JUAREZ DE OLIVEIRA

UMA ARQUITETURA DE SEGURANÇA UTILIZANDO COMPUTAÇÃO CONFIÁVEL E ZERO TRUST

CURITIBA
2024

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central
Edilene de Oliveira dos Santos CRB 9 / 1636

O48a
2024

Oliveira, Juarez de
Uma arquitetura de segurança utilizando computação confiável e zero trust / Juarez de Oliveira ; orientador: Eduardo Kugler Viegas ; coorientador: Altair Olivo Santin -- 2024
68 f. : il. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2024
Bibliografia: f. 65-67

1. Informática. 2. Computadores – Medidas de segurança. 3. Validação de dados. 4. Software – Confiabilidade. I. Viegas, Eduardo Kugler. II. Santin, Altair Olivo. III. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática. IV. Título

CDD 20. ed. – 004

Pontifícia Universidade Católica do Paraná
Escola Politécnica
Programa de Pós-Graduação em Informática

Curitiba, 15 de maio de 2024.

36-2024

DECLARAÇÃO

Declaro para os devidos fins, que **Juarez de Oliveira** defendeu a dissertação de Mestrado intitulada "**UMA ARQUITETURA DE SEGURANÇA UTILIZANDO COMPUTAÇÃO CONFIÁVEL E ZERO TRUST**", na área de concentração Ciência da Computação no dia 28 de fevereiro de 2024, no qual foi aprovado.

Declaro ainda, que foram feitas todas as alterações solicitadas pela Banca Examinadora, cumprindo todas as normas de formatação definidas pelo Programa.

Por ser verdade firmo a presente declaração.

_____
Prof. Dr. Emerson Cabrera Paraiso
Coordenador do Programa de Pós-Graduação em Informática

JUAREZ DE OLIVEIRA

UMA ARQUITETURA DE SEGURANÇA UTILIZANDO COMPUTAÇÃO CONFIÁVEL E ZERO TRUST

Projeto de Dissertação de Mestrado apresentado ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

Orientador: Dr. Eduardo Kugler Viegas
Coorientador: Dr. Altair Olivo Santin

Pontifícia Universidade Católica do Paraná
Programa de Pós-Graduação em Informática

CURITIBA
2024

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CWE	Common Weakness Enumerations
CIS	Center for Internet Security
CSRF	Cross site request forgery
CVE	Common Vulnerabilities and Exposures
HOTP	Time-based One-Time Password
IAM	Identity and Access Management
IdP	Identity Provider
MFA	Multiple Factors-Authentication
NIST	National Institute of Standards and Technology
OTP	One Time Password
OWASP	Open Worldwide Application Security Project
PAM	Privileged Acess Management
SAML	Security Assertion Markup Language
SASE	Secure Access Service Edge
SGX	Intel Software Guard Extension
SP	Service Provider
SSH	Secure Shell
TEE	Trusted Execution Environment
TOTP	HMAC-based One-Time Password
VPN	Virtual Private Network
XSS	Cross-site scripting
ZT	Zero Trust
ZTNA	Zero Trust Network Access

SUMÁRIO

“O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001”.

“This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001".

DEDICATÓRIA

Dedico este trabalho à minha família, em especial à minha esposa Irene, pelo apoio incondicional durante o período de estudos.

AGRADECIMENTOS:

Agradeço fortemente aos meus orientadores Prof. Dr. Eduardo Kugler Viegas e Prof. Dr. Altair Olivo Santin, por terem acreditado no meu potencial e me guiado pelo árduo caminho da pesquisa científica.

RESUMO

É recomendado que o acesso a aplicações críticas utilize autenticação com múltiplos fatores, protegida por ambiente de execução confiável (TEE) em diferentes canais de comunicação. Adicionalmente, os segredos de acesso devem ser protegidos em um Vault que armazena os usuários privilegiados e códigos de acesso, diminuindo a exposição de credenciais, além de permitir a alteração periódica destes segredos. Neste trabalho, analisamos e empregamos estas camadas de proteção e adicionamos o *One-Time Password* para proteção do ambiente, sem necessidade de intervenção humana e sem diminuir o tempo de validade das credenciais do cofre de senhas. Para fortalecer as camadas de segurança foi implementado o conceito de Zero Trust, sendo demonstrado por meio do modelo do adversário e *pentest* que o ambiente é seguro em aplicações web e de maneira interativa com o *remote shell*.

Palavras-chave: One-Time Password, Trusted Execution Environment, Credential Vault, Mitre Att&ck, Zero Trust.

ABSTRACT

It is recommended that access to critical applications use multi-factor authentication, protected by a trusted execution environment (TEE) using different communication channels. Additionally, access secrets must be protected in a Vault that stores privileged users and access codes, reducing the exposure of credentials, in addition to allowing these secrets to be changed periodically. In this work we analyze and employ all these layers of protection and add OTP (One-Time Password) to protect the environment without requiring human intervention and without decreasing the validity time of Vault credentials. To strengthen the security layers, the concept of Zero Trust was implemented, demonstrating through the adversary model and pentest that the environment is secure in web applications and interactively with the remote shell.

Keywords: *One-Time Password, Trusted Execution Environment, Credential Vault, Mitre Att&ck, Zero Trust*

Capítulo 1
Introdução

Este capítulo apresenta a contextualização do trabalho, motivação, objetivos e contribuição científica da pesquisa.

1.1. Contextualização

Tradicionalmente, cibersegurança está preocupada com a construção de diversas barreiras de segurança, principalmente em redes de computadores, em especial no emprego de *firewalls*. Além disto, as aplicações utilizam de criptografia no hardware, em sistemas, armazenamento e no transporte de dados.

Recentemente, a indústria tem desenvolvido equipamentos com *softwares* que permitem a execução em ambiente considerado seguro, por meio dos enclaves. No caso do hardware, o TEE *(Trusted Execution Environment)* ou Ambiente de Execução Confiável, oferece recurso que permite a execução de códigos contendo segredos, sendo armazenados na memória de maneira criptografada.

O objetivo do uso do TEE é impedir que *softwares* maliciosos, não autorizados, tenham acesso a áreas de memória onde estejam armazenados os segredos ou informações sensíveis, como números de cartões de crédito, por exemplo.

Mesmo em ambientes mais protegidos por mecanismos de segurança, incluindo a criptografia, o roubo de credenciais ou de sessões autenticadas, podem se tornar frequentes. A autenticação de usuários com *multi-factor authentication* (MFA) é um dos métodos mais adotados para mitigar estes problemas (CIS Security, 2023).

Entretanto, a autenticação MFA não está imune a violação por golpes, sendo necessário um bom treinamento dos usuários para que consigam se proteger, em especial de engenharia social.

A autenticação centralizada, em especial no modelo SSO (*single-sign-on*), é considerada boa prática pelos principais *frameworks* de cibersegurança, como o C*IS*

Controls (CIS Security, 2023) e o *MITRE ATT&ACK* (Mitre, 2023), isto se deve ao fato de que é bastante desafiador para o usuário, poder gerenciar logins e senhas para cada aplicação que possua acesso dentro do ambiente corporativo, por exemplo. Permitir que o usuário se conecte a diversas aplicações sem pedir uma nova senha a todo instante, melhora a experiência do usuário e diminui o tráfego de *logins* e senhas pelas redes.

Ao usar o modelo centralizado de gerenciamento de identidade e acesso (IAM – *Identity and Access Management*), os administradores de rede têm mais controle sobre o tempo de expiração da senha e os dispositivos MFA (múltiplos fatores de autenticação), bloqueando dispositivos suspeitos que tentam acesso não autorizado e verificar as permissões dos usuários dentro de um sistema.

Além disso, o IAM pode usar uma infraestrutura interna (geralmente não recomendada) ou externa (recomendado) de um Provedor de identidade (IdP). A justificativa é que é difícil para um invasor controlar um servidor IAM porque o movimento lateral é complicado, considerando um domínio IdP remoto e seguro.

A movimentação lateral, no contexto da segurança cibernética, refere-se ao movimento de um invasor dentro da rede de uma organização após ter obtido acesso inicial a um sistema. Em vez de avançar diretamente em direção ao seu alvo final, o invasor se move lateralmente pela rede, explorando e comprometendo outros sistemas e recursos internos. Por meio deste movimento lateral, um adversário pode obter acesso não autorizado a um dispositivo ou sistema por meio de uma falha no aplicativo ou até mesmo roubo de credenciais. Posteriormente, ele pode se mover por outros sistemas ou dispositivos, ganhando mais poder sobre credenciais privilegiadas.

O Gerenciamento de Acesso Privilegiado (PAM) fornece proteção das credenciais privilegiadas por meio de um componente Vault (cofre) de senha (Cheng, Li, Wang, Chu, & Liang, 2021). Ao focar na redução da superfície de ataque e no roubo de credenciais, é importante considerar um prazo de validade para as credenciais (senha/código). No contexto deste trabalho, um *vault* é a função principal de um PAM, também conhecido como *password safe* ou cofre de senhas.

A imposição de um prazo de validade para credenciais pode aumentar a segurança, mas diminui a usabilidade e a produtividade, devido à necessidade de novas melhores práticas para proteger essas credenciais. Por exemplo, um tempo de expiração de um segredo comumente utilizado no mercado é de 8 horas. Isto significa que um invasor pode

causar danos ao sistema durante esse período se ele ou ela se esconder ou for um mau funcionário (um *insider*).

Outra possibilidade é que um hacker ataque o PAM para obter o segredo e acessar um sistema. Neste caso, o invasor teve 8 horas para tentar. Portanto, a validade das credenciais é complicada pois se o administrador reduzir este tempo, o usuário terá que alterar os segredos (senha/código) no *vault* com maior frequência, assim que a validade expirar.

Este trabalho propõe um mecanismo seguro para manter a usabilidade e a produtividade, ao mesmo tempo que melhora a segurança do *vault* de forma não interativa.

Para diminuir ainda mais a superfície de ataque podemos utilizar o conceito de *Zero Trust* (Bobbert & Scheerder, 2022), permitindo somente o acesso ao recurso computacional ao usuário previamente autenticado e autorizado, ou seja, não seria possível a execução de uma varredura nas portas do ativo para localizar eventuais vulnerabilidades ou serviços abertos.

Neste trabalho busca-se responder à seguinte questão: Como é possível prover um mecanismo para aumentar a segurança no processo de autenticação, de forma interativa quando possível, visando a *non-interference* na produtividade e usabilidade de acordo com a abordagem de uso?

1.2. Motivação

Apesar da aplicação de diversos tipos de controles e ferramentas, o número de ataques cibernéticos bem-sucedidos e que levam a perda de dados, com consequente perda financeira e de imagem para as organizações, permanece em crescimento ano após ano.

A interação humana nas permissões de acessos a sistemas impõe mais um fator de risco, mesmo que os engenheiros e administradores de sistemas, considerados peças-chaves, sejam bem treinados.

A atuação de um adversário interno (“*insider*”) ou ainda falhas inerentes aos seres humanos, como distração, pouco treinamento, sobrecarga de trabalho entre outros, pode ser decisiva para a ocorrência de graves incidentes de segurança.

A criptografia de dados em uso ainda continua com um nível baixo de utilização no mundo real, o que nos permite avançar os estudos e conhecer eventuais limitações para criação de ambientes de execução confiável (TEE). Os testes realizados com a solução Intel SGX (Intel, 2023) em sistema operacional Linux mostraram-se promissoras, conseguindo proteger aplicações portadas para plataformas específicas.

No entanto, a tecnologia Intel SGX pode apresentar algumas dificuldades na sua aplicação em ambiente de produção, e tem limites de recursos de memória limitados a 512 MB por socket de processador (Muhammad El-Hindi, 2022).

Proteger um sistema para que ele tenha o menor nível de exposição possível dentro do contexto de sua utilização é de suma importância, principalmente porque é muito difícil mantê-lo sem nenhuma vulnerabilidade o tempo todo. Neste contexto, a aplicação de uma arquitetura de *Zero Trust* é o modelo indicado.

O próprio governo americano determinou que suas organizações públicas façam uso dessa arquitetura, tendo editado a norma SP-800-207 por meio do NIST (NIST, 2020). O *Zero Trust* não é uma solução de software específica, mas um conjunto de boas práticas cujo princípio é ‘nunca confie, sempre verifique’.

Portanto, publicar apenas o serviço para o qual o usuário possui permissão de acesso, verificando sua identidade e nível de permissões constantemente, enquanto considera os requisitos mínimos de segurança do dispositivo, é a abordagem adequada.

A proteção de segredos, mesmo tendo evoluído nos últimos anos, ainda requer especial atenção na implementação de qualquer arquitetura ou infraestrutura computacional. O duplo fato de autenticação pode ser estendido para APIs Rest na comunicação entre aplicações, por meio do protocolo WebAuthn (Houssain, Houssain, Houssain, Sohag, & Hahman, 2018).

Uma forma cada vez mais comum nas empresas de maior porte é o PAM (*Privileged Access Management*), um sistema que permite o uso de um cofre de senhas na administração de segredos (*vault*), como usuários para conexão de bancos de dados, acessos via SSH, APIs e outros.

O uso de um provedor de autenticação confiável pode não ser o suficiente em caso de uma aplicação mal configurada ou em alguns tipos de ataque, que levam ao roubo de credenciais ou a tokens de autenticação. Por isso, proponho a implementação de mais um nível de autenticação, baseado em códigos efêmeros.

Nos testes realizados neste trabalho durante a pesquisa inicial, as bibliotecas em Python que implementam o TOTP e o HOTP mostraram-se promissoras para uso em aplicações web (Python PyOTP, 2023) e, quando aliado a não intervenção humana, o chamado *passwordless*, pode ser uma forma de diminuir alguns ataques como os de engenharia social e os de força bruta.

Na proteção dos sistemas mais críticos convém que seja feita uma arquitetura multicamadas, onde diversos fatores de autenticação, proteção de segredos, proteção das aplicações por hardware e *Zero Trust* sejam implementadas.

Usar diversas camadas de proteção diminui a superfície de ataque, dificulta a movimentação lateral em caso de um ataque bem-sucedido a um dos ativos e melhora a resiliência do ambiente computacional.

1.3. Objetivos

1.3.1. Objetivo geral

O objetivo geral é propor uma arquitetura de segurança multicamadas, empregando o *Zero Trust* para gestão de segredos com autenticação sem interação humana, em ambiente de execução confiável.

1.3.2. Objetivos Específicos

O objetivo geral se desdobrará nos seguintes objetivos específicos:

I. Elaborar uma arquitetura multicamadas baseada em *Zero Trust* para melhorar a segurança.

II. Elaborar a gestão de segredos usando OTP em ambiente de execução confiável.

III. Desenvolver um mecanismo de autenticação sem interação humana, baseado em OTP.

IV. Avaliar um modelo de ameaças para o ambiente proposto.

1.4. Contribuições

Como contribuições destacamos a arquitetura de segurança multicamadas. Essa arquitetura visa implementar o conceito de confiança zero, aplica a proteção de segredos em ambiente de execução confiável, provê mais uma forma de autenticação sem interação humana, reduzindo a probabilidade de ataques, conforme o modelo de ameaças apresentado.

O presente trabalho resultou na publicação de um artigo em uma conferência Qualis A3 além de um periódico atualmente em processo de revisão.

