



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
DEPARTAMENTO DE COMPUTAÇÃO



TRABALHO DE CONCLUSÃO DE CURSO

**Analizador de políticas de controle de acesso para
plataformas de computação em nuvem**

Autor: Túlio Giovani Ferreira Bittar

Orientadora: Sílvia Calmon de Albuquerque

16 de Dezembro de 2022, Belo Horizonte



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
DEPARTAMENTO DE COMPUTAÇÃO



Túlio Giovani Ferreira Bittar

**Analizador de políticas de controle de acesso para
plataformas de computação em nuvem**

Trabalho de Conclusão de Curso apresentado ao
Curso de Engenharia de Computação do Centro
Federal de Educação Tecnológica de Minas Gerais,
como requisito parcial para a obtenção do título de
Bacharel em Engenharia de Computação.
Orientadora: Sílvia Calmon de Albuquerque

16 de Dezembro de 2022, Belo Horizonte

Avaliação do Trabalho de Conclusão de Curso

Aluno: Túlio Giovani Ferreira Bittar

Título do trabalho: Analisador de políticas de controle de acesso para plataformas de computação em nuvem

Data da defesa: 01/12/2022

Horário: 10:30

Local da defesa: Online, via Conferência Web RNP

O presente trabalho foi avaliado pela seguinte banca:

Professora Silvia Calmon de Albuquerque - Orientadora
Departamento de Computação
Centro Federal de Educação Tecnológica de Minas Gerais

Professor Adelson de Paula Silva - Membro da banca de avaliação
Departamento de Computação
Centro Federal de Educação Tecnológica de Minas Gerais

Professor Mateus Felipe Tymburibá Ferreira - Membro da banca de avaliação
Departamento de Computação
Centro Federal de Educação Tecnológica de Minas Gerais

16 de Dezembro de 2022, Belo Horizonte

Resumo

A popularização da computação em nuvem e sua grande adesão por empresas ao redor do mundo traz consigo o risco de vazamentos de dados e impactos de infraestrutura, passíveis de prejuízos milionários ou extorsões por dinheiro. Nas plataformas de computação em nuvem, os clientes são responsáveis pela gestão da identidade e acesso, o que pode se tornar um desafio, por sua complexidade, granularidade e por demandar conhecimento por parte dos clientes. Sendo assim, este trabalho apresenta uma ferramenta que visa facilitar a gestão de identidade e acessos na plataforma AWS, líder mundial em computação em nuvem, além de promover o conceito de menor privilégio e boas práticas de segurança junto aos clientes. A solução consiste no desenvolvimento de um sistema capaz de analisar e identificar permissões consideradas críticas, não autorizadas e de alto risco, em políticas anexadas a usuários. Ela se diferencia de trabalhos relacionados na sua habilidade de detectar mais de 500 dessas ações, entre as mais de 13 mil ações existentes. A metodologia adotada para conduzir este trabalho envolveu a criação de uma base de dados, com todos os tipos de permissões existentes na plataforma AWS, o desenvolvimento do analisador de políticas propriamente dito, o armazenamento das análises em um diretório, no qual o cliente possa pesquisar todas as análises executadas pela ferramenta e, por fim, a sua publicação em um repositório de código aberto. Os testes mostraram-se precisos, ao gerar resultados de fácil interpretação, além de provar a capacidade de tratar erros de formatação das políticas. O trabalho contribui para que o profissional de Segurança da Informação seja capaz de identificar um acesso crítico, antes de concedê-lo para um usuário ou serviço, e esteja apto a definir medidas cabíveis em favor da segurança de seu ambiente. Adicionalmente, a ferramenta pode ser utilizada para avaliar o risco da implantação de *softwares* comerciais, tais como *Commvault* e *Palo Alto Prisma Cloud*, no ambiente de computação em nuvem dos clientes.

Palavras-chave: Segurança. Computação em nuvem. Gestão de Identidades. Vazamento de Dados. Menor Privilégio.

