corretamente integradas, deixando as outras funcionalidades do monitorizador livres de erros 2 – O script deve retornar dados limpos e válidos de cada uma das fontes num formato estruturado (JSON) 3 – Erros de comunicação com as APIs devem ser corretamente tratados.
REQ-09 Adaptação do script Monitorizador como uma API Restful	Desenvolver uma API (RESTful) utilizando a framework Flask que retorne o output do script Monitorizador, com rotas específicas e segura, numa resposta JSON. Criação de dois endpoints: um para o varrimento de domínios e subdominios e outro para o varrimento de ip´s	Alta/Alto	1 - A API deve disponibilizar dois endpoints funcionais e documentados: • POST /monitorizador/dom • POST /monitorizador/IP 2 - A resposta retornada por cada endpoint deve ser equivalente ao output original do script Monitorizador, mantendo a estrutura de dados e formato JSON. 3 - A API deve implementar mecanismos robustos de tratamento de erros e validação de inputs, assegurando que entradas inválidas não causam falhas de execução.

			4 - A solução deve ser validada através de testes funcionais utilizando o Postman ou em scripts de teste desenvolvidos
REQ-10 Implementação do OpenCTI na DigitalOcean	Configurar e disponibilizar a instância do OpenCTI na plataforma de cloud DigitalOcean garantindo acesso externo à máquina.	Médio/Médio	1 – O OpenCTI deve ser acessível externamente através de outra máquina via browser. 2 – Todas as funcionalidades disponíveis na instância local do OpenCTI devem estar igualmente operacionais na instância implementada na cloud.
REQ-11 Implementação de um feed de IoC´s	Construção de dois scripts: IoC's OpenCTI+MISP IoC's OxSI_f33d	Médio/Médio	1 - Scripts devem ser capazes de: • Extrair IPs, domínios, hashes da API do OpenCTI. • Extrair IoCs por país. • Retirar IoCs do feed OxSI_f33d por intervalo de tempo. 2 - Dados devem estar formatados corretamente.
REQ-12 Desenvolvimento de Endpoints para Consulta de CVEs, Feed OxSl_f33d e Leak-Lookup	Desenvolver novos endpoints na API para além dos dois criados anteriormente (consulta de Domínios e Ips) para permitir a consulta de vulnerabilidades (CVEs), o feed externo OxSl_f33d com as informações já previamente tratadas e a base de dados Leak-Lookup.	Médio/Médio	1 - A API deve disponibilizar mais três endpoints funcionais e documentados: • POST /cves • POST /oxsl_feed • POST /leak_lookup 2 - A resposta retornada por cada endpoint deve ser equivalente ao output original do script, mantendo

			a estrutura de dados e formato JSON. 3 - A API deve implementar mecanismos robustos de tratamento de erros e validação de inputs, assegurando que entradas inválidas não causam falhas de execução. 4 - A solução deve ser validada através de testes funcionais utilizando o Postman ou em scripts de teste desenvolvidos
REQ-13 Desenvolvimento de um Script de Web Scraping ao PishTank	Desenvolver um script em Python com capacidade de realizar scraping ao site PishTank com o BeautifulSoup para recolha automatizada de URLs maliciosos, classificadas como tentativas de phishing.	Baixa/Baixo	1 - Script deve usar BeautifulSoup para extrair URLs maliciosos do PhishTank. 2 - Deve conseguir executar scraping sem erros. 3 - Dados devem ser guardados ou retornados em formato estruturado. 4 - Deve lidar com casos de falha de conexão ou mudanças na estrutura do site.
REQ-14 Desenvolvimento de Endpoints para Consulta de dados relacionados com o feed do opencti	Desenvolver novos endpoints na API, para além dos já criados anteriormente, que permitam a consulta de domínios, hashes e IPs recolhidos no dia pelo OpenCTI, os 10 principais ataques com destino a um determinado país, e os ataques dirigidos a um país durante um período de tempo definido.	Médio/Médio	1 - A API deve disponibilizar os novos endpoints funcionais e documentados 2 - A resposta retornada por cada endpoint deve ser equivalente ao output original do script, mantendo a estrutura de dados e formato JSON. 3 - A API deve implementar mecanismos robustos de tratamento de erros e validação de inputs, assegurando que entradas inválidas não causam falhas de execução. 4 - A solução deve ser validada através de testes

			funcionais utilizando o Postman ou em scripts de teste desenvolvidos

Tabela 3-Requisitos Não Funcionais

ID do Requisito	Descrição do Requisito	Prioridade/Impacto
REQ-15 A API deverá ter uma base estável para futuras expansões.	A API deverá ser desenvolvida com uma arquitetura escalável, suportando por isso a futura adição de novas integrações sem comprometer o desempenho da mesma e por sua vez da aplicação.	Alta/Médio
REQ-16 A API deverá suportar, de forma estável, o grande volume de dados recebidos	A API deverá ser capaz de processar e suportar grandes volumes de dados , garantindo a estabilidade da aplicação mesmo em cenários de alta demanda de informação e assegurando assim a continuidade da mesma	Alta/Alto
REQ-17 A API deverá ser capaz de normalizar os dados recebidos de diversas plataformas.	A API deverá ser capaz de normalizar os dados recolhidos de diferentes pontos de informação, convertendo as informações para um formato padronizado. Esta normalização facilitará a correlação e análise cruzada entre os dados de diversas fontes	Alta/Alto
REQ-18	A API deverá fornecer um guia detalhado dos passos	Média/Médio

A API deverá incluir um guia de instalação detalhado	para a instalação da mesma, bem como um manual de configuração para o correto funcionamento da API, este guia deve retratar passo a passo a instalação inicial e a resolução de problemas comuns	
REQ-19 A API deverá conter um guia de utilização completo	Para o correto uso da API, esta deverá incluir um manual de utilização que descreva por completo as funcionalidades da API , os exemplos dos endpoints e melhores práticas	Média/Médio
REQ-20 Implementar limites de requisições por IP ou chave de API para evitar ataques de Denial of Service	A API deverá conter mecanismos que permitem que haja um controlo de requisições por IP ou chave de API para que o serviço esteja sempre disponível evitando ataques de Denial of Service	Alta/Alto

3.1.2 Descrição detalhada dos requisitos principais

Neste ponto, descrevemos com maior exatidão os requisitos considerados mais importantes, indicando as suas dependências, objetivos e critérios de aceitação. Esta descrição complementa a visão geral apresentada no ponto anterior.

[REQ-06] Limpeza e Operacionalização do script Monitorizador fornecido pela Cybers3c

Este requisito tem como objetivo principal a revisão completa do *script* Monitorizador, originalmente desenvolvido pela CyberS3c. Uma vez que o *script* não se encontra funcional, será essencial identificar e resolver erros de execução, conflitos de dependências, bibliotecas desatualizadas e problemas de estrutura de código que impeçam o seu correto funcionamento.

A limpeza e documentação do código serão cruciais para um melhor entendimento do funcionamento do Monitorizador. Essa reestruturação será acompanhada de perto pela CyberS3c, de forma a garantir que nenhuma funcionalidade anteriormente desenvolvida seja comprometida. As alterações no código serão implementadas de forma

incremental, permitindo que a CyberS3c valide e ajuste os objetivos para este requisito em cada etapa.

O objetivo final é que o *script* funcione corretamente e de forma eficiente, e que seja possível integrá-lo com novos módulos. Isso proporcionará uma base sólida para os restantes desenvolvimentos e futuras integrações na plataforma.

[REQ-07 & REQ-08] Integração de APIs de Infraestrutura, Vulnerabilidades, Leaks e Segurança Web

Este requisito contempla a expansão das capacidades do script Monitorizador através da integração de fontes externas de threat intelligence, com foco na exposição de infraestrutura digital, vulnerabilidades conhecidas, leaks de credenciais e segurança de domínios.
No âmbito do REQ-07, serão incorporadas APIs que fornecem dados sobre infraestrutura e riscos de segurança, incluindo:

- Shodan – para identificação de dispositivos expostos e portas abertas;
- Leak-Lookup – para deteção de credenciais comprometidas;
- ExploitDB – para consulta de exploits públicos associados a CVEs;
- BlackListChecker – para verificação de IPs e domínios em listas negras.

Já no âmbito do REQ-08, será realizada a integração de ferramentas de análise de segurança web, tais como:

- URLscan.io – para visualização e reputação de páginas web;
- Mozilla HTTP Observatory – para análise de cabeçalhos de segurança e boas práticas;
- VirusTotal – para deteção de conteúdos maliciosos;
- Web-Check e SecurityTrails – para insights sobre configuração de domínios, tecnologias utilizadas e possíveis vulnerabilidades.

Estas integrações irão reforçar a profundidade da análise automatizada de ativos digitais, permitindo à API fornecer aos utilizadores informações contextuais, técnicas e acionáveis, com o objetivo de detetar vulnerabilidades, riscos e exposições em tempo real.

[REQ-09] Adaptação do script Monitorizador como uma API Restful

Este requisito exige a transformação do script Monitorizador numa API *Restful*, utilizando a *framework* Flask. Esta API deverá, inicialmente, possuir duas rotas seguras e bem definidas: uma para os Domínios e outra para os IPs. Estas funcionalidades, já

desenvolvidas no script original, deverão ter uma resposta bem estruturada em JSON. Adicionalmente, a API deverá ser capaz de lidar com tratamento de exceções e incluir documentação básica integrada.

[REQ-14] A API deverá suportar, de forma estável, o grande volume de dados recebidos

Este requisito visa garantir que a API seja capaz de suportar eficientemente um grande volume de dados, considerando a integração com múltiplas APIs externas, fontes públicas e as diversas funções implementadas no módulo Monitorizador.
Para manter a estabilidade e o desempenho da aplicação, mesmo em cenários de elevada carga, é essencial a implementação de *threads* ou paralelismo. Isso permitirá que múltiplas operações de recolha e análise de dados ocorram de forma simultânea, reduzindo significativamente o tempo de resposta.
Adicionalmente, o código deverá ser otimizado, com a remoção de redundâncias e a reestruturação de blocos condicionais complexos (como if/else encadeados), garantindo maior legibilidade e eficiência. Estas medidas são fundamentais para evitar atrasos na execução e assegurar que a API se mantém responsiva mesmo com consultas intensivas e chamadas concorrentes a várias fontes externas.

[REQ-15] A API deverá ser capaz de normalizar dos dados recebidos de diversas plataformas

Este requisito visa Como a API irá consumir e correlacionar dados de diferentes fontes de threat intelligence (OpenCTI, MISP, Shodan, etc.), será essencial aplicar uma camada de normalização que traduza formatos distintos para um modelo comum de dados. A uniformização facilitará análises cruzadas, criação de relatórios e correlação entre eventos e indicadores de comprometimento (IoCs), além de melhorar a eficácia dos alertas e respostas automatizadas.

3.1.3 Casos de Uso/*User Stories*

Nesta seção, são apresentados cenários de utilização real da solução desenvolvida, esta representação permite contextualizar os requisitos descritos nos pontos anteriores e também compreender melhor o seu impacto. Serão abordados casos de uso em forma de diagramas de casos de uso que vão estar disponíveis nas imagens imediatamente abaixo e que refletem situações práticas, facilitando a compreensão do contexto de aplicação da API e das funcionalidades associadas
Neste contexto definimos um Atores como sendo:

- Utilizador da plataforma (Exemplo: No VIRIATUS fazer um varrimento externo a um domínio)

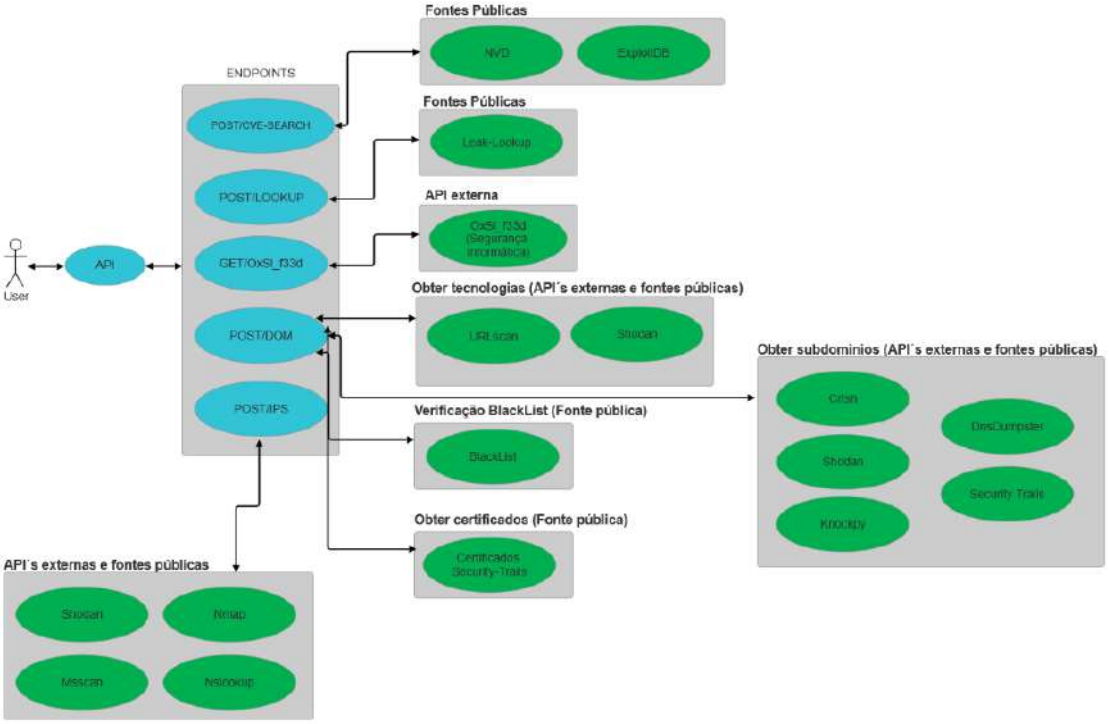


Figura-7 Use Case Request API

3.2 Modelação

Neste capítulo, é apresentado o diagrama de entidade-relação do nosso projeto, que representa a estrutura da base de dados da aplicação. A Figura 8-Diagrama Entidade-Relação ilustra este modelo. O diagrama foi desenvolvido de forma a garantir que os dados estejam organizados e normalizados, cumprindo a Terceira Forma Normal.

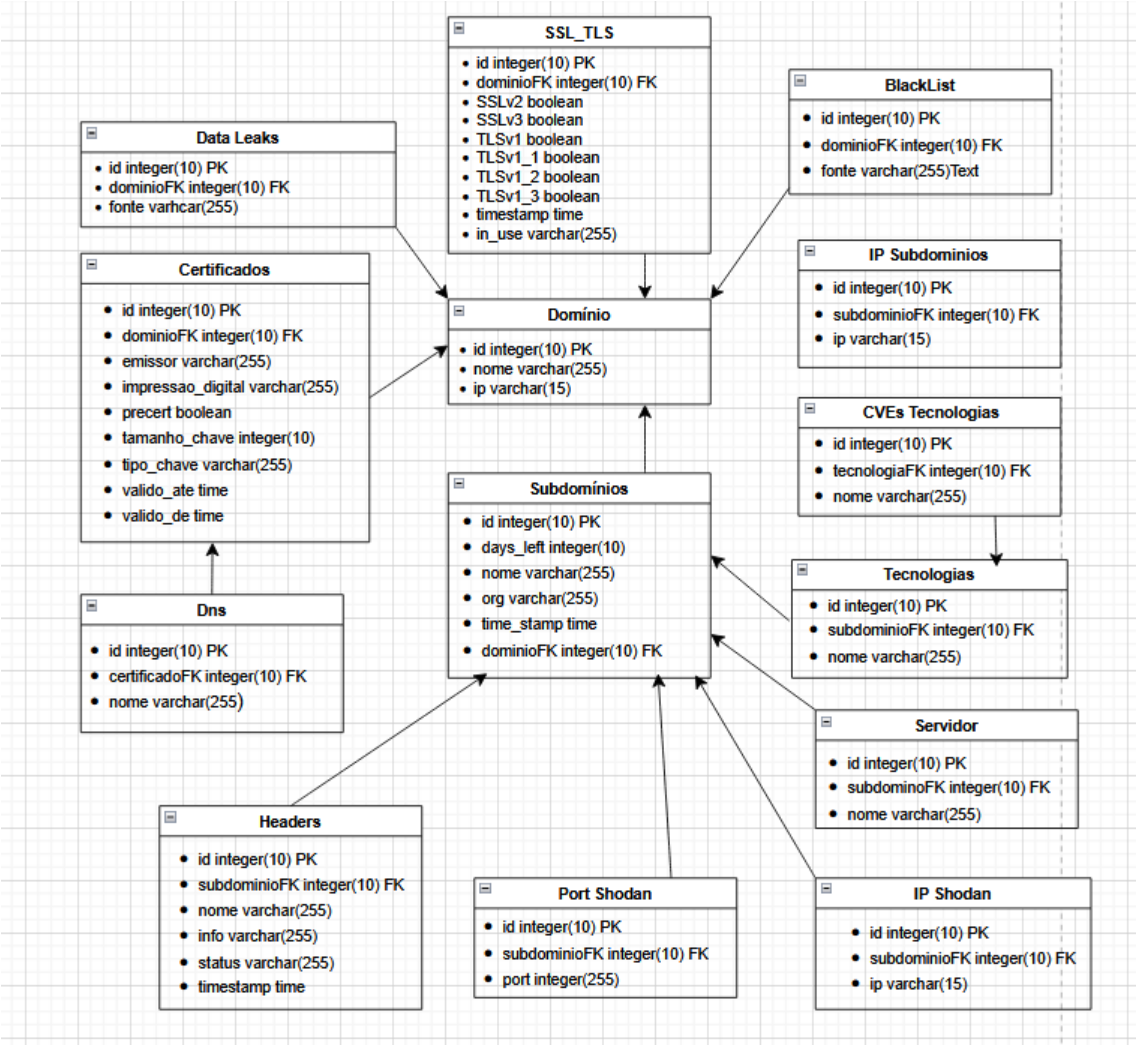


Figura 8-Diagrama Entidade-Relação

3.3 Protótipos de Interface

Neste capítulo, é apresentado o mapa aplicacional da API, ilustrando a sua estrutura e as suas interações. A Figura 9-Mapa Aplicacional API demonstra este modelo. Este mapa

reflete os componentes da API e ilustra a forma como a navegação ocorre entre eles, bem como a interação esperada entre os diferentes módulos.

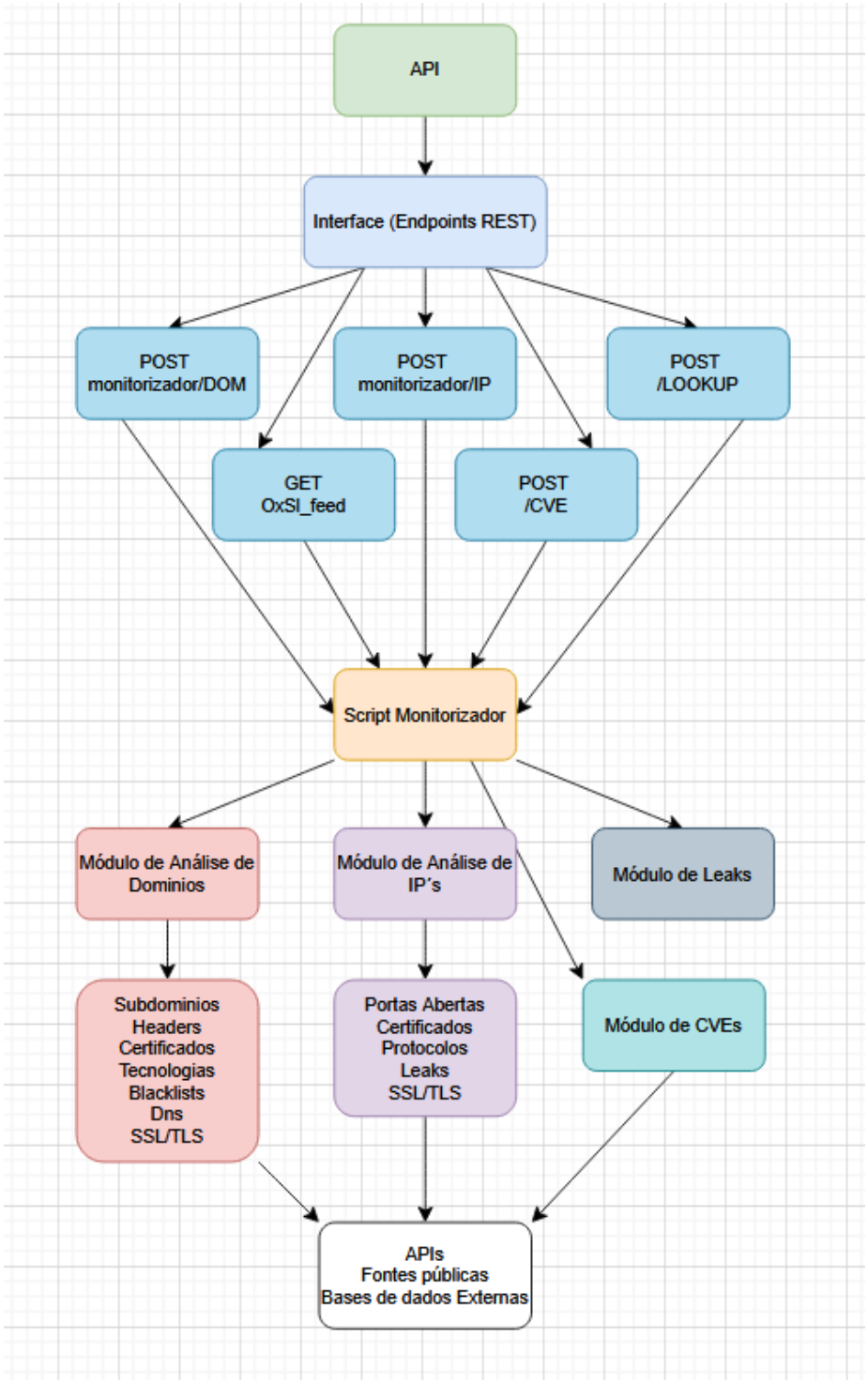


Figura 9-Mapa Aplicacional API

4 Solução Proposta

4.1 Apresentação

A nossa solução proposta visa o aprimoramento da plataforma VIRIATUS, desenvolvida pela CyberS3c, através de uma API. Esta API oferece uma solução abrangente para a gestão de ciberameaças e proteção de ativos, centralizando e organizando dados sobre vulnerabilidades e IoCs, correlacionando-os de modo a ampliar a compreensão sobre ataques e ameaças. Através desta abordagem, as organizações podem mitigar os seus riscos, tomar decisões mais informadas de forma proativa e garantir segurança e escalabilidade. Espera-se, assim, oferecer uma abordagem eficaz e simples que facilite todas as tarefas de gestão de vulnerabilidades externas relacionadas com a plataforma VIRIATUS. Visto que nos foi disponibilizado um código inicial pela empresa parceira, foi desenvolvido no **Anexo A** uma comparação que descreve o estado inicial do código disponibilizado comparativamente ao código final com as respetivas alterações realizadas, dando assim uma visão geral e adaptada das modificações executadas.
Nos subcapítulos seguintes, serão abordados os seguintes tópicos:

- **Subcapítulo 4.2:** Aborda a arquitetura utilizada para a solução, as tecnologias a empregar no desenvolvimento do Trabalho Final de Curso (TFC) e a fundamentação das principais opções construtivas.
- **Subcapítulo 4.3:** Contém uma breve descrição das tecnologias e ferramentas utilizadas, com a respetiva justificação de uso.
- **Ponto 4.4:** Descreve o ambiente produtivo da solução a desenvolver, indicando os recursos necessários para a sua exploração produtiva.
- **Subcapítulo 4.5:** Apresenta todas as áreas de conhecimento relevantes para o projeto e a sua justificação de aplicação.
- **Ponto 4.6:** Detalha cada um dos componentes, realçando aspetos técnicos da sua implementação.

O nosso trabalho apresenta algumas restrições devido à parceria com a empresa CyberS3c, o que impossibilita a publicação do código-fonte num repositório público como o GitHub. Assim, para permitir a compreensão e análise da nossa solução, será disponibilizado um vídeo com uma pequena demonstração do funcionamento dos componentes desenvolvidos. Adicionalmente, a documentação técnica completa da API (Anexo C) e os guias de instalação e configuração de plataformas cruciais como o OpenCTI (Anexo D) encontram-se igualmente disponíveis para consulta, fornecendo os detalhes necessários sobre a implementação e utilização.

https://youtu.be/CD5KoiqI1Uc

4.2 Arquitetura

A imagem abaixo ilustra a arquitetura da API, que será o ponto principal do nosso projeto. A arquitetura apresentada procura representar, de forma aproximada, o esqueleto da API. Esta estrutura visa garantir uma integração com a máxima eficiência

e segurança entre os componentes da mesma, assegurando que a recolha e análise de dados da infraestrutura externa ocorra de forma coordenada e escalável.

