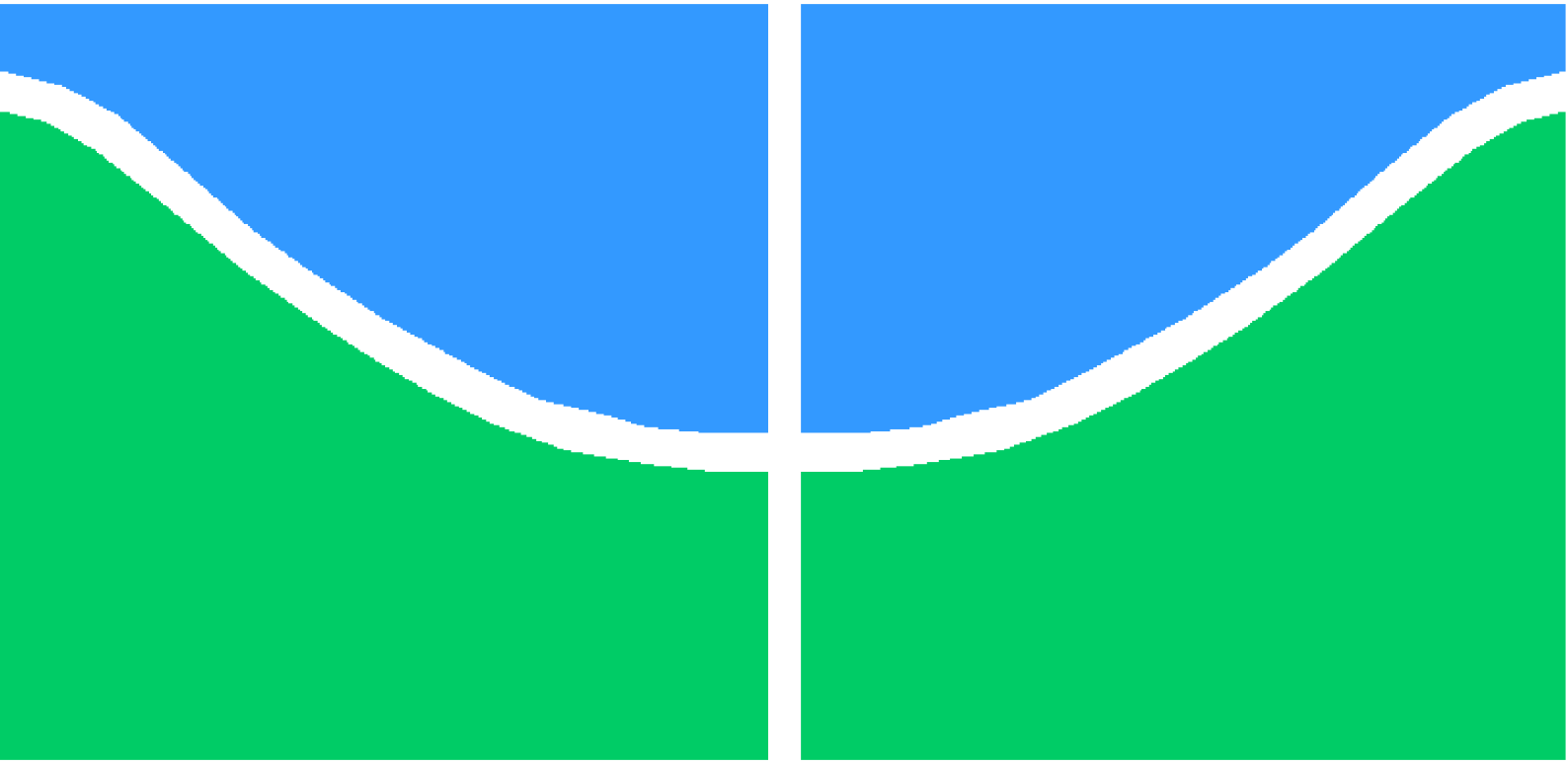


Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Engenharia de Software

Segurança no Desenvolvimento Web: Análise de casos e um guia para mitigar exposições de dados sensíveis

Autor: Álvaro Leles Guimarães

Orientadora: Profª. Drª. Elaine Venson

Brasília, DF

2023

Álvaro Leles Guimarães

Segurança no Desenvolvimento Web: Análise de casos e um guia para mitigar exposições de dados sensíveis

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Profª. Drª. Elaine Venson

Brasília, DF

2023

Álvaro Leles Guimarães

Segurança no Desenvolvimento Web: Análise de casos e um guia para mitigar exposições de dados sensíveis

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Brasília, DF
2023

Agradecimentos

Primeiramente, gostaria de agradecer a minha orientadora, Prof[a]. Dr[a] Elaine Venson por suas orientações, paciência e conhecimento. Sua vontade em compartilhar seu conhecimento foi crucial para o desenvolvimento deste trabalho. Suas sugestões e feedbacks construtivos me ajudaram a seguir e melhorar com minha pesquisa, aprofundando minha compreensão sobre o tema.

Em seguida, agradeço aos meus colegas de curso e amigos que me apoiaram ao longo deste processo. Dividir a experiência do curso com eles, além de bons momentos, foram essenciais para manter minha motivação quando enfrentei desafios.

Além disso, agradeço à minha família pelo amor incondicional, apoio constante e compreensão durante minha jornada acadêmica. Seu incentivo e apoio foram fundamentais para que eu pudesse me dedicar aos estudos e alcançar meus objetivos. Agradeço também a minha namorada, por todo apoio, compreensão, paciência e conselhos ao longo deste tempo.

"Diga-me e eu esquecerei. Ensina-me e eu poderei lembrar. Envolva-me e eu aprenderei."
(Benjamin Franklin)

Resumo

O grande volume de informações que trafegam em sistemas de software faz com que a segurança seja um desafio para a área de Engenharia de Software. Parte dessas informações são classificadas como dados sensíveis, demandando que mecanismos de segurança estejam presentes para evitar a sua exposição. As aplicações web, em especial, são impactadas ainda por um ambiente de concorrência e necessidade de rapidez no lançamento, muitas vezes levando ao neglicenciamento da segurança. Estas aplicações envolvem uma interface de interação com o usuário final, em uma relação de comunicação cliente e servidor, onde informações trafegam a todo momento. Contudo, informações sensíveis não devem ser vistas ou manipuladas por um usuário final ou agente malicioso. Informações sensíveis expostas evidenciam uma das vulnerabilidades mais comuns na área de desenvolvimento web, a exposição de dados sensíveis, causadas principalmente por falhas de criptografia. O objetivo deste trabalho é propor um guia de desenvolvimento de aplicações web, focado no combate à exposição de dados sensíveis. A metodologia empregada será descritiva, a partir da revisão documental, com a investigação de casos reais de exposição de dados no mercado mundial, além da coleta de ações de mitigação e aprendizados adquiridos.

Palavras-chave: Segurança; Desenvolvimento Web; Exposição de Dados Sensíveis; Falhas de Criptografia.

Abstract

The increasing volume of software systems that handle information poses a significant security challenge for the field of Software Engineering. The sensitivity of this information necessitates the development of robust security measures to prevent data exposure. Among these software systems are web applications, which are particularly affected by a competitive environment and the need for rapid deployment, often leading to security negligence. These applications involve a user-facing interface, establishing a client-server communication relationship where information is constantly exchanged. However, sensitive information should not be accessible or manipulated by end users or malicious agents. The exposure of sensitive information highlights one of the most common vulnerabilities in web development, namely, sensitive data exposure caused mainly by cryptographic failures. This work aims to propose a guide for the development of web applications, focusing on combating sensitive data exposure. The methodology employed is descriptive, involving a documentary review, investigation of real-world cases of data exposure in the global market, and the collection of mitigation actions and lessons learned.

Keywords: Security; Web Development; Sensitive Data Exposure; Cryptographic Failures

Lista de ilustrações

Figura 1 – Informações pessoais identificáveis. 23
Figura 2 – Microsoft SDL. 28
Figura 3 – Exemplo de dados vazados. 37
Figura 4 – Linha do tempo dos ataques ao Yahoo. 40

Lista de tabelas

Tabela 1 – Objetivos e métodos 31
Tabela 2 – Consolidação dos problemas 62

Lista de abreviaturas e siglas

API	Application Programming Interface
CVE	Common Vulnerabilities and Exposures
CWE	Common Weakness Enumeration
DKIM	DomainKeys Identified Mail
DLL	Dynamic-link library
DMARC	Domain-based Message Authentication, Reporting & Conformance
DNS	Domain Name System
HTTP	Hypertext Transfer Protocol
OTP	One-Time Password
OWASP	The Open Worldwide Application Security Project
PII	Personally Identifiable Information
SDL	Security Development Lifecycle
SERPRO	Serviço Federal de Processamento de Dados
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SPF	Sender Policy Framework
TCC	Trabalho de Conclusão de Curso
UnB	Universidade de Brasília

Sumário

1 INTRODUÇÃO 15
1.1 Contexto 15
1.2 Problema 16
1.3 Objetivos 17
1.3.1 Objetivo Geral 17
1.3.2 Objetivos Específicos 17
1.4 Metodologia 17
1.5 Organização do Trabalho 18

2 REFERENCIAL TEÓRICO 19
2.1 Considerações Iniciais do Capítulo 19
2.2 Segurança de Software 19
2.2.1 Falha de segurança 19
2.2.2 Vulnerabilidades em APIs 20
2.2.3 Vulnerabilidades de vazamento de informações 21
2.2.4 CVE 21
2.2.5 CWE 21
2.3 Desenvolvimento Web 21
2.3.1 Segurança no Desenvolvimento Web 22
2.3.2 Dados sensíveis 23
2.3.3 Exposição de dados sensíveis 24
2.4 OWASP 24
2.4.1 OWASP Top Ten 25
2.5 Criptografia 25
2.6 Práticas de Desenvolvimento Seguro 26
2.6.1 SDL 26
2.6.2 Prevenções para exposição de dados sensíveis 28

3 METODOLOGIA 30
3.1 Considerações Iniciais do Capítulo 30
3.2 Descrever e detalhar o que são Exposição de Dados Sensíveis e Falhas de Criptografia 31
3.3 Pesquisar e relatar detalhadamente casos reais envolvendo este tipo de vulnerabilidade, incluindo causas raiz, impactos e consequências dos incidentes 32

3.4 Identificar possíveis medidas que poderiam ter evitado a exposição dos dados nos casos relatados . . . 33
3.5 Relacionar os casos levantados com o material teórico estudado . . . 33
3.6 Propor um guia prático de desenvolvimento seguro focado nas vulnerabilidades especificadas, com base no que foi encontrado nos casos analisados . . . 34

4 ANÁLISE DE CASOS . . . 35
4.1 Considerações Iniciais do Capítulo . . . 35
4.2 LinkedIn . . . 35
4.2.1 Causas raiz . . . 35
4.2.2 Métodos de ataque utilizados . . . 36
4.2.3 Tipos de dados expostos . . . 36
4.2.4 Atitudes tomadas pela empresa . . . 37
4.2.5 Consequências . . . 38
4.2.6 Estratégias de mitigação . . . 38
4.3 Yahoo . . . 39
4.3.1 Causas raiz . . . 40
4.3.2 Métodos de ataque utilizados . . . 41
4.3.3 Tipos de dados expostos . . . 41
4.3.4 Atitudes tomadas pela empresa . . . 42
4.3.5 Consequências . . . 42
4.3.6 Estratégias de mitigação . . . 43
4.4 eBay . . . 43
4.4.1 Causas raiz . . . 44
4.4.2 Métodos de ataque utilizados . . . 44
4.4.3 Tipos de dados expostos . . . 44
4.4.4 Atitudes tomadas pela empresa . . . 45
4.4.5 Consequências . . . 45
4.4.6 Estratégias de mitigação . . . 45
4.5 Equifax . . . 46
4.5.1 Causas raiz . . . 46
4.5.2 Métodos de ataque utilizados . . . 46
4.5.3 Tipos de dados expostos . . . 47
4.5.4 Atitudes tomadas pela empresa . . . 47
4.5.5 Consequências . . . 48
4.5.6 Estratégias de mitigação . . . 48
4.6 JPMorgan Chase . . . 48
4.6.1 Causas raiz . . . 49
4.6.2 Métodos de ataque utilizados . . . 49

4.6.3 Tipos de dados expostos . . . 50
4.6.4 Atitudes tomadas pela empresa . . . 50
4.6.5 Consequências . . . 50
4.6.6 Estratégias de mitigação . . . 51
4.7 Target . . . 51
4.7.1 Causas raiz . . . 51
4.7.2 Métodos de ataque utilizados . . . 52
4.7.3 Tipos de dados expostos . . . 53
4.7.4 Atitudes tomadas pela empresa . . . 53
4.7.5 Consequências . . . 54
4.7.6 Estratégias de mitigação . . . 54

5 ANÁLISE COMPARATIVA . . . 55
5.1 Considerações Iniciais do Capítulo . . . 55
5.2 Comparações . . . 55
5.2.1 LinkedIn . . . 55
5.2.1.1 Problemas . . . 55
5.2.1.2 Soluções . . . 55
5.2.1.3 Mitigação baseada no referencial teórico . . . 55
5.2.2 Yahoo . . . 56
5.2.2.1 Problemas . . . 56
5.2.2.2 Soluções . . . 56
5.2.2.3 Mitigação baseada no referencial teórico . . . 57
5.2.3 eBay . . . 57
5.2.3.1 Problemas . . . 57
5.2.3.2 Soluções . . . 57
5.2.3.3 Mitigação baseada no referencial teórico . . . 58
5.2.4 Equifax . . . 58
5.2.4.1 Problemas . . . 58
5.2.4.2 Soluções . . . 58
5.2.4.3 Mitigação baseada no referencial teórico . . . 59
5.2.5 JPMorgan Chase . . . 59
5.2.5.1 Problemas . . . 59
5.2.5.2 Soluções . . . 60
5.2.5.3 Mitigação baseada no referencial teórico . . . 60
5.2.6 Target . . . 60
5.2.6.1 Problemas . . . 60
5.2.6.2 Soluções . . . 61
5.2.6.3 Mitigação baseada no referencial teórico . . . 61
5.2.7 Consolidação . . . 62

5.3 Considerações Finais do Capítulo . . . 63

6 CONCLUSÃO . . . 64

REFERÊNCIAS . . . 66

APÊNDICE A – GUIA DE MEDIDAS DE SEGURANÇA CONTRA AMEAÇAS DE EXPOSIÇÃO DE DADOS SENSÍVEIS . . . 76

A.1 Introdução . . . 76
A.1.1 Segurança no desenvolvimento web . . . 76
A.1.2 Objetivo . . . 76
A.2 Referencial . . . 77
A.3 Criptografia e Categorização de Dados . . . 78
A.3.1 Dados em Repouso . . . 78
A.3.2 Dados em Movimento . . . 78
A.3.3 Dados em Uso . . . 79
A.4 *Malwares* . . . 79
A.4.1 *Antimalware* . . . 80
A.4.2 Detecção e Resposta em *Endpoints* . . . 80
A.4.3 Detecção e Resposta de Rede . . . 81
A.4.4 Isolamento Remoto de Navegador . . . 81
A.4.5 Interface de Ambiente de Trabalho Virtual . . . 82
A.5 Falhas na Administração e Uso de Sistemas . . . 82
A.6 *Phishing* . . . 83
A.6.1 Autenticação de Dois Fatores . . . 83
A.6.2 Chaves de Segurança . . . 84
A.6.3 Tokens OTP Dedicados . . . 85
A.6.3.1 Autenticação de Dois Fatores em Aplicativos Móveis . . . 85
A.6.3.2 OTP Baseado em SMS . . . 86
A.6.3.3 OTP Baseado em E-mail . . . 86
A.6.4 Autenticação de Múltiplos Fatores . . . 86
A.6.5 Proteger os Domínios com SPF, DKIM e DMARC . . . 87
A.6.6 Domínios Parecidos . . . 88
A.6.7 Preenchimento de Credenciais e Controle de Conta . . . 88
A.6.8 Gerenciadores de Senhas . . . 89
A.6.9 Defesas Adicionais . . . 90
A.6.9.1 Treinamento e Testes . . . 90
A.6.9.2 Verificações de Complexidade de Senha . . . 90
A.6.9.3 Rotação de Senhas . . . 91

A.7 Ferramentas de Terceiros . . . 91
A.7.1 Segurança de Fornecedores . . . 92
A.7.2 Desenvolvedores, Parceiros e Clientes . . . 92
A.8 Centralização de Rede . . . 93
A.9 Vulnerabilidades de Software . . . 93
A.9.1 Vulnerabilidades de Primeira Parte . . . 94
A.9.1.1 Fase de Desenvolvimento . . . 94
A.9.1.2 Fase de Testes . . . 95
A.9.1.3 Fase de Produção . . . 96
A.9.2 Vulnerabilidades de Terceiros . . . 96
A.9.2.1 Identificação e Validação . . . 96
A.9.2.2 Priorização . . . 97
A.9.2.3 Atualização de *Endpoints* . . . 97
A.10 Considerações Finais . . . 98

1 Introdução

1.1 Contexto

Nos últimos anos, o mundo tem experimentado um grande aumento no desenvolvimento e lançamento de sistemas em diversas áreas. A cada lançamento, os concorrentes sentem mais depressa a necessidade de desenvolverem sistemas ou aplicações tão relevantes ou superiores. Muitas vezes, essa necessidade faz com que a área de segurança seja negligenciada, o que é um desafio para a área de Engenharia de Software.

Dentre os sistemas de software, o desenvolvimento de aplicações web cresceu rapidamente, dado a fatores como: possibilidade de atualização rápida e fácil; disponibilidade quase instantânea em nível global, e não ter necessidade de instalação nas máquinas dos usuários (AOKI; CARVALHO, 2016). Porém, estas aplicações estão sujeitas a muitas vulnerabilidades de segurança, causadas por um ou mais tipos diferentes de falhas introduzidas ao longo de seu desenvolvimento, como a exposição de dados sensíveis e falhas de criptografia (The OWASP Foundation, 2021b).

Aplicações web precisam trafegar e processar informações entre suas pontas cliente e servidor. Algumas dessas informações são consideradas sensíveis, como senhas, propriedades identificadoras de objetos, ou outras informações necessárias para a interface que está sendo apresentada, mas que não podem ser acessíveis ao usuário final (FAHL, 2021). Caso essas informações sejam acessadas pelo usuário ou alteradas de forma não orgânica, é possível dizer que houve uma exposição de dados sensíveis. A principal causa desse tipo de vulnerabilidade está relacionada às falhas de criptografia (The OWASP Foundation, 2021a). Aplicações web que trafegam dados sensíveis entre cliente e servidor devem ocultar e criptografar essas informações, impossibilitando que possam ser compreendidas, alteradas ou enviadas para o servidor por um usuário comum ou agente malicioso.

Com o crescimento e globalização de aplicações web, mais usuários e mais dados sensíveis são inseridos e trafegados pelos sistemas, tornando imprescindível a utilização de mecanismos de segurança para preservar esse tipo de dado. Caso vulnerabilidades presentes no desenvolvimento inicial não sejam removidas, com a ocorrência de um ataque, o prejuízo pode ser expressivo. Segundo o relatório de custo de violação de dados da IBM (IBM, 2023), um vazamento de dados nos EUA custa em média para as empresas 9,44 milhões de dólares. O prejuízo causado por uma exposição de dados envolve desde a perda de confiança por parte dos clientes, e consequentemente a perda dos mesmos, até o esforço e o investimento necessários para remodelar a aplicação com objetivo de construir um sistema mais preparado a resistir a ataques desse tipo.

O relatório também aponta a importância da rápida identificação e mitigação do problema. Um vazamento contido em até 200 dias pode gerar uma economia de até 1,12 milhões de dólares. Quanto antes uma exposição for identificada, mais rápido é possível bloquear a aplicação e diminuir a quantidade de dados expostos, além de diminuir os prejuízos organizacionais como a perda de credibilidade com os usuários (IBM, 2023).

1.2 Problema

Com o aumento exponencial da quantidade de dados gerados e armazenados, é mais complicado para as organizações gerenciar e proteger essas informações adequadamente (DO et al., 2017). Além do aumento na quantidade de dados, há também um aumento na complexidade dos sistemas, que incluem várias camadas de software, infraestrutura e integrações (REINA et al., 2020). Essa complexidade aumenta os desafios na implementação de práticas de segurança eficazes.

Erros humanos também são fatores que contribuem com a ocorrência de vulnerabilidades. Em um ambiente altamente competitivo, organizações muitas vezes sentem a pressão para lançar produtos e serviços rapidamente. Essa necessidade pode levar a adoção de atalhos e negligência nas práticas de segurança.

Essa pressão cai sobre os desenvolvedores, que como todo ser humano, cometem erros. Uma configuração incorreta ou escolha inadequada de configurações de segurança podem resultar na exposição acidental de dados. Os desenvolvedores podem ainda não ter conhecimento suficiente sobre as melhores práticas de segurança ou não receber treinamento adequado (XIAO; WITSCHEY; MURPHY-HILL, 2014). Isso pode gerar falhas na codificação, como vulnerabilidades ou falta de validação adequada de dados de entrada.

Entre 2010 e 2020 foram reportados casos de exposição de dados em empresas como LinkedIn (PEREZ, 2016), Yahoo (FIEGERMAN, 2016), Facebook (CADWALLADR; GRAHAM-HARRISON, 2018) e eBay (COLóN, 2014), e muitos outros podem ser citados.

Dada a quantidade significativa de casos de grandes empresas ou instituições que passaram por esse tipo de exposição, esforços tem sido empreendidos no sentido de tentar mitigar os riscos relacionados a essas vulnerabilidades. Exemplos de iniciativas são a criação de listas de vulnerabilidades e falhas mais frequentes (The MITRE Corporation, 2023b) e a promoção de práticas de desenvolvimento seguro de software que podem ser aplicadas para evitar essas situações (Microsoft, 2023b).

Com esse conjunto de conhecimentos, deve ser possível propor guias que permitam um desenvolvimento que acople segurança de dados desde o início do ciclo de vida de um produto.

Com base nestas constatações, este trabalho tem como base a seguinte pergunta:

- Levando em conta casos reais de vazamento de informações, quais são as falhas de desenvolvimento que levam à exposição de dados sensíveis em aplicações web e que práticas podem preveni-las?

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo deste trabalho é propor um guia de desenvolvimento seguro de aplicações web, focado no combate à exposição de dados sensíveis, com base em análises de casos reais de exposição no cenário mundial.

1.3.2 Objetivos Específicos

Para alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

- Descrever e detalhar o que são Exposição de Dados Sensíveis e Falhas de Criptografia;
- Pesquisar e relatar detalhadamente casos reais envolvendo este tipo de vulnerabilidade, incluindo causas raiz, impactos e consequências dos incidentes;
- Identificar possíveis medidas que poderiam ter evitado a exposição dos dados nos casos coletados;
- Relacionar os casos levantados com o material teórico estudado;
- Propor um guia prático de desenvolvimento seguro focado nas vulnerabilidades especificadas, com base no que foi encontrado nos casos analisados.

1.4 Metodologia

A metodologia definida para o desenvolvimento do projeto tem como objetivo a proposição de um guia de desenvolvimento web, com o intuito de evitar exposição de dados. Esta abordagem é composta por pesquisas bibliográfica e documental. A pesquisa bibliográfica tem como foco a definição do problema, além de outros termos ou conteúdos relevantes para atingir os objetivos. A pesquisa documental busca encontrar casos associados às vulnerabilidades abordadas, com o intuito de realizar uma análise minuciosa dos motivos e resoluções dos mesmos. Este estudo faz uso de coleta e análise de casos, definição da vulnerabilidade escolhida, avaliação dos casos com base na descrição da vulnerabilidade

e proposição de práticas que promovam a prevenção de vulnerabilidades relacionadas à exposição de dados no desenvolvimento web.

1.5 Organização do Trabalho

Este trabalho está organizado da seguinte forma:

- **Capítulo 2 - Referencial Teórico:** Apresenta definições chave relacionadas à segurança de software, desenvolvimento web, comunidade de software, criptografia e práticas de desenvolvimento seguro. Inclui também a definição de falhas, a definição de dados sensíveis e as abordagens de segurança web.
- **Capítulo 3 - Metodologia:** Apresenta a abordagem escolhida pra conduzir o estudo, que inclui pesquisas bibliográfica e documental, além do estabelecimento de passos para coleta e análise de casos.
- **Capítulo 4 - Análise de Casos:** Apresenta os casos reais de exposição de dados de empresas, levantando detalhes como causas raiz, métodos de ataque utilizados, tipos de dados expostos, as atitudes tomadas pela empresa e suas consequências e as estratégias de mitigação específicas para cada caso.
- **Capitulo 5 - Análise Comparativa:** Traz uma analise agrupada dos casos, resumindo seus problemas e soluções em categorias em comum e os relacionando com o referencial teórico levantado. Ao fim, traz uma consolidação dos problemas e os categoriza quanto a fonte do problema.
- **Capítulo 6 - Conclusão:** Nesse Capítulo é feita a conclusão final do trabalho, evidenciando os resultados e objetivos atingidos.