Tabela 1 - Publicação

Título	**Qualis**	**Local**	**Autores**
A non-Interactive One-Time Password-based Method to Enhance the Vault Security	A3	*The 38th International Conference on Advanced Information Networking and Applications (AINA)*	Juarez de Oliveira, Altair Olivo Santin, Eduardo Kugler Viegas e Pedro Horchulhack

1.5. Estrutura do Documento

Este trabalho está organizado da seguinte maneira:

- O capítulo 1 faz uma breve Introdução visando contextualizar a elaboração deste trabalho.
- O capítulo 2 traz a fundamentação teórica.
- O capítulo 3 descreve trabalhos relacionados.
- O capítulo 4 apresenta o tópico “Segurança de Sistemas Vault” descrevendo os principais aspectos técnicos.
- O capítulo 5 apresentamos o “Protótipo” da arquitetura de segurança proposta.
- O capítulo 6 fazemos uma “Análise de Segurança” com alguns testes que apoiam a demonstração da viabilidade.

- O capítulo 7 apresenta a “Conclusão”.

Capítulo 2
Fundamentação Teórica

Este capítulo apresenta alguns conceitos utilizados na concepção da arquitetura de segurança, como computação confiável, protocolos de autenticação e autorização, múltiplos fatores de autenticação, o *framework* de segurança CIS Controls, o PAM, as vulnerabilidades de segurança em servidores e a arquitetura Zero Trust.

2.1. Computação Confiável

Computação Confiável refere-se à proteção de dados em uso por meio de execução de computação em um Ambiente de Execução Confiável (TEE), atestado e baseado em hardware (CCC, 2023). Significa que, mesmo que um adversário tenha acesso à memória do computador, aproveitando-se de alguma brecha em outro sistema ou acesso indevido, não conseguirá acesso às informações em processamento. Esta arquitetura, presente em boa parte dos processadores fabricados na última década, ainda não conta com ampla utilização.

Intel SGX (*Software Guard Extensions*) é uma tecnologia de computação confiável, que consiste em criar um enclave seguro no qual as aplicações possam executar operações confidenciais, impedindo rapidamente o acesso de outros aplicativos, como visto na Figura 1.

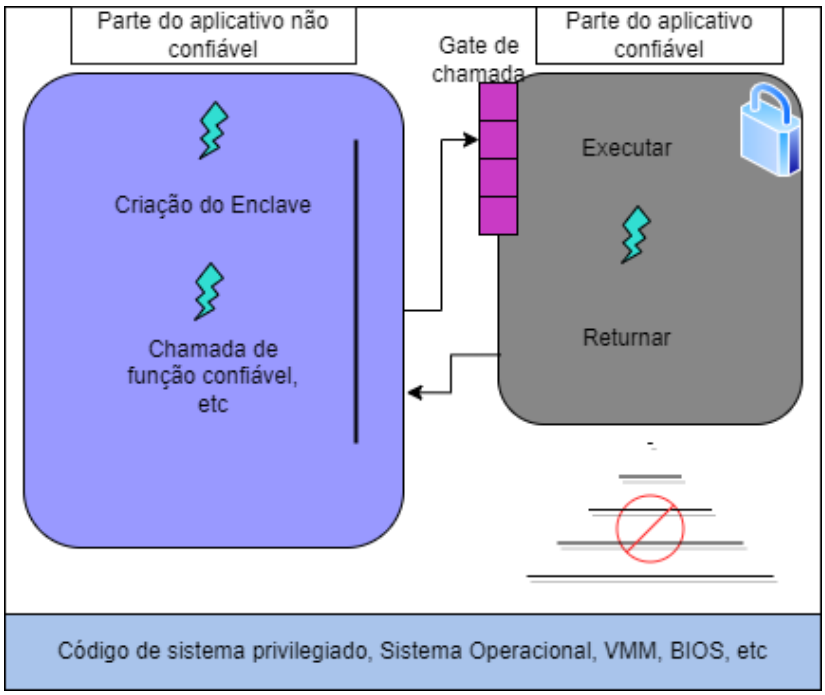

Figura 1 - Visão geral da arquitetura Intel SGX. Fonte: adaptado de Intel (Intel, 2023).

Um enclave é uma região de memória isolada, protegida por hardware, onde os aplicativos armazenam dados confidenciais e executam parte do seu código.

O Intel SGX proporciona:

a) Isolamento, com a criação de enclaves seguros, que podem executar códigos criptografados e sem acesso pelo sistema operacional;

b) Inicialização segura, com o código do enclave selado com uma chave de criptografia única que está associada ao hardware e é usado para verificar a integridade do enclave antes que ele seja carregado na memória;

c) Criptografia e autenticação, com todos os dados dentro do enclave criptografados, e as operações dentro do enclave autenticadas para garantir a integridade e a confidencialidade dos dados, em processos chamados de atestação, que pode ser local ou remota;

d) Gerenciamento de chaves de criptografia usadas para proteger os enclaves manuseadas pelo hardware SGX e não podendo ser acessadas nem manipuladas pelo sistema operacional ou outros aplicativos (Intel, 2023).

Para implementação de TEE (*Trusted Execution Environment*) usando Intel SGX com um maior nível de portabilidade, podemos utilizar o projeto Gramine, que é um

libOS, um *light guest operating system*, cuja função é abstrair a complexidade de um sistema operacional completo na implementação do SGX. Pode ser instalado no sistema hospedeiro ou ainda rodar em container Docker (Intel Gramine GSC, 2023).

O Gramine possui um diretório com alguns exemplos de código que podem ser usados para teste. Gramine é a continuidade do projeto Graphene, inicialmente desenvolvido por engenheiros da Intel.

2.2. Protocolos de Autorização e Autenticação

O protocolo OAuth 2.0, que significa *Open Authorization* 2.0, é um padrão de autorização amplamente utilizado na integração de sistemas e na segurança de aplicações web. Ele permite que usuários concedam acesso a determinadas informações ou recursos sem compartilhar suas credenciais de login, aumentando a segurança e a praticidade no gerenciamento de autenticação.

O seu funcionamento envolve três principais entidades: o Cliente, o Servidor de Autorização e o Servidor de Recursos. O Cliente é a aplicação que busca acesso aos recursos protegidos em nome do usuário. O Servidor de Autorização é responsável por autenticar o usuário e conceder ou negar o pedido de autorização. O Servidor de Recursos, por sua vez, armazena e gerencia os recursos protegidos que o Cliente está tentando acessar.

O processo inicia-se com o Cliente solicitando autorização ao Servidor de Autorização, fornecendo informações como identificação, escopo de acesso desejado e redirecionamento para retorno.

O Servidor de Autorização autentica o usuário, verifica a validade da solicitação e, se aprovado, redireciona o Cliente com um código de autorização.

Em seguida, o Cliente troca esse código de autorização por um token de acesso diretamente com o Servidor de Autorização. Esse token é um tipo de credencial que o Cliente usa para acessar os recursos protegidos no Servidor de Recursos, sem a necessidade de compartilhar credenciais do usuário (Auth0, 2023).

Os tokens Oauth 2.0 podem ser de diferentes tipos, como token de acesso, token de atualização e token de ID, cada um com sua finalidade específica. Além disso, o protocolo

oferece a possibilidade de renovar tokens, melhorando a segurança ao evitar a exposição prolongada de credenciais.

O OpenID Connect (OIDC) é um protocolo de autenticação que funciona como uma camada de autenticação em cima do protocolo OAuth 2.0, adicionando uma camada de autenticação ao processo. Após uma autenticação bem-sucedida, o OIDC emite um ID Token, que é um token JWT (JSON Web Token) contendo informações sobre a identidade do usuário, como nome, endereço de e-mail, e outros atributos relevantes.

O protocolo de autenticação SAML (*Security Assertion Markup Language*) é um outro padrão aberto para a troca de informações de autenticação e autorização entre partes, geralmente em ambientes de autenticação única (Single Sign-On - SSO).

O SAML utiliza XML para troca de informações e permite que entidades autenticadoras, conhecidas como provedores de identidade (IdPs), forneçam informações de autenticação a serviços confiáveis, conhecidos como provedores de serviços (SPs).

Neste trabalho optou-se pela utilização do Oauth2 em vez do SAML, pela melhor compatibilidade com provedores de identidade IdP atualmente.

2.3. MFA – Múltiplos Fatores de Autenticação

MFA (múltiplo fatores de autenticação) são formas de se inserir mais um componente/fator no processo de autenticação, além do usuário e da senha. Podemos assim enviar mais um código por meio de mensagens SMS, gerar um código em um aplicativo de celular, enviar um código por email, entre outros.

Uma biblioteca comum entre desenvolvedores Python para MFA é a PyOTP (Python PyOTP, 2023), que permite o gerenciamento de códigos OTP dos tipos TOTP (OTP baseado em tempo) e HOTP (OTP baseado em HMAC), tecnologia que detalhamos mais a frente.

Os Mecanismos de Senhas Únicas (OTP, do inglês *One-Time Password*) representam uma abordagem eficaz para aumentar a segurança no processo de autenticação, especialmente em ambientes online.

Esses mecanismos geram códigos de senha únicos que só são válidos por um curto período, tornando mais difícil para os invasores comprometerem as credenciais de um usuário.

Existem dois tipos de OTP: TOTP (Time-based One-Time Password) e HOTP (HMAC-based One-Time Password).

O TOTP utiliza um algoritmo que gera códigos temporários baseados no tempo. Geralmente, essa abordagem é implementada através do uso de um aplicativo de autenticação em um dispositivo móvel.

O servidor e o dispositivo compartilham uma chave secreta, e o código gerado pelo aplicativo muda a cada período definido de tempo, geralmente 30 segundos. Isso significa que mesmo se um código for interceptado, ele se tornará obsoleto em breve, aumentando a segurança do processo.

Por outro lado, o HOTP não é baseado no tempo, mas sim em um contador que é incrementado a cada solicitação de código. Assim como no TOTP, o servidor e o dispositivo compartilham uma chave secreta, mas a validade do código não é limitada pelo tempo.

Esse método é mais adequado para situações em que a sincronização de tempo entre o servidor e o dispositivo pode ser um problema, embora a principal desvantagem seja que os códigos permanecem válidos até serem usados ou expirarem, o que pode representar um risco maior em caso de interceptação.

Assim, a diferença fundamental entre TOTP e HOTP está na maneira como os códigos são gerados. O TOTP é baseado no tempo, enquanto o HOTP é baseado em um contador que é incrementado com cada uso, pois ambos os mecanismos de OTP têm se mostrado eficazes na proteção contra os ataques de *phishing* e captura de senha, proporcionando uma camada adicional de segurança além das tradicionais senhas estáticas.

A escolha entre TOTP e HOTP depende das necessidades específicas de cada aplicação e do equilíbrio desejado entre segurança e usabilidade. Em muitos casos, a implementação de OTP é combinada com outros métodos de autenticação, como senhas tradicionais, para criar um sistema multifatorial robusto e adaptável (TWILIO, 2023).

2.4. CIS Controls

CIS Controls (Center for Internet Security) (CIS Security, 2023) é reconhecido por toda a comunidade de cibersegurança como um bom ponto de partida para controles tecnológicos para serem aplicados em um ambiente computacional, sendo que os 18 grupos de controles que podem ser aplicados a 3 tamanhos de organizações diferentes: IG1, IG2 e IG3.

O CIS possui uma ferramenta chamada *CIS Navigator*, que permite a comparação dos controles do CIS com os de outros frameworks. É um dos frameworks de cibersegurança mais utilizados no mundo para desenhar controles de prevenção e mitigação. Na tabela a seguir (Tabela 2) estão listados os 18 controles definidos nesta versão:

Tabela 2 - Controles do CIS versão 8. Fonte: adaptado de CIS Security (CIS Security, 2023)

Controle 01. Inventário e controle de ativos corporativos.
Controle 02. Inventário e controle de ativos de software.
Controle 03. Proteção de dados.
Controle 04. Configuração segura de ativos corporativos e software.
Controle 05. Gestão de contas.
Controle 06. Gestão do controle de acesso.
Controle 07. Gestão contínua de vulnerabilidades.
Controle 08. Gestão de registros de auditoria.
Controle 09. Proteções de e-mail e navegador Web.
Controle 10. Defesas contra malware.
Controle 11. Recuperação de dados.
Controle 12. Gestão da infraestrutura de rede.
Controle 13. Monitoramento e defesa da Rede.
Controle 14. Conscientização sobre segurança e treinamento de competências.
Controle 15. Gestão de provedor de serviços
Controle 16. Segurança de aplicações.
Controle 17. Gestão de respostas a incidentes.
Controle 18. Testes de invasão.

Neste trabalho, considera-se principalmente os controles 05 e 06, pois é importante que em cada projeto de infraestrutura ou cibersegurança, fazer a análise de compatibilidade com algum framework de segurança, validando que os parâmetros seguidos sejam aqueles reconhecidos como boas práticas por toda a comunidade, mesmo na proposição de uma solução diferente da usual de mercado.

2.5. PAM – Privileged Access Management

PAM (*Privileged Access Management, Key Vault*) é um processo que se preocupa com o uso de credenciais privilegiadas. A principal ferramenta deste processo é o Vault, também conhecido por cofre de senhas (*password safe*) ou simplesmente Vault (OneIdentity, 2023).

No Vault são armazenados segredos como as chaves de acesso, usuários administradores, usuários de acesso remoto como SSH, pares de chave/valor, usuários de bancos de dados, entre outros, cujo fluxo pode ser conferido na Figura 2.

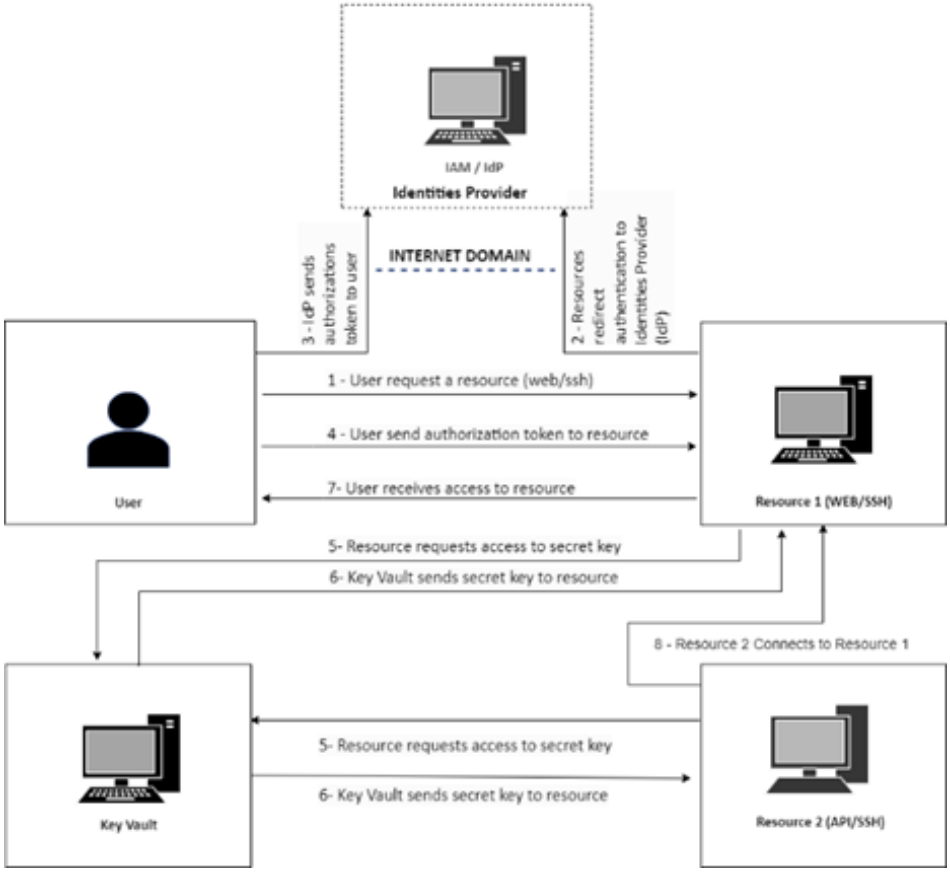

Figura 2 - Fluxo de autenticação do vault padrão. fonte: do autor.

