UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Análise e Proposta de Melhorias em Simulacros de Phishing

Fernando Jorge Fernandes Gonçalves

Mestrado em Engenharia Informática

Versão Pública

Trabalho de Projeto orientado por:
Professor Doutor Luis Antunes

2023

Agradecimentos

Ao longo destes dois anos suei, ri, chorei, caí e levantei-me. É com satisfação que vejo esta etapa da minha vida a chegar ao fim.

Na faculdade, quero agradecer ao meu orientador, Professor Doutor Luis Antunes por me conseguir fazer perceber o caminho certo e que eu era capaz.

Na Emvenci, quero agradecer ao Alexandre Aniceto pela orientação que me deu na resolução de vários tipos de problemas. Quero agradecer também aos meus colegas da empresa Carlos Carvalho, Diogo Silva, Luís Carneiro, Maria Silva, Miguel Silva, Tiago Murteira e Vasile Parasca por toda a ajuda e diversão.

Quero agradecer à minha namorada Mariana Pernes por me ajudar sempre em tudo o que era preciso e a melhorar em tudo.

Quero agradecer especialmente aos meus pais, Joaquim Gonçalves e Lurdes Fernandes não só por me terem apoiado e incentivado a começar e finalizar esta etapa como também tudo na minha vida.

Quero também agradecer aos meus amigos que me ajudaram e mostraram sempre disponibilidade.

Por último quero agradecer ao meu filho Miguel Gonçalves por apenas, com a sua existência e sem nenhuma palavra, me ajudar a passar qualquer barreira nesta vida.

Um grande obrigado a todos.

Dedicatória.

Resumo

O *Phishing* é uma das técnicas de ciberataque mais utilizadas atualmente, contando com várias formas de obter informações sensíveis sobre as vítimas. Revela-se uma problemática pertinente a nível global, que atinge tanto indivíduos como empresas, com uma incidência tendencialmente crescente. Deste modo a presente dissertação foca essencialmente a análise e melhoria do simulador de *Phishing* da EMVENCI, tendo em conta as necessidades atuais do mercado. Este estudo permite o investimento em técnicas preventivas adequadas. Uma forma de prevenção proposta é a melhoria nos programas de treino para os colaboradores que, ao fornecerem uma maior capacidade de deteção de *Phishing*, provocam uma diminuição do impacto negativo destes ataques na empresa. Entre as melhorias implementadas, destacam-se a inclusão de simulacros de treino de *Phishing* com suporte de dispositivos USB e NFC e a criação de grupos dinâmicos de utilizadores (Dynamic groups).

Palavras-chave: *Phishing*, Treino, Ciberataque, USB, NFC

Abstract

Nowadays Phishing is one of the most commonly used cyberattack techniques, with various ways of obtaining sensitive information about victims. It is a relevant global issue that affects both individuals and companies with an increasing incidence. This dissertation essentially focuses on the analysis and improvement of EMVENCI's Phishing simulator, taking into account current market needs. This project focuses on the appropriate preventive techniques. One proposed form of prevention is the improvement of training programs for the company's employees by providing greater detection capability of Phishing attacks. Doing this will lead to a decrease in the negative impact on the company. Among these improvements, the main ones are the addition of Phishing training simulations that support the use of USB and NFC devices and the implementation of Dynamic Groups of users.

Keywords: *Phishing*, Training, Cyberattack, Improvements

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas

BEC – Business Email Compromise
CEO – Chief Executive Officer
CFO – Chief Financial Officer
CRUD – Create, Read, Update, and Delete
CSV – Comma-Separated Values
HTML – Hypertext Markup Language
HTTP – Hypertext Transfer Protocol
IM – Instant Messaging
JSON – JavaScript Object Notation
JWT – JSON Web Token
KB – Knowledge Base
NFC – Near Field Communication
PDF – Portable Document Format
REST – Representational State Transfer
RGPD – Regulamento Geral de Proteção de Dados
RPC – Remote Procedure Call
SAAS – Software as a Service
SMS – Short Message Service
UA – URL Agent
URL – Uniform Resource Locator
USB – Universal Serial Bus
WIFI – Wireless Fidelity

Capítulo 1

Introdução

A presente dissertação em Engenharia Informática, orientada pelo Professor Doutor Luis Antunes e supervisionada por Alexandre Aniceto, CEO da EMVENCI, é dedicada à análise e melhoria dos simuladores de *Phishing* da EMVENCI. A EMVENCI é uma empresa de cibersegurança que desenvolve a Cybersecurity Cloud, uma plataforma em SaaS *(Software as a Service)* com vários módulos: simulador de *Phishing*, formação de cibersegurança (*eLearning*), gestor de políticas de segurança, plataforma de registo dos requisitos RGPD (Regulamento Geral de Proteção de Dados), gestor de vulnerabilidades e plataforma de centralização e gestão de *logs*. O projeto tem como foco a análise do módulo de simulacros de *Phishing*, a fim de acompanhar as tendências mais recentes de ataques de *Phishing* e treinar os colaboradores de forma adequada.

1.1 Motivação

Atualmente, o *Phishing* é uma das técnicas mais comuns de ciberataque, que utiliza várias estratégias para obtenção de informações sensíveis sobre os utilizadores, tais como nome de utilizador, palavras-passe, detalhes dos cartões de crédito, entre outros [35]. Devido ao aumento da incidência destes ataques revela-se cada vez mais importante o seu estudo e mitigação [6]. Em consequência deste aumento exponencial e generalizado, têm surgido novas variantes de ataques de *Phishing*: *Vishing, Smishing, Pharming, Whaling, eFax, Instant Messaging* (IM), etc. [2].

Face ao impacto negativo destes ataques nas empresas, surge a necessidade de investir em técnicas preventivas, destacando-se os programas de treino para os colaboradores. Estes programas fornecem aos colaboradores uma maior capacidade de deteção de ataques de *Phishing*, diminuindo o seu impacto na empresa. Uma vez explicitada a pertinência desta problemática, o objetivo deste projeto passa por analisar os simulacros de *Phishing* já existentes e propor melhorias com funcionalidades novas e adaptadas às necessidades atuais do mercado.

1.2 Objetivos e contribuições

Os objetivos deste projeto são: explorar os requisitos funcionais existentes; analisar a discrepância entre as capacidades atuais e os ataques simulados de acordo com as tendências atuais; propor uma arquitetura para as funcionalidades de *Dynamic Groups* e campanhas de USB e NFC; avaliar e priorizar as opções disponíveis com definição de tarefas e nível de esforço; prototipar e desenvolver os casos propostos e, por fim, desenhar testes para validar os objetivos finais do projeto.

As minhas contribuições para este projeto foram: análise e desenvolvimento dos *Dynamic Groups*; análise e desenvolvimento das campanhas de USB e NFC e, por fim, desenho de teste para ambas as soluções.