2 Referencial Teórico

2.1 Considerações Iniciais do Capítulo

Neste capítulo, são abordados os assuntos relacionados à segurança de aplicações web e exposições de dados. Primeiro serão abordados pontos referentes à segurança de software como um todo, passando por definições do que é segurança, o que são falhas, tipos de falhas, e esforços da comunidade para mitigação de vulnerabilidades. Em seguida, é abordado o desenvolvimento de aplicações web, abordando definições de segurança neste ambiente específico. Nesta Seção também é abordado o que são dados sensíveis e sua exposição. A Seção 2.4 aborda a fundação OWASP e seu projeto Top Ten, que é o ponto de partida desse trabalho. A Seção 2.5 aborda criptografia, recurso essencial quando se aborda dados sensíveis, com definições iniciais do que é, e como se aplica em sistemas de software. Por fim, há a Seção de práticas de desenvolvimento seguro, com pontos relevantes para garantir segurança de dados.

2.2 Segurança de Software

Um software pode ser considerado seguro se estiver satisfazendo um critério de segurança específico, determinado por requisitos de confidencialidade, integridade e disponibilidade para os dados e funcionalidades do sistema. Entretanto, essas combinações de requisitos podem entrar em conflito entre si, por exemplo, restringir um sistema ao detectar um ataque é bom para a confidencialidade e integridade do sistema, mas ruim para a disponibilidade (PIESSENS, 2021).

2.2.1 Falha de segurança

Uma falha de segurança acontece quando a aplicação não satisfaz seu critério de segurança e uma vulnerabilidade é causada a partir dessa falha de desenvolvimento. Geralmente não é fácil apontar o que causou uma vulnerabilidade, pois não é trivial determinar quais partes do código geram falhas de segurança que consequentemente expõem uma dada vulnerabilidade, apenas sabe-se que violações de segurança são causados por erros de projeto ou desenvolvimento (WILLIAMS, 2021). Outro fato que costuma ocorrer é que uma vulnerabilidade pode precisar de alterações em múltiplos pontos do código, onde em cada um desses pontos, uma estratégia de mitigação diferente pode ser necessária. Cada execução de um software pode satisfazer ou não a especificação de segurança explicada

anteriormente, no momento que há uma execução que não satisfaz essa especificação, há uma falha de segurança (PIESSENS, 2021).

Em geral, muito do trabalho da segurança de software é focado em evitar que falhas e vulnerabilidades conhecidas estejam presentes nas aplicações. Muitas das vulnerabilidades mais conhecidas hoje possuem já boa compreensão de como são causadas e o que pode ser feito para evitá-las (PIESSENS, 2021).

Mesmo com a compreensão das vulnerabilidades e esforço para se aplicar correções em sistemas, desenvolvedores podem erroneamente resolver um problema ou falha de segurança adicionando outros problemas na solução, podendo ocorrer principalmente em situações onde a necessidade para a correção do problema original era de muita urgência e devido a necessidade, foi apressado o processo de desenvolvimento e lançamento ou disponibilidade do código.

Há também a ocorrência da não aplicação de atualizações de segurança, seja por relutância ou não acompanhamento das atualizações que saem para o software em uso. O principal problema é que quando uma vulnerabilidade é reportada, agentes maliciosos trabalham em cima de toda informação disponível sobre a vulnerabilidade, sabendo que muitos usuários não irão realizar a atualização quando estiver disponível (WILLIAMS, 2021).

2.2.2 Vulnerabilidades em APIs

Uma API (*Application Programming Interface*, ou, Interface de Programação de Aplicações) é uma interface de comunicação entre dois ou mais componentes de software, como por exemplo uma aplicação desktop e uma biblioteca de serviços web. Hoje em dia, praticamente todas aplicações fazem uso do consumo de APIs. Toda API possui uma especificação de sua utilização, informando o que precisa ser fornecido a ela e o que ela devolverá para quem a chamou. APIs também possuem critérios de segurança, que quando não são satisfeitos também expõem falhas de segurança, ou vulnerabilidades.

É importante destacar que algumas APIs possuem mais cuidados e detalhes de segurança do que outras. Um exemplo de tipo de API que deve possuir extremo cuidado em termos de segurança são justamente APIs de serviços de segurança, como as que fornecem funcionalidades de criptografia, controle de acesso ou autenticação. Quando se tem aplicações que consomem esse tipo de serviço, essas devem usar as APIs seguindo estritamente as especificações da mesma, pra não permitir que critérios de segurança sejam violados. Cabe aos desenvolvedores dessas APIs escolherem entre flexibilizar o uso da API para facilitar o consumo dos serviços ou restringir suas especificações para garantir que seus critérios de segurança não sejam quebrados (PIESSENS, 2021). Existem pesquisas de

que comumente desenvolvedores cometem erros no consumo de APIs desse tipo, expondo vulnerabilidades (EGELE et al., 2013).

2.2.3 Vulnerabilidades de vazamento de informações

Vulnerabilidades relacionadas ao vazamento de informações são aquelas que violam políticas de confidencialidade no fluxo de informações. Elas podem ser entendidas como a violação de um critério de segurança, porém, quando há esse tipo de vulnerabilidade, pode ter sido necessário que múltiplas execuções e tentativas de roubar informações tenham ocorrido em cima da exploração dessa falha de segurança, obtendo pequenas quantidades de dados de cada vez. Isso torna mais complicado que estratégias de mitigação sejam elaboradas para esse tipo de vulnerabilidade, já que pode ser difícil identificar esses pequenos vazamentos (PIESSENS, 2021).

2.2.4 CVE

A MITRE Corporation é uma organização estadunidense que tem como foco a área de tecnologia de informação e segurança, responsável por manter o sistema do CVE. O CVE (*Common Vulnerabilities and Exposures*) é um banco de dados que armazena vulnerabilidades de segurança registradas pela comunidade. Esse banco permite que constantemente pesquisadores e desenvolvedores puxem referências de especificações de software e hardware relacionados a uma vulnerabilidade, também é possível obter dados de como ela ocorre e suas consequências. Quando disponível, apresenta também recomendações de correção ou mitigação (The MITRE Corporation, 2023a).

2.2.5 CWE

A comunidade atual de software pesquisa e divulga listas e categorias de fraquezas que se tornam úteis para identificação, prevenção e mitigação das mesmas. Fraquezas são pontos sensíveis de um programa que podem gerar vulnerabilidades. Um exemplo dessas listas é a CWE (Common Weakness Enumeration), que busca enumerar quais são as fraquezas que mais tem sido identificadas e relatadas tanto em aplicações de software quanto de hardware. Ela pode ser uma referência para trabalhos de segurança focados em identificação e mitigação de vulnerabilidades (The MITRE Corporation, 2023b).

2.3 Desenvolvimento Web

Atualmente, a web é um dos meios primários com que os usuários interagem com a Internet e aplicações de software. E quanto mais aplicações web se tornam disponíveis, maior a importância da segurança das informações. Essas aplicações (clientes) costumam

interagir com interfaces de aplicativos (servidor) usando tecnologias web, as APIs. Esse fenômeno, conhecido como *webification* afeta o ecossistema web (FAHL, 2021).

2.3.1 Segurança no Desenvolvimento Web

Nos anos 90, o foco de segurança no ambiente web estava no lado do servidor das aplicações e na infraestrutura. As aplicações web basicamente renderizavam sites estáticos em nenhum conteúdo dinâmico. Porém, nos anos 2000, com o surgimento e amadurecimento do uso de linguagens de script, como PHP, os servidores dessas aplicações tiveram que lidar com ataques de injeção (FAHL, 2021).

Hoje, a maioria das aplicações web possui quantidades consideráveis de código rodando no lado do cliente ao invés do servidor. Com o suporte dos navegadores para linguagens como Flash e JavaScript, muitos recursos puderam ser adicionados ao cliente, mas por outro lado a amplitude de possibilidades de ataques a aplicações web cresceu drasticamente. Novos tipos de ataques surgiram e tantos os fabricantes dos navegadores quanto os desenvolvedores das aplicações tiveram que tomar medidas visando a segurança de suas informações (FAHL, 2021).

Navegadores web se comunicam com serviços back-end usando tecnologias próprias da web. Essa comunicação é principalmente baseada no HTTP (Hypertext Transfer Protocol) e na sua versão mais segura conhecida como HTTPS (Hyper Text Transfer Protocol Secure). O lado do cliente costuma utilizar HTML, JSON, XML e JavaScript, este último podendo ser utilizado também no lado servidor (FAHL, 2021).

O HTTP é a forma mais utilizada para troca de documentos e dados entre servidores e clientes no meio web. Tendo como base o TCP/IP, o lado do cliente envia solicitações HTTP para um servidor HTTP, e este servidor retorna uma resposta HTTP com base no que foi solicitado (FAHL, 2021). Clientes HTTP podem enviar qualquer tipo de conteúdo para o servidor, e o servidor responderá a essas requisições com o conteúdo solicitado, caso seja encontrado, ou com mensagens de erro se for o caso.

Atualmente, é comum que as aplicações web façam conexões bidirecionais, onde não só o cliente envie dados para o servidor como também que o servidor envie dados para o cliente a qualquer momento. Isso é possível graças ao protocolo WebSocket, que basicamente é uma evolução do HTTP, que permite que ambas as partes possam trafegar dados a qualquer momento sem a necessidade de novas conexões (FAHL, 2021).

O HTTPS é o protocolo de rede seguro mais utilizado por aplicações web. Ele utiliza um esquema de segurança semelhante ao protocolo TLS para encapsular o HTTP, adicionando autenticação de servidor, integridade e confidencialidade para os dados que estão sendo transmitidos. O HTTPS protege os dados enviados via HTTP contra interceptação e adulteração, tentando evitar ataques de captura entre as pontas de comunicação.

Esse encapsulamento protege as URLs, cabeçalhos e o conteúdo do HTTP, mas não criptografa endereços IP e portas dos clientes e servidores (FAHL, 2021).

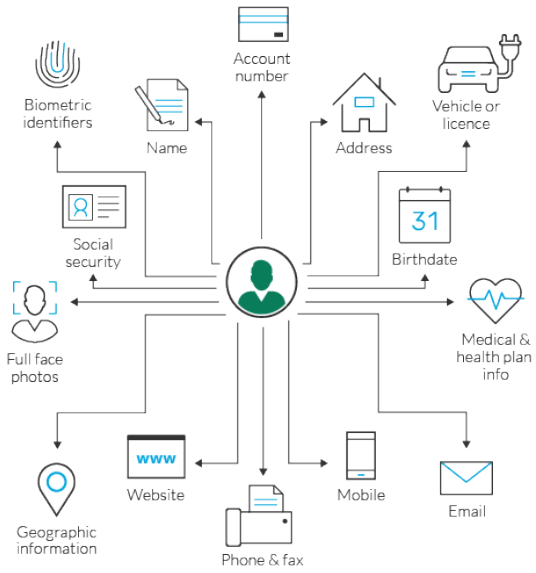
2.3.2 Dados sensíveis

Um dado sensível se refere a qualquer informação que deve estar protegida do acesso não autorizado. Alguns tipos de dados são claramente considerados sensíveis: dados bancários quaisquer, sejam números de cartões, dados de conta ou agência ou mesmo valores de transação. Mesmo se essas informações tiverem que ser mostradas, armazenadas ou trafegadas parcialmente, estas devem ser consideradas dados sensíveis. Senhas e códigos internos da aplicação também são dados comumente entendidos como sensíveis.

Porém, dados pessoais também devem ser considerados dados sensíveis. Informações como nome e sobrenome, data de nascimento e e-mail são informações pessoais, e só devem ser conhecidos pelos envolvidos em dado contexto de aplicação. Agentes maliciosos podem usar dados como esses, caso os recuperem, como ponto de partida para agrupar indivíduos baseado em outras informações roubadas, criando assim perfis para aplicar golpes específicos em cada perfil.

De acordo com o Departamento de Trabalho dos Estados Unidos (United States Department of Labor, 2022), são consideradas informações pessoais identificáveis (PII): nome completo, endereço de residência, telefone, e-mail, data de nascimento, estado civil, números de documentos, fotos e históricos de emprego e escolares, além de outras como ilustrado na Figura 1. Informações desse tipo devem ser consideradas sensíveis e também devem passar por regras de proteção, impedindo sua exposição de acessos não autorizados.

Figura 1 – Informações pessoais identificáveis.



Fonte: (Imperva, 2023)

A Lei Geral de Proteção de Dados (BRASIL, 2018) busca proteger direitos de privacidade, implementando níveis de segurança jurídica, padronizando práticas de uso de dados pessoais de brasileiros. Esta lei define que deve haver consentimento do titular dos dados sobre o que está sendo fornecido por ele, além de garantias como exclusão dos dados ou revogação do consentimento.

Vulnerabilidades que envolvam exposição desse tipo de dado podem, de acordo com a lei brasileira, gerar multas de até 2% do faturamento anual da organização responsável pelos dados (SERPRO, 2023).

2.3.3 Exposição de dados sensíveis

Em aplicações web, é possível armazenar informações no lado do cliente, mas isso pode permitir que agentes maliciosos tentem manipular esses dados. É importante então que essas áreas de armazenamento sejam protegidas. Para garantir a integridade dessas informações, uma das formas mais difundidas é adicionar assinaturas criptografadas a elas, e sempre verificá-las ao serem mandadas de volta para o servidor (FAHL, 2021).

Ao invés de tentar burlar a criptografia, agentes maliciosos podem tentar roubar dados diretamente no servidor, trafegando entre as pontas ou diretamente no navegador da aplicação. Vale ressaltar que esse tipo de ataque se beneficia da não encriptação de dados sensíveis, ou, quando há criptografia, são utilizadas chaves ou algoritmos fracos (The OWASP Foundation, 2017).

Uma vulnerabilidade como a exposição de dados é delicada por ser muito custosa. Em 2018 o instituto Poneman realizou um estudo (Ponemon Institute, 2018b) indicando que em média, há um custo de 7,9 milhões de dólares nos Estados Unidos quando uma empresa passa por um vazamento de dados. Além do dinheiro que precisa ser investido para solucionar o problema, existe também perda financeira com a desconfiança e abandono dos usuários, devido ao dano reputacional que um vazamento de dados leva (WILLIAMS, 2021).

Outro ponto que contribui para o encarecimento deste tipo de vulnerabilidade é o tempo que se leva para identificá-lo, segundo o mesmo estudo, a identificação de um vazamento de dados pode levar cerca de 200 dias, além de mais 70 dias para encontrar a fonte do problema e consertá-la.

2.4 OWASP

A OWASP (T*he Open Worldwide Application Security Project*) é uma fundação internacional que pesquisa e fornece para a comunidade informações práticas sobre segurança de software.

2.4.1 OWASP Top Ten

O OWASP Top Ten é um projeto que tem como objetivo produzir, para os desenvolvedores, um documento de conscientização de segurança de aplicações web. Este documento busca identificar quais são as 10 vulnerabilidades mais presentes nas aplicações web no ano de pesquisa, detalhando causas, como prevenir, exemplos de cenário de ataque e as CWEs relacionadas (The OWASP Foundation, 2021b).

O documento de 2021 registrou as seguintes vulnerabilidades:

- *Broken Access Control.*
- *Cryptographic Failures.*
- *Injection.*
- *Insecure Design.*
- *Security Misconfiguration.*
- *Vulnerable and Outdated Components.*
- *Identification and Authentication Failures.*
- *Software and Data Integrity Failures.*
- *Security Logging and Monitoring Failures.*
- *Server-Side Request Forgery.*

2.5 Criptografia

Criptografia é uma área que envolve questões matemáticas, além de conhecimentos em computação, software e hardware. Segundo Smart (2021), uma definição inata para criptografia é a de que um agente malicioso não deve conseguir recuperar uma chave criptográfica, nem mensagens criptografadas.

Desenvolvedores costumam utilizar algoritmos criptográficos já existentes fazendo uso de bibliotecas ou de APIs. Para tornar esses algoritmos utilizáveis por usuários em geral, é necessário considerar sua usabilidade e o projeto dessas APIs, pois além de serem seguras, essas bibliotecas precisam possuir certa facilidade de uso.

Essa espécie de disputa, "facilidade versus segurança", gera rupturas entre o quão seguro um algoritmo poderia ser se perfeitamente utilizado e o quão flexível seu uso deve ser para não limitar de forma excessiva quem busca utilizar esses algoritmos, e são nessas rupturas que surgem as vulnerabilidades de segurança. Exemplos como uso indevido de

bibliotecas criptográficas, má gestão de chaves e utilização de algoritmos básicos dentro de sistemas complexos são alguns dos mais comuns na origem de vulnerabilidades de software.

Uma aplicação de software precisa entender a criptografia como uma ferramenta que faz parte de um conjunto de outras ferramentas focadas em garantir a segurança de um sistema. É necessário proteger uma dada informação sensível desde o momento que ela sai de uma ponta, trafega, e chega até seu destino. Para Paterson (PATERSON, 2021), para assegurar um dado, é necessário usar ferramentas criptográficas no armazenamento desse dado, quando ele está em repouso, em trânsito, no processo de comunicação ou no ambiente de computação em nuvem.

2.6 Práticas de Desenvolvimento Seguro

Dada a grande quantidade de *feedbacks* de falhas e vulnerabilidades encontradas em sistemas, práticas vem sendo propostas que, se aplicadas desde o início do desenvolvimento de um sistema, podem evitar exposição de dados.

Segundo Williams (WILLIAMS, 2021), o ciclo de vida de segurança de software deve ser proativo, buscando eliminar vulnerabilidades desde a fonte, ao invés de tratar as consequências quando o problema acontecer.

2.6.1 SDL

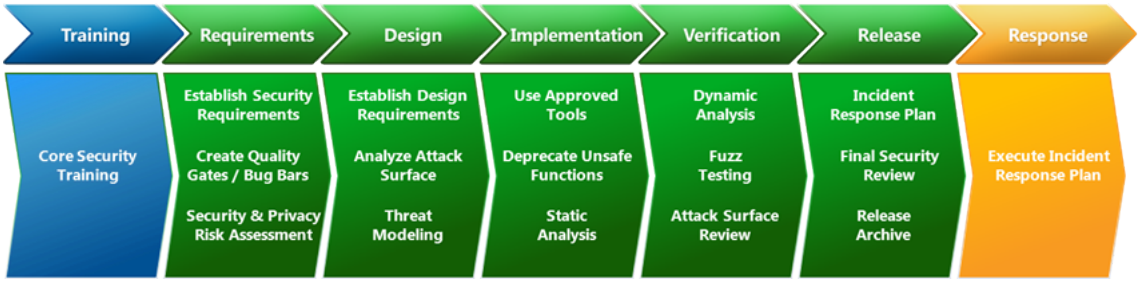
A metodologia de desenvolvimento seguro da Microsoft (SDL - *Security Development Lifecycle*) (Microsoft, 2023b) faz uso de práticas para tornar o desenvolvimento de seus produtos mais seguros, representadas na figura 2. Sendo elas:

- **Disponibilização de treinamento:** Geralmente, a formação atual de desenvolvedores de software não inclui questões ou áreas de segurança digital. Além disso, modelos de ataque e ferramentas de segurança mudam e se atualizam com frequência (WILLIAMS, 2021). Sabendo disso, empresas devem estimular e fornecer fontes de conhecimento como cursos e treinamentos para atualizar seus desenvolvedores, afim de aprimorar técnicas de detecção e prevenção de vulnerabilidades.
- **Definição de requisitos de segurança:** Requisitos de segurança devem ser estabelecidos desde o início de um projeto, tanto do planejamento quanto desenvolvimento, e estes requisitos devem estar abertos a atualizações constantes, com o objetivo de refletirem as funcionalidades do projeto e as ameaças conhecidas no momento.

- **Definição de métricas:** A equipe de gerenciamento de um projeto deve definir valores mínimos que devem ser atendidos em relação à métricas de segurança (MORRISON et al., 2018). É importante ter um rastreamento bem definido para definições como priorização e gerenciamento de riscos.
- **Modelagem de ameaças:** As equipes devem considerar possíveis motivações de atacantes, além das forças e fraquezas do sistema, contemplando os limites de confiança e o fluxo de dados do sistema.
- **Estabelecer requisitos de projeto:** A equipe deve estar orientada a fazer implementação de recursos bem projetados em termos de segurança. Tanto a arquitetura como o projeto devem resistir às ameaças conhecidas no ambiente operacional pretendido.
- **Definir e usar padrões de criptografia:** A criptografia é um recurso importante para garantir segurança e privacidade de dados sensíveis. Especialistas devem ser consultados para utilização correta e uso de bibliotecas adequadas.
- **Gerenciar o risco de segurança do uso de componentes de terceiros:** Componentes de terceiros podem ter vulnerabilidades. A equipe de desenvolvimento deve ter um controle preciso desses componentes, utilizando ferramentas de varredura de vulnerabilidades e ter um plano de resposta caso uma vulnerabilidade seja descoberta.
- **Usar ferramentas aprovadas:** A equipe deve publicar uma lista com as ferramentas aprovadas e as verificações de segurança associadas. Deve-se utilizar as versões mais recentes das ferramentas, aproveitando as funcionalidades de análise de segurança e proteções mais recentes.
- **Executar testes de segurança de análise estática (SAST):** Testes de análise estática buscam encontrar padrões de codificação insegura e que não sigam políticas de codificação segura. Ferramentas que executam esse tipo de teste podem ser integrados ao pipeline de *commits* e implantação.
- **Executar testes de segurança de análise dinâmica (DAST):** Testes de análise dinâmica verificam em tempo de execução o software compilado para verificar funcionalidades que só são aparentes com a integração e execução dos componentes. Esse tipo de teste pode detectar problemas de memória, problemas de autenticação, ataques de injeção e outros critérios de segurança.
- **Realizar teste de penetração:** Testes de penetração simulam ações de um atacante. Este tipo de teste pode descobrir outras formas de vulnerabilidade, desde pequenos erros de implementação até grandes falhas de codificação ou outros tipos de fraqueza.

- **Estabelecer um Processo Padrão de Resposta a Incidentes:** A equipe deve estar preparada para ataques inevitáveis. Devem possuir um plano de respostas a incidentes. Este plano deve incluir a quem contatar em caso de emergência de segurança, estabelecer protocolo para mitigação eficiente de vulnerabilidades, resposta e comunicação com clientes e implantação rápida de correção.

Figura 2 – Microsoft SDL.



Fonte: (Microsoft, 2023b)

2.6.2 Prevenções para exposição de dados sensíveis

De acordo com a OWASP (The OWASP Foundation, 2021a), algumas práticas que podem prevenir a exposição de dados sensíveis são:

- Classificar os dados processados, armazenados ou transmitidos pela aplicação. Identificar quais dados são sensíveis de acordo com leis de privacidade, requisitos regulamentares ou regras de negócio.
- Não armazenar dados sensíveis sem necessidade. Deve-se descartar este tipo de dados o mais rápido possível. Dados que não são retidos não podem ser vazados.
- Criptografar todos os dados sensíveis armazenados.
- Utilizar algoritmos, protocolos e chaves atualizados. Fazer uma gestão adequada das chaves.
- Criptografar dados transmitidos com protocolos seguros, como o TLS, utilizando cifras adequadas, priorização de cifras pelo servidor e parâmetros seguros. Reforçar a criptografia utilizando diretivas como *HTTP Strict Transport Security* (HSTS).
- Desativar armazenamento em cache em respostas que contenham dados sensíveis.
- Aplicar controles de segurança necessários de acordo com a classificação dos dados.
- Não utilizar protocolos legados como FTP ou SMTP no transporte de dados sensíveis.

- Armazenar senhas utilizando funções de hash adaptativas como Argon2, scrypt, bcrypt or PBKDF2.
- Vetores de inicialização devem ser escolhidos de forma apropriada para cada modo de operação. Em muitos modos, deve-se utilizar um gerador de números pseudoaleatórios criptograficamente seguro (CSPRNG). Um mesmo vetor de inicialização nunca deve ser usado duas vezes para uma mesma chave.
- Sempre utilizar criptografia com autenticação.
- Chaves devem ser geradas criptograficamente de forma aleatória e armazenadas na memória como *arrays* de bytes. Se uma senha for utilizada, ela deve ser convertida em uma chave por meio de mecanismos apropriados de derivação de chaves baseada em senha.
- Utilizar aleatoriedade criptográfica quando necessário e que não tenha sido inicializada de forma previsível ou com baixa entropia. A maioria das APIs modernas não exigem que o desenvolvedor inicialize o gerador de números pseudoaleatórios criptograficamente seguro com uma semente para obter segurança.
- Evitar utilizar funções criptográficas e esquemas de *padding* obsoletos como MD5, SHA1, PKCS #1.
- Verificar de forma independente a eficácia das configurações.