Vault é um sistema de gerenciamento de identidade e criptografia que pode usar senha ou atuar sem senha (*passwordless*). Os mecanismos de autenticação multi-fator (MFA) combinam um ou mais fatores de autenticação além do nome de usuário e senha. Junto com isso é possível, por exemplo, enviar um código SMS, gerar um código de autenticação em um aplicativo e enviar um código para um endereço de e-mail, entre outros. Geralmente, MFA é uma imposição ao acessar o Vault.

Com o Vault podemos alterar automaticamente esses segredos (senhas ou códigos), periodicamente ou a cada acesso. Isto impede que, ao ter acesso indevido a um desses códigos ou senhas, o adversário possa reutilizá-las para algum acesso indevido ou elevação de privilégios no mesmo ou em outro sistema.

O gerenciamento de acesso privilegiado é um importante controle previsto no CIS Controls V8, mais especificamente no Controle 6 – Gestão de Controle de Acesso, que prevê a centralização dos usuários, uso do MFA, além de um controle maior sobre os usuários administradores.

2.6. Vulnerabilidades em Servidores

Os sistemas e hardwares apresentam vulnerabilidades que potencialmente podem ser exploradas por adversários. Neste trabalho nos limitaremos a citar as vulnerabilidades do CVE de softwares e vulnerabilidades de aplicações web.

CVEs – *Common Vulnerabilities and Exposure* (Mitre, 2023) é catálogo com as vulnerabilidades encontradas em software de qualquer fabricante. A centralização das CVEs ajuda os profissionais a terem acesso à documentação das fragilidades e seus impactos, incluindo mitigação, especialemente quando vinculadas ao *National Vulnerability Database* do NIST (NIST, 2023).

Às CVEs é atribuído um *score* CVSS – *Common Vulnerability Scoring System*, cujos valores vão de 1 a 10. Ter uma vulnerabilidade com score alto (10) não significa necessariamente que o ativo será comprometido, mas é um indicador de que providências devem ser tomadas.

Os fabricantes avaliam a falha reportada, registram a CVE, quando possível disponibilizam um patch de correção ou informam maneiras de mitigação. Quando uma CVE ainda não tem correção ela é conhecida por CVE Zero-Day.

O TOP 10 OWASP (OWASP Top10, 2023) é um documento de conscientização para desenvolvedores de aplicações web. Nele são listadas as 10 falhas mais recorrentes que afetam estas aplicações. A fragilidade mais recorrente no ano de 2021 foi a *Broken Access Control* – A01. As 34 CWEs – *Common Weakness Enumerations*, que tem mais recorrências de qualquer outra.

Um exemplo de Zero Day e com score alto é a CVE-2022-3181 (Mitre, 2023), que atinge os servidores web Apache2 *(mod_proxy X-Forwarded-For dropped by hop-by-hop mechanism*; Figura 2 : *Resource Web*) em suas versões com um score 9.8.

Análises de cibersegurança devem levar em consideração a superfície de exposição, em geral, percebida nas vulnerabilidades exploradas diretamente ou resultantes de movimentação lateral. A movimentação lateral também diminui a eficácia das camadas de segurança que são interpostas considerando sua disposição numa arquitetura de segurança.

O MITRE ATT&CK (Mitre, 2023) é uma base de conhecimento de acesso global de técnicas e táticas de adversários criada a partir de observações do mundo real. Nela podemos entender a dinâmica da maioria dos ataques conhecidos e as formas mais adequadas de fazer a segurança que deve ser utilizada na análise de cibersegurança.

A base de conhecimentos MITRE ATT&CK (Mitre, 2023) trabalha com TTP, *Tatics, Techniques and Defenses*. Tática é o objetivo do adversário, a razão para que ele realize uma ação, por exemplo, obter acesso a uma credencial. Técnica (*techniques*) é como este adversário atinge um objetivo ao realizar uma ação. Defesa (*defense*) são as ações de mitigação que podem ser usadas pela organização para impedir o uso de alguma técnica ou subtécnica de ataque (Mitre, 2023).

Na modelagem do adversário, as análises de segurança cibernética devem considerar o ataque superfície reconhecida nas vulnerabilidades exploradas diretamente ou resultantes de movimento lateral (capacidade adversária).

Além disso, o movimento lateral reduz a eficiência das camadas de segurança sobrepostas, considerando sua disposição em uma arquitetura de segurança.

A avaliação do nível de segurança de um ambiente computacional que se deseja proteger é muito importante e o PTES (PTES, 2023) devem ser empregados para que o teste seja feito da maneira sistêmica, permitindo inclusive que a repetitividade da avaliação seja possível.

O PTES estabelece uma sequência prática, numa estrutura de *cyber kill chain*, com os seguintes elementos*: Pre-engagement Interactions, Intelligence Gathering, Threat Modeling, Vulnerability Analysis, Exploitation, Post Exploitation and Reporting*, que permitem sistematizar o processo de avaliação.

2.7. Arquitetura Zero Trust

De acordo com o Instituto Nacional de Padrões e Tecnologia NIST, em seu documento SP 800-207 (NIST, 2020) , *Zero Trust* (ZT) é o termo para "*um conjunto em evolução de paradigmas de segurança cibernética que movem as defesas de perímetros estáticos baseados em rede para se concentrarem em usuários, ativos e recursos*".

De acordo com esta publicação do NIST, a arquitetura *Zero Trust* é aderente aos seguintes princípios:

a) **Todas as fontes de dados e serviços computacionais são consideradas recursos**, então uma rede pode ser composta por diversas classes de dispositivos.

b) **Todas as comunicações são protegidas independentemente da localização do recurso**, logo, a localização do recurso por si só não implica em confiança.

c) **O acesso a recursos empresariais individuais é concedido por sessão**. Por isso, a confiança no solicitante é avaliada antes que o acesso seja concedido, sendo os privilégios mínimos para o uso.

d) **O acesso aos recursos é determinado por políticas dinâmicas**, incluindo o estado observável da identidade do cliente, da aplicação/serviço e do ativo solicitante, podendo incluir outros atributos comportamentais e ambientais.

e) **A organização monitora e mede a integridade e a postura de segurança de todos os ativos próprios e associados**. Nesta perspectiva nenhum ativo é inerentemente confiável, logo a postura de segurança do ativo é avaliada a cada solicitação de recurso.

f) **Toda a autenticação e autorização de recursos são dinâmicas e fortemente aplicadas anteriormente à concessão do acesso**, formando um

ciclo de obtenção de acesso, verificação e avaliação de ameaças, adaptação e reavaliação contínua da confiança na comunicação.

g) **A organização coleta o máximo de informações possível sobre o estado atual dos ativos**, infraestrutura de rede e comunicações e as utiliza para melhorar sua postura de segurança.

Nessa arquitetura, utilizando-se de ferramentas apropriadas como as soluções comerciais de SASE ou ZTNA, podemos autenticar não somente o usuário, mas o host (sistema operacional ou dispositivo) que ele está utilizando.

Pode ser possível ainda criar diversas políticas de acesso, baseadas em sistema operacional, localização geográfica, horário, entre outras, fazendo o controle de acesso ao ativo de uma forma mais granular.

2.8. Considerações Finais

Neste capítulo analisamos algumas das camadas de proteção de segurança, protocolos e tecnologias que, mais a frente, serão utilizadas como base na construção de nossa arquitetura.

A computação confiável, cujo uso ainda é bem pouco disseminado no mercado, pode ajudar a proteger parte dos códigos em execução, diminuindo consideravelmente a possibilidade de acesso aos dados que estão em processamento no host, como ataques de Buffer Overflow e DMA (*Direct Memory Access*).

Protocolos de autenticação e autorização, como o Oauth2 e OIDC são geralmente usados em processos de autenticação entre os recursos e os provedores de identidade (IdP), permitindo ainda o uso de múltiplos fatores de autenticação (MFA), para colocar mais uma camada no nível de autenticação.

Apresentamos uma visão geral sobre a arquitetura Zero Trust, suas possibilidades de implementação, suas tecnologias e como o ZT pode ser usado para diminuir a superfície de exposição do ativo, dificultando atividades de reconhecimento por parte de um atacante.

Trazemos ainda análise de vulnerabilidades Web (top 10 OWASP,) e de softwares (CVEs), com suas classificações, onde são catalogadas e os métodos que os atacantes podem utilizar para atingir seus objetivos (MITRE ATT&CK).

O framework CIS Controls foi apresentado como uma metodologia eficaz na implementação de barreiras de segurança. Este modelo é utilizado pela maioria das organizações por se constituir em boas práticas que são definidas por profissionais da área de cibersegurança de todo o mundo. Atualmente encontra-se na versão 8.

O PAM, que é um subprocesso ou subsistema do processo de Gestão de Identidade e Acessos, será utilizado na arquitetura proposta para melhorar a proteção dos segredos.

Assim, esses elementos em conjunto permitem o fortalecimento das camadas de segurança que apresentamos em nossa arquitetura.

Capítulo 3
Trabalhos Relacionados

Neste capítulo são descritos alguns artigos relacionados aos temas necessários para compreensão deste trabalho, sendo separados por tópicos “Computação Confiável”, “Zero Trust”, “Autenticação e MFA” e “PAM e IdP”.

3.1. Computação Confiável

WU, CAI e LI, no trabalho “*SGX-UAM: A Secure Unified Access Management Scheme With One Time Passwords via Intel SGX*” (Wu, H.J., & Li, 2021), implementam um sistema de gerenciamento de acesso unificado e seguro, chamado por eles de SGX-UAM, no qual utilizam um ambiente confiável com SGX com autenticação OAuth2. Esta implementação resiste à maioria dos ataques de clientes, ataques MITM, ataques de phishing, a maioria dos ataques de DDoS e de repetição, onde outras implementações genéricas de UAM são vulneráveis.

O *Confidential Computing Consortium* (CCC) faz uma análise do uso atual das tecnologias de computação confiável (CCC, 2023), comparando-os com outras tecnologias que visam a proteção de dados, como a criptografia homomórfica, e o TPM (*Trusted Platform Modules*).

São descritas as principais características dos ambientes de execução confiável (TEE) em hardware, como a confidencialidade do código, lançamento autenticado, programabilidade, atestabilidade e a recuperabilidade. No modelo de ameaças deles a computação confiável visa reduzir a capacidade do operador/proprietário dos dados de uma plataforma de acessar esses recursos de uma maneira que seja logicamente ou economicamente viável.

Falcão et al (Falcão, Silva, Luz, & Brito, 2022) destacam os desafios da migração do modelo tradicional de computação para o modelo de computação em nuvem, a necessidade de implementação de um modelo de confiança zero, que aborda algumas

questões de segurança conhecidas, como movimentação lateral, mas discute sobre a complexidade da complexidade extra da gestão de identidade.

Aborda ainda quais estratégias mais amplas, seja para proteção de código, dados e credenciais em aplicativos. Destas análises surgiu o SPIRE, uma implementação do Zero Trust com computação confidencial, por meio da solução SCONE, que é uma implementação de ambientes TEE em containers (Falcão, Silva, Luz, & Brito, 2022).

3.2. Zero Trust

Como as diretivas de alto nível relativas ao risco, segurança cibernética e conformidade são operacionalizadas no centro nervoso de qualquer organização acima de uma certa complexidade? Como podemos medir a eficácia da segurança em soluções tecnológicas de forma comprovada e medida? E como essa tecnologia pode ser alinhada com a governança e metas financeiras do conselho? Estas são perguntas efetuadas por qualquer CEO, CIO ou CISO de uma organização.

O artigo "*Zero Trust Validation: from Practice to Theory*" (Bobbert & Scheerder, 2022) investiga as limitações das abordagens atuais de Zero Trust, que normalmente abordam segmentação de rede, recursos acessados de maneira segura e o conceito de "sempre verifique, nunca confie".

São analisados os críticos de sucesso no ZT desenvolvido pela ON2IT '*Zero Trust Innovators*', descrevendo ainda um projeto de um artefato com estrutura de Zero Trust, com uma validação empírica de estratégicas de ZT. Por fim, o artigo visa obter uma aceitação mais ampla do ZT.

O resultado da pesquisa é uma estrutura proposta, associada à tecnologia, por meio do princípio de confiança zero, abrangendo múltiplas camadas da organização para entender e alinhar os riscos cibernéticos com a capacidade de atuação da organização no combate a esses riscos.

Segundo a pesquisa, a maturidade da segurança da informação, que pode ser exponenciada com o uso do Zero Trust, pode diminuir os riscos e os custos da organização com segurança. Esta pesquisa contribuiu para a análise de importância do Zero Trust no contexto organizacional de diminuição da exposição ao risco.

No artigo "*Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust*" (Buck, Olenberger, Schweizer, Völter, & Eymann, 2021), os autores realizam um revisão multivocal da literatura sobre o Zero Trust, tanto relacionadas a pesquisa acadêmica, quanto relacionadas a pesquisas práticas.

Os resultados mostram que a literatura acadêmica se concentra principalmente nas melhorias de arquitetura e desempenho do Zero Trust. Por outro lado, a literatura orientada para a prática teve mais foco nas vantagens organizacionais de confiança zero e em possíveis estratégias de migração.

No entanto, as análises econômicas e os estudos relacionados com usuários têm sido pouco avaliados, tanto pela abordagem acadêmica quanto pela abordagem prática. Entre as contribuições do artigo estão a sumarização dos artigos relacionados com o Zero Trust.

3.3. Autenticação e MFA

O OAuth 2.0 é um protocolo de autorização fortemente utilizado em aplicações web, em especial por redes sociais e provedores de identidade. O artigo "*OAuth-SSO: A Framework to Secure the OAuth based SSO Service for Packaged Web Applications*" (Houssain, Houssain, Houssain, Sohag, & Hahman, 2018) analisa o processo de autenticação de logon único, chamado de SSO (Single-Sign-On) e propõe uma modificação do fluxo de execução do Oauth em aplicativos.

O trabalho apresentou resultados positivos na proteção contra os seguintes ataques: código de autorização de roubo e desvio, roubo de token de acesso via XSS, espionagem ou roubo de token de acesso e ataque de CSRF, fazendo pequenas modificações no fluxo do OAUTH.

FIDO2/WebAuth é uma abordagem muito promissora podem substituir as senhas em algum momento uma vez que também permite autenticação sem senha, além de funcionam como um segundo fator para qualquer autenticação web. Embora isso resolva alguns problemas de senhas usando criptografia de chave pública/privada e criptografia forte e mecanismos de autenticação como a biometria, não consegue detectar um invasor após um login bem-sucedido.

