Abordagem confiança zero aplicada a ambientes computacionais *big data*: um estudo de caso*

Júnia Maísa Oliveira[1], **Vinícius Rodrigues Oliveira**[1],
Daniel Fernandes Macedo[1], **Dorgival Guedes**[1] **José Marcos Nogueira**[1]

[1]Departamento de Ciência da Computação - Instituto de Ciências Exatas
Universidade Federal de Minas Gerais (UFMG)
CEP 31270-901 – Belo Horizonte, MG, Brazil

{juniamaisa, oliveiravinicius, damacedo, dorgival, jmarcos}@dcc.ufmg.br

Resumo. *O tema deste artigo é o estudo e a aplicação dos princípios da abordagem de confiança zero (*zero trust*) para segurança cibernética a um ambiente de* big data*. A partir da definição de um ambiente computacional* big data *baseado em um ambiente real de produção, definimos uma arquitetura de segurança baseada na confiança zero e a instanciamos para o ambiente em questão, abordando alguns casos de autenticação e autorização. Para a instanciação da arquitetura, várias ferramentas de software são utilizadas, diversas delas já disponíveis no ambiente real. As soluções propostas são ilustradas por casos de uso. A conclusão é que essa a tarefa é complexa, porém factível, e que pode ser implementada com as ferramentas usuais de ambientes* big data*.*

Abstract. *The subject of this article is the study and application of the principles of the zero trust approach to cybersecurity in a* big data *environment. From the definition of a* big data *computing environment based on a real production environment, a zero trust security architecture is defined and instantiated for the environment in question. For the instantiation of the architecture, several software tools are used, several of them already available in the environment of reference. The proposed solutions are illustrated by use cases, approaching in some cases authentication and authorization. The conclusion is that this task is complex, but feasible, and that it can be implemented with good use of the usual tools of* big data *environments.*

1. Introdução

A confiança zero (Zero Trust) é um paradigma de segurança cibernética que focaliza a proteção de recursos (ativos), sendo baseada na premissa de que a confiança nunca é concedida implicitamente, mas deve ser avaliada continuamente. A Arquitetura de Confiança Zero é uma abordagem ponta-a-ponta para a segurança de dados e recursos de uma instituição, que abrange identidade (entidades pessoais e não pessoais), credenciais, gerenciamento de acesso, operações, terminais, ambientes de hospedagem e infraestrutura de interconexão[S. Rose et al. 2020].

Nos ambientes computacionais, de maneira geral e tradicionalmente, os esquemas de proteção dos ativos concentram-se na defesa de perímetro, na qual os usuários,

*Gostaríamos de agradecer o suporte das seguintes agências brasileiras de fomento: CAPES, CNPq, FAPEMIG e FAPESP.

uma vez autenticados, recebem autorização de acesso a uma ampla coleção de recursos. Como resultado, pode ocorrer movimento lateral não autorizado dentro da rede, o que tem sido um dos maiores desafios de segurança para as organizações. Os *firewalls* de perímetro provêm *gateways* normalmente robustos, o que ajuda a bloquear os invasores da Internet.Entretanto, são menos úteis para detectar e bloquear ataques de dentro da rede e não podem proteger os usuários fora do perímetro (por exemplo, trabalhadores remotos e serviços baseados na nuvem).

Esse problema torna-se particularmente crítico em ambientes nos quais há uma diversidade de categorias de usuários dos sistemas e aplicações. Entre essas pode-se citar a de usuários da própria organização, incluindo administradores, desenvolvedores e usuários comuns, a de usuários de organizações externas de desenvolvimento de software e prestação de serviço, a de usuários de organizações parceiras e a categoria de usuário de clientes. Esses usuários, seja de qual categoria forem, em princípio fazem acesso aos sistemas e aplicações tanto de dentro da organização, no caso dentro do perímetro de segurança, como de fora.

O objetivo deste trabalho é apresentar um estudo e aplicação da abordagem de confiança zero para segurança de sistemas computacionais a um ambiente computacional de *big-data* com acesso interno entre módulos e externo via web. Essa abordagem e a arquitetura associada serão apresentados ao longo deste texto. No caso deste trabalho, o estudo considera uma organização real com um parque computacional de porte médio, com uma organização orientada a *big data*, com extensa aplicação de virtualização de sistemas (máquinas virtuais, majoritariamente), com grandes volumes de dados (texto, imagens, vídeos, áudios), com bancos de dados em contínuo crescimento, com vários dados altamente sensíveis). As categorias de usuários e a origem dos acessos são condizentes com o que foi apresentado no parágrafo anterior. A referida organização, uma instituição pública do estado de Minas Gerais, usa os esquemas tradicionais de proteção, baseados em perímetro de segurança, *firewalls*, *gateways* de proteção, redes virtuais privadas (VPN), além dos mecanismos usuais de identificação e autenticação de usuários ou clientes. Dentro do ambiente computacional convivem sistemas e aplicações com objetivos e funções diversas. Existem aplicações que são acessadas via web e outras por meio de interfaces de acesso específicas. Há aplicações ou sistemas que são acessados apenas para quem está dentro da organização.

Este texto está organizado como a seguir, após esta seção de introdução: a Seção 2 apresenta a abordagem de confiança zero para segurança cibernética de ambientes computacionais. A Seção 3 apresenta o contexto computacional e de segurança. A Seção 4 mostra a arquitetura de segurança, bem como os casos de uso e as soluções desenvolvidas. A Seção 5 apresenta alguns trabalhos relacionados. A Seção 6 conclui o trabalho.