3 Metodologia

3.1 Considerações Iniciais do Capítulo

Neste capítulo é descrita a metodologia utilizada para auxiliar no desenvolvimento do projeto. Metodologia esta que deve ser classificada quanto sua natureza, abordagem, tipo e procedimentos de investigação.

- **Natureza:** Aplicada, pois tem como objetivo gerar conhecimentos práticos que podem auxiliar a resolver problemas específicos.
- **Abordagem:** Qualitativa, pois leva em consideração relações entre o mundo real e aqueles que interagem diretamente com ele, buscando identificar fenômenos e atribuir relações a estes fenômenos (MINAYO, 2012). A pesquisa não obrigatoriamente fará uso de métodos estatísticos, utilizando do ambiente natural para coletar dados.
- **Tipo:** Descritiva, pois faz uso de recursos como pesquisa bibliográfica e documental.
- **Meios de investigação:** Pesquisa documental e Pesquisa bibliográfica

O objetivo deste trabalho é a proposição de um guia de desenvolvimento seguro em aplicações web, focado em prevenir exposições de dados sensíveis. Este objetivo passa pela identificação e análise de casos reais onde houve a ocorrência desta vulnerabilidade.

A pesquisa bibliográfica é realizada fazendo levantamento de um referencial teórico já analisado e publicado, com o objetivo de conhecer o que já foi estudado sobre um assunto, selecionando e organizando materiais que contribuam para a construção do projeto. É realizada em cima de pesquisas anteriores ou outros documentos impressos como artigos e livros (SOUSA; OLIVEIRA; ALVES, 2021).

A pesquisa documental também faz uso de um texto como objeto de estudo, porém, a definição de documento expande o conceito de textos escritos/impressos abordados na pesquisa bibliográfica. Enquanto a pesquisa bibliográfica trabalha com fontes como livros e artigos, a pesquisa documental considera documento materiais como vídeos, slides ou reportagens, sendo estas fontes de informações e esclarecimentos para chegar a conclusões (Sá-SILVA; ALMEIDA; GUINDANI, 2009).

A Tabela 1 demonstra a relação das metodologias com os objetivos específicos estabelecidos.

Objetivo 1: Descrever e detalhar o que são Exposição de Dados Sensíveis e Falhas de Criptografia	Pesquisa e Análise Bibliográfica
Objetivo 2: Pesquisar e relatar detalhadamente casos reais envolvendo este tipo de vulnerabilidade, incluindo causas raiz, impactos e consequências dos incidentes	Pesquisa e Análise Documental
Objetivo 3: Identificar possíveis medidas que poderiam ter evitado a exposição dos dados nos casos relatados	Análise de casos e tomada de decisão
Objetivo 4: Relacionar os casos levantados com o material teórico estudado	Análise de casos e pesquisa e análise bibliográfica
Objetivo 5: Propor um guia prático de desenvolvimento seguro focado nas vulnerabilidades especificadas, com base no que foi encontrado nos casos analisados	Desenvolvimento do guia com base nos resultados obtidos

Tabela 1 – Objetivos e métodos

Abaixo é descrito como a metodologia escolhida será utilizada para atingir cada um dos objetivos estipulados.

3.2 Descrever e detalhar o que são Exposição de Dados Sensíveis e Falhas de Criptografia

Este objetivo é atingido com a realização de pesquisa bibliográfica sobre definições das vulnerabilidades, incluindo contextos sobre os quais as vulnerabilidades estão inseridas. Além das definições, a realização de uma revisão de literatura em cima de outras informações relevantes ao contexto de segurança, desenvolvimento web, dados sensíveis e criptografia se mostra importante para melhor compreensão e análise de casos relacionados.

Será considerado um dado sensível:

- Dados bancários, como partes ou todo número de um cartão, dados referentes a conta ou agência bancária ou valores de transações;
- Senhas em geral;
- Códigos internos da aplicação, como a propriedade “id” de um objeto
- Informações pessoais identificáveis (PII), como nome, endereço, telefone, e-mail, entre outros (United States Department of Labor, 2022).

3.3 Pesquisar e relatar detalhadamente casos reais envolvendo este tipo de vulnerabilidade, incluindo causas raiz, impactos e consequências dos incidentes

Com o uso da pesquisa documental, será realizado uma busca por reportagens, documentos judiciais e outros materiais de divulgação que evidenciem casos de empresas onde houve exposição de dados sensíveis. Além de materiais que evidenciem os casos, também será buscado documentos onde foram feitos análises sobre o caso, identificando vulnerabilidades e falhas que puderam ser exploradas e como foram exploradas. Também será de interesse materiais que evidenciem as consequências da exposição para a empresa, seja com perda de usuários ou necessidade de investimento de capital.

Este objetivo será atingido a partir dos seguintes passos:

- Seleção de casos de estudo: Identificação e seleção de casos reais de exposição de dados sensíveis que podem ser analisados.
  - Deve-se analisar a relevância dos casos e a disponibilidade de informações para análise.
- Coleta de dados: Coleta de informações sobre os casos selecionados, que podem incluir documentos como relatórios de incidentes, notícias e artigos. Deve conter os seguintes pontos:
  - **Causas raiz:** Fatores ou elementos que permitiram a possibilidade de exposição dos dados sensíveis. A identificação das causas permite entender os pontos fracos em sistemas, processos ou práticas que permitiram a existência da vulnerabilidade.
  - **Métodos de ataque utilizados:** Técnicas e abordagens utilizadas pelos agentes maliciosos para explorar as falhas existentes e obter acesso aos dados sensíveis.
  - **Tipos de dados expostos:** As informações confidenciais cuja exposição causou danos aos proprietários dos dados e quais as características dessas informações (PIIs, dados bancários, informações médicas, senhas).
  - **Atitudes tomadas pela empresa:** Medidas de resposta à exposição da vulnerabilidade como ações de resposta à violação, notificação aos usuários afetados, aplicação de melhorias das medidas de segurança, responsabilização quanto ao incidente.
  - **Consequências:** Que indivíduos foram afetados e como, que danos foram causados às organizações ou empresas, quais impactos negativos ocorreram

a terceiros ou parceiros comerciais, caso existam, como empresas terceirizadas responsáveis pela segurança. A compreensão das consequências, diretas ou indiretas, auxiliam na avaliação do impacto e na conscientização sobre a importância de se cuidar de dados sensíveis.

3.4 Identificar possíveis medidas que poderiam ter evitado a exposição dos dados nos casos relatados

Com base nos casos coletados e detalhados, um estudo deve ser realizado em cada um deles, buscando encontrar e hipotetizar uma ou mais soluções onde a exposição dos dados não aconteceria. A base para a hipótese dessas soluções é o conhecimento adquirido durante a pesquisa bibliográfica realizada para cumprir com o primeiro objetivo. Uma relação entre práticas que previnam e mitiguem a vulnerabilidade em questão e o caso abordado deverá ser realizada, identificando qual ou quais práticas melhor se associam, além de propor um exemplo prático demonstrando a solução do problema.

Os seguintes pontos devem ser seguidos:

- Análise dos casos: Análise aprofundada de cada caso, examinando as vulnerabilidades e falhas que permitiram a exposição dos dados sensíveis.
- Identificação de medidas de segurança: Identificar pontos em que medidas poderiam ter sido aplicadas para evitar ou mitigar o problema.
  - Aqui devem ser considerados aspectos técnicos, processuais e humanos envolvidos.

3.5 Relacionar os casos levantados com o material teórico estudado

Tendo coletado e analisado os casos, serão feitas relações com os mesmos e as práticas de desenvolvimento seguro encontradas e estudadas na fase de levantamento do referencial teórico, buscando mostrar a utilidade dessas práticas, além de demonstrar como os problemas encontrados possuem estratégias de prevenção claras.

O objetivo desta fase é conectar os estudos bibliográficos com os documentais, demonstrando a importância do conteúdo bibliográfico para o desenvolvimento de sistemas com fortes características de segurança.

3.6 Propor um guia prático de desenvolvimento seguro focado nas vulnerabilidades especificadas, com base no que foi encontrado nos casos analisados

Com base nas análises e identificação de medidas dos casos e no conhecimento adquirido com os estudos relacionados ao tema e chegando a conclusões e padronizações relacionando os problemas às soluções, será possível determinar linhas de desenvolvimento e medidas de segurança que auxiliem no ciclo de vida de aplicações web.

A junção destas práticas com informações coletadas ao longo da pesquisa produzirão um guia prático de desenvolvimento, com soluções técnicas, práticas recomendadas e estratégias de mitigação que permitirão a contextualização, a prevenção e a solução de problemas relacionados à vulnerabilidade conhecida como exposição de dados sensíveis.

4 Análise de Casos

4.1 Considerações Iniciais do Capítulo

Nos últimos anos, não raros foram os casos noticiados e estudados de empresas multinacionais que enfrentaram vazamentos de dados. Analisar e compreender esses casos é importante para que erros como esses não sejam repetidos, e para evitar que problemas ocorram no futuro, primeiro deve-se conhecer o passado e lembrar dele.

Entretanto, grandes casos de vazamento de dados são acompanhados de muitos detalhes, então também é importante simplificar, abstrair detalhes e entender principalmente quais foram as causas raiz de um vazamento de dados.

Neste Capítulo, serão analisados alguns dos grandes casos de vazamento de dados de empresas conhecidas, tendo foco em suas causas raiz, os métodos de ataque utilizados, quais tipos de dados foram expostos, atitudes tomadas e consequências enfrentadas pela empresa, além de estratégias de mitigação para os problemas apontados. Os casos foram selecionados com base na quantidade de pessoas afetadas, na repercussão e no peso que o nome das empresas carregam.

4.2 LinkedIn

O LinkedIn é uma rede social profissional voltada para a conexão de profissionais de diversas áreas. É uma plataforma onde seus usuários listam suas experiências profissionais, habilidades, formação acadêmica e outros detalhes relevantes (LinkedIn, 2023), além de preencherem informações pessoais. A plataforma permite que o usuário defina quem pode ver suas informações.

Em 2021, dois vazamentos de dados ocorreram em menos de três meses na plataforma, em abril (Cybernews Team, 2021) e em junho (MORRIS, 2021). Em junho, foi divulgado que PIIs de 700 milhões de usuários foram vazados, o que corresponde a cerca de 93% dos seus usuários, com dados atualizados de 2020 e 2021 (MORRIS, 2021). O ataque foi realizado por um agente malicioso com o nome “TomLiner” (GIBSON et al., 2021).

4.2.1 Causas raiz

O LinkedIn oferece um serviço de API, onde disponibiliza produtos voltados para áreas específicas como: Consumidor, Marketing, Vendas, Plugins e Contratação. Para oferecer esse tipo de funcionalidade, a empresa optou por abrir mão da privacidade de

alguns dados dos usuários para permitir que desenvolvedores usassem essas informações para montar estatísticas, além de compor outras ferramentas de análises de dados, o que permitiu ao LinkedIn uma maneira adicional de gerar receita indireta (GIBSON et al., 2021). O vazamento acabou por mostrar, porém, o quão invasivo a empresa se permitiu ser, dados os tipos de informações divulgadas.

Segundo o próprio atacante, os dados vazados foram obtidos explorando essa API do LinkedIn (TAYLOR, 2021).

4.2.2 Métodos de ataque utilizados

Em abril de 2021, dados de milhões de usuários de do LinkedIn apareceram para venda online na *dark web*, similar a um incidente ocorrido com o Facebook na mesma época (VANIAN, 2021b). Ambas empresas tiveram seus dados obtido por *scrapping* (VANIAN, 2021a).

Scrapping é uma técnica usada para extrair informações de sites de maneira automatizada. Essa técnica faz uso de programas ou scripts para percorrer páginas web, capturando seus dados e os organizando em formatos úteis, como agrupando informações similares. Apesar de útil, seu uso deve ser realizado de forma ética e respeitando termos de uso de dados dos sites.

Para obter acesso aos dados, o agente malicioso fez uma combinação de dados recuperados a partir da API do LinkedIn com APIs de outros sites que também tiveram vazamentos de dados semelhantes, como o Facebook, montando assim um banco de dados mais robusto de PIIs (Informações Pessoalmente Identificáveis).

O atacante então divulgou, na *dark web* informações de 1 milhão, dos 700 milhões de usuários, para validar a veracidade das informações (TAYLOR, 2021).

Em relação à forma de se obter os dados, o agente malicioso utilizou uma *third party*, ou ferramenta de terceiros, para abusar da API do LikedIn (TAYLOR, 2021). A API permitiu que essa ferramenta fizesse um *scrapping* e agrupamento dos dados gerados tanto pela API do LinkedIn quanto de outras fontes e compilasse essas informações em uma lista robusta de dados sobre os usuários.

4.2.3 Tipos de dados expostos

Os dados vazados eram compostos por informações pessoais tais como (Cybernews Team, 2021):

- Nomes completos;
- Números de telefone;

- Endereços;
- E-mails;
- Dados de localizações;
- Nomes de usuário e URLs de perfis;
- Experiências pessoais e profissionais, além de históricos;
- Gêneros;
- Nomes de usuário e contas de outras redes sociais.

A Figura 3 traz um exemplo de dados compartilhados pelo atacante na *dark web*, exemplificando os tipos de dados expostos.

Figura 3 – Exemplo de dados vazados.

Fonte: (GIBSON et al., 2021)

4.2.4 Atitudes tomadas pela empresa

Inicialmente, o LinkedIn negou que os dados estariam atualizados. Negou também que muitos dos dados foram obtidos por meio do *scrapping* de seu próprio site (MORRIS, 2021). Posteriormente a empresa alegou que os dados obtidos eram dados disponíveis publicamente e que apenas foram obtidos em larga escala, alegando que não houve exposição de dados e muito menos de dados privados. Porém, admitiu que foi realizado *scrapping* de seu site e que os dados foram obtidos de seus servidores. Afirmaram também que quando este tipo de ação é tomada e os dados são usados para propósitos que a empresa não concorda, a empresa trabalha para impedir e responsabilizar os agentes (LinkedIn, 2021).

4.2.5 Consequências

Esses vazamentos impactaram não só a empresa, mas também a visão de segurança de dados no mundo, principalmente levando em conta o número de usuários expostos nas duas ocorrências, chegando a mais de 700 milhões de usuários comprometidos, equivalente a 93% dos usuários do LinkedIn (MORRIS, 2021).

Um vazamento deste tamanho tem impactos tanto para os usuários quanto para a empresa. Os usuários tiveram suas informações expostas, podendo virar alvo de tentativas de golpe, além de perderem confiança na estrutura de segurança do LinkedIn. A empresa por outro lado passou por uma violação de dados que afetou a maior parte de seus usuários, exigindo o estabelecimento de um processo de recuperação da confiança de seus usuários para que continuem a compartilhar suas informações com a plataforma (GIBSON et al., 2021).

Analisando do ponto de vista dos usuários, a principal consequência é ter seus dados públicos (JOHNSON, 2021). Com isso, cada um desses usuários estão mais expostos a ataques de *phishing* vinculados aos seus dados. Agentes que tenham a intenção de aplicar golpes puderam ter acesso a dados pessoais sensíveis como nome completo, endereço, e-mails, telefones, cargos profissionais, além de outros, informações essas que não eram públicas por escolha do usuário. Com essas informações, ataques de *phishing* conseguem ser mais convincentes. Outro ponto que deve ser destacado é que com informações de empresas como nomes, endereços e cargos se tornando públicos, é mais fácil para que hackers coletem mais detalhes sobre essas empresas, repetindo processos de *scrapping*.

A perda de reputação também foi um impacto significativo, especialmente ao se considerar que é uma empresa relacionada à Microsoft. Tendo sofrido dois vazamentos em questão de poucos meses, criou-se uma dúvida pública do cuidado que Microsoft e LinkedIn possuem com os dados de seus usuários. Na violação de dados mais antiga, a Microsoft apenas ofereceu monitoramento gratuito para pessoas afetadas (JOHNSON, 2021). Apesar disso contribuir com a opinião pública, não é o suficiente para reverter os danos causados (GIBSON et al., 2021).

4.2.6 Estratégias de mitigação

Apesar do vazamento de dados não ter sido realizado através de um *hack*, é importante implementar métodos que não permitam que usos abusivos de uma API aconteçam, como foi o caso. Algumas estratégias de mitigação seriam: utilizar autenticação e autorização, estabelecer limites na quantidade de dados que podem ser obtidos, monitorar as APIs em relação a seu uso.

A ação mais consistente que se pode tomar nesse tipo de caso é fortalecer a segurança no acesso da API. Como a API serve como uma porta entre o usuário e os dados

que se busca acessar, é nesse ponto que conceitos como autenticação e autorização podem ser aplicados para aumentar a segurança em solicitações (Malwarebytes Labs, 2021). Não fazer uso desses tipos de conceitos permitem que qualquer pessoa acesse os dados disponibilizados pela empresa, permitindo que um agente malicioso faça, sem dificuldade, um *scrapping* na API. A autenticação e a autorização podem ser implementados fazendo uso de chaves criptografadas para usar a API, além de um sistema de autenticação de identidade antes de ter acesso aos dados. Além disso, o servidor deve autorizar cada requisição feita para a API. Isso torna mais difícil que sejam realizadas grandes quantidades de chamadas para a API (GIBSON et al., 2021).

Outra ação que pode ser tomada é monitorar a API, rastreando atividades anormais. Isso irá permitir que a empresa seja alertada quando um grande número de dados está sendo solicitado de uma só vez, podendo interromper essa solicitação antes que os dados sejam passados para quem está fazendo a solicitação. Esse tipo de ação serve como uma contingência para quando a autenticação falhar, prevenindo intrusões por meio de bloqueio automático de atividades suspeitas (GIBSON et al., 2021). A empresa pode implementar um rastreamento de consumo de dados em seus servidores, assim como ferramentas da Azure fazem com gerenciamento de bancos de dados (Microsoft, 2023a) e *websites* (Microsoft, 2023c), permitindo que o administrador acompanhe em valores quanto está sendo exigido e consumido de seus servidores, assim como quanto está sendo gravado no mesmo. Pode também determinar valores arbitrários que indicam quando esse consumo entra em nível de alerta e deve ser acompanhado ou interrompido.

Pode-se também estabelecer um limite de dados que pode ser obtido por requisição na API. Esse tipo de trava impediria que dados de 90% dos usuários fossem vazados de uma só vez. É interessante monitorar usuários que estão lendo grandes quantidades de dados e diminuir a quantidade que pode ser lida. Esse tipo de abordagem possibilitaria inclusive implementar níveis de serviço, sendo ao mesmo tempo uma oportunidade de negócio e regulamentação de como a API está sendo usada (GIBSON et al., 2021).

Caso nenhum método de controle seja implementado, são grandes as chances de ocorrência de episódios de abuso como este.

4.3 Yahoo

Fundada em 1994, o Yahoo é um portal web que fornece serviços como buscador, e-mail, notícias e publicidade.

Em setembro de 2016, a empresa anunciou que um hacker invadiu seus sistemas em 2014 e roubou informações de mais de 500 milhões de contas (TRAUTMAN; ORMEROD, 2017). Investigações relacionadas apontaram que os responsáveis pelos ataques foram dois

agentes russos, Dmitry Dokuchaev e Igor Sushchin e dois hackers independentes, Alexsey Belan e Karim Baratov (DASWANI; ELBAYADI, 2021).

Até o momento em que este evento foi anunciado, este caso era o maior caso de vazamento de dados até então (PERLROTH, 2016). Mas em dezembro de 2016, a empresa anunciou a ocorrência de outro vazamento, este em 2013, que afetou entre 1 bilhão (MCMILLAN; KNUTSON; SEETHARAMAN, 2016) e 3 bilhões (DASWANI; ELBAYADI, 2021) de contas. Uma linha do tempo com os principais eventos desse caso pode ser vista na Figura 4.

Figura 4 – Linha do tempo dos ataques ao Yahoo.

Fonte: (DASWANI; ELBAYADI, 2021)

4.3.1 Causas raiz

O vazamento de 2014 teve início com um funcionário da empresa sendo vítima de *phishing* (DASWANI; ELBAYADI, 2021), técnica de engenharia social que engana um usuário com o objetivo de obter informações confidenciais, como logins e senhas.

De acordo com uma investigação do Yahoo, um agente malicioso havia obtido acesso a algumas contas de usuários explorando a ferramenta de gerenciamento de contas da empresa. A equipe de segurança chegou a conclusão de que o atacante extraiu cópias de arquivos de backup do banco de dados que possuía dados pessoais dos usuários do Yahoo (TRAUTMAN; ORMEROD, 2017).

Quanto ao vazamento de 2013, segundo investigações realizadas pela Verizon, após a compra do Yahoo, a empresa sofreu um roubo de seus códigos-fontes, o que permitiu que o agente malicioso fabricasse *cookies* do site do Yahoo (DASWANI; ELBAYADI, 2021).

A base de dados do Yahoo armazenava senhas criptografadas com funções *hash* (DASWANI; ELBAYADI, 2021), que são funções que embaralham textos onde dada uma senha, é fácil calcular matematicamente se ela corresponde ao resultado *hash* armazenado.

4.3.2 Métodos de ataque utilizados

Um dos atacantes copiou a base de dados do Yahoo, com informações de 500 milhões de contas, para seu computador, além do código-fonte que o Yahoo usava para gerar os *cookies* de seu site (DASWANI; ELBAYADI, 2021).

Cookies são pedaços de código que são armazenados na memória cache de um navegador, para que o site não exija login a cada acesso, e segundo o diretor de segurança da informação, Bob Lord, o ataque de 2013 foi realizado com a utilização de *cookies* falsificados (THIELMAN, 2016).

Neste tipo de ataque, o agente tenta acessar contas de usuários sem possuir a senha adequada, contando que o *cookie* falso o identifique como dono de uma conta de e-mail válida, o que permitiu que os atacantes acessassem as contas do Yahoo mesmo não possuindo suas senhas (DASWANI; ELBAYADI, 2021).

Os atacantes fizeram uso também da ferramenta de gerenciamento de contas do Yahoo, que tem como funcionalidade editar contas de usuários. Os agentes maliciosos procuraram pelas contas de endereço de e-mail de recuperação dos usuários (DASWANI; ELBAYADI, 2021).

Por exemplo, se os atacantes estivessem interessados em acessar as contas de usuários da UnB, os e-mails de recuperação desses usuários provavelmente seguiriam o padrão nome@unb.br. Buscando por todas as contas que tivessem um e-mail de recuperação @unb.br, os atacantes teriam acesso as informações de todos os funcionários e estudantes da universidade.

Os agentes maliciosos tiveram acesso a informações de muitas organizações pesquisando com base nos e-mails de recuperação das mesmas (DASWANI; ELBAYADI, 2021).

Os atacantes fizeram uso de aplicações *malware* de limpeza de *logs* para esconder suas atividades dentro da rede do Yahoo (DASWANI; ELBAYADI, 2021).

4.3.3 Tipos de dados expostos

A informação roubada incluía dados como (TRAUTMAN; ORMEROD, 2017):

- Nomes;
- Datas de nascimento;

- Números de telefone;
- E-mails;
- Senhas criptografadas;
- Perguntas e respostas de segurança, tanto criptografadas como descriptografadas.

Este último item possui uma sensibilidade em especial já que perguntas de segurança podem dar acesso a redefinições de senhas.

4.3.4 Atitudes tomadas pela empresa

O relatório da investigação feita pela empresa registra e detalha os esforços de certas áreas da empresa para tentar responder às ameaças, mas registra também negligência em outras áreas executivas, que "deixaram de agir de maneira suficiente, com base no conhecimento interno da equipe de segurança da empresa" (TRAUTMAN; ORMEROD, 2017).

De acordo com um relatório preenchido pela empresa, alguns dos executivos de maior relevância falharam em compreender e investigar o problema corretamente que foi informado a eles pela equipe de segurança (DASWANI; ELBAYADI, 2021).

A principal atitude a ser destacada foi a demora por parte do Yahoo em divulgar os vazamentos de dados, principalmente quando se leva em conta o grande número de usuários afetados. A empresa demorou mais de dois anos para divulgar que dados tinham sido roubados de seus servidores (DASWANI; ELBAYADI, 2021).

4.3.5 Consequências

A empresa enfrentou consequências financeiras sem precedentes, além de perder credibilidade e reputação da marca (DASWANI; ELBAYADI, 2021).

O Yahoo teve que pagar multas que chegaram a centenas de milhões de dólares e obedecer a ações judiciais, além de reduzirem a avaliação da empresa e perder a confiança de seus usuários e clientes, chegando a perder efetivamente alguns destes.

Especialistas em áreas de segurança já criticavam os baixos níveis de proteção de serviços do Yahoo como o e-mail, já que era observável a ocorrência de spams, além de outros ataques baseados em e-mail, depois da divulgação dos ataques, essas críticas aumentaram (THIELMAN, 2016), reforçando a má publicidade.

No segundo semestre de 2016, o Yahoo estava em processo de ser adquirido pela empresa Verizon (HACKETT, 2016), e o anúncio dos vazamentos impactou negativamente no valor de compra negociado (ATKINSON, 2016).