1.3 Planeamento

O projeto da presente dissertação, enquadrado no estágio na EMVENCI, foi planeado de acordo com a organização e métodos da empresa onde são realizados ciclos de trabalho de 2 semanas. Como se pode verificar no gráfico de Gantt, representado na Figura 1.1, o meu planeamento é o seguinte:

- Análise dos requisitos funcionais existentes. Fazer o alinhamento com as últimas tendências e enquadramento no sistema empresarial atual (12/09/2022 a 06/01/2023);
- Análise da discrepância entre as capacidades atuais e os ataques simulados de acordo com as últimas tendências (1/12/2022 a 13/01/2023);
- Proposta da arquitetura de funcionalidades para mitigar os ataques (16/01/2023 a 03/02/2023);
- Análise e priorização das opções com definição de tarefas e nível de esforço, de modo a planear e organizar em *Sprints* de desenvolvimento (06/02/2023 a 10/02/2023);
- Prototipar e desenvolver os casos propostos (13/02/2023 a 21/04/2023);
- Desenhar testes para validar os objetivos finais do projeto (24/04/2023 a 26/05/2023);
- Escrita e unificação do projeto de Dissertação (a partir de 26/05/2023).

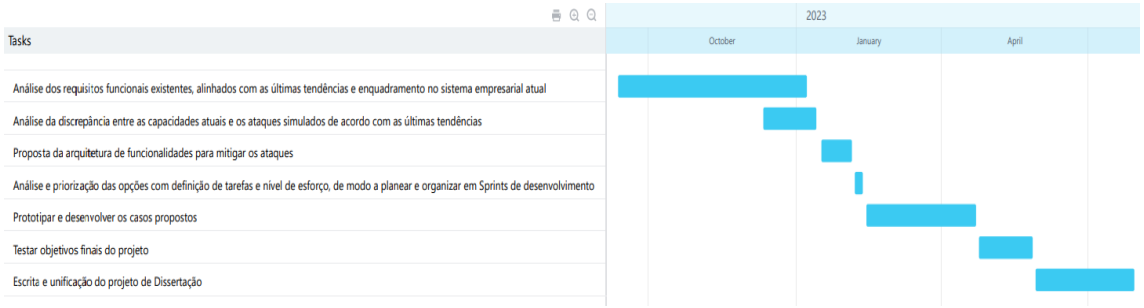

Figura 1.1: Gráfico de Gantt

1.4 Estrutura do Documento

Este documento encontra-se estruturado em sete capítulos. O primeiro capítulo aborda a apresentação da EMVENCI e a explicação da necessidade de explorar os desafios atuais relacionados com o Phishing, que constituem os objetivos centrais deste projeto. No segundo capítulo, denominado ”Conceitos e Definições”, são expostos os diversos tipos de ataques de *Phishing* e as medidas existentes para combatê-los. Pretende-se também descrever as atividades desempenhadas na empresa no âmbito de simulacros e treinos, bem como os módulos disponíveis, os benefícios da plataforma e as tecnologias empregues. No terceiro capítulo, é apresentada uma breve contextualização dos simulacros de Phishing executados na empresa, tanto ao nível do utilizador como dos alvos, de modo a proporcionar uma melhor compreensão do que será abordado. No quarto capítulo, são levantados os requisitos de análise para ambas as melhorias implementadas, incluindo os requisitos funcionais, não funcionais e histórias de utilizador contendo várias ações possíveis (*user stories*). No quinto capítulo, expõe-se o planeamento de ambas as melhorias, bem como uma análise preliminar das alterações. No sexto capítulo, é apresentada a implementação e descrição dos componentes desenvolvidos, além dos diferentes testes a serem realizados na solução. O sétimo e último capítulo abarca uma conclusão acerca do trabalho realizado, bem como uma reflexão sobre possíveis trabalhos futuros e melhorias a serem implementadas.

Capítulo 2

Conceito e Definições

Neste capítulo, explora-se o amplo espectro do mundo do Phishing, abrangendo desde os diversos tipos de ataques, como *Spear Phishing, Vishing, Smishing, Whaling, Wi-Fi Phishing, Business Email Compromise, NFC Phishing* e *USB Phishing*, até às medidas de combate existentes para diversas ameaças. Além disso, são analisados casos de estudo de treino contra *Phishing* e discutidas as tecnologias e arquiteturas usadas na Cybersecurity Cloud, como SaaS e *Multi-Tenancy*, arquitetura REST, linguagem Golang, *Clean Architecture* e base de dados MariaDB.

Também se aprofundam as metodologias de desenvolvimento de *software*, comparando abordagens tradicionais e ágeis, e destacando as suas vantagens e desvantagens. É exposta a autenticação por token JWT e o seu papel na garantia da segurança. Por fim, são investigadas a tecnologia NFC e as suas *tags*, bem como os treinos atualmente realizados na Cybersecurity Cloud.

2.1 Ataques de *Phishing*

O *Phishing* é um tipo de ataque informático que existe há muito tempo e tem evoluído significativamente desde os anos 90 [34]. Com a evolução das formas de prevenção, os ataques de *Phishing* também se têm tornado cada vez mais sofisticados, surgindo novas variantes como o *Spear Phishing, Vishing, Smishing, Whaling, WiFi Phishing, Business Email Compromise* (BEC), USB *Phishing,* NFC *Phishing e QRCode Phishing* descritas em seguida [12, 3, 28, 26].

2.1.1 *Spear Phishing*

O *Spear Phishing* é um tipo de ciberataque realizado através do envio de *e-mails* fraudulentos, direcionado a indivíduos ou organizações específicas. O conteúdo dos *e-mails* é aparentemente inofensivo e pode até incluir informações pessoais, como nome e profissão, para não levantar suspeitas. Para facilitar o trabalho de busca por informações detalhadas sobre os alvos, o atacante pode utilizar, por exemplo, uma plataforma conhecida como Linkedin. Adicionalmente, há que ter em conta que estes *e-mails*, à primeira vista, parecem ter sido enviados por pessoas conhecidas pelas vítimas, aumentando substancialmente a probabilidade de abrirem o *e-mail*. De notar que o foco em alvos específicos requer, naturalmente, mais tempo e esforço, no entanto pode vir a ser mais recompensador [2].

2.1.2 *Vishing*

O *Vishing* é uma técnica de *Phishing* que envolve sobretudo o uso da voz - daí o termo *"Vishing"*, derivado das palavras *"Voice"* e *"Phishing"* [2]. A *Voice over Internet Protocol*, ou VoIP, em conjunto com outras tecnologias modernas, permite esconder a localização real do atacante e reduz o custo de realizar chamadas a um nível mínimo, praticamente insignificante. O atacante utiliza técnicas de engenharia social para tentar obter dados confidenciais da vítima, como informações pessoais e financeiras, com o objetivo de obter uma recompensa [10].

2.1.3 *Smishing*

Smishing é um tipo de *Phishing* que utiliza serviços de mensagem de texto (SMS) para enganar as vítimas. Existem dois métodos principais de *Smishing*. O primeiro consiste em enviar uma mensagem de texto com uma suposta origem confiável, como um banco ou uma empresa nacional, com uma hiperligação que redireciona para um *website* malicioso de forma a obter os dados inseridos. O segundo método envolve o envio de mensagens de texto com *malware*. Quando a vítima clica na hiperligação ou acede ao *website*, o *malware* é instalado no seu dispositivo, permitindo ao atacante ter acesso aos seus dados pessoais ou até mesmo controlar o dispositivo. Isto permite que o atacante se autentique como a vítima e consiga até fazer compras em seu nome [42].