2. A abordagem confiança zero para segurança cibernética

O foco inicial da abordagem de confiança zero para a segurança cibernética deve ser a restrição dos recursos somente àqueles que necessitam acessá-los e a concessão apenas dos privilégios mínimos (por exemplo, ler, escrever, excluir) necessários para executar a missão[Haber 2020]. As definições operacionais de confiança zero e de arquitetura de confiança zero são as seguintes. A *confiança zero* (ZT - do inglês *Zero Trust*) provê uma coleção de conceitos e ideias projetados para reduzir a incerteza pela imposição de de-

cisões de acesso por requisição a sistemas e serviços de informação em face de uma rede vista como comprometida. Já a *arquitetura de confiança zero* (ZTA - do inglês *Zero Trust Architecture*) é um plano de segurança cibernética de uma organização, que utiliza conceitos de confiança zero, e que abrange relacionamentos de componentes, planejamento de fluxos de trabalho e políticas de acesso. Portanto, uma **organização confiança zero** consiste em uma infraestrutura de rede (física e virtual) e em políticas operacionais existentes colocadas em prática como produto de um plano de arquitetura de confiança.

Existe uma série de **princípios básicos** aos quais uma arquitetura confiança zero deve ser aderente, definidos pelo NIST e aqui apresentados sucintamente [S. Rose et al. 2020]: (1) todas as fontes de dados e serviços de computação são consideradas recursos; (2) toda a comunicação é protegida, independentemente da localização da rede de comunicação; (3) o acesso a recursos individuais da organização é concedido por sessão; (4) o acesso aos recursos é determinado por uma política dinâmica - incluindo o estado observável da identidade do cliente, do aplicativo e do ativo solicitado - e pode incluir outros atributos comportamentais; (5) a organização garante que todos os dispositivos proprietários e associados estejam no estado mais seguro possível, e monitora os ativos para garantir que eles permaneçam no estado mais seguro possível; (6) toda autenticação e autorização de recurso é dinâmica e rigorosamente aplicada antes do acesso ser permitido; (7) a instituição coleta o máximo possível de informações sobre o estado atual da infraestrutura e da comunicação da rede, e as utiliza para melhorar sua postura de segurança.

Adicionalmente, existem pressupostos sobre a infraestrutura da rede da organização: (8) a rede privada da organização não é considerada uma zona de confiança implícita; (9) dispositivos na rede podem não pertencer ou não ser configurados pela organização; (10) nenhum recurso de rede é inerentemente confiável. Existem também pressupostos para a infraestrutura de rede de propriedade de terceiros: (11) nem todos os recursos da organização estão localizados em infraestrutura da própria organização; (12) usuários remotos não podem confiar totalmente na conexão da rede local.

Diversos componentes lógicos compõem uma arquitetura de confiança zero em uma organização. Esses componentes podem ser operados como um serviço local ou por meio de um serviço baseado em nuvem. Na Figura 1, baseada no documento do NIST acima citado, tem-se um modelo (ideal) de estrutura conceitual que mostra o relacionamento básico entre os componentes lógicos e suas interações. O ponto de decisão de política (PDP) aqui é dividido em dois componentes lógicos: o mecanismo de políticas e o administrador de políticas, abaixo definidos. Os componentes lógicos da arquitetura de confiança zero usam um plano de controle separado para se comunicar, enquanto os dados das aplicações trafegam em um plano de dados. Ilustrando, um usuário ou máquina precisa fazer acesso a um recurso da organização; o acesso é concedido pelo Ponto de Decisão de Política (PDP) e habilitado pelo Ponto de Execução de Política correspondente (PEP). O esquema deve garantir que o usuário seja autêntico e a solicitação seja válida. O PDP julga se o sujeito pode acessar o recurso, o que implica que a confiança zero se aplica a duas áreas básicas: autenticação e autorização..

Os componentes da arquitetura de segurança zero estão aqui descritos [S. Rose et al. 2020] [Bethlehem 2020]:

1. Mecanismo de política (PE - *Policy Engine*) - concede (ou não) acesso a um re-

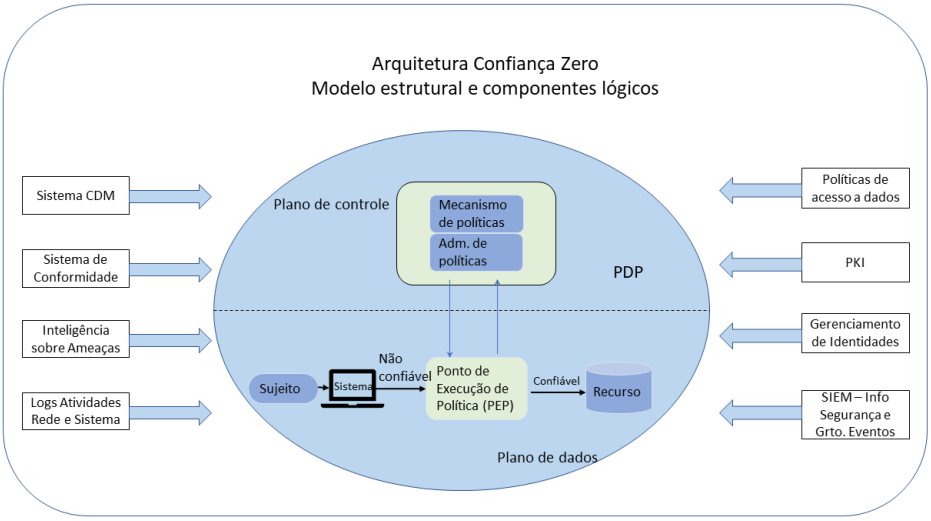

Figura 1. Modelo estrutural e componentes da arquitetura de confiança zero

curso para um determinado sujeito;