4.3.6 Estratégias de mitigação

Em conceitos técnicos, o Yahoo falhou em proteger seus sistemas principalmente por causa de *phishing*, usos de *malware* e um ataque que forjava *cookies*.

A forma mais disseminada de se evitar ataques de *phishing* é educar e treinar pessoas. Por ser um ataque que envolve engenharia social, mais do que conhecimentos técnicos acerca do problema, usuários e funcionários precisam estar cientes acerca dos artifícios que um agente malicioso usa para recuperar informações sigilosas. Dinâmicas de jogos e programas interativos são exemplos de atitudes que uma empresa pode implementar para preparar seus funcionários para este tipo de ataque (SUMNER; YUAN, 2019).

Quanto ao uso de *malwares* nos servidores de uma empresa, cabe à equipe de segurança monitorar seus sistemas, fazendo uso de análises estáticas e dinâmicas. Análises estáticas consistem em analisar os códigos-fonte dos sistemas que estão ativos no servidor das empresas, verificando as DLLs (bibliotecas compartilhadas) chamadas pelo programa, além de URLs acessadas pelo mesmo. Em alguns casos, essa simples análise pode ser o suficiente para determinar quais ações escondidas uma aplicação está executando em um servidor. As análises dinâmicas devem ser aplicadas em situações onde o código do *malware* não pode ser acessado. Nesses casos, deve se analisar a execução do *malware* e o comportamento do mesmo no sistema. Antes de realizar a execução do *malware* para analisá-lo, ferramentas *anti-malware* e de *debug* devem estar pré-instaladas, para impedir que sua execução não danifique o sistema (CHAKKARAVARTHY; SANGEETHA; VAIDEHI, 2019).

Para o problema dos *cookies*, o Yahoo deveria fazer uso de uma chave secreta que fosse conhecida apenas pela empresa na geração de *cookies*, para impedir que eles fossem forjados. Esta chave não estaria explicita no código fonte e nem seria armazenada com o mesmo, evitando expor uma vulnerabilidade no design da aplicação (DASWANI; ELBAYADI, 2021).

4.4 eBay

O eBay é um mercado online que permite que as pessoas realizem transações comerciais entre si. Para garantir certa confiabilidade, o eBay utiliza sistemas de reputação onde os usuários acumulam classificações com base em suas transações, onde tanto os compradores avaliam os vendedores, como os vendedores também avaliam os compradores.

No primeiro trimestre de 2014, um ataque de *spear-phishing* causou a exposição de cerca de 145 milhões de contas de usuários do eBay (ROBERTS, 2018).

4.4.1 Causas raiz

Sendo o *phishing* o método principal de ataque utilizado, a principal causa raiz está na exploração da psicologia básica humana. Ao considerar que o e-mail recebido parecerá ser de uma fonte confiável, é inevitável que haja pessoas que se tornarão vítimas, independentemente do quão cientes estejam dos perigos das ameaças à segurança (PARMAR, 2012).

4.4.2 Métodos de ataque utilizados

O *spear-phishing* é uma evolução do *phishing* tradicional. Enquanto o *phishing* envia e-mails para milhares de usuários, esperando que alguns deles caiam no golpe, os ataques de *spear-phishing* são direcionados e envolvem indivíduos específicos dentro de organizações específicas para que baixem aplicações *malware* para dentro de suas máquinas (PARMAR, 2012). Esse tipo de ataque exige uma pesquisa sobre as informações das vítimas como nome, cargo, interesses e até detalhes da empresa onde a vítima trabalha, o que torna o golpe mais convincente.

Não se tem certeza se com esse ataque, um *malware* foi instalado na máquina da empresa ou se um funcionário-chave forneceu suas credenciais de login, mas por meio deste tipo de ataque os agentes maliciosos conseguiram acesso a rede da empresa. Os atacantes passaram meses dentro da rede do eBay, coletando informações pessoais de cada usuário do eBay (ROBERTS, 2018).

4.4.3 Tipos de dados expostos

O eBay declarou que os tipos de dados expostos foram (ROBERTS, 2018):

- Nomes de clientes;
- E-mails;
- Endereços;
- Números de telefone;
- Datas de nascimento;
- Senhas (KOLEVSKI et al., 2021).

Cabe destacar que, com exceção das senhas, nenhuma dessas informações estava criptografada. As informações financeiras estavam criptografadas, e portanto, não foram comprometidas (ROBERTS, 2018).

4.4.4 Atitudes tomadas pela empresa

A empresa se manifestou admitindo que as informações pessoais dos usuários foram obtidas, mas negou qualquer risco de perda relacionado a informações financeiras, já que os detalhes das contas eram mantidos separados das informações financeiras (HOLM; MACKENZIE, 2014).

O eBay notificou seus clientes sobre a violação e os instruiu a alterar suas credenciais de login para evitar riscos adicionais, apesar do atraso de dois meses dessa notificação em relação ao período de descoberta da mesma (HOLM; MACKENZIE, 2014).

4.4.5 Consequências

A empresa teve que enfrentar ações judiciais coletivas por conta da exposição de dados pessoais de seus clientes (KOLEVSKI et al., 2021). Estima-se que o eBay tenha arcado com custos de cerca de 300 milhões de dólares (ROBERTS, 2018).

Apesar do grande número de usuários afetados, não são muitos os relatos acerca das consequências que a empresa sofreu em questões como perda de clientes ou perda de reputação. Apesar disso, dada as ações judiciais e ao valor financeiro estimado voltado para pagar essas ações, pode-se acreditar que o eBay passou sim por perda de credibilidade por parte de seus clientes.

4.4.6 Estratégias de mitigação

Para combater ataques de engenharia social como o *spear-phishsing*, o eBay deveria aumentar a conscientização acerca dos perigos de ataques como esse e educar continuamente seus funcionários e usuários (PARMAR, 2012).

Mesmo que a educação seja a principal ferramenta contra ataques de engenharia social, deve-se ter em mente que basta um funcionário ser vítima desse tipo de ataque para que toda a empresa esteja em risco de perdas em termos de credibilidade, confiabilidade e financeiros. Logo, a educação não é a única solução. A empresa também deveria aplicar estratégias de segurança *endpoint*, que é onde a maior parte dos dados de uma rede se encontra (PARMAR, 2012).

Além de educar seus funcionários, o eBay deveria ter criptografado os dados pessoais de seus usuários assim como fez com os financeiros, além de utilizar autenticações de dois fatores para criar uma barreira para ataques que invadem contas de funcionários (ROBERTS, 2018).

A maior lição deste caso é como é necessário ter uma cultura de segurança em todas as cadeias de uma empresa. Treinamentos e dinâmicas educacionais contra ataques

de engenharia social são técnicas eficazes de ensino para funcionários contra ataques deste tipo.

4.5 Equifax

A Equifax é uma agência de crédito dos Estados Unidos, responsável por coletar e manter informações de crédito sobre consumidores e empresas. A empresa armazena informações como histórico de pagamento de empréstimos, cartões de crédito, contas de serviços públicos e transações financeiras.

Em 2017 a Equifax sofreu um vazamento de dados que durou da metade de maio até julho e afetou mais de 145 milhões de pessoas nos Estados Unidos, no Reino Unido e no Canadá (Equifax, 2017).

Em julho daquele ano, a empresa descobriu que agentes maliciosos haviam explorado uma vulnerabilidade em um site para acessar determinados arquivos (Equifax, 2017).

4.5.1 Causas raiz

Os agentes maliciosos exploraram a vulnerabilidade Apache Struts CVE-2017-5638 (Equifax, 2017).

O Apache Struts é um framework web Java usado para construir aplicações web em larga escala, comumente utilizado em aplicações de governo, financeiras ou outras aplicações empresariais de grande porte (LUSZCZ, 2018).

Foi identificada nesse framework uma vulnerabilidade mapeada como CVE-2017-5638, onde um erro relacionado a um parser utilizado pelo framework pode ser explorado para executar código arbitrário (LUSZCZ, 2018). Esse framework contém uma biblioteca chamada *Object Graph Navigation Language* (OGNL), sendo essa a tecnologia atacada e explorada no caso.

O OGNL possuía um defeito relacionado às mensagens de erro de *parse* na funcionalidade padrão de *upload* de arquivos (LUSZCZ, 2018).

4.5.2 Métodos de ataque utilizados

A vulnerabilidade CVE-2017-5638 do Apache Struts foi explorada utilizando técnicas de injeções em cima da biblioteca OGNL. O OGNL é uma linguagem de expressão que permite a configuração de propriedades de objetos e a execução de métodos de classes Java, podendo ser explorado para ataques de execução de código remoto contra servidores Apache (LUSZCZ, 2018).

Os atacantes enviaram solicitações manipuladas para fazer *upload* de um arquivo para um servidor vulnerável que usava um *plug-in* baseado em um algorítimo específico (*Jakarta Multipart parser*) para processar a solicitação desse arquivo. Nesse processo, o atacante conseguia enviar um código malicioso no cabeçalho da requisição para executar comandos no servidor vulnerável (LUSZCZ, 2018).

Os agentes maliciosos passaram dois meses, desde o primeiro acesso, aumentando seus privilégios no sistema e explorando a rede da Equifax, até conseguirem começar a acessar arquivos com informações de identificação pessoal (PIIs) (WANG; JOHNSON, 2018).

4.5.3 Tipos de dados expostos

De acordo com a Equifax, os hackers tiveram acesso a (GRESSIN, 2017):

- Nomes;
- Números de Seguro Social;
- Datas de nascimento;
- Endereços;
- Informações de carteiras de motorista;
- Números de cartões de crédito.

4.5.4 Atitudes tomadas pela empresa

A Equifax contratou uma empresa de segurança cibernética independente para fazer revisões e determinar o alcance da intrusão e que dados foram afetados (Equifax, 2017).

A empresa emitiu uma notificação para seus clientes e construiu um site de assistência aos usuários afetados, explicando o vazamento, o que a empresa estava fazendo para contornar o problema e o que os usuários deveriam fazer para proteger suas informações.

A empresa disponibilizou um ano de monitoramento de crédito em suas plataformas, além de outros serviços gratuitos (GRESSIN, 2017).

A empresa desligou os responsáveis pelos cargos de Diretor de Informações e Diretor de Segurança, substituindo por membros que consideraram de mais confiança (Equifax, 2017).

4.5.5 Consequências

A Equifax perdeu clientes para concorrentes. Um concorrente em específico se beneficiou significativamente devido à violação de dados da Equifax, já que mais de 100 mil consumidores se cadastraram como clientes dessa concorrente dentro de uma semana do anúncio da violação da Equifax (DASWANI; ELBAYADI, 2021).

A empresa enfrentou ações judiciais movidas por consumidores que tiveram seus dados violados, sendo obrigada a pagar multas e restituir usuários (Equifax, 2020).

4.5.6 Estratégias de mitigação

Em março de 2017, a Equifax recebeu um comunicado de que havia uma correção para uma vulnerabilidade no *Apache Struts* e emitiu uma notificação interna solicitando a aplicação dessa correção. Porém, a empresa só aplicou essa correção em julho de 2017, o que deu tempo suficiente para que o ataque acontecesse (WANG; JOHNSON, 2018). A empresa poderia ter evitado o ataque caso tivesse aplicado a correção com mais velocidade.

A empresa estava utilizando mecanismos de segurança em final de vida, como o detector de vulnerabilidades do McAfee, além de certificados de segurança vencidos, o que impediu a empresa de detectar o problema com maior antecedência (DASWANI; ELBAYADI, 2021). Utilizando sistemas de detecção de ameaças e certificados de segurança atualizados, o ataque poderia ter sido detectado com maior antecedência, diminuindo a amplitude do vazamento.

Também é possível evidenciar a falta de contramedidas, como segmentação de rede, monitoramento de integridade de arquivos e proteção de credenciais de banco de dados. Características como estas não estavam em vigor, o que permitiu que os atacantes se movimentassem com quase nenhuma restrição pela rede da Equifax (DASWANI; ELBAYADI, 2021).

Por fim, pensando na causa raiz, que foi a vulnerabilidade no *parser* padrão do Struts, caso a empresa tivesse implementado seu próprio algoritmo de *parser*, os atacantes possivelmente falhariam na invasão da rede (LUSZCZ, 2018).

4.6 JPMorgan Chase

A JPMorgan Chase é uma empresa multinacional de serviços financeiros fundada nos Estados Unidos. A empresa opera em áreas como serviços bancários, gestão de ativos, banco de investimento e serviços comerciais.

Em 2014, a empresa sofreu um vazamento de dados onde nomes e e-mails de mais de 80 milhões de usuários foram roubados, por conta de invasões nas contas de mais de 90

funcionários. É interessante notar que a empresa gasta cerca de 250 milhões de dólares por ano em segurança. O ataque começou com um fornecedor de serviços que hospedava uma plataforma online de corridas beneficentes do banco (DASWANI; ELBAYADI, 2021).

4.6.1 Causas raiz

A violação tem início com uma entidade que organizou uma corrida beneficente do banco, que tinha como objetivo gerar receita para ser repassada para instituições de caridade locais. Para participar da corrida, era necessário se registrar no site do evento, que era hospedado pela Simmco Data Systems. O evento contou com vários participantes, incluindo funcionários da empresa.

Em abril de 2014, atacantes comprometeram o certificado do site da Simmco, quebrando a confiabilidade entre os visitantes do site e o site, o que permitiu que o tráfego entre os lados do cliente e do servidor fossem interceptados, incluindo credenciais de login feitas por funcionários da empresa. Um dos fatores chave para o ataque foi o fato de que muitos funcionários estavam usando as mesmas credenciais tanto para seus logins corporativos quanto para o site do evento (DASWANI; ELBAYADI, 2021).

Nem a Simmcon nem o JPMorgan Chase notaram que uma violação estava acontecendo em seus sistemas até que a Hold Security, uma empresa de segurança, descobriu um repositório online com mais de um bilhão de credenciais de login de mais de 400 mil sites, incluindo o Simmco Data Systems. Em agosto de 2014, a Hold Security entrou em contato com consultores de segurança do JPMorgan Chase informando que foram encontradas credenciais de login dos participantes do seu evento e o certificado da Simmco no repositório em questão. Nesse mesmo período, a equipe de segurança do JPMorgan Chase estava ciente de que a rede da empresa estava indicando um tráfego de rede incomum (DASWANI; ELBAYADI, 2021).

4.6.2 Métodos de ataque utilizados

Por quatro meses os agentes maliciosos testaram credenciais roubadas do ataque à Simmco em portais de login do JPMorgan Chase. Nesse mesmo período, a empresa estava implementando autenticação de dois fatores em alguns de seus servidores, para exigir que além de credenciais de login, um usuário precise também de um código de verificação de uso único para entrar no sistema. Entretanto, em um dos servidores onde essa atualização não tinha sido implantada, os atacantes conseguiram utilizar as credenciais roubadas para acessar a rede da empresa (DASWANI; ELBAYADI, 2021).

Além disso, os *hackers* apagaram arquivos de registro dos seus movimentos dentro da rede da empresa, fazendo uso de *malwares* específicos para essa finalidade (DASWANI; ELBAYADI, 2021).

4.6.3 Tipos de dados expostos

Os dados contidos nos servidores eram (DASWANI; ELBAYADI, 2021):

- Nomes;
- E-mails;
- Endereços residenciais;
- Números de telefone.

A JPMorgan Chase afirmou que a violação se limitou a informações pessoais e que nenhuma informação financeira foi comprometida.

4.6.4 Atitudes tomadas pela empresa

O Diretor de Operações da JPMorgan Chase e o Diretor de Segurança da Informação abriram investigações para rastrear as origens dos ataques e tentar identificar os atacantes, chegando a mais de 10 endereços IP de outros países. Também descobriram que esses mesmos endereços IP estavam em comunicação com a rede da empresa por meses (DASWANI; ELBAYADI, 2021).

A empresa também colaborou proativamente com órgãos públicos como o FBI para analisar a extensão do vazamento e rastrear os atacantes. A empresa identificou e resolveu as vulnerabilidades da sua rede, apesar de não informar publicamente como, e deu declarações afirmando que dobraria o orçamento de segurança da empresa para 500 milhões de dólares anuais (DASWANI; ELBAYADI, 2021).

4.6.5 Consequências

Assim que o vazamento foi divulgado, os dias seguintes foram de muitas reclamações e críticas em relação à empresa, evidenciando uma perda imediata da reputação da empresa (SYED; DHILLON, 2015).

O vazamento e todo seu desenrolar midiático obrigou a empresa a declarar que aumentaria o investimento em segurança em 250 milhões de dólares por ano (DASWANI; ELBAYADI, 2021). No caso do vazamento em questão, um incentivo na cultura de utilização segura de senhas e um pequeno investimento em segurança na estrutura do site do evento poderiam ter evitado o vazamento. Logo, a empresa passou a redirecionar anualmente uma quantidade excessiva de recursos por conta de um incidente que poderia ser resolvido com pouco investimento.

4.6.6 Estratégias de mitigação

Neste caso, o vazamento ocorreu através de uma aplicação de terceiros que apoiava uma necessidade da empresa e deveria ter sido tratada como se fosse uma ferramenta da empresa. Deveria ter sido exigido da plataforma externa o mesmo alto nível de segurança que era utilizado na empresa, para que funcionalidades externas não se tornem o ponto fraco na segurança de uma empresa (DASWANI; ELBAYADI, 2021).

Outro ponto de destaque neste vazamento foi que o mesmo se tornou possível pois os funcionários da empresa utilizaram as mesmas credenciais de acesso a rede corporativa para criar contas em sites de terceiros. Essa prática de reutilização de credenciais contribuiu para deixar a empresa exposta. Criar e incentivar a cultura de uso correto e seguro de senhas é uma maneira eficaz de evitar que credenciais da empresa sejam comprometidas. Além disso, fazer uso de autenticação de dois fatores em todos os servidores da empresa também garantiriam a segurança mesmo com o roubo de credenciais (DASWANI; ELBAYADI, 2021).

4.7 Target

A Target é uma empresa varejista dos Estados Unidos, atualmente a oitava maior do país.

A empresa foi vítima de vazamento de dados no final de 2013, onde *hackers* ucranianos invadiram a empresa por meio da Fazio Mechanical Services, empresa que gerenciava o sistema de aquecimento, ventilação e ar-condicionado de 1800 lojas da empresa. Os *hackers* roubaram mais de 40 milhões de números de cartão de crédito de clientes e informações pessoais de 70 milhões de clientes (DASWANI; ELBAYADI, 2021).

4.7.1 Causas raiz

Uma grande empresa de varejo como a Target trabalha com muitos fornecedores para manter seu negócio adequadamente funcionando. Clientes da Target podem acessar seu site principal para obter informações como o catálogo de produtos, como entrar em contato com suporte ao cliente e outros serviços. Porém, até 2013, além de fornecer informações aos clientes em seu site, a Target também disponibilizava informações para seus fornecedores em seu site público, como instruções sobre envio de faturas, ordens de serviço e pagamentos. A Target disponibilizava uma detalhada documentação interna para seus fornecedores em sites de acesso público sem qualquer autenticação ou autorização para acessar dados com essa sensibilidade. Esse site público levava ainda para páginas de gerenciamento de instalações da empresa, contendo listas completas de fornecedores da empresa (DASWANI; ELBAYADI, 2021).

Baixando essas listas de fornecedores era possível visualizar os metadados do arquivo. Metadados são dados que descrevem dados, por exemplo, uma foto tirada no celular contém, além do dado que é a foto, metadados como o local que a foto foi tirada, configurações da câmera no momento da foto, o tamanho da foto em bytes. Os metadados das listas de fornecedores da Target continham informações como data de criação, nome do usuário que editou o arquivo e nome da rede da empresa. Com uma simples pesquisa, era possível saber que o nome do usuário do arquivo estava atrelado a um funcionário da Target. Com tantas listas disponíveis e tantos metadados relacionados, os atacantes conseguiram juntar muita informação sobre a empresa e seus funcionários (DASWANI; ELBAYADI, 2021). A própria empresa se colocou em situação de vulnerabilidade deixando tantos dados públicos.

Além disso, quase todos os contratantes da Target utilizavam o mesmo sistema de faturamento, sistema esse que permitia que os contratantes fizesse o *upload* de faturas para que pudessem acompanhar o trabalho e receber o pagamento da empresa (DASWANI; ELBAYADI, 2021).

4.7.2 Métodos de ataque utilizados

Utilizando dos dados de fornecedores da empresa, agentes maliciosos aplicaram um ataque de *malware* via e-mail contra a Fazio Mechanical Services, um dos fornecedores mais simples da Target (DASWANI; ELBAYADI, 2021). O ataque se trata do envio de e-mails para vítimas contendo links ou anexos com *malware*, que são programas com objetivo de danificar ou recuperar de forma ilícita informações de um computador. Caso a vítima abra esses links ou anexos, o *malware* é baixado e executado.

O ataque foi bem sucedido em algum dos funcionários, e o *malware Citadel* foi baixado no computador da vítima. O *Citadel* é um programa robô que rouba senhas, e, estando presente no computador do funcionário, só foi necessário que o mesmo fizesse login na rede da Target uma vez. Ao interceptar o login, o *Citadel* recuperou as credenciais de acesso do funcionário. Com essas credenciais, os atacantes conseguiram fazer login na rede da empresa (DASWANI; ELBAYADI, 2021).

Investigações sobre o ataque sugerem que os agentes maliciosos aproveitaram de uma falha no sistema de faturamento da Target e fizeram o *upload* de um arquivo PHP executável que permitiu que fossem executados comandos quaisquer. Os atacantes conseguiram entrar na rede da Target e obter acesso a tokens de hash de administradores da rede. Conectados como administradores, os atacantes criaram sua própria conta de administrador e tiveram total acesso na rede da empresa (DASWANI; ELBAYADI, 2021).

Tendo acesso aos sistemas de venda da Target, os atacantes instalaram uma ferramenta de *scrapping* de memória nesses pontos. A ferramenta utilizada foi o BlackPOS,

que registrava os números de cartão de crédito presentes na memória dos pontos de venda da empresa (DASWANI; ELBAYADI, 2021).

4.7.3 Tipos de dados expostos

Os dados roubados dos servidores da Target incluíam (DASWANI; ELBAYADI, 2021):

- Números de cartão de crédito;
- Registros de transações de compras;
- Nomes de clientes;
- E-mails;
- Números de telefone;
- Endereços físicos.

4.7.4 Atitudes tomadas pela empresa

A Target possuía mecanismos de defesa em vigor como ferramentas *anti-malware* e monitoração especializada da rede. Esses mecanismos acusaram problemas e a equipe de monitoramento informou a sede da empresa.

Porém, o aviso enviado foi de um *malware* genérico, e por ser enviado justamente no dia da *Black Friday*, a empresa não deu a atenção devida para o problema. Uma porta-voz da Target chegou a mencionar que foi detectado um acesso estranho na rede, mas por terem poucos dados de atividade registrados e detectados, a empresa optou por não fazer um acompanhamento imediato (FINKLE; HEAVEY, 2014).

O ataque começou na metade de novembro de 2013, um mês depois, a Target percebeu que sua rede havia sido invadida e divulgou a notícia sobre a invasão. Poucos dias depois de descobrir o vazamento de dados, a empresa contratou especialistas em segurança da Verizon para fazer uma auditoria em sua rede.

Após o vazamento e todo o julgamento que se seguiu, a empresa investiu em melhorias de segurança em sua rede como (DASWANI; ELBAYADI, 2021):

- Melhoria de monitoramento e registros;
- Implantação de listas de aplicativos que podem ser executados em seus pontos de venda;
- Segmentação da rede;

- Limitação em acesso de documentos.

A empresa investiu centenas de milhões de dólares em segurança e pediu para a Verizon auditar sua rede novamente, onde a mesma confirmou que a rede da Target passou a ser mais robusta e menos sucetível a vazamento de dados (DASWANI; ELBAYADI, 2021).

4.7.5 Consequências

Os clientes da empresa tiveram seus cartões de créditos expostos e sofreram prejuízos financeiros por conta de compras grandes que não haviam feito e até mesmo zeramento de contas bancárias (DASWANI; ELBAYADI, 2021).

A Target foi responsabilizada pela violação e foi obrigada a pagar reparações aos clientes afetados. Além de restituições financeiras referentes ao prejuízo individual de cada pessoa afetada, a empresa também foi obrigada a pagar pelos custos de reemissões de cartões. A Visa, por exemplo, recebeu 67 milhões de dólares da Target como parte de um acordo de conciliação. A empresa também fechou acordo de 10 milhões de dólares em uma ação coletiva.

Ao final, estima-se que o vazamento de dados custou mais de 250 milhões de dólares para a empresa. As vendas em dezembro, quando a invasão foi divulgada, caíram em 4%. Cargos de liderança da empresa como CEO foram demitidos e substituídos.

4.7.6 Estratégias de mitigação

Informações devem ser fornecidas apenas com base no mínimo que é necessário que determinado público ou perfil de acesso saiba, o mínimo necessário para que sejam realizados seus trabalhos (DASWANI; ELBAYADI, 2021). Neste caso em específico, não havia necessidade de que qualquer pessoa com acesso a internet soubesse ou tivesse acesso à lista de todos os fornecedores da empresa.

