PIPA: Uma solução integradora de políticas de controle de acesso a recursos e de gerenciamento de identidades. *

Júnia Maísa Oliveira, Emanuela Ferraz, Vinícius Rodrigues Oliveira, Daniel Fernandes Macedo, José Marcos Nogueira

[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, jmarcos}@dcc.ufmg.br

emanuellaferraz@ufmg.br

Resumo. *Este trabalho apresenta uma ferramenta integradora sistemas de controle de acesso a recursos e gerenciamento de identidades denominada PIPA - Plataforma Integrada de Políticas de Acesso - para um ambiente computacional composto de uma variedade de sistemas e aplicações com restrições de acesso. Cada sistema em um ambiente computacional complexo, como o big data, possui uma forma de associar políticas de acesso aos usuários, o que pode ser dispendioso para o administrador de sistema; no caso, o administrador precisa acessar cada um dos sistemas para atribuir políticas para cada usuário, uma tarefa trabalhosa e sujeita a erros quando a quantidade de aplicações e usuários é grande. O PIPA é uma solução para o problema de gerenciamento de políticas de acesso em ambientes computacionais que possuem diversos sistemas e serviços, como máquinas virtuais, sistemas operacionais, páginas web, nuvem, entre outros. Com o PIPA, o administrador do sistema realiza o gerenciamento de políticas de forma integrada, com uma única ferramenta. A ferramenta, já desenvolvida em sua primeira versão, encontra-se em teste e já mostra que pode ser efetiva em um ambiente de produção.*

1. Introdução

Este trabalho apresenta a Plataforma Integrada de Políticas de Acesso (PIPA), que tem como objetivo a integração do gerenciamento de identidades de usuários e a autorização de acesso a recursos de um ambiente computacional *de big data* composto de serviços de diferentes naturezas (plataformas de processamento de grandes volumes de dados, plataformas de armazenamento de código, aplicações de *big data* em produção, etc). A plataforma visa facilitar o cadastramento de usuários e a atribuição de políticas de acesso aos diversos sistemas de um ambiente computacional, como o apresentado em [Oliveira et al. 2022].

A diversidade de sistemas e aplicações que podem ser integradas a um ambiente computacional pode sujeitá-lo a vulnerabilidades diversas. Isso pode ocorrer quando cada sistema ou aplicação possui mecanismos de autorização e autenticação próprios. Quando isso acontece em ambientes computacionais complexos, os usuários muitas vezes utilizam

*Os autores agradecem as seguintes instituições brasileiras pelo apoio: MPMG - Ministério Público de Minas Gerais, CAPES, CNPq, FAPEMIG e FAPESP.

senhas fracas, o administrador do ambiente tem dificuldades de gerenciar as políticas de acesso aos sistemas, senhas passam a ser anotadas em papel ou compartilhadas em arquivos no computador, entre outras atitudes que afetam a segurança do ambiente computacional. Além do mais, tem-se as vulnerabilidades causadas pelo gerenciamento de políticas descentralizadas em sistemas pertencentes a ambientes computacionais complexos. Um outro fator importante é a alta carga de trabalho, no qual um administrador de sistema, ao cadastrar um novo usuário, gastará um tempo proporcional ao número de sistemas existentes no ambiente computacional, pois precisará acessar cada sistema individualmente e atribuir as políticas individualmente.

O problema tratado no trabalho é: *como construir uma solução de gerenciamento de identidades e políticas de acesso para um ambiente computacional com uma diversidade de sistemas, cada um com suas peculiaridades relativas ao controle de acesso a seus recursos, com um número elevado de usuários internos e externos, de forma a possibilitar uma operação cotidiana eficiente e segura?* Entre os desafios está a descrição e mapeamento de políticas de acesso unificadas para os diversos sistemas com suas especifidades. A implementação de interfaces entre módulos também configura dificuldades a serem vencidas.

O trabalho está organizado da seguinte forma: a Seção 2 apresenta os antecedentes do trabalho, a Seção 3 apresenta a ferramenta, a Seção 4 apresenta os requisitos computacionais necessários para o funcionamento da ferramenta, a Seção 5 apresenta os links de acesso aos manuais e repositório da ferramenta, a Seção 6 conclui o texto.