2. Administrador de políticas (PA - *Policy Administrator*) - estabelece ou bloqueia o caminho de comunicação entre um sujeito e um recurso;
3. Ponto de execução de política (PEP - *Policy Enforcement Point*) - habilita, monitora e encerra conexões entre um sujeito e um recurso da organização;
4. Sistema de diagnóstico e mitigação contínuos (CDM - *Continuous Diagnostics and Mitigation*) - reúne informações sobre o estado atual dos ativos da organização e aplica atualizações aos componentes de configuração e de software;
5. Sistema de conformidade do setor (*Industry compliance system*) - garante que a organização permaneça em conformidade com os regimes regulatórios relevantes;
6. Informações de inteligência sobre ameaças (*Threat intelligence feeds*) - fornece informações de fontes internas ou externas que ajudam o mecanismo de políticas a tomar decisões de acesso;
7. Políticas de acesso a dados (*Data access policies*) - são os atributos, regras e políticas sobre o acesso aos recursos da organização;
8. Infra-estrutura de chave pública da organização (PKI - *Public Key Infrastructure*) - gera e registra certificados emitidos pela organização;
9. Sistema de gerenciamento de identidades (ID *management system*): cria, armazena e gerencia contas de usuário corporativo e registros de identidade (por exemplo, servidor LDAP - *lightweight directory access protocol*);
10. *Logs* de atividades de rede e sistema (*Network and system activity* logs) - sistema corporativo que agrega *logs* de ativos, tráfego de rede, ações de acesso a recursos e outros eventos relativos à segurança;
11. Sistema de informações de segurança e gerenciamento de eventos (SIEM - *Security Information and Event Management*) - coleta informações de segurança para análise posterior.

3. Ambiente e arquitetura de estudo

O ambiente computacional objeto do estudo é aqui brevemente apresentado, com o intuito de mostrar a sua complexidade, organização e tecnologias utilizadas. Todas

as atividades do ambiente computacional são implementadas em um ambiente de nuvem privada. A infraestrutura computacional está organizada em um *cluster* de hiperconvergência, instalado dentro do *data center* da organização; há duas LANs virtuais separadas, uma para o *data center* como um todo e outra específica para o ambiente de *big data*, que é acessível pelos usuários internos. O acesso externo é provido por meio de redes virtuais (VPN) *site-to-site* e redes *Multiprotocol Label Switching* (MPLS). O acesso público a aplicações é feito via mecanismos de proteção de acesso (proxy reverso).

A arquitetura do sistema é baseada em camadas, sendo cada uma delas uma abstração de alto nível para um conjunto de etapas com ferramentas distintas. A Figura 2 ilustra essa arquitetura e as interações entre as camadas. A camada de Fonte de Dados realiza a coleta dos dados, os quais são enviados para a camada Zona Intermediária, onde aguardam para serem processados e incorporados ao ambiente de processamento *big data*, o *data lake* (lago de dados). Uma vez no *data lake*, os dados são limpos e refinados, seja por processos internos de transformação dos dados, seja por aplicações de análises e exploração de dados da camada Analytics. Ao final do fluxo, os dados são copiados para o Repositório de Dados, o repositório final, onde serão utilizados pelas Aplicações Web.

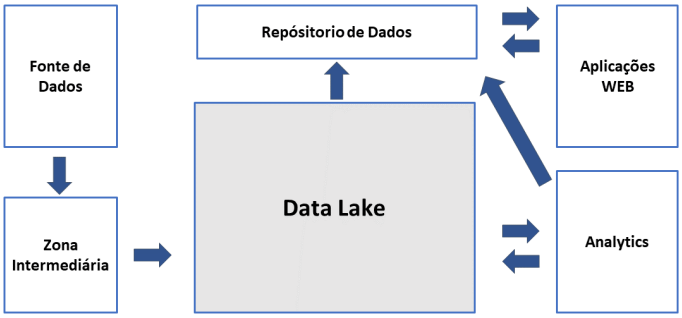

Figura 2. Arquitetura de dados/serviços

O modo atual de proteção e segurança não é satisfatório devido a especificidades de setores críticos quanto à sensibilidade dos dados e aplicações, bem como pela diversidade de tipos e origens de usuários do sistema. Nesse ambiente computacional não há uma política específica de segurança da informação (PSI); os controles de acesso a sistemas que precisam de autenticação são efetuados a partir de um banco de dados LDAP ou pelo registro de usuários no próprio sistema operacional onde um serviço está instalado. Mesmo para *ssh* e *login*, o sistema operacional tenta a autenticação via LDAP, primeiro, e só depois verifica se há usuários criados localmente.

O acesso externo a aplicações de acesso controlado ainda é baseado em listas de endereços IPs e portas específicas, de forma a limitar o local de acesso dos usuários de fora do ambiente computacional, visando criar uma ideia de um perímetro de segurança estendido. A proteção de acesso a dados e aplicações é garantida pelas ferramentas usuais de controle de proteção de ambientes em rede baseados no sistema operacional Linux.

No ambiente do *data lake*, a segurança dos dados é feita pelo Apache Ranger, uma ferramenta de segurança abrangente de dados [Ranger 2021]. Com ela, regras de acesso podem ser assinaladas de forma flexível a regiões dos dados (colunas, células, etc.) com base na identificação do usuário executando um determinado programa. Para as aplicações Web, o controle de acesso às informações é efetuado pela lógica da aplicação.