2.1.4 *Whaling*

O *Whaling* é uma técnica baseada em *Spear Phishing* que visa atingir alvos de alto nível, como um CEO, CFO ou outras pessoas possuidoras de uma grande riqueza e influência, também conhecidas como "baleias"ou *"whales"*. Para efetuar o ataque, os atacantes colecionam informações sobre as vítimas, por exemplo, através de redes sociais. Quando se trata de indivíduos-alvo inseridos no mundo empresarial, o objetivo é roubar informações confidenciais da empresa, uma vez que essas pessoas têm acesso à maioria dos dados da empresa [10, 23].

2.1.5 *WiFi Phishing*

O *Wi-Fi Phishing* é uma forma de ataque de *Phishing* que geralmente ocorre em *hotspots* públicos, com o objetivo de enganar as vítimas, fazendo com que revelem informações sensíveis como credenciais de autênticação ou dados pessoais. A técnica mais comum é instalar *malware* no dispositivo da vítima para recolher essas informações ou direcioná-la para *websites* maliciosos. Para além desta técnica, existem também outras que intercetam o tráfego de rede em *hotspots* públicos, para roubar informações pessoais transmitidas pelos utilizadores [2].

2.1.6 *Business Email Compromise* (BEC)

O *Business Email Compromise* (BEC) é uma técnica de *Phishing* que visa atacar empresas, ao invés de indivíduos, para obter dinheiro ou informações confidenciais. A engenharia social é um elemento central deste tipo de ataque, pois os atacantes realizam tentativas de aceder aos dados

de funcionários, posteriormente procuram estabelecer relacionamentos com eles para ganhar a sua confiança e, por fim, conseguem obter dinheiro ou informações confidenciais [1, 13].

2.1.7 *NFC Phishing*

A *Near Field Communication* (NFC) é uma tecnologia de comunicação por proximidade, que permite a troca de dados entre dispositivos com um simples toque entre os mesmos. Entre os dispositivos compatíveis com esta tecnologia estão os *tags* NFC - dispositivos de pequena dimensão (e.g., autocolante anti-roubo usado em lojas) que armazenam dados que podem ser enviados a um leitor de NFC quando o utilizador os aproxima. Facilmente conseguimos perceber como é que esta tecnologia pode ter tanto de utilidade como de suscetibilidade a burlas. Surge então uma nova forma de *Phishing*, o *Phishing* NFC. Tal como o BEC, esta é uma técnica baseada em engenharia social, uma vez que exige o estabelecimento de uma relação interpessoal para o sucesso do ataque. Esta técnica de *Phishing* é simples e económica, basta um pequeno esforço do atacante, que tenta persuadir as vítimas a ler uma *tag* NFC que, se alterada ou substituída, pode levar as vítimas a revelar os seus dados pessoais ou direcioná-las para aplicações maliciosas [28, 38].

2.1.8 *USB Phishing*

O *USB Phishing* é uma método de *Phishing* baseado na técnica da carta perdida [31, 16], que consiste em deixar cartas perdidas, como o próprio nome indica, em locais estratégicos e observar o comportamento das pessoas (i.e., se retornam a carta ao destinatário). Essencialmente é o que sucede com o USB *Phishing*, sendo que, em vez de cartas, o atacante utiliza uma *pen drive* USB - um dispositivo de armazenamento que contém uma memória *flash* com uma interface USB [22]. Assim, pretende-se que a vítima insira a *pen drive* no seu dispositivo e, deste modo, o atacante terá acesso direto aos sistemas pessoais ou empresariais da mesma. Este acesso permitirá ao atacante aceder a dados confidenciais da vítima, bem como instalar *backdoors* para obter acesso contínuo no sistema [26].

2.1.9 *QRCode Phishing (QRishing)*

Os *Quick Response Codes* QR *Codes*, conhecidos como códigos QR, são códigos de barras tridimensionais em preto e branco que armazenam informações. Para aceder a estes dados, é necessário ter um leitor ótico que descodifique o código. Ao fazê-lo, o leitor vai abrir o URL ou redirecionar para uma aplicação sem a autorização do utilizador. Por conseguinte, apresenta aqui logo à partida uma grande suscetibilidade a burlas, uma vez que além de não ser possível decifrar os códigos manualmente, os códigos QR geralmente têm *link hiding* e *URL shorteners* associados. Assim, o atacante pode facilmente criar códigos QR e espalhá-los em locais estratégicos, por exemplo aparentando ser uma publicidade ou até mesmo um anúncio de uma empresa. O ataque é bem sucedido quando a vítima faz *scan* do código e, ao ser redirecionada para um *website* malicioso, podem ser solicitados os seus dados pessoais ou até mesmo ser inserido *malware* no seu dispositivo [2, 26, 40].

2.2 Medidas de Combate ao *Phishing*

Atualmente, existem duas metodologias para prevenir ataques de *Phishing*: técnicas e não técnicas [5]. As metodologias técnicas incluem: uso de filtros de *e-mail*, geralmente utilizados para redirecionar para a caixa de *spam*; *software* de análise de *e-mails* com aprendizagem automática (*machine learning*), para uma melhor identificação de padrões e nomenclaturas comuns; extensões de navegadores e métodos de autenticação de dois fatores [21]. As metodologias não-técnicas incluem: políticas e procedimentos em ambientes empresariais para detetar, mitigar e relatar ataques de *Phishing* antecipadamente; uso de senhas fortes e diversificadas, geradas por geradores de palavras-passe, para manter os dados dos utilizadores mais seguros e treino de utilizadores para detetar e evitar ataques [20]. A melhor proposta para uma empresa é a utilização combinada das duas abordagens. Desta forma é possível não só ter colaboradores treinados para situações conhecidas, mas também sistemas automatizados de deteção e *feedback* [11, 32, 37].

2.3 Casos de Estudo

Como mencionado anteriormente, a necessidade dos treinos dos colaboradores surge do aumento da incidência dos ataques de *Phishing* nas empresas. Vários estudos comprovam o potencial de utilizar a intervenção de treino de colaboradores como um meio de combater ataques de *Phishing* [41, 25, 4]. Em seguida encontram-se três tipos de treino de colaboradores que, testados em ambiente real, se revelaram mais eficazes quando comparados a métodos tradicionais de treino não interativos como, por exemplo, o envio tradicional de *e-mails* de treino.

Wen et al. (2019) desenvolveram uma nova metodologia de treino de utilizadores, que ultrapassa o método tradicional de ensino - o jogo What.Hack, uma forma lúdica e ativa de treinar os colaboradores. Este jogo não só ensina os conceitos de *Phishing*, como simula os ataques atuais num momento de *role-play*; o objetivo é incentivar o jogador (utilizador) a defender-se deste tipo de ataques de cibersegurança [41]. Este estudo comprova a eficácia deste jogo pois, devido em grande parte ao seu *design* mais atrativo, aumenta a capacidade dos colaboradores para reconhecer e evitar ataques de *Phishing*. Estes autores também constataram um facto interessante - esta metodologia não só se revela mais eficaz do que uma forma básica de treino (e.g., enviar *e-mails* informativos), mas também quando comparados a jogos com um desenho competitivo (i.e., que não simulam o *Phishing* através de *role-play*).