O acesso não autorizado em uma ferramenta específica da empresa permitiu acesso a toda rede da Target. Segmentar a rede da empresa em zonas e com diferentes níveis de confiança teria impedido que os atacantes tivessem total acesso a rede atacando apenas um ponto específico (DASWANI; ELBAYADI, 2021).

5 Análise Comparativa

5.1 Considerações Iniciais do Capítulo

Tendo levantado casos de empresas reconhecidas no cenário global e evidenciado casos de vazamento detalhando os mesmos, esta Seção tem o objetivo de trazer comparações entre os casos, buscando encontrar pontos em comum entre eles, sejam nas causas raiz ou nos métodos de ataque utilizados, para que seja possível identificar quais as características que são exploradas com maior frequência em ataques de aplicações. Além disso, aqui também serão referenciadas estratégias de mitigação para cada caso, baseadas no material levantado no Capítulo 2.

5.2 Comparações

5.2.1 LinkedIn

5.2.1.1 Problemas

- Dados sensíveis descriptografados;
- Abdicação deliberada da privacidade dos dados dos usuários em uma API;
- Abuso de API utilizando *scrapping*;

5.2.1.2 Soluções

- Implementar autenticação e autorização no uso da API;
- Monitorar a API, cortando a disponibilização dos dados caso algo de estranho seja observado;
- Estabelecer limite de dados que podem ser obtidos pela API de uma só vez.

5.2.1.3 Mitigação baseada no referencial teórico

O abuso de API realizado no LinkedIn poderia ter sido evitado com a aplicação da metodologia de desenvolvimento seguro da Microsoft, o SDL (Microsoft, 2023b), onde as seguintes práticas e contextos do problema poderiam ser seguidas:

- Definição de requisitos de segurança: A funcionalidade da API que liberava as informações dos usuários da empresa foi desenvolvida posteriormente em relação ao

core da aplicação, sendo assim, caso requisitos de segurança claros fossem estabelecidos e seguidos desde o início do projeto, esta funcionalidade, da forma que foi desenvolvida, não passaria nesses requisitos.

- Modelagem de ameaças: Para o desenvolvimento de uma funcionalidade que tem como objetivo dar acesso público a dados, seria necessário considerar as motivações de atacantes, e que o interesse desse tipo de agente seria despertado pelas funcionalidades dessa API, e nesse momento, pensar a respeito dos limites de confiança e do fluxo de dados da aplicação.

Além das práticas da metodologia da Microsoft, pontos levantados pela OWASP (The OWASP Foundation, 2021b) em relação à prevenção de exposição de dados sensíveis também poderiam ter sido aplicados nesse caso, como a classificação dos dados processados, identificando dados sensíveis de acordo com leis de privacidade. Como os dados vazados se classificam como informações pessoais identificáveis (PIIs), os mesmos deviam ser classificados como sensíveis, e por consequência, deveriam estar criptografados.

Dada a classificação dos dados, uma outra discussão é se esses dados deveriam estar sendo divulgados ao público dessa forma. Mesmo que uma funcionalidade devolvesse esses dados, era esperado ao menos, segundo as recomendações da OWASP, que um processo de autenticação fosse implementado para consumir os dados, de forma criptografada.

5.2.2 Yahoo

5.2.2.1 Problemas

- Dados sensíveis descriptografados;
- *Phishing*;
- Uso de *malwares*;
- Vulnerabilidade de software. (Falsificação de *cookies*).

5.2.2.2 Soluções

- Treinamento com os funcionários;
- Monitoramento do sistema;
- Uso de criptografia na geração de *cookies*.

5.2.2.3 Mitigação baseada no referencial teórico

Casos que envolvem *phishing* tem como principal ponto a falta de treinamento ou cultura de segurança da empresa. A metodologia SDL (Microsoft, 2023b) traz a disponibilização de treinamento como a primeira prática para o desenvolvimento seguro de produtos. Além de cursos e treinamentos que atualizem o conhecimento técnico dos funcionários, é necessário treinar os funcionários a reconhecerem táticas de engenharia social, de forma que os mesmos não sejam pontos de vulnerabilidade dentro da empresa.

Em relação ao uso de *cookies* forjados, uma modelagem de ameaças, também proposta pelo SDL, poderia ter levado a empresa a identificar que existia uma fraqueza no sistema, onde um vazamento do seu código-fonte era o suficiente para comprometer uma funcionalidade tão ligada a autenticação no sistema. Para este mesmo problema, também deveria ter sido utilizado criptografia na chave geradora dos *cookies*, como recomenda tanto o SDL quanto a OWASP (The OWASP Foundation, 2021b).

Como este caso envolve uso de *malwares*, era crucial que testes de segurança periódicos, tanto estáticos quanto dinâmicos fossem executados no sistema (Microsoft, 2023b). A execução regular de testes poderia ter alertado o time de segurança sobre a presença de programas maliciosos em seus sistemas, provocando a empresa a investigar o estado da aplicação.

Um ponto muito relevante neste caso foi o tempo necessário para que a empresa descobrisse a invasão e para divulgar ao público, tanto que "negligência" é uma das palavras-chave que descrevem esse caso. O SDL recomenda a prática do estabelecimento de processos padrão de resposta a incidentes (Microsoft, 2023b), onde a equipe de segurança deve estar preparada para ataques, sabendo que protocolos devem ser seguidos em casos como este, mitigando vulnerabilidades e comunicando rapidamente seus clientes.

5.2.3 eBay

5.2.3.1 Problemas

- Dados sensíveis descriptografados;
- *Phishing* (*Spear-phishing*);
- Uso de *malwares*;

5.2.3.2 Soluções

- Treinamento com os funcionários;
- Fazer uso de criptografia em PIIs.

5.2.3.3 Mitigação baseada no referencial teórico

O eBay passou por um ataque mais sofisticado de *phishing*, o *spear-phishing*, onde, estando o atacante munido de informações específicas sobre seu alvo, ele consegue elaborar um ataque mais direcionado, e tendo informações da empresa e das possíveis vítimas, consegue trabalhar em um ataque mais convincente (PARMAR, 2012). Um detalhe interessante desse tipo de ataque é que para conseguir as informações específicas das vítimas para executar o *spear-phishing*, uma violação de dados anterior deve ter acontecido, evidenciando os perigos e consequências da exposição de dados pessoais identificáveis, onde um vazamento de dados pode ser o ponto de partida de outro ataque.

Assim como em outros casos envolvendo *phishing*, o treinamento de funcionários e a implantação de uma cultura de segurança na empresa são práticas que mitigam e evitam vazamentos de dados causados por métodos de ataque que envolvem engenharia social. Além do *phishing*, o uso de *malwares* também se assemelha ao caso anterior, onde o monitoramento do sistema e a realização periódica de testes de segurança poderiam ter mitigado os efeitos do ataque, como recomendado pelo SDL (Microsoft, 2023b).

Após descobrir o vazamento de dados em seu sistema, o eBay levou dois meses para conseguir tomar medidas de resolução e mitigação do problema, evidenciando a falta de estabelecimento de processos padrão de resposta a incidentes, como também recomenda a metodologia de desenvolvimento seguro da Microsoft (Microsoft, 2023b), onde além da demora, não foi capaz de propor processos de mitigação eficientes, se limitando a instruir os clientes a alterar suas credenciais de login (HOLM; MACKENZIE, 2014).

Por fim, este é mais um caso onde dados pessoais identificáveis não são corretamente classificados como sensíveis, mantendo-os descriptografados, indo contra as recomendações da OWASP no que se refere a exposição de dados sensíveis (The OWASP Foundation, 2021b).

5.2.4 Equifax

5.2.4.1 Problemas

- Dados sensíveis descriptografados;
- Vulnerabilidade de software (No *framework* utilizado);
- Falhas de administração de sistema (Uso de mecanismos de segurança obsoletos);
- Rede centralizada;

5.2.4.2 Soluções

- Manter as bibliotecas utilizadas sempre atualizadas em questões de segurança;

- Utilizar mecanismos de segurança atualizados.
- Segmentação de rede.
- Implementação de algorítimos próprios.

5.2.4.3 Mitigação baseada no referencial teórico

Dos casos estudados, a Equifax foi a que mais forneceu informações a seus clientes, além de propor respostas rápidas a partir do momento de descoberta do vazamento dos dados, onde isso foi possível graças a ajuda de uma empresa de segurança especializada contratada (Equifax, 2017).

Apesar disso, a empresa poderia ter tomado ainda outras atitudes, baseadas no protocolo SDL (Microsoft, 2023b) ou nas recomendações da OWASP (The OWASP Foundation, 2021b) que teriam evitado ou mitigado o vazamento de informações.

Nesse caso, onde ocorreu a exploração de uma vulnerabilidade na biblioteca utilizada pela aplicação, a realização periódica de testes de segurança estáticos e dinâmicos poderiam antecipar a identificação do problema, permitindo diminuir a quantidade de dados vazados. Além disso, a equipe poderia ter investido na realização de testes de penetração, buscando por possíveis vulnerabilidades e se protegendo das encontradas.

Neste ataque, os agentes maliciosos conseguiram, além de obter acesso a dados pessoais identificáveis, obter acesso a números de cartões de crédito descriptografados. Tanto o SDL quanto a OWASP recomendam, com bastante importância, que dados de tamanha sensibilidades estejam sempre criptografados, seja em seu repouso no banco de dados ou em qualquer tipo de tráfego. Esse tipo de exposição pode causar grandes prejuízos financeiros nas vítimas, demonstrando o nível da irresponsabilidade por parte da empresa em não proteger corretamente dados com tamanha sensibilidade.

Por fim, cabe destacar ainda que a empresa tanto atrasou a aplicação de uma correção para a vulnerabilidade na biblioteca utilizada como também estava utilizando mecanismos de segurança vencidos ou em final de vida, indo contra a prática de utilizar ferramentas aprovadas e atualizadas do SDL (Microsoft, 2023b) e a recomendação da OWASP (The OWASP Foundation, 2021b).

5.2.5 JPMorgan Chase

5.2.5.1 Problemas

- Dados sensíveis descriptografados;
- Comprometimento em ferramentas de terceiros;

- Falhas de funcionários no sistema (Uso de credenciais repetidas em diferentes sistemas);
- Uso de *malwares*.

5.2.5.2 Soluções

- Implementação de regras de segurança mesmo em ferramentas de terceiros;
- Treinamento de funcionários;
- Autenticação em dois fatores.

5.2.5.3 Mitigação baseada no referencial teórico

As principais causas raiz desse caso envolvem falta de cultura de segurança por parte dos funcionários da empresa e o uso de produtos de terceiros em seus projetos. Assim como em outros casos, fornecer cursos e treinamentos de segurança para os funcionários de uma empresa os torna mais preparados para fazerem parte da organização (Microsoft, 2023b) sem se tornarem pontos fracos da mesma.

Já em relação ao uso de ferramentas de terceiros, o SDL possui duas práticas recomendadas relacionadas (Microsoft, 2023b), onde a primeira recomenda o gerenciamento do risco de segurança do uso de componentes de terceiros, onde a equipe de desenvolvimento deve fazer uma gestão minuciosa das ferramentas externas, realizando varreduras constantes e planejar planos de resposta para vulnerabilidades. A outra prática recomendada pelo SDL é utilizar apenas ferramentas aprovadas, onde toda ferramenta deve passar por uma análise da equipe de segurança e projetar verificações nas ferramentas.

5.2.6 Target

5.2.6.1 Problemas

- Dados sensíveis descriptografados;
- Falhas de administração de sistema (Divulgação desnecessária de informações);
- *Phishing*;
- Comprometimento em ferramentas de terceiros;
- Rede centralizada;
- Uso de *malwares*.

5.2.6.2 Soluções

- Expor apenas informações necessárias;
- Segmentação de rede;

5.2.6.3 Mitigação baseada no referencial teórico

Neste caso, o problema estava na forma em que a empresa pensou o sistema, não identificando os males que uma exposição pública de certos arquivos em sua aplicação poderiam causar caso um agente malicioso tivesse acesso a eles. Tendo como base as recomendações da metodologia de desenvolvimento seguro da Microsoft (Microsoft, 2023b), é possível identificar algumas práticas que poderiam ter evitado o vazamento de dados.

Primeiramente, faltou realizar uma modelagem de possíveis ameaças, além de testes de penetração, considerando o que atacantes poderiam explorar a partir das informações públicas do sistema. Também é possível identificar a falta de estabelecimento de requisitos de projeto, no que se refere a desenvolver recursos bem projetados em termos de segurança, já que o sistema permitiu que o upload de arquivos indevidos fosse realizado em sua rede. O fato de toda a rede da empresa estar interligada, permitindo que um acesso não autorizado em um ponto do sistema permitisse acesso a todos os módulos da empresa evidencia pouco esforço gasto na definição de requisitos de segurança, onde não foi considerado segmentar a rede da empresa em módulos separados, para evitar o comprometimento total da organização.

Um ponto relevante é que a partir da exposição desnecessária de informações, os agentes maliciosos conseguiram identificar possíveis vítimas para um ataque de *phishing*, seria possível então ter seguido mais uma prática do SDL e fornecer treinamento para os clientes da empresa, os capacitando para identificar e se prevenir de ataques de engenharia social.

5.2.7 Consolidação

A Tabela 2 agrupa os problemas buscando sumarizar e apontar os problemas mais frequentes, além de categorizá-las quanto a sua questão principal.

Problema	Qtd.	Categoria
Dados sensíveis descriptografados	6	Questão humana
Uso de *malwares*	4	Questão tecnológica
Falhas de administração/funcionários no sistema	3	Questão humana
Phishing	3	Questão humana
Comprometimento em ferramentas de terceiros	2	Questão arquitetural
Rede centralizada	2	Questão arquitetural
Vulnerabilidade de software	2	Questão tecnológica
Abdicação deliberada de dados	1	Questão humana
Abuso de funcionalidade	1	Questão tecnológica

Tabela 2 – Consolidação dos problemas

A coluna “Categoria” tem o objetivo de identificar a questão principal que rodeia o problema da linha, onde “humana” significa que o problema está relacionado a decisões tomadas por uma pessoa ou ataques de engenharia social, ou seja, que envolvem pessoas. Uma questão “tecnológica” se refere a problemas relacionados a um conhecimento técnico que não possui influência de atores “não atacantes”, como usuários ou desenvolvedores. São problemas onde foi necessário possuir um conhecimento mais avançado em software para realizar o ataque. Por fim, “arquitetural” se refere a problemas onde tomadas de decisão sobre a arquitetura do projeto na fase de desenvolvimento levaram a possibilidade do ataque, como decisões de que ferramentas de terceiros utilizar ou como distribuir os subsistemas de um projeto na rede.

É importante notar que na tabela acima, o problema “Dados sensíveis descriptografados” está categorizado como “questão humana” pois o dados não foram criptografados por uma escolha dos desenvolvedores. Em todos os casos analisados, apenas os dados descriptografados foram expostos, não havendo em nenhum dos casos uma quebra de criptografia, o que evidencia a importância de classificar dados sensíveis corretamente e criptografá-los, além de mostrar a força que a criptografia de dados possui.

Enquanto que os problemas “Uso de *malwares*”, “Vulnerabilidade de software” e “Abuso de funcionalidade” estão mais intimamente ligados a questões tecnológicas, os problemas “Falhas de administração/funcionários no sistema”, “*Phishing*” e “Abdicação deliberada de dados” estão ligados a engenharia social ou deslizes de natureza básica na administração de um sistema.

A partir do agrupamento dos dados coletados, percebe-se que a maioria dos casos de vazamento de dados está diretamente relacionada ou a falhas técnicas que a primeira vista não são simples de se notar ou a situações onde não havia nenhuma ou pouca cultura relacionada a aspectos de segurança sobre como se portar dentro de um sistema de informações ou sobre como administrar um.

5.3 Considerações Finais do Capítulo

Baseado na análise comparativa realizada, chega-se a conclusão de que é necessário monitorar redes e sistemas além de estar atento a possíveis vulnerabilidades, mas também é de grande necessidade promover um treinamento entre os funcionários de uma empresa que lida com dados sensíveis sobre que práticas e cuidados devem ser tomados para evitar a exposição de suas informações.

Analisar cada um dos casos individualmente mais uma vez neste capítulo, mas buscando relacioná-lo com o referencial teórico levantado permitiu explorar ainda mais cada um dos casos, identificando novas maneiras de mitigá-los e identificando novas formas de relacionar causas raiz comuns com outras soluções. Com isso, ao final do referencial teórico levantado, dos casos explorados e analisados e a análise comparativa realizada neste capítulo, foi possível propor um guia de medidas de segurança, apresentado no Apêndice A.1, consolidando todo o trabalho realizado até aqui.

6 Conclusão

Ao longo deste trabalho, foi dedicada uma atenção à análise aprofundada de exposições de dados sensíveis e falhas de criptografia, explorando casos reais que evidenciaram a vulnerabilidade crítica desses incidentes. O objetivo foi desenvolver um entendimento completo dessas ameaças, não apenas destacando as causas raiz, impactos e consequências, mas também propondo soluções tangíveis e práticas para fortalecer a segurança da informação.

Foi examinado de perto os pilares fundamentais da segurança de software, práticas de desenvolvimento web, recomendações da OWASP, princípios de criptografia e metodologias de desenvolvimento seguro. Esta base teórica foi essencial para fundamentar as análises e fornecer a estrutura necessária para a construção de medidas de segurança eficazes.

A pesquisa de casos reais proporcionou uma visão valiosa sobre a complexidade e as consequências das exposições de dados sensíveis e falhas de criptografia. A identificação de causas raiz, impactos nos agentes relacionados e as consequências legais e reputacionais, ilustram claramente a necessidade de abordagens robustas de segurança.

Ao propor um guia prático de desenvolvimento seguro, o objetivo foi realizar algo a mais do que apenas documentar problemas, buscando soluções palpáveis para eles. Cada medida sugerida tem como base a experiência adquirida a partir dos casos analisados e a aplicação direta dos princípios levantados no referencial teórico.

A análise comparativa entre os casos estudados fazendo uso do referencial teórico destacou a relevância e a aplicabilidade dos conceitos teóricos. Observou-se que as vulnerabilidades expostas nos casos poderiam ter sido mitigadas ou prevenidas através da implementação aplicada de práticas de segurança bem estabelecidas.

Conclui-se que a segurança de dados sensíveis não é apenas uma preocupação técnica, mas uma responsabilidade ética e legal. A integridade e confidencialidade dos dados são essenciais para a confiança nas transações digitais e para a preservação da privacidade individual. O guia proposto visa capacitar desenvolvedores e profissionais de segurança a adotarem medidas proativas, buscando criar um ambiente digital mais seguro.

Os resultados deste trabalho representam um ponto de partida. A rápida evolução do cenário cibernético exige respostas contínuas e adaptativas. Também representam um chamado aos profissionais da área a se comprometerem com a aprendizagem contínua, a integração de práticas de segurança em seus fluxos de trabalho e a contribuição para a construção de um ecossistema digital mais seguro e confiável.

Espera-se que este trabalho possa servir de inspiração para novas pesquisas, discussões e principalmente ações concretas para proteger a integridade dos dados sensíveis na atual era digital.

Referências

ALKHALIL, Z. et al. Phishing Attacks: A Recent Comprehensive Study and a New Anatomy. *Frontiers in Computer Science*, v. 3, 2021. ISSN 2624-9898. Disponível em: <https://www.frontiersin.org/articles/10.3389/fcomp.2021.563060>. Citado na página 83.

AOKI, E. K.; CARVALHO, A. H. P. d. Práticas de segurança para o desenvolvimento de sistemas Web. *FaSCi-Tech*, v. 1, n. 5, set. 2016. ISSN 2176-9427. Disponível em: <https://www.fatecsaocaetano.edu.br/fascitech/index.php/fascitech/article/view/47>. Citado na página 15.

ATKINSON, C. *Verizon wants $1B discount on Yahoo deal after reports of hacking, spying*. 2016. Disponível em: <https://nypost.com/2016/10/06/verizon-wants-1b-discount-on-yahoo-deal-after-hacking-reports/>. Citado na página 42.

BEJTLICH, R. *The Practice of Network Security Monitoring: Understanding Incident Detection and Response*. [S.l.]: No Starch Press, 2013. Google-Books-ID: QdLclhJhQecC. ISBN 978-1-59327-509-9. Citado na página 81.

BISHOP, M. About Penetration Testing. *IEEE Security & Privacy*, v. 5, n. 6, p. 84–87, nov. 2007. ISSN 1558-4046. Conference Name: IEEE Security & Privacy. Disponível em: <https://ieeexplore.ieee.org/abstract/document/4402456>. Citado 2 vezes nas páginas 95 e 96.

BOBROVSKIS, S.; JURENOKS, A. A Survey of Continuous Integration, Continuous Delivery and Continuous Deployment. 2018. Citado na página 95.

BRASIL. *L13709*. 2018. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm>. Citado na página 24.

CADWALLADR, C.; GRAHAM-HARRISON, E. Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach. *The guardian*, v. 17, n. 1, p. 22, 2018. Citado na página 16.

CANONGIA, C.; JUNIOR, R. M. Segurança cibernética: o desafio da nova Sociedade da Informação. *Parcerias Estratégicas*, v. 14, n. 29, 2009. Disponível em: <http://cdi.mecon.gov.ar/bases/doc/parceriasest/29.pdf#page=23>. Citado na página 76.

CHAKKARAVARTHY, S. S.; SANGEETHA, D.; VAIDEHI, V. A Survey on malware analysis and mitigation techniques. *Computer Science Review*, v. 32, p. 1–23, maio 2019. ISSN 1574-0137. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1574013718301114>. Citado na página 43.

COLóN, M. *European data authorities to probe eBay data breach*. 2014. Disponível em: <https://www.scmagazine.com/brief/breach/european-data-authorities-to-probe-ebay-data-breach>. Citado na página 16.

Cybernews Team. *LinkedIn Data Breach - 500M Records Leaked and Being Sold*. 2021. Disponível em: <https://cybernews.com/news/stolen-data-of-500-million-linkedin-users-being-sold-online-2-million-leaked-as-proof-2/>. Citado 2 vezes nas páginas 35 e 36.

DASWANI, N.; ELBAYADI, M. *Big Breaches: Cybersecurity Lessons for Everyone*. Berkeley, CA: Apress, 2021. ISBN 978-1-4842-6654-0 978-1-4842-6655-7. Disponível em: <https://link.springer.com/10.1007/978-1-4842-6655-7>. Citado 32 vezes nas páginas 40, 41, 42, 43, 48, 49, 50, 51, 52, 53, 54, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96 e 97.

DENCHEVA, L. *Comparative analysis of Static application security testing (SAST) and Dynamic application security testing (DAST) by using open-source web application penetration testing tools*. Dissertação (Mestrado) — Dublin, National College of Ireland, ago. 2022. Disponível em: <https://norma.ncirl.ie/5956/>. Citado na página 95.

DO, C. T. et al. Game Theory for Cyber Security and Privacy. *ACM Computing Surveys*, v. 50, n. 2, p. 30:1–30:37, 2017. ISSN 0360-0300. Disponível em: <https://dl.acm.org/doi/10.1145/3057268>. Citado na página 16.

EGELE, M. et al. An Empirical Study of Cryptographic Misuse in Android Applications. In: *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*. New York, NY, USA: Association for Computing Machinery, 2013. (CCS '13), p. 73–84. ISBN 978-1-4503-2477-9. Event-place: Berlin, Germany. Disponível em: <https://doi.org/10.1145/2508859.2516693>. Citado na página 21.

EMINAğAOğLU, M.; UçAR, E.; EREN, The positive outcomes of information security awareness training in companies – A case study. *Information Security Technical Report*, v. 14, n. 4, p. 223–229, nov. 2009. ISSN 1363-4127. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1363412710000099>. Citado na página 82.

Equifax. *Consumer Notice*. 2017. Disponível em: <https://www.equifaxsecurity2017.com/consumer-notice>. Citado 3 vezes nas páginas 46, 47 e 59.

Equifax. *Equifax Data Breach Settlement | Am I Affected?* 2020. Disponível em: <https://www.equifaxbreachsettlement.com/>. Citado na página 48.

FAHL, S. The Cyber Security Body of Knowledge v1.1.0, 2021. In: . University of Bristol, 2021. Section: Web & Mobile Security. Disponível em: <https://www.cybok.org/>. Citado 4 vezes nas páginas 15, 22, 23 e 24.

FARUK, M. J. H. et al. Malware Detection and Prevention using Artificial Intelligence Techniques. In: *2021 IEEE International Conference on Big Data (Big Data)*. [s.n.], 2021. p. 5369–5377. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9671434>. Citado na página 80.

FIEGERMAN, S. *Yahoo says 500 million accounts stolen*. 2016. Disponível em: <https://money.cnn.com/2016/09/22/technology/yahoo-data-breach/>. Citado na página 16.