As aplicações acessam os dados com identificadores de usuário a elas associados. Algumas aplicações mantêm registros de usuários em bases de dados específicas, sendo que esses usuários têm acesso apenas à interface das aplicações e não diretamente aos sistemas da infraestrutura.

4. Confiança zero no ambiente alvo

Apresenta-se aqui uma proposta de instanciação da referida arquitetura de confiança zero específica para o referido ambiente na camada **Data Lake**, **Aplicações Web** e **Analytics**. Ressalta-se que este estudo de caso está em contínuo desenvolvimento.

4.1. Arquitetura do sistema

Ao se definir uma arquitetura a partir de um modelo estrutural genérico, o ambiente computacional alvo e tecnologias específicas devem ser levados em consideração para nortear a atividade. Para o caso específico, foram selecionadas, após análise detalhada, ferramentas de software para instanciar a arquitetura, conforme mostrado em seção abaixo. O mapeamento das ferramentas e fontes de dados e políticas existentes no modelo estrutural e componentes lógicos é mostrado na Figura 3. Os componentes em branco são considerados na atual fase da arquitetura, enquanto os componentes em azul estão em fase de instanciação e não serão abordados neste artigo. Como mostra a figura, o protótipo, no momento, tem um foco maior na autenticação dos usuários e na autorização dinâmica. Tal decisão é devida a serem estas funcionalidades de implementação mais direta e trazerem benefícios imediatos à segurança do ambiente. Está sendo construído um sistema de *data analytics* para análise de padrões de comportamento dos usuários.

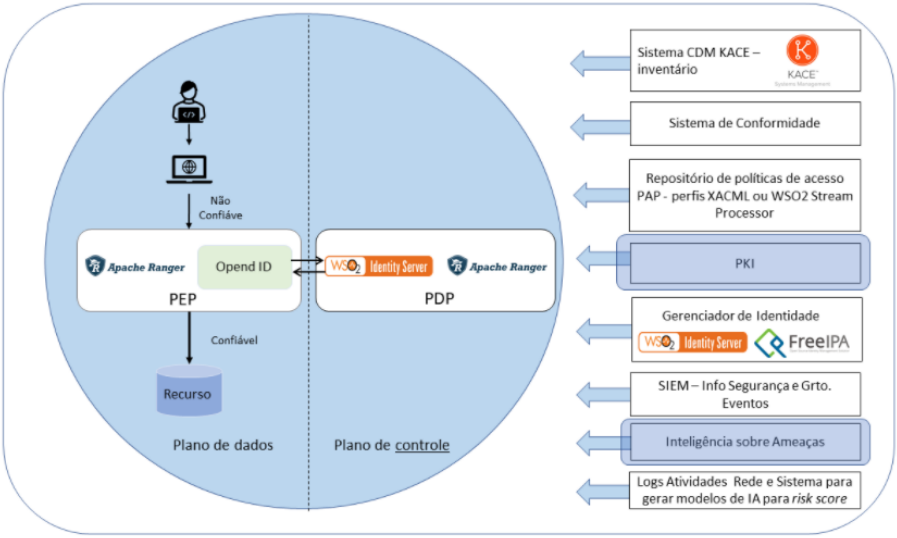

Figura 3. Mapeamento no modelo estrutural das tecnologias do protótipo.

4.2. Ferramentas habilitadoras e plataformas consideradas

As ferramentas de software empregadas no protótipo são brevemente descritas a seguir.

AIDE (*Advanced Intrusion Detection Environment*) [RedHat] – cria um banco de dados de arquivos do sistema e o usa para garantir a integridade dos arquivos e diretórios e detectar intrusões no sistema.

Apache Hadoop [Apache Hadoop 2021] – é um ecossistema de código aberto da *Apache Software Foundation* para *big data*. No ambiente, o elemento utilizado do Hadoop é o seu sistema de arquivos, o Hadoop *Distributed File System* (HDFS).

Apache Ranger [Ranger 2021] – é um /it framework para habilitar, monitorar e gerenciar de forma centralizada a segurança de dados de ecossistema Apache Hadoop. Algumas características do Ranger são: administração de segurança; controle de acesso de grão fino e padronizado para todo o ecossistema Hadoop; auditoria.

FreeIPA [FreeIPA 2021] – é uma solução integrada de identidade e autenticação para ambientes de rede Linux/UNIX. Um servidor FreeIPA fornece autenticação centralizada, autorização e informações de conta; armazena dados sobre usuários, grupos, hosts e outros objetos necessários para gerenciar a segurança de uma rede de computadores.

Microsegmentação [VMWare 2021] – divide um *data center* em segmentos de segurança distintos, indo até o nível de carga de trabalho individual. Os controles de segurança e os serviços são definidos para cada segmento.

OpenID Connect 1.0 [Connect 2017] – é uma camada de identidade no topo do protocolo OAuth 2.0. Ele permite que os clientes verifiquem a identidade do usuário final com base na autenticação realizada por um servidor de autorização, bem como obtenham informações básicas de perfil do usuário final de maneira interoperável e semelhante ao estilo de arquitetura de software REST.

OpenSCAP [SCAP 2022]– é uma biblioteca que usa SCAP (*Security Content Automation Protocol*) mantida pelo NIST (*National Institute of Standards and Technology*). Ela realiza varreduras de configuração e de vulnerabilidades em um sistema local, para validar o conteúdo de conformidade da configuração e para gerar relatórios e guias com base nessas varreduras e avaliações.