No artigo "*FIDOnuous: A FIDO2/WebAuthn Extension to Support Continuous Web Authentication*" (Klieme, Wilke, Dornick, & Meinel, 2020) é avaliada a extensão que permite uma proteção para autenticação contínua em navegadores e também em aplicativos para dispositivos móveis.

Henricks e Kettani analisam os benefícios e características da autenticação por múltiplos fatores, incluindo o uso de biometria (Henricks & Kettani, 2019). O MFA é apresentado como uma forte camada adicional de segurança adicionado a autenticação convencional por senha.

A proposta de henricks e Kettani coloca ainda a perspectiva de exposição futura de características de DNA humano, o que diminuiria seu nível atual de segurança para uso como um fator de autenticação seguro.

3.4. PAM e IdP

A norma ISO 27001, editada pela ISO (*International Stardards Organization*), é um dos principais guias de boas práticas sobre segurança da informação existentes no mundo. Este framework implementa o SGSI ou Sistema de Gestão de Segurança da Informação.

Um dos controles estabelecidos por esta norma é o Controle de Acesso que, em uma das suas abordagens, inclui o PAM (*Privileged Access Management).* Contudo, muitas organizações podem não ter conhecimento técnico ou recursos suficientes implantar e administrar este tipo de solução.

No trabalho "*Assessing Privileged Access Management (PAM) using ISO 27001:2013 Control*", os autores (Purba & Soetomo, 2018) falam sobre a importância da organização atender aos requisitos de conformidade legal e também proteger os seus ativos críticos quanto a vazamentos de dados. Para isso, elaboraram uma matriz de requisitos para apoiar as organizações na aquisição e implementação de uma solução de PAM.

Embora o serviço de computação em nuvem seja atualmente bastante difundido, devido à sua flexibilidade e facilidades na hora de provisionar os recursos, muitas pessoas não se sentem confortáveis quando enviam seus dados pessoais para serviços hospedados desta maneira.

Um vasto histórico de vazamento de dados pessoais tem resultado em perdas para empresas e para seus clientes. O trabalho "*Enhancing Privacy on Identity Providers*" (Weingärtner, & Westphal, 2014) aborda a segurança e a privacidade das informações pessoais no gerenciamento de provedores de identidade (IdP), visando conscientizar sobre as possibilidades de coleta excessiva de dados.

Então, os próprios provedores de identidade, que deveriam ser fonte de segurança, acabam por ser ponto de vulnerabilidade.

A importância da proteção dos dados pessoais, dentro do contexto de cibersegurança, aumenta a cada dia, aumentando os riscos e desafios para as organizações. O artigo "*Public Institutions Updated Enhanced Biometric Security, Zero Trust Architecture and Multi-Factor Authentification*" (Iordache, Dragomir, & Marian, 2022) apresenta a possibilidade de introdução da autenticação multifatorial, usando biometria como padrão para melhorar a segurança cibernética em instituições públicas da Romênia.

A autenticação biométrica seria usada ainda para restringir privilégios numa arquitetura de confiança zero. além da melhoria da segurança, o modelo também atenderia os padrões do regulamento geral de proteção de dados da União Europeia (GDPR).

Os provedores de serviços (SP), que administram diversos ambientes de cliente, necessitam de regras e meios mais seguros quando fornecem acesso a estes serviços aos seus funcionários. Isso porque ele fica responsável em caso de incidentes, podendo perder credibilidade ou arcar com indenizações.

Com base em uma revisão literária que apontou os principais problemas, o artigo "*Privileged access management model for a managed service provider*" (Tuononem, 2023) buscou definir uma série de requisitos de segurança e de conformidade, servindo de base para a implementação do processo e também para o desenvolvimento contínuo do processo de gestão de acessos privilegiados.

Tabela 3 - Características dos Trabalhos Relacionados.

AUTORES	COMPUTAÇÃO CONFIÁVEL	ZERO TRUST	AUTENT. / MFA	PAM E IDP	AVALIAÇÃO DE SEGURANÇA
(Wu, H.J., & Li, 2021)	SIM	NÃO	NÃO	SIM	NÃO
(CCC, 2023)	SIM	NÃO	NÃO	NÃO	NÃO
(Bobbert & Scheerder, 2022)	NÃO	SIM	NÃO	NÃO	NÃO
(Falcão, Silva, Luz, & Brito, 2022)	SIM	NÃO	NÃO	SIM	SIM
(Houssain, Houssain, Houssain, Sohag, & Hahman, 2018)	NÃO	NÃO	SIM	SIM	SIM
(Henricks & Kettani, 2019)	NÃO	NÃO	SIM	SIM	NÃO
(Klieme, Wilke, Dornick, & Meinel, 2020)	NÃO	NÃO	SIM	SIM	NÃO
(Weingärtner, & Westphal, 2014)	NÃO	NÃO	SIM	SIM	NÃO
(Iordache, Dragomir, & Marian, 2022)	NÃO	SIM	SIM	SIM	NÃO
(Buck, Olenberger, Schweizer, Völter, & Eymann, 2021)	NÃO	SIM	NÃO	NÃO	NÃO
(Tuononem, 2023)	NÃO	NÃO	NÃO	SIM	NÃO
(Purba & Soetomo, 2018)	NÃO	NÃO	NÃO	SIM	NÃO
(Oliveira, Santin, Viegas, & Horchulhack, 2024)	SIM	SIM	SIM	SIM	SIM

3.5. Considerações Finais

Nos trabalhos relacionados distribuídos entre os tópicos Computação Confiável, Zero Trust, Autenticação e MFA, e PAM e IdP, buscamos analisar como a comunidade científica tem trabalhado com esses assuntos, quais os *gaps* em suas implementações permitem que os ciber-ataques continuem com elevada taxa de sucesso.

Neste sentido, vimos que a Computação Confiável, embora promissora, ainda não conseguiu alcançar lugar de destaque, mas contém diversos elementos que podem apoiar na melhoria de uma arquitetura de segurança, como proteção de memória e proteção contra elevação de privilégios.

A arquitetura de *Zero Trust* tem elementos fortes que permitem a proteção do ambiente computacional, em especial na proteção contra movimentação lateral e acesso privilegiado.

Autenticação centralizada ou por SSO, mesmo com o uso de MFA, ainda têm sido alvo constante de ataques, mas ainda parece ser a metodologia mais adequada, independente do uso de outras soluções de *passwordless*, como certificados digitais.

O processo de IAM (*Identity and Access Management*), apoiado por soluções de IdP e PAM, além da questão da gestão administrativa dos acessos, permite um monitoramento mais efetivo da ação de autenticação de usuários e concessão de privilégios, embora em determinadas situações, como no caso do acesso via SSH, podem não apresentar a mesma comodidade do acesso direto feito pelos administradores.

Capítulo 4

Segurança de Sistemas Vault

Neste capítulo apresentamos um modelo de aprimoramento de ambiente computacional, que contém elementos de vault, aplicações web, aplicação OTP e conceito de confiança zero.

Apresentamos o método de proteção, um modelo do adversário e considerações acerca do modelo de arquitetura.

4.1. Um método não interativo baseado em senha única

Esta seção descreve um método não interativo baseado em senha única, no qual há sempre um Vault para guardar e gerir os segredos (tokens/códigos, senhas etc.), diminuindo a superfície de exposição - credenciais armazenadas em arquivos e variáveis de ambiente.

Um servidor web será o único ponto de acesso de aplicação, exposto na Internet, fazendo com que os serviços sejam publicados em outras máquinas e conectados ao servidor via API (Figura 3). A autenticação de usuários pode ser por um IdP interno ou externo (passos 1-3), dependendo da necessidade de integração com outras aplicações ou das políticas interna / cultura da empresa.

A geração do token OTP (Figura 3, passos 7, 11) é feita no servidor do ambiente de execução confiável (TEE) para MFA, o que impede o acesso indevido à memória RAM e, consequentemente, à semente do gerador de OTP.

Esta abordagem foi incorporada em nossa proposta para impor dificuldades adicionais de ataque a um adversário por lidarmos com ambiente críticos, com informações muito sensíveis.

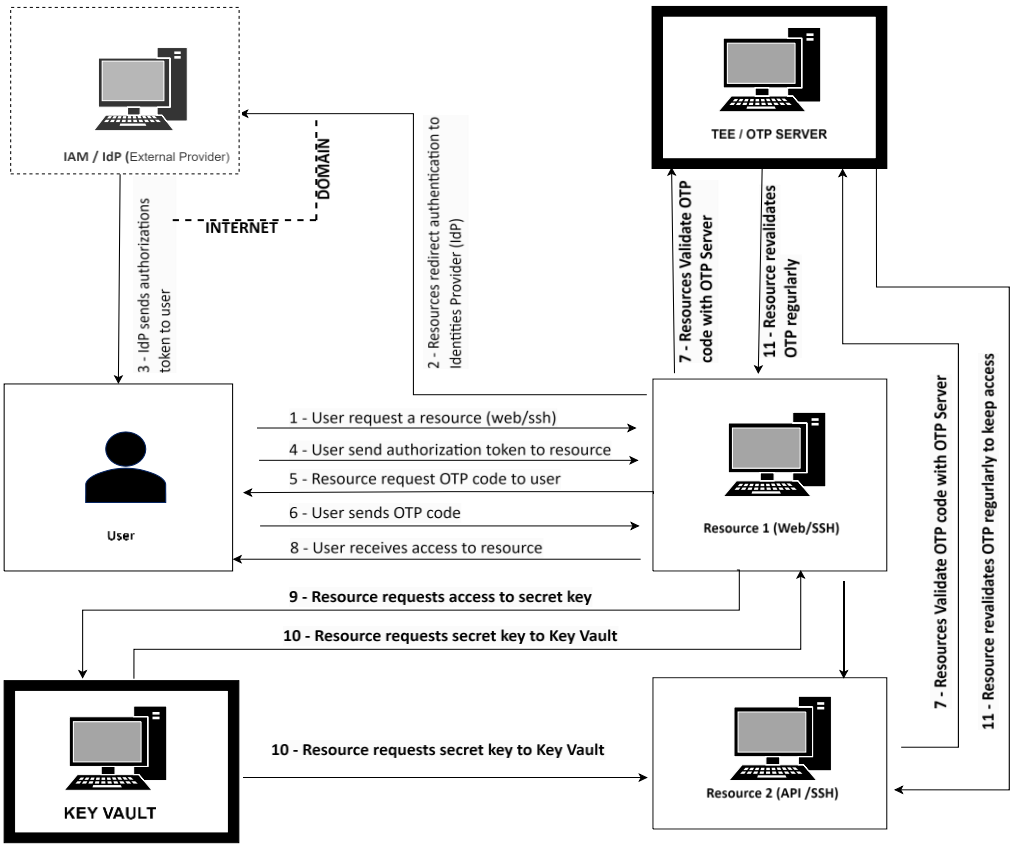

Figura 3 - Visão geral de melhoria na segurança do vault. Fonte: autor

Nos dois casos, se usa esteganografia para proteger os códigos em trânsito nas APIs, por isto a implementação no TEE é mais recomendada por permitir a implantação facilitada e segura deste controle.

Após a autenticação, seja do usuário da aplicação web, do usuário de shell seguro ou outro serviço não-interativo, o tempo da validade da credencial ou token (Figura 3, passos 9-10) é bastante importante, pois é a janela de ataque para um adversário.

Isto significa que se o token for interceptado, o atacante vai poder utilizá-lo ou até alterá-lo durante este tempo, muitas vezes sem a percepção do usuário. Calcular este tempo e ter outros fatores de autenticação além da senha, pode fazer a diferença entre ter o serviço comprometido ou não.

A dupla validação de credenciais, adicional a autenticação (Figura 4, etapas 1 a 3, eventos detalhados de userToken para Data e de SendOTPCode para authOTPuser), ocorre periodicamente no servidor web para acesso à API (Figura 4, passos 5 a 7 e passos 8 a 11).aumenta o nível de segurança e de garantia de que não houve roubo de credenciais ou mesmo da sessão.

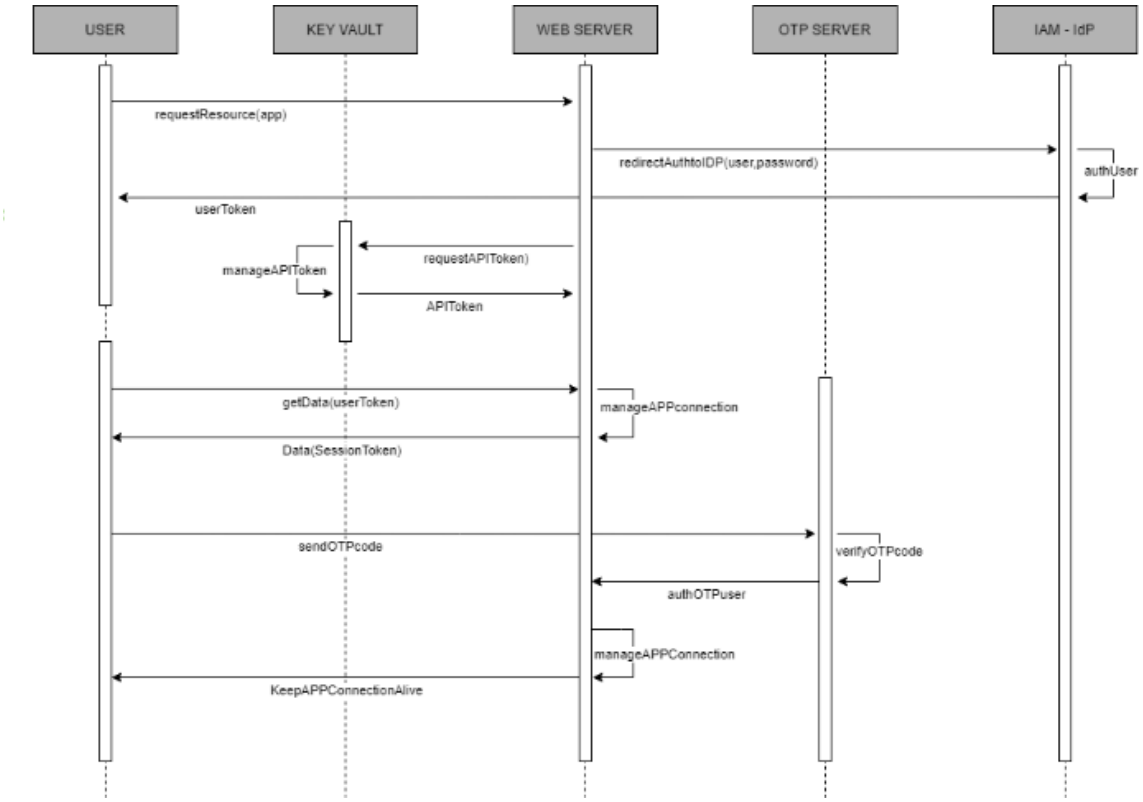

Figura 4 – Autenticação web centralizada com OTP server e Vaut. Fonte: do autor.

O monitoramento contínuo da autenticação por OTP permite que ações sejam tomadas em caso de fragilidade de segurança de forma imediata, como o fechamento (desconexão) da sessão ou de algum serviço como o SSH, por exemplo.

A Figura 5 representa o fluxo de autenticação e interação do usuário para acesso ao serviço SSH, considerando os elementos da proposta.