2. O ambiente alvo do estudo

O ambiente computacional objeto de estudo que motivou o desenvolvimento da plataforma é um ecossistema de big data, com aplicações web, máquinas virtuais (VM), computadores com sistemas operacionais Linux ou Windows, bem como mecanismos distintos de autenticação e autorização de acesso. O ambiente computacional objeto de estudo é apresentado em [Oliveira et al. 2022]. Atualmente o ambiente conta com aplicações de processamento de dados de *big data*, que seguem o modelo de autenticação Linux; um ambiente de ciência de dados baseado em *notebooks* e um sistema de controle de versionamento de código, seguindo um sistema de autenticação baseado em Active Directory; e finalmente versões de desenvolvimento e de produção de soluções de *big data* que empregam protocolos de autenticação próprios para aplicações Web. Cada um desses sistemas possui mecanismos próprios para a definição de políticas de autorização, que empregam por sua vez diferentes representações das políticas de acesso.

O gerenciamento de identidades de usuários dos sistemas do ambiente computacional alvo é feito por um ou mais sistemas que gerenciam as identidades de usuários quanto a autenticação e autorização de acesso a recursos. Os acessos aos sistemas diferenciam-se de acordo com a seus tipos que, atualmente, no caso do ambiente de estudo, são: web, ambiente de big data e máquinas virtuais.

A tarefa mais comum do administrador de sistemas, nesses ambientes, é a atribuição de políticas já existentes a um usuário ou a um grupo de usuários. Isto ocorre pois as políticas de acesso em já são em geral definidas no momento de implantação do sistema. São definidos perfis de acesso para cada sistema, aos quais os usuários devem ser mapeados. O ambiente de *big data* permite tanto o acesso de usuários locais (pertencen-

tes à organização) quanto o acesso de usuários de organizações parceiras, de forma que a quantidade de usuários ativos pode chegar a milhares. Assim, a criação e a remoção de contas são as ações mais frequentes no dia-a-dia do administrador.

A criação de novos usuários nesse ambiente computacional ocorre tipicamente da seguinte forma. Os usuários que necessitam fazer acesso a alguma VM do ambiente o solicitam via o preenchimento de um formulário online. O administrador de sistemas verifica os dados do novo usuário no formulário e, em seguida, cadastra o usuário e sua senha no sistema utilizado para o gerenciamento de informações de segurança para ambientes Linux/UNIX e, no caso, a ferramenta FreeIPA, um provedor de identidades (IdP) [FreeIPA 2021]. O administrador do sistema associa o usuário recém criado a um ou mais grupos de políticas padrão ou por *default*. Entretanto, com apenas esta associação inicial, o usuário não consegue executar ações no ambiente computacional.

O FreeIPA aplica o conceito de política de acesso, pela qual se define o que o usuário pode ou não fazer em um determinado recurso (Ex.: ler/escrever). Grupos de usuários do FreeIPA podem ser associados a políticas de acesso. Um usuário i associado a um grupo j FreeIPA está sujeito à política de acesso à qual o grupo j está associado. No FreeIPA não é possível dar permissão a grupos externos, como é o caso de usuários de um Active Directory (AD). Neste caso, os usuários do AD deverão ser adicionados a grupos de políticas do FreeIPA para que os usuários recebam políticas.

Atualmente a autorização de acesso no ambiente computacional estudado é realizada de maneira distinta para (1) as aplicações web, (2) o ecossistema de big data e (3) os sistemas operacionais Windows e Linux. O administrador acessa cada um dos mecanismo de gerenciamento de políticas para atribuir as políticas de acesso necessárias ao novo usuário do ambiente computacional.