XACML (*eXtensible Access Control Markup Language*) [Hanna 2021] – é um sistema baseado em XML para controle de acesso. O XACML oferece uma maneira padronizada de obter autorização externalizada e dinâmica. Decisões de autorização são feitas por um serviço de autorização em tempo de execução com base em políticas que determinam quais ações um usuário ou serviço pode executar em um determinado ativo de informação e em um contexto específico.

WSO2 Identity Server, ou WSO2 IS [Karunarathna and Karunaratne 2017] – é um software baseado em padrões abertos e princípios de código aberto que tem a finalidade de simplificar as atividades relacionadas ao Gerenciamento de Identidades e Controle de Acesso (IAM). O WSO2 *Identity Server* possibilita o *single sign on* (SSO) entre aplicações, federação de identidades, autenticação forte adaptativa, administração de identidades, gerenciamento de contas, provisionamento de identidades, controle de acesso de grão fino, monitoramento, relatórios e auditoria.

4.3. Protótipos para o estudo de caso

Para o estudo de caso, foi concebido e implementado um protótipo aplicado a processos de negócios específicos, onde são abordados aspectos como identidade, autenticação simples e múltiplos fatores, *login* único, federação/autenticação federada, autorização e microsegmentação de rede. Os protótipos são orientados para aplicações web, para ambientes computacionais em nuvem e ambiente de *big data*.

A Figura 4 apresenta a estrutura e os componentes atualmente em uso no protótipo para aplicações web. As aplicações web localizam-se parte no ponto de execução de política - PEP e parte na representação dos recursos sendo acessados. Na parte relativa ao plano de dados encontra-se o Ponto de Execução de Políticas (PEP), que realiza solicitações de identificação (OpenID Connect e XACML) ao Ponto de Decisão de Políticas (PDP), situado no plano de controle. A solicitação OpenID Connect é a requisição de autenticação do usuário e a solicitação XACML é a requisição de autorização de uso de recursos.

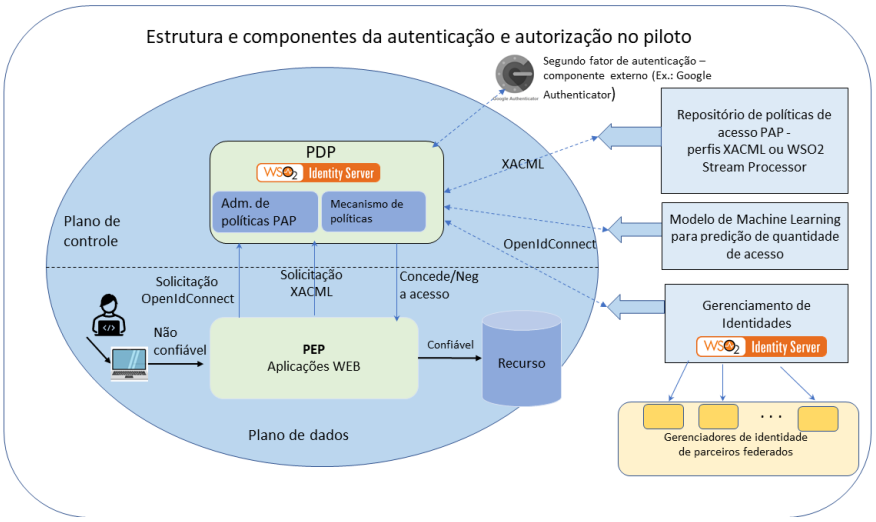

Figura 4. Estrutura e componentes de autenticação e autorização do protótipo.

No plano de controle tem-se o Ponto de Decisão de Política (PDP), que toma as decisões sobre as requisições feitas pelo ponto de execução de política (PEP) com base em repositórios de políticas. As decisões tomadas são tipicamente ”conceder”ou ”negar”acesso a recursos. O PDP também é o elemento que se comunica com agentes externos do WSO2 IS como, por exemplo, o Google Authenticator, e com o gerenciador de identidade de parceiros federados para a realização de autenticação. O diagrama também mostra as tecnologias ou protocolos usados na comunicação entre os diversos módulos da arquitetura. Ademais, indica que a autenticação federada acontece pela via do gerenciador de identidades, no caso o WSO2 Identity Server. Com o propósito de ilustração da aplicação da proposta de arquitetura, apresentam-se a seguir quatro exemplos na forma de caso de uso e a descrição da solução correspondente.

4.4. Caso de uso 1: *login* único, identidade federada e autenticação de múltiplos passos

Considere o caso em que um usuário faz acesso a um primeiro serviço ou aplicação e, em seguida, faz acesso a um segundo serviço. Esse usuário utiliza o mesmo navegador para acessar ambas as aplicações, não deve ter que fazer mais de um *login*, pode ser de origem interna ou pertencer a qualquer organização que utiliza o sistema. Ademais, conforme o contexto do momento e da operação a ser executada, suas credenciais podem ser alteradas e, consequentemente, a decisão de autorização ou negação tem que ser revista.