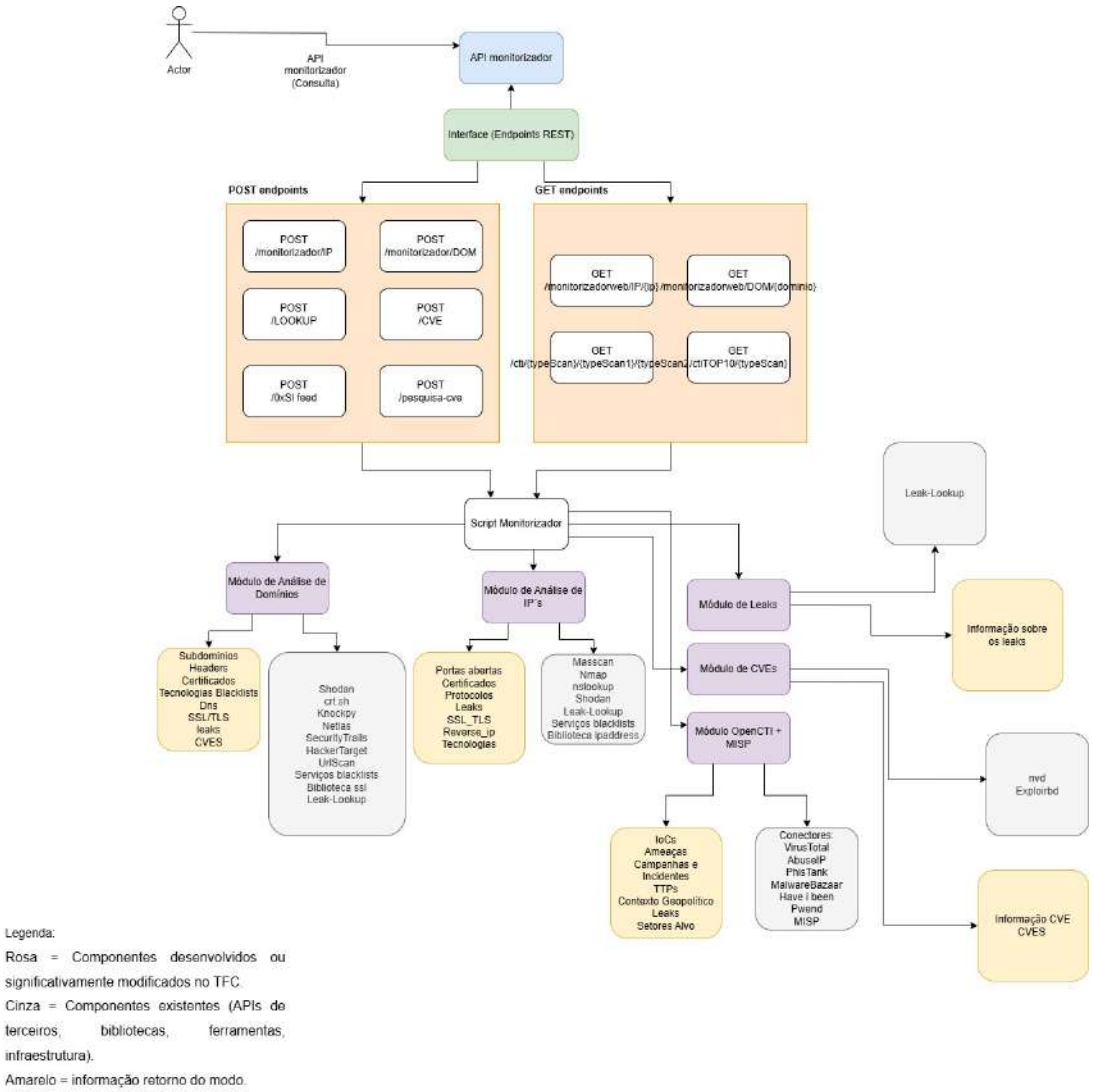


Figura 10-Arquitetura da API

4.3 Tecnologias e Ferramentas Utilizadas

4.3.1 Tecnologias

Python - Python é uma linguagem de programação de alto nível, conhecida pela sua simplicidade e legibilidade. É versátil, utilizada para diversos fins como desenvolvimento *web*, ciência de dados, inteligência artificial e automação. Esta linguagem possui uma vasta gama de bibliotecas e *frameworks*, o que facilita a criação de APIs REST com rapidez e segurança, preenchendo assim os nossos requisitos.

Flask - O Flask é um *microframework web* desenvolvido em Python, ideal para a criação de APIs e aplicações *web* leves. O Flask oferece simplicidade, flexibilidade e controlo total sobre os componentes utilizados

4.3.2 Ferramentas Utilizadas

MISP - O MISP é uma solução de *software* de código aberto para recolher, armazenar, distribuir e partilhar indicadores de cibersegurança e ameaças relacionadas com a análise de incidentes de cibersegurança e análise de *malware*. O MISP foi concebido por e para analistas de incidentes, profissionais de segurança e de TI, ou especialistas em engenharia reversa de *malware*, com o objetivo de apoiar as suas operações diárias, permitindo a partilha eficiente de informações estruturadas.

OpenCTI - O OpenCTI é uma plataforma de código aberto que permite às organizações gerir os seus conhecimentos e observáveis de *threat intelligence*. Foi criada para estruturar, armazenar, organizar e visualizar informações técnicas e não técnicas sobre ciberameaças. Além disso, o OpenCTI pode ser integrado com outras ferramentas e aplicações, como o MISP.

Shodan - O Shodan é um motor de busca que permite aos utilizadores encontrar dispositivos específicos como computadores, servidores, *routers*, *webcams* e dispositivos IoT, entre outros, conectados à rede. Foi desenvolvido para gerir e analisar os *banners* que os servidores enviam de volta aos clientes. Desta forma, possibilita analisar que dispositivos estão conectados, onde estão localizados, que *software* estão a executar e que portas estão abertas.

Exploit-DB - O Exploit Database é um repositório de *exploits* relativos a vulnerabilidades de segurança conhecidas. Foi criado com o intuito de servir como recurso centralizado para investigadores de segurança, profissionais de testes de intrusão e entusiastas da área da segurança. O mesmo disponibiliza o acesso ao código e às técnicas que exploram falhas nos sistemas.

NVD - A *National Vulnerability Database* é uma base de dados mantida pelo governo dos EUA que reúne informações sobre vulnerabilidades de segurança de *software* e *hardware*. Fornece classificações de risco, descrições técnicas e métricas.

0xSl_f33d- O 0xSl_f33d**,** é um feed criado e desenvolvido pela Segurança Informática, é um repositório público que agrega campanhas de phishing e malware direcionadas a cidadãos portugueses, sendo alimentado por sensores automáticos e contribuições da comunidade.

Leek lookup - O Leak Lookup é uma ferramenta *online* que permite verificar se os dados de um endereço de *e-mail*, um domínio ou um *username* foram expostos em fugas de informação. Baseia-se em bases de dados públicas de *leaks* e fornece detalhes sobre quais serviços foram comprometidos.

Securytytrails - O SecurityTrails é uma plataforma de cibersegurança que fornece dados abrangentes sobre domínios, endereços IP e informações DNS, tanto atuais quanto históricos.

UrlScan- O UrlScan é um serviço *online* que analisa e inspeciona URLs, permitindo identificar conteúdos maliciosos e suspeitos.

Blacklistchecker- O Blacklist Checker é uma ferramenta que permite verificar se um domínio ou endereço IP está presente em listas negras de reputação na *internet*, consultando múltiplas bases de dados em simultâneo.

4.4 Ambientes de Teste e de Produção

A solução da API de *Threat Intelligence* será implementada em dois tipos de ambientes distintos: um ambiente de teste e um ambiente de produção. O ambiente de teste será utilizado para validação, desenvolvimento e integração de novas funcionalidades antes de serem transpostas para o ambiente de produção. Este último, por sua vez, será configurado para suportar uma operação contínua, garantindo disponibilidade e desempenho. A existência destes dois tipos de ambientes é essencial para assegurar uma solução robusta, segura e com capacidade de escalabilidade, contribuindo para a redução de riscos operacionais e para a integração contínua.

4.5 Abrangência

Nome das Disciplinas Associadas:

- Engenharia de Software
- Engenharia de Requisitos e Testes
- Sistema de Informação na Nuvem
- Computação Distribuída
- Programação Web

Neste projeto, serão aplicados os conhecimentos adquiridos ao longo do percurso académico, abrangendo áreas como programação *web*, computação distribuída e desenvolvimento da API REST, que é o ponto central do nosso projeto. Serão também utilizados conceitos de sistemas de informação na *cloud*, com o objetivo de criar uma infraestrutura de servidor que aloje *feeds* que alimentam a API principal. Além disso, serão integrados princípios de Engenharia de *Software* e Engenharia de Requisitos e Testes, que, em conjunto, desempenham um papel crucial na organização do projeto, na definição de requisitos e na distribuição das tarefas essenciais para o seu desenvolvimento de maneira correta e organizada.

4.6 Componente

4.6.1 **API CENTRAL**

A API centraliza todas as funcionalidades do serviço que iremos implementar, permitindo a integração com a plataforma. Utilizando uma linguagem de programação

como Python e com o auxílio de *frameworks*, busca-se aumentar a simplicidade, eficiência e agilidade de diversos processos. Esta API oferece também um alto nível de desempenho para garantir respostas rápidas a requisições frequentes.Foi também desenvolvido um documento completo sobre a implementação da API, contendo a explicação de todas as suas funções e funcionalidades, bem como os passos para a sua instalação e configuração, disponível no **Anexo C.**

4.6.1.1 Monitorizador

Um dos componentes centrais da nossa API consiste em dois *endpoints* dedicados à monitorização e recolha de informação sobre domínios (e respetivos subdomínios) e sobre IPs. Estes constituem o núcleo do nosso projeto e, consequentemente, da própria API. Encontram-se divididos nos seguintes *endpoints*:

- http://127.0.0.1:5000/monitorizador/IP
- http://127.0.0.1:5000/monitorizador/DOM

Ambos os *endpoints* foram desenvolvidos utilizando o método POST, permitindo o envio para a API de um JSON com uma lista de IPs ou, alternativamente, uma lista de domínios (podendo incluir os seus subdomínios). Este modelo possibilita uma análise mais abrangente e completa. Abaixo, apresenta-se uma maquete do JSON retornado pela API:

```
{
    "dominios": [
        {
            "domain": ,
            "ip": '',
            "time": ,
            "data_leaks": ,
            "certificados_securitytrails": ,
            "subdominios": [
                {
                    "nome": ,
                    "ip": '',
                    "ip_shodan": [],
                    "start_date": ,
                    "valid_until": ,
                    "days_left": ,
                    "org_name": ,
                    "time": ,
                    "headers": [
                        {
                            "header": ,
                            "info": ,
                            "status": ,
                            "time":
                        },

                    ],
                    "headers_shodan": [],
                    "ports_shodan": [],
                    "technologies_shodan": [],
                    "technologies_cves_shodan": null,
                    "servers": [

                    ],
                    "technologies_cves": {
                        "UIKit": [

                        ],

                    },
                    "tls_versions_shodan": [],
                    "services_shodan": []
                },
```


Figura 11-Json de Retorno Domínios

No caso dos domínios, o retorno em formato JSON inclui toda a informação disponível recolhida de várias fontes externas. Além disso, são identificados todos os subdomínios válidos, sendo igualmente recolhida informação detalhada sobre cada um deles.

Figura 12-Json Retorno IPS

No caso dos IPs, o retorno em formato JSON inclui todos os dados relevantes obtidos a partir de diversas fontes externas, assim como através de métodos de varrimento de portas (como Nmap e Masscan). Esta abordagem permite ainda identificar IPs associados ao pesquisado, juntamente com a respetiva análise detalhada.

4.6.1.2 Pesquisa de Fugas de informação

Um dos componentes da nossa API é um *endpoint* desenvolvido para verificar se um domínio, IP, nome de utilizador ou *e-mail* está presente em fugas de dados, retornando, em formato JSON, os dados encontrados. Isto permite prevenir acessos indevidos, utilizando o seguinte *endpoint*:

- http://127.0.0.1:5000/monitorizador/LOOKUP/

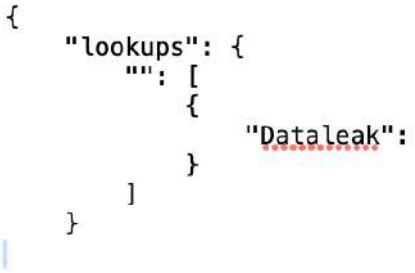


Figura 13-Json Retorno Leaks

Neste caso, o retorno em formato JSON inclui todos os *data leaks* encontrados relacionados com o elemento pesquisado.

4.6.1.3 Pesquisa de CVES por software

Um dos componentes da nossa API é um endpoint desenvolvido para, através de um *software* ou palavra-chave, retornar todos os CVEs associados, bem como a respetiva descrição, utilizando o seguinte endpoint:

- http://127.0.0.1:5000/monitorizador/pesquisar-cve/

Figura 14-Json Retorno CVEs

Neste caso, o retorno em formato JSON inclui todos os CVEs encontrados relacionados com o *software* pesquisado, juntamente com a descrição de cada um e o respetivo grau de gravidade.

4.6.1.4 Pesquisa de CVES

Um dos componentes da nossa API é um endpoint desenvolvido para fornecer toda a informação sobre um determinado CVE e todos os *exploits* associados ao CVE pesquisado, através do seguinte endpoint:

- http://127.0.0.1:5000/monitorizador/cve/

Figura 15-Json Retorno CVES informação

Neste caso, o retorno em formato JSON inclui todos os *exploits* associados ao CVE pesquisado, contendo informações relevantes sobre os mesmos, bem como o respetivo código.

4.6.1.5 Monitorizadorweb

Um dos componentes centrais da nossa API consiste em dois endpoints dedicados à monitorização e recolha de informação sobre domínios (e respetivos subdomínios) e sobre endereços IP. Estes constituem o núcleo do nosso projeto e, consequentemente, da própria API.
A pedido da empresa parceira, foi solicitado o desenvolvimento de uma versão dos dois endpoints principais utilizando o método GET, o que permite a visualização e realização de pedidos à API diretamente através de um *browser*.

Os endpoints encontram-se divididos da seguinte forma:

- http://127.0.0.1:5000/monitorizadorweb/IP/0.0.0.0/
- http://127.0.0.1:5000/monitorizadorweb/DOM/example.com/

Ambos os endpoints foram desenvolvidos à semelhança dos endpoints do monitorizador anteriormente explicados, mas agora adaptados ao método GET.

Abaixo apresenta-se uma maquete do JSON retornado pela API:

Figura 16-Monitorizador Domínios

No caso dos domínios, o retorno em formato JSON inclui toda a informação disponível recolhida de várias fontes externas. Além disso, são identificados todos os subdomínios válidos, sendo igualmente recolhida informação detalhada sobre cada um deles.

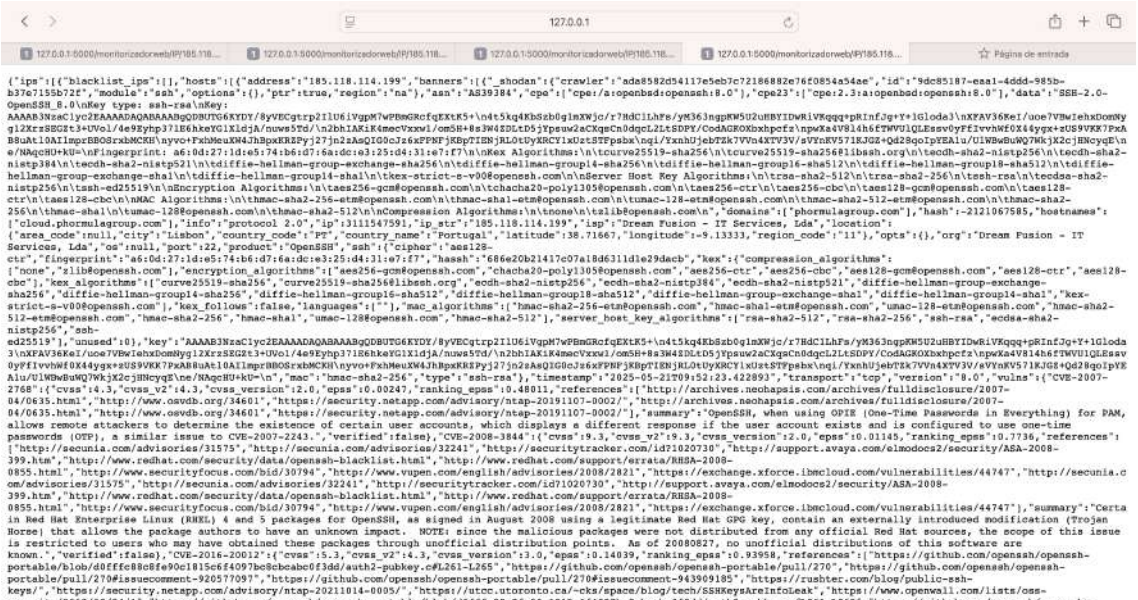


Figura 17-Monitorizador IPS

No caso dos IPs, o retorno em formato JSON inclui todos os dados relevantes obtidos a partir de diversas fontes externas, bem como através de métodos de varrimento de portas (como o Nmap e o Masscan). Esta abordagem permite ainda identificar IPs associados ao endereço pesquisado, juntamente com a respetiva análise detalhada.

4.6.1.6 OxSl_f33d

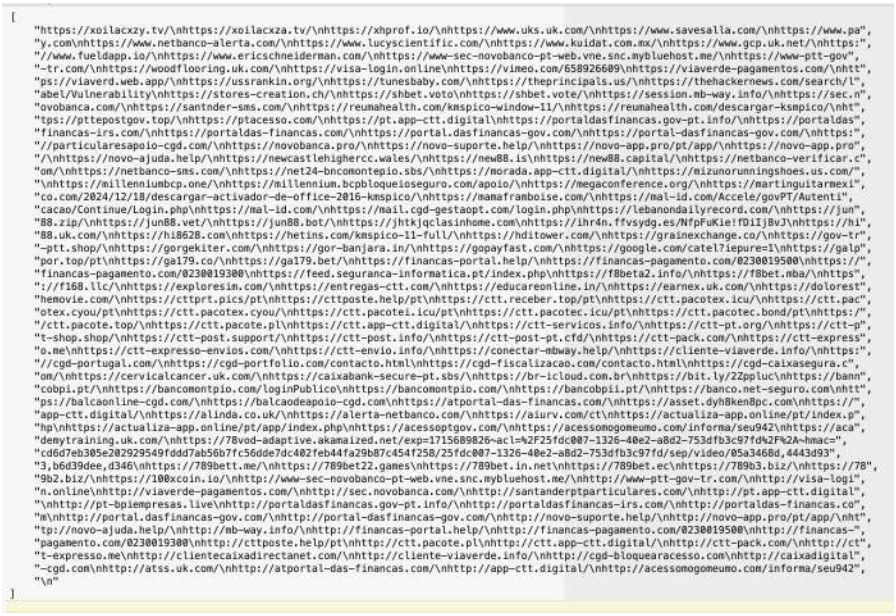
Um dos componentes da nossa API é um endpoint desenvolvido para obter, através de um *feed* de [Segurança Informática] , todos os domínios reportados como *phishing* no dia em questão.

Este recurso permite obter uma visão atualizada sobre os novos métodos utilizados em ataques maliciosos, contribuindo assim para uma melhor compreensão do impacto destas ameaças.

Adicionalmente, este endpoint tem a vantagem de focar-se no mercado nacional, alinhando-se com o principal objetivo do nosso projeto.

Pode ser acedido através do seguinte endpoint:

- http://127.0.0.1:5000/OxSl_f33d/



[
"https://xoilacxzy.tv/\nhttps://xoilacxza.tv/\nhttps://xhprof.io/\nhttps://www.uks.uk.com/\nhttps://www.savesalla.com/\nhttps://www.pa",
"y.com\nhttps://www.netbanco-alerta.com/\nhttps://www.lucyscientific.com/\nhttps://www.kuidat.com.mx/\nhttps://www.gcp.uk.net/\nhttps:",
"//www.fueldapp.io/\nhttps://www.ericschneiderman.com/\nhttps://www-sec-novobanco-pt-web.vne.snc.mybluehost.me/\nhttps://www-ptt-gov",
"-tr.com/\nhttps://woodflooring.uk.com/\nhttps://visa-login.online\nhttps://vimeo.com/658926609\nhttps://viaverde-pagamentos.com/\nhtt",
"ps://viaverd.web.app/\nhttps://ussrankin.org/\nhttps://tunesbaby.com/\nhttps://theprincipals.us/\nhttps://thehackernews.com/search/l",
"abel/Vulnerability\nhttps://stores-creation.ch/\nhttps://shbet.voto\nhttps://shbet.vote/\nhttps://session.mb-way.info/\nhttps://sec.n",
"ovobanca.com/\nhttps://santnder-sms.com/\nhttps://reumahealth.com/kmspico-window-11/\nhttps://reumahealth.com/descargar-ksmpico/\nht",
"tps://pttepostgov.top/\nhttps://ptacesso.com/\nhttps://pt.app-ctt.digital\nhttps://portaldasfinancas.gov-pt.info/\nhttps://portaldas",
"financas-irs.com/\nhttps://portaldas-financas.com/\nhttps://portal.dasfinancas-gov.com/\nhttps://portal-dasfinancas-gov.com/\nhttps:",
"//particularesapoio-cgd.com/\nhttps://novobanca.pro/\nhttps://novo-suporte.help/\nhttps://novo-app.pro/pt/app/\nhttps://novo-app.pro",
"/\nhttps://novo-ajuda.help/\nhttps://newcastlehighercc.wales/\nhttps://new88.is\nhttps://new88.capital/\nhttps://netbanco-verificar.c",
"om/\nhttps://netbanco-sms.com/\nhttps://net24-bncomontepio.sbs/\nhttps://morada.app-ctt.digital/\nhttps://mizunorunningshoes.us.com/",
"\nhttps://millenniumbcp.one/\nhttps://millennium.bcpbloqueioseguro.com/apoio/\nhttps://megaconference.org/\nhttps://martinguitarmexi",
"co.com/2024/12/18/descargar-activador-de-office-2016-kmspico/\nhttps://mamaframboise.com/\nhttps://mal-id.com/Accele/govPT/Autenti",
"cacao/Continue/Login.php\nhttps://mal-id.com/\nhttps://mail.cgd-gestaopt.com/login.php\nhttps://lebanondailyrecord.com/\nhttps://jun",
"88.zip/\nhttps://jun88.vet/\nhttps://jun88.bot/\nhttps://jhtkjqclasinhome.com\nhttps://ihr4n.ffvsydg.es/NfpFuKiefID1jBvJ\nhttps://hi",
"88.uk.com/\nhttps://hi8628.com\nhttps://hetins.com/kmspico-11-full/\nhttps://hditower.com/\nhttps://grainexchange.co/\nhttps://gov-tr",
"-ptt.shop/\nhttps://gorgekiter.com/\nhttps://gor-banjara.in/\nhttps://gopayfast.com/\nhttps://google.com/catel?iepure=1\nhttps://galp",
"por.top/pt\nhttps://ga179.co/\nhttps://ga179.bet/\nhttps://financas-portal.help/\nhttps://financas-pagamento.com/0230019500\nhttps://",
"financas-pagamento.com/0230019300\nhttps://feed.seguranca-informatica.pt/index.php\nhttps://f8beta2.info/\nhttps://f8bet.mba/\nhttps",
":://f168.llc/\nhttps://exploresim.com/\nhttps://entregas-ctt.com/\nhttps://educareonline.in/\nhttps://earnex.uk.com/\nhttps://dolorest",
"hemovie.com/\nhttps://cttprt.pics/pt\nhttps://cttposte.help/pt\nhttps://ctt.receber.top/pt\nhttps://ctt.pacotex.icu/\nhttps://ctt.pac",
"otex.cyou/pt\nhttps://ctt.pacotex.cyou/\nhttps://ctt.pacotei.icu/pt\nhttps://ctt.pacotec.icu/pt\nhttps://ctt.pacotec.bond/pt\nhttps:/",
"/ctt.pacote.top/\nhttps://ctt.pacote.pl\nhttps://ctt.app-ctt.digital/\nhttps://ctt-servicos.info/\nhttps://ctt-pt.org/\nhttps://ctt-p",
"t-shop.shop/\nhttps://ctt-post.support/\nhttps://ctt-post.info/\nhttps://ctt-post-pt.cfd/\nhttps://ctt-pack.com/\nhttps://ctt-express",
"o.me\nhttps://ctt-expresso-envios.com/\nhttps://ctt-envio.info/\nhttps://conectar-mbway.help/\nhttps://cliente-viaverde.info/\nhttps:",
"//cgd-portugal.com/\nhttps://cgd-portfolio.com/contacto.html\nhttps://cgd-fiscalizacao.com/contacto.html\nhttps://cgd-caixasegura.c",
"om/\nhttps://cervicalcancer.uk.com/\nhttps://caixabank-secure-pt.sbs/\nhttps://br-icloud.com.br\nhttps://bit.ly/2Zpp1uc\nhttps://bann",
"cobpi.pt/\nhttps://bancomontpio.com/loginPublico\nhttps://bancomontpio.com/\nhttps://bancobpii.pt/\nhttps://banco.pet-seguro.com\nhtt",
"ps://balcaonline-cgd.com/\nhttps://balcaodeapoio-cgd.com\nhttps://atportal-das-financas.com/\nhttps://asset.dyh8ken8pc.com\nhttps://",
"app-ctt.digital/\nhttps://alinda.co.uk/\nhttps://alerta-netbanco.com/\nhttps://aiurv.com/ct\nhttps://actualiza-app.online/pt/index.p",
"hp\nhttps://actualiza-app.online/pt/app/index.php\nhttps://acessoptgov.com/\nhttps://acessomogomeumo.com/informa/seu942\nhttps://aca",
"demytraining.uk.com/\nhttps://78vod-adaptive.akamaized.net/exp=1715689826~acl=%2F25fdc007-1326-40e2-a8d2-753dfb3c97fd%2F%2A~hmac=",
"cd6d7eb305e202929549fddd7ab56b7fc56dde7dc402feb44fa29b87c454f258/25fdc007-1326-40e2-a8d2-753dfb3c97fd/sep/video/05a3468d,4443d93",
"3,b6d39dee,d346\nhttps://789bett.me/\nhttps://789bet22.games\nhttps://789bet.in.net\nhttps://789bet.ec\nhttps://789b3.biz/\nhttps://78",
"9b2.biz/\nhttps://100xcoin.io/\nhttp://www-sec-novobanco-pt-web.vne.snc.mybluehost.me/\nhttp://www-ptt-gov-tr.com/\nhttp://visa-logi",
"n.online\nhttp://viaverde-pagamentos.com/\nhttp://sec.novobanca.com/\nhttp://santanderptparticulares.com/\nhttp://pt.app-ctt.digital",
"\nhttp://pt-bpiempresas.live\nhttp://portaldasfinancas.gov-pt.info/\nhttp://portaldasfinancas-irs.com/\nhttp://portaldas-financas.co",
"m\nhttp://portal.dasfinancas-gov.com/\nhttp://portal-dasfinancas-gov.com/\nhttp://novo-suporte.help/\nhttp://novo-app.pro/pt/app/\nht",
"tp://novo-ajuda.help/\nhttp://mb-way.info/\nhttp://financas-portal.help/\nhttp://financas-pagamento.com/0230019500\nhttp://financas-",
"pagamento.com/0230019300\nhttp://cttposte.help/pt\nhttp://ctt.pacote.pl\nhttp://ctt.app-ctt.digital/\nhttp://ctt-pack.com/\nhttp://ct",
"t-expresso.me\nhttp://clientecaixadirectanet.com/\nhttp://cliente-viaverde.info/\nhttp://cgd-bloqueearacesso.com\nhttp://caixadigital",
"-cgd.com\nhttp://atss.uk.com/\nhttp://atportal-das-financas.com/\nhttp://app-ctt.digital/\nhttp://acessomogomeumo.com/informa/seu942",
"\n"
]

Figura 18-OxSl_f33d

Neste caso, o retorno em formato JSON inclui todos os domínios associados a campanhas de *phishing* reportadas no dia da requisição.