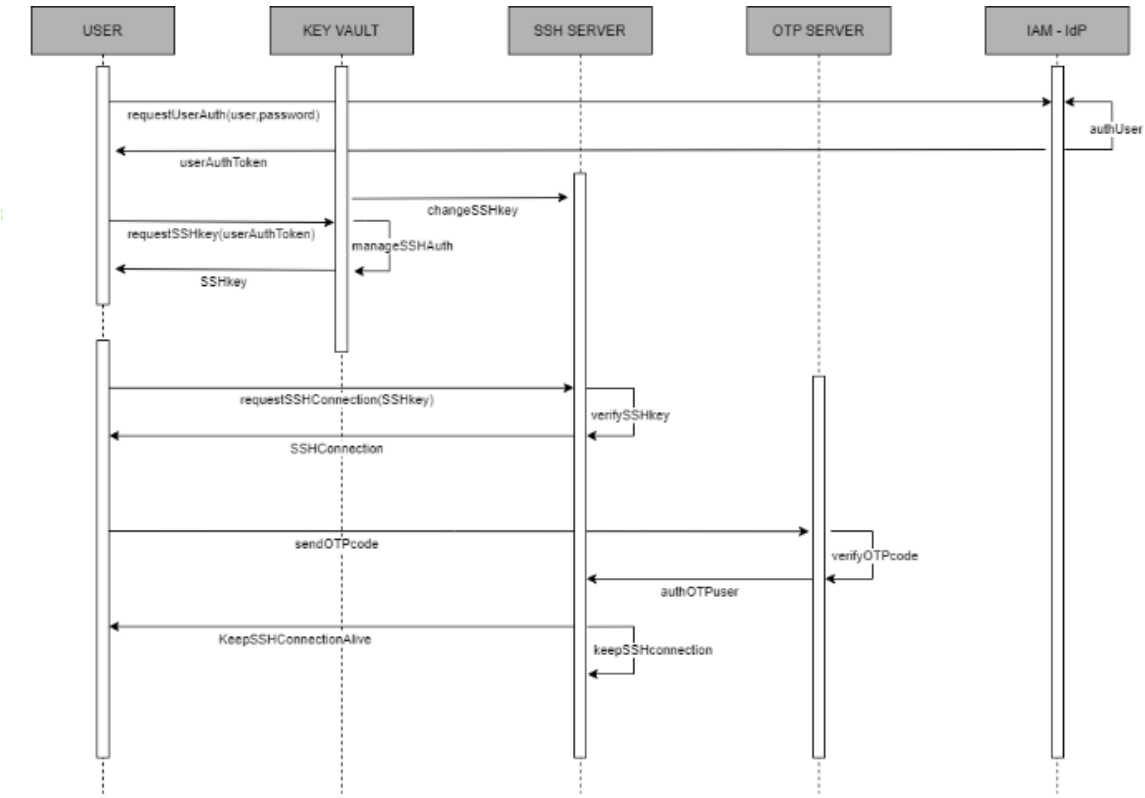

Figura 5 – Autenticação centralizada do serviço SSH com servidor de aplicação OTP. Fonte: do autor.

Após a autenticação do SSH (eventos de requisição do UserAuth para SSHConnection, Figura 5) no IdP, que dá acesso ao Vault para retirar a credencial de acesso de único uso, um script de verificação do OTP é acionado (evento sendOTPcode), aguardando que o usuário faça a segunda validação (evento authOTPuser).

Com a inserção deste mecanismo de autenticação MFA, em ambiente de execução segura (TEE) se faz uma segunda validação em autenticação web e SSH, garantindo que este código OTP só enviado de maneira não-interativa e independente do controle do usuário, se este se passou pelo IdP.

Assim, se um adversário roubar uma credencial obtida no vault dentro da validade, não conseguirá utilizá-la, pois faltará o OTP correspondente -- que só é gerado para quem passou pelos passos 1-4 da Figura 3.

Caso essa autenticação por OTP falhe, automaticamente a sessão/conexão é bloqueada e, portanto, o recurso crítico estará protegido.

4.3. O modelo do adversário

São objetivos do adversário fazer o reconhecimento do ambiente *(ID: TA0043 in MITRE Att&ck)* (Mitre, 2023) como identificar quais serviços estão disponíveis, quais portas estão abertas e como se dá o processo de autenticação.

Ao conhecer as características e fraquezas do ativo alvo, o atacante elabora uma estratégia para explorá-las, como a execução de ferramentas ou scripts automatizados para quebrar senhas, explorar alguma CVE que permita o acesso remoto ou ainda aplicar engenharia social para conseguir as credenciais válidas de algum usuário legítimo.

Mas suas capacidades de atuação dependem diretamente como o ativo está protegido por firewalls, configurações restritivas nos servidores WEB, uso de MFA na autenticação, limite no número de conexões por minuto, uso de WAF (*Web Application Firewall*) e assim por diante.

Para análise das capacidades do atacante (*adversary*), consideraremos (i) o atacante na Internet, com acesso somente ao servidor web, e (ii) na rede interna, com acesso às portas do SSH e Web.

O ambiente de execução confiável (TEE) permite uma proteção adequada da execução dos códigos em memória. Porém, pode não está disponível em todos os computares, já que não executa em qualquer sistema de virtualização, considerando o modelo INTEL/SGX.

A interceptação da autenticação MFA pode ocorrer por meio de um *keylogger*, instalado em um sistema infectado. Quando o código de MFA é enviado por um sistema externo, como um email ou SMS, pode ocorrer a interceptação caso o provedor do serviço não esteja adequadamente protegido.

Como ação de mitigação pode-se investir no treinamento dos usuários para que removam os seus cartões de autenticação quando não estiverem em uso – TTP 1111 (Mitre, 2023). Para detectar tal ação maliciosa, precisamos monitorar as APIs de acesso ao sistema, o que pode ter alguma complexidade.

Em outra ação do adversário, o material de autenticação alternativa, como um *hash* de senha, um token Kerberos ou um token OTP de aplicativo, pode ser roubado por meio de técnicas de acesso a credenciais, como acesso indevido à memória RAM ou a arquivos de configuração no computador.

Para mitigar este problema precisamos atuar na gestão de contas privilegiadas, limitando a ação de cada conta a requisitos mínimos. Os requisitos de menor privilégio devem ser aplicados às contas de usuários comuns, impedindo que estes tenham permissões de administradores ou acessos a dados que, não são estritamente necessários para desempenhar suas atividades ou são de outros proprietários (Mitre, 2023).

A detecção pode ocorrer por meio do monitoramento da criação de sessões e dos *logs* de acesso a recursos.

Considerando a técnica *Remote Services: SSH* do Mitre – ID: T1021.004 (Mitre, 2023), o adversário pretende abusar da informação de algum usuário legitimo para ter acesso à rede por meio do protocolo SSH. Um dos grupos que explora essa técnica é o APT39.

Como mitigação o MITRE Att&ck propõe desativar o serviço SSH em sistemas onde ele não é requerido, usar MFA e limitar o número de usuários com acesso ao serviço. A detecção é feita pelo monitoramento da criação da sessão, da criação do usuário e ainda da conexão de rede.

O Vault pode ajudar nos processos de mitigação e o MFA com OTP ajuda no monitoramento e desativação do serviço em caso de uso indevido.

Neste trabalho buscamos responder à seguinte questão: Como é possível prover um mecanismo para aumentar a segurança no processo de autenticação, de forma interativa quando possível, visando a *non-interference* na produtividade e usabilidade de acordo com a abordagem de uso?

4.4. Discussão

Em relação aos trabalhos relacionados, na nossa proposta aplicamos o MFA como mais uma camada de proteção, desvinculada diretamente à autenticação do provedor de identidades (IAM), embora seja adequado que o usuário use também o MFA do provedor.

Consideramos a autenticação de um provedor externo como seguro, nos limites do que a tecnologia atual permite, porque para ataques de movimentos laterais e elevação de privilégios, por exemplo, é muito difícil controlar o provedor de IAM, ao contrário do caso de gestão local de contas.

Nós também protegemos o servidor OTP em um ambiente TEE para aumentar sua resiliência a ataques, devido a criptografia de memória e outras restrições do enclave.

Na proposta apresentada, o Vault é um dos componentes principais da arquitetura, devido às suas características de integração com diversos outros tipos de mecanismos de cibersegurança.

Utilizar uma solução de *Zero Trust* pode ajudar de forma definitiva na diminuição da superfície de ataque, em especial para usuários e recursos internos. Mesmo que um usuário externo não seja autenticado pela solução de *Zero Trust*, os ativos internos podem ser.

4.5. Considerações Finais

Podemos concluir que o aumento no número das camadas de segurança, assim como a análise detalhada da aplicação de cada uma delas pode permitir um ganho considerável na proteção do ambiente computacional.

Consideramos as práticas de mercado, as ferramentas e tecnologias disponíveis que não apresentem por si só um aumento de custos e planejamos sua implementação.

Capítulo 5
Protótipo

Neste capítulo elaboramos um protótipo com alguns dos principais pontos de atenção em um ambiente computacional, principalmente no que diz respeito à proteção de segredo e autenticação.

5.1. Prototipação:

Construímos um protótipo de servidor de aplicação web rodando em uma máquina virtual com o sistema operacional Ubuntu Server 22.04. 5. Também existe um Apache2 servidor web, servidor openSSH, um aplicativo cliente OTP feito em Python e um Agente cliente SSH do Vault (*vault-ssh-helper*).

O provedor de identidade é considerado uma entidade externa. Um serviço de *Zero Trust* baseado em nuvem pública foi utilizado em complemento aos demais recursos.

O vault é executado em uma máquina virtual com sistema operacional Ubuntu Server 22.04, baseada na solução Hashicorp Vault (Hashicorp, 2023). Este servidor também possui um cliente Vault SSH (*vault-ssh-helper*) e um aplicativo cliente OTP em Python.

O Vault foi incluído nesta arquitetura porque o PAM é uma das melhores práticas para gerenciamento de acesso fornecido no *CIS Controls*, melhorando significativamente a proteção contra acesso não autorizado ao sistema.

Para complementar e integrar o ambiente com uma abordagem completa de *Zero Trust*, utilizamos o Twingate (TWINGATE, 2023), uma solução baseada em nuvem pública que pode ser utilizada de forma gratuita para pequenos ambientes.

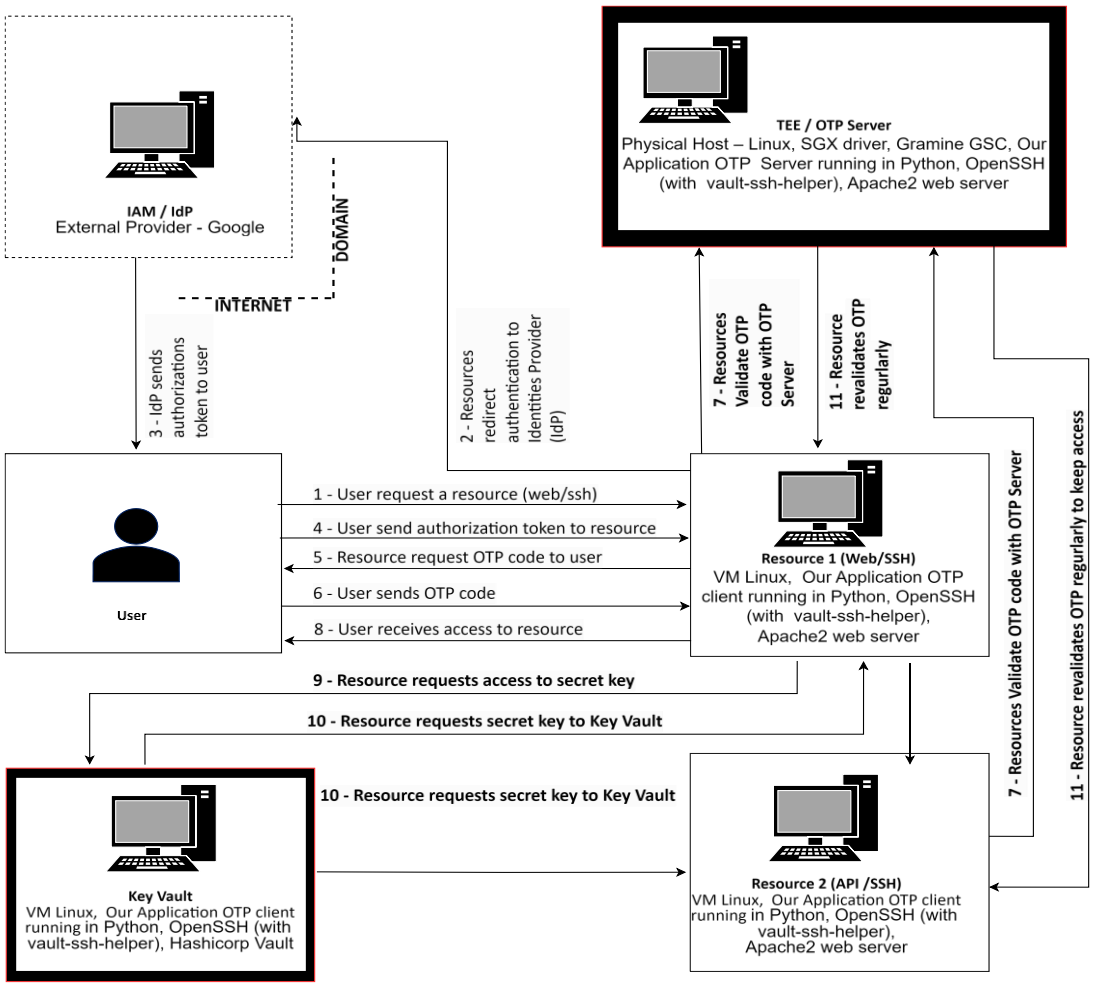

Figura 6 – Visão Geral do Protótipo. Fonte: do autor.

O servidor com a aplicação OTP Server, desenvolvida em Python com o Biblioteca PyOTP (Python PyOTP, 2023), roda em uma máquina física com sistema operacional Centos 8 Stream, que utiliza um processador Intel(R) Core(TM) i7-7500U de 2,70 GHz, com INTEL Gramine GSC em contêiner Docker, além de drives Intel/SGX, para implementar o TEE.

Este servidor físico também roda uma API feita em Python, que, em neste contexto, serve como uma aplicação crítica. Ele também executa um cliente Vault SSH (*vault-ssh-helper*).

Paralelamente à autenticação API, o usuário autentica usando um provedor de identidade (IdP). Isto é seguido pela autenticação com o OTP, isoladamente, para garantir

que as credenciais de acesso não foram comprometidas. Esta revalidação de a autenticação com o OTP pode ocorrer no login e periodicamente.

A conexão SSH é feita usando um código/senha único gerado pelo Vault. Após o processo de autenticação convencional, um pós-script é acionado internamente no servidor alvo, validando o OTP, que será executado no servidor seguro (SGX-Gramine). A conexão SSH só será concluída se isso ocorre a segunda validação (Figura).

Também podemos solicitar uma nova validação periódica. A tela de validação exibida ao usuário pode ser vista na figura 7, que ainda mostra o código que estaria correto, mas apenas para ação no modo de depuração.