Abstract

The popularization of cloud computing and its wide adoption by companies around the world brings with it the risk of data leaks and impacts in infrastructure, which can cause millionaire losses or extortion for money. In cloud computing platforms, customers are responsible for identity and access management, whose responsibility often becomes a challenge, due to its complexity, granularity and demand for knowledge from customers. Therefore, this project presents a tool that aims to facilitate identity and access management on the AWS platform, a world leader in cloud computing, in addition to promoting the concept of least privilege and good security practices to customers. The solution consists of developing a system capable of analyzing and identifying permissions considered critical, unauthorized and high risk in policies attached to users. It differs from related works in its ability to detect more than 500 of these actions, among more than 13,000 existing actions. The methodology adopted to execute this project involved the creation of a database, with all types of permissions existing in the AWS platform, the development of the policy analyzer itself, the storage of the analyzes in a directory, where the client can search all the analyzes performed by the tool and, finally, its publication in an open source repository. The tests proved to be accurate, delivering results that were easy to interpret and proving the ability to handle policy formatting errors. The work contributes to the Information Security professional being able to identify a critical access before granting it to a user or service, hence being able to perform the appropriate measures in favor of the security of their environment. Additionally, the tool can be used to assess the risk of deploying commercial *software*, such as *Commvault* and *Palo Alto Prisma Cloud*, in customers' cloud computing environment.

Keywords: Security. Cloud Computing. Identity Management. Data Leakage. Least Privilege.

Lista de Abreviaturas e Siglas

API	Application Programming Interface
AWS	Amazon Web Services
CEO	Chief Executive Officer
CSV	Comma Separated Values
EC2	Elastic Compute Cloud
GB	Gigabytes
HTML	HyperText Markup Language
IaaS	Infrastruture As A Service
IAM	Identity And Access Management
JSON	JavaScript Object Notation
LGPD	Lei Geral de Proteção de Dados Pessoais
MFA	Multi Factor Authentication
PaaS	Platform As A Service
PII	Personal Identifiable Information
S3	Simple Storage Service
SaaS	Software As A Service
SDK	Software Development Kit
SNS	Simple Notification Service
SQS	Simple Queue Service
URL	Uniform Resource Locator
VPC	Virtual Private Cloud

Lista de Figuras

1	Matriz de Responsabilidade Compartilhada	9
2	Relação de Confiança de uma Função do IAM	16
3	Exemplo de política do IAM	17
4	Política do IAM para o serviço S3	19
5	Metodologia utilizada	22
6	Arquitetura da ferramenta	23
7	Comando para gerar nova tabela de ações	24
8	Tabela de ações por serviço	24
9	Tabela de ações finalizada	25
10	Mensagem de erro	27
11	Histórico de resultados das análises	28
12	Demonstração de execução da ferramenta	29
13	Política para o serviço DynamoDB	30
14	Resultado da análise da política para o serviço S3	31
15	Arquitetura de Trabalhos Futuros	34

Sumário

1	Introdução	9
2	Fundamentação Teórica	12
2.1	Computação em nuvem	12
2.1.1	Tipos de computação em nuvem	13
2.2	Gerenciamento de identidade e acesso	14
2.3	Elementos de identidade	14
2.3.1	Usuários	14
2.3.2	Grupos de usuários	14
2.3.3	Funções	15
2.4	Elementos de gerenciamento de acesso	16
2.4.1	Estrutura de documento de política JSON	17
2.4.2	Expressões Regulares em políticas do IAM	18
2.5	Simulador de políticas do IAM	18
3	Trabalhos Relacionados	20
4	Metodologia	22
5	Desenvolvimento do Trabalho	23
5.1	Definição da arquitetura	23
5.2	Tabela de ações	24
5.3	Lista de ações críticas	25
5.4	Analizador de políticas	26
5.5	Histórico de análises	27
5.6	Disponibilização em repositório público	28
6	Testes e Resultados	28
6.1	Testes com modelos de Políticas	28
6.2	Casos de Uso	31
6.3	Desafios Durante o Desenvolvimento	32
7	Conclusão e Trabalhos Futuros	33
7.1	Futuras implementações	33