4.6.2 Módulo de Corelacionamento de Dados

Este módulo será responsável por processar e correlacionar dados provenientes de várias fontes externas, como vulnerabilidades e IoCs, com o objetivo de gerar *insights* mais profundos e robustos. Este módulo será suportado num servidor na *cloud* e, por sua vez, os dados serão inseridos na API central. Será desenvolvido com recurso ao OpenCTI (cuja documentação de instalação e configuração detalhada se encontra no

Anexo D) e a *feeds* de informação, incluindo o *feed* do MISP, associado a todos os *feeds* que este já possui.

Figura 19-OpenCTI

Figura 20-Feeds Opencti

Este módulo será responsável por processar e correlacionar dados provenientes de várias fontes externas, como vulnerabilidades e IoCs, com o objetivo de gerar *insights* mais profundos e robustos. Este módulo será suportado num servidor na *cloud* e, por sua vez, os dados serão inseridos na API central. Será desenvolvido com recurso ao OpenCTI e a *feeds* de informação, incluindo o *feed* do MISP, associado a todos os *feeds* que este já possui.

4.6.2.1 Ctiweb

Este componente foi desenvolvido para permitir o acesso a todos os *hashes*, domínios ou endereços IP identificados no dia que foram correlacionados no OpenCTI.

Os respetivos endpoints são os seguintes:

- http://127.0.0.1:5000/ctiweb/hashes/
- http://127.0.0.1:5000/ctiweb/IPS/
- http://127.0.0.1:5000/ctiweb/DOM/

Figura 21-cti IPS

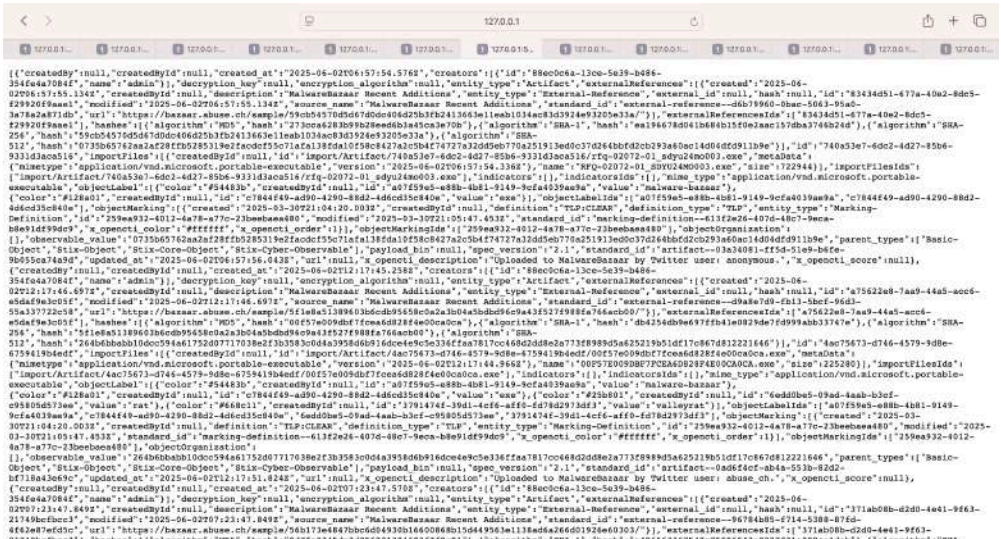


Figura 22-cti hashes

```
{
    "2025-05-26T00:00:00Z": [
        "backup-tlscom.sytes.net",
        "admin.668608.xyz",
        "image.windowstimes.online",
        "1212tank.activitydmy.icu",
        "c43f5d6e73a7eb.ccega6r0yph8.com",
        "0ac0568239f8978.ccega6r0yph8.com",
        "mohsar.ddns.net",
        "784564141.ccega6r0yph8.com",
        "api.xwphd.com",
        "outlook-office.micsrosoftonline.com",
        "bkp.windowstimes.me",
        "times.windowstimes.me",
        "magic-telecom.ddns.net",
        "images.windowstimes.online",
        "pub-ce02802067934e0eb072f69bf6427bf6.r2.dev",
        "term-restore-satisfied-hence.trycloudflare.com",
        "times.windowstimes.online"
    ]
}
```


Figura 23-cti domains

4.6.2.2 ctiTop10

Este endpoint foi desenvolvido para devolver os 10 principais ataques direcionados a um determinado país.

O endpoint correspondente é o seguinte:

- http://127.0.0.1:5000/ctiTop10/Portugal/

4.6.2.3 Ctipais

Este endpoint foi desenvolvido para devolver os ataques direcionados a um determinado país durante um período de tempo específico. É possível definir um intervalo temporal concreto (por exemplo, entre dois meses), obter apenas os ataques mais recentes até uma determinada data ou, ainda, escolher se se pretende devolver todos os ataques ou apenas os primeiros resultados da pesquisa. O endpoint correspondente é o seguinte:

- http://127.0.0.1:5000/ctipais/Portugal/03-2025/04-2025/
- http://127.0.0.1:5000/ctipais/Portugal/03-2025/
- http://127.0.0.1:5000/ctipaisfim/Portugal/04-2025/
- http://127.0.0.1:5000/ctipais/Portugal/


```
{"Portugal":[{"TLP":"TLP:CLEAR","atacante":"Shenzhen Haimaiyunxiang Media Co., Ltd.","confian\u00e7a":100,"criado":"2025-04-
01T08:50:47.803Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Intrusion-Set"},
{"TLP":"TLP:CLEAR","atacante":"SYS01","confian\u00e7a":100,"criado":"2025-04-
01T10:54:36.991Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"Hadoken","confian\u00e7a":100,"criado":"2025-04-
01T02:42:06.290Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"FlyGram","confian\u00e7a":100,"criado":"2025-04-
01T05:48:43.607Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"Zombinder","confian\u00e7a":100,"criado":"2025-04-
01T02:42:07.552Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"BadBazaar","confian\u00e7a":100,"criado":"2025-04-
01T05:49:03.727Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"Coper","confian\u00e7a":100,"criado":"2025-04-
01T09:31:46.534Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Intrusion-Set"},
{"TLP":"TLP:CLEAR","atacante":"GymDrop","confian\u00e7a":100,"criado":"2025-04-
01T02:42:09.947Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},{"TLP":"TLP:CLEAR","atacante":"Cerberus (ex-
Amnesia)","confian\u00e7a":100,"criado":"2025-04-01T11:27:00.874Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Intrusion-Set"},
{"TLP":"TLP:CLEAR","atacante":"BugDrop","confian\u00e7a":100,"criado":"2025-04-
01T02:42:12.584Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"}]}
```


Figura 24-ctitop10 Portugal

{"Portugal":[{"TLP":"TLP:CLEAR","atacante":"Shenzhen Haimaiyunxiang Media Co., Ltd.","confian\u00e7a":100,"criado":"2025-04-
01T08:50:47.800Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Intrusion-Set"},
{"TLP":"TLP:CLEAR","atacante":"SYS01","confian\u00e7a":100,"criado":"2025-04-
01T10:54:36.991Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"Hadoken","confian\u00e7a":100,"criado":"2025-04-
01T02:42:06.290Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"FlyGram","confian\u00e7a":100,"criado":"2025-04-
01T05:48:43.607Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"Zombinder","confian\u00e7a":100,"criado":"2025-04-
01T02:42:07.552Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"BadBazaar","confian\u00e7a":100,"criado":"2025-04-
01T05:49:03.727Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"Coper","confian\u00e7a":100,"criado":"2025-04-
01T09:31:46.534Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Intrusion-Set"},
{"TLP":"TLP:CLEAR","atacante":"GymDrop","confian\u00e7a":100,"criado":"2025-04-
01T02:42:09.947Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},{"TLP":"TLP:CLEAR","atacante":"Cerberus (ex-
Amnesia)","confian\u00e7a":100,"criado":"2025-04-01T11:27:00.874Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Intrusion-Set"},
{"TLP":"TLP:CLEAR","atacante":"BugDrop","confian\u00e7a":100,"criado":"2025-04-
01T02:42:12.584Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"}]}

Figura 25- ctipais Portugal

{"Portugal":[{"TLP":"TLP:CLEAR","atacante":"Lampion","confian\u00e7a":100,"criado":"2025-05-
06T15:37:27.633Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"Lampion","confian\u00e7a":100,"criado":"2025-05-
06T15:37:26.990Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Intrusion-Set"}]}

Figura 26-ctipais Portugal-04-2025/05-2025

{"Portugal":[{"TLP":"TLP:CLEAR","atacante":"Lampion","confian\u00e7a":100,"criado":"2025-05-
06T15:37:27.633Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"Lampion","confian\u00e7a":100,"criado":"2025-05-
06T15:37:26.990Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Intrusion-Set"}]}

Figura 27-ctipais Portugal-04-2025

Todas estas variações de endpoints relativas à aplicação de *threat intelligence* foram desenvolvidas no formato GET. Isso permite que as requisições sejam feitas através de motores de busca. Além disso, optou-se por este método uma vez que, devido à natureza da origem dos dados, não é necessário enviar uma lista com diversos *inputs*.

5 Testes e Validação

Este capítulo é de extrema importância para a realização do projeto como um todo. É através dos testes e da validação que garantimos que a solução desenvolvida cumpre os objetivos definidos, demonstrando a sua pertinência e o cumprimento dos critérios de aceitação. A relevância deste capítulo é ainda maior dada a área do nosso projeto, a cibersegurança, onde é essencial minimizar todos os erros possíveis.
Além disso, a parceria com a CyberS3c acarreta uma maior responsabilidade para assegurar que todos os requisitos propostos funcionem como pretendido. Nesse âmbito, diversos testes foram realizados durante todo o desenvolvimento do projeto até à data. Uma das principais exigências da empresa parceira foi que a API desenvolvida não entrasse em *crash* em circunstância alguma.
Dessa forma, foram realizados vários testes nesse sentido. Os testes desenvolvidos tiveram como objetivo cobrir todos os *endpoints* implementados e funcionais até à data.

***Endpoints* Testados:**

- **/monitorizador/DOM/** – Responsável por retornar informações relativas ao domínio analisado.

- **/monitorizador/IP/** – Responsável por retornar informações relativas ao IP analisado.

- **/LOOKUP/** – Executa pesquisas sobre um domínio ou IP para verificação cruzada em fontes externas.

- **/CVES/** – Lista vulnerabilidades conhecidas (Common Vulnerabilities and Exposures) associadas a um determinado ativo.

- **/pesquisar-cve/** – Permite pesquisa detalhada de vulnerabilidades com base em palavras-chave ou identificadores específicos.

- **/monitorizadorweb/DOM/exemple/** – Responsável por retornar informações relativas ao domínio analisado via *web*.

- **/monitorizadorweb/IP/0.0.0.0/** – Responsável por retornar informações relativas ao IP analisado via *web*.

- **/0xSl_f33d/** – Responsável por retornar informações de domínios portugueses associados a *phishing* no dia de hoje.

- **/ctiweb/hashes/** – Responsável por retornar informações de *hashes* associados pelo OpenCTI, também analisando os *endpoints* DOM e IP.

- **/ctipais/Portugal/** – Responsável por devolver informações relativas aos ataques direcionados a Portugal, recolhidas pelo OpenCTI durante um determinado período de tempo.

- **/ctiTOP10/** – Responsável por devolver informações relativas aos top 10 ataques direcionados a Portugal, recolhidas pelo OpenCTI.

Como o projeto se foca no desenvolvimento de uma API, os testes foram direcionados para casos de uso reais. No Anexo , estão disponíveis todos os casos de testes realizados, bem como os resultados esperados e obtidos.

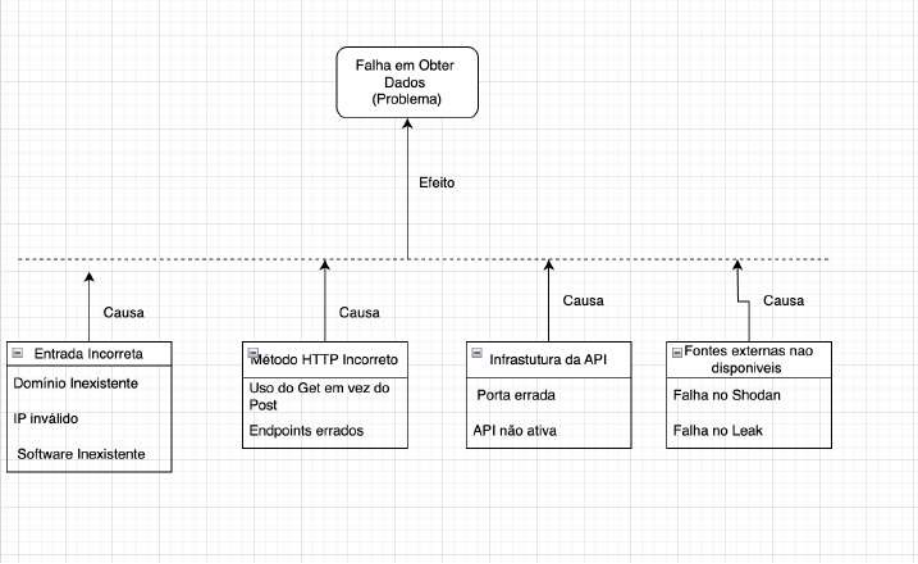


Figura 28-Diagrama causa-efeito

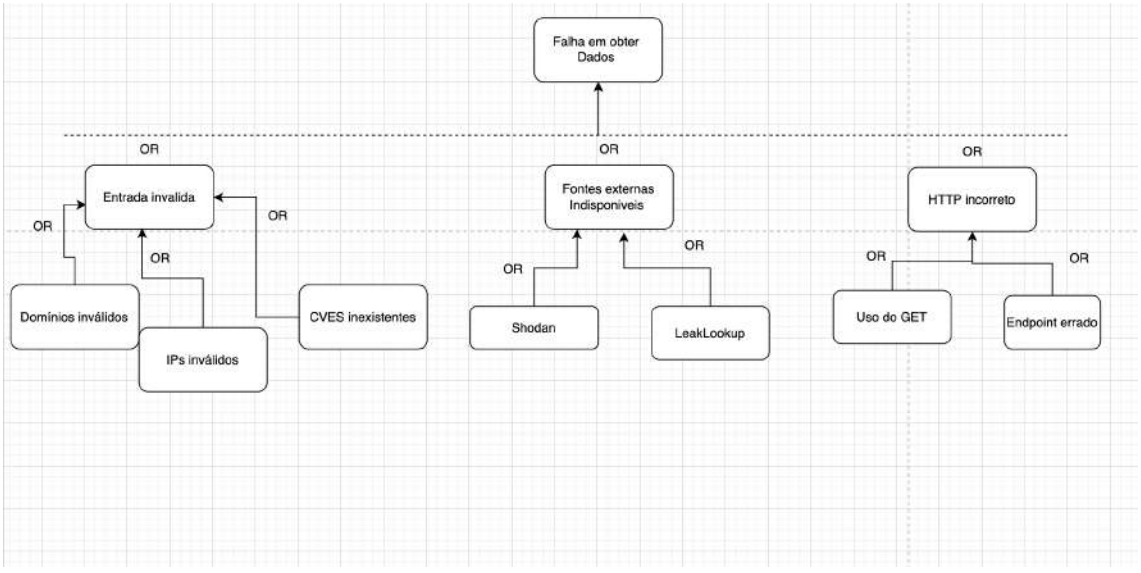


Figura 29-Diagrama Fault Tree Analysis

Com o desenvolvimento de ambos os modelos formais, foi possível elaborar testes destinados à verificação de falhas que poderiam ocorrer durante a utilização da API (conforme apresentado no Anexo)

Os testes foram realizados num computador Mac equipado com um processador M1 e 16 GB de memória RAM. Durante a execução de pedidos à API, foram monitorizados os níveis de utilização do processador, da internet e da RAM.

6 Método e Planeamento

6.1 Planeamento inicial

O plano de trabalho e o cronograma proposto para o Trabalho Final de Curso foram realizados em formato Gant, utilizando a aplicação [ProjectLibre]para o planeamento do trabalho. Além disso, foi utilizado o [Redmine] para a gestão das tarefas atribuídas semanalmente pela CyberS3c, permitindo um desenvolvimento contínuo do projeto.
A dificuldade inicial do projeto foi a compreensão do mesmo, uma vez que trabalhar num projeto com uma empresa introduz alguma complexidade até se alcançar o alinhamento necessário. Entender e instalar as tecnologias iniciais foi um processo complicado, sobretudo porque nunca tínhamos tido contacto prévio com a área de segurança cibernética.

Lista de Entregáveis:

- Entrega do Relatório Intercalar 1.º Semestre
- Entrega do Relatório Intercalar 2.º Semestre
- Entrega do Relatório Final
- Entrega do Projeto
-

Lista de Tarefas:

Para uma melhor compreensão do planeamento, segue abaixo uma Figura 30-Gant que apresenta o mesmo, elaborada no [ProjectLibre], representando o planeamento provisório e inicial definido por nós em colaboração com a CyberS3c.

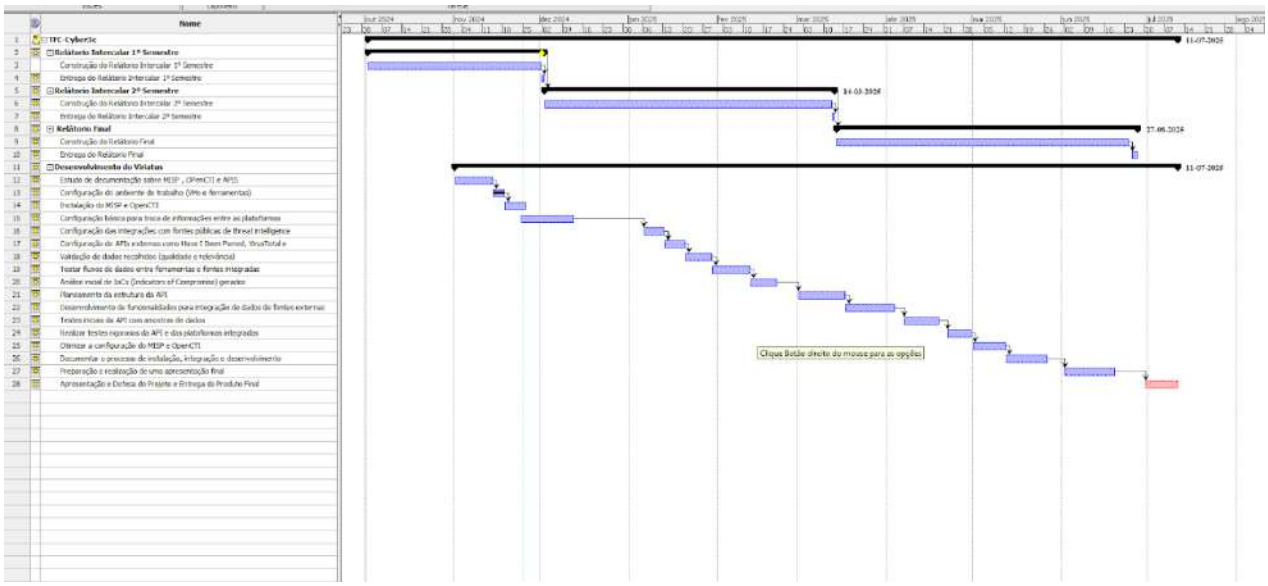


Figura 30-Gant

6.1.1 Planeamento Orientado à Disponibilização Pública

Este plano detalha uma abordagem estratégica para a hipotética disponibilização pública da solução construída. O foco principal reside na realização de testes

abrangentes, na garantia de qualidade e aceitação por parte dos utilizadores, e na preparação meticulosa para a eventual implementação em ambiente real.

Para estruturar o calendário necessário a este planeamento, será demonstrada a utilização do Gráfico de Gantt, uma ferramenta amplamente reconhecida na gestão de projetos. A metodologia de construção do Gantt será similar à aplicada no planeamento inicial do projeto, permitindo uma visualização clara das fases, tarefas e prazos envolvidos neste processo de preparação para a disponibilização.

Lista de Entregáveis:

- Plano de Testes Abrangente
- Critérios de Qualidade e Aceitação
- Plano de Implementação
- Documentação de Suporte e Utilizador
- Relatórios de Validação e Aceitação
- Preparação do Ambiente de Produção
- Definição do Plano de Suporte e Manutenção

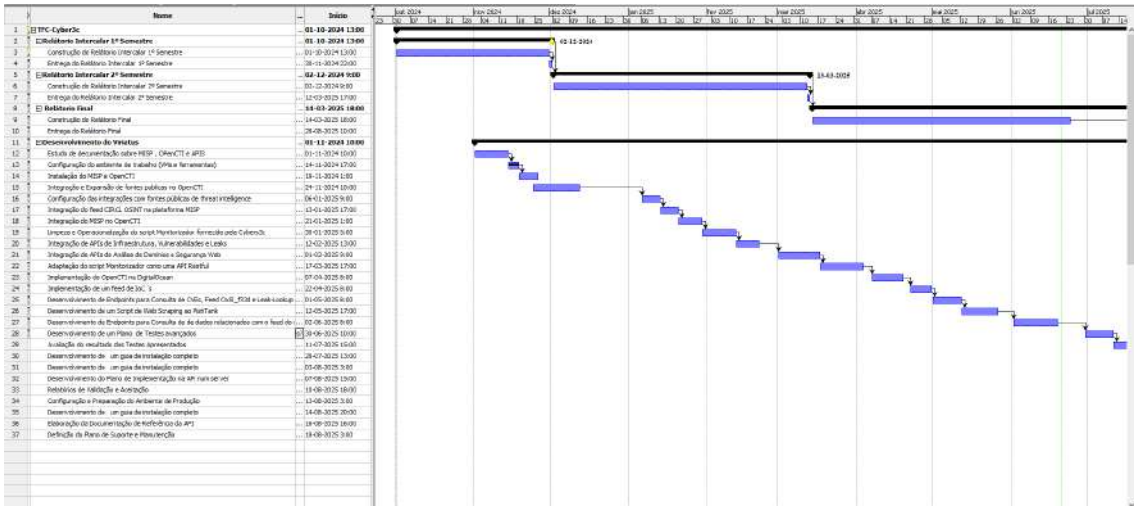


Figura 31-Gant

6.1.2 Análise Crítica ao Planeamento

Este capítulo analisa e critica o planeamento do projeto, focando-se em como o progresso foi acompanhado, nas dificuldades encontradas e nas alterações que foram introduzidas ao plano inicial.
O acompanhamento do progresso do projeto foi realizado com a plataforma Redmine, que se revelou fundamental para o controlo e gestão das tarefas. Através desta ferramenta, os pontos principais do projeto foram registados como assuntos, servindo como uma referência central para o acompanhamento das atividades. Além disso, o Redmine permitiu a criação de subtarefas à medida que surgiam novas necessidades, bem como a atualização contínua do progresso. Esta abordagem garantiu que o trabalho fosse monitorizado de forma eficaz, permitindo uma gestão mais ágil e adaptativa.

Adicionalmente, foram realizadas duas reuniões semanais com a empresa parceira CyberS3c, o que contribuiu significativamente para a resolução de dúvidas e para a discussão do progresso do trabalho. Estas reuniões asseguraram um acompanhamento contínuo e eficaz, permitindo uma rápida adaptação e resolução de eventuais obstáculos.

A nível organizacional, a colaboração com a CyberS3c foi um fator importante para a boa gestão do tempo e a inexistência de problemas relacionados com a coordenação do trabalho. Este suporte foi fundamental para que o planeamento inicial fosse maioritariamente seguido, sem alterações significativas. No entanto, surgiram alguns imprevistos que tiveram impacto no progresso do projeto.

Entre os principais imprevistos, destacaram-se problemas com APIs externas que, em alguns casos, já não se encontravam operacionais ou não retornavam a informação necessária. Por exemplo, a API Fishtank deixou de funcionar corretamente, o que obrigou à procura de alternativas viáveis. Além disso, a API Shodan, utilizada para recolher dados sobre domínios, não fornecia informação suficientemente relevante, o que levou à necessidade de investigar outras soluções. Embora estas limitações tenham causado atrasos e exigido mais tempo de pesquisa e adaptação, foi possível ultrapassá-las através da implementação de alternativas adequadas.

A dificuldade mais significativa do projeto surgiu durante a configuração do OpenCTI e a sua integração com o MISP. Esta plataforma exigia configurações específicas e a utilização de um sistema operativo Ubuntu Server, com o qual a equipa não possuía experiência prévia. Acresce que a arquitetura da máquina era limitada à arquitetura x86, o que obrigou, após a instalação inicial do OpenCTI, a reinstalar o sistema numa máquina compatível com essa arquitetura. Após esta instalação, verificou-se que a máquina com o MISP não conseguia comunicar com o servidor onde o OpenCTI estava instalado. Após diversas tentativas e testes, concluiu-se que a melhor solução seria hospedar ambas as máquinas num servidor *cloud*, mais concretamente na DigitalOcean, tirando partido de um crédito gratuito de 200 dólares. Esta mudança estratégica implicou uma nova fase de investigação sobre o funcionamento da plataforma e uma nova reinstalação de todas as máquinas e respetivas configurações.

Contudo, o maior contratempo surgiu quando o OpenCTI esgotou toda a memória disponível no servidor *cloud*, o que resultou na perda dos contentores associados e na necessidade de uma nova reinstalação completa. Este imprevisto teve um impacto considerável no progresso do projeto, uma vez que exigiu tempo e recursos adicionais para ser resolvido.

Outra alteração relevante ao plano inicial foi a introdução dos novos *endpoints* monitorizadorweb, desenvolvidos com o método GET, resultante de uma mudança de visão por parte da empresa parceira. Esta alteração implicou a redefinição de parte do planeamento do projeto, bem como a introdução de dois novos requisitos funcionais, representando uma revisão do planeamento inicial para acomodar essas novas exigências.