Kumaraguru et al. (2009) criaram um sistema de treino que, de uma forma diferente da dita "tradicional", ensina os utilizadores a detetar ataques de *Phishing*. Este treino, intitulado PhishGuru, consiste no envio de *e-mails* aos colaboradores de uma forma aleatória - i.e., sem o utilizador saber que está a ser testado e treinado. Estes *e-mails* contém um URL de *Phishing* simulado e, caso o utilizador clique no URL, surge uma mensagem de treino. Os materiais de treino apresentados nestas mensagens não são meramente informativos, têm um caráter lúdico; são apresentadas fai-

xas de banda desenhada que define *Phishing*, explica os passos a seguir para evitar ser enganado num ataque de *Phishing* e ilustra a facilidade com que os criminosos realizam este tipo de ataques [25]. Estes autores encontraram uma significativa redução no número de ataques de *Phishing* bem-sucedidos relatados depois do treino com o PhishGuru.

Alnajim e Munro (2009) elaboraram uma abordagem *anti-phishing*, com o intuito de verificar se o utilizador está atento e é capaz de detetar e evitar *Phishing* enquanto navega na *internet*. O funcionamento desta abordagem é semelhante ao treino mencionado anteriormente - se o utilizador tentar submeter dados sensíveis a um *website* de *Phishing*, é apresentada uma mensagem de treino para um melhor entendimento acerca dos *websites* de *Phishing*, i.e., o que são, como funcionam e como podem ser detetados e evitados [4]. Nos casos em que o utilizador reconhece o *Phishing*, evitando-o, não é realizada nenhuma intervenção, permitindo ao utilizador continuar a navegar normalmente na internet. Estes autores descrevem o processo desta nova abordagem, baseada no uso de listas negras para detetar os *websites* de *Phishing* e com três componentes essenciais: *Proxy,* URL *Agent* (UA) e *Knowledge Base* (KB). Como referido anteriormente, a intervenção de acordo com esta abordagem tem lugar entre a *internet* e os utilizadores - quando o utilizador abrir a página URL no *browser* e clicar para tentar submeter a informação, o UA verifica se o URL se encontra na lista negra ou não; se não estiver, o *Proxy* permite prosseguir com o processo de submissão; se estiver na lista negra, o *Proxy* impede que a informação seja submetida e é apresentada uma mensagem de treino, com vista a uma melhor compreensão acerca do *Phishing* em geral, bem como formas de o detetar eficazmente no futuro. Esta abordagem *anti-phishing* é inovadora no sentido em que permite um processo de treino *anti-phishing* contínuo (i.e., não é realizado num único momento, mas ao longo do tempo), o que significa que sempre que o utilizador tentar submeter informação a um *websites* de *Phishing*, será alertado e treinado [4].

2.4 Cybersecurity Cloud

A Cybersecurity Cloud é uma plataforma SaaS e *Multi-Tenant* desenvolvida pela EMVENCI onde o módulo de simulacros de *Phishing* é a solução de maior sucesso da empresa. Esta solução baseia-se na configuração de simulacros para identificar se os colaboradores a ser testados clicam em certas hiperligações e submetem dados. A implementação a nível do *backend* é feita em *Go (Golang)* com o uso de *Clean-Architecture*, tem uma arquitetura RESTful e usa uma base de dados *MariaDB*.

2.4.1 Plataforma *SaaS e Multi-Tenancy*

A plataforma *Multi-Tenant* SaaS *(Software as a Service)* tem como objetivo a existência de apenas uma instância de *software* que atende a vários clientes. Neste tipo de plataforma, *tenants* são clientes que podem utilizar o *software* concorrentemente e que são independentes de outros clientes [9]. A sua arquitetura possui três componentes principais: aplicação, armazenamento de dados (contém os dados específicos de cada *tenant*) e armazenamento de dados partilhado (contém

dados compartilhados por todos os *tenants*) interligados a apenas uma instância da aplicação ao invés de três, como representa a Figura 2.1. Estas componentes são divididas para proporcionar uma plataforma escalável e flexível [24]. Apesar de existirem vários benefícios em utilizar este tipo de plataforma, também se verificam, naturalmente, algumas desvantagens, e a Cybersecurity Cloud não é exceção. Assim, deve ter-se em conta que, para utilizar a plataforma *Multi-Tenant* SaaS, é necessário isolar e gerir os dados de forma eficiente [9], para evitar inconsistências nos dados de cada *tenant*.

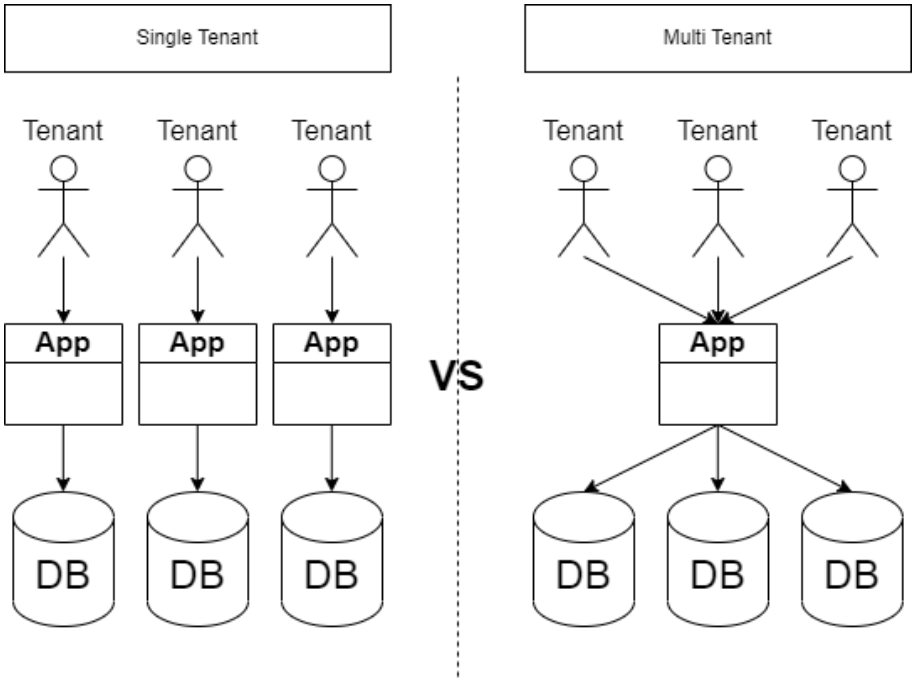

Figura 2.1: *Single-Tenant vs Multi-Tenant*

2.4.2 Arquitetura REST

A arquitetura REST *(Representational State Transfer)* é cada vez mais utilizada em serviços *web*, devido à sua prevalência em relação ao RPC *(Remote Procedure Call)* [39]. Esta arquitetura permite que o servidor seja abstraído através da utilização de recursos variados; assim, o cliente pode interagir com o servidor de forma *stateless* [17], através de pedidos a uma interface. Na arquitetura REST, o HTTP *(Hypertext Transfer Protocol)* passa a ser um protocolo de transferência de estado, ao invés de apenas um protocolo de transporte de dados [19]. As operações GET, POST, PUT e DELETE são utilizadas para realizar pedidos e obter respostas REST: o GET obtém o estado atual de um recurso; o POST transfere um novo estado para um recurso; o PUT cria um novo recurso e o DELETE remove um recurso existente [30]. Com esta arquitetura, a Cybersecurity Cloud consegue separar a camada de interface (desenvolvida em Angular), a camada de negócio (desenvolvida em Golang) e a camada de dados (usando MariaDB).