1. Introdução

Computação em nuvem já é uma realidade: diversas formas de uso e novas aplicações surgem e a demanda por profissionais que entendam a mudança e preparem as organizações para esse novo paradigma da computação só aumenta [1]. A popularidade dos serviços baseados em nuvem deve-se a sua acessibilidade e baixo custo [2]. A nuvem torna possível que usuários acessem a informação, de qualquer lugar e a qualquer momento, o que elimina a necessidade de estarem próximos aos computadores que armazenam tais dados. Estabelecida a conexão com a Internet, o usuário pode acessar os serviços de computação em nuvem a partir de vários tipos de *hardware* - um computador, *laptop*, *tablet* ou *smartphone* [3].

Serviços de computação em nuvem trabalham com um modelo de responsabilidade compartilhada. Como referência, considerou-se o serviço de nuvem *Amazon Web Services* (AWS) - líder mundial em serviços de computação em nuvem - e plataforma escolhida para a elaboração deste trabalho. A AWS, basicamente, divide o seu ambiente em duas camadas [4]:

- Segurança **da** nuvem - infraestrutura que suporta os serviços da AWS;
- Segurança **na** nuvem - tudo o que a AWS disponibiliza ao cliente para que seja feita a gestão.

A Figura 1 mostra, detalhadamente, o modelo explicado. Como se percebe, o cliente é o responsável pela gestão da identidade, pelo acesso de seus usuários e serviços, pela proteção de seus próprios dados, além da criptografia e configurações de rede. Isso significa que a AWS não se responsabiliza, por exemplo, por vazamentos de dados sensíveis, arquivos não criptografados expostos para a Internet ou configurações de rede inadequadas. Ficam evidentes, portanto, na plataforma, as responsabilidades de cada uma das partes envolvidas na matriz.

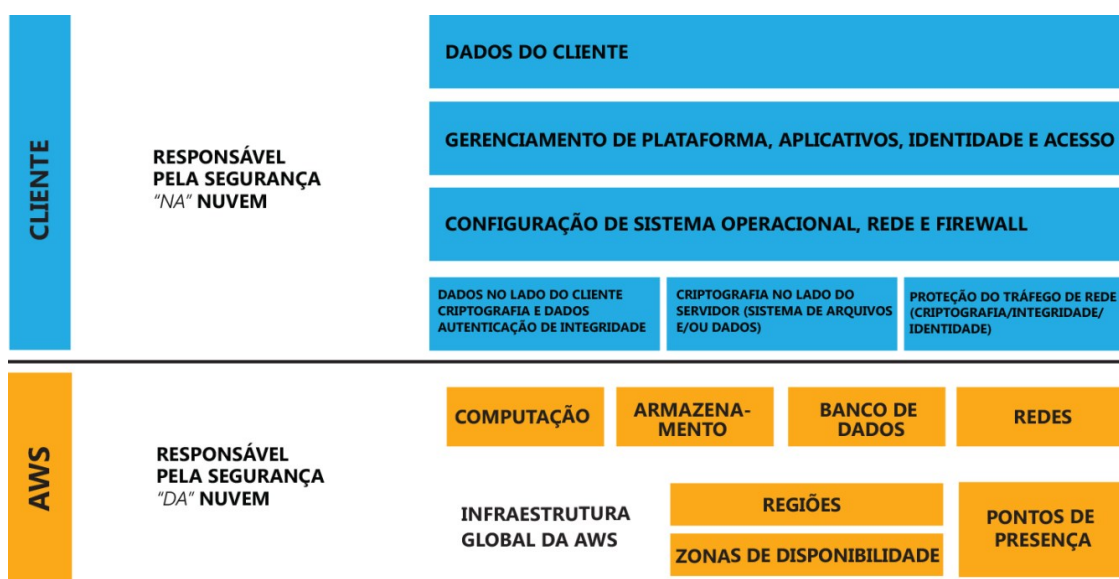


Figura 1. Matriz de Responsabilidade Compartilhada

A computação em nuvem tem desafiado todas as áreas de tecnologia a inovarem a cada dia, em especial, a área de segurança da informação. Com a existência de leis