FINKLE, J.; HEAVEY, S. Target says it declined to act on early alert of cyber breach. *Reuters*, mar. 2014. Disponível em: <https://www.reuters.com/article/target-breach-idINDEEA2C0LV20140313>. Citado na página 53.

GIBSON, B. et al. Vulnerability in Massive API Scraping: 2021 LinkedIn Data Breach. In: *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*. [S.l.: s.n.], 2021. p. 777–782. Citado 6 vezes nas páginas 35, 36, 37, 38, 39 e 77.

GRESSIN, S. The equifax data breach: What to do. *Federal Trade Commission*, v. 8, 2017. Disponível em: <https://www.penncommunitybank.com/wp-content/uploads/2019/12/The-Equifax-Data-Breach_-What-to-Do-_-Consumer-Information.pdf>. Citado na página 47.

HACKETT, R. *AOL CEO Tim Armstrong 'Cautiously Optimistic' About Verizon Closing the Yahoo Deal*. 2016. Disponível em: <https://fortune.com/2016/12/06/yahoo-verizon-aol-ceo-tim-armstrong-optimistic/>. Citado na página 42.

HASSAN, W. U.; BATES, A.; MARINO, D. Tactical Provenance Analysis for Endpoint Detection and Response Systems. In: *2020 IEEE Symposium on Security and Privacy (SP)*. [s.n.], 2020. p. 1172–1189. ISSN: 2375-1207. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9152771>. Citado na página 81.

HOLM, E.; MACKENZIE, G. The importance of mandatory data breach notification to identity crime. In: *2014 Third International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*. Beirut, Lebanon: IEEE, 2014. p. 6–11. ISBN 978-1-4799-3906-0 978-1-4799-3905-3. Disponível em: <http://ieeexplore.ieee.org/document/6913963/>. Citado 2 vezes nas páginas 45 e 58.

HOLM, H. et al. A quantitative evaluation of vulnerability scanning. *Information Management & Computer Security*, v. 19, n. 4, p. 231–247, jan. 2011. ISSN 0968-5227. Publisher: Emerald Group Publishing Limited. Disponível em: <https://doi.org/10.1108/09685221111173058>. Citado na página 96.

HU, J.; WANG, H.; LIU, Y. Strengthening Digital Marketing Security Website Threat Isolation and Protection Using Remote Browser Isolation Technology. *Computer-Aided Design and Applications*, p. 56–74, nov. 2023. ISSN 16864360. Disponível em: <https://cad-journal.net/files/vol_21/Vol21NoS4.html>. Citado 2 vezes nas páginas 81 e 82.

IBM. *Cost of a data breach 2022*. 2023. Disponível em: <https://www.ibm.com/reports/data-breach>. Citado 2 vezes nas páginas 15 e 16.

Imperva. *What is Personally Identifiable Information | PII Data Security | Imperva*. 2023. Disponível em: <https://www.imperva.com/learn/data-security/personally-identifiable-information-pii/>. Citado na página 23.

IMTIAZ, N.; THORN, S.; WILLIAMS, L. A comparative study of vulnerability reporting by software composition analysis tools. In: *Proceedings of the 15th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. New York, NY, USA: Association for Computing Machinery, 2021. (ESEM '21), p. 1–11. ISBN 978-1-4503-8665-4. Disponível em: <https://doi.org/10.1145/3475716.3475769>. Citado na página 94.

JOHNSON, T. *What are the implications of LinkedIn's latest data breach?* 2021. Disponível em: <https://omdia.tech.informa.com/blogs/2021/what-are-the-implications-of-linkedins-latest-data-breach>. Citado na página 38.

KARA, I. A basic malware analysis method. *Computer Fraud & Security*, v. 2019, n. 6, p. 11–19, jun. 2019. ISSN 1361-3723. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1361372319300648>. Citado na página 80.

KOLEVSKI, D. et al. Cloud Data Breach Disclosures: the Consumer and their Personally Identifiable Information (PII)? In: *2021 IEEE Conference on Norbert Wiener in the 21st Century (21CW)*. [S.l.: s.n.], 2021. p. 1–9. ISSN: 2643-4482. Citado 2 vezes nas páginas 44 e 45.

LAZAROVITZ, L. Deconstructing the SolarWinds breach. *Computer Fraud & Security*, v. 2021, n. 6, p. 17–19, jan. 2021. ISSN 1361-3723. Publisher: Elsevier. Disponível em: <https://www.magonlinelibrary.com/doi/abs/10.1016/S1361-3723%2821%2900065-8>. Citado na página 91.

LEUNG, H.; WHITE, L. A study of integration testing and software regression at the integration level. In: *Proceedings. Conference on Software Maintenance 1990*. [s.n.], 1990. p. 290–301. Disponível em: <https://ieeexplore.ieee.org/abstract/document/131377>. Citado na página 95.

LI, J. *Vulnerabilities Mapping based on OWASP-SANS: a Survey for Static Application Security Testing (SAST)*. 2020. Disponível em: <https://arxiv.org/abs/2004.03216v2>. Citado na página 94.

LinkedIn. *An update on report of scraped data*. 2021. Disponível em: <https://news.linkedin.com/2021/june/an-update-from-linkedin>. Citado na página 37.

LinkedIn. *Sobre o LinkedIn*. 2023. Disponível em: <https://about.linkedin.com/pt-br>. Citado na página 35.

LUSZCZ, J. Apache Struts 2: how technical and development gaps caused the Equifax Breach. *Network Security*, v. 2018, n. 1, p. 5–8, jan. 2018. ISSN 1353-4858, 1872-9371. Disponível em: <http://www.magonlinelibrary.com/doi/10.1016/S1353-4858%2818%2930005-9>. Citado 4 vezes nas páginas 46, 47, 48 e 77.

Malwarebytes Labs. *Second colossal LinkedIn "breach"in 3 months, almost all users affected*. 2021. Disponível em: <https://www.malwarebytes.com/blog/news/2021/06/second-colossal-linkedin-breach-in-3-months-almost-all-users-affected>. Citado na página 39.

MCGRAW, G. Software security. *Building security in*, 2006. Citado na página 94.

MCGRAW, G. Silver Bullet Talks with the IEEE Center for Secure Design. *IEEE Security & Privacy*, v. 12, n. 6, p. 9–12, nov. 2014. ISSN 1558-4046. Conference Name: IEEE Security & Privacy. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7006434>. Citado na página 94.

MCMILLAN, R.; KNUTSON, R.; SEETHARAMAN, D. Yahoo Discloses New Breach of 1 Billion User Accounts. *Wall Street Journal*, dez. 2016. ISSN 0099-9660. Disponível em: <http://www.wsj.com/articles/yahoo-discloses-new-breach-of-1-billion-user-accounts-1481753131>. Citado na página 40.

MEYER, K. et al. Decentralizing Control and Intelligence in Network Management. In: SETHI, A. S.; RAYNAUD, Y.; FAURE-VINCENT, F. (Ed.). *Integrated Network Management IV: Proceedings of the fourth international symposium on integrated network management, 1995*. Boston, MA: Springer US, 1995, (IFIP — The International Federation for Information Processing). p. 4–16. ISBN 978-0-387-34890-2. Disponível em: <https://doi.org/10.1007/978-0-387-34890-2_1>. Citado na página 93.

Microsoft. *Banco de Dados SQL do Azure – Serviço de Banco de Dados de Nuvem Gerenciado | Microsoft Azure*. 2023. Disponível em: <https://azure.microsoft.com/pt-br/products/azure-sql/database>. Citado na página 39.

Microsoft. *Microsoft Security Development Lifecycle - Windows Security*. 2023. Disponível em: <https://learn.microsoft.com/en-us/windows/security/security-foundations/msft-security-dev-lifecycle>. Citado 12 vezes nas páginas 16, 26, 28, 55, 57, 58, 59, 60, 61, 77, 94 e 95.

Microsoft. *Serviço de Aplicativo Web | Microsoft Azure*. 2023. Disponível em: <https://azure.microsoft.com/pt-br/products/app-service/web>. Citado na página 39.

MINAYO, M. C. d. S. Análise qualitativa: teoria, passos e fidedignidade. *Ciência & Saúde Coletiva*, v. 17, p. 621–626, mar. 2012. ISSN 1413-8123, 1678-4561. Disponível em: <https://www.scielo.br/j/csc/a/39YW8sMQhNzG5NmpGBtNMFf/?lang=pt>. Citado na página 30.

MORRIS, C. *LinkedIn data theft exposes personal information of 700 million people*. 2021. Disponível em: <https://fortune.com/2021/06/30/linkedin-data-theft-700-million-users-personal-information-cybersecurity/>. Citado 3 vezes nas páginas 35, 37 e 38.

MORRISON, P. et al. Mapping the field of software life cycle security metrics. *Information and Software Technology*, v. 102, p. 146–159, out. 2018. ISSN 0950-5849. Disponível em: <https://www.sciencedirect.com/science/article/pii/S095058491830096X>. Citado na página 27.

Norton. *What is a computer virus?* 2018. Disponível em: <https://br.norton.com/blog/malware/what-is-a-computer-virus>. Citado na página 79.

O'GORMAN, G.; MCDONALD, G. Ransomware: A Growing Menace. 2012. Citado na página 79.

PAN, Y. Interactive Application Security Testing. In: *2019 International Conference on Smart Grid and Electrical Automation (ICSGEA)*. [s.n.], 2019. p. 558–561. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8901378>. Citado na página 95.

PARMAR, B. Protecting against spear-phishing. *Computer Fraud & Security*, v. 2012, n. 1, p. 8–11, jan. 2012. ISSN 13613723. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1361372312700076>. Citado 3 vezes nas páginas 44, 45 e 58.

PATERSON, K. G. The Cyber Security Body of Knowledge v1.1.0, 2021. In: . University of Bristol, 2021. Section: Applied Cryptography. Disponível em: <https://www.cybok.org/>. Citado 2 vezes nas páginas 26 e 78.

PEREZ, S. *117 million LinkedIn emails and passwords from a 2012 hack just got posted online*. 2016. Disponível em: <https://techcrunch.com/2016/05/18/117-million-linkedin-emails-and-passwords-from-a-2012-hack-just-got-posted-online/>. Citado na página 16.

PERLROTH, N. Yahoo Says Hackers Stole Data on 500 Million Users in 2014. *The New York Times*, set. 2016. ISSN 0362-4331. Disponível em: <https://www.nytimes.com/2016/09/23/technology/yahoo-hackers.html>. Citado na página 40.

PIESSENS, F. The Cyber Security Body of Knowledge v1.1.0, 2021. In: . University of Bristol, 2021. Section: Software Security. Disponível em: <https://www.cybok.org/>. Citado 3 vezes nas páginas 19, 20 e 21.

PIETERVANHOVE. *Always Encrypted com enclaves seguros - SQL Server*. 2023. Disponível em: <https://learn.microsoft.com/pt-br/sql/relational-databases/security/encryption/always-encrypted-enclaves?view=sql-server-ver16>. Citado na página 79.

Ponemon Institute. *Data Risk in the Third-Party Ecosystem: Third Annual Study*. 2018. Disponível em: <https://www.ponemon.org/userfiles/filemanager/nvqfztft3qtufvi5gl60/>. Citado 2 vezes nas páginas 94 e 96.

Ponemon Institute. *Ponemon Institute Cost of a Data Breach Study 2018*. 2018. Disponível em: <https://securityintelligence.com/series/ponemon-institute-cost-of-a-data-breach-2018/>. Citado na página 24.

REINA, G. et al. The moving target of visualization software for an increasingly complex world. *Computers & Graphics*, v. 87, p. 12–29, abr. 2020. ISSN 0097-8493. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0097849320300078>. Citado na página 16.

RESCORLA, E. *The Transport Layer Security (TLS) Protocol Version 1.3*. [S.l.], 2018. Num Pages: 160. Disponível em: <https://datatracker.ietf.org/doc/rfc8446>. Citado na página 79.

ROBERTS, S. Learning lessons from data breaches. *Network Security*, v. 2018, n. 11, p. 8–11, nov. 2018. ISSN 1353-4858, 1872-9371. Disponível em: <http://www.magonlinelibrary.com/doi/10.1016/S1353-4858%2818%2930111-9>. Citado 4 vezes nas páginas 43, 44, 45 e 77.

RUNESON, P. A survey of unit testing practices. *IEEE Software*, v. 23, n. 4, p. 22–29, jul. 2006. ISSN 1937-4194. Conference Name: IEEE Software. Disponível em: <https://ieeexplore.ieee.org/abstract/document/1657935>. Citado na página 95.

SALEMI, M.; SADRE, R.; LEGAY, A. "Automated rules generation into Web Application Firewall using Runtime Application Self-Protection. 2020. Disponível em: <https://dial.uclouvain.be/downloader/downloader.php?pid=thesis%3A25351&datastream=PDF_01>. Citado na página 96.

SERPRO. *LGPD - Lei Geral de Proteção de Dados Pessoais | Serpro*. 2023. Disponível em: <https://www.serpro.gov.br/lgpd>. Citado na página 24.

SHARMA, R.; SIBAL, R.; SABHARWAL, S. Software vulnerability prioritization using vulnerability description. *International Journal of System Assurance Engineering and Management*, v. 12, n. 1, p. 58–64, fev. 2021. ISSN 0976-4348. Disponível em: <https://doi.org/10.1007/s13198-020-01021-7>. Citado na página 97.

SMART, N. The Cyber Security Body of Knowledge v1.1.0, 2021. In: . University of Bristol, 2021. Section: Cryptography. Disponível em: <https://www.cybok.org/>. Citado na página 25.

SOUSA, A. S. d.; OLIVEIRA, G. S. d.; ALVES, L. H. A PESQUISA BIBLIOGRÁFICA: PRINCÍPIOS E FUNDAMENTOS. *Cadernos da FUCAMP*, v. 20, n. 43, mar. 2021. ISSN 2236-9929. Number: 43. Disponível em: <https://revistas.fucamp.edu.br/index.php/cadernos/article/view/2336>. Citado na página 30.

SUMNER, A.; YUAN, X. Mitigating Phishing Attacks: An Overview. In: *Proceedings of the 2019 ACM Southeast Conference*. Kennesaw GA USA: ACM, 2019. p. 72–77. ISBN 978-1-4503-6251-1. Disponível em: <https://dl.acm.org/doi/10.1145/3299815.3314437>. Citado na página 43.

SYED, R.; DHILLON, G. Dynamics of Data Breaches in Online Social Networks: Understanding Threats to Organizational Information Security Reputation. 2015. Citado na página 50.

SYSTEMS, Z. P. E. *Centralized vs. Distributed Network Management: Which One to Choose?* 2021. Disponível em: <https://zpesystems.com/centralized-vs-distributed-network-management-zs/>. Citado na página 93.

Sá-SILVA, J. R.; ALMEIDA, C. D. d.; GUINDANI, J. F. Pesquisa documental: pistas teóricas e metodológicas. *Revista Brasileira de História & Ciências Sociais*, v. 1, n. 1, jul. 2009. ISSN 2175-3423. Number: 1. Disponível em: <https://periodicos.furg.br/rbhcs/article/view/10351>. Citado na página 30.

TAHBOUB, R.; SALEH, Y. Data leakage/loss prevention systems (DLP). In: *2014 World Congress on Computer Applications and Information Systems (WCCAIS)*. IEEE, 2014. p. 1–6. Disponível em: <https://ieeexplore.ieee.org/abstract/document/6916624.> Citado na página 83.

TAYLOR, S. *New LinkedIn Data Leak Leaves 700 Million Users Exposed*. 2021. Disponível em: <https://restoreprivacy.com/linkedin-data-leak-700-million-users/>. Citado na página 36.

The MITRE Corporation. *CVE - CVE*. 2023. Disponível em: <https://cve.mitre.org/>. Citado na página 21.

The MITRE Corporation. *CWE - Common Weakness Enumeration*. 2023. Disponível em: <https://cwe.mitre.org/>. Citado 2 vezes nas páginas 16 e 21.

The OWASP Foundation. *OWASP Top 10 - 2017*. OWASP, 2017. Original-date: 2016-08-30T15:46:11Z. Disponível em: <https://github.com/OWASP/Top10>. Citado na página 24.

The OWASP Foundation. *A02 Cryptographic Failures - OWASP Top 10:2021*. 2021. Disponível em: <https://owasp.org/Top10/A02_2021-Cryptographic_Failures/>. Citado 2 vezes nas páginas 15 e 28.

The OWASP Foundation. *OWASP Top Ten*. 2021. Disponível em: <https://owasp.org/www-project-top-ten/>. Citado 8 vezes nas páginas 15, 25, 56, 57, 58, 59, 77 e 78.

THIELMAN, S. Yahoo hack: 1bn accounts compromised by biggest data breach in history. *The Guardian*, v. 15, p. 2016, 2016. Citado 2 vezes nas páginas 41 e 42.

THOMAS, K.; MOSCICKI, A. *New research: How effective is basic account hygiene at preventing hijacking*. 2019. Disponível em: <https://security.googleblog.com/2019/05/new-research-how-effective-is-basic.html>. Citado na página 85.

TRAUTMAN, L. J.; ORMEROD, P. C. Corporate Directors' and Officers' Cybersecurity Standard of Care: The Yahoo Data Breach. 2017. Citado 5 vezes nas páginas 39, 40, 41, 42 e 77.

United States Department of Labor. *Guidance on the Protection of Personal Identifiable Information*. 2022. Disponível em: <http://www.dol.gov/general/ppii>. Citado 2 vezes nas páginas 23 e 31.

VANIAN, J. *Data from half a billion LinkedIn users has been scraped and put online*. 2021. Disponível em: <https://fortune.com/2021/04/08/linkedin-user-data-breach-leak-hackers/>. Citado na página 36.

VANIAN, J. *Everything to know about Facebook's huge data leak*. 2021. Disponível em: <https://fortune.com/2021/04/07/facebooks-data-leak-everything-to-know/>. Citado na página 36.

VERMA, D. et al. Lack of software engineering practices in the development of bioinformatics software. *ICCGI*, v. 2013, p. 57–62, 2013. Disponível em: <https://www.academia.edu/download/85340444/download_full.pdf#page=71>. Citado na página 93.

WAGNER, P. SSRN Scholarly Paper, *Third Party Breaches - A Survey of Threats and Recommendations*. Rochester, NY: [s.n.], 2021. Disponível em: <https://papers.ssrn.com/abstract=3782822>. Citado na página 97.

WANG, P.; JOHNSON, C. CYBERSECURITY INCIDENT HANDLING: A CASE STUDY OF THE EQUIFAX DATA BREACH. *Issues In Information Systems*, 2018. ISSN 15297314. Disponível em: <https://iacis.org/iis/2018/3_iis_2018_150-159.pdf>. Citado 2 vezes nas páginas 47 e 48.

WATERSON, D. Managing endpoints, the weakest link in the security chain. *Network Security*, v. 2020, n. 8, p. 9–13, ago. 2020. ISSN 1353-4858. Publisher: Elsevier. Disponível em: <https://www.magonlinelibrary.com/doi/abs/10.1016/S1353-4858%2820%2930093-3>. Citado na página 97.

WILLIAMS, L. The Cyber Security Body of Knowledge v1.1.0, 2021. In: . University of Bristol, 2021. Section: Secure Software Lifecycle. Disponível em: <https://www.cybok.org/>. Citado 4 vezes nas páginas 19, 20, 24 e 26.

WIRTH, N. A Brief History of Software Engineering. *IEEE Annals of the History of Computing*, v. 30, n. 3, p. 32–39, jul. 2008. ISSN 1934-1547. Conference Name: IEEE Annals of the History of Computing. Disponível em: <https://ieeexplore.ieee.org/abstract/document/4617912>. Citado na página 93.

XIAO, S.; WITSCHEY, J.; MURPHY-HILL, E. Social influences on secure development tool adoption: why security tools spread. In: *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. New York, NY, USA: Association for Computing Machinery, 2014. (CSCW '14), p. 1095–1106. ISBN 9781450325400. Disponível em: <https://dl.acm.org/doi/10.1145/2531602.2531722>. Citado na página 16.

Apêndices

APÊNDICE A – Guia de Medidas de Segurança Contra Ameaças de Exposição de Dados Sensíveis

A.1 Introdução

A.1.1 Segurança no desenvolvimento web

Aplicações web desempenham um papel fundamental em muitos aspectos do cotidiano além de ambientes de trabalho. Presentes em plataformas de comércio eletrônico ou sistemas de gerenciamento de dados, sites de notícias ou redes sociais, as aplicações web tornaram-se parte integrante de muitos setores. Entretanto, mesmo que esses ambientes ofereçam benefícios, as aplicações web também apresentam desafios significativos em termos de segurança.

A crescente interconexão e acessibilidade dessas aplicações via internet levantam preocupações em questões de segurança. Dada a natureza das aplicações web, que coletam, processam e armazenam dados sensíveis dos usuários, elas se tornam alvos atrativos para potenciais ataques. Questões relacionadas à privacidade e segurança cibernética são, portanto, temas críticos a serem abordados no desenvolvimento e na manutenção de aplicações web (CANONGIA; JUNIOR, 2009).

A.1.2 Objetivo

O objetivo deste guia é servir como uma espécie de dicionário, onde é possível identificar um problema e encontrar descrições práticas de segurança relacionadas ao problema. Este documento pode ter utilidade para quaisquer indivíduos direta ou indiretamente ligados à produção de aplicações web, principalmente àqueles que possuem foco em segurança dessas aplicações.

O uso deste guia não exige que seja seguida uma ordem, onde, assim como um dicionário, caso queira ler a respeito de um problema específico, basta ir até a seção correspondente e esta seção trará toda informação necessária reunida neste trabalho. Contudo, para aqueles que tiverem interesse em ter noções acerca de vários tipos de problemas que acometem o universo do desenvolvimento web, a leitura completa do guia também é de grande valia.

A.2 Referencial

Para a produção deste guia, foram realizadas pesquisas no ambiente bibliográfico acerca de boas práticas para o desenvolvimento web focado em se precaver em relação à exposição de dados sensíveis. Esse guia se baseia na metodologia de desenvolvimento seguro da Microsoft, o SDL (Microsoft, 2023b) e nas práticas de desenvolvimento seguro recomendadas pela OWASP (The OWASP Foundation, 2021b), com foco na prevenção da exposição de dados sensíveis.

Além da base bibliográfica, esse guia também se escora na pesquisa documental, em cima de análises de casos de exposição de dados sensíveis em empresas globais de grande porte. Os casos analisados para a construção desse guia pertencem às seguintes empresas:

- LinkedIn, plataforma profissional para *networking* e oportunidades de carreira, que passou por vazamentos de dados em 2021 (GIBSON et al., 2021);
- Yahoo, portal de internet que oferece serviços variados como mecanismo de busca, e-mail, notícias e entretenimento, que passou por vazamentos nos anos de 2013 e 2016 (TRAUTMAN; ORMEROD, 2017);
- eBay, plataforma de comércio eletrônico que facilita a compra e venda de produtos entre usuários, que sofreu um ataque em 2014 (ROBERTS, 2018);
- Equifax, empresa de serviços de informação de crédito que avalia e fornece relatórios de crédito para indivíduos e empresas, auxiliando em decisões de crédito e gestão de riscos financeiros, que teve dados vazados em 2017 (LUSZCZ, 2018);
- JPMorgan Chase, instituição financeira global, que oferece serviços bancários, financeiros e de gestão de ativos para clientes corporativos, institucionais e individuais, que teve dados expostos em 2014 (DASWANI; ELBAYADI, 2021);
- Target, rede varejista americana, contando com lojas físicas e online, que teve seus dados expostos em 2013 (DASWANI; ELBAYADI, 2021).

As recomendações presentes a seguir foram inspiradas nesses casos, o que não significa que este documento seja um guia definitivo, mas sim que foi baseado em casos reais, buscando trazer práticas que podem prevenir algumas das várias situações reais de ataque em plataformas web.

A principal referência para a construção desse guia é o livro “Big Breaches - Cybersecurity Lessons for Everyone” (DASWANI; ELBAYADI, 2021), onde mais de 9 mil vazamentos foram estudados, trazendo os mais relevantes para o livro e ao final sugerindo práticas de segurança para evitar as principais causas raiz encontradas em sua pesquisa.

A.3 Criptografia e Categorização de Dados

Dados podem estar em alguns estados, e cada estado apresenta características e cuidados diferentes. Podem estar em repouso, armazenados em algum local, em movimento, transmitidos pela rede, ou em uso, na memória de um dispositivo, e a confidencialidade de dados considerados sensíveis deve ser protegida em qualquer um desses estados (DASWANI; ELBAYADI, 2021).

O objetivo desta seção é apresentar conceitos sobre a disposição de dados sensíveis em diferentes estados. Para obter informações mais técnicas sobre criptografia, recomenda-se pesquisar e estudar algorítimos de criptografia em materiais bibliográficos ou em centros de recomendações focados na segurança de dados sensíveis, como a OWASP (The OWASP Foundation, 2021b).