3. Apresentação da ferramenta

O PIPA é uma plataforma computacional para gerenciamento de identidades e políticas de acesso a recursos computacionais por usuários e sistemas. Como indicado anteriormente, uma tarefa muito comum em ambientes de *big data* com muitos usuários é a atribuição de políticas a usuários. O PIPA possibilita de maneira eficiente a seu operador a definição e atribuição de políticas de acesso a usuários aos diversos sistemas (aplicações) do ambiente computacional. A sua arquitetura, componentes da ferramenta, funcionalidades, desenvolvimento e modo de uso são apresentados a seguir. Os principais elementos conceituais ou atores relacionados ao PIPA são os seguintes:

- **Usuários.** Classificam-se em usuários comuns, usuário administrador (ou só administrador), grupo de usuários, usuário-sistema.
- **Grupo de usuários.** conjunto de usuários com características comuns de origem ou função.
- **Sistemas ou aplicações do ecossistema de big data.** São as aplicações de processamento do ambiente de big data, tais como GitHub, Ranger, FreeIPA e Jenkins.
- **Política de acesso.** Denota as operações ou ações permitidas e/ou proibidas a um usuário para um determinado sistema ou aplicação. Ex: 1) ler, 2) escrever, 3) ler e escrever. (ex: ler, escrever, ler, apagar, copiar).
- **Projeto.** Um tipo de grupo de usuários para sistemas específicos. Conceito utilizado pelo GitLab, por exemplo.

O usuário principal do PIPA é o administrador de sistemas. Os clientes, aqueles que serão associados a políticas de acesso e a aplicações, são chamados de usuários comuns ou simplesmente usuários. O administrador de sistemas interage com o PIPA para gerenciar autorizações e identidades de usuários, às quais devem estar nele cadastrados.

O uso do conceito de grupos e políticas no PIPA decorre da visão centrada nas aplicações do ambiente computacional. Os grupos são em geral construídos de forma a identificar quais regras se aplicam a um certo usuário em um certo sistema. Grosso modo, cada usuário pode pertencer a um ou mais grupos, que servem para mapear conjuntos de usuários a políticas específicas. Os grupos, por sua vez, podem estar associados a um conjunto de políticas, que em geral são específicas de uma aplicação. Ver a Figura 1.

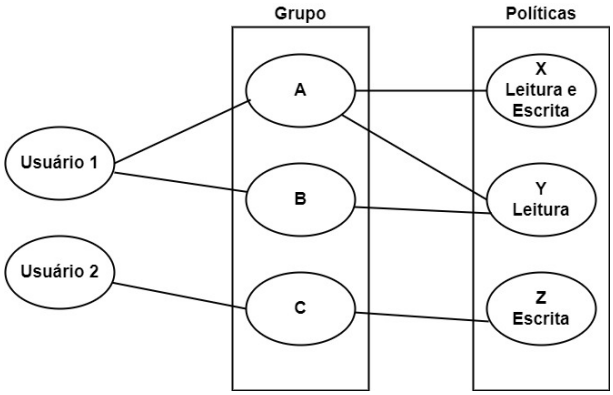

Figura 1. Relação lógica (associação) entre usuários, grupos e políticas no PIPA.

3.1. Estrutura organizacional

O PIPA foi concebido para operar em um ambiente distribuído, composto por diversos sistemas e interagir principalmente com o provedor de identidade FreeIPA, que armazena as credenciais dos usuários no seu LDAP o qual é utilizado pelas demais aplicações ou serviços no ambiente computacional. Ele interage com os diversos sistemas do ambiente para a criação de políticas de autorização e para a associação de usuários e grupos às políticas de autorização.

A Figura 2 apresenta a organização estrutural do PIPA, considerando a integração de algumas aplicações. Ele comunica-se diretamente com as aplicações ou sistemas FreeIPA, GitLab, Jenkins e Apache Ranger. O gerenciador de identidades WSO2 no momento está sendo considerado para ser integrado ao PIPA. A versão que será demonstrada ainda não inclui nem o Apache Jenkins nem o Apache Ranger, que no momento estão em fase de integração.

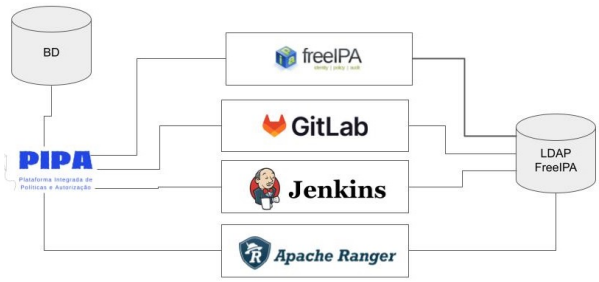

Figura 2. Organização estrutural do PIPA.