Apesar destes desafios, o projeto foi desenvolvido com sucesso. Os *endpoints* já se encontram numa versão estável e cumprem os critérios de aceitação definidos. O planeamento inicial foi cumprido na sua maioria, com exceção da introdução das reuniões semanais e da utilização do Redmine para acompanhar o progresso de forma mais eficaz, o que se revelou uma melhoria significativa no processo.

#	Tipo	Estado	Prioridade	Assunto	Atribuído a	Alterado
1462	Support	In Progress	Normal	Implementação de um feed de IoC's	Miguel Lourenço	22/04/2025 17:03
1434	Support	In Progress	Normal	Obter informação através das API's web-check, security-trails, blacklist-checker	Miguel Lourenço	18/03/2025 20:46
1433	Support	Resolved	Normal	Obter informação através das API's URLscan, HTTP observatory e Virus total	Vasco Pereira	18/03/2025 17:01
1412	Support	In Progress	Normal	Adaptação do script Monitorizador e construção de uma API RESTful	Vasco Pereira	07/03/2025 19:07
1379	Support	In Progress	Normal	Obter informação sobre uma CVE com ExploitDB	Miguel Lourenço	17/04/2025 16:36
1346	Support	In Progress	Normal	Obter informação através da API do Leak-Lookup	Miguel Lourenço	18/03/2025 17:02
1345	Support	In Progress	Normal	Obter informação através da API do PhishTank	Miguel Agostinho	07/03/2025 15:53
1272	Support	Closed	Normal	Obter informação através da API do HaveIBeenPwned	Miguel Lourenço	16/12/2024 11:05
1269	Support	In Progress	Normal	Obter informação através da API do Shodan	Vasco Pereira	07/03/2025 15:50
1264	Support	Resolved	Normal	Integração do MISP no OpenCTI	Vasco Pereira	12/01/2025 23:34
1263	Support	Resolved	Normal	Integração de novas fontes publicas no OpenCTI	Miguel Lourenço	08/12/2024 14:36
1256	Support	Resolved	Normal	Integração do feed CIRCL OSINT na plataforma MISP	Vasco Pereira	29/11/2024 00:28
1255	Support	Resolved	Normal	Integração de fontes publicas no OpenCTI	Miguel Lourenço	30/11/2024 18:01
1233	Support	Resolved	Normal	Instalação do OpenCTI numa VM de Testes	Miguel Lourenço	06/12/2024 22:15
1232	Support	Resolved	Normal	Instalação do MISP em Máquina Virtual	Vasco Pereira	07/12/2024 18:34

Figura 32-Tarefas Redmine

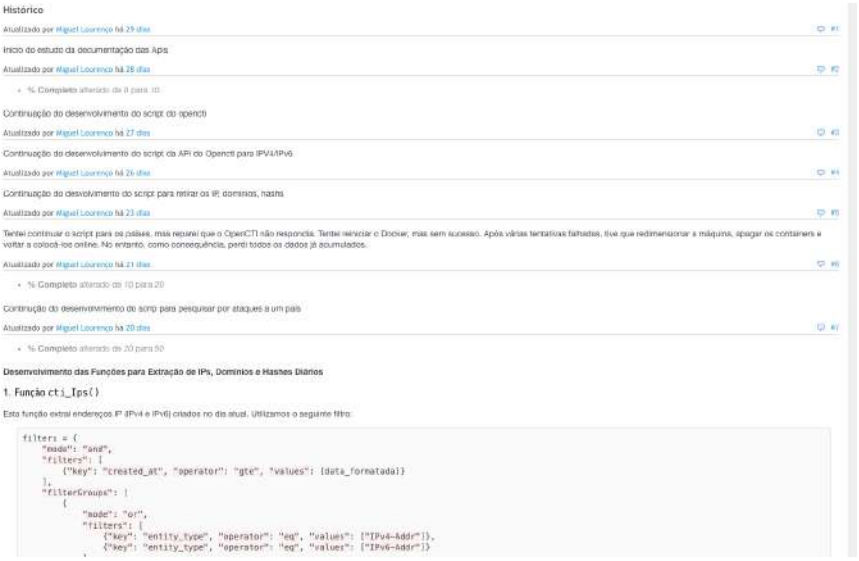


Figura 33-Exemplo de histórico da tarefa

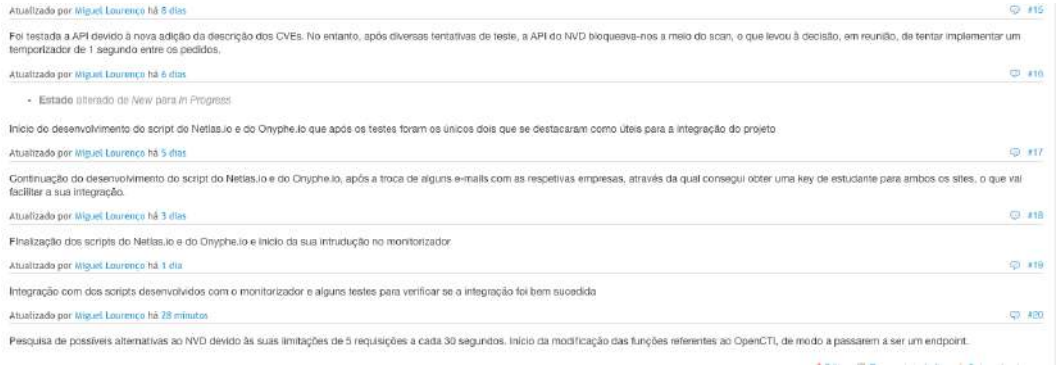


Figura 34-Exemplo de histórico da tarefa

Em suma, o projeto foi concluído com sucesso, apesar das dificuldades e imprevistos que surgiram ao longo do seu desenvolvimento. As alterações ao plano inicial, como a migração para um servidor *cloud* e a reconfiguração dos sistemas, revelaram-se necessárias para ultrapassar os desafios técnicos. O acompanhamento contínuo, através das reuniões semanais e do uso da plataforma Redmine, permitiu manter o controlo

sobre o progresso e garantir a execução eficiente das tarefas. Embora tenham sido necessários alguns ajustamentos, o planeamento inicial foi, em grande medida, respeitado e as dificuldades enfrentadas foram superadas de forma eficaz, sem comprometer os objetivos definidos pela equipa e pela entidade parceira.

7 Resultados

7.1 Resultados dos Testes

Este capítulo analisa os resultados obtidos através dos testes realizados. Efetuamos um conjunto de testes funcionais à API desenvolvida para validar seu comportamento em diferentes cenários de utilização. Estes testes incidiram sobre consultas a domínios, endereços IP, CVEs e indicadores de ameaças cibernéticas, simulando interações típicas dos utilizadores e avaliando a resiliência da API perante erros e falhas externas, como a indisponibilidade de serviços de terceiros.
Todos os testes foram conduzidos internamente, sem a participação de avaliadores externos. Contudo, asseguramos uma avaliação criteriosa de cada requisito funcional, com registo rigoroso dos *inputs*, ações executadas, *outputs* esperados e resultados efetivamente obtidos.
A realização exaustiva de testes ao longo do desenvolvimento do projeto foi um dos principais requisitos da empresa parceira, com o intuito de garantir que cada fase cumprisse os objetivos definidos. Os resultados apresentados a seguir refletem os testes descritos no Capítulo 5 e detalhados no Anexo A.

7.1.1 Resultados Detalhados dos Testes:

A tabela seguinte resume os resultados obtidos na avaliação funcional dos *endpoints* /monitorizador/DOM/ e /monitorizadorweb/DOM/, correspondentes aos testes T01 a T07 e T14. Estes testes foram concebidos para verificar o comportamento da API perante diferentes tipos de entrada (domínios válidos, inválidos, mistos), cenários de erro (utilização incorreta de métodos HTTP ou *endpoints* mal formados) e indisponibilidade de fontes externas.

Tabela 4-monitorizador DOM

Cenário	Outputs	Outcame
Domínio válido	ulusofona.pt	JSON completo com informação do domínio e subdomínios
Domínio inválido	ulufona.tfdv	JSON vazio, sem erro
Domínios válidos e inválidos	lusfona.pt, ulusofona.pt	JSON com dados apenas dos domínios válidos
API externa indisponível	ulusofona.pt	JSON com os dados das fontes disponíveis
Método HTTP incorreto (GET em vez de POST)	ulusofona.tp	Mensagem “Método não permitido” + código 405
Endpoint incorreto (/DOT em vez de /DOM)	ulusofona.pt	Mensagem de erro e código 400 predefinido

Os testes realizados aos *endpoints* de consulta de domínios demonstraram que a API se comporta de forma robusta e consistente. Foram corretamente tratadas as situações de

sucesso (com domínios válidos), falhas previsíveis (como domínios inválidos ou métodos incorretos), e exceções (como a falha de uma API externa). Todas as respostas devolvidas foram adequadas ao cenário previsto, sem a ocorrência de falhas inesperadas, garantindo a fiabilidade e estabilidade da solução implementada.
Estes resultados validam o correto cumprimento dos requisitos definidos para esta funcionalidade.

A tabela seguinte apresenta os resultados obtidos nos testes aos *endpoints* de consulta de IPs, nomeadamente /monitorizador/IP/ e /monitorizadorweb/IP/. Correspondentes aos testes T08 a T10 e T15, estes testes visam verificar o comportamento da API perante IPs válidos, inválidos ou mal formatados, bem como a sua capacidade de filtrar corretamente os dados devolvidos quando é fornecida uma mistura de IPs válidos e inválidos.

Tabela 5- Monitorizador IP

Cenário	Outputs	Outcames
IP válido (IPv4)	213.58.148.218	JSON com toda a informação sobre o IP
IP inválido ou mal formatado	1.1.1.fd	JSON vazio, sem erro
IPs mistos (válidos e inválidos)	65.21.239.46 fe80::10c3:4ecd:3114:baef%en6 1.1.1. fd 34	JSON contendo apenas o IP válido (API suporta apenas IPv4)

Os testes realizados ao *endpoint* de consulta de IPs demonstram um comportamento robusto e consistente da API. Foi confirmada a capacidade do sistema para validar corretamente endereços IPv4, rejeitar entradas inválidas sem gerar erros, e lidar com múltiplos *inputs* de forma eficaz. Além disso, ficou demonstrado que a API está preparada para ignorar *inputs* não suportados, como endereços IPv6, mantendo a estabilidade e precisão dos dados devolvidos.
Estes resultados validam o correto cumprimento dos requisitos definidos para esta funcionalidade.

A tabela seguinte apresenta os resultados obtidos nos testes aos *endpoints* /pesquisar-cve/ (pesquisa por *software*) e /CVE/ (consulta por identificador específico de vulnerabilidade), correspondentes aos testes T11, T12 e T13. Estes testes visam validar a resposta da API perante a pesquisa de *software* existente ou inexistente, bem como a obtenção detalhada de informação sobre um CVE específico.

Tabela 6- CVEs

Cenário	Output	Outcame
Software existente	Font Awesome	JSON com todos os CVEs associados

Software inexistente	UBornto 20.04	JSON vazio, sem erro
Consulta a CVE específico	CVE- 2021-44228	JSON com todos os detalhes sobre o CVE pedido

Os testes realizados demonstram que a API lida corretamente com diferentes tipos de consulta ao repositório de vulnerabilidades (CVEs). A funcionalidade de pesquisa devolve os resultados esperados tanto para *software* conhecido como para nomes inexistentes, tratando as exceções de forma silenciosa e controlada (sem erros inesperados). A consulta direta por identificador CVE revelou-se eficaz, retornando dados completos e detalhados.
Estes resultados validam o correto cumprimento dos requisitos definidos para esta funcionalidade.

A tabela seguinte apresenta os resultados obtidos nos testes ao *endpoint* de consulta de *data leaks*, nomeadamente /LOOKUP/, correspondentes ao teste T13. Estes testes visam verificar o comportamento da API perante domínios com e sem histórico de vazamentos de dados, assegurando que o sistema responde adequadamente em ambos os casos.

Tabela 7- Lookup

Cenário	Output	Outcame
Domínio com histórico de leaks	lusofona.pt	JSON com todos os registos de dataleaks relacionados com o domínio
Domínio sem histórico de leaks	-	JSON vazio, sem erro.

Os testes realizados demonstram que a API é capaz de identificar corretamente domínios com registos de *data leaks*, apresentando essa informação de forma completa e estruturada. Quando não existem dados disponíveis, a resposta da API mantém-se consistente, devolvendo um JSON vazio sem gerar erros ou comportamentos inesperados. Estes resultados validam o correto cumprimento dos requisitos definidos para esta funcionalidade.

A tabela seguinte apresenta os resultados obtidos nos testes aos *endpoints* da componente de inteligência de ameaças cibernéticas, nomeadamente /0xSl_f33d/, /ctiTOP10/, /ctiweb/ e /ctipais/, correspondentes aos testes T16 a T26. Estes testes visam verificar o comportamento da API na disponibilização de dados atualizados sobre indicadores de *phishing*, ciberataques a países, e informações relacionadas com *hashes*, IPs e domínios suspeitos, assegurando que o sistema responde adequadamente em todos os casos.

Tabela 8- Feeds

Cenário	Output	Outcame
Indicadores de phishing em domínios portugueses	-	JSON com domínios suspeitos (OxSl_f33d)

Ciberataques a um pais durante um período de tempo	Portugal, 04-2025, 05-2025 Polland,	JSON com os ataques, ao pais durante o tempo determinado
Hashes, IPs e domínios suspeitos	-	JSON com dados relevantes Diários
Ciberataques a um pais durante um período de tempo	Portugal, 07-2025	JSON vazio sem qualquer erro.

Os testes realizados aos *endpoints* de *Cyber Threat Intelligence* demonstram que a API integra corretamente diversas fontes externas, disponibilizando dados atualizados sobre ciberameaças. A resposta das APIs foi consistente e adequada em todos os cenários testados, fornecendo informações estruturadas sobre indicadores de *phishing*, ataques a países num dado período, e dados diários sobre *hashes*, IPs e domínios suspeitos. Estes resultados validam o correto cumprimento dos requisitos definidos para esta funcionalidade.

7.1.2 Conclusão dos Resultados dos Testes

Os resultados obtidos no conjunto global de testes revelaram que os *endpoints* desenvolvidos responderam de forma consistente e conforme o esperado em todos os cenários testados. Essa consistência valida a fiabilidade da solução e reforça a confiança na relevância da informação disponibilizada para efeitos de monitorização e análise de ameaças em contexto real.
Os testes foram realizados repetidamente ao longo de todas as fases do desenvolvimento do projeto, garantindo que eventuais alterações no código ou na infraestrutura não comprometessem o funcionamento correto da API. Essa abordagem contínua à validação foi alinhada com as exigências da empresa parceira, que destacou desde o início a importância de assegurar a estabilidade e robustez da solução em ambientes reais.

7.1.3 Avaliação de Desempenho e Utilização de Recursos

No que diz respeito ao desempenho, monitorizamos o tempo de resposta e a eficiência no processamento das consultas. Os resultados estão descritos abaixo, o que evidencia a capacidade da API para responder de forma eficaz às diferentes solicitações.

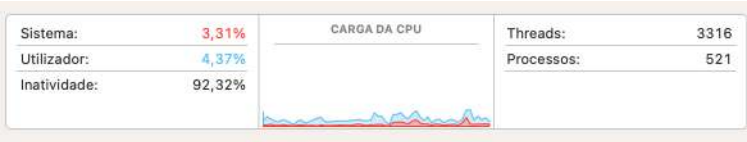


Figura 35-Utilização do CPU

Figura 36-Utilização da Internet

Figura 37-Utilização da RAM

Com base nas imagens apresentadas, é possível constatar que, ao longo da execução dos testes, os principais recursos computacionais utilizados foram a memória RAM e a ligação à internet, enquanto o processador manteve níveis de utilização relativamente baixos. Esta observação permite inferir que a latência verificada nas respostas da API está, maioritariamente, associada à qualidade da ligação à internet, e não ao processamento local.

Assim, conclui-se que a velocidade de resposta da API é fortemente influenciada pela capacidade e desempenho da conexão de rede do *host*. Em ambientes com ligações instáveis ou de baixa largura de banda, é expectável um aumento do tempo de resposta, mesmo que os recursos locais estejam disponíveis. Este fator deverá ser tido em consideração na eventual migração da API para produção ou em cenários com elevada concorrência de pedidos.

7.2 Cumprimento de requisitos

Tabela 9- Cumprimentos de requisitos funcionais

Requisito	Grau de Realização
REQ-01- Instalação do MISP numa Máquina Virtual de Testes	Realizado
REQ-02- Instalação do OpenCTI numa Máquina Virtual de Testes	Realizado
REQ-03- Integração e Expansão de fontes publicas no OpenCTI	Realizado
REQ-04- Integração do feed CIRCL OSINT na plataforma MISP	Realizado
REQ-05- Integração do MISP no OpenCTI	Realizado parcialmente

REQ-06- Limpeza e Operacionalização do script Monitorizador fornecido pela Cybers3c	Realizado
REQ-07- Integração de APIs de Infraestrutura, Vulnerabilidades e Leaks	Realizado
REQ-08- Integração de APIs de Análise de Domínios e Segurança Web	Realizado
REQ-09- Adaptação do script Monitorizador como uma API Restful	Realizado
REQ-10- Implementação do OpenCTI na DigitalOcean	Realizado
REQ-11- Implementação de um feed de IoC´s	Realizado
REQ-12- Desenvolvimento de Endpoints para Consulta de CVEs, Feed OxSl_f33d e Leak-Lookup	Realizado parcialmente
REQ-13- Desenvolvimento de um Script de Web Scraping ao PishTank	Abandonado
REQ-14- Desenvolvimento de Endpoints para Consulta de dados relacionados com o feed do opencti	Realizado

Tabela 10- Cumprimento de requisitos não funcionais

Requisito	Grau de Realização
REQ-15- A API deverá suportar, de forma estável, o grande volume de dados recebidos	Realizado
REQ-16- A API deverá ser capaz de normalizar os dados recebidos de diversas plataformas	Realizado

REQ-17- A API deverá incluir um guia de instalação detalhado	Realizado
REQ-18- A API deverá conter um guia de utilização completo	Realizado
REQ-19- Implementar limites de requisições por IP ou chave de API para evitar ataques de Denial of Service	Realizado
REQ-20- Implementar limites de requisições por IP ou chave de API para evitar ataques de Denial of Service	Não realizado

A seguir, apresentamos a justificação para os requisitos que não atingiram o grau de "realizado". Apesar de a maioria ter sido desenvolvida com o apoio da empresa parceira, o que resultou em grande parte do projeto ser concluído com sucesso, pequenos imprevistos levaram à realização parcial, abandono ou não realização dos requisitos 5, 12, 13 e 20.

REQ-05

O objetivo inicial do requisito 5 era conectar o MISP, que estaria em uma máquina diferente e em redes distintas. No entanto, devido a problemas de bloqueio no *router* de um dos membros da equipa, não foi possível executar integralmente o plano original de conectar os dois sistemas em computadores e redes separadas.
Como alternativa, a conexão do MISP com o OpenCTI foi realizada em máquinas virtuais distintas, mas na mesma rede. Por essa razão, o objetivo inicial foi alterado, e o requisito é considerado **parcialmente realizado**.

REQ-12

No requisito 12, o objetivo inicial era desenvolver os *endpoints* Consulta de CVEs, Feed OxSl_f33d e Leak-Lookup, os quais foram concluídos com sucesso. No entanto, o problema surgiu na Consulta de CVEs, onde a meta era, a partir da pesquisa de uma tecnologia específica, obter todos os seus CVEs associados, bem como a respetiva descrição. Embora isto tenha sido inicialmente realizado com sucesso, o principal suporte para o funcionamento deste *endpoint* é a NVD (*National Vulnerability Database*). Devido a recentes tensões políticas nos Estados Unidos, a NVD perdeu financiamento e, consequentemente, tem apresentado uma API instável em termos de disponibilidade, o que torna a acessibilidade deste *endpoint* por vezes incerta.
Apesar da procura por algumas alternativas, não foi encontrada até à data uma solução que permita a substituição integral. Isso torna este requisito **parcialmente realizado**

REQ-13

No requisito 13, tínhamos como objetivo desenvolver um *script* para o Phishtank. No entanto, desde o início de dezembro, o *website* desativou a criação de contas para novos membros, o que impossibilitou o plano original.
Apesar disso, um *script* de *web scraping* foi desenvolvido com o intuito de contornar a situação. Contudo, por aconselhamento da empresa parceira, foi recomendado não continuar com o desenvolvimento, uma vez que o *website* poderia bloquear o acesso devido ao *scraping*. Assim, o projeto foi **abandonado**.

REQ-20

No requisito 20, o objetivo inicial era assegurar que a API estivesse protegida contra ataques de *Denial of Service* (DoS), entre outros, o que implicaria a implementação de tais medidas assim que a API fosse disponibilizada na *internet*. No entanto, após reuniões com a empresa parceira, foi-nos indicado que a API ficaria disponível apenas internamente nos seus sistemas.

Por esse motivo, não foi considerada necessária a implementação de métodos de limitação de requisições nem a adição de *keys* de utilizador, o que levou à **não realização** desses requisitos.

8 Conclusão

8.1 Conclusão

Este Trabalho Final de Curso (TFC) permitiu desenvolver e integrar um conjunto de ferramentas e fontes de inteligência orientadas para a análise e monitorização de domínios e endereços IP, com base em dados OSINT (Open Source Intelligence). O projeto teve como principal objetivo disponibilizar uma visão abrangente sobre a superfície de exposição de ativos online, recorrendo a diversas fontes públicas de informação.

Além da API para monitorização de domínios e IPs, que constitui o núcleo central da solução, o projeto integrou também plataformas de *threat intelligence* como o MISP (*Malware Information Sharing Platform*) e o OpenCTI (*Open Cyber Threat Intelligence*). Isso permitiu enriquecer o ecossistema de dados e estabelecer ligações relevantes entre indicadores. Foram ainda utilizadas outras fontes OSINT, como *feeds* de segurança nacionais (nomeadamente o fornecido pelo blog Segurança Informática), bem como serviços públicos e bases de dados relativas a *data leaks* e CVEs, que permitiram identificar vulnerabilidades conhecidas associadas a tecnologias ou ativos específicos. A integração desses elementos contribuiu significativamente para reforçar a capacidade de análise da ferramenta e apoiar a tomada de decisões em contextos de segurança ofensiva e defensiva.

Em termos de concretização do plano inicialmente delineado, grande parte dos objetivos propostos foi atingida. Contudo, ao longo do desenvolvimento, foi necessário fazer alguns ajustamentos, o que é expectável num projeto com esta complexidade e com uma forte componente prática. Alguns requisitos acabaram por não ser totalmente implementados, sobretudo devido à indisponibilidade de determinadas APIs em formato gratuito ou acessível, o que obrigou à redefinição de prioridades e à adaptação do âmbito do trabalho.

O projeto manteve-se fiel ao plano inicial, sofrendo poucas alterações ao longo do desenvolvimento. A principal evolução foi a adição de novos *endpoints* e a integração de APIs adicionais, muitas delas sugeridas ou implementadas por iniciativa própria, com o objetivo de tornar a solução mais completa e funcional.

Durante o desenvolvimento, registou-se uma evolução significativa tanto a nível técnico como prático. Ganhamos experiência prática em diversas áreas, nomeadamente na análise de *banners*, certificados SSL, listas negras (*blacklists*), CVEs, tratamento de dados de segurança e no manuseamento de uma plataforma de *threat intelligence*. A nível técnico, foi possível reforçar competências na linguagem Python, no consumo e integração de APIs, técnicas de *scraping*, manipulação e normalização de dados. Em termos de *cloud*, adquirimos conhecimentos no *deployment* de infraestruturas, nomeadamente ao implementar uma máquina virtual com o OpenCTI a correr num ambiente *cloud*.

Além disso, evoluímos também na gestão de requisitos, execução de testes e no nosso primeiro contacto direto com um cliente real (CyberS3c), o que acrescentou uma dimensão prática e comunicacional importante ao projeto. No âmbito das ferramentas e comandos utilizados, destacam-se as experiências práticas com *scanners* de rede como o Nmap e o Masscan, essenciais para a recolha de informação, incluindo a identificação de portas abertas, análise de registos DNS e cabeçalhos HTTP, fundamentais para a avaliação de segurança.

Se o TFC voltasse ao início, uma das principais melhorias passaria por uma definição mais clara e rigorosa dos requisitos desde as fases iniciais. Verificamos que, já numa fase avançada do projeto, surgiram alterações por parte da entidade parceira (CyberS3c), o que implicou refatorações substanciais e não planeadas. Uma melhor delimitação dos objetivos e das expectativas desde o início teria evitado ambiguidades e teria sido particularmente útil na fase final de validação. Além disso, iniciar mais cedo o desenvolvimento da API teria permitido uma fase de testes mais extensa e um maior número de iterações e melhorias.

Entre as principais dificuldades, destacam-se a adaptação de código externo (nomeadamente o *script* fornecido pela CyberS3c**,** que exigiu modificações relevantes para se integrar na lógica da API) e a integração de diversas APIs públicas com estruturas de resposta heterogéneas. Por fim, garantir o desempenho da API em chamadas simultâneas e perante grandes volumes de informação revelou-se um desafio técnico importante**,** que exigiu a introdução de *threads*, eliminação de redundâncias e reorganização de estruturas condicionais complexas que comprometiam a *performance*.

Em síntese, este projeto constituiu uma experiência profundamente enriquecedora**,** não apenas a nível técnico, mas também no desenvolvimento de competências de planeamento, adaptação e colaboração com entidades externas, culminando numa solução funcional e com aplicabilidade real no contexto da cibersegurança.