2.4.3 Linguagem Golang

A linguagem Go, também conhecida como Golang, é a linguagem utilizada no desenvolvimento de *backend* na Cybersecurity Cloud. Foi desenvolvida pela Google em 2007 como uma linguagem de *backend* para criar *software confiável*, eficiente e de compilação rápida. Sendo um projeto *open-source*, o número de colaboradores a trabalhar no mesmo aumentou rapidamente, acelerando o desenvolvimento e utilização desta linguagem [7]. Go é *Statically-typed*, ou seja, o tipo das variáveis é verificado em tempo de compilação e não em tempo de execução; eficiente na recolha de lixo e suporta concorrência. Para além disto, é bastante intuitiva, permite formatar o código automaticamente e detetar erros em tempo de compilação. Todas estas características fazem com que seja fácil programar uma aplicação em Go por vários programadores em simultâneo [14].

2.4.4 *Clean-Architecture*

A *Clean-Architecture* é um modelo de organização e estruturação do código de uma aplicação, que possibilita a separação em camadas do *software* [18]. Essencialmente, esta arquitetura visa simplificar o desenvolvimento, manutenção, testes e integração da aplicação.

Atualmente, está a decorrer um processo de migração completa para esta arquitetura na Cybersecurity Cloud, para melhorar a eficiência e estrutura da plataforma. A arquitetura *Clean-Architecture* separa os diferentes acessos da aplicação em pacotes diferentes. Os acessos à base de dados são realizados através de repositórios, que são expostos a outras camadas através de interfaces específicas e onde os dados são geridos por meio de entidades. Os acessos aos repositórios são, por sua vez, realizados através de serviços (contêm a lógica de negócio), que também são expostos por meio de interfaces específicas e onde os dados são geridos através de *viewmodels*. Os acessos aos serviços são realizados através de *handlers*, que recebem e enviam respostas a pedidos HTTP feitos à aplicação, expostos através de uma arquitetura REST. Ao separar todas essas camadas em pacotes de acordo com a lógica de negócio, é possível obter um sistema coeso, limpo e de fácil gestão, como é exemplificado na Figura 2.2.

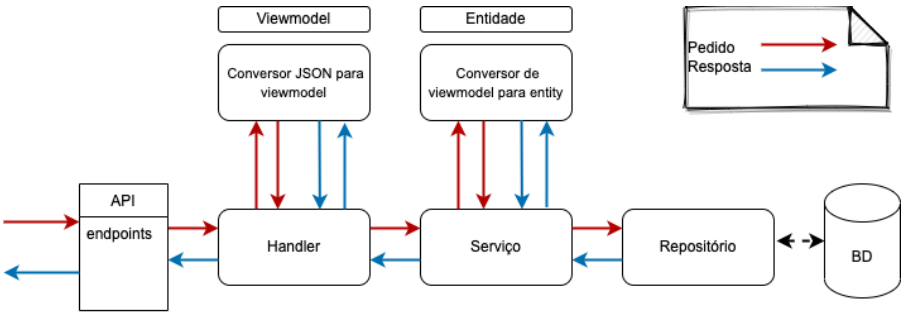

Figura 2.2: Exemplo de Pedido e Resposta *Clean Arquitecture*

2.4.5 Base de Dados MariaDB

MySQL é um sistema de gestão de dados relacional, crucial no desenvolvimento de inúmeras aplicações da atualidade [15] e utilizado na Cybersecurity Cloud. Com a venda deste sistema para a Oracle, começaram a surgir algumas críticas de utilizadores, nomeadamente uma insatisfação com a qualidade em geral e com a demora nos processos de melhoria. Como resposta a esta insatisfação, foi criado o MariaDB, um *fork* do código-fonte do MySQL; projetado para facilitar a vida dos programadores, incluindo várias melhorias como o aumento do desempenho e testabilidade, facilidade de uso e diminuição de *bugs* e alertas [8].

2.4.6 Autenticação por Token JWT

A autenticação JWT (JSON Web Token) é um método de autenticação cada vez mais utilizado em aplicações *web*, usado também na Cybersecurity Cloud. Este método envolve o uso de tokens assinados de forma a verificar a identidade do utilizador. O *token* é um objeto JSON que contém informações sobre o utilizador, além de um código de autenticação assinado que é utilizado para verificar a autenticidade do utilizador. Este *token* é gerado pelo servidor quando o utilizador faz *login* e é enviado para o cliente, para ser armazenado na *cache* ou no cabeçalho HTTP. A cada pedido subsequente ao servidor, o *token* é enviado de volta para verificar a autenticação e autenticidade do mesmo. Uma grande vantagem no uso destes *tokens* é que contêm uma data de expiração que ajuda a manter a segurança quanto ao forjamento e reutilização por utilizadores maliciosos.

2.5 Metodologias de Desenvolvimento de *Software*

Todos os projetos de *software* seguem um sistema metodológico de forma a chegarem a um estado final, garantindo qualidade, controlo e produtividade ao longo do processo. Atualmente, existem várias metodologias que permitem chegar a este estado, sendo que se subdividem em duas grandes categorias: metodologias tradicionais e metodologias ágeis [27].

2.5.1 Metodologia Tradicional

As metodologias tradicionais seguem um modelo sequencial e linear, no qual cada fase tem, obrigatoriamente, que estar concluída para que seja possível iniciar a próxima. É o caso dos modelos Waterfall, V-Model, Incremental, Spiral e Rational Unified Process [36]. Nos modelos supramencionados, as fases estão bem definidas:

- Levantamento de requisitos;
- Análise dos requisitos: arquitetura, desenho e tecnologia a serem utilizados;
- Projeto: criação de um plano detalhado para a implementação;
- Implementação: desenvolvimento do *software*;

- Testes: realização de testes para garantir que a implementação vai de encontro aos requisitos levantados;
- Setup: instalação e configuração do software em ambiente de produção;
- Manutenção: realização de pequenas correções de *bugs* para o correto funcionamento do *software*.

2.5.2 Metodologias Agile

As metodologias ágeis são metodologias flexíveis e iterativas que, no seu conjunto, permitem obter resultados incrementais rapidamente. Esta rapidez nos resultados deve-se, sobretudo, ao facto de todas as fases serem revistas no mesmo ciclo de trabalho - que, geralmente, tem iterações de duas a quatro semanas. Entre os exemplos mais conhecidos, destaca-se o Scrum, Kanban, Extreme Programming, Lean Software Development e Crystal [36]. Cada uma destas metodologias tem as suas próprias práticas e abordagens, no entanto, todas seguem o Manifesto Agile, que tem como principais valores e princípios:

- Valorização de entreajuda na equipa;
- Desenvolvimento rápido e iterativo;
- Colaboração com o cliente e capacidade de fazer alterações rapidamente;
- Reflexões, planeamentos e ajustes periódicos.

Em face disto, é possível chegar à conclusão que as metodologias ágeis têm a capacidade de entregar *software* com baixos custos, rapidez e versatilidade de requisitos, o que faz com que seja a metodologia de eleição na Cybersecurity Cloud.