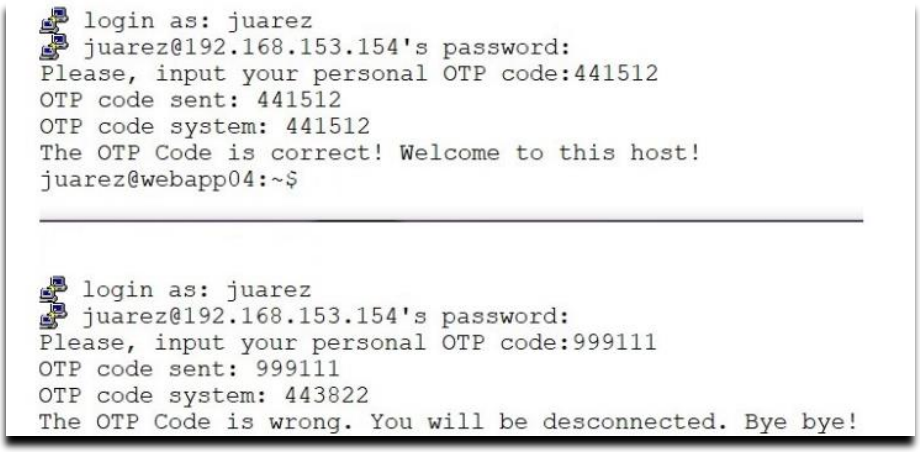

Figura 76 - SSH centralizado com autenticação OTP. Fonte: do autor

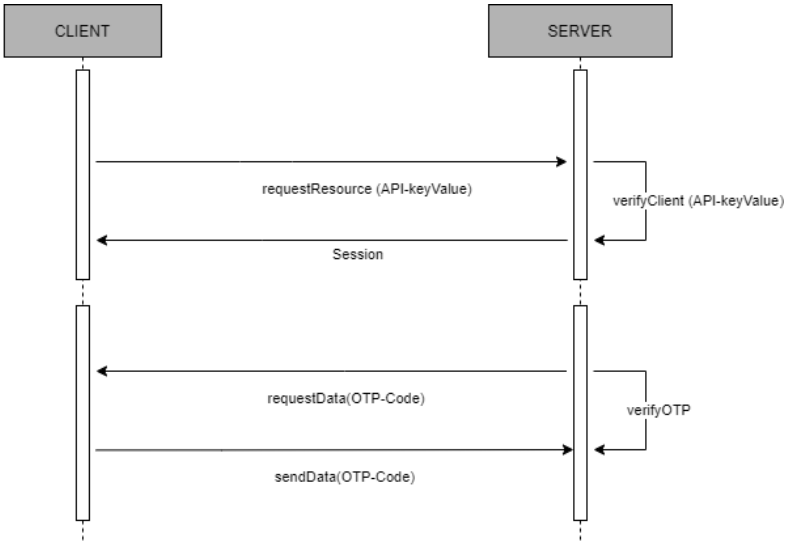

Figura 8 – Meta-protocolo e autenticação de API. Fonte: do autor.

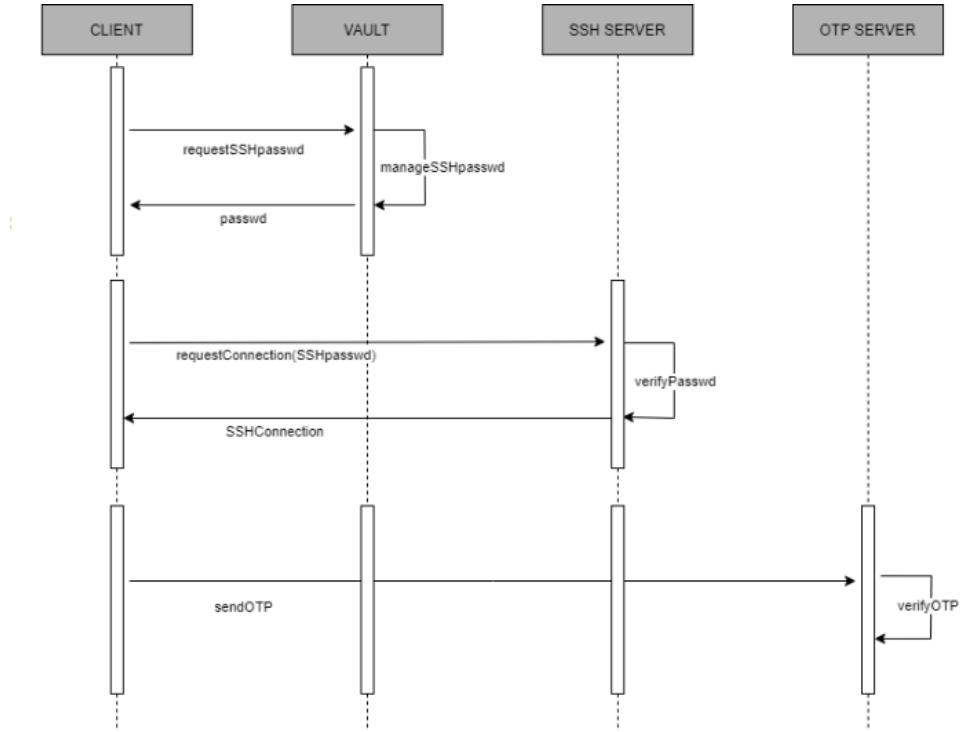

Figura 09 – Meta-protocolo de autenticação do SSH. Fonte: do autor.

Paralelo a autenticação da API, o usuário se autentica utilizando um provedor de identidade (IdP) 10.

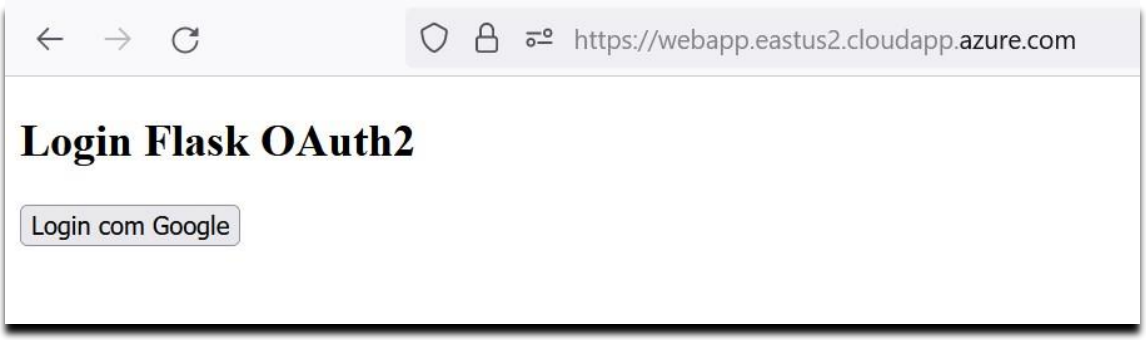

Figura 10 - Tela da aplicação web de teste. Fonte: do autor.

Em seguida, ocorre mais uma vez uma autenticação com o OTP, de forma isolada (Figura 11), para garantir que as credenciais de acesso não foram comprometidas. Esta revalidação da autenticação com o OTP pode ocorrer não só no login, mas periodicamente.

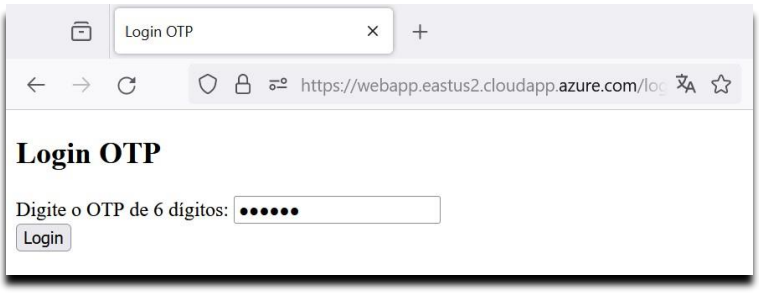

Figura 11 - Tela de login do OTP. Fonte: do autor.

Caso não haja nenhum problema com a autenticação do IdP e o usuário também digite o código OTP correto (código gerado por nossa aplicação Python), o usuário ganha os direitos de acesso (Figura).

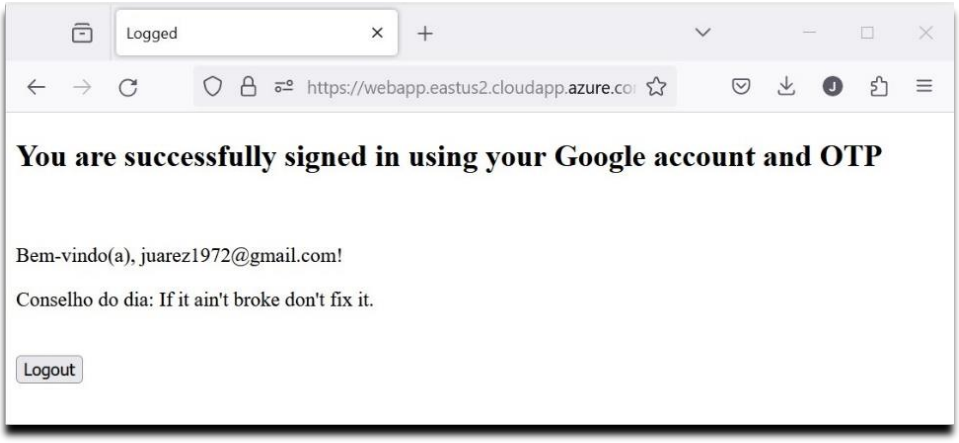

Figura 12 - Tela do Servidor Web após login. Fonte: do autor.

Mesmo com autorização do IdP, mas sem o código OTP correto, o usuário é redirecionado novamente para a tela de login do provedor de identidade, conforme apresentado nos logs do servidor web (figura 13).

```
WARNING: This is a development server. Do not use it in a production deployment. Use a
 production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 582-713-822
127.0.0.1 - - [02/Jan/2024 12:07:55] "GET /login HTTP/1.1" 302 -
Dados do user_info.data :< {'id': '117821956065434764353', 'email': 'juarez1972@gmail.
com', 'verified_email': True, 'picture': 'https://lh3.googleusercontent.com/a-/ALV-UjX
PtcuUMktmQ0DrpFL7YBZviZTGVAuRhawtn6G8L3C6E5xd=s96-c'} >
Dados da headers do Token JWT:< {'Authorization': 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6
IkpXVCJ9.eyJlbWFpbCI6Imp1YXJlejE5NzJAZ21haWwuY29tIn0.Gx0B0K1dV2KKz9Q2vg_Uz8ZLejEqIT_Rv
Vy6ESdVTn8'} >
127.0.0.1 - - [02/Jan/2024 12:07:59] "GET /login/authorized?code=4%2F0AfJohXmab-5-MzB4
duK4vq3ztRDh25Z1gKalM5KD67RfzixzPoHTsivws4f4hDUOvt7Scw&scope=email+openid+https%3A%2F%
2Fwww.googleapis.com%2Fauth%2Fuserinfo.email&authuser=0&prompt=none HTTP/1.1" 200 -
127.0.0.1 - - [02/Jan/2024 12:08:04] "POST /login/otp-token HTTP/1.1" 302 -
127.0.0.1 - - [02/Jan/2024 12:08:04] "GET / HTTP/1.1" 200 -
```

Figura 13 - Logs do Servidor Web com redirect após OTP incorreto. Fonte: do autor.

A integração entre a aplicação OTP no servidor seguro e o Vault foi feita utilizando-se a biblioteca HVAC (Hvac, 2023). Assim, podemos inserir a senha da conexão inicial sem armazenar em um arquivo ou variável de ambiente, conforme trecho de código nafigura 14.

```
#init_server()
import hvac
import sys
import os

client = hvac.Client(url='https://192.168.153.200:8200')
print(f" Is client authenticated: {client.is_authenticated()}")

kv_configuration = client.secrets.kv.v2.read_configuration(
    mount_point='kv',
)
print('Config under path "kv": max_versions set to "{max_ver}"'.format(
    max_ver=kv_configuration['data']['max_versions'],
))
```

Figura 14 - Código para conexão com o vault. Fonte: adaptado de Hvac (Hvac, 2023)

Outra possibilidade é usar a esteganografia, ocultar as informações de conexão em um arquivo, dificultando o invasor para capturar as credenciais. Se as credenciais forem apenas inseridas, se o aplicativo for reiniciado, ele solicitará que o operador atue novamente.

Neste trabalho, utilizamos um Contêiner Docker com INTEL Gramine GSC (Intel Gramine GSC, 2023), o que pode nos trazer portabilidade e menor dependência do sistema operacional hospedeiro.

Instalar clientes OTP e monitorar o uso de aplicativos foi possível em uma rede local. Para ser estendido a clientes web externos, para os quais a autenticação sem interação não é possível, o usuário precisa inserir o Código OTP de vez em quando. Isto pode levar a algum uso hostil, mas aumentar significativamente a segurança (Figura);

Na Figura temos a conferência de um código OTP um aplicativo de OTP executando no TEE, cuja validação ocorreu com sucesso:

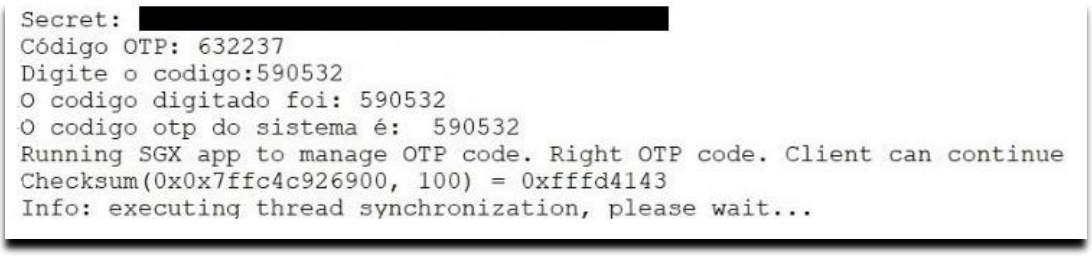

Figura 15 – Autenticação no servidor OTP OK. Fonte: do autor.

Na figura 16 é mostrada uma validação sem sucesso de um código OTP para um aplicativo de OTP executando no TEE.

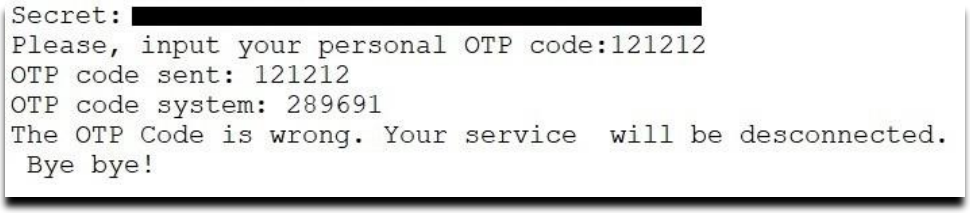

Figura 16 – Autenticação no servidor OTP não OK. Fonte: do autor.