8.2 Trabalhos Futuros

O futuro do projeto depende essencialmente das decisões estratégicas da CyberS3c, já que a continuidade e evolução da API são de sua responsabilidade direta. A versão atual da API está funcional e concluída, cumprindo os requisitos definidos para esta fase. No entanto, a continuação da parceria com a instituição de ensino, bem como a evolução da solução, estará sujeita à vontade e disponibilidade da CyberS3c em manter essa colaboração.
Caso a parceria se prolongue, abre-se a possibilidade de futuros alunos darem seguimento ao trabalho desenvolvido, contribuindo para seu aperfeiçoamento técnico e funcional. Entre os possíveis desenvolvimentos futuros, destacam-se a criação de novos *endpoints* focados na correlação avançada de dados, o que permitirá uma análise mais rica e contextualizada dos indicadores de ameaça.
Além disso, pode ser considerada a implementação de uma vertente gráfica complementar, nomeadamente através do desenvolvimento de uma aplicação *web*. Esse componente visual teria como objetivo facilitar a interpretação dos dados gerados

pela API, permitindo navegação mais intuitiva, agregação visual de informação e, eventualmente, funcionalidades de filtragem, alerta e *reporting* em tempo real. Tal evolução contribuiria para tornar a solução mais acessível a utilizadores com menor literacia técnica e reforçaria sua aplicabilidade em contextos operacionais.

Bibliografia

[CyberS3c] Sobre nós - CyberS3c. (2023, August 8). CyberS3c. https://www.cybers3c.pt/sobre-nos/ , acedido em Nov.2024.

[NIS 2] *NIS 2 Directive*. (n.d.). Www.nis-2-Directive.com. https://www.nis-2-directive.com/, acedido em Abr.2025.

[ProjectLibre] Our products | Projectlibre. (n.d.). Www.projectlibre.com. https://www.projectlibre.com/products , acedido em Nov.2024

[Redmine] Overview - Redmine. (n.d.). Www.redmine.org. https://www.redmine.org/ ,acedido em Nov.2024

[ODS9] Indústria, Inovação e Infraestruturas • ODS - BCSD Portugal. (n.d.).https://ods.pt/objectivos/9-inovacao-e-infraestruturas/ , Nov.2024

[ODS16] Paz e Justiça • ODS - BCSD Portugal. (n.d.). https://ods.pt/objectivos/16-paz-e-justica/
, acedido em Nov.2024

[Luís Rato] Security, I. T. (n.d.). Como a inteligência artificial impacta a cibersegurança. IT Security. https://www.itsecurity.pt/news/analysis/como-a-inteligencia-artificial-impacta-a-ciberseguranca , acedido em Nov.2024

[IDC] securityMagazine. (2024, October 30). *IDC prevê que investimento em cibersegurança atinja 250 milhões em Portugal até ao final do ano - Security Magazine*. Security Magazine. https://www.securitymagazine.pt/2024/10/30/idc-preve-que-investimento-em-ciberseguranca-atinja-250-milhoes-em-portugal-ate-ao-final-do-ano/?utm_source=chatgpt.com, acedido em Abr.2025

[IBM] *IBM Report: Escalating Data Breach Disruption Pushes Costs to New Highs*. (2024). IBM Newsroom. https://newsroom.ibm.com/2024-07-30-ibm-report-escalating-data-breach-disruption-pushes-costs-to-new-highs? ,acedido em Abr.2025

[crescimento anual] *Canalys Newsroom - Heightened threat levels drive cybersecurity spending to US$19 billion in Q2 2023*. (2023). @Canalys. https://www.canalys.com/newsroom/cybersecurity-market-Q2-2023? , acedido em Abr.2025

[Expresso] Oliveira, T. (2024, July 15). *37% das empresas portuguesas pretende investir até €30 mil em cibersegurança*. Expresso. https://expresso.pt/iniciativaseprodutos/projetos-expresso/2024-07-15-37-das-empresas-portuguesas-pretende-investir-ate-30-mil-em-ciberseguranca-d5d1e955 ,acedido em Abr.2025

[Check Point] Europa, T., & Europa, T. (2021, January 29). *72% das empresas portuguesas não têm políticas de cibersegurança*. TV Europa. https://www.tveuropa.pt/noticias/72-das-empresas-portuguesas-nao-tem-politicas-de-ciberseguranca/? , acedido em Abr.2025

[DORA] PricewaterhouseCoopers. (2023). *Digital Operational Resilience Act | Advisory | Serviços | PwC Portugal*. PwC. https://www.pwc.pt/pt/servicos/advisory/digital-operational-resilience-act.html , acedido em Abr.2025

[Cybersecurity Ventures] *Topic: Cybersecurity in Europe*. (2024). Statista. https://www.statista.com/topics/12924/cybersecurity-in-europe/#topicOverview ,acedido em Abr.2025

[Statista Research Department] Stu Sjouwerman. (2023, January 21). *Cybercrime The World's Third Largest Economy After the U.S. and China*. Knowbe4.com; KnowBe4, Inc. https://blog.knowbe4.com/cybercrime-the-worlds-third-largest-economy-after-the-u.s.-and-china ,acedido em Abr.2025

[Portugal Digital Awards] *Finalistas 2024 - Portugal Digital Awards*. (2024, November 27). Portugal Digital Awards. https://www.portugaldigitalawards.pt/finalistas-2024/ ,acedido em Abr.2025

[DORA] EIOPA. (2025). *Digital Operational Resilience Act (DORA)*. Www.eiopa.europa.eu. https://www.eiopa.europa.eu/digital-operational-resilience-act-dora_en , acedido em Abr.2025

[Segurança Informática] Segurança Informática. (2025). Feed de notícias sobre segurança informática. *www.seguranca-informatica.pt*. https://feed.seguranca-informatica.pt/, acedido em Jun.2025.

Anexo A

Este anexo dedica-se a detalhar as intervenções significativas realizadas no *script* Monitorizador original da CyberS3c. Este *script*, que na sua versão inicial apresentava desafios de funcionalidade e eficiência, bem como limitações estruturais, foi alvo de uma análise exaustiva e de uma profunda refatoração. O principal propósito desta intervenção foi não só torná-lo plenamente funcional e eficiente, resolvendo os problemas inerentes de execução, dependências e estrutura de código, mas também expandir significativamente as suas capacidades. Isto incluiu a adição de diversas novas fontes de informação, enriquecendo os dados recolhidos e ampliando a cobertura de monitorização. Mais crucialmente, o *script* foi submetido a uma reestruturação completa para o transformar numa API (Application Programming Interface). Esta metamorfose confere-lhe maior modularidade, facilita a sua integração com outros componentes da solução e permite uma interação padronizada, robusta e escalável.

Contexto do Problema Original

O *script* Monitorizador original, fornecido pela CyberS3c, apresentava uma série de falhas e limitações que comprometiam a sua funcionalidade e aplicabilidade no contexto do projeto. Tratava-se de código com cerca de quatro anos de desenvolvimento, contendo já funcionalidades não operacionais.
No seu estado inicial (AS-IS), o código fornecido não funcionava corretamente para os dois modos de varrimento previstos (IPs e domínios). Carecia de otimização, e o acesso às suas funcionalidades era restrito ao terminal, dado que não existia uma interface programática (API). O armazenamento dos dados varridos era realizado numa base de dados local, o que comprometia a escalabilidade e dificultava a manutenção do sistema. Adicionalmente, a ausência de documentação e de mecanismos adequados de tratamento de erros impunha significativas limitações operacionais. As integrações com plataformas externas de *Threat Intelligence* eram limitadas a Shodan e crt.sh (para subdomínios), restringindo o alcance e a profundidade da análise.

Gestão de Dependências (requirements.txt): O ficheiro requirements.txt, responsável por listar as dependências do *script*, encontrava-se significativamente incompleto. Esta lacuna dificultava a instalação correta e a gestão das bibliotecas necessárias para o seu funcionamento.

Problemas de Fiabilidade na Escrita para a Base de Dados:

- O módulo principal do *script*, Monitorizador, era originalmente concebido para retornar dados para uma base de dados interna, mas nem sempre o fazia de forma fiável.
- O MonitorizadorDomínio frequentemente não retornava a informação para a base de dados e, quando o fazia, nem sempre preenchia todos os campos necessários.
- O MonitorizadorIPs também não registava toda a informação na base de dados interna.

Performance (Tempo de Execução):O *script* apresentava um problema crítico de desempenho, demorando um tempo excessivo para realizar uma pesquisa básica. Esta lentidão tornava a sua utilização impraticável e ineficiente.

Integrações e Funcionalidades Existentes (e as suas Limitações): O *script* original possuía funcionalidades básicas de recolha de informação, nomeadamente para IPs e Domínios. No entanto, a sua implementação e fiabilidade de registo na base de dados eram inconsistentes. As principais capacidades eram:

- **Para IPs:** Procura de portas abertas (utilizando Masscan); Pesquisa de certificados SSL/TLS; Enumeração de protocolos; Verificação do estado das portas (e.g., *open*, *filtered*); Consulta de *blacklists* de IPs. O *script* era concebido para imprimir estes dados e registá-los na base de dados interna.
- **Para Domínios:** Procura de subdomínios (incluindo verificação de certificados) com crt.sh; Verificação de *blacklists* de IPs associados aos domínios; Análise de versões SSL/TLS; Verificação de cabeçalhos de segurança; Deteção de *typosquatting*. Tal como para IPs, a informação recolhida era impressa e destinada a ser registada na base de dados interna. Apesar destas funcionalidades, as integrações e o processo de registo de dados revelavam-se limitados e inconsistentes, necessitando de uma revisão e expansão para o contexto atual do projeto.

Intervenções e Melhorias Implementadas

De modo a superar as problemáticas previamente identificadas e a converter o *script* numa solução significativamente mais robusta e alinhada com as necessidades atuais, foram levadas a cabo as seguintes intervenções e melhorias:

Gestão de Dependências (requirements.txt): O ficheiro requirements.txt, responsável por listar as dependências do *script*, foi totalmente reformulado. Esta intervenção garantiu que todas as dependências necessárias para o funcionamento do *script* ficassem devidamente listadas e geridas, eliminando problemas de instalação e compatibilidade.

Criação de um Módulo de Configuração (configs.yaml): Foi criado um novo módulo, o ficheiro configs.yaml, dedicado à centralização de configurações essenciais. Esta abordagem permite gerir configurações relativas a ferramentas como o Masscan (evitando a necessidade de modificar diretamente o código-fonte) e otimizar a conexão para o OpenCTI, promovendo maior flexibilidade e facilidade de manutenção.

Reestruturação e Refatoração do Código: O código foi extensivamente refatorado e as funções foram documentadas internamente para facilitar a sua compreensão e manutenção. Foram removidas todas as referências diretas a bases de dados, uma vez que o objetivo do trabalho se destinava à criação de uma API, não ao armazenamento interno persistente. A informação recolhida passou a ser estruturada em formato JSON, sendo retornada diretamente por cada *endpoint* da API e podendo ser guardada com o

nome do IP/domínio analisado e a hora da pesquisa. Esta alteração exigiu uma profunda modificação na lógica das funções e adaptações significativas no código.

Otimização de Performance: Para melhorar a otimização do desempenho, foi introduzida a utilização de *threads* em áreas específicas, como nas chamadas a APIs externas para a obtenção de subdomínios e na busca de informação detalhada de cada subdomínio. Estas áreas demonstravam ser particularmente morosas na versão original. Além disso, diversas outras secções do código foram otimizadas, e funcionalidades obsoletas foram removidas para aumentar a eficiência geral.

Transformação para API: O código original, com a sua arquitetura concebida para armazenamento em base de dados relacional, não se alinhava com o novo paradigma de uma API RESTful eficiente. Embora uma adaptação pudesse ser considerada, a lentidão inerente motivou uma conversão completa da lógica de gestão de dados para o formato JSON, com a informação a ser agora retornada através de ficheiros JSON. A implementação e criação da API RESTful foi realizada através do *framework* Flask, disponibilizando diversos *endpoints* que permitem a integração com o produto desenvolvido pela empresa.

Adição e Integração de Novas Fontes de Informação: Foram integradas no *script* diversas novas fontes de recolha de informação, bem como novos pontos para pesquisa. As novas fontes foram desenvolvidas com *scripts* dedicados, visando maximizar a quantidade e a abrangência da informação recolhida para o projeto.

Implementação de Funcionalidades Robustas para IPs e Domínios: As funcionalidades dos módulos MonitorizadorDomínio e MonitorizadorIPs foram significativamente melhoradas, focando na sua fiabilidade e na amplitude da informação recolhida. As principais adições e melhorias incluem:

- **Para Domínios:** Pesquisa de subdomínios expandida com Shodan e SecurityTrails, além da já existente crt.sh.
- **Novas Fontes de Informação Integradas:** Shodan, LeakLookup, APIs de *blacklists* adicionais (complementando as já procuradas), NVD para pesquisa de CVEs associadas a tecnologias detetadas, Netlas, Onyphe, URLScan e Mozilla HTTP Observatory
- **Novos Pontos de Pesquisa:** ExploitDB: Permite a pesquisa detalhada de *exploits* associados a CVEs; NVD: Para pesquisa de CVEs de tecnologias específicas; LeakLookup: Possibilita a pesquisa de *leaks* associados a endereços de e-mail, domínios e IPs; Módulos de *Threat Intelligence* do OpenCTI e *feeds* de segurança informática: Para recolha e análise aprofundada de informação de ameaças. Estas modificações e adições permitiram que a API se tornasse muito mais completa a nível de informação e consideravelmente mais útil para os utilizadores.

Documentação e Testes: Todo o código foi extensivamente documentado internamente, tornando a sua compreensão muito mais intuitiva. As funções foram cuidadosamente organizadas e divididas em ficheiros e pastas adequadas à sua função. Adicionalmente, foi produzida uma documentação completa sobre todas as funções do

código, e desenvolvido um ficheiro README abrangente, contendo informações essenciais sobre a instalação e a execução da solução.

Integrações e Funcionalidades: O script final possui funcionalidades de recolha de informação, nomeadamente para IPs e Domínios. As principais capacidades são:

- **Para IPs:** Procura de portas abertas (utilizando Masscan); Pesquisa de certificados SSL/TLS; Enumeração de protocolos; Verificação do estado das portas (e.g., open, filtered); Consulta de blacklists; Vulnerabilidades; Informações de Localização; Informações Organizacionais e de Domínios.
- **Para Domínios:** Procura de subdomínios (incluindo verificação de certificados); Verificação de blacklists de IPs associados aos domínios; Análise de versões SSL/TLS; Verificação de cabeçalhos de segurança; Deteção de typosquatting; Registros DNS; Vulnerabilidades de Tecnologia; Informações Detalhadas de Resposta HTTP/HTTPS; Tecnologias Identificadas.

Foram também desenvolvidas outras funcionalidades com *endpoints* específicos, como:

- /cves**:** Consulta de vulnerabilidades com base em tecnologias detetadas.
- /oxsl_feed**:** Acesso ao *feed* OxSI_f33d com filtragem por data.
- /leak_lookup**:** Verificação de exposição de *emails* / credenciais.
- Consulta dos IoCs (Indicadores de Compromisso) recolhidos no dia pelo OpenCTI.
- *Top 10* dos principais ataques a um país.
- Listagem de ataques por país num período definido.

Anexo B

Tabela 11-Testes

ID	Cenário	Imputs	Ação	Resultado Esperado	Resultado Obtido
T01	Consulta a. Domínios válidos	ulusofona.pt	POST/http://127.0.0.1:5000 /monitorizador/DOM	Retorno de um Json com a informação completa sobre o domínio e subdomínio	200 Foi obtido com sucesso o Json com a informação completa do ulusofona.pt bem como os seus subdomínios
T02	Consulta a Domínios inválidos	ulufona.tfdv	POST/http://127.0.0.1:5000 /monitorizador/DOM	Retorno de um Json vazio uma vez que o domínio é invalido	200 Foi Obtido um Json vazio, sem qualquer erro
T03	Consulta a Domínios validos e inválidos	lusfona.pt ulusofona.pt	POST/http://127.0.0.1:5000 /monitorizador/DOM	Retorno de um Json Contento. Apenas a informação valida	200 Foi obtido um Json contento apenas a informação do domínio valido
T04	Consulta a Domínios com alguma das APIs de fontes externas indisponíveis - Shodan	ulusofona.pt	POST/http://127.0.0.1:5000 /monitorizador/DOM	Retorno do um Json Contento a informação das restantes fontes externas	200 Foi obtido um Json contento a informação sobre o domínio
T05	Consulta a Domínios com alguma das APIs de fontes externas indisponíveis - LeakLookup	ulusofona.pt	POST/http://127.0.0.1:5000 /monitorizador/DOM	Retorno de um Json contento a informação sobre o domínio mesmo que a não seja possível a conexão ao leaklookup	200 Foi obtido um Json contento a informação sobre o domínio
T06	Consulta escolhendo mal o método neste caso GET	Ulusofona.tp	GET//http://127.0.0.1:5000 /monitorizador/DOM	Retorno de uma mensagem "Método não permitido" e Código 405	Foi obtida a mensagem esperada assim bem como o código 405

T07	Consulta a API enviando, mas o requisito /DOM/	ulusofona.pt	POST//http://127.0.0.1:5000 /monitorizador/DOT	Retorno de uma mensagem "Método não permitido "	Foi obtida a mensagem esperada bem como o código 400 predefinido
T08	Consulta de Ips válidos	213.58.148.218	POST///http://127.0.0.1:5000 /monitorizador/IP	Retorno de um Json com a informação completa sobre o respetivo IP	200 Foi obtido um Json com a informação completa sobre o IP
T09	Consulta de IP invalido	1.1.1.fd 34	POST///http://127.0.0.1:5000 /monitorizador/IP	Retorno de um Json vazio uma vez que os IPS são inválidos	200 Foi Obtido um Json vazio, sem qualquer erro
T10	Consulta de IPS válidos e inválidos	65.21.239.46- Válido fe80::10c3:4ecd:3114: baef%en6 1.1.1. fd 34	POST///http://127.0.0.1:5000 /monitorizador/IP	Retorno de um Json contento apenas a informação do IP valido uma vez que API apenas suporta ipv4	200 Foi obtido um Json contento apenas a informação do IP valido
T11	Consulta de CVEs sobre um determinado software existente	Font Awesome	POST///http://127.0.0.1:5000 /pesquisar-cve/	Retorno de um Json com todos os CVES sobre o software bem como a sua descrição	200 Foi obtido um Json com todos os CVES
T12	Consulta de CVEs sobre um determinado software inexistente	UBornto 20.04	POST///http://127.0.0.1:5000 /pesquisar-cve/	Retorno de um Json vazio	200 Foi obtido um Json vazio sem qualquer erro
T13	Consulta a informação sobre um CVE específico	CVE- 2021-44228	POST///http://127.0.0.1:5000 /CVE/	Retorno de um Json com toda a Informação sobre os CVES	200 Foi obtido um Json com toda a informação sobre o determinado CVE
T14	Consulta de Leaks de um determinado Domínio	ulusofona.pt	POST///http://127.0.0.1:5000 /LOOKUP/	Retorno de um Json com todos os dataleaks do domínio	200 Foi obtido um Json com todos os domínios para o domínio
T15	Consulta a. Domínio	apdp.pt	GET/http://127.0.0.1:5000	Retorno de um Json com a informação	200

	válidos		/monitorizadorweb/DOM/apdp.pt/	completa sobre o domínio e respetivos subdomínios	Foi obtido com sucesso o Json com a informação completa do domínio apdp.pt bem como os seus subdomínios
T16	Consulta a IP válido	185.118.114.199	GET/http://127.0.0.1:5000 /monitorizadorweb/IP/185.118.114.199	Retorno de um Json com a informação completa sobre o IP	200 Foi obtido com sucesso o Json com a informação completa do IP
T17	Consulta ao OxSl_f33d		GET /http://127.0.0.1:5000/OxSl_f33d/	Retorno de um Json com a informação com domínios portugueses associados a phings no dia de hoje	200 Foi obtido com sucesso o Json com a informação completa
T18	Consulta ao ctiTOP10	Portugal	GET /http://127.0.0.1:5000/ctiTOP10/Portugal/	Retorno de um Json com a informação com os 10 ataques ao pais	200 Foi obtido com sucesso o Json com a informação completa
T19	Consulta ao Ctiweb	hashes	GET /http://127.0.0.1:5000/ctiweb/hashes/	Retorno de um Json com a informação com os sobre os hashes	200 Foi obtido com sucesso o Json com a informação completa
T20	Consulta ao Ctiweb	DOM	GET /http://127.0.0.1:5000/ctiweb/DOM/	Retorno de um Json com a informação com os sobre os domínios s	200 Foi obtido com sucesso o Json com a informação completa
T21	Consulta ao Ctiweb	IP	GET /http://127.0.0.1:5000/ctiweb/IP/	Retorno de um Json com a informação com os sobre os ips	200 Foi obtido com sucesso o Json com a informação completa
T22	Consulta ao ctipais	Portugal	GET /http://127.0.0.1:5000/ctipais/Portugal/	Retorno de um Json com a informação com os ataques destinados ao pais	200 Foi obtido com sucesso o Json com a informação completa
T23	Consulta ao ctipais	Polland	GET /http://127.0.0.1:5000/ctipais/Polland/	Retorno de um Json com a informação com os ataques destinados ao pais	200 Foi obtido com sucesso o Json com a informação completa

T24	Consulta ao ctipais	Portugal 04-2025 05-2025	GET /http://127.0.0.1:5000/ctipais/Portugal/04-2025 /05-2025/	Retorno de um Json com a informação com os ataques destinados ao país durante o determinado período selecionado	200 Foi obtido com sucesso o Json com a informação completa
T25	Consulta ao ctipais	Portugal 04-2025	GET /http://127.0.0.1:5000/ctipais/Portugal/04-2025	Retorno de um Json com a informação com os ataques destinados ao pais apartir do mês selecionado	200 Foi obtido com sucesso o Json com a informação completa
T26	Consulta ao ctipaisfim	Portugal 05-2025	GET /http://127.0.0.1:5000/ctipaisfim/Portugal/05-2025/	Retorno de um Json com a informação com os ataques destinados ao pais ate do mês selecionado	200 Foi obtido com sucesso o Json com a informação completa
T27	Consulta ao ctipais	Portugal 07-2025	GET /http://127.0.0.1:5000/ctipais/Portugal/07-2025	Retorno de um Json com a informação vazio uma vez que ainda não estamos no mês 7	200 Foi obtido com sucesso o Json vazio sem erros.
T28	Consulta de Leaks de um determinado Domínio		POST///http://127.0.0.1:5000 /LOOKUP/	Retorno de um Json Vazio uma vez que o domínio não tem leaks	200 Foi obtido um Json vazio sem qualquer erro.

Abaixo são apresentados alguns dos testes destacados na tabela acima. No entanto, no caso do JSON completo sobre os domínios e subdomínios, foi necessário realizar alguns cortes, não sendo possível apresentar na íntegra o JSON devolvido, uma vez que o mesmo continha cerca de 39 mil linhas, o que tornaria impensável adicioná-lo na totalidade.

```
{
    "dominios": [
        {
            "domain": "ulusofona.pt",
            "ip": "193.137.75.244",
            "time": "2025-04-16 14:47:56",
            "data_leaks": null,
            "certificados_securitytrails": null,
            "subdominios": [
            {
                "nome": "dcc.ulusofona.pt",
                "ip": "193.137.75.175",
                "ip_shodan": [
                    "193.137.75.175"
                ],
                "start_date": "None",
                "valid_until": "None",
                "days_left": "Falha ao verificar certificado SSL",
                "org_name": "None",
                "time": "2025-04-16 14:51:50",
                "headers": [
                    {
                        "header": "x-frame-options",
                        "info": "is missing",
                        "status": "WARN",
                        "time": "2025-04-16 14:51:50"
                    },
                    {
                        "header": "strict-transport-security",
                        "info": "is missing",
                        "status": "WARN",
                        "time": "2025-04-16 14:51:50"
                    },
                    {
                        "header": "access-control-allow-origin",
                        "info": "is missing",
                        "status": "OK",
                        "time": "2025-04-16 14:51:50"
                    },
                    {
                        "header": "content-security-policy",
                        "info": "is missing",
                        "status": "WARN",
                        "time": "2025-04-16 14:51:50"
                    },
                    {
                        "header": "x-xss-protection",
                        "info": "is missing",
                        "status": "WARN",
                        "time": "2025-04-16 14:51:50"
                    },
                    {
                        "header": "x-content-type-options",
                        "info": "is missing",
                        "status": "WARN",
                        "time": "2025-04-16 14:51:50"
                    },
                    {
                        "header": "x-powered-by",
                        "info": "is missing",
                        "status": "OK",
                        "time": "2025-04-16 14:51:50"
                    },
                    {
                        "header": "server",
                        "info": "contains value 'Apache'",
                        "status": "WARN",
                        "time": "2025-04-16 14:51:50"
                    },
                    {
                        "header": "HTTPS supported",
                        "info": null,
                        "status": "OK",
                        "time": "2025-04-16 14:51:50"
                    },
                    {
                        "header": "HTTPS valid certificate",
                        "info": null,
```


Figura 38- Resultado do Teste T01:Validação Domínio Válido

```
                    {
                        "header": "HTTP -> HTTPS redirect",
                        "info": null,
                        "status": "OK",
                        "time": "2025-04-16 14:51:50"
                    }
                ],
                "headers_shodan": [],
                "ports_shodan": [],
                "technologies_shodan": [
                    "",
                    "Link: <https://dcc.ulusofona.pt/wp-json/wp/v2/pages/12>; rel=\"alternate\"; title=\"JSON\"; type=\"application/json\"\r",
                    "HTTP/1.1 200 OK\r",
                    "Date: Mon, 07 Apr 2025 07:20:25 GMT\r",
                    "Link: <https://dcc.ulusofona.pt/>; rel=shortlink\r",
                    "Link: <https://dcc.ulusofona.pt/wp-json/>; rel=\"https://api.w.org/\"\r",
                    "Content-Type: text/html; charset=UTF-8\r",
                    "Transfer-Encoding: chunked\r",
                    "\r",
                    "Server: Apache\r",
                    "Vary: Accept-Encoding\r"
                ],
                "technologies_cves_shodan": {
                    "Apache": []
                },
                "servers": [],
                "technologies_cves": {},
                "tls_versions_shodan": [
                    "versions",
                    "jarm",
                    "trust",
                    "acceptable_cas",
                    "handshake_states",
                    "ja3s",
                    "chain",
                    "ocsp",
                    "cert",
                    "tlsext",
                    "chain_sha256",
                    "mlpr",
                    "cipher"
                ],
                "services_shodan": [
                    "None: Apache httpd"
                ]
```