A.3.1 Dados em Repouso

Em incidentes onde dispositivos são perdidos ou roubados, caso os dados armazenados não estejam criptografados, têm-se uma violação de dados. A criptografia de dados armazenados pode ser feita pelo sistema operacional ou pelo próprio disco rígido, criptografando os dados no disco com uma chave de criptografia que não deve ser armazenada de forma clara no dispositivo. Sistemas operacionais modernos oferecem algum tipo de criptografia em nível de armazenamento e ativar essa funcionalidade de criptografia evita que incidentes de perda ou roubo de dispositivos se tornem violações de dados (DASWANI; ELBAYADI, 2021).

Quando a criptografia é usada como uma ferramenta para proteger a confidencialidade dos dados, é de extrema importância escolher o local ideal onde armazenar as chaves de criptografia e decidir quais pessoas terão acesso a elas. Embora sejam mais incomuns, ataques que roubam fisicamente um dispositivo que armazena dados sensíveis ainda existem, portanto, ao utilizar dispositivos que contenham informações sensíveis, é necessário cuidar para que os dados estejam criptografados a nível de disco rígido (DASWANI; ELBAYADI, 2021).

A.3.2 Dados em Movimento

No momento que que os dados sensíveis estão sendo trafegados de um dispositivo para outro através da rede, é necessário garantir que esses dados não podem ser capturados por agentes externos a essa comunicação. Para isso, pode-se criptografar os dados com uma chave que é conhecida apenas pelas extremidades da comunicação (PATERSON, 2021). Os dados são então criptografados antes da transmissão pela rede e só são descriptografados ao serem recebidos pelo interessado legítimo.

Muitos são os protocolos que podem ser utilizados para garantir a integridade dos dados durante o tráfego, mas um dos mais utilizados pelos navegadores e servidores da web é o TLS (DASWANI; ELBAYADI, 2021). Após estabelecer uma conexão de rede, o TLS utiliza uma chave pública que combina com a chave privada conhecida apenas pelas extremidades da comunicação, para criptografar o tráfego de dados nesse canal (RESCORLA, 2018).

A.3.3 Dados em Uso

Por fim, é necessário garantir a segurança de dados sensíveis também quando estão em uso. Para isso, recomenda-se não descriptografar dados sensíveis na memória padrão do dispositivo, mas sim deixar para processar esses dados criptografados em um enclave seguro, que é uma região protegida da memória do dispositivo, que apresenta um ambiente de execução confiável, com sua própria CPU e memórias dedicadas. A memória nesse ambiente e os registradores do processador usados no enclave seguro são criptografados com chaves que são inacessíveis fora desse ambiente (PIETERVANHOVE, 2023).

A.4 *Malwares*

Mesmo que sempre surjam novos softwares maliciosos, ainda é possível impedir que eles sejam executados, detectando sua existência na rede e então os neutralizar. Agentes maliciosos sempre irão existir, criando novos *malwares*, logo, deve-se investir em impedir a execução desses programas, detectando-os e neutralizando-os, impedindo o *malware* de cumprir sua função e entregar algo de valor para o atacante.

Malwares são nomeados e categorizados com base no que fazem ou como se espalham. Um *ransomware*, por exemplo, tem esse nome pois criptografa dados com uma chave desconhecida pelo proprietário do sistema, inutilizando os sistemas que precisam desses dados, e não fornece ao proprietário do sistema a chave de descriptografia enquanto o autor ou operador do *malware* seja pago com um resgate. Em inglês, a palavra "resgate" é "ransom", por isso o nome, *ransomware* (O'GORMAN; MCDONALD, 2012). Os vírus possuem esse nome por causa da forma que se espalham, já que são *malwares* que podem se replicar com a ajuda de um intermediário humano, que insere um dispositivo infectado no computador, por exemplo (Norton, 2018).

Ransomwares mais sofisticados conseguem criptografar e excluir backups. Então além de implementar defesas robustas contra *malwares*, é necessário efetivamente testar se é possível recuperar dados dos backups assim que forem gerados e configurados. Dessa forma, caso seja necessário se recuperar de um ataque de *ransomware*, os responsáveis pelo backup existente já confirmaram que é possível recuperar esses dados. Muitas empresas

configuram um sistema de backup, mas não o testam regularmente e não conseguem recuperar os dados quando necessário (DASWANI; ELBAYADI, 2021).

A.4.1 *Antimalware*

O principal motivo para a ocorrência de tantas violações de dados envolvendo *malwares* se dá, em parte, no fato de que as organizações não implementam defesas *antimalware* suficientes, e mesmo as implementadas, não são robustas o suficiente (DASWANI; ELBAYADI, 2021).

Atualmente, os atacantes não apenas desenvolvem um *malware* e o lançam na expectativa de atingir o alvo. Eles geram muitas variações do *malware* e o executam através dos escâneres de detecção que a organização alvo pode utilizar. Somente ao desenvolver uma variante suficientemente diferente, que não possua uma assinatura conhecida que possa ser detectada pelos escâneres da organização é que o *malware* é implantado (KARA, 2019). Dessa forma, o *malware* pode infectar e realizar suas atividades sem ser detectado por alguns dias, semanas ou meses.

Esse modelo de detecção de *malware* baseado na sua assinatura tem se tornado obsoleto ao longo dos anos, então se tornou necessário utilizar abordagens mais inovadoras, como o uso de inteligência artificial, para detectar os *malwares* que escapam dos escâneres conhecidos (FARUK et al., 2021). Recomenda-se então que sejam utilizados produtos *antimalware* que façam uso de tecnologias como a inteligência artificial para detectar programas maliciosos, mas sem desprezar os escâneres que detectam *malwares* mais básicos.

Uma observação quanto ao uso dos programas *antimalware* diz respeito a limitação dos recursos locais onde esse programa está instalado. Programas desse tipo que aproveitam recursos em nuvem tendem a ser mais eficazes do que os que usam recursos locais. Sendo assim, a análise que pode ser realizada no centro de dados do fornecedor do *antimalware* na nuvem pode ser feita de forma mais aprofundada do que a análise no dispositivo local (DASWANI; ELBAYADI, 2021).

Produtos *antimalware* são conhecidos como proteção de *endpoint* e possuem recursos adicionais como proteções para navegação segura. Dessa forma, se um usuário acessar um link ou domínio conhecido por distribuir programas maliciosos, o software de proteção *endpoint* pode bloquear o acesso ao site antes que o *malware* tenha a oportunidade de chegar na maquina do usuário (DASWANI; ELBAYADI, 2021).

A.4.2 Detecção e Resposta em *Endpoints*

Além das defesas *antimalware* em execução no *endpoint*, o software de detecção e resposta de *endpoint* tem se tornado mais conhecido recentemente. Softwares desse tipo

proporcionam visibilidade às equipes de segurança em relação às atividades e eventos do sistema que podem ajudar a descobrir violações de segurança, além de detectarem programas maliciosos. Softwares de detecção e resposta oferecem benefícios como indicadores de comprometimento ou indicadores de ataque, identificação de processos ativos no *endpoint*, identificação de conexções de rede entre o *endpoint* e máquinas internas ou externas, acesso ao histórico de contas que realizaram login no *endpoint*, identificação da criação de arquivos compactados que um invasor pode usar para extração de dados (HASSAN; BATES; MARINO, 2020).

A.4.3 Detecção e Resposta de Rede

O ponto negativo das defesas orientadas a *endpoint* é que elas atuam justamente apenas no final do sistema, ou seja depois que o problema já pode ter passado pela rede. Sendo assim, é importante utilizar também defesas *antimalware* na rede e inspecionar código e dados que estão trafegando pela mesma, com o objetivo de identificar e bloquear um programa malicioso antes que ele chegue a um *endpoint*. Esses são os objetivos de ferramentas de detecção e resposta de rede (BEJTLICH, 2013). Métricas relevantes para avaliar a eficiência desse tipo de ferramenta incluem o tempo médio de detecção, taxa de falsos positivos e falsos negativos (DASWANI; ELBAYADI, 2021).

Entretanto, é importante lembrar que mesmo utilizando dessas ferramentas, que a taxa de detecção nunca é de 100%, por isso se torna tão importante combinar as ferramentas e estratégias.

A.4.4 Isolamento Remoto de Navegador

Existem ainda tecnologias que isolam o *malware*, impedindo-o de ser transmitido para a rede de uma organização e de alcançar um *endpoint*. Ferramentas de isolamento remoto de navegador trabalham baseados na ideia de que o navegador web é o maior responsável por transmissões de programas maliciosos, já que muitos usuários leem seus e-mails e passam a maior parte do tempo online usando o navegador e consequentemente resultando em downloads de *malwares* (HU; WANG; LIU, 2023).

Outros pontos de infecção comuns como páginas web infectadas, seja por download direto, widgets ou anúncios, também podem resultar no download de programas maliciosos para a máquina do usuário por meio do navegador web. Plug-ins que os navegadores usam para renderizar conteúdo podem ter também vulnerabilidades de software que podem ser exploradas para enviar *malwares*. Embora seja possível desativar esses plug-ins nos *endpoints*, é mais seguro controlar a configuração do navegador e seus plug-ins em um servidor na nuvem e permitir que os *endpoints* apenas vejam e interajam com a exibição (DASWANI; ELBAYADI, 2021).

A tecnologia de isolamento remoto de navegador isola fisicamente o navegador, executando-o em um servidor separado e enviando apenas os pixels de exibição para o dispositivo do usuário. Os usuários conseguem interagir com a página da web e os pixels de exibição como se o navegador em seu dispositivo estivesse renderizando o conteúdo, mas o risco associado à renderização desse conteúdo e ao uso de quaisquer plug-ins ou downloads é transferido para um servidor protegido na nuvem (HU; WANG; LIU, 2023).

Apesar de uma latência seja adicionada com o uso dessa tecnologia, devido à renderização real não ocorrer no dispositivo do usuário, esse tipo de prática ainda é válida. Outro ponto negativo é o aumento da largura de banda de rede e os custos associados, pois o conteúdo precisa primeiro ser baixado em um servidor na nuvem para depois ser descarregado no *endpoint* (DASWANI; ELBAYADI, 2021).

A.4.5 Interface de Ambiente de Trabalho Virtual

Em situações onde não é necessário acessar aplicativos localmente nos *endpoints*, utilizar interfaces de ambientes de trabalho virtuais é uma prática que não isola apenas o navegador fisicamente, mas também todos os aplicativos do ambiente de trabalho e os dados desses aplicativos. Sendo assim, todos os aplicativos e dados estão em um servidor na nuvem, e apenas os pixels de exibição são enviados para o *endpoint*.

Essa prática pode ajudar a prevenir infecções por *malware*, já que a navegação web não ocorre no dispositivo final do usuário, mas apenas no servidor em nuvem protegido. O ponto negativo é a complexidade, já que todos os aplicativos do ambiente de trabalho precisam ser virtualizados, ao invés de apenas o navegador (DASWANI; ELBAYADI, 2021).

A.5 Falhas na Administração e Uso de Sistemas

Este tópico possui ligação com outros como *phishing* ou vulnerabilidades no desenvolvimento de softwares pois aborda falhas que um integrante de uma organização pode cometer enquanto desenvolve ou usa um sistema. Logo, muitas das tecnologias discutidas em outras seções deste guia também ajudarão a evitar erros relacionados aos integrantes de uma organização.

Treinar todos os funcionários e parceiros de uma organização em conscientização de segurança é uma prática muito importante para evitar erros de administração ou uso de um sistema (DASWANI; ELBAYADI, 2021). Esse tipo de treinamento aumenta o envolvimento dos envolvidos com a empresa, além de capacitá-los em habilidades como identificar e evitar ataques de engenharia social aos quais são constantemente expostos (EMINAğAOğLU; UçAR; EREN, 2009).

Além de treinar os funcionários, a organização pode também fazer uso de ferramentas de prevenção de perda de dados, que são ferramentas que detectam quando informações sensíveis possivelmente estão sendo enviadas para destinatários externos à empresa (TAHBOUB; SALEH, 2014). Além de detectar, essas ferramentas também bloqueiam essa saída de dados, então, se por exemplo, um funcionário acidentalmente enviar um e-mail com informações anexadas referentes aos salários dos funcionários da empresa para um destinatário que não está identificado como parte da organização, a ferramenta identifica que o e-mail para qual o arquivo está sendo enviado não faz parte da empresa e bloqueia esse envio, evitando uma exposição de dados sensíveis.

A.6 *Phishing*

Ataques de *phishing* fazem parte da internet desde os anos 90 (ALKHALIL et al., 2021). Quando começou, era possível criar ataques de *phishing* com facilidade, pois o SMTP (*Simple Mail Transfer Protocol*, ou Protocolo Simples de Transferência de Correio), utilizado para enviar e receber e-mails, não autenticava o remetente do e-mail. Sendo assim, qualquer pessoa podia mandar um e-mail para qualquer outra, alegando ser quem quisesse e afirmando ser de qualquer organização que quisesse, possibilitando que impostores executasse ataques de *phishing* (DASWANI; ELBAYADI, 2021).

Em um ataque de *phishing* bem-sucedido, a vítima confia no e-mail recebido pois acredita que a identidade do suposto remetente na seção “De:” do e-mail está correta. Além disso, a vítima, confiando na veracidade do e-mail, clica em um link que redireciona para um formulário de login em um site falso e inadvertidamente entrega sua senha ao atacante.

A partir disso, a primeira recomendação contra esse tipo de ataque é implementar autenticação de dois fatores ao invés de contar apenas com a senha, fazendo com que seja necessário não apenas saber algo (a senha), mas também ter algo (chave de segurança ou um dispositivo móvel) ou ser algo (utilizando biometria) (DASWANI; ELBAYADI, 2021).

A.6.1 Autenticação de Dois Fatores

As principais razões que impulsionam a necessidade de se usar autenticação de dois fatores são as muitas violações que geram mais credenciais comprometidas, alimentando o conjunto de dados roubados dos atacantes, o que fortalece as próximas tentativas de invadir uma organização. Soma-se a isso o fato de que clientes e funcionários geralmente não escolhem senhas fortes (DASWANI; ELBAYADI, 2021).

Toda organização poderia fazer uso de autenticação de dois fatores com certa facilidade, visto que estão disponíveis no mercado várias opções para tipos de projetos diferentes, como o Microsoft Office 365 e o Slack para ferramentas de produtividade e

colaboração, AWS e Azure para serviços em nuvem e o GitHub para repositórios de código-fonte.

Habilitar autenticação de dois fatores utilizando os dispositivos móveis de seus funcionários é recomendável já que a maioria dos funcionários possuem esses dispositivos. Independentemente das prioridades de uma organização, a defesa contra ataques de *phishing* pode ser rapidamente aprimorada ao habilitar autenticação de dois fatores por meio de aplicativos autenticadores de dispositivos móveis que podem ser baixados na maioria dos celulares (DASWANI; ELBAYADI, 2021).

A.6.2 Chaves de Segurança

Um dos fatores secundários de autenticação mais bem avaliados até o momento para eliminar ameaças de *phishing* é o uso de chaves de segurança em hardware (DASWANI; ELBAYADI, 2021).

Similar a um motorista que insere uma chave para ligar um carro, os funcionários utilizam uma chave de segurança em seu ambiente de trabalho (laptops, desktops, celulares) para poderem fazer login em sites e aplicativos da organização. Uma chave de segurança é geralmente um token, similar a um pendrive, que autentica um funcionário apenas quando está conectado e o funcionário toca na chave.

Este guia não tem como objetivo explicar tecnicamente como funciona o processo, mas a essência é que o hardware da chave de segurança é utilizado para gerar uma confirmação autenticada de que o usuário legítimo deseja fazer login em um site específico, e não um criado por um impostor. A assinatura digital gerada pelo hardware da chave de segurança funciona como uma segunda senha que permite apenas ao usuário legítimo fazer login, apenas no site legítimo.

Utilizar abordagens de autenticação de dois fatores no geral não tornam um site à prova de *phishing*. Os atacantes sempre podem configurar um site falso que encaminha credenciais de um usuário para o site legítimo, força o site legítimo a enviar um código de dois fatores para o usuário como um SMS, e depois solicita que o usuário informe o código de dois fatores no site falso, fornecendo ao atacante ambos fatores necessários para se passar pelo usuário no site legítimo. Porém, fazendo uso de chaves de segurança, como navegador da web verifica o se o nome de domínio do site corresponde ao nome no certificado do site, um atacante precisaria falsificar também o certificado do site falso para enganar a chave de segurança (DASWANI; ELBAYADI, 2021).

Até o momento, as chaves de segurança oferecem a defesa mais eficaz contra *phishing*. O Google relatou que, após implantar chaves de segurança no início de 2017 para seus mais de 85 mil funcionários, eles não registraram nenhum ataque de *phishing* bem-sucedido ou invasão de conta mais de um ano depois. Relatou também, em 2019,

que as chaves de segurança eram 100% eficazes contra bots automatizados, ataques de *phishing* em massa e ataques direcionados (THOMAS; MOSCICKI, 2019).

A.6.3 Tokens OTP Dedicados

Uma opção para um segundo fator baseado em algo que o usuário tem são tokens OTP (One-Time Password ou Senha de Uso Único), onde, depois que um usuário insere suas credenciais, eles são solicitados a inserir um código gerado por um sistema dedicado. Esse código é unico e válido apenas por um curto período, o que fornece uma camada adicional de segurança.

Códigos OTP permitem que o cliente prove ao servidor que ele conhece uma chave compartilhada usada para gerar criptograficamente um código OTP. Esse código difere a cada vez, mas é baseado na chave compartilhada. Esse processo fornece uma forma dinâmica e temporária de autenticação, uma vez que o código muda regularmente, tornando mais difícil para os invasores preverem ou reproduzirem o código.

Para configurar um OTP, o usuário se registra no servidor e um nome de usuário, senha e uma chave compartilhada são acordados. Uma vez registrado, o usuário pode fazer login com os dois fatores. Primeiro, o servidor solicita um nome de usuário e senha, o primeiro fator. O servidor verifica as credenciais, e se forem consideradas autênticas, o servidor verifica então um segundo fator, o código OTP, único a partir do dispositivo que o gerou (DASWANI; ELBAYADI, 2021).

A.6.3.1 Autenticação de Dois Fatores em Aplicativos Móveis

À medida que dispositivos móveis passaram a poder executar aplicativos de software, é compreensível que aplicações que aproveitem do poder computacional desses dispositivos passaram a gerar código OTP. Google Authenticator e Microsoft Authenticator são exemplos de aplicativos autenticadores móveis de dois fatores.

Alguns aplicativos autenticadores oferecem funcionalidades que permitem aos usuários simplesmente aprovar um login ao receber uma solicitação de autenticação via push no celular. Internamente, esses aplicativos fornecem o código de autenticação ao servidor sem a interação direta do usuário, tornando o processo de login mais conveniente. No entanto, quando um atacante obtém acesso às credenciais de um usuário, eles podem persuadir esse usuário a clicar em "aprovar", mesmo quando o usuário não está fazendo login, dando ao atacante acesso à conta. Usuários muitas vezes clicam rapidamente em qualquer coisa que atrapalhe o uso de seus celulares (DASWANI; ELBAYADI, 2021).

Portanto, em ambientes sensíveis, pode valer a pena exigir que os funcionários digitem os códigos dos aplicativos autenticadores, ao invés de simplesmente clicar para aprovar um login.

A.6.3.2 OTP Baseado em SMS

Em casos em que o usuário não tem um aplicativo autenticador em seu celular, um servidor pode enviar o código de dois fatores para o celular do usuário via SMS (Short Message Service, ou Serviço de Mensagens Curtas).

Um ponto negativo é que o SMS não é criptografado e pode ser suscetível a ataques *man-in-the-middle*, onde um terceiro captura dados que em tráfego. Por exemplo, se um usuário possui um malware em seu celular que pode ler todas suas mensagens SMS e consequentemente, todos os códigos de dois fatores enviados por esse método (DASWANI; ELBAYADI, 2021).

A.6.3.3 OTP Baseado em E-mail

Outro método de autenticação de dois fatores para fazer login em um site sensível ou corporativo é enviar o código de segundo fator para o e-mail do usuário ao invés do SMS. Se o usuário puder comprovar que pode fazer login em seu e-mail registrado e acessar o código OTP enviado para o e-mail, será permitido fazer o login. Entretanto, se um atacante conseguir comprometer o e-mail do usuário, o atacante também poderá roubar os códigos de dois fatores enviados para o e-mail (DASWANI; ELBAYADI, 2021).

A.6.4 Autenticação de Múltiplos Fatores

A autenticação de múltiplos fatores se refere ao uso de mais de um fator para autenticação. A autenticação de dois fatores é uma forma de autenticação de múltiplos fatores, porém, mais do que apenas dois fatores podem ser utilizados na autenticação de múltiplos fatores.

Outros fatores que podem ser utilizados ao realizar uma autenticação são (DASWANI; ELBAYADI, 2021):

- Tipo de dispositivo e características de configuração do dispositivo;
- Localização do usuário;
- Comportamento do usuário em aplicativos (por exemplo, com que frequência o usuário acessa um aplicativo);
- Características do usuário (por exemplo, padrões de digitação);
- Impressão digital (TouchID);
- Reconhecimento facial (FaceID).

Dessa forma, mesmo que um atacante consiga roubar as credenciais de um usuário, e também consiga comprometer um código OTP de segundo fator, é possível acreditar que um ou mais fatores adicionais (como os mencionados acima) revelem que na verdade é um invasor tentando fazer login. Por exemplo, se o atacante não usar o mesmo tipo de celular que o usuário legítimo, então, ao incorporar características sobre o dispositivo como parte da verificação de autenticação, a tentativa de login pode ser bloqueada.

A.6.5 Proteger os Domínios com SPF, DKIM e DMARC

Quando bancos começaram a oferecer serviços aos consumidores online, alguns dos ataques de *phishing* mais populares envolviam o envio de e-mails que se faziam passar pelo banco para os consumidores, solicitando que clicasse em um link no e-mail para fazer login no site do banco. O e-mail poderia ser enviado a partir do domínio do banco no campo "De:"do e-mail (por exemplo, "De: Suporte ao Cliente<suporte@banco.com>"), pois o remetente não é autenticado como parte do SMTP. Usuários que recebem esses e-mails de *phishing* tendem a acreditar que um e-mail está realmente sendo enviado pelo banco quando o domínio pode ser facilmente falsificado (DASWANI; ELBAYADI, 2021).

Existem alguns protocolos que dificultam a falsificação de e-mails de organizações. Para evitar que um golpista envie um e-mail ilegítimo se passando por uma organização, é possível utilizar os padrões SPF, DKIM e DMARC.

O SPF (Sender Policy Framework) permite que organizações especifiquem quais endereços de IP estão autorizados a enviar e-mails em nome da organização. A lista de IPs autorizados pode ser adicionada aos registros DNS da organização. Os registros DNS são normalmente usados para traduzir nomes de domínio em endereços IP quando um cliente deseja se conectar a um servidor.

Com o SPF, quando um programa de e-mails recebe uma mensagem, ele pode procurar qual é o endereço IP autorizado correspondente ao domínio do remetente e verificar se o IP real de onde a mensagem foi enviado é o mesmo que o IP listado em seu registro DNS. Se for, o e-mail é considerado válido. Caso contrário, o e-mail é rotulado como spam ou é excluído. O DMARC (Domain-based Message Authentication, Reporting, and Conformance, ou Mensagem Baseada em Domínio de Autenticação, Relatório e Conformidade) pode ser utilizado para especificar o que um programa de e-mail deve fazer se o endereço IP não corresponder (DASWANI; ELBAYADI, 2021).

Ao invés de apenas autenticar e-mail com base no endereço IP, o protocolo DKIM (DomainKeys Identified Mail) pode ser usado para assinar digitalmente todas as mensagens legítimas originadas de uma organização, utilizando criptografia de chaves públicas e privadas para assinar as mensagens . Os e-mails são assinados usando uma chave provida conhecida apenas pela organização. A chave pública correspondente é publicada nos regis-

tros DNS da organização, e os programas de e-mail podem verificar as assinaturas digitais em e-mails assinados com DKIM usando a chave pública (DASWANI; ELBAYADI, 2021).

A.6.6 Domínios Parecidos

Atacantes podem forjar e-mails de domínios semelhantes. Por exemplo, utilizar "site5eguro.com" ao invés de "siteseguro.com".

Para criar o e-mail de *phishing* mais convincente possível, pode-se imaginar que um atacante deseje que esse e-mail pareça ser de "siteseguro.com". No entando, se SPF, DKIM e DMARC forem utilizados, um atacante pode ter que recorrer ao uso de um domínio semelhante, como "site5eguro.com" a ponto da diferença não ser percebida rapidamente. Além disso, os atacantes podem aproveitar vários conjuntos de caracteres internacionais para registrar domínios parecidos que tenham caracteres extremamente similares aos caracteres do idioma correspondente, tornando a diferença ainda menos perceptível. Por fim, um atacante pode até criar registros SPF, DKIM e DMARC para os domínios semelhantes a fim de aumentar a legitimidade de seus e-mails de ataque (DASWANI; ELBAYADI, 2021).