Na Figura , podemos verificar a rede SDWAN criada após a autenticação do host Kali Linux (*headless mode*) no Twingate. Isso permite o acesso a outros hosts, de acordo com a política criada, independente de qual rede local se encontram.

```
┌──(root㉿ kali)-[/home/juarez]
└─# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qle
n 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group def
ault qlen 1000
    link/ether 08:00:27:cb:7e:f5 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
       valid_lft 78834sec preferred_lft 78834sec
    inet6 fe80::5024:9282:1e23:521e/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group def
ault qlen 1000
    link/ether 08:00:27:c8:17:3b brd ff:ff:ff:ff:ff:ff
    inet 192.168.153.100/24 brd 192.168.153.255 scope global dynamic noprefixroute et
h1
       valid_lft 535sec preferred_lft 535sec
    inet6 fe80::b263:70d1:5799:f1ec/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
4: sdwan0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UN
KNOWN group default qlen 500
    link/none
    inet 100.96.0.2/32 scope global noprefixroute sdwan0
       valid_lft forever preferred_lft forever
    inet6 fe80::cbeb:5d22:ac0a:828c/64 scope link noprefixroute
       valid_lft forever preferred_lft forever

┌──(root㉿ kali)-[/home/juarez]
└─# twingate status
online
```

Figura 17 - Rede SDWAN estabelecida com o Twingate. Fonte: do autor.

O modo *headless* do Twingate é uma forma de autenticar o host, por meio de uma chave de API, sem a necessidade de se digitar uma senha de usuário.

5.2. Avaliação

Considerando um invasor na Internet, com acesso apenas ao servidor web em porta 443, mesmo que ela consiga capturar ou ter acesso ao login do usuário e senha, o MFA (SSO) do provedor deve reduzir significativamente a superfície de exposição.

Se o invasor conseguir roubar a sessão, seja por fraquezas no navegador ou implantando malware que explora as 10 principais vulnerabilidades listadas na OWASP (OWASP Top10, 2023), por exemplo, a autenticação adicional realizada pelo software de *token* OTP desconecta automaticamente a conexão após decorrido o tempo configurado.

Para autenticação API, que em nossa proposta está em um host diferente, sem acesso externo da Internet, usamos apenas autenticação de *token* OTP sem interação humana, o que impede o invasor de manter a conexão – mesmo em rede local e de posse dos dados de autenticação (chave, secreto), pois o *token* OTP é enviado junto com a solicitação e pode usar esteganografia para tornar sua descoberta ainda mais difícil.

Outra alternativa adotada neste trabalho é enviar o código TOP fora da banda (em paralelo e em outro canal). Em outras palavras, toda vez que uma solicitação autêntica de código é feita através de uma API, um código HOTP é enviado de acordo como sequência de códigos OTP enviados durante a interação do código com o servidor.

Portanto, se um invasor tentar invocar uma API (ou seja, um intruso fingindo ser um código de programa autêntico), ela não terá o código de sequência HOTP correto para autenticar a solicitação e a operação não será confirmada.

Foi implementada a validação manual do código de sequência HOTP em nossa proposta para acompanhar essas ações. Para mais detalhes, veja a Figura : – verificação de um código OTP de um aplicativo OTP em execução no TEE, que foi validado com sucesso, e Figura : – verificação de um código OTP de um OTP aplicação em execução no TEE, que foi não validada com sucesso.

5.3. Considerações finais:

A criação de um ambiente computacional para apresentação de um conceito foi tarefa árdua, principalmente para a utilização de aplicações em ambiente de computação confiável.

Devido à sua pouca adoção, os principais materiais disponíveis, assim como a base de conhecimento para correção de bugs é feita principalmente pelo fabricante.

Ainda assim, foi possível executar todas as ferramentas, aplicativos e configurações propostas.

Capítulo 6
Análise de segurança

Este capítulo apresenta os testes realizados para comprovar a eficiência das camadas de proteção planejadas para o modelo.

6.1. O modelo de Ameaças

Podemos descrever um modelo de ameaças de acordo com a OWASP (OWASP), decompondo o ambiente apresentado na Figura , o qual é composto por aplicações web, serviço de SSH nos hosts, um servidor de aplicação OTP, um servidor Vault, um provedor de identidades IdP externo e um gateway de *Zero Trust* também externo.

Determinamos as principais ameaças como a quebra de controle de acesso (OWASP Top10, 2023), que se refere a um conjunto de 34 CWE (*Common Weakness Enumerations*) como violações do princípio do menor privilégio, ultrapassagem de controles de acesso por meio de alteração de URL, acesso indevido à APIs por problemas de configuração, manipulação de metadados de cookies e tokens, entre outros.

A captura de pacotes na rede, onde os dados podem ser alterados ou acessados de forma indevida.

O escaneamento de portas, onde o atacante consegue escanear os serviços publicados pelos hosts, visando localizar vulnerabilidades específicas de cada serviço.

A elevação de privilégios, quando o atacante consegue personificar um usuário de serviço ou um usuário administrador para ter acesso a configurações, dados ou executar aplicativos de forma indevida.

Os ataques de força bruta, onde o atacante se vale de listas de senhas e usuários para tentar o login nas aplicações.

As principais dependências externas identificadas são o Provedor de Identidades (Idp); o firewall de Internet; o gateway de Zero Trust; o Vault (cofre de senhas); e ainda as bibliotecas python PyOTP e HVAC.

Como ativos *(assets)* temos Servidores Web; Servidor de aplicação Python (OTP); Usuário anônimo; Usuário Autenticado pelo IdP; Usuário Administrador do site; e o Usuário de banco de dados;

Os ativos possuem os seguintes níveis de confiança:

a) Um usuário anônimo consegue visualizar a página de login na aplicação Web, já que é a única parte da arquitetura publicada na Internet;

b) Um usuário autenticado pelo IdP e pelo MFA gerado pela aplicação OTP consegue visualizar os recursos da aplicação web, como os providos por APIs e bancos de dados, desde que as permissões tenham sido concedidas pelo administrador do site;

c) Um usuário autenticado no processo de autenticação do SSH, que compreende um IdP, um token fornecido pelo Vault e um token MFA fornecido pela aplicação OTP, consegue acesso ao shell do host, mas só executa aplicações de privilégio elevado se estiver no grupo Sudo;

d) Um usuário administrador do site pode visualização dos recursos de bancos de dados, APIs e adicionar permissões para novos usuários;

e) Um processo de usuário do servidor web (www-data) consegue visualização dos tokens de autorização e tokens OTP.

6.2. Avaliação e testes

A segurança baseada em camadas é recomendada como boas práticas em segurança cibernética. Os procedimentos (mitigação no ATT&CK do MITRE) são projetados para garantir que se uma camada (por exemplo, categoria de ferramenta) falhar, não deverá comprometer a segurança geral.

Nós consideramos nosso modelo mais seguro porque adicionamos duas camadas: (i) razoável proteção de dados em um TEE para aplicações de sistema e (ii) implementação de outro fator de autenticação com OTP, fora de banda (paralelo) ao padrão autenticação de aplicativo/API.

Também implementamos regras tradicionais de acesso externo através do firewall para permitir acesso às portas TCP para o serviço web e internamente para SSH, Vault e o Servidor OTP. O acesso externo para clientes aplicativos só poderá ser feito através de um único ponto (servidor web), e SSH, Vault e o gerador OTP não são acessados diretamente. O objetivo foi reduzir a superfície de exposição.

Avaliamos se o nível de segurança era o esperado. Para isso foram conduzidos testes de intrusão para comparar e identificar as capacidades do invasor, o que poderia comprometer a superfície de exposição. Conduzimos uma varredura externa para avaliar as portas, serviços e vulnerabilidades.

O teste foi realizado utilizando a ferramenta NMAP da internet. O servidor web – único ponto de contato direto com a Internet, revelou porta 443 (HTTPS) aceitando conexões com a Internet. Portas 4443 (aplicativo OTP) e 8201 (key vault) foram identificados internamente. Além disso, a porta 22 (SSH) foi acessível. Isso significa que a segurança baseada em firewall funcionou corretamente.

Outros testes visam imitar as ações do adversário, considerando o alcance dos servidores expostos na Internet e, em outros casos, um *insider* na rede local. Neste caso, utilizamos algumas das ferramentas descritas no PTES (PTES, 2023) “Exploração Personalizada” e “Pegadas Externas”.

Para avaliar a possibilidade de roubo de credenciais ou outras informações enviadas pela rede local, coletamos dados usando TCPDump com acesso ao web servidor. Neste caso foi possível perceber a criptografia dos dados em trânsito, o que dificulta o acesso imediato aos dados transmitidos na rede, seja por meio do protocolo HTTPS ou por meio de uma conexão TLS ao soquete OTP.

Foi realizado um *dump* de memória para avaliar a possibilidade de acessar o OTP na memória. Embora esta seja uma possibilidade de roubo de informações, este tipo de ataque requer um nível de acesso privilegiado (atacante com altas capacidades) e bibliotecas de suporte instaladas no sistema de destino.

Depois de executar o despejo de memória com o contêiner Gramine GSC, executamos os testes com a ferramenta Volatilidade. O teste confirmou que era impossível acessar o conteúdo da memória como texto não criptografado devido à criptografia SGX.

Primeiro atacamos o servidor SSH sem configuração de OTP, utilizando um usuário válido, avaliando a possibilidade de acesso não autorizado por meio de um dicionário de senhas. Após alguns minutos, o acesso foi revogado e a conexão foi interrompida.

O SSH teve uma configuração padrão neste teste, com usuário e senha. Esse teste foi realizado da seguinte forma: usando uma lista de usuários que continha um dos válidos usuários para a conexão, rodamos a ferramenta Hydra, que acessou com sucesso o servidor SSH. Neste teste, não houve outras configurações de segurança no servidor SSH, como limites de tentativas, blocos de IP de origem ou autenticação MFA no PAM.

Num segundo teste do servidor SSH, para avaliar a nova camada de proteção estabelecido com o OTP, consideramos que o invasor é um *insider* ou obteve algum acesso explorando outra vulnerabilidade CVE em um dos sistemas internos servidores.

Usando a ferramenta Metasploit em um Kali Linux com acesso à rede, atacamos com nomes de usuário e senhas válidos fornecidos pelo Vault. A conexão foi bem-sucedida porque foi a primeira tentativa de login.

Quando a tentativa foi repetida, a senha já havia sido alterada. Isso ocorre porque o Vault altera a senha para cada conexão. Então, após autenticação SSH válida, um pós-script verificou o código OTP em execução no TEE, interrompendo a conexão.

Nos testes com Metasploit (Rapid7 Metasploit, 2023), embora a sessão possa ser criada quando o usuário e os logins corretos forem obtidos, o Meterpreter não poderá obter um *shell.*

Foi conduzido um ataque de força bruta com a ferramenta Hydra para avaliar a tentativa acessar o servidor web, considerando que o adversário está na Internet, com a porta 443 exposta. O ataque usou uma lista de usuários e senhas válidas e obteve acesso sem configurar o OTP.

Neste caso, os ataques de força bruta não produziram resultados, pois os provedores de identidade (IAM/IdP) aceitam apenas algumas tentativas de conexões para o mesmo usuário. Ainda assim, devemos considerar a possibilidade de que o adversário pode ter adquirido as credenciais por outros meios, como engenharia social.

A seguir, habilitamos o segundo fator com OTP para estabelecer uma re-autenticação processar e encerrar a conexão imediatamente se não tiver êxito. É possível solicitar ao usuário uma re-autenticação periódica além da autenticação padrão atualização de token.

No entanto, isso pode gerar algumas experiências hostis para o usuário, devendo haver uma análise criteriosa do aumento da segurança e usabilidade ou interferência na produtividade, pois pode causar indisponibilidade, o que também é um problema de segurança.

Para avaliar a metodologia de autenticação contínua usando APIs sem interação humana, consideramos o cliente OTP autenticado periodicamente enviando seu código OTP (fora da banda). No servidor OTP em execução no TEE, o serviço verifica se o código da sequência HOTP é válido e executa alguma ação pré-configurada se não estiver.

Neste caso, forjamos o envio de um código incorreto. O servidor executou a tarefa esperada corretamente, exibindo uma mensagem e executando um script. O tempo de verificação de autenticação entre o servidor web e o gerador OTP depende do serviço que está sendo considerado. Por exemplo, no caso do SSH, uma verificação a cada minuto pode ser suficiente.

No caso de um banco de dados, alterar a senha ou queda de conexão a cada minuto pode fazer com que o serviço seja indisponível e, desta vez, deve demorar algumas horas. No caso de acesso à API, um minuto também pode ser suficiente se apenas o acesso de um servidor para outro é considerado.

Se o acesso for de um cliente externo, um tempo maior poderá ser mais apropriado. Isso significa que o tempo da sessão é um parâmetro que precisa a ser definido pelo administrador para cada caso.

No acesso SSH em servidores Linux, é sempre apropriado colocar outras proteções barreiras que impedem tentativas de acesso de força bruta, como Fail2ban (Fail2ban, 2023) ou Proteção EDR (*Endpoint Detection and Response*).

Outras travas de segurança podem ser aplicadas, como configurar os arquivos *hosts.deny* e *hosts.allow* em SSH e regras de correspondência de localização no servidor web Apache2, por exemplo. O uso de firewall regras e proteção em hosts também são altamente recomendadas.

Nos servidores web pode ser impostas ainda outras regras para um ambiente de produção, como por exemplo a aplicação das políticas de *Headers X-Frame-Options*, *Content Security Policies* e *Same-origin*, cujas boas práticas de configuração podem ser encontradas no site da Fundação Mozilla (Mozilla, 2024).

O uso do Vault também se mostrou muito adequado, com diversas possibilidades para integração com scripts e SSH, reduzindo significativamente o acesso a credenciais que foram previamente armazenados como texto não criptografado em arquivos, variáveis de ambiente, entre outros, e que foram utilizados diversas vezes devido aos longos tempos de validade das credenciais.

Infelizmente não testamos o cofre de senhas do Vault , pois não queremos discutir possíveis vulnerabilidades nesta solução. No entanto, este controle de segurança está previsto nos controles CIS e outras estruturas como prática recomendada, sendo o Hashicorp Vault, utilizado no protótipo, uma das ferramentas mais utilizadas no mercado.

A utilização de diversas camadas de proteção impõe ao intruso/atacante uma maior capacidade de explorar uma superfície de exposição restrita e controlada. Isso faz com que seja difícil para eles atingirem seus objetivos, seja pela proteção imposta por cada camada de segurança, ou porque o tempo se torna um dos fatores críticos nos desafios a serem superados devido às frequentes mudanças de códigos/senhas que pode ocorrer com cada acesso.

O objetivo da proposta que motivou a questão de pesquisa, foi avaliar como adicionar um novo mecanismo de proteção envolvendo criptografia de memória e autenticação *out-of- band*, com mecanismos de OTP aumentaria a segurança do sistema baseado em um sistema de PAM (Vault), sem diminuir a produtividade e usabilidade do cliente.

Testes efetuados com uso da solução de *Zero Trust* Twingate (TWINGATE, 2023), a partir de um host com Kali Linux, conseguimos verificar que, após a autenticação do host do Kali (*headless mode*) Figura , o acesso por SSH pode ser feito para um servidor web hospedado em nuvem pública, mesmo com a porta 22 aberta apenas para o ambiente interno.

No escaneamento de portas com NMAP aparecem a maioria das portas TCP como abertas e as portas que o host tem acesso como filtradas. Dessa forma, quando o serviço de autenticação do Zero Trust é parado, nenhuma porta mais é mostrada, comprovando a redução da superfície de ataque com o Zero Trust. O ponto de vista utilizado aqui foi de um atacante (*insider*), que estava dentro de um host autenticado e com permissão de acesso ao host testado.