Figura 39- Resultado do Teste T01:Validação Domínio Válido

Figura 40- Resultado do Teste T01:Validação Domínio Válido

Teste 2

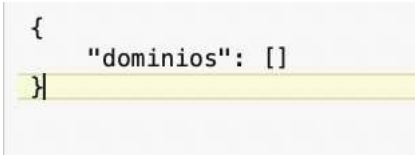


Figura 41- Resultado do Teste TO2: Validação de Domínio Inválido

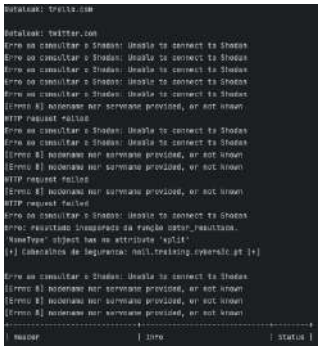


Figura 42- Resultado do Teste T04: Erro no Shodan

```
            "status": "WARN",
            "time": "2025-04-16 15:14:26"
        },
        {
            "header": "access-control-allow-origin",
            "info": "is missing",
            "status": "OK",
            "time": "2025-04-16 15:14:26"
        },
        {
            "header": "content-security-policy",
            "info": "is missing",
            "status": "WARN",
            "time": "2025-04-16 15:14:26"
        },
        {
            "header": "x-xss-protection",
            "info": "is missing",
            "status": "WARN",
            "time": "2025-04-16 15:14:26"
        },
        {
            "header": "x-content-type-options",
            "info": "is missing",
            "status": "WARN",
            "time": "2025-04-16 15:14:26"
        },
        {
            "header": "x-powered-by",
            "info": "is missing",
            "status": "OK",
            "time": "2025-04-16 15:14:26"
        },
        {
            "header": "server",
            "info": "contains value 'cloudflare'",
            "status": "WARN",
            "time": "2025-04-16 15:14:26"
        },
        {
            "header": "HTTPS supported",
            "info": null,
            "status": "OK",
            "time": "2025-04-16 15:14:26"
        },
        {
            "header": "HTTPS valid certificate",
            "info": null,
            "status": "FAIL",
            "time": "2025-04-16 15:14:26"
        },
        {
            "header": "HTTP -> HTTPS redirect",
            "info": null,
            "status": "OK",
            "time": "2025-04-16 15:14:26"
        }
    ],
    "headers_shodan": [],
    "ports_shodan": [],
    "technologies_shodan": [],
    "technologies_cves_shodan": null,
    "servers": [],
    "technologies_cves": {},
    "tls_versions_shodan": [],
    "services_shodan": []
```


Figura 43- Resultado do Teste T05: Erro numa API Externa


```
{
    "dominios": [
        {
            "domain": "cybers3c.pt",
            "ip": "172.67.74.154",
            "time": "2025-04-16 15:10:51",
            "data_leaks": [
                {
                    "Dataleak": "promo.com"
                },
                {
                    "Dataleak": "canva.com"
                },
                {
                    "Dataleak": "trello.com"
                },
                {
                    "Dataleak": "twitter.com"
                }
            ],
            "certificados_securitytrails": null,
            "subdominios": [
                {
                    "nome": "anciber.cybers3c.pt",
                    "ip": "104.26.12.174",
                    "ip_shodan": [],
                    "start_date": "None",
                    "valid_until": "None",
                    "days_left": "Falha ao verificar certificado SSL",
                    "org_name": "None",
                    "time": "2025-04-16 15:14:23",
                    "headers": [
                        {
```


Figura 44- Resultado do Teste T05: Erro numa API Externa (Shodan)

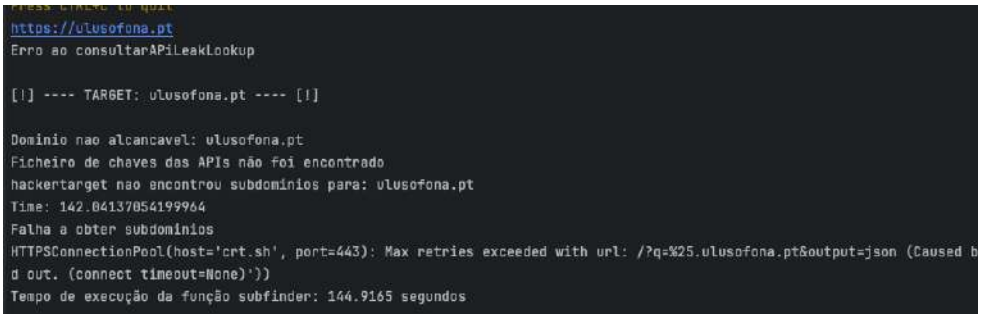


Figura 45- Resultado do Teste T05: Erro na Requisição do LeakLookup

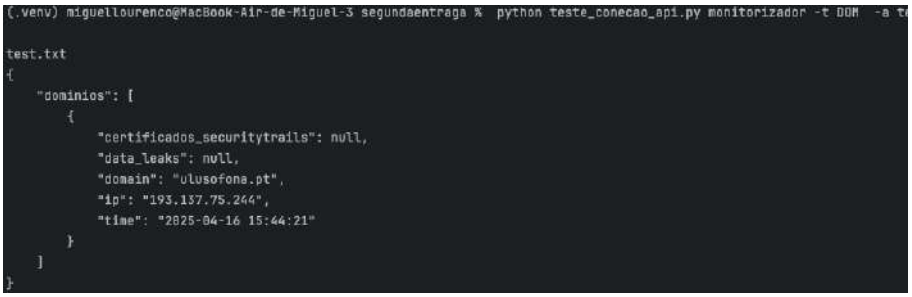


Figura 46- Resultado do Teste T05: Erro numa API Externa

Figura 47- Resultado do Teste T08: Validação de IP Válido

Figura 48- Resultado do Teste T08: Validação de IP Válido

Figura 49- Resultado do Teste T08: Validação de IP Válido

Figura 50- Resultado do Teste T10: Validação de IP Inválido

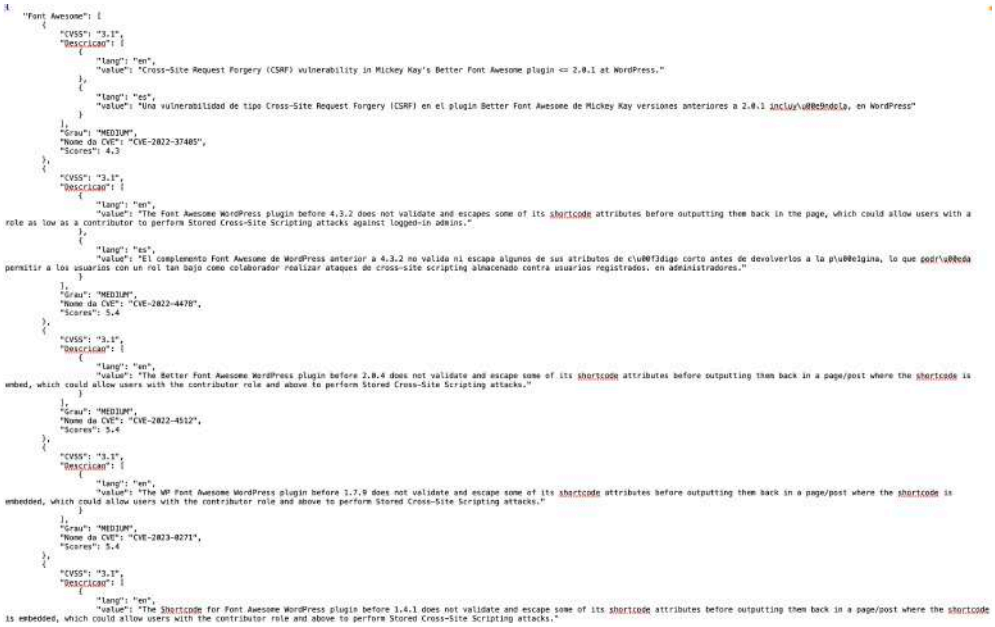


Figura 51- Resultado do Teste T11: Pesquisa de CVEs por Software

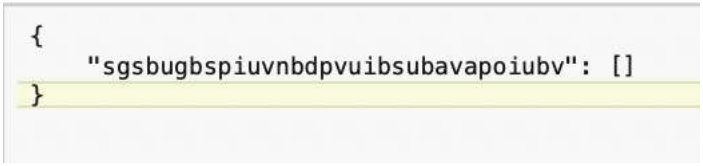


Figura 52- Resultado do Teste T12: Pesquisa Inválida de CVEs por Software

Figura 53- Resultado do Teste T13: Pesquisa por CVE

Figura 54- Resultado do Teste T13: Pesquisa por *Leak*

Figura 55- Resultado do Teste T28: Pesquisa por CVE Inválido

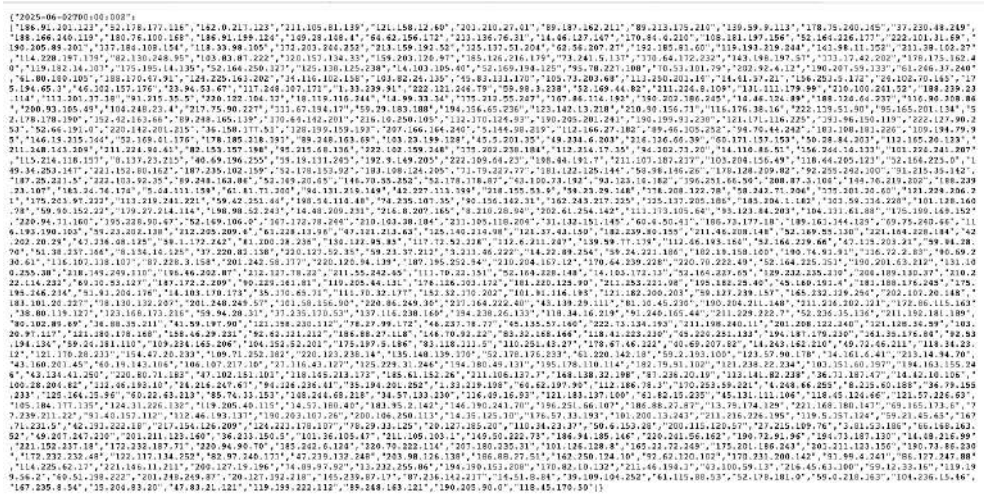


Figura 56- Resultado do Teste T21: Pesquisa de IPs no CTI

Figura 57- Resultado do Teste T19: Pesquisa de *Hashes* no CTI

Figura 58- Resultado do Teste T20: Pesquisa de Domínios no CTI

Figura 59- Resultado do Teste T22: Pesquisa dos 10 Principais Ataques em Portugal (CTI)

{"Portugal":[{"TLP":"TLP:CLEAR","atacante":"Shenzhen Haimaiyunxiang Media Co., Ltd.","confian\u00e7a":100,"criado":"2025-04-
01T08:50:47.800Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Intrusion-Set"},
{"TLP":"TLP:CLEAR","atacante":"SYS01","confian\u00e7a":100,"criado":"2025-04-
01T10:54:36.991Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"Badoken","confian\u00e7a":100,"criado":"2025-04-
01T02:42:06.290Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"FlyGram","confian\u00e7a":100,"criado":"2025-04-
01T05:48:43.607Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"ZoomBinder","confian\u00e7a":100,"criado":"2025-04-
01T02:42:07.552Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"BadBazaar","confian\u00e7a":100,"criado":"2025-04-
01T05:49:03.727Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"Coper","confian\u00e7a":100,"criado":"2025-04-
01T09:31:46.534Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Intrusion-Set"},
{"TLP":"TLP:CLEAR","atacante":"GymDrop","confian\u00e7a":100,"criado":"2025-04-
01T02:42:09.947Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},{"TLP":"TLP:CLEAR","atacante":"Cerberus (ex-
Amnesia)","confian\u00e7a":100,"criado":"2025-04-01T11:27:00.874Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Intrusion-Set"},
{"TLP":"TLP:CLEAR","atacante":"BugDrop","confian\u00e7a":100,"criado":"2025-04-
01T02:42:12.584Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},{"TLP":"TLP:CLEAR","atacante":"XOCTOPUS -
80669","confian\u00e7a":100,"criado":"2025-04-01T09:31:48.647Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"TyToxic","confian\u00e7a":100,"criado":"2025-04-
01T16:51:51.612Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},{"TLP":"TLP:CLEAR","atacante":"Hadoken Security
Group","confian\u00e7a":100,"criado":"2025-04-01T02:42:05.158Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Intrusion-Set"},
{"TLP":"TLP:CLEAR","atacante":"ScarletStealer","confian\u00e7a":100,"criado":"2025-04-
01T10:54:35.639Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},{"TLP":"TLP:CLEAR","atacante":"OS
X","confian\u00e7a":100,"criado":"2025-04-01T05:48:52.032Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"Exobot","confian\u00e7a":100,"criado":"2025-04-
01T09:31:49.678Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"ToxicPanda","confian\u00e7a":100,"criado":"2025-04-
01T16:51:50.797Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"BugSleep","confian\u00e7a":100,"criado":"2025-04-
01T13:52:25.259Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"Lampion","confian\u00e7a":100,"criado":"2025-05-
06T15:37:27.633Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"Lampion","confian\u00e7a":100,"criado":"2025-05-
06T15:37:26.990Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Intrusion-Set"},
{"TLP":"TLP:CLEAR","atacante":"Acrid","confian\u00e7a":100,"criado":"2025-04-
01T10:54:33.350Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"antidot","confian\u00e7a":100,"criado":"2025-04-
01T17:56:35.610Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},{"TLP":"TLP:CLEAR","atacante":"Metamorfo -
80455","confian\u00e7a":100,"criado":"2025-04-01T11:30:24.705Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"RedLine","confian\u00e7a":100,"criado":"2025-04-
01T11:29:33.962Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"LabHost","confian\u00e7a":100,"criado":"2025-04-
01T09:57:34.852Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Intrusion-Set"},{"TLP":"TLP:CLEAR","atacante":"Rhadamanthys
Stealer","confian\u00e7a":100,"criado":"2025-04-01T11:32:50.566Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"Dracula Stealer (Samurai)","confian\u00e7a":100,"criado":"2025-04-
01T11:32:04.804Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"GREF","confian\u00e7a":100,"criado":"2025-04-
01T05:48:39.767Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Intrusion-Set"},
{"TLP":"TLP:CLEAR","atacante":"ToxicPanda","confian\u00e7a":100,"criado":"2025-04-
01T16:51:48.040Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Intrusion-Set"},
{"TLP":"TLP:CLEAR","atacante":"DUCKTAIL","confian\u00e7a":100,"criado":"2025-04-
01T06:13:40.004Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},

Figura 60- Resultado do Teste T23: Pesquisa CTI por País – Portugal

{"Portugal":[{"TLP":"TLP:CLEAR","atacante":"Lampion","confian\u00e7a":100,"criado":"2025-05-
06T15:37:27.633Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Malware"},
{"TLP":"TLP:CLEAR","atacante":"Lampion","confian\u00e7a":100,"criado":"2025-05-
06T15:37:26.990Z","criador":"AlienVault","rela\u00e7\u00e3o":"targets","tipo_alvo":"Country","tipo_ataque":"Intrusion-Set"}]}

Figura 61- Resultado do Teste T24: Pesquisa CTI por País – Portugal (04/2025 a 05/2025)

Figura 62- Resultado do Teste T24: Pesquisa CTI por País – Polónia (04/2025 a 05/2025)

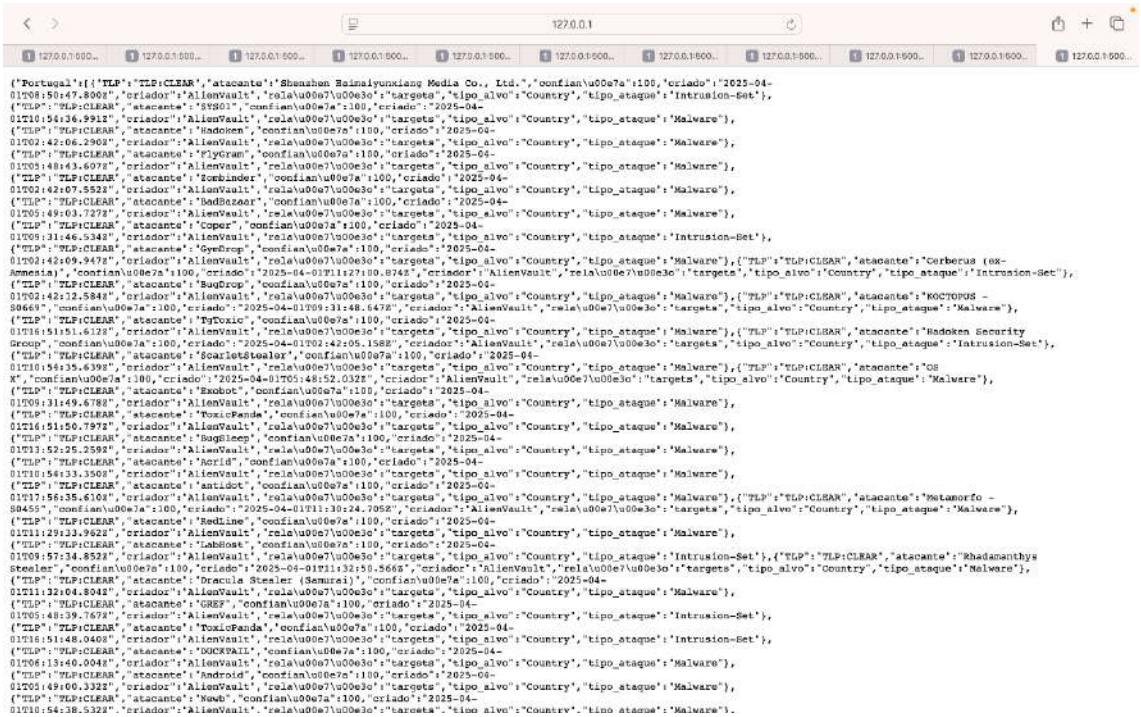


Figura 63- Resultado do Teste T25: Pesquisa CTI por País – Portugal (04/2025)

Figura 64- Resultado do Teste T27: Pesquisa CTI por País – Portugal (07/2025)

Anexo C

Documentação da API Monitorizador

O objetivo deste anexo é a uma exposição da documentação desenvolvida para o monitorizador ,documentação esta que tinha como objetivo clara e completa sobre o desenvolvimento da API realizado ao longo dos últimos meses. Esta documentação visa descrever detalhadamente os principais ficheiros envolvidos no projeto, bem como as funções de maior relevância para o seu funcionamento. Serão ainda apresentados todos os endpoints disponíveis, com a devida explicação dos seus parâmetros e respostas esperadas. Adicionalmente, este anexo incluirá um guia prático que descreve, passo a passo, o processo de instalação da API, contemplando os requisitos necessários, configurações e dependências. É importante notar que este guia foi ligeiramente adaptado em relação ao original para se adequar ao formato e propósito deste anexo.

Guia de Instalação do Monitorizador

Pré-Requisitos

Para executar o projeto localmente, é necessário garantir que o ambiente cumpre os seguintes requisitos, de modo a permitir a replicação completa deste guia.
Sistemas Operativos:

1. *Linux* (Ubuntu 22.04 ou superior 22.04 ou 24.04 LTS)
2. *macOS*

Todos os sistemas operativos recomendados foram devidamente testados embora seja tecnicamente possível utilizar outra distribuição Linux, não se pode garantir o funcionamento integral do sistema conforme o esperado.

É necessário que a máquina cumpra os seguintes requisitos:

1. *Python* 3.10
2. *Git*
3. Uma IDE à escolha (recomenda-se o *VSCode* ou o *PyCharm*)
4. *Postman* (opcional)
5. *ExploitDB*
6. *Masscan*

Instalação dos principais requisitos:

Python

Para instalar o *Python*, é necessário aceder ao terminal da máquina e executar os seguintes comandos:

- sudo apt update
- sudo apt install python3

Git

Para instalar o *Git*, é necessário aceder ao terminal da máquina e executar o seguinte comando:

- sudo apt install git

Exploibd

Para instalar o *ExploitDB*, recomenda-se consultar a documentação oficial, de forma a garantir uma instalação completa e atualizada. A documentação está disponível em:

- https://www.exploit-db.com/searchsploit

Masscan

Para instalar o *Masscan*, é necessário aceder ao terminal da máquina e executar os seguintes comandos:

- sudo apt update
- sudo apt install masscan

Para além das ferramentas mencionadas acima, é também necessário requerer e configurar as chaves de API, de forma a garantir o funcionamento adequado dos serviços utilizados. Esta configuração deve ser efetuada no ficheiro: api_keys.yaml

Keys:

- *Shodan*
- *HackerTarget*
- *SecurityTrails*
- *urlSan.Io*
- *OxSi_f33d*
- *Lookup*
- *Blacklist Checker*
- *Opencti*

```
shodan:
  key: "A_TUA_API_KEY_AQUI"
```

Figura 65-Exemplo de como configurar as keys

É necessário configurar o ficheiro configs.yaml com a interface de rede correta. Para identificar a interface em uso, pode utilizar o seguinte comando no terminal:

- ip a

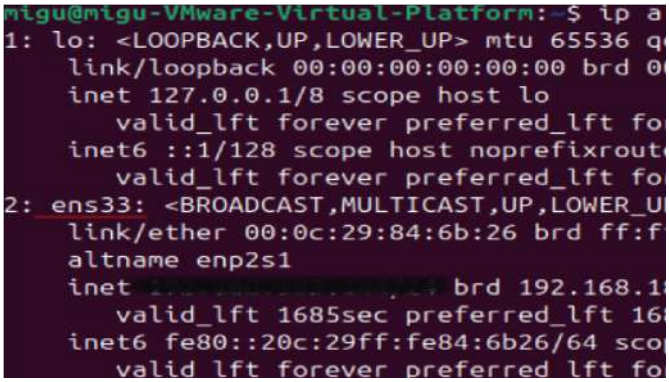


Figura 66-Exemplo de requisição da interface de internet ens33

- masscan_interface: "ens33" (exemplo)

É ainda necessário configurar no mesmo ficheiro o url da máquina que contem o opencti de modo

```
opencti_url: 'http://139.59.163.170:8080/'
```

Figura 67-Exemplo de url Opencti

Após garantir que todas as dependências mencionadas foram corretamente instaladas, podemos proceder com a instalação do projeto.

9 Inicialização da API

Clonar repositório

Para clonar o repositório para a máquina local, deve-se executar o seguinte comando no terminal:

• git clone https://github.com/duke-the-1998/TFC-Lusofona-API

Criar e Ativar um ambiente Virtual

O *Python* utiliza ambientes virtuais, nos quais é possível instalar as bibliotecas necessárias para a execução do projeto. Para criar e ativar o ambiente virtual, devem ser utilizados os seguintes comandos:

- sudo apt install python3.12-venv
- python -m venv venv
- source venv/bin/activate

Instalar dependências

O repositório já inclui as dependências necessárias para o funcionamento do projeto. Para instalá-las, deve-se executar o seguinte comando:

• pip install -r requirements.txt

Como correr a API

Após concluir todos os procedimentos acima descritos, podemos proceder à execução do projeto. O mesmo deve ser executado a partir do terminal.
Como no Linux não é possível utilizar o sudo fora do ambiente virtual, e como o sudo é necessário para a execução de certos componentes no diretório principal do projeto, deve-se executar o seguinte comando:

- which python

Deste modo, será possível obter o caminho até ao *Python* dentro do ambiente virtual. Após isso, podemos executar o seguinte comando:

- sudo /home/user/PycharmProjects/TFC-Lusofona-API/.venv/bin/python monitorizador.py

Este comando irá iniciar o servidor *Flask* local na porta 5000, acessível em http://127.0.0.1:5000/. No entanto, a porta pode ser alterada no ficheiro monitorizador.py.

Figura 68-Porta padrão flask

Requests à API

Existem três formas principais de interagir com os endpoints da API:

Postman

Pode utilizar o Postman para testar os endpoints expostos pela API.

Exemplo de URL de request:

- http://127.0.0.1:5000/monitorizador/DOM

Neste exemplo:

- 5000 que é a porta definida no servidor *Flask*
- monitorizador/DOM representa o *endpoint* que queremos fazer um *request*

O método HTTP a ser utilizado deve ser o POST. No *Postman*, essa opção pode ser selecionada no menu suspenso à esquerda do campo onde é inserido o URL da requisição.
Após definir o método HTTP e o URL, é necessário inserir no corpo da requisição (*body*) uma lista de IPs ou domínios que se deseja consultar, no formato de lista, como exemplificado abaixo:

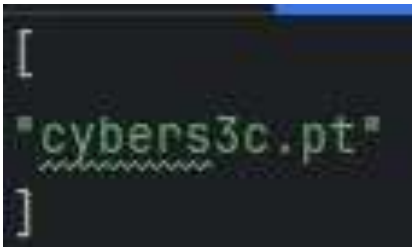


Figura 69-Exemplo de requisição

script teste_conecao_api.py

Foi também desenvolvido um script para facilitar a realização de testes automáticos através da linha de comandos.

Exemplo de execução:

- python teste_conecao_api.py monitorizador -t DOM -a test.txt

Neste exemplo:

- DOM é o tipo de scan que se quer realizar

- test.txt é onde vai estar a lista dos alvos que queremos analisar

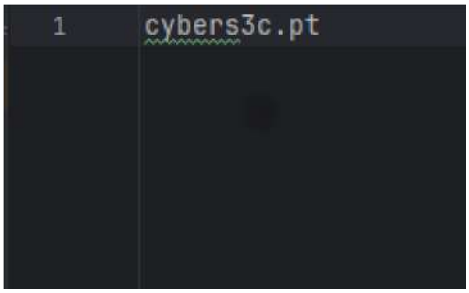


Figura 70-Exemplo de requisição

Request browser

É possível requisitar alguns endpoints diretamente através do *browser*; no entanto, essa abordagem está limitada aos endpoints que utilizam o método GET, como é o caso de monitorizadorweb e outros semelhantes.