2.6 Near Field Communication e Tags NFC

O NFC (*Near Field Communication*) é uma tecnologia de comunicação sem fios, de curto alcance, que através de uma simples aproximação permite a troca de dados entre dispositivos compatíveis com NFC. Esta tecnologia é amplamente utilizada em dispositivos móveis como *smartphones* e *tablets*.

As *tags* NFC são pequenas etiquetas que contêm um *micro-chip* com uma antena, que permite a comunicação por NFC. As *tags* podem ser programadas com informações, como um URL, um número de telefone ou um simples texto, e podem ser lidas por dispositivos móveis com NFC. As *tags* NFC são geralmente utilizadas para ativar ações específicas no dispositivo móvel, como abrir uma página *web* ou uma aplicação, ou configurar automaticamente as definições do dispositivo [33].

2.7 Treinos Realizados pela Cybersecurity Cloud

Atualmente, a Cybersecurity Cloud oferece aos seus clientes vários serviços de teste e treino de colaboradores, entre os quais se destacam os módulos de *Phishing* e *eLearning* como os mais bem sucedidos. Neste sentido, e para uma melhor compreensão das atividades realizadas na EMVENCI a nível de treinos, importa esclarecer alguns conceitos:

- Campanha: simulacro de *Phishing* direcionado a um grupo específico;
- *Templates*: vários formatos de *e-mails* ou SMS que podem ser selecionados para envio numa campanha;
- Questionário: questões apresentadas com um formato de resposta de escolha múltipla, apresentadas quando a vítima falha no simulacro de *Phishing* para testar o seu conhecimento e capacidades de deteção destes ataques;
- *Report*: documento CSV que contém os resultados das campanhas com dados analisáveis, como o número de pessoas alvo e o número de pessoas que falharam no simulacro de *Phishing*;
- Conteúdo de consciencialização: vídeos educativos sobre vários temas e conceitos relacionados com o *Phishing*;
- Landing Page: página HTML que tem formulários e informações específicas para inserir dados;
- Utilizador: cliente da Cybersecurity Cloud que consegue criar e gerir simulacros;
- Alvo: pessoa ou dispositivo adicionados pelo utilizador que recebem um simulacro como vitimas.

Na Cybersecurity Cloud, o módulo de *Phishing* permite a realização de simulacros, que incluem o envio de *e-mails* e SMS (inofensivos) a um grupo específico de indivíduos. É possível selecionar o modelo que se deseja utilizar, definir os destinatários, adicionar *links* e até incluir questionários no final da campanha. Desta forma, os clientes da plataforma podem testar os funcionários e avaliar a suscetibilidade e consciencialização de cada colaborador relativamente ao *Phishing*. O módulo de *eLearning* permite o envio de vídeos educativos aos clientes e a realização de questionários após a visualização do conteúdo, a fim de testar o conhecimento adquirido.

Capítulo 3

Contextualização com os Simulacros

A presente secção está omissa devido à confidencialidade inerente ao projeto.

Capítulo 4

Análise dos Requisitos

A presente secção está omissa devido à confidencialidade inerente ao projeto.

Capítulo 5

Desenho da Solução

A presente secção está omissa devido à confidencialidade inerente ao projeto.

Capítulo 6

Implementação

A presente secção está omissa devido à confidencialidade inerente ao projeto.

6.1 Testes

Realizar testes é um procedimento essencial no desenvolvimento de *software*, com o objetivo de avaliar o desempenho da solução proposta e averiguar se esta atende aos requisitos estabelecidos anteriormente. Isso também permite identificar áreas de aperfeiçoamento para assegurar maior qualidade no produto final. Os testes *end-to-end* são um tipo de teste de *software* que visa garantir o correto funcionamento do sistema. Estes testes simulam a experiência de um utilizador final, interagindo com o máximo de camadas do sistema possíveis para identificar possíveis problemas. É fundamental realizar este tipo de testes para garantir a qualidade do software.

Na Tabela 6.1 estão representados os diversos testes que têm como objetivo testar a implementação dos Dynamic Groups. Esta tabela tem o identificador do teste, a descrição e o seu objetivo.

Tabela 6.1: Testes end-to-end *Dynamic Groups*

ID	Descrição	Objetivo
EE01	Filtrar um grupo pelo seu impacto de segurança.	Na criação de um grupo escolher um determinado valor para o filtro de impacto de segurança e retornar as pessoas que tenham esse valor.
EE02	Filtrar um grupo pela sua probabilidade de risco.	Na criação de um grupo escolher um determinado valor para o filtro de probabilidade de risco e retornar as pessoas que tenham esse valor.
EE03	Filtrar um grupo pelo seu impacto de risco.	Na criação de um grupo escolher um determinado valor para o filtro de impacto de risco e retornar as pessoas que tenham esse valor.
EE04	Filtrar um grupo pelo seu risco.	Na criação de um grupo escolher um determinado valor para o filtro de risco e retornar as pessoas que tenham esse valor.
EE05	Filtrar pessoas que participaram numa determinada campanha de *Phishing*	Na criação de um grupo escolher um determinado valor para o filtro de campanha de *Phishing* e retornar as pessoas que tenham esse valor
EE06	Filtrar pessoas que participaram numa determinada campanha de *eLearning*.	Na criação de um grupo escolher um determinado valor para o filtro de campanha de *eLearning* e retornar as pessoas que tenham esse valor.
EE07	Filtrar pessoas pela sua data de criação.	Na criação de um grupo escolher um determinado valor para o filtro de data de criação e retornar as pessoas que tenham esse valor.
EE08	Filtrar pessoas que abriram o *e-mail* numa campanha.	Na criação de um grupo escolher o valor de aberto para o filtro e retornar as pessoas que tenham esse valor.
EE09	Filtrar pessoas que submeteram dados numa campanha.	Na criação de um grupo escolher o valor de submeteu dados para o filtro e retornar as pessoas que tenham esse valor.
EE10	Filtrar pessoas que clicaram no *link* de uma campanha.	Na criação de um grupo escolher o valor de clicou *link* para o filtro e retornar as pessoas que tenham esse valor.
EE11	Filtrar pessoas que visualizou o conteúdo de consciencialização de uma campanha.	Na criação de um grupo escolher o valor de visualizou o conteúdo para o filtro e retornar as pessoas que tenham esse valor.
EE12	Filtrar pessoas que passaram no questionário de uma campanha.	Na criação de um grupo escolher o valor de passou para o filtro e retornar as pessoas que tenham esse valor.
EE13	Filtrar pessoas que reportaram uma campanha.	Na criação de um grupo escolher o valor de reportou para o filtro e retornar as pessoas que tenham esse valor.

Na Tabela 6.2 estão representados os diversos testes que têm como objetivo testar a implementação das campanhas de USB e NFC. Esta tabela tem o identificador do teste, a descrição e o seu objetivo.