A solução concebida para esse caso, integrada na arquitetura de confiança zero, utiliza os recursos de gerenciamento de identidades da ferramenta WSO2 Identity Server. A primeira requisição do usuário vai para a Aplicação 1 (Requisição 1); em seguida, vai para um segundo serviço, no caso Aplicação 2 (Requisição 2). Primeiramente, ao acessar Aplicação 1, a requisição do usuário é imediatamente direcionada para o provedor de identidade, no caso o WSO2 Identity Server, para quem são fornecidas as credenciais para ser autenticado; o WSO2 IS salva em *cookies* do navegador as informações de *login* do usuário; em seguida o usuário é redirecionado para a Aplicação 1. Quando a Aplicação 2 for acessada, na Requisição 2, o usuário será redirecionado novamente para o provedor de identidade, que verifica se já existem *cookies* salvos no navegador referente ao usuário autenticado na Aplicação 1. Caso positivo, não solicita novamente as credenciais do usuário, direcionando-o então para a Aplicação 2.

O caso de autenticação de múltiplos passos exercita conceito de autenticação adaptativa, no qual um usuário é autenticado conforme a necessidade de acesso a recursos e em função do contexto. A solução concebida para esse caso funciona da seguinte forma: Um usuário, no Contexto 1, ao solicitar acesso ao recurso Perfil na aplicação web consegue acessar sem problemas, somente com autenticação básica já realizada. Já por outro lado, no Contexto 2, o usuário ao solicitar acesso ao recurso Alterar Senha, que pode ser considerado mais sensível, necessita ademais fornecer uma senha OTP do Google *Authenticator* para poder acessar o recurso.

Para o caso de identidade federada, consideremos um caso no qual uma Organização A e uma Organização B, têm provedores de identidade e são parceiras. A solução concebida para esse caso, integrada à arquitetura de confiança zero, funciona como a seguir. O usuário da Organização B acessa a aplicação da Organização A e informa ao provedor de identidade da Organização A que deseja se autenticar utilizando o provedor de identidade (idP) da sua organização de vínculo. O usuário fornece suas credenciais ao provedor de identidade da sua organização, o qual fica responsável pela autenticação do usuário e por informar ao idP da Organização A que o usuário foi autenticado, repassando informações do usuário autenticado.

O caso de uso apresenta a autenticação federada, autenticação adaptativa e autenticação de múltiplos fatores. Em comparação com os acessos tradicionais que solicitam login e senha, a adição das autenticações traz benefícios à segurança dos sistemas e dados. As autenticações adicionais tornam o processo de *login* dependente de duas ou mais informações do usuário, o que requer maior esforço de um *hacker* para coleta de informações de acesso ao sistema. O SSO (*Single Sign-On*), ao permitir que o usuário efetue *login* apenas uma vez para acessar mais de um serviço, traz mais segurança, pois estimula os usuários a utilizarem senhas mais fortes.

4.5. Caso de uso 2: aprendizado de máquina na autorização de acesso para fluxos

Uma das funções mais importantes na abordagem de confiança zero é a autorização de fluxos de comunicação entre usuários e aplicações e entre máquinas e processos, para atender ao princípio de que ninguém é confiável, nem mesmo processos em execução no ambiente interno. O resultado da função é ou autoriza ou nega uma solicitação de acesso. Tal função faz parte do mecanismo de política na arquitetura, parte integrante do plano de controle, que decide conceder (ou não) acesso a um recurso para um determinado sujeito. Associado ao mecanismo de política pode existir um algoritmo de atribuição de confiança a usuários, que pode ser tão sofisticado quanto se queira. A utilização de Inteligência Artificial no contexto de segurança cibernética possibilita rapidez na detecção e bloqueio de invasões e fornece proteção contra ameaças sofisticadas[Wirkuttis and Klein 2017].

No caso, está sendo desenvolvido um algoritmo baseado em aprendizado de máquina que considera o comportamento histórico dos usuários. Os dados considerados para o aprendizado e treinamento são obtidos a partir de *logs* de acesso a algumas aplicações do sistema, no caso também chamadas de aplicações finalísticas ou aplicações web, conforme mostrado na seção de apresentação do contexto. Esses *logs*, ou registros de operações de acesso às aplicações finalísticas, são utilizados para desenvolver modelos de aprendizado de máquina que, por sua vez, serão utilizados para gerar *scores* de confiabilidade de usuários, os quais norteiam as tomadas de decisão relativas às demandas de autenticação e de autorização de acesso. A geração de *score* de confiabilidade de usuário é feita por um algoritmo de atribuição de confiança, que faz parte do Mecanismo de Política (PE- *Police Engine*), um componente lógico da arquitetura de confiança zero.

Para elaboração do algoritmo de autorização ou negação de fluxos de comunicação entre máquinas virtuais e processos sendo executados no ambiente, são considerados os *logs* dos fluxos entre os módulos da parte interna do ambiente apresentado na Figura 3. A partir do estudo desses *logs*, um modelo de aprendizado de máquina é criado. Diferentemente do modelo anterior, que visa a segurança do ponto de vista do usuário final, este modelo trata da segurança dos processos internos do ambiente, delimitando a criticidade do processo à confiança do usuário que tenta acessar. A criticidade do processo é determinada pela frequência de acesso pelo usuário naquele determinado horário, seu *score* e a rede que origina o acesso. A atribuição de confiança a aplicações também faz parte da abordagem de confiança zero.

A estratégia utilizada para o caso de uso, inicialmente, foi desenvolver um algoritmo de aprendizado de máquina para a predição da quantidade de acessos, por usuário e por horário utilizando séries temporais. Foram utilizados também algoritmos de *cluster* de aprendizado de máquina para entender o comportamento não conhecido de acesso dos usuários, com objetivo de classificar o que é ou não um comportamento incomum, resultando em atribuição de *scores* aos usuários.