O PIPA conta um banco de dados, um provedor de identidades, uma interface de usuário e aplicações integradas. Esses componentes estruturais da plataforma estão a seguir descritos:

- **Banco de dados de usuários e políticas.** Armazena políticas e dados de usuários em processo de cadastramento (usuários pendentes). Um usuário, uma vez cadastrado, tem seus dados removidos deste banco de dados.
- **Provedor de identidades (IdP)** O FreeIPA fornece mecanismos para o PIPA criar/cadastrar usuários no banco de dados. Armazena dados de usuários, grupos, hosts e políticas gerenciados pelo FreeIPA.
- **Banco de dados do provedor de identidades** Associado ao FreeIPA, armazena dados de usuários, grupos, hosts e políticas gerenciados.
- **Interface dos usuários do ambiente.** A operação do sistema é feita por meio de uma interface humano-computador baseada em web.
- **Aplicações do ambiente computacional.** São aplicações ou sistemas que requerem controle de acesso para seu uso. O controle de acesso é definido por políticas específicas de cada sistema. Essas políticas de autorização devem ser instaladas nos sistemas.

3.2. Componentes integrados

As aplicações objeto de integração ao PIPA, numa primeira fase, estão a seguir brevemente descritas. Com a evolução do ambiente computacional e a consequente agregação de mais aplicações, plataformas, etc, o processo de integração continua.

- **FreeIPA.** O FreeIPA é uma solução integrada de gerenciamento de informações de segurança para ambientes Linux/UNIX. Consiste em uma interface web e ferramentas de administração baseadas em linhas de comando. Um servidor FreeIPA fornece autenticação centralizada, autorização e informações de conta, armazenando dados sobre usuários, grupos, hosts e outros objetos necessários para gerenciar os aspectos de segurança de uma rede de computadores [FreeIPA 2021].
- **Plataforma GitLab.**[1] GitLab é uma plataforma de desenvolvimento de software de código aberto considerado no ambiente de estudo e teste. Ele pode ser hospedado em servidores próprios, em um contêiner ou em um provedor de nuvem. O GitLab atribui suas políticas de acesso por projetos. Um projeto pode ser considerado como um agrupamento de usuários para efeito de controle de acesso.
- **Jenkins.**[2] O Jenkins é outro sistema considerado no ambiente computacional de estudo e teste. O Jenkins possui particularidades de permissão de acesso e gerenciamento de identidades. O gerenciamento de identidade é feito nas configurações de políticas próprias do Jenkins. O Jenkins automatiza o processo de implantação das aplicações desenvolvidas no GitLab.
- **Apache Ranger.**[3] O Apache Ranger gerencia autorizações de acesso às aplicações do ambiente big data. Políticas são atribuídas aos usuários para cada aplicação que utiliza o Apache Ranger como mecanismo de autorização. É possível atribuir grupos aos usuários e atribuir políticas aos grupos. O usuário é criado pelo administrador e as políticas são atribuídas ao usuário de acordo com cada aplicação que ele necessita acessar.

[1] https://gitlab.com/gitlab-org/gitlab
[2] https://www.jenkins.io
[3] https://ranger.apache.org/

- **WSO2 IS.**[4] é um software baseado em padrões abertos e princípios de código aberto com a finalidade de simplificar as atividades relacionadas ao Gerenciamento de Identidades e Controle de Acesso (IAM).O PIPA foi concebido para operar em um ambiente distribuído, composto por diversos sistemas e interagir principalmente com o provedor de identidade FreeIPA, que armazena as credenciais dos usuários.