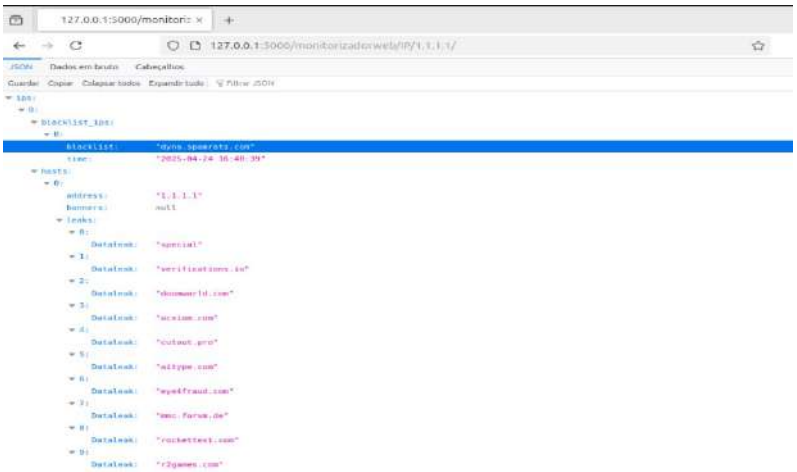


Figura 71-Exemplo de Requisição

Endpoints da API

Monitorização de Ips

Este *endpoint* retorna informações detalhadas sobre um determinado IP fornecido, bem como sobre os IPs associados. Foi desenvolvido com o método POST, permitindo receber um ficheiro JSON contendo uma lista de IPs.

Parâmetros:

- **Ip's (obrigatório)**: Os endereções Ip´s a ser analisados. Exemplo: ["8.8.8.8"]

Exemplo de Requisição:

- http://127.0.0.1:5000/monitorizador/IP

Figura 72-Exemplo de Requisição Post

Monitorizador de IPS via Web

Este *endpoint* retorna informações detalhadas sobre um determinado IP fornecido e os IPs a ele associados. Ao contrário do anterior, foi desenvolvido com o método GET, sendo, por isso, possível analisar apenas um endereço IP de cada vez.

Parâmetros

- **IP (obrigatório):** IP a ser analisado, como no exemplo: **/1.1.1.1/**

Exemplo de Requisição:

- http://127.0.0.1:5000/monitorizadorweb/IP/1.1.1.1/

Monitorização de Domínios

Este *endpoint* retorna informações detalhadas sobre um determinado domínio fornecido, bem como sobre todos os seus subdomínios. Foi desenvolvido com o método POST, permitindo o envio de um ficheiro JSON contendo uma lista de domínios ou subdomínios.

Parâmetros :

- **domínios(obrigatório):**Domínios a analisar. Exemplo: ["cybers3c.pt", "teste.pt"]

Exemplo de Requisição:

- http://127.0.0.1:5000/monitorizador/DOM

Monitorizador de Domínios via Web

Este *endpoint* retorna informações detalhadas sobre um determinado domínio fornecido, bem como os seus subdomínios associados. Ao contrário do anterior, foi desenvolvido com o método GET, sendo por isso possível analisar apenas um domínio de cada vez.

Parâmetros:

- **domínio (obrigatório):** Domínio a ser analisado, como no exemplo: **/exemple.com/**

Exemplo de Requisição:

- http://127.0.0.1:5000/monitorizadorweb/DOM/exemple.com/

LOOKUP

Este *endpoint* retorna os *data leaks* associados a um determinado IP, domínio, nome de utilizador (*username*) ou endereço de e-mail. Foi desenvolvido com o método POST, permitindo o envio de um ficheiro JSON com uma lista de IPs, domínios, *usernames* e e-mails para os quais se pretende obter essa informação.

Parâmetros:

- **IP, Domínio, Username ou mail (obrigatório)** a ser analisado

Exemplo de requisição

- http://127.0.0.1:5000/ LOOKUP/

CVE

Este *endpoint* retorna os scripts e *exploits* associados a um determinado CVE pesquisado. Foi desenvolvido com o método POST, permitindo o envio de um ficheiro JSON com uma lista de CVEs para consulta.

Parâmetros:

- **Cves (Obrigatório)**: CVE a ser analisado como no exemplo: ["cve-202154"]

Exemplo de requisição:

- http://127.0.0.1:5000/ CVE/

pesquisar-cve

Este *endpoint* retorna todos os CVEs associados a uma determinada palavra-chave ou tecnologia. Foi desenvolvido com o método POST, permitindo o envio de um ficheiro JSON contendo uma lista de palavras-chave para procurar os CVEs correspondentes.

Parâmetros:

- **Palavras (Obrigatório)**: Palavrava a ser analisada como no exemplo: ["Ubuntu "]

Exemplo de requisição:

- http://127.0.0.1:5000/ pesquisa-cve/

OxSl_feed
Este *endpoint* retorna à consulta do OxSl_f33d da segurança informática em formato JSON. Foi desenvolvido com o método POST.

Exemplo de Requisição:

- http://127.0.0.1:5000/OxSl_feed/

ctipais
Este *endpoint* retorna à consulta sobre o *feed* do OpenCTI relacionado a ataques direcionados a um país durante um período definido.

Parâmetros:

- **/ctipais/<typeScan>/<typeScan2>/<typeScan3>/**

O primeiro parâmetro (<typeScan>) é obrigatório e corresponde ao nome do país.

Exemplo:

- "United Kingdom of Great Britain and Northern Ireland"

O segundo (<typeScan2>) e terceiro parâmetro (<typeScan3>) correspondem à data do período de procura. Ambos devem seguir o formato MM-AAAA.

Exemplo:

- "03-2025"

Exemplos de Requisição:

- http://127.0.0.1:5000/ctipais/Portugal/03-2025/04-2025/
- http://127.0.0.1:5000/ctipais/Portugal/None/03-2025/

- http://127.0.0.1:5000/ctipais/Portugal/03-2025/

ctiTOP10
Este *endpoint* retorna à consulta sobre o *feed* do *OpenCTI* relacionado aos 10 principais ataques direcionados a um determinado país. Foi desenvolvido com o método GET.

Parâmetros:

- **ctiTOP10/<typeScan>/ (Obrigatório)** como no exemplo – Portugal

Exemplos de Requisição:

- http://127.0.0.1:5000/ctiTOP10/Portugal/

ctiweb
Este *endpoint* retorna a consulta sobre os *feeds* do *OpenCTI* relacionados aos *hashes*, domínios e IPs do último mês. Foi desenvolvido com o método GET.

Parâmetros:

Este endpoint oferece **três subopções** para retornar o feed de **hashes**, **IPs** ou **domínios**:

- O parâmetro **<typeScan>** é **obrigatório** e define o tipo de feed a ser retornado. Os valores possíveis são:
  hashes
  DOM
  IPS

Exemplos de Requisição:

- http://127.0.0.1:5000/ctiweb/DOM/
- http://127.0.0.1:5000/ctiweb/IPS/
- http://127.0.0.1:5000/ctiweb/hashes/

Descrição do Código das Funções Principais e Ficheiros

Diretório Core.py

Dom_checker.py

Este *script* é responsável por realizar o processamento completo das informações de um domínio e dos seus subdomínios para resposta ao *endpoint* /monitorizador/DOM. Este *script* equipare-se ao ips.py (que trata os dados associados ao *endpoint* de endereços IP). Ele processa operações como deteção de subdomínios, análise de SSL, verificação de *typosquatting* e listas negras. A seguir estão presentes as funções e a suas respetivas descrições:

is_valid_domain

Esta função verifica se um domínio é válido através de uma expressão regular r"^((?!-)[A-Za-z0-9]{1,63}(?<!-)\.)+[A-Za-z]{2,6}". Recebe como parâmetro um nome de domínio e retorna verdadeiro se o domínio for válido, ou falso caso contrário. Esta função é bastante importante para a validação da lista de domínios fornecida pelo utilizador, garantindo que os domínios enviados para as próximas etapas do script estejam corretamente formatados.

clear_url

Esta função é responsável por receber uma *string* que contém um domínio ou subdomínio e limpar o URL para obter apenas o nome do domínio. Remove prefixos como www. e extrai apenas a parte principal do domínio. Esta função é utilizada para limpar tanto os domínios como os subdomínios, garantindo uma camada adicional de validação e assegurando que todas as entradas estejam padronizadas antes de serem processadas pelas próximas etapas do script, resultando numa resposta mais organizada e limpa.

simplify_list

Esta função tal como o nome indica, simplifica uma lista de listas, removendo os duplicados. É utilizada no processamento de subdomínios, dado um determinado domínio. Como o processamento de subdomínios é realizado através de várias fontes públicas e APIs, é comum existirem duplicados que precisam de ser eliminados.

get_crtsh_subdomains

Esta função obtém os subdomínios de um determinado domínio através da API do serviço crt.sh. O processamento da consulta é realizado no ficheiro crtsh.py, sendo que esta função invoca o método crtshAPI().search() para recolher os subdomínios associados ao domínio alvo. Os dados são retornados sob a forma de uma lista limpa e padronizada.

get_all_subdomains

Esta função é responsável por obter todos os subdomínios a partir de um determinado domínio. Tal como mencionado anteriormente, estes subdomínios são recolhidos a

partir de várias fontes e APIs, como *crt.sh, Knockpy, Dnsdumpster, Shodan e SecurityTrails*. Como os subdomínios são obtidos de diversas fontes, a lista necessita de ser validada e retirados os duplicados . A função conta ainda com um mecanismo de *threads* para minimizar o tempo de resposta.

check_reason
Esta função traduz as mensagens de erro para mensagens mais compreensíveis facilitando a identificar a razão de falha no código, assegurando uma resposta mais limpa e organizada.

process_subdomain
Esta função é uma das partes centrais do *script*, sendo responsável por analisar um determinado subdomínio, recolhendo diversas informações como endereço IP, certificados SSL, cabeçalhos de segurança, tecnologias utilizadas e CVEs associados. Estas informações são obtidas a partir de várias fontes e APIs (como *Shodan, Netlas*, entre outras), integradas noutras funções e scripts. A função retorna um dicionário com todos os dados relevantes sobre esse subdomínio. Esta função é chamada para cada um dos subdomínios encontrados.

processar_subdominio
Esta função obtém informações de diversas fontes externas (*ONYPHE, Netlas e Shodan*) para um subdomínio específico. Após recolher esses dados, invoca a função process_subdomain, passando-lhe todos os parâmetros necessários.

api_keys
Esta função obtém as chaves das APIs a partir do ficheiro YAML, para que possam ser utilizadas nas várias funções do *script.*

subdomains_finder_dnsdumpster

Esta função obtém os subdomínios de um determinado domínio através da API do *HackerTarget (DNSDumpster*). A função envia uma requisição à API, utilizando uma chave da API obtida pela função *hackertarget_key*(), e processa a resposta que contém pares de subdomínios e IPs. A lista que esta função retorna contém apenas os nomes dos subdomínios.

ssl_version_suported
Esta função verifica quais as versões SSL/TLS que estão a ser utilizadas num determinado domínio. Para isso, estabelece uma conexão com o domínio através da biblioteca padrão ssl do *Python*. Após a conexão, chama a função *check_ssl_version*() para identificar as versões suportadas e em uso.

check_ssl_version

Esta função verifica quais as versões SSL/TLS que estão a ser usadas para o hostname. Utiliza a biblioteca padrão ssl do Python e recebe como parâmetro a conexão SSL realizada pela função ssl_version_supported. Esta análise permite avaliar se o serviço está a utilizar versões seguras e atualizadas do protocolo.

db_insert_domain
Esta função insere o domínio na estrutura de dados principal jsonDominio e as suas informações detalhadas, como IP, data *leaks* e certificados *securitytrails*. Estas informações são recolhidas a partir de várias fontes externas e APIs, através de chamadas a outras funções.

blacklisted
Esta função verifica se o IP de um domínio está listado em uma ou mais *blacklists* DNS, que são usadas para identificar fontes de spam ou que não sejam confiáveis. A função consulta uma lista de mais de 60 serviços de *blacklist* e utiliza *threads* para fazer essas consultas de forma eficiente.

Insert_headers
Esta função é responsável por verificar e inserir os cabeçalhos de segurança HTTP para um subdomínio. Utiliza a classe *SecurityHeaders* para realizar uma análise dos cabeçalhos de segurança presentes na resposta HTTP do subdomínio. Para cada cabeçalho de segurança, é verificado se ele está definido e qual é o seu conteúdo. Cada resultado recebe um status de acordo com a presença e validade do cabeçalho, sendo classificado como OK (quando está corretamente definido) ou WARN (quando está ausente ou mal configurado). Todos os resultados são organizados numa tabela visual exibida no terminal e armazenados numa lista de dicionários.

typo_squatting_api
Esta função tem como objetivo identificar possíveis domínios de *typo-squatting* relacionados a um domínio. Utiliza a API do serviço DNS *Twister*, que gera variações do domínio original (com erros de digitação comuns) e verifica se esses domínios estão registados e resolvem para algum endereço IP.

ip_models.py
Neste *script* estão definidas todas as classes referentes aos IPs, como ModelHost, ModelPort e ModelInfo, que serão utilizadas através de funções nos ficheiros ips.py e utils.py.

Ips.py
Este *script* é responsável por realizar o processamento completo das informações dos ips para resposta ao endpoint /monitorizador/IP. Este script equipara-se ao dom_checker.py (que trata os dados associados ao *endpoint* dos domínios e seus subdomínios). Ele processa operações como masscan, nmap, reverse IP, listas negras,

leaks, banners, tecnologias e análise SSL. A seguir estão presentes as funções e as suas respetivas descrições.

validate_ip_address
Esta função verifica se um determinado IP é válido através de uma biblioteca do *Python* chamada *ipaddress* e do *isinstance()*, uma função que verifica se um objeto é instância de uma classe ou não. Assim, através destas duas ferramentas, a função recebe como parâmetro um IP e retorna verdadeiro se for válido, falso caso contrário. Esta função é bastante importante para a validação da lista de IPs que o utilizador envia, garantindo que os IPs que estão a ser enviados para as próximas etapas do script estejam bem formatados.

validate_network
Esta função é responsável por verificar se uma determinada rede IP é válida através da biblioteca do *Python ipaddress* e a função integrada desta biblioteca *ip_network*. Através desta função, se o valor fornecido não for uma rede válida, será lançada uma exceção *ValueError* e o *try-except* captura esse erro, retornando false nesse caso. Esta função é bastante importante para a validação da lista de redes IPs, garantindo que as redes IPs que estão a ser enviadas para as próximas etapas do script estejam bem formatadas.

is_private
Esta função verifica se um determinado IP é privado através da biblioteca do *Python ipaddress* e a função integrada desta biblioteca *ip_address().is_private*. Através desta função, se o valor fornecido for um IP privado, retorna verdadeiro; caso contrário, retorna falso. Esta função é bastante útil na função *reverse_ip_lookup* para verificar se o IP dado não é privado, permitindo assim executar o comando *nslookup.*

ip_range_cleaner
Esta função é responsável por estender uma gama de endereços IP fornecida em formato de rede, utilizando a biblioteca *ipaddress* do *Python* e a função integrada desta biblioteca *ip_network().hosts()* para gerar todos os endereços IP válidos. Todos os endereços gerados são escritos no ficheiro cleanIPs.txt, permitindo acumular os resultados para utilização noutras funções. Esta função é especialmente útil para preparar listas de IPs que serão posteriormente analisadas ou submetidas a varreduras.

Class Importer
Esta classe é responsável por definir a estrutura geral para importação e processamento de informações relacionadas a *hosts* e IPs. Isto inclui informações como: endereço IP, portas abertas (protocolo, estado, descrição e SSL associados), *leaks, banners*, tecnologias associadas, sistema operativo e organização associada.

Class NmapXMLImporter

Esta classe é uma subclasse de Importer que é responsável por ler e interpretar ficheiros de output do Nmap no formato XML, extraindo dados dos hosts e das suas respetivas portas.

Configsa
Esta função tem como objetivo obter a configuração da interface através do ficheiro configs.yaml. Ela abre o ficheiro localizado em /core/configs.yaml e obtém uma string com a interface do *masscan.* Esta função é útil para executar o comando *masscan,* pois define qual é a interface de rede que será utilizada durante o processo de varrimento. Caso o ficheiro de configuração não seja encontrado, a função retorna None.

ip_scan
Esta função é responsável por executar os comandos *masscan* e *nmap* através da biblioteca do *Python subprocess,* que possibilita a execução de comandos diretamente no terminal. A função utiliza o *Masscan* para realizar uma varredura rápida de todas as portas no IP fornecido. Em seguida, as portas detectadas são extraídas e utilizadas para realizar um scan mais detalhado com o *Nmap*, cujos resultados são guardados num ficheiro XML.

blacklistedIP
Esta função verifica se o IP está listado em uma ou mais *blacklists* DNS, que são utilizadas para identificar fontes de spam ou IPs não confiáveis. Ela consulta uma lista de mais de 60 serviços de *blacklist* e utiliza *threads* para realizar essas consultas de forma eficiente.

reverse_ip_lookup
Esta função tem como objetivo realizar uma pesquisa de DNS reverso para um determinado endereço

IP público, ou seja, tentar obter o nome de domínio associado a esse IP. A função verifica primeiro se o IP fornecido não pertence a uma rede privada. Caso seja um IP público, executa o comando *nslookup*.
Esta operação é útil para identificar o *hostname* de um IP.

utils.py

Este script atua como intermediário para executar os *scripts* responsáveis pelo processamento de informações para os *endpoints* dos domínios e IPs (*dom_checker.py e ips.py*).

run_ips
Esta função é responsável por executar as funções relativas ao processamento das informações para o endpoint dos IPs. Antes de executar essas funções, o IP fornecido é validado, garantindo que o código não apresenta erros e que seja mais organizado. As funções chamadas pelo *run_ips* fazem parte do script *ips.py*, que executa operações como varreduras ao *ip, lookup* reverso e verificação em *blacklists.*

run_domains
Esta função tem como objetivo executar as funções relativas ao processamento das informações para o *endpoint* dos domínios, equiparando-se à função *run_ips*, mas só que opera para os domínios. Antes de executar, o domínio fornecido é validado, garantindo que o código não apresenta erros e que seja mais organizado. As funções chamadas pelo *run_domains* fazem parte do script *dom_checker.py*, que executa operações como verificação de certificados, subdomínios, tecnologias, cves associados e verificação em *blacklists*.

is_subdomain
Esta função verifica se um subdomínio é válido através de uma expressão regular r'[0-9a-zA-Z.\-]*\.[09a-zA-Z\-]*\.\w+'. Recebe como parâmetro um nome de um subdomínio e retorna verdadeiro se o subdomínio for válido, ou falso caso contrário. Esta função é essencial para a validação da lista de domínios fornecida pelo utilizador, verificando se esta contém algum subdomínio. Esta verificação assegura que as etapas seguintes do script estejam devidamente formatadas e preparadas para serem executadas.

is_main_domain
Esta função verifica se um domínio é válido através de uma expressão regular “r'^[a-z0-9](?:[a-z0-9]{0,61}[a-z0-9])?\.[a-z0-9][a-z0-9-]{0,61}[a-z0-9]". Recebe como parâmetro um nome de domínio e retorna verdadeiro se o domínio for válido, ou falso caso contrário. Esta função é bastante importante para a validação da lista de domínios fornecida pelo utilizador, garantindo que os domínios enviados para as próximas etapas do *script* estejam corretamente formatados.

get_main_domain
Esta função é responsável por extrair o domínio principal a partir de um subdomínio fornecido como parâmetro. Esta funcionalidade é especialmente útil para processar a lista de domínios fornecida pelo utilizador, permitindo que, caso algum dos itens seja um subdomínio, seja automaticamente convertido no respetivo domínio principal.

treat_domains
Esta função tem como objetivo validar a lista de domínios fornecida pelo utilizador, identificando quais dos elementos são domínios principais e quais são subdomínios. Caso sejam detetados subdomínios, estes são convertidos para os respetivos domínios principais. Esta verificação e validação é realizada com recurso às funções *get_main_domain, is_main_domain e is_subdomain*.

valid_TLD
Esta função é responsável por devolver uma lista com os domínios válidos, verificados através de uma consulta DNS. Para cada domínio fornecido, tenta resolver o nome através do sistema DNS. Apenas os domínios que obtêm uma resolução com sucesso são considerados válidos e adicionados à lista final.

delete_aux_files
Esta função apaga ficheiros auxiliares temporários, como *cleanIPs.txt*, scans.txt e *mscan.json*, caso existam. Informa que todos os ficheiros foram apagados após a execução.

clean_useless_files
Esta função apaga o ficheiro cleanIPs.txt se ele existir. Caso contrário, exibe uma mensagem informando que o ficheiro não foi encontrado.

Security_headers.py
Neste ficheiro, estão inseridas, dentro da classe *SecurityHeaders,* algumas funções relacionadas com os cabeçalhos de segurança.

evaluate_warn
Esta função avalia o risco de um cabeçalho HTTP com base no seu conteúdo, definindo um sinalizador de alerta se for considerado inseguro. Retorna um dicionário com o estado de alerta, a presença do cabeçalho e o seu conteúdo.

test_https
Esta função verifica se um site suporta HTTPS e se o seu certificado SSL é válido. Retorna um dicionário com dois campos: *supported* (suporte a HTTPS) e *certvalid* (validação do certificado).

test_http_to_https
Esta função testa se um endereço HTTP redireciona para HTTPS, seguindo até 5 redirecionamentos.

Devolve *True* se o redirecionamento ocorrer com sucesso, ou *false* caso contrário.

check_headers
Esta função verifica a presença e o estado de cabeçalhos de segurança importantes. Permite seguir redirecionamentos e identificar falhas na segurança dos cabeçalhos HTTP/HTTPS.

create_json.py

guardar
Esta função tenta abrir e gravar os dados de domínios, previamente recolhidos e armazenados na variável jsonDominios, num ficheiro chamado teste.json. Caso haja algum erro durante o processo, este será tratado e apresentado.

guardar_json_ips

Esta função tenta abrir e gravar os dados de IPs, previamente recolhidos e armazenados na variável jsonIps, num ficheiro chamado testeIp.json. Caso haja algum erro durante o processo, este será tratado e apresentado.

clean_json_IPS
Esta função é utilizada para limpar o ficheiro JSON referente aos IPs, de modo a não ficar informação residual para próximas chamadas à API.

clean_json
Esta função é utilizada para limpar o ficheiro JSON referente aos domínios, de modo a não ficar informação residual para próximas chamadas à API.

configs.yaml
Neste ficheiro, estão contidas as configurações referentes à API, neste caso, a interface de rede, especi ficada como *masscan_interface*.

api_keys.yaml
Neste ficheiro, estão presentes todas as chaves de APIs que terão de ser configuradas para o devido funcionamento da API.

Diretório crtsh

crtshAPI
A classe *crtsAPI* contém a função referente à pesquisa, sendo responsável por realizar consultas à API para obter dados sobre certificados e subdomínios a partir de um domínio fornecido.

Search
Esta função pesquisa o domínio na base de dados crt.sh, permitindo incluir ou excluir certificados expirados. Retorna uma lista de objetos com informações detalhadas sobre os certificados encontrados.

crtsh_cert_info.py

flatten
Esta função recebe listas ou sublistas aninhadas e as expande, retornando uma estrutura plana, sem a necessidade de lidar com listas opcionais aninhadas.

check_expiration_date
Esta função recebe a data de expiração de um certificado SSL e retorna o número de dias restantes até a expiração.

check_cert

Esta função verifica o certificado SSL de um domínio, retornando detalhes como a data de expiração, a data de início e o nome da organização, ou um erro caso a verificação falhe.

check_cert_output
Esta função extrai e formata as informações do certificado SSL, incluindo a data de expiração, a data de início e o nome da organização, além de calcular o tempo restante até a expiração.

Diretório knockpy

config.py
Este ficheiro contém as configurações necessárias para poder executar as funções do *Knockp*.

Dns_socket.py

parse_dns_string
Esta função interpreta uma sequência de bytes DNS codificada, convertendo-a num nome de domínio legível, com suporte a apontadores de reutilização.

class StreamReader

Esta classe conta com diversas funções diversas utilizadas no *knockpy*.

reuse
Esta função converte pós num índice, ajusta-o se necessário e reutiliza dados a partir desse ponto no *buffer* para analisar uma *string* DNS.

make_dns_query_domain
Esta função constrói e codifica uma *string* DNS no formato de consulta, separando o domínio em partes com comprimento prefixado.

make_dns_request_data
Esta função cria uma mensagem de requisição DNS no formato binário, com cabeçalho e dados da consulta especificados.

add_record_to_result
Esta função adiciona ao resultado um registo DNS do tipo A ou CNAME, convertendo os dados conforme o tipo.

parse_dns_response
Esta função processa a resposta a uma consulta DNS, extraindo e organizando registos do tipo A e CNAME num dicionário de resultados.

dns_lookup
Esta função realiza uma consulta DNS para um domínio, enviando a requisição para o servidor especificado por *address*.

_gethostbyname_ex
Esta função resolve um domínio através de DNS, devolvendo o nome, os aliases e os endereços IPv4 associados.

wordlist.txt
Este ficheiro contém todas as palavras utilizadas para conseguir encontrar os subdomínios no processo de análise.

knockpy.py
Este ficheiro contém presentes diversas classes com as respetivas funções utilizadas na busca dos subdomínios através do *Knockpy*.

ClassRequestdns
Esta função resolve o nome de domínio alvo para obter o endereço IP. Utiliza um servidor DNS personalizado se definido na configuração (*config["dns"]*), caso contrário, utiliza o DNS padrão do sistema.

https
Esta função tenta estabelecer uma ligação HTTPS com o URL fornecido, utilizando um *user-agent* aleatório e um tempo limite definido na configuração. Retorna uma lista com o código de estado HTTP e o valor do cabeçalho *"Server",* ou uma lista vazia em caso de erro.

http
Esta função tenta estabelecer uma ligação HTTP com o URL fornecido, utilizando um *user-agent* aleatório e um tempo limite definido na configuração. Retorna uma lista com o código de estado HTTP e o valor do cabeçalho *"Server",* ou uma lista vazia em caso de erro.

bs4scrape
Esta função tenta extrair subdomínios de uma página HTML, identificando ligações que apontem para subdomínios do alvo fornecido. Devolve uma lista de subdomínios encontrados na resposta HTML, se o estado da resposta for 200.

Class Wordlist Local

Esta função lê ficheiros locais linha por linha, devolvendo apenas as linhas não vazias.

google
Esta função realiza uma pesquisa no Google para encontrar subdomínios do domínio indicado, utilizando *scraping* com *BeautifulSoup* para extrair os resultados.

duckduckgo
Esta função realiza uma pesquisa no *DuckDuckGo* para encontrar subdomínios do domínio indicado, utilizando *scraping* com *BeautifulSoup* para extrair os resultados.

get
Esta função recolhe palavras de várias fontes (local, *Google e DuckDuckGo*), conforme definido na configuração, para formar uma lista de palavras úteis na descoberta de subdomínios.