O algoritmo de aprendizado de máquina em desenvolvimento visa limitar automaticamente o acesso de usuários. Caso dados de um *login* estejam na posse de um *hacker*, o algoritmo detectará a discrepância de comportamento e efetuará o bloqueio do usuário automaticamente. O interessante desse algoritmo é a sua capacidade e autonomia para bloquear acessos de acordo com o contexto do usuário, sem a necessidade um gestor de sistemas, bem como a proteção sistemática durante as 24 horas do dia. Uma desvantagem

a ser considerada é a demanda de processamento dos dados e treinamento do algoritmo de aprendizado de máquina, visto que *logs* de acesso possuem alta densidade de dados.

4.6. Caso de uso 3: políticas de acesso no ambiente de *big data*

Para o estudo de caso do ambiente de *big data*, um modelo conceitual foi desenvolvido para um protótipo, no qual são abordados os aspectos de autorização relevantes.

A ferramenta mais utilizada neste caso é o Apache Ranger, um software que provê acesso a vários componentes no ecossistema Hadoop, havendo funções que se encaixam no plano de controle e outras no plano de dados. A decisão é tomada a partir de informações do repositório de políticas e de um *score* de confiança gerado dinamicamente por um modelo de aprendizado de máquina, elaborado anteriormente com base na análise de *logs* de acesso do Ranger.

Um protótipo foi concebido e aplicado a um processo de negócio específico, onde são abordados aspectos de autorização através do Apache Ranger. A proposta do protótipo foi demonstrar a possibilidade de criar políticas dinâmicas com linguagem de programação no Apache Ranger. Neste protótipo específico não foram utilizados modelos de aprendizagem de máquina para geração de *scores* de confiança, o que deverá ser feito no futuro, considerando também modelos estatísticos para a autorização. No contexto utilizando o Apache Ranger é possível incorporar regras dinâmicas, visto que ele conta com uma estrutura de extensibilidade para avaliar esse tipo de regra, denominado *Condition Evaluator* [Balaji Ganesan and Alok Lal 2015].

O caso de uso hipotético é o da criação e uso de uma política por um administrador no Apache Ranger, pela inserção de valores associados a riscos. Neste exemplo, valores maiores ou iguais a 5 são considerados de alto risco e o acesso ao recurso solicitado é negado; valores abaixo de 5 são considerados de baixo risco e o acesso ao recurso solicitado é permitido.

A autenticação baseada em contexto, ou fino grão, inibe ações suspeitas e os seus impactos ao ambiente computacional independentemente da politica de acesso estabelecida. Isso é possível, pois a partir do contexto de ambiente e usuário, o acesso pode ser bloqueado ou permitido. O controle de acesso encontrado com mais frequência é baseado no método *Role Based Access Control* (RBAC), o qual restringe o acesso à rede com base nas funções de usuários individuais de uma organização, ou seja, restringe por políticas. A autenticação baseada em contexto ao ser comparada com a autorização (RBAC), proporciona maior segurança ao sistema, pois o acesso torna-se dependente de um histórico de utilização, ao qual dificilmente um invasor tem acesso.

4.7. Caso de uso 4: nuvem computacional

Este caso de uso engloba máquinas virtuais e sistema operacional. Desenvolvemos um modelo conceitual em conformidade com as recomendações de confiança zero e com base em ferramentas de mercado. As ferramentas selecionadas, bem como seu uso no caso, são descritas a seguir.

As ferramentas selecionadas para o caso de uso foram o FreeIPA, como gerenciador de identidade para os sistemas operacionais do ambiente, e técnica de microsegmentação, que segue o princípio de privilégio mínimo. O OpenSCAP foi integrado ao sistema operacional para verificar as configurações e vulnerabilidades no

Tabela 1. Situação da segurança antes e depois da adoção da confiança zero

	Ambiente computacional	**Antes da abordagem ZT**	**Proposta da abordagem ZT**
1	Aplicações Web	Autenticação por usuário e senha	Autenticação Multifatores; SSO; Autenticação federada
2	Big Data	Autorização RBAC (*Role Based Access Control*	Autorização de fino grão baseada em contexto
3	Nuvem	*Demilitarized Zone* (DMZ)	Microsegmentação

sistema local, além de validar e gerar relatórios sobre o conteúdo de conformidade de configuração. O AIDE foi integrado à solução para funcionar como um SIEM(*Security Information and Event Management*), no qual as informações coletadas sobre intrusões do sistema para análises e tomada de decisões de segurança.

A microsegmentação de privilégio mínimo proposta, ao isolar os pontos críticos da rede, evita que o *malware* consiga acesso a ferramentas de gerenciamento e controle do ambiente computacional. Isso é importante pois impede a perda de controle do ambiente computacional. O método *Demilitarized Zone* (DMZ), tradicionalmente utilizado em nuvem, permite que um dispositivo da rede seja removido do *firewall* externo e seja exposto completamente à Internet, tornando o dispositivo vulnerável[Linksys 2022].

Os casos de uso baseados na abordagem de confiança zero apresentam benefícios para segurança de gerenciamento de sistemas e dados. A Tabela 1 apresenta a situação da segurança para os ambientes abordados nos casos de uso, antes e depois da adoção da proposta de arquitetura.

5. Trabalhos relacionados

O trabalho [Teerakanok et al. 2021] aborda a migração da arquitetura baseada em perímetro para a arquitetura de confiança zero, propondo um processo de migração em três etapas: avaliação dos usuários e ambiente computacional; avaliação de risco; e implantação e revisão. O nosso trabalho aborda as três etapas ao propor e instanciar uma arquitetura baseada em tecnologias comerciais aplicadas ao ambiente de *big data* real.