3.3. Funcionalidades

A função precípua do PIPA é o gerenciamento de políticas de acesso a recursos computacionais por usuários e sistemas por meio da integração de sistemas. Para isso, possibilita a seu operador a definição e atribuição de políticas de acesso a usuários aos diversos sistemas (aplicações) do ambiente computacional estudado. As funcionalidades do PIPA, do ponto de vista do usuário, são as seguintes:

- **Solicitação de acesso ao ambiente computacional.** Os usuários comuns solicitam a criação de um login de acesso ao ambiente computacional.
- **Visualização de lista de usuários que solicitaram acesso ao ambiente computacional.** O usuário administrador visualiza a lista dos usuários comuns que solicitaram login de acesso ao ambiente computacional.
- **Visualização de dados de usuários.** O usuário administrador visualiza as informações de um usuário comum específico.
- **Atribuição de usuário a grupo.** O usuário administrador atribui um usuário a um grupo existente, o qual deve conter politicas de acesso vinculado. Ao ser incluído em um grupo, o usuário estará sujeito às políticas de acesso do grupo.
- **Exclusão de usuário de um grupo.** O usuário administrador pode excluir um ou mais usuários comuns pertencentes a um grupo especificado. Quando excluído, um usuário comum não fica sujeito a políticas de acesso do grupo e, com isso, não consegue acessar os sistemas relacionados do ambiente computacional .
- **Exclusão de usuário.** O usuário administrador exclui do sistema um usuário selecionado.

Todas as operações realizadas no PIPA são registradas inicialmente em seu banco de dados. Em seguida, são registradas no banco de dados do FreeIPA ou no sistema de armazenamento políticas próprias do GitLab. A Figura 3 ilustra a tela principal do PIPA, a qual dá acesso ao administrador de sistemas aos principais fluxos de tarefas possíveis na plataforma, tal como descrito acima.

3.4. Desenvolvimento

Do ponto de vista operacional, o PIPA é estruturado em duas partes. A primeira parte trata da interação com usuários, seja o administrador e ou sejam os usuários que solicitam cadastramento no PIPA. A segunda parte consiste do código para execução das operações solicitadas e sua interação com as diversas aplicações ou sistemas. A primeira parte, tratada aqui com frontend, foi desenvolvida utilizando a biblioteca React.js, uma biblioteca JavaScript para desenvolvimento de interfaces de usuário. A segunda parte, tratada aqui como backend, foi desenvolvida utilizando Flask[5], um micro-framework ou

[4]https://is.docs.wso2.com/en/5.9.0/

[5]https://flask.palletsprojects.com/en/2.2.x/

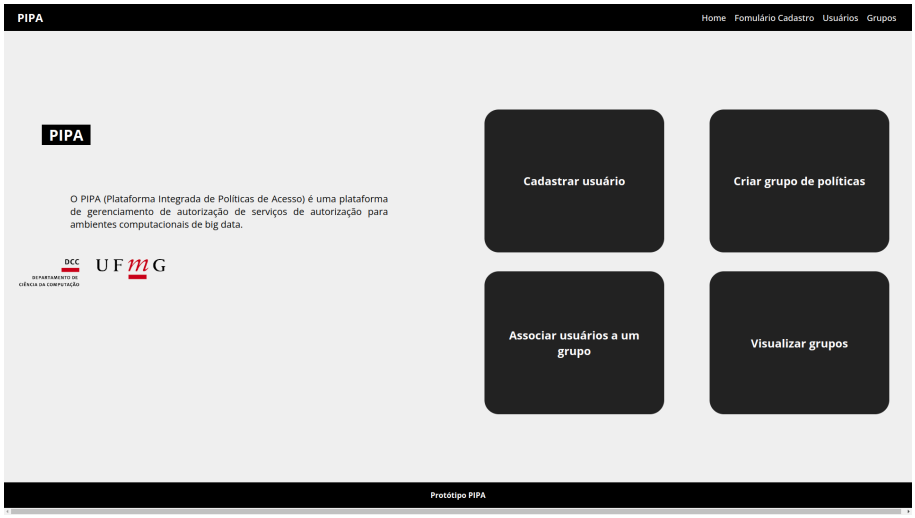

Figura 3. Tela home do PIPA que dá acesso às suas principais funcionalidades.

conjunto de ferramentas do Python usadas para desenvolvimento web. O PIPA tem um repositório de dados baseado no PostgreSQL[6], um sistema de gerenciamento de banco de dados objeto-relacional de código aberto que constitui o repositório de dados de usuários e grupos de políticas.