Tabela 6.2: Testes *end-to-end* USB e NFC

ID	Descrição	Objetivo
EE14	Criar uma campanha de USB ou NFC.	Criar uma campanha de USB ou NFC dando todos os valores necessários e conseguir visualizar a campanha como ”Criada”.
EE15	Alterar valores de uma campanha de USB ou NFC.	Tendo criado já uma campanha, aceder à mesma e alterar qualquer um dos seus valores e guardar essa alteração.
EE16	Eliminar uma campanha de USB ou NFC.	Tendo criado já uma campanha eliminar a mesma e todos os dispositivos relacionados com ela.
EE17	Fazer o *launch* de uma campanha de USB ou NFC.	Tendo criado uma campanha, ser possível fazer o *launch* da mesma.
EE18	Visualizar o conteúdo de consciencialização de uma campanha de USB ou NFC.	Abrir o link respetivo a qualquer dispositivo e visualizar o conteúdo de consciencialização.
EE19	Responder ao questionário de uma campanha de USB ou NFC.	Abrir o *link* respetivo a qualquer dispositivo e responder ao questionário.
EE20	Fazer o download do link de um dispositivo de uma campanha de USB ou NFC.	Tendo feito o ”Launch”da campanha, ser possível fazer o *download* do ficheiro ou *link* relativo a qualquer dispositivo.
EE21	Visualizar informações de eventos relativos a dispositivos numa campanha de USB ou NFC.	Estando a campanha em progresso ser possível verificar quantos alvos abriram o *link*.

Todos estes testes foram baseados nos requisitos identificados no Capitulo **??**. Ao combinar a primeira análise com os testes realizados neste capítulo, é possível obter uma versão consistente, correta e funcional tanto para a melhoria dos *Dynamic Groups* quanto para a nova funcionalidade das campanhas de USB e NFC. Assim com esta proposta de testes é possível que a solução seja validada pela equipa de *testers* da empresa de forma a prosseguir com uma versão de produção.

Capítulo 7

Conclusão

Este capítulo apresenta uma visão geral do projeto realizado na EMVENCI, incluindo um resumo do trabalho desenvolvido e das decisões importantes tomadas ao longo do processo. Além disso, são apresentadas ideias para trabalhos futuros que surgiram durante o estágio e também no final do projeto.

7.1 Trabalho Realizado

O presente trabalho foi desenvolvido como parte integrante do projeto de conclusão do Mestrado em Engenharia Informática e focou-se na análise e otimização do *software* da EMVENCI, em particular na área de simulacros de *Phishing*. Após uma análise das funcionalidades e módulos atuais em vigor na empresa, identificou-se a implementação de *Dynamic Groups* como uma estratégia para melhorar a formação de colaboradores em simulacros, pois agora é possível fazer uma análise mais precisa e aplicar as conclusões retiradas de uma campanha prévia para a seguinte. Considerando o planeamento previamente estabelecido, pode-se afirmar que a implementação alcançou o sucesso esperado de acordo com as funcionalidades e testes esperados.

As campanhas de USB e NFC são focadas na realização de simulacros de *Phishing* através de *pen drives* e *tags* de NFC. Primeiramente, procedeu-se à análise e identificação das necessidades de evolução para o mundo físico, com o intuito de ampliar o leque de testes disponíveis para os colaboradores. Uma vez validadas essas necessidades e requisitos, procedeu-se à implementação de soluções para ambos os tipos de campanhas.

No caso das campanhas de USB, agora é possível inserir numa *pen drive* um ficheiro correspondente a uma campanha, permitindo o teste e treino dos colaboradores. Em relação às campanhas de NFC, implementou-se um sistema que permite o acesso a todas as configurações e informações de uma campanha, e ainda, com o apoio de uma aplicação móvel desenvolvida por outro colega, uma solução para escrever e ler de tags NFC.

Para garantir a eficácia das soluções, desenhei um conjunto de testes específicos para cada uma delas, de forma a auxiliar a equipa de *testers* da empresa a validar as implementações. Estas duas soluções enfatizaram a colaboração, uma vez que outros membros da equipa contribuíram com o desenho gráfico, que será posteriormente implementado pelo *frontend*. Além disso, serão incorpo-

radas as funcionalidades associadas às campanhas de USB e NFC na aplicação móvel, e ambas as soluções serão submetidas a testes pela equipa de *testers* da empresa. O principal propósito desta solução é proporcionar uma versão segura, funcional e eficaz de campanhas de *Phishing* utilizando dispositivos USB e NFC, concluindo-se que esse objetivo foi plenamente alcançado.

Este projeto proporcionou uma valiosa oportunidade de aquisição de novos conhecimentos sobre segurança informática, metodologias de trabalho, tecnologias usadas, bem como sobre trabalho em equipa e integração na empresa.

7.2 Trabalho Futuro

O futuro desenvolvimento da solução para as campanhas de USB e NFC passa por três pontos principais:

Possibilitar a exportação automática do ficheiro HTML para várias pen drives simultaneamente, por meio de uma *dock station*. Com isso, o utilizador não precisa fazer o *download* de cada ficheiro e fazer *upload* em cada *pen drive*, poupando tempo e trabalho.

Permitir a implementação de um *autorun* nas pen drives de forma a registar o evento de um alvo abrir uma *pen drive* não fidedigna no computador. Por exemplo, seria possível adicionar um novo evento, como ”PEN OPENED”, para obter um registo ainda mais detalhado das ações dos utilizadores.

Até ao momento, a empresa não armazena dados dos utilizadores para criar perfis dos alvos. No entanto, é possível tentar criar perfis dos alvos nas campanhas de USB ou NFC, juntando os IPs públicos e privados ou armazenando dados inseridos, exceto as senhas. Desta forma, seria possível realizar uma análise posterior mais aprofundada nas campanhas.

Bibliografia

[1] Norah Saud Al-Musib, Faeiz Mohammad Al-Serhani, Mamoona Humayun, and NZ Jhanjhi. Business email compromise (bec) attacks. *Materials Today: Proceedings*, 2021.

[2] Rana Alabdan. Phishing attacks survey: types, vectors, and technical approaches. *Future Internet*, 12(10):168, 2020.

[3] Zainab Alkhalil, Chaminda Hewage, Liqaa Nawaf, and Imtiaz Khan. Phishing attacks: A recent comprehensive study and a new anatomy. *Frontiers in Computer Science*, 3:563060, 2021.

[4] Abdullah Alnajim and Malcolm Munro. An anti-phishing approach that uses training intervention for phishing websites detection. In *2009 Sixth International Conference on Information Technology: New Generations*, pages 405–410. IEEE, 2009.

[5] Mohamed Alsharnouby, Furkan Alaca, and Sonia Chiasson. Why phishing still works: User strategies for combating phishing attacks. *International Journal of Human-Computer Studies*, 82:69–82, 2015.

[6] Anti-Phishing Working Group APWG. *Phishing Activity Trends Report 1st Quarter 2022*. 2022.

[7] Ivo Balbaert. *The way to Go: A thorough introduction to the Go programming language*. IUniverse, 2012.

[8] Daniel Bartholomew. Mariadb vs. mysql. *Dostopano*, 7(10):2014, 2012.

[9] Cor-Paul Bezemer and Andy Zaidman. Multi-tenant saas applications: maintenance dream or nightmare? In *Proceedings of the joint ercim workshop on software evolution (evol) and international workshop on principles of software evolution (iwpse)*, pages 88–92, 2010.

[10] Vaishnavi Bhavsar, Aditya Kadlak, and Shabnam Sharma. Study on phishing attacks. *Int. J. Comput. Appl*, 182:27–29, 2018.

[11] Junaid Ahsenali Chaudhry and Robert G Rittenhouse. Phishing: Classification and countermeasures. In *2015 7th International Conference on Multimedia, Computer Graphics and Broadcasting (MulGraB)*, pages 28–31. IEEE, 2015.