O trabalho [Tao, Yang and Lei, Zhu and Ruxiang, Peng 2018] propõe a incorporação de algoritmos ao Apache Ranger para a realização do controle de acesso de *big data* em três etapas: (i) reconhecimento de contexto do usuário com base em confiança zero; (ii) controle de grão fino da autenticação de acesso a dados; e (iii) auditoria de acesso a dados com base no tráfego de rede. Experimentos conduzidos demonstraram que a proposta pode identificar a maioria dos riscos de segurança de dados. No presente trabalho esses e outros aspectos de segurança são incorporados ao ambiente de *big data*, além de sugerir estruturas em forma de caso de uso, para autenticação e autorização utilizando tecnologias comercialmente disponíveis.

Os autores [Yao et al. 2020] apresentam um sistema dinâmico de controle de acesso e autorização a partir de retratos de comportamento dos usuários em tempo real para obtenção de dinamismo no controle de acesso e refinamento na autorização. Os autores não aplicam em um contexto real. Nosso trabalho apresenta um caso de uso

para controle de acesso e autorização de usuário utilizando algoritmos de aprendizado de máquina.

Embora os trabalhos descritos apresentem formas de migrar para uma arquitetura de confiança zero e formas de controle de acesso mais refinados, eles não apresentam casos de usos concretos. Nosso trabalho trata de casos de usos empregando software *open source*.

6. Conclusão

Este trabalho apresentou um estudo e aplicação da abordagem de confiança zero a um ambiente computacional de *big data*, abordando principalmente o controle de acesso a recursos internos e externos. No caso de acessos externos, foram propostas soluções para aplicações web. Para acessos internos, foram propostas soluções de políticas dinâmicas baseadas em *risk score* e microsegmentação da rede. A instanciação da arquitetura de confiança zero empregou diversas ferramentas, algumas delas já oferecendo mecanismos de segurança, o que facilitou grandemente essa complexa tarefa. Para trabalhos futuros serão utilizados algoritmos de aprendizagem de máquina para geração de *scores* de confiança, considerando também modelos estatísticos para a autorização, que serão incorporados ao Apache Ranger para realizar análise de requisição. Ademais, será estudada a efetividade das soluções propostas implementadas em um ambiente real, possivelmente de produção.

Referências

Apache Hadoop (2006-2021). The Apache™ Hadoop® project develops open-source software. Disponível em: https://hadoop.apache.org. (Acesso: 23.02.2022).

Balaji Ganesan and Alok Lal (2015). Dynamic Policy Hooks in Ranger - Configure and Use. Disponível em: https://cwiki.apache.org/confluence/display/RANGER/Dynamic+Policy+Hooks+in+Ranger+-+Configure+and+Use. (Acesso: 20.02.2022).

Bethlehem, D. (2020). The key components and functions in a zero trust architecture. https://cpl.thalesgroup.com/blog/encryption/key-components-function-in-zero-trust-architecture . (Acesso: 19.04.2022).

Connect, O. (2017). Openid connect 1.0. Disponível em: https://openid.net/connect/. (Acesso: 20.02.2022).

FreeIPA (2021). What is freeipa? Disponível em: https://www.freeipa.org/page/About#What_is_FreeIPA.3F. (Acesso: 21.02.2022).

Haber, M. J. (2020). Zero trust. In *Privileged Attack Vectors*, pages 295–304. Springer.

Hanna, K. T. (2021). Xacml (extensible access control markup language). Disponível em: https://www.techtarget.com/searchcio/definition/XACML. (Acesso: 15.02.2022).

Karunarathna, I. and Karunaratne, I. (2017). O que é o wso2 identity server? https://wso2.com/library/articles/2017/08/o-que-e-o-wso2-identity-server. (Acesso: 19.02.2022).

Linksys (2022). Ativando o recurso de dmz em sua conta na nuvem da linksys. https://www.linksys.com/br/support-article?articleNum=142514. (Acesso: 19.04.2022).

Ranger, A. (2021). Ranger. Disponível em: https:https://ranger.apache.org/. (Acesso: 20.02.2022).

RedHat. Checking integrity with aide. Disponível em: https://access.redhat.com/documentation/en–us/red_hat_enterprise_linux/7/html/security_guide/sec–using-aide. (Acesso: 20.02.2022).

S. Rose et al. (2020). *MDraft (2nd) NIST Special Publication 800-207 Zero Trust Architecture*. https://doi.org/10.6028/NIST.SP.800-207-draft2.

SCAP, O. (2022). Scan your system. https://www.open-scap.org . (acesso: 19.04.2022).

Tao, Yang and Lei, Zhu and Ruxiang, Peng (2018). Fine-grained big data security method based on zero trust model. In *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 1040–1045.

Teerakanok, S., Uehara, T., and Inomata, A. (2021). Migrating to zero trust architecture: Reviews and challenges. *Security and Communication Networks*, 1.

VMWare (2021). What is micro–segmentation? Disponível em: https://www.vmware.com/br/topics/glossary/content/microsegmentation.html/. (Acesso: 15.02.2022).

Wirkuttis, N. and Klein, H. (2017). Artificial intelligence in cybersecurity. *Cyber, Intelligence, and Security*, 1(1):103–119.

Yao, Q., Wang, Q., Zhang, X., and Fei, J. (2020). Dynamic access control and authorization system based on zero-trust architecture. In *2020 International Conference on Control, Robotics and Intelligent System*, pages 123–127.