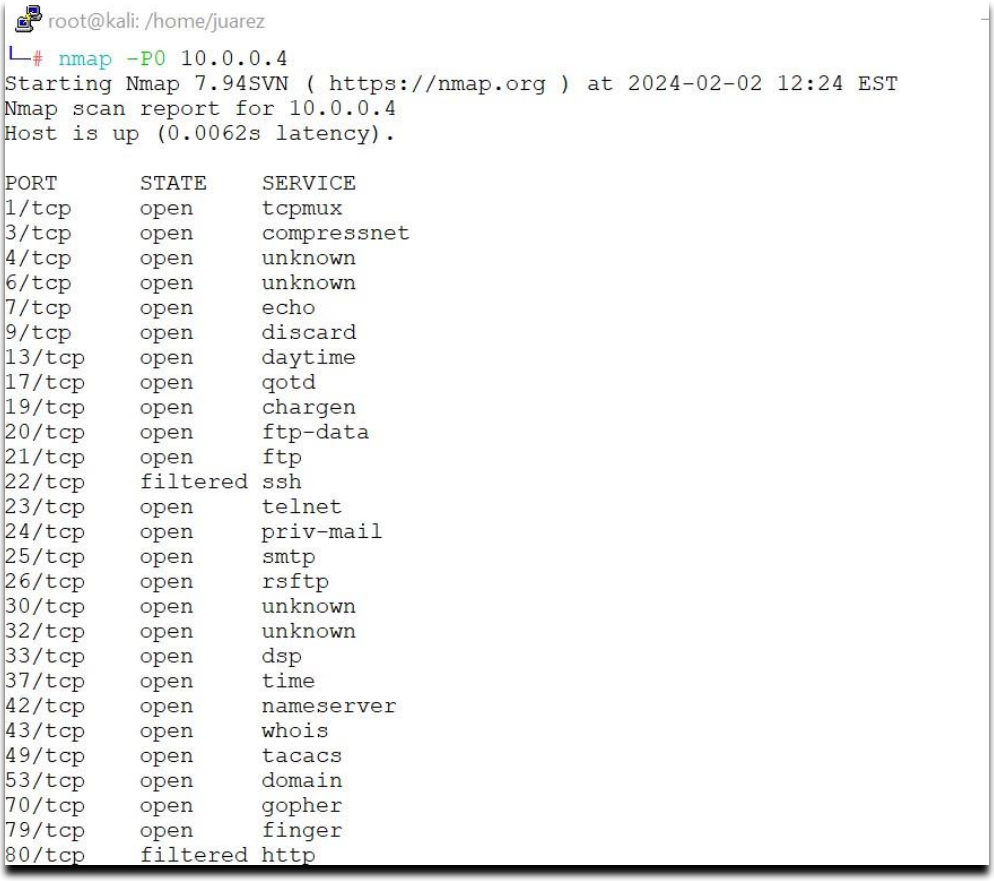

Figura 18 - Nmap para webserver com zerotrust. fonte: autor.

Para responder à questão de pesquisa, conseguimos inserir um protocolo de autenticação mecanismo que funciona em paralelo com a abordagem tradicional sem mudando o modus operandi dos mecanismos baseados em PAM na proposta modelo. Portanto, não alteramos o tempo de validade das credenciais PAM (no mercado a validade média é de 8 horas).

Ainda assim, com o mecanismo OTP, as validações são feitas fora da banda com tempo mais frequente, em torno de unidades de por minuto, aumentando significativamente a necessidade do invasor de capacidades para explorar superfície de exposição do sistema. A validade temporal do novo mecanismo também pode ser configurada de acordo com a necessidade de cada aplicação.

Pelo tipo dos testes executados e do ambiente controlado, consideramos que não é possível se estabelecer uma análise quantitativa sobre os ganhos de segurança, apenas qualitativo nos aspectos testados. Para se estabelecer uma análise quantitativa seria necessário a exposição dos ambientes por um tempo prolongado aos atacantes, ou ainda

ser aplicada a arquitetura a um ambiente real em produção que já possuísse histórico de incidentes registrado.

6.3. Determinação das contramedidas e mitigação

Algumas contramedidas são indicadas de acordo com o modelo apresentado, sendo que algumas foram implementadas somente devido ao modelo proposto de melhoria.

A autenticação em IdP externo permite terceirizar parte da estrutura de segurança para uma entidade externa confiável. Isto permite aumentar o foco na proteção de outras partes da arquitetura.

O uso de mais uma camada de MFA (e.g. OTP) independente do IdP, a qual implementamos por meio de uma aplicação que roda em um ambiente de computação confiável, permite colocar uma barreira extra na proteção de usuários.

A criptografia dos dados em trânsito, feita por meio do uso de HTTPS e TLS visa impedir o acesso aos dados que transitam pela rede. Contudo, configurações extras de autenticação e configurações seguras nos serviços de SSH e WEB se fazem necessário. Utilizamos as configurações seguras adotadas pela Fundação Mozilla (Mozilla, 2024) para o servidor Apache2.

Um gateway Zero Trust para usuários internos e acesso aos ativos da rede local permitiu a redução da superfície de ataque, permitindo que somente hosts e usuários autenticados tenham acesso inicial a serviços como SSH e APIs.

O uso de TEE para proteção da aplicação Python de OTP impede ataques de acesso direto à memória, fazendo com que os dados da aplicação sejam criptografados durante o uso.

Para o protocolo SSH, além da utilização do Vault (agente *vault-ssh-helper*), que emite senha de acesso de uso único para cada sessão, nosso modelo implementou uma segunda autenticação por MFA por meio do aplicativo OTP. Essa segunda etapa foi implementada após o processo de autorização de PAhM *(Pluggable Authentication Modules)* do Linux. Além disso, o acesso de login do usuário *root* foi desabilitado. Um limite de 5 tentativas de conexão estabelecido.

Um Firewall no ambiente de nuvem permite a publicação apenas das portas 80 (usada para redirecionamento) e 443 (HTTPS) para Internet. Desta forma, os usuários, que não

utilizam o gateway Zero Trust, acessem somente a aplicação Web e não tenham acesso à porta do SSH.

Capítulo 7
Conclusão

A execução de aplicativos em um ambiente de execução confiável pode ser um desafio para administradores de sistemas de computador, pois adiciona mais variáveis ao esquema, especialmente quando consideramos a necessidade de gerenciamento rotineiro de vulnerabilidades, motivo pelo qual escolhemos utilizar o libOS Gramine.

O uso do Vault para credenciais já está consolidado nos principais setores de segurança cibernética frameworks, e é uma prática recomendada que deve ser observada. Ajustando a sessão tempos de aplicações e serviços não é uma tarefa simples, mas é essencial para proteção do ambiente.

Como usamos uma estratégia de validação TOP fora da banda em nosso trabalho, testamos tempos baixos e funcionou corretamente, embora estes os parâmetros são configuráveis. Esta estratégia também desafia os atacantes que deveriam obter o controle deste sistema para comprometer a segurança da proposta baseada em OTP.

O aplicativo servidor OTP, criado em Python, permitiu a validação da autenticação do host, sem interação humana, num modelo que também implementa o princípio do Zero Trust de ‘nunca confie, sempre verifique.

De maneira complementar, a implantação do Zero Trust com o Twingate permitiu um nível menor de exposição da arquitetura.

O mecanismo proposto mostrou-se viável para os objetivos do projeto, principalmente porque a análise de segurança e os testes de invasão mostraram que a segurança aumentou e não houve alteração no tempo de validade do PAM, favorecendo a usabilidade da segurança e o usuário produtividade.

Para trabalhos futuros vislumbramos a possibilidade de criação de um serviço de vault em TEE, assim como é possível desenvolver uma outra solução de gateway Zero Trust que funcione somente em ambiente *on-premisses*.

Referências

Iordache, C. A., Dragomir, A. V., & Marian, C. V. (2022). Public Institutions Updated Enhanced Biometric Security, Zero Trust Architecture and Multi-Factor Authentification. *2022 International Symposium on Electronics and Telecommunications (ISETC)*, (pp. 1-4). Timisoara - Romania.

Auth0. (2023). *Auth0 Oauth2*. Fonte: Auth0: https://auth0.com/pt/intro-to-iam/what-is-oauth-2

Bobbert, Y., & Scheerder, J. (2022). Zero Trust Validation: from Practice to Theory. *2022 IEEE 29th Annual Software Technology Conference (STC).*

Buck, C., Olenberger, C., Schweizer, A., Völter, F., & Eymann, T. (2021). Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust. *Science Direct*.

Bursell, M. (2021). *Trust in Computer Systems and The Cloud.* New Jersey: Wiley.

CCC. (2023). *A Technical Analysis of Confidential Computing.* Fonte: Confidential Computing Consortium: https://confidentialcomputing.io/wp-content/uploads/sites/10/2023/03/CCC-A-Technical-Analysis-of-Confidential-Computing-v1.3_unlocked.pdf

Cheng, H., Li, W., Wang, P., Chu, C.-H., & Liang, K. (2021). Incrementally Updateable Honey Password Vaults. *30th USENIX Security Symposium.*

CIS Security. (2023). *CIS Security Controls*. Fonte: CIS Security Controls: https://www.cisecurity.org/controls

Fail2ban. (2023). *Fail2ban*. Fonte: Fail2ban: https://github.com/fail2ban/fail2ban

Falcão, E., Silva, M., Luz, A., & Brito, A. (2022). Supporting Confidential Workloads in SPIRE. *IEEE International Conference on Cloud Computing Technology and Science (CloudCom).*

Garbis, J., & Chapman, J. W. (2021). Privileged Access Management. Em J. Garbis, & J. W. Chapman, *Zero Trust Security - An Enterprise Guide.* Berkeley - CA: Apress.

Hashicorp. (2023). *Hashicorp Vault*. Fonte: Hashicorp Vault: https://developer.hashicorp.com/vault/docs/install

Henricks, A., & Kettani, H. (2019). ACM ISSM. *On Data Protection Using Multi-Factor Authentication*.

Houssain, H., Houssain, A., Houssain, Z., Sohag, H. I., & Hahman, S. (2018). OAuth-SSO: A Framework to Secure the OAuthbased SSO Service for Packaged Web Applications. *17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications*.

Hvac. (2023). *hvac*. Fonte: hvac: https://hvac.readthedocs.io/en/stable/#

Intel. (2023). *Intel SGX*. Fonte: Intel SGX: https://www.intel.com/content/www/us/en/architecture-and-technology/software-guard-extensions.html

Intel Download01. (2023). *Download01 Intel SGX*. Fonte: Download01 Intel SGX: https://download.01.org/intel-sgx/

Intel Gramine. (2023). *Gramine Project*. Fonte: Gramine Project: https://gramineproject.io/

Intel Gramine GSC. (2023). *Gramine GSC Project*. Fonte: Gramine GSC Project: https://gramine.readthedocs.io/projects/gsc/en/latest/

Klieme, E., Wilke, J., Dornick, N. v., & Meinel, C. (2020). FIDOnuous: A FIDO2/WebAuthn Extension to Support Continuous Web Authentication. *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom).*

Klieme, E., Wilke, J., Dornick, N. v., & Meinel, C. (2020). FIDOnuous: A FIDO2/WebAuthn Extension to Support Continuous Web Authentication. *IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom).*

Mitre. (2023). *CVE-2022-31813*. Fonte: Mitre CVE: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-31813

Mitre. (2023). *Mitre Att&ack Reconnaissance*. Fonte: Reconnaissance: https://attack.mitre.org/tactics/TA0043/

Mitre. (2023). *Mitre Att&ck Home*. Fonte: Mitre Att&ck: https://attack.mitre.org/

Mitre. (2023). *Mitre Att&ck Remote Services SSH*. Fonte: Remote Services: SSH: https://attack.mitre.org/techniques/T1021/004/

Mitre. (2023). *Mitre Att&ck TTP 1111*. Fonte: TTP 1111: https://attack.mitre.org/techniques/T1111/

Mitre. (2023). *Mitre Att&ck TTP 1550*. Fonte: TTP1550: https://attack.mitre.org/techniques/T1550/

Mitre. (2023). *Mitre CVE*. Fonte: Mitre CVE: https://cve.mitre.org/

Mitre. (2023). *Mitre TTP Based Hunting*. Fonte: Mitre TTP Based Hunting: https://www.mitre.org/sites/default/files/2021-11/prs-19-3892-ttp-based-hunting.pdf

Mozilla. (2024). *Mozilla Foundation.* Fonte: Mozilla Developer: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers

Muhammad El-Hindi, T. Z. (2022). Benchmarking the Second Generation of Intel SGX Hardware. *Data Management on New Hardware (DaMoN'22),*.

NIST. (2020). *Special Publications.* Fonte: nvlpubs.nist.gov: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf

NIST. (2023). *Nist NVD*. Fonte: Nist NVD: https://nvd.nist.gov/vuln

Oliveira, J. d., Santin, A. O., Viegas, E. K., & Horchulhack, P. (2024). A non-Interactive One-Time Password-based Method to Enhance the Vault Security. *Advanced Information Networking and Applications - AINA* (pp. 201-213). Springer Nature Switzerland.

OneIdentity. (2023). *Enhanced Privilege Access Management.* Fonte: OneIdentity KupppingerCole WhitePaper: https://www.oneidentity.com/whitepapert/white-paper-kuppingercole-report-highlights-the-need-for-enhanced-pam-8142554

OWASP. (s.d.). *OWASP Threat Modeling.* Fonte: OWASP: https://owasp.org/www-community/Threat_Modeling_Process

OWASP Top10. (2023). *Owasp Top10*. Fonte: Owasp Top10: https://owasp.org/www-project-top-ten/

Peral, A. B., Orozco, A. S., García Villalba, L. J., & Kim, T.-h. (2018). Distributed One Time Password Infrastructure for Linux Environments - Entropy 2018. *Entropy 20, no. 5: 319. https://doi.org/10.3390/e20050319*, 5.

PTES. (2023). *Pentest Standard*. Fonte: Pentest Standard: http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines

Purba, A., & Soetomo, M. (2018). Assessing Privileged Access Management (PAM) using ISO. *Proceedings of Annual Conference on Management and Information Technology (ACMIT) 2018.*

Python PyOTP. (2023). *Python PyOTP*. Fonte: Python PyOTP: https://pyauth.github.io/pyotp/

Rapid7 Metasploit. (2023). *Metasploit*. Fonte: Metasploit: https://www.metasploit.com/

Tuononem, H. (2023). Privileged access management model for a managed service provider. *Master's Degree Programme in Information Technology, Cyber Security. Master´s thesis*.

TWILIO. (2023). *Twilio TOTP*. Fonte: Twilio: https://www.twilio.com/docs/glossary/totp

TWINGATE. (2023). *Twingate.* Fonte: Twingate: https://www.twingate.com/docs

Weidman, G. (2021). *Penetration Testing: A hands-on Introduction to Hacking.* São Paulo - SP: Novatec.

Weingärtner, , R., & Westphal, C. M. (2014). Enhancing Privacy on Identity Providers. *SECURWARE - The Eighth International Conference on Emerging Security Information, Systems*, (pp. 84 - 88).

Wu, L., H.J., C., & Li, H. (2021). SGX-UAM: A Secure Unified Access Management Scheme With One Time Passwords via Intel SGX. *IEEE Access*.