Class outputprogressPrint
Esta função é utilizada para atualizar a linha de comando (ou terminal) com um texto que muda dinamicamente, sem criar linhas.

jsonizeRequestData
Esta função organiza e estrutura informações sobre subdomínios, aliases, IPs, código HTTP e servidor no formato JSON, associando-as ao domínio de destino.

linePrint
Esta função formata e imprime uma linha de dados sobre um endereço IP, subdomínios, código HTTP, servidor e domínio. Ajusta o espaçamento de cada elemento com base no comprimento máximo definido, garantindo uma apresentação organizada e legível.

Class Startscan
Esta função realiza a verificação de DNS e HTTP(S) para um determinado domínio e subdomínio e, dependendo dos resultados obtidos, formata e armazena a informação num dicionário de resultados.

knockpy
Esta função executa uma varredura de subdomínios para um determinado domínio, usando uma lista de palavras (*wordlist*) obtida de fontes locais, *Google e DuckDuckGo*.

Diretório shodan

Neste diretório, estão presentes as funções referentes ao *Shodan*, uma das componentes principais do monitorizador. Estão presentes funções tanto para a procura de informações referentes a um determinado IP ou domínio, como também para a procura de subdomínios.

API
Esta função devolve a chave necessária para conectar à API. A chave está presente no ficheiro das chaves *(keys).* Em caso de erro, a função retorna *None.*

shodan_subdomains
Esta função procura identificar os subdomínios de um domínio através de uma consulta à API do *Shodan.* Retorna uma lista com os subdomínios encontrados ou uma lista vazia em caso de erro.

search_domain_info
Esta função obtém informações detalhadas sobre um determinado domínio ou subdomínio, realizando uma consulta à API do *Shodan*. Ela retorna um conjunto de listas contendo informações relacionadas aos IPs, subdomínios, cabeçalhos de segurança, tecnologias, portas abertas, versões TLS e serviços, com itens não repetidos. Em caso de erro, retorna listas vazias.

search_ip_info
Esta função recolhe informações detalhadas sobre um IP, realizando uma consulta à API do *Shodan*. Ela retorna um JSON contendo informações sobre o IP, portas abertas, banners, localização, organização, sistema operativo e tecnologias associadas.

Diretório unused
Neste diretório estão presentes todos os scripts não usados

Diretório CTI
Neste diretório estão presentes todos os scripts desenvolvidos para retirar informação e processá-la de diversas fontes externas utilizadas ao longo do projeto.

Leak_lookup.py

valida_tipoScan
Esta função valida se o tipo recebido é um e-mail, IP, domínio ou nome de utilizador.

valida_email
Esta função verifica se o e-mail é válido.

valida_ip
Esta função verifica se um IP é válido.

verify_domain
Esta função verifica se é um domínio válido.

consultarAPiLeakLookup
Esta função faz uma consulta à API do *Lookup*, de modo a retornar os *leaks* para um domínio, IP, email ou nome de utilizador.

guardar_json
Esta função guarda o resultado da pesquisa num ficheiro JSON.

lookup_api
Esta função principal vai solicitar os parâmetros, como a informação a extrair e o formato em que se pretende armazenar, realiza a consulta dos mesmos e processa os resultados.

Consulta_dominio_ip_api
Esta função processa os dados retornados para consultas de domínio ou IP e guarda os resultados numa lista.

Consulta_email_username_api
Esta função processa os dados retornados para consultar email ou nome de utilizador e guarda os resultados numa lista.

black.py

consulta_black
Esta função faz uma pesquisa através do *BlacklistCheckers* para verificar se um domínio ou IP estão presentes em alguma *blacklist* conhecida.

exploidb.py

Neste ficheiro estão presentes as funções referentes ao *Exploid*.

cves
Esta função recolhe *exploits* associados a um determinado CVE e retorna um dicionário com as informações dos mesmos, como *cve, exploit_title, exploit_id, exploit_link, date_published, date_added, date_updated, author, type, platform, tags, aliases e zexploit_code.*

get_exploit_data
Esta função pesquisa *exploits* relacionados a uma determinada CVE na base de dados *Exploit-DB*, utilizando a ferramenta *searchsploit*, e devolve os resultados no formato JSON.

get_exploit_code

Esta função retorna o código-fonte de um *exploit* da base de dados *Exploit-DB*, com base no seu id.

my_cve.py

Neste script estão presentes todas as funções de requisição de informação à API do NVD.

consultar_cveTec

Esta função pesquisa CVEs relacionadas com um termo fornecido, utilizando a API da *NVD,* e devolve uma lista de identificadores CVE encontrados.

consultar_cv

Esta função pesquisa CVEs relacionadas com um termo, usando a API da *NVD*, e devolve as vulnerabilidades encontradas com detalhes como descrição, versão do CVSS, pontuação e gravidade.

guardar_json

Esta função guarda os resultados da pesquisa com a informação de uma CVE no formato JSON.

netlas_domain.py

Neste script estão presentes todas as funções de requisição de informação à API do *Netlas.*

netlas_connection

Esta função estabelece uma ligação com a API do *Netlas.*

netlas_domain

Esta função consulta a API do *Netlas* para obter informações sobre um domínio e devolve a resposta ou *None* em caso de erro.

netlas_lookup

Esta função processa os dados obtidos da API do *Netlas* sobre um domínio, extraindo e organizando as informações de portas, *software* e DNS.

onyphe_domains.py

conecao

Esta função realiza uma pesquisa na API da *Onyphe* com base num domínio, devolvendo os resultados no formato JSON.

domains

Esta função obtém informações detalhadas sobre um domínio ou subdomínio, incluindo dados de rede, certificados, produtos, protocolos e respostas *HTTP, scan_info, TLS, ports e transport*, guardando tudo num dicionário que é depois retornado.

securitytrails.py

obter_dados_certificado
Esta função consulta a API da *SecurityTrails* para obter informações sobre certificados SSL válidos de um domínio, retornando detalhes como datas de validade, entidade emissora, tipo e tamanho da chave, impressões digitais , tudo retornado num dicionário.

obter_sub_dominios
Esta função consulta a API da *SecurityTrails* para obter todos os prefixos dos subdomínios ativos de um domínio fornecido e devolve uma lista desses subdomínios completos ou retorna uma lista vazia em caso de erro.

urlScan.py

consulta_urlScan
Esta função envia um pedido à API do *URLScan* para analisar um domínio ou URL e retorna o link da API com os resultados da análise, se o pedido for bem-sucedido.

obter_resultado
Esta função processa os resultados da análise feita pela API, devolvendo uma lista de servidores identificados e outra com as tecnologias detetadas do domínio, retornando duas listas com as respetivas informações ou *None* em caso de erro.

Diretório iocs_feed

Neste diretório estão presentes todos os *IOCs* para retornar informação através dos *endpoints* da API, contando com informação vinda do *OpenCTI* e respetivos dados.

Api_open_cti_country.py

CTI_pais
Esta função pesquisa os 10 ataques com maior nível de confiança relacionados com um país específico, utilizando a API do *OpenCTI*, e devolve as informações mais relevantes sobre cada ataque, como atacante, tipo de ataque, confiança e TLP.

CTI_pais2
Esta função permite pesquisar os ataques relacionados com um país específico, utilizando a API do *OpenCTI*. A função oferece a possibilidade de filtrar os ataques dentro de um intervalo de datas, com base nos parâmetros mes_inicio e mes_fim, caso sejam fornecidos. Os dados dos ataques são extraídos e organizados num dicionário, que

contém as informações detalhadas sobre o atacante, o tipo de ataque, o alvo, a confiança, TLP *(Traffic Light Protocol)* e a data de criação.

Api_open_cti_ob.py

cti_Ips
Esta função recolhe todos os indicadores de endereços IP (IPv4 e IPv6) criados hoje na plataforma *OpenCTI*. Os dados são armazenados e retornados via dicionário.

cti_domains
Esta função recolhe todos os indicadores de domínios criados hoje na plataforma *OpenCTI*. Os dados são armazenados e retornados via dicionário.

cti_hashes
Esta função recolhe todos os indicadores do tipo *hashes* criados hoje na plataforma *OpenCTI*. Os dados são armazenados e retornados via dicionário.

OxSl_f33d.py

obter_user_api_key
Esta função devolve a chave necessária para conectar à API, caso esta esteja presente no ficheiro das chaves. Em caso de erro, retorna *None.*

obter_password_api_key
Esta função devolve a chave necessária para conectar à API, caso esta esteja presente no ficheiro das chaves. Em caso de erro, retorna *None.*

consultar0xsl_f33d_api
Esta função estabelece ligação à API do OxSI_f33d, recolhe os dados da última semana e retorna-os numa lista de *strings*.
consultar0xsl_f33d
Esta função estabelece ligação à API do OxSI_f33d, recolhe os dados de um determinado período, e permite opcionalmente indicar um título ou URL a ser pesquisado, retornando-os numa lista de *strings* com os dados encontrados.

guardar_lista
Esta função guarda os dados retornados obtidos num ficheiro.

Monitorizador.py

Este script é o responsável pela API REST desenvolvida em *Flask*, que fornece diversos endpoints, como análise de ameaças, verificação de vulnerabilidades (CVEs), *feed* de inteligência de ameaças, informações relacionadas a IPs, domínios e *dataleaks*. Cada

função no monitorizador.py é responsável por um *endpoint* (rota) da API. Abaixo está uma breve descrição de cada função:

run_0xSl_f33d
Esta função consulta o *feed* do projeto 0xSl, desenvolvido e suportado pela SegurancaInformatica.pt. Os dados retornados correspondem, por padrão, à última semana (esse período pode ser ajustado na função consultar0xsl_f33d_api()). O *feed* inclui URLs reportadas por utilizadores portugueses, relacionadas com campanhas de *phishing e malware*. Os dados são retornados em formato JSON através do endpoint: /0xSl_feed/[GET].

run_cti
Esta função permite consultar dados da plataforma *OpenCTI* conforme o tipo de scan especificado no *endpoint*, aceitando um JSON como entrada para configuração dos parâmetros. Estão disponíveis quatro tipos de scan: TOP10, que é o único que requer um parâmetro de entrada (o país) e retorna os 10 ataques com maior nível de confiança associados a esse local; *hashes,* que recolhe todos os indicadores baseados em *hashes* do dia atual; *domain,* que consulta indicadores do tipo domínio; e ips, que consulta indicadores do tipo IP. Todos os resultados são devolvidos em formato JSON através do *endpoint*:
/cti/TypeScan/[POST].

Cti
Esta função oferece as mesmas funcionalidades da *run_cti(TypeScan),* com exceção do tipo de scan TOP10, que não é suportado nesta versão e, portanto, não permite a consulta por país. A principal diferença é que esta função utiliza um *endpoint* do tipo GET, permitindo que seja acedida diretamente através do navegador. Os resultados mantêm o formato da função *run_cti* e podem ser obtidos através do endpoint: /ctiweb/TypeScan[GET].

pais,pais2, pais3 ,pais4
Estas quatro funções partilham a mesma funcionalidade do tipo de scan TOP10 da função *run_cti(),* mas estão disponíveis através de endpoints do tipo GET, permitindo o seu acesso direto via browser. A principal diferença é que estas variantes permitem, opcionalmente, definir um limite temporal: o parâmetro typeScan2 indica o mês de início e typeScan3 o mês de fim da consulta. Caso não sejam fornecidos, não é aplicado qualquer limite temporal, sendo retornados todos os resultados disponíveis. Os resultados têm o mesmo formato da função *run_cti()* e podem ser obtidos através dos seguintes endpoints: /ctipais/typeScan/typeScan2/typeScan3/[GET], /ctipais/typeScan/typeScan2/[GET], /ctipais/[GET].

run_cvescript

Esta função permite consultar dados sobre vulnerabilidades CVE através do módulo *exploitdb.* Aceita um JSON com uma lista de identificadores CVE como entrada e devolve os resultados em formato JSON. As consultas são realizadas via o endpoint /CVE/[POST].

run_cve
Esta função permite consultar dados sobre vulnerabilidades CVE utilizando a API oficial do NVD (*National Vulnerability Database*). Aceita um JSON contendo uma lista de identificadores CVE como entrada e devolve os resultados em formato JSON. Cada resultado inclui uma descrição resumida da vulnerabilidade, o CVSS (*Common Vulnerability Scoring System*), os valores de severidade ,bem como informação adicional relevante sobre o tipo de falha (fraude, execução remota, etc.). As consultas são efetuadas através do endpoint /pesquisar-cve/[POST].

run_lookupScript
Esta função permite consultar informações associadas a emails, nomes de utilizador, domínios e endereços IP através da API do *LeakLookup*. A entrada é feita em formato JSON, indicando o tipo de entidade a consultar (por exemplo: email, username, domínio ou ip) e o respetivo valor. Os resultados, também em formato JSON, são obtidos através do *endpoint* /LOOKUP/[POST].

run_monitorizador
Esta função permite consultar domínios e endereços IP, utilizando dados processados pelos ficheiros *dom_checker.py e ips.py,* os quais serão explicados mais à frente neste documento. Suporta dois tipos de scan: IP ou DOM. A entrada, composta pelos IPs ou domínios a analisar, deve ser fornecida em formato JSON, e os resultados, também em formato JSON, são disponibilizados através do *endpoint* /monitorizador/TypeScan[POST].

Monitorizador
Esta função oferece as mesmas funcionalidades da *run_monitorizador(),* com a principal diferença de que pode ser acedida diretamente através do browser, uma vez que utiliza o método GET. Os resultados são devolvidos em formato JSON e o acesso é feito através do *endpoint*: /monitorizadorweb/TypeScan/Type_scan_web/[GET], onde o TypeScan é o tipo de scan a ser realizado e o *type_scan_web* representa o conteúdo a ser analisado (ips ou domínios).

Teste_conecao_api.py

Este script foi desenvolvido para tornar o teste de todos os endpoints da API o mais fácil possível, com exceção do monitorizadorweb. Desta forma, é possível fazer a requisição dos endpoints através de código, tornando mais fácil realizar as consultas.

Anexo D

Este anexo tem como objetivo apresentar o documento realizado com o intuito de proceder à instalação do OpenCTI numa máquina virtual (VM). O propósito é configurar uma plataforma de *threat intelligence* que permita a ingestão e análise de dados. É importante notar que a documentação original sofreu pequenas alterações para ser adaptada a este anexo.

Instação do Docker

Começamos por instalar o Docker na nossa máquina: **sudo apt install docker-compose**.

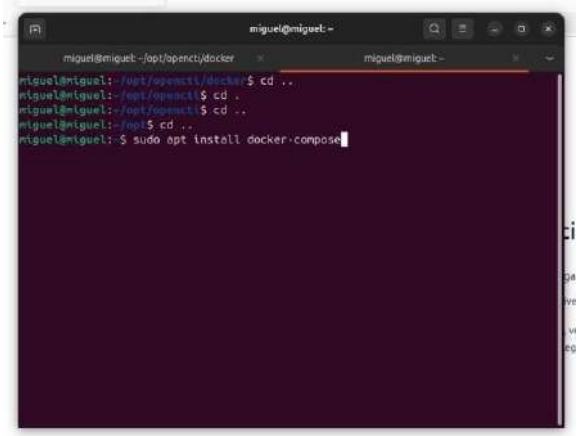


Figura 73-Instalação do Docker-compose.

Após a instalação bem-sucedida do Docker, vamos agora criar o caminho de pastas até ao local onde iremos clonar o repositório Git da OpenCTI, utilizando os comandos **sudo mkdir opt** e **sudo mkdir opencti**.

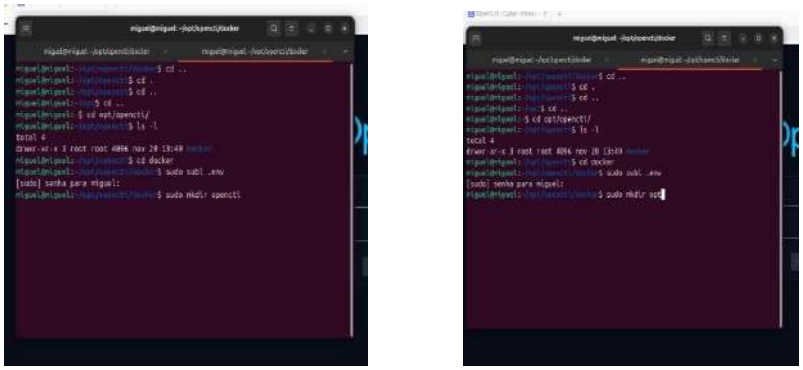


Figura 74-Criação das pastas para clonar o repositório do Git.

Instalação do Repositório git

Depois de criar o caminho, copiamos o repositório Git com o comando **git clone** https://github.com/OpenCTI-Platform/docker.git e, em seguida, acedemos à pasta com o comando **cd Docker**

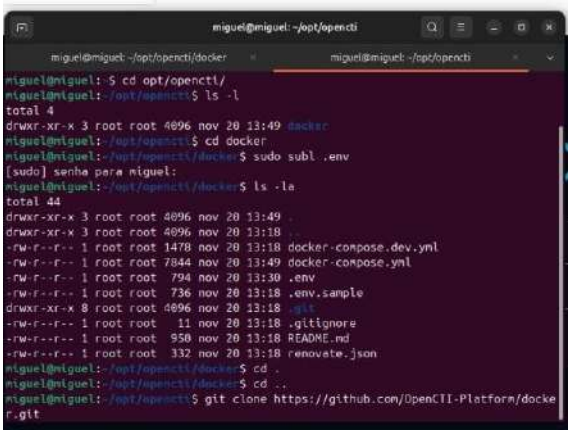


Figura 75-Clonar o git do openCTI.

Configuração do OpenCTI

Utilizando o comando **ls -la** para verificar todos os ficheiros presentes, observamos um. env.sample, que contém um exemplo do ficheiro que teremos de configurar para que a instalação seja bem-sucedida.

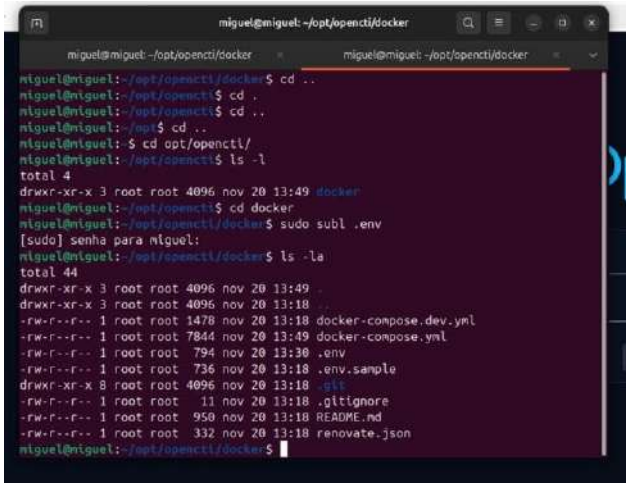


Figura 76-Verificar ser o env.sample esta presente

Passamos, então, a fazer uma cópia do ficheiro, renomeando-o para. env, que será o ficheiro que o sistema irá utilizar na configuração.

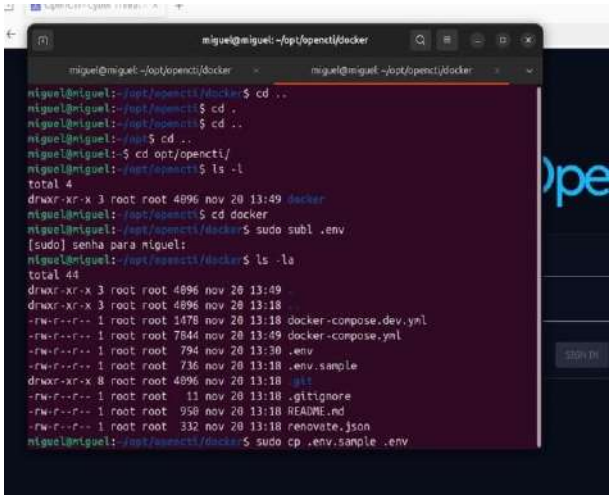


Figura 77-Criação do ficheiro chamado. env para as configurações do OpenCTI

Abrimos o ficheiro diretamente na consola ou utilizando o [Sublime Text] para ser mais fácil fazer a edição: sudo subl .env.

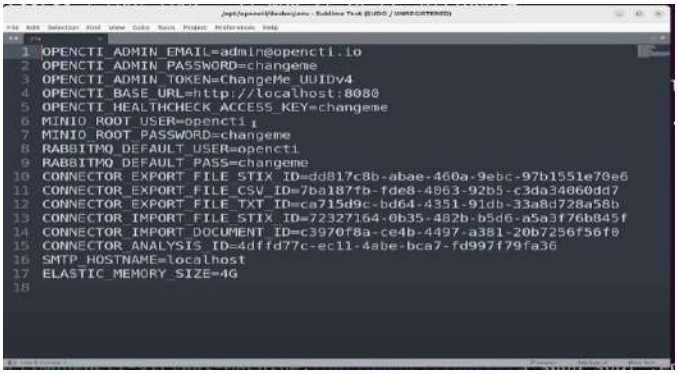


Figura 78-Configurações do OpenCTI

Iremos, então, configurar todas as linhas que contiverem “changeme”. No atributo TOKEN, devemos consultar [uuidv4] e copiar o token que aparece. Após isso, devemos atualizar a página (refresh) disponível nesse mesmo site e colar o novo token no campo HEALTHCHECK_ACCESS_KEY, substituindo os restantes parâmetros "changeme", de forma a ficar semelhante à **Erro! A origem da referência não foi encontrada.** Após concluir esta configuração, devemos guardar as alterações do ficheiro e fechá-lo.

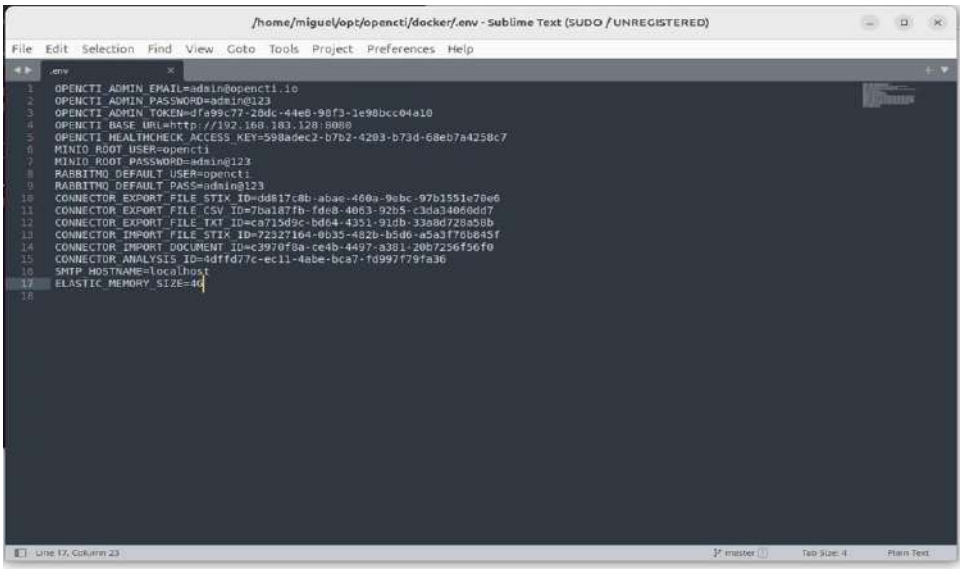


Figura 79-Configurações do OpenCTI

Iniciação do Docker com o OpenCTI

Após a configuração, iremos iniciar o serviço Docker utilizando o comando **sudo systemctl start docker.service** e, para verificar se está operacional, utilizamos o comando **sudo systemctl status docker.service**.

Figura 80-Verificar o funcionamento do Docker.

Para terminar a instalação, iremos, então, executar o comando **sudo docker-compose up -d**, que, da primeira vez, poderá demorar algum tempo. Após a conclusão, a instalação estará finalizada.

```
miguel@miguel:~/opt/opencti/docker$ sudo docker-compose up -d
docker_rabbitmq_1 is up-to-date
docker_redis_1 is up-to-date
docker_minio_1 is up-to-date
docker_elasticsearch_1 is up-to-date
docker_opencti_1 is up-to-date
docker_connector-import-document_1 is up-to-date
docker_connector-export-file-stix_1 is up-to-date
docker_connector-export-file-txt_1 is up-to-date
docker_connector-analysis_1 is up-to-date
docker_worker_1 is up-to-date
docker_worker_2 is up-to-date
docker_worker_3 is up-to-date
docker_connector-export-file-csv_1 is up-to-date
docker_connector-import-file-stix_1 is up-to-date
```

Figura 81-Utilização do Docker-compose up para iniciar o OpenCTI.

Abertura do OpenCTI

Abrimos o motor de busca e acedemos a uma das seguintes páginas: http://localhost:8080/ ou http://ipdamaquina:8080/ . Introduzimos as credenciais que foram configuradas no ficheiro. env e, assim, estaremos dentro do OpenCTI, concluindo a instalação.

Figura 82-Página inicial do openCTI

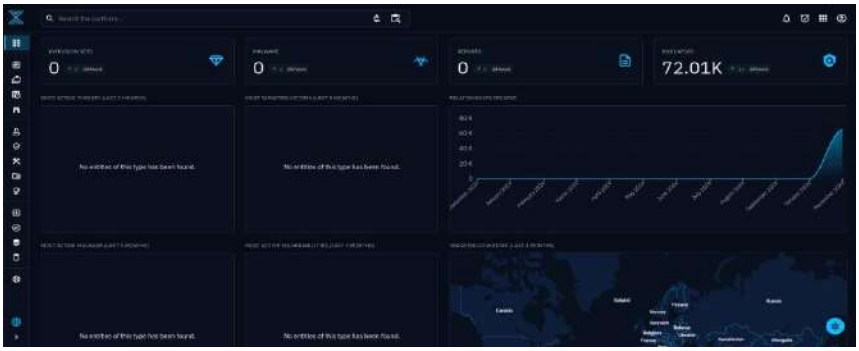


Figura 83-Página inicial do OpenCTi