[12] Kang Leng Chiew, Kelvin Sheng Chek Yong, and Choon Lin Tan. A survey of phishing attacks: Their types, vectors and technical approaches. *Expert Systems with Applications*, 106:1–20, 2018.

[13] Cassandra Cross and Rosalie Gillett. Exploiting trust for financial gain: An overview of business email compromise (bec) fraud. *Journal of Financial Crime*, 27(3):871–884, 2020.

[14] Alan AA Donovan and Brian W Kernighan. *The Go programming language*. Addison-Wesley Professional, 2015.

[15] Paul DuBois. *MySQL*. Pearson Education, 2008.

[16] David P Farrington and Barry J Knight. Stealing from a"lost"letter: Effects of victim characteristics. *Criminal Justice and Behavior*, 7(4):423–436, 1980.

[17] Xinyang Feng, Jianjing Shen, and Ying Fan. Rest: An alternative to rpc for web services architecture. In *2009 First International Conference on future information networks*, pages 7–10. IEEE, 2009.

[18] Vinicius Barros Silva Ferreira, Carlos Antônio Ferreira, and Eliana Tiba Gomes Grande. Estado da arte da pesquisa em: Clean architecture e princípios de solid. *Research, Society and Development*, 11(16):e335111637198–e335111637198, 2022.

[19] David Gourley, Brian Totty, Marjorie Sayer, Anshu Aggarwal, and Sailu Reddy. *HTTP: the definitive guide*. "O'Reilly Media, Inc.", 2002.

[20] Jason Hong. The state of phishing attacks. *Communications of the ACM*, 55(1):74–81, 2012.

[21] Markus Jakobsson and Steven Myers. *Phishing and countermeasures: understanding the increasing problem of electronic identity theft*. John Wiley & Sons, 2006.

[22] Hanjae Jeong, Younsung Choi, Woongryel Jeon, Fei Yang, Yunho Lee, Seungjoo Kim, and Dongho Won. Vulnerability analysis of secure usb flash drives. In *2007 IEEE International Workshop on Memory Technology, Design and Testing*, pages 61–64. IEEE, 2007.

[23] P Kalaharsha and Babu M Mehtre. Detecting phishing sites–an overview. *arXiv preprint arXiv:2103.12739*, 2021.

[24] Sungjoo Kang, Sungwon Kang, and Sungjin Hur. A design of the conceptual architecture for a multitenant saas application platform. In *2011 First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering*, pages 462–467. IEEE, 2011.

[25] Ponnurangam Kumaraguru, Justin Cranshaw, Alessandro Acquisti, Lorrie Cranor, Jason Hong, Mary Ann Blair, and Theodore Pham. School of phish: a real-world evaluation of anti-phishing training. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, pages 1–12, 2009.

[26] Elmer Evert Hendrik Lastdrager. From fishing to phishing. 2018.

[27] Ahmed Lawal and Richard Chukwu Ogbu. A comparative analysis of agile and waterfall software development methodologies. *Bakolori Journal of General Studies*, 11(2):1–2, 2021.

[28] Gerald Madlmayr, Josef Langer, Christian Kantner, and Josef Scharinger. Nfc devices: Security and privacy. In *2008 Third International Conference on Availability, Reliability and Security*, pages 642–647. IEEE, 2008.

[29] Mike McCormick. Waterfall vs. agile methodology. *MPCS, N/A*, 3, 2012.

[30] M Melnichuk, Yu Kornienko, and O Boytsova. Web-service. restful architecture. *Automation of technological and business processes*, 10(1), 2018.

[31] Curtis B Merritt and Richard G Fowler. The pecuniary honesty of the public at large. *The Journal of Abnormal and Social Psychology*, 43(1):90, 1948.

[32] Swapan Purkait. Phishing counter measures and their effectiveness–literature review. *Information Management & Computer Security*, 2012.

[33] Anusha Rahul, Sethuraman Rao, ME Raghu, et al. Near field communication (nfc) technology: a survey. *International Journal on Cybernetics & Informatics (IJCI)*, 4(2):133, 2015.

[34] Koceilah Rekouche. Early phishing. *arXiv preprint arXiv:1106.4692*, 2011.

[35] Ashina Sadiq, Muhammad Anwar, Rizwan A Butt, Farhan Masud, Muhammad K Shahzad, Shahid Naseem, and Muhammad Younas. A review of phishing attacks and countermeasures for internet of things-based smart business applications in industry 4.0. *Human behavior and emerging technologies*, 3(5):854–864, 2021.

[36] Soobia Saeed, NZ Jhanjhi, Mehmood Naqvi, and Mamoona Humayun. Analysis of software development methodologies. *International Journal of Computing and Digital Systems*, 8(5):446–460, 2019.

[37] Steve Sheng, Ponnurangam Kumaraguru, Alessandro Acquisti, Lorrie Cranor, and Jason Hong. Improving phishing countermeasures: An analysis of expert interviews. In *2009 eCrime Researchers Summit*, pages 1–15. IEEE, 2009.

[38] Manmeet Mahinderjit Singh, KAAK Adzman, and Rohail Hassan. Near field communication (nfc) technology security vulnerabilities and countermeasures. *International Journal of Engineering & Technology*, 7(4.31):298–305, 2018.

[39] Beng Hang Tay and Akkihebbal L Ananda. A survey of remote procedure calls. *ACM SIGOPS Operating Systems Review*, 24(3):68–79, 1990.

[40] Timothy Vidas, Emmanuel Owusu, Shuai Wang, Cheng Zeng, Lorrie Faith Cranor, and Nicolas Christin. Qrishing: The susceptibility of smartphone users to qr code phishing attacks. In *International Conference on Financial Cryptography and Data Security*, pages 52–69. Springer, 2013.

[41] Zikai Alex Wen, Zhiqiu Lin, Rowena Chen, and Erik Andersen. What. hack: engaging anti-phishing training through a role-playing phishing simulation game. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

[42] Ezer Osei Yeboah-Boateng and Priscilla Mateko Amanor. Phishing, smishing & vishing: an assessment of threats against mobile devices. *Journal of Emerging Trends in Computing and Information Sciences*, 5(4):297–307, 2014.

ANEXO .1


```
{
   "test":false,
   "name":"Campanha NFC Teste",
   "type":"NFC",
   "smtp_id":0,
   "schedule_complete":"2023-10-10T13:16:00.000Z",
   "awareness_page_toggle":true,
   "from_address2":"empresanow.com",
   "page":{
      "id":9,
      "name":"Ebay - Verify Your Account",
      "type":"public"
   },
   "questionnaire_id":1,
   "questionnaire_restriction_type":"public",
   "questionnaire_passing_score":51,
   "awareness_page":{
      "name":"",
      "type":"private",
      "awareness_content":{
         "id":64,
         "type":"public"
      },
      "skip_entry":true,
      "entry_text":"",
      "button_text1":"Start",
      "button_text2":"In Progress",
      "button_text3":"Read and Understood",
      "button_text4":"Exit",
      "disclaimer":""
   },
   "target_devices":[
      {
         "name":"teste1",
         "description":"teste",
         "device_type":"nfc_tag"
      },
      {
         "name":"teste2",
         "description":"teste2",
         "device_type":"nfc_tag"
      }
   ]
}
```