Além de proteger seus domínios legítimos contra falsificações, é importante também registrar proativamente o maior número possível de domínios semelhantes e erros ortográficos dos domínios de sua organização. Também é uma boa prática monitorar se e quando atacantes registram domínios semelhantes aos da organização. Como pode haver um número infinito de variações, dados os muitos conjuntos de caracteres internacionais, serviços de monitoramento de marca e domínio podem ser usados para ajudar na detecção de registros de domínios maliciosos.

Uma última dica é ajudar os funcionários a identificar quando os e-mails não são enviados por partes internas legítimas, marcando o e-mails como "externos", inserindo marcações como "[EXTERNO]" na linha de assunto do e-mail. Essas marcações podem ser implementadas identificando mensagens que não tenham uma assinatura DKIM válida pela própria organização como externas (DASWANI; ELBAYADI, 2021).

A.6.7 Preenchimento de Credenciais e Controle de Conta

Sendo o objetivo de um atacante adquirir credenciais válidas usuários, uma vez obtidas essas informações, eles podem tentar assumir a conta do usuário. Atacantes podem também comprar grandes quantidades de credenciais na *dark web*. Mesmo que essas credenciais adquiridas não sejam para um site sensível específico, os usuários tendem a reutilizar as mesmas senhas em vários sites, e as senhas roubadas nas várias violações que já ocorreram podem funcionar nesses sites sensíveis.

Quando os atacantes testam grandes números de credenciais roubadas em sites-alvo, ocorre um ataque chamado "preenchimento de credenciais". Para se proteger desse tipo de ataque é necessário, principalmente, identificar o uso de robôs e verificar o uso por funcionários de senhas roubadas (DASWANI; ELBAYADI, 2021).

A detecção anti robôs identifica tentativas automatizadas de atacantes de usar credenciais roubadas em um site. Algumas empresas como a Cloudflare oferecem soluções de detecção e mitigação anti robôs que não apenas ajudam a detectar tentativas de ataque de preenchimento de credenciais em um site, como também ajudam a lidar com *scrapping* e controle de contas, que é quando um ataque de preenchimento de credenciais resulta em um login bem-sucedido (DASWANI; ELBAYADI, 2021).

Como os funcionários podem reutilizar as mesmas senhas tanto na rede corporativa quanto em contas pessoais online, é importante identificar quando isso acontece e fazer com que alterem as senhas corporativas. Atacantes inevitavelmente testarão credenciais roubadas na rede corporativa. Tudo que um atacante precisa é encontrar um funcionário que tenha reutilizado tal senha para assumir uma conta na rede corporativa, obter acesso ao e-mail corporativo e depois tentar expandir sua presença na rede usando o conteúdo disponível na caixa de entrada do funcionário comprometido.

Para identificar casos de reutilização de senha, é necessário verificar todas as senhas dos funcionários em dumps de senha da *dark web*. Alguns serviços online podem ajudar nessa tarefa (DASWANI; ELBAYADI, 2021). É importante também que em cada vez que um funcionário altere sua senha, essa nova senha seja verificada em repositórios de senhas roubadas.

A.6.8 Gerenciadores de Senhas

Gerenciadores de senhas são aplicativos que funcionários e usuários podem usar para gerar e gerenciar automaticamente senhas fortes e complexas, únicas para cada site que utilizam. Alguns gerenciadores de senhas também tem integração com navegadores, de modo que o gerenciador pode ajudar a verificar se um site é legítimo antes de enviar uma senha para ele. Quando um gerenciador de senhas gera uma senha para um site no momento do registro, ele também pode registrar o domínio do site legítimo. Se posteriormente o usuário clicar em um link para um site impostor, o site impostor não corresponderá ao domínio legítimo, e o gerenciador de senhas pode aconselhar o usuário a não enviar suas credenciais. Como o gerenciador faz essa verificação com técnicas byte a byte, ele não será enganado por domínios parecidos (DASWANI; ELBAYADI, 2021).

Gerenciadores de senhas exigem o uso consistente por parte dos funcionários para cada site online para funcionar como uma defesa eficaz. Para se defender de um ataque

de *phishing* é necessário acertar todas as vezes. Para um atacante ter sucesso, basta um deslize para se ter um comprometimento inicial.

A.6.9 Defesas Adicionais

Além de defesas e estratégias de mitigação orientadas a tecnologia, existem também defesas e estratégias de mitigação orientadas a processos e pessoas.

A.6.9.1 Treinamento e Testes

Um treinamento *anti-phishing* é um treinamento para funcionários no qual eles são ensinados sobre os sinais comuns a serem observados em e-mails que podem indicar ataques de *phishing* ou *spear phishing*. Alguns exemplos desses sinais são domínios desconhecidos ou incorretos usados no remetende ou em links em um e-mail, anexos inesperados ou pedidos que incluem um senso artificial de urgência, como por exemplo, "você precisa responder imediatamente ou sua conta será desativada"(DASWANI; ELBAYADI, 2021).

Esse treinamento geralmente é seguido por testes de *phishing* enviados pela equipe de segurança da organização. Esses testes são elaborados para parecerem com e-mails de *phishing* de atacantes, com a exceção de serem inofensivos, e possuirem o objetivo de verificar se os funcionários conseguem evitá-los. Se um funcionário abrir um e-mail de teste de *phishing*, clicar em um link no e-mail, clicar em um anexo ou inserir suas credenciais em um site impostor vinculado ao e-mail, ocorre um momento de aprendizado em que o funcionário é informado de que caiu em um ataque de *phishing* de teste. Se o funcionário não cai no ataque, mas ao invés disso relata o e-mail à equipe de segurança da organização, é um sinal de que o trinamento *anti-phishing* está funcionando, e o funcionário está menos suscetível a um ataque deste tipo.

Com treinamento suficiente, os funcionários podem se tornar parte da linha de defesa da organização, sendo o último elemento na detecção de ataques de *phishing* que não foram filtrados da caixa de entrada dos funcionários por meio de outras contramedidas (DASWANI; ELBAYADI, 2021).

Realizando treinamentos *anti-phishing* pelo menos uma vez ao ano e enviando testes de *phishing* periodicamente, deve ser possível medir se os funcionários de uma organização estão realmente se tornando menos suscetíveis a ataques desse tipo ao longo do tempo.

A.6.9.2 Verificações de Complexidade de Senha

Alguns sistemas exigem que os funcionários ou usuários definam senhas que atendam a um conjunto de restrições de complexidade como número de caracteres, possuir letras e números, letras maiúsculas e minúsculas ou símbolos especiais.

Essas verificações de complexidade de senha ajudam a evitar que os usuários escolham senhas frequentemente usadas, mas fracas, como “12345678” ou “senha”. No entanto, essas verificações de complexidade por si só não são suficientes. Dada a quantidade de combinações que um processador pode tentar em um período relativamente curto, é, de certa forma, fácil realizar ataques de força bruta, onde se descobre a senha apenas por tentar todas as combinações possíveis.

Entretanto, verificações de complexidade de senha não são recomendadas pelas diretrizes do Instituto Nacional de Padrões e Tecnologia, pois levam a um comportamento de senha inadequado a longo prazo. Além disso, sua eficácia contra a maioria dos ataques sofisticados também é baixa (DASWANI; ELBAYADI, 2021). Caso esse tipo de verificação seja utilizado, o ideal é que seja em conjunto com outras defesas.

A.6.9.3 Rotação de Senhas

Alguns sistemas exigem que os usuários troquem suas senhas periodicamente. Por exemplo, em algumas empresas, os funcionários devem trocar suas senhas a cada 90 dias. Geralmente, os funcionários consideram essas políticas como uma inconveniência, e elas não são muito eficazes, já que os funcionários tendem a escolher senhas semelhantes às anteriores ou simplesmente alternam entre duas ou três senhas apenas. As orientações do Instituto Nacional de Padrões e Tecnologia também desaconselham esse tipo de prática, pois também leva a um comportamento inadequado de senha a longo prazo (DASWANI; ELBAYADI, 2021).

A.7 Ferramentas de Terceiros

Muitas violações de dados começam com comprometimentos envolvendo terceiros. Dificilmente uma organização consegue funcionar completamente sem utilizar nenhuma ferramenta de terceiros. Terceiros se enquadram como fornecedores ou parceiros e geralmente, quanto maior a organização, mais ela pode depender de terceiros. Um exemplo é o vazamento de dados que ocorreu na organização SolarWinds, em 2021, onde aproximadamente 18 mil organizações que tinham a SolarWinds como terceira podem ter sido afetadas(LAZAROVITZ, 2021).

Se um terceiro tiver acesso a dados ou acesso à rede e esse terceiro for comprometido, a organização que está trabalhando com esse terceiro provavelmente será comprometida também. Se um terceiro não cumprir com requisitos de segurança, isso pode comprometer os requisitos de segurança da organização em si. Todo terceiro pode se tornar o elo mais fraco (DASWANI; ELBAYADI, 2021).

A.7.1 Segurança de Fornecedores

O tipo mais comum de terceiro é um fornecedor. Para avaliar e gerenciar o risco devido a fornecedores, é essencial fazer um inventário de todos os relacionamentos com fornecedores de terceiros. Dependendo do que é adquirido deles e da natureza do relacionamento, pode ser necessário avaliar a segurança desse terceiro no momento em que o contrato inicial é criado e periodicamente depois.

Por exemplo, se alguma informação sensível é compartilhada com fornecedores, um vazamento que acometer esse fornecedor pode significar um vazamento na organização, já que a organização é a fonte dos dados vazados (DASWANI; ELBAYADI, 2021).

Alguns pontos que devem ser considerados nas relações com fornecedores são:

- Quais dados serão fornecidos e quais serão recebidos? Esses dados são sensíveis? Dados sensíveis estão sendo fornecidos ao fornecedor? Caso esse fornecedor seja comprometido, a organização também estará comprometida?
- Qual o nível de acesso a recursos da organização que o fornecedor tem? Acesso à rede? Acesso à conta? APIs?

Dependendo das respostas para essas perguntas, é necessário considerar como os dados serão armazenados e criptografados pelo fornecedor, como os dados serão protegidos em repouso e em tráfego, como as organizações se comunicarão, como os segmentos de rede que lidam com dados sensíveis serão segregados do restante da rede do fornecedor.

É recomendável pesquisar sobre um fornecedor antes de fechar contrato com ele, buscando por outros projetos em que o mesmo participou, além de auditorias já realizadas e buscando por *feedbacks* de outras organizações que já trabalharam com esse fornecedor.

Dependendo do quanto é necessário utilizar dos serviços de um fornecedor, pode ser necessário assinar ou renovar o contrato com esse fornecedor mesmo que alguns requisitos de segurança não estejam sendo cumpridos. Nesses casos, é importante monitorar os serviços que passam por esse fornecedor, assim como os dados, e periodicamente realizar monitoramentos para certificar que os dados continuam seguros ou promover melhorias (DASWANI; ELBAYADI, 2021).

A.7.2 Desenvolvedores, Parceiros e Clientes

Se uma organização permite que desenvolvedores de terceiros ou parceiros acessem dados sobre os clientes, é importante avaliar e monitorar as atividades desses terceiros.

Embora seja comum e difundido que as práticas de segurança dos fornecedores que recebem dados sensíveis devem ser avaliadas, o mesmo vale para os clientes. Se a organização vende dados para seus clientes e um desses clientes for comprometido, o

incidente pode ser atribuído à organização que vendeu os dados. Sendo assim, além de avaliar a segurança de fornecedores, é importante também avaliar a segurança dos clientes antes de fornecer dados a eles (DASWANI; ELBAYADI, 2021).

A.8 Centralização de Rede

Uma rede centralizada é uma rede que foi construída ao redor de um único servidor central, que lida com o gerenciamento e o processamento dos dados e das funcionalidades de um sistema. Apesar dessa arquitetura proporcionar uma implantação mais rápida e fácil, um gerenciamento mais fácil de usuários e ser mais barata, ela também tem a desvantagem de possuir um único ponto de falha em toda a rede, ou seja, se o servidor for invadido, toda a rede será invadida (SYSTEMS, 2021).

Com todos os dados de um sistema reunidos em um único servidor, significa que tanto dados não sensíveis quanto dados sensíveis estão compartilhando o mesmo espaço, representando um risco de segurança. Nessa situação, se um atacante conseguir acessar o centro de dados único, ele terá acesso a tudo a partir desse local, ao invés de precisarem enfrentar o obstáculo de se mover por diferentes servidores para encontrar o que desejam (SYSTEMS, 2021).

Tendo isso em mente, uma recomendação é implementar uma rede descentralizada, com mais de um servidor, onde os servidores são independentes. Nessa arquitetura, se um servidor cair ou for invadido, não afetará os outros servidores e nem permitirá que um atacante tenha acesso a informações fora do servidor invadido (MEYER et al., 1995).

Além da proteção contra invasões generalizadas, redes descentralizadas contam com o benefício de serem mais fáceis de escalar, a medida que o sistema cresce. Há também menos gargalos em uma rede descentralizada, pois o tráfego total do sistema está distribuído em diferentes servidores, a depender da funcionalidade demandada (MEYER et al., 1995).

A.9 Vulnerabilidades de Software

A engenharia de software é uma área considerada por muitos como difícil e exige competências como criatividade, lógica e certas vezes grande complexidade. Soma-se a isso o fato de ser uma área relativamente nova se comparada a outras engenharias como a engenharia civil (WIRTH, 2008). A maioria dos governos ainda não adota códigos de desenvolvimento de software seguro e para a maioria dos casos, não é necessário possuir uma licença para desenvolver aplicações de software que sustentam redes elétricas, comércio eletrônico ou sistemas de comunicação, além de outras áreas consideradas de infraestrutura crítica (VERMA et al., 2013). Desenvolvedores e usuários de software devem

ter ciência que todo software possui vulnerabilidades, seja por conta de bugs ou falhas de design (MCGRAW, 2006).

Em organizações que desenvolvem a própria aplicação, o software pode estar suscetível às chamadas vulnerabilidades de software de primeira parte, ou seja, vulnerabilidades que estão presentes no próprio código da aplicação da organização e que podem ser exploradas por agentes maliciosos (DASWANI; ELBAYADI, 2021).

Já em organizações que utilizam software desenvolvido por terceiros, o que corresponde à maioria delas (Ponemon Institute, 2018a), é necessário estar atento às vulnerabilidades de software de terceiros. Essas vulnerabilidades podem ser identificadas tanto pela organização desenvolvedora do software quanto por outros pesquisadores de segurança em algum momento após seu lançamento.

A.9.1 Vulnerabilidades de Primeira Parte

A.9.1.1 Fase de Desenvolvimento

Na fase de desenvolvimento de uma aplicação, a melhor prática para identificar proativamente vulnerabilidades de design de segurança é realizar uma revisão do design de segurança do produto (DASWANI; ELBAYADI, 2021). Este processo envolve revisar os documentos de design e arquitetura do produto antes mesmo do desenvolvimento em código começar. Defeitos identificados nesta etapa podem economizar custos de manutenção no futuro, assim como a dor de cabeça causada por um defeito exposto após o lançamento do produto. Uma fonte relevante para agregar segurança ao produto é o Centro IEEE para Design Seguro (MCGRAW, 2014), que fornece códigos de desenvolvimento para assuntos como internet das coisas e software para dispositivos médicos, além de orientações que auxiliam a evitar falhas de design de segurança em softwares.

Depois do desenvolvimento do código-fonte, testes estáticos de segurança auxiliam a encontrar vulnerabilidades no código sem a necessidade de executá-lo (Microsoft, 2023b). Vulnerabilidades como estouro de buffer e injeção de código são exemplos de problemas que testes estáticos ajudam a identificar, a partir de características do código-fonte (LI, 2020).

Além dos testes estáticos, análises de composição de software também auxiliam na identificação do uso de componentes de terceiros, como bibliotecas ou *frameworks*, que foram escritos pela própria organização (IMTIAZ; THORN; WILLIAMS, 2021).

Por fim, revisões manuais de código devem ser realizadas por desenvolvedores, sejam eles internos ou externos, para identificar vulnerabilidades no código-fonte. Trechos de código que realizam funções sensíveis à segurança dos dados como manipulação ou criptografia deles precisam ser revisados manualmente. Apesar dessa prática consumir tempo, é

necessário alocar um revisor especializado neste processo para identificar vulnerabilidades sutis (DASWANI; ELBAYADI, 2021).

A.9.1.2 Fase de Testes

Após o desenvolvimento de um sistema, ele precisa ser testado, seja por testes unitários (RUNESON, 2006) ou por um conjunto de testes onde o sistema maior tem responsabilidade, como testes de integração (LEUNG; WHITE, 1990). Além destes, testes dinâmicos e interativos de segurança também agregam na busca de vulnerabilidades durante o desenvolvimento do produto (Microsoft, 2023b).

Testes dinâmicos de segurança têm como objetivo procurar vulnerabilidades executando conjuntos automatizados de testes no código-fonte. Esses testes são considerados como testes de caixa-preta, já que envolvem o envio de entradas para o programa em execução e a observação de saídas, sem envolver uma análise do próprio código em si, ao contrário dos testes estáticos, que são classificados como caixa-branca e procuram por vulnerabilidades analisando o código-fonte (DENCHEVA, 2022).

Testes interativos são semelhantes aos testes dinâmicos, mas ao invés de testar conjuntos de testes automatizados pré elaborados, é combinado testes humanos com os automatizados para tentar encontrar vulnerabilidades. Testes interativos são classificados como caixa-branca já que costumam analisar também o código-fonte, de modo que, ao encontrar uma vulnerabilidade, é possível identificar o local no código que produziu a vulnerabilidade. É importante ter em mente que testes interativos exigem um conjunto abrangente de testes automatizados e manuais para resultar em conclusões assertivas (PAN, 2019).

Testes de penetração consistem em pessoas qualificadas no assunto de testes que utilizam um conjunto de técnicas de teste, como as dinâmicas ou interativas para tentar encontrar vulnerabilidades na aplicação. Geralmente esse tipo de teste é executado um pouco antes do lançamento do produto e tem como objetivo simular ataques de agentes maliciosos, preparando o produto para o contato com os variados tipos de pessoas que podem utilizar a aplicação (BISHOP, 2007).

Integração contínua e implantação contínua (CI/CD) são conceitos que se tornaram muito populares no desenvolvimento de software, promovendo agilidade no desenvolvimento e refatoração de aplicações, além de velocidade na implantação das mesmas. Esses conceitos tratam de pequenas alterações frequentes em repositórios de código e automação na implantação do código nos mais diversos ambientes, automatizando tanto a testagem, fazendo uso de testes contínuos, quanto a implantação do programa em produção (BOBROVSKIS; JURENOKS, 2018).

Ao construir pipelines de CI/CD, os desenvolvedores podem escolher quais ferramentas de testes estáticos e dinâmicos irão utilizar para monitorar as frequentes e contínuas modificações de código em desenvolvimento (DASWANI; ELBAYADI, 2021).

A.9.1.3 Fase de Produção

Depois que um projeto de software foi testado e colocado em produção, técnicas de autodefesa da aplicação em tempo de execução podem ser utilizadas para identificar ataques que tentem explorar vulnerabilidades não descobertas pelos desenvolvedores nas fases anteriores. As tecnologias de autodefesa analisam a entrada real dos usuários e monitoram ou bloqueiam ataques realizados (SALEMI; SADRE; LEGAY, 2020).

Nesta fase, também é possível realizar testes de penetração para encontrar novas vulnerabilidades (BISHOP, 2007), mas nesse momento, é necessário ter extrema cautela ao fazê-lo, já que a exploração de uma nova vulnerabilidade em ambiente de produção pode colocar os dados reais em risco.

A.9.2 Vulnerabilidades de Terceiros

Muitos negócios dependem de software hoje em dia (Ponemon Institute, 2018a), entretanto, todos os softwares possuem bugs, e alguns desses bugs podem representar vulnerabilidades de segurança que precisam ser identificadas (DASWANI; ELBAYADI, 2021). Dessas vulnerabilidades, as de maior gravidades precisam ser corrigidas ou mitigadas o quanto antes, para que uma exploração da vulnerabilidade não aconteça e consequentemente, um vazamento de dados.

A.9.2.1 Identificação e Validação

A principal tecnologia necessária para identificar vulnerabilidades em software de terceiros é o escâner de vulnerabilidades. Esses escâneres conseguem analisar a rede e identificar quais máquinas estão em execução e quais softwares estão sendo utilizados nelas (DASWANI; ELBAYADI, 2021). Esse processo é realizado através dos endereços de IP disponíveis na rede escaneada, se comunicando com as aplicações em execução em cada porta desses endereços, obtendo informações sobre as aplicações por meio de respostas e comportamentos. Com base nessas informações, o escâner consegue identificar vulnerabilidades nas aplicações que estão em execução na rede (HOLM et al., 2011).

Após identificar as vulnerabilidades, é necessário validá-las, para descartar situações de falso positivo. Depois de validada, cada vulnerabilidade precisa ser rastreada, pois deixar de resolver uma vulnerabilidade que seja pode ser o suficiente para que um atacante comprometa o sistema (DASWANI; ELBAYADI, 2021).

Além dos falsos positivos, é preciso lidar também com os falsos negativos, que são situações onde o escâner não detectou uma vulnerabilidade. Para evitar esse tipo de problema, é importante que uma organização faça uso de mais de um escâner simultaneamente em sua rede (DASWANI; ELBAYADI, 2021).

A.9.2.2 Priorização

O rastreamento de vulnerabilidades por parte dos escâneres pode resultar na identificação de muitas vulnerabilidades, com isso, se faz necessário priorizar quais vulnerabilidades serão corrigidas primeiro. Para isso deve-se analisar questões como existência da vulnerabilidade em várias organizações simultaneamente, existência de ataques para a vulnerabilidade específica, facilidade de exploração da vulnerabilidade, etc (SHARMA; SIBAL; SABHARWAL, 2021). O mercado oferece uma variedade de produtos que auxiliam as organizações a priorizarem o risco das vulnerabilidades de terceiros, levando em consideração o contexto da arquitetura de rede da organização.

Tendo priorizado as vulnerabilidades, é necessário determinar o esforço necessário para corrigir as vulnerabilidades de maior risco, testar a correção e avaliar o impacto que a implementação dessa correção terá nos usuários e nos sistemas dependentes (WAGNER, 2021). É importante considerar de forma cuidadosa esses impactos, pois a correção de uma vulnerabilidade pode exigir um grande esforço em diversas áreas do sistema dependentes do ponto corrigido, podendo ser necessário estimar, desenvolver, testar e implantar novas alterações em ambientes de produção (DASWANI; ELBAYADI, 2021).

A.9.2.3 Atualização de *Endpoints*

Uma classe de vulnerabilidades importante que precisa ser gerenciada são as vulnerabilidades nos pontos finais (*endpoint*), dispositivos ou aplicações finais em um canal de comunicação (WATERSON, 2020). Fornecedores de sistemas operacionais por exemplo, frequentemente identificam vulnerabilidades em seus softwares.

Correções para corrigir vulnerabilidades em *endpoints* devem ser implementadas regularmente. Sistemas operacionais disponibilizam atualizações ou correções regularmente, e estas devem ser aplicadas nos *endpoints* da rede com certa frequência, para garantir que uma vulnerabilidade que já possui correções disponíveis não se mantenha exposta na rede por muito tempo (WATERSON, 2020).

A.10 Considerações Finais

Tendo apresentado formas de evitar as principais causas raiz que assolam grandes empresas quando se fala de exposição de dados sensíveis, têm-se a esperança de que esse material possa ajudar desenvolvedores e líderes de organizações a mitigar o risco de terem seus dados comprometidos.

Quanto mais informados forem os agentes que desenvolvem ou manipulam os sistemas de dados, mais raros serão os casos de organizações, grandes ou pequenas, passando por violações de dados. E quanto mais o conhecimento puder ser propagado, tanto dentro da própria organização quanto entre organizações, mais será possível construir uma cultura de segurança na comunidade de software.

Em um mundo onde constantemente vê-se uma evolução de ameaças cibernéticas, a educação contínua dos agentes envolvidos é crucial. Medidas como treinamentos regulares de conscientização e simulações de ataques não apenas fortalecem as defesas de uma organização como também capacitam os membros de equipes a se tornarem defensores ativos da segurança de dados.

A implementação eficaz das medidas discutidas neste guia requer uma abordagem abrangente e sistemática. As políticas de segurança de uma organização devem ser reavaliadas periodicamente, com uma integração contínua de tecnologias de segurança mais recentes, sabendo que o nível das ameaças também evoluem periodicamente.

Por fim, deve estar claro que a prevenção de exposições de dados sensíveis é uma responsabilidade compartilhada por todos os membros de uma organização. Cada medida proposta neste guia propõe um avanço em direção a um ambiente cibernético mais seguro e resiliente.

O fim deste guia traz então um chamado para que as organizações adotem uma postura proativa de um bem tão valioso: os dados sensíveis de seus clientes e parceiros. A segurança cibernética não é apenas uma prioridade, é um compromisso constante com a confiança e a integridade.