A primeira versão do PIPA conta com uma interface humano-computador gráfica voltada para a utilização do administrador da plataforma, a qual possibilita a execução das suas principais tarefas. Pela interface gráfica, o administrador preenche um formulário para criação de um usuário; preenche um formulário para a criação de um grupo de políticas, selecionando os projetos do GitLab e o grupo do FreeIPA aos quais os usuários deste grupo terão acesso; obtém uma lista de usuários com seus respectivos dados, tendo a possibilidade de editar seus dados, apagá-los e os criar no FreeIPA e no GitLab; lista os grupos do PIPA, também com acesso aos dados de cada um e possibilidade de apagá-los; seleciona usuários para os associar a grupos do PIPA.

O backend do PIPA é uma API REST que possibilita a outras aplicações utilizarem as funcionalidades do PIPA sem conhecer os detalhes de sua implementação. Sendo assim, ele pode ser acoplado a aplicações ou serviços já existentes sem muito esforço. Cada endpoint da API do PIPA tem um nome, uma descrição e métodos suportados. Os endpoints são apresentados conforme o manual de uso disponível a partir da Seção 5.

Algumas dificuldades foram encontradas no desenvolvimento do PIPA. Como exemplo, a compreensão e obtenção de informações sobre os diferentes sistemas integrados ao PIPA. Existem fatores limitantes em relação às chamadas de APIs que são disponibilizadas pelos sistemas integrantes do PIPA que não atendem a todas as funcionalidades que são possíveis executar via da interface web dos sistemas. Além do mais, a documentação das APIs não tem boas descrições sobre os métodos disponibilizados, o que dificultou o entendimento.

3.5. Modo de uso

Para utilizar o PIPA, o usuário deverá ter nome de usuário e senha válidos. Ao efetuar login, o usuário é autenticado pelo WSO2. A autorização é feita internamente na

[6]https://www.postgresql.org

aplicação, aplicando regras preestabelecidas que são baseadas no método RBAC (Role-Based Access Control). Sendo assim, o acesso é concedido ou negado de acordo com o papel (role) do usuário. Para um ambiente de teste, três máquinas virtuais foram configuradas, cada uma delas com as seguintes aplicações: (i) FreeIPA Server; (ii) GitLab; e (iii) Plataforma PIPA. Os sistemas Jenkins e Ranger estão sendo integrados no momento. Outros mais poderão ser integrados, conforme sejam agregados ao ambiente computacional.

4. Requisitos para a demonstração

Um computador apto a executar o PIPA com acesso à Internet e os seguintes programas instalados: Python 3, PostgreSQL e Node.js. As variáveis de ambiente devem ser criadas de acordo com os sistemas que se deseja integrar ao PIPA.

5. Links para acesso ao material: códigos, manuais, vídeos

- **URL do código fonte, documentação e manual da ferramenta:** https://github.com/WinetLabUFMG/PIPA
- **URL de vídeo sobre a instalação e as funcionalidades:** https://youtu.be/Vx9CEWh_mxM

6. Conclusão

A ferramenta desenvolvida, denominada de Plataforma Integrada de Políticas de Acesso (PIPA), atua em ambientes computacionais complexos, com grande número de usuários de origens e funções diversas e uma variada gama de aplicações com políticas de acesso específicas. O PIPA gerencia as identidades e políticas de acesso a recursos. A ferramenta integra os diversos gerenciadores de políticas de acesso de sistemas de um ambiente computacional complexo.

Uma primeira versão foi desenvolvida como prova de conceito. As ações executadas pelo sistema são armazenadas no software FreeIPA e as políticas são atribuídas aos diversos sistemas corretamente. Durante o desenvolvimento do PIPA foi necessário compreender as diferenças entre os atributos de cada API para modelar os *endpoints* do *backend*. Como trabalho futuro está o atendimento de requisitos de segurança necessários na comunicação entre as APIs, tarefa já em desenvolvimento. A integração de novas aplicações está no horizonte, havendo duas integrações em andamento.

Referências

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

Oliveira, J. M., Oliveira, V. R., Macedo, D. F., Guedes, D., and Nogueira, J. M. (2022). Abordagem confiança zero aplicada a ambientes computacionais big data: um estudo de caso. In *Anais do XXVII Workshop de Gerência e Operação de Redes e Serviços*, pages 127–140. SBC.