regulatórias e auditorias, a preocupação com o acesso às informações armazenadas na nuvem tem sido um tema abordado constantemente. Funcionários que possuem acesso a dados de aplicações em nuvem, a partir de dispositivos como *laptops* e *smartphones*, podem representar ameaças de segurança, principalmente, quando a negligência do colaborador e o mau uso de credenciais estão envolvidos.

Em 29 de dezembro de 2020, ocorreu um vazamento de 574GB de dados envolvendo a plataforma de computação em nuvem AWS e uma empresa de serviços financeiros chamada Prisma Promotora [5]. O time de pesquisa da vpnMentor [6], liderado por Noam Rotem, relatou que, por falta de cuidados da parte do cliente, o volume de dados estava armazenado na plataforma AWS, sem a devida proteção. Entre os dados vazados, havia fotos de documentos de identidade, fotos de cartões de crédito, comprovantes de situação cadastral na Receita Federal, planilhas contendo informações de cadastro de clientes, entre outros. Relatos sobre vazamentos de dados e brechas de segurança vêm aumentando assustadoramente nos últimos tempos, o que reforça a importância de uma gestão de acessos bem elaborada no ambiente da nuvem.

De acordo com o Art. 46 da Lei Nº13.709, de 14 de agosto de 2018, mais conhecida como Lei Geral de Proteção de Dados Pessoais (LGPD), os agentes de tratamento devem adotar medidas de segurança - técnicas e administrativas - aptas a proteger os dados pessoais de acessos não autorizados e de situações acidentais ou ilícitas de destruição, perda, alteração, comunicação ou qualquer forma de tratamento inadequado ou ilícito. Portanto, tendo em vista proteger a informação de acessos não autorizados, e, a fim de reduzir os riscos associados aos serviços de nuvem, empresas necessitam de uma solução de *Identity and Access Management* (IAM) robusta, adaptável, escalável e segura perante os usuários [2].

O IAM - Gerenciamento de Identidade e Acesso - é um serviço da AWS muito eficiente para o controle de usuários e acessos. É a porta de entrada para a AWS, através da qual todos os usuários recebem suas credenciais e permissões para acessar os mais de 300 serviços da plataforma. Ele permite ainda a criação de usuários, grupos, funções e políticas para gerenciar os acessos de forma inteligente e escalável. No entanto, como essa é uma responsabilidade da parte do cliente (segurança **na** nuvem), faz-se necessário entender todos os seus recursos e como configurá-los corretamente. O controle de acesso para um ambiente tão amplo envolve alta complexidade.

O acesso a um recurso fora do escopo de trabalho de uma área pode acarretar grandes prejuízos para uma empresa. Por esse motivo, empresas têm procurado alternativas para mitigar o risco de usuários conseguirem acessos não autorizados. Este trabalho apresenta uma solução capaz de analisar políticas de acesso do IAM, focando na identificação de ações que apresentem algum tipo de risco para as organizações.

O AWS IAM permite que o próprio usuário tenha a liberdade de elaborar sua política de acesso, definindo os serviços e o tipo de acesso (leitura ou escrita) necessários para a execução de suas tarefas. No entanto, a geração de políticas pode se tornar complexa para um usuário que não esteja familiarizado com os serviços de computação em nuvem e com o funcionamento do serviço IAM. Vale ressaltar que o acesso aos serviços da AWS são baseados em chamadas de *Application Programming Interface* (API). No IAM, existem milhares de ações responsáveis por gerenciar cada serviço da AWS e cada ação representa

uma chamada de API específica.

Diante disso, existe uma grande dificuldade na criação de políticas de controle de acesso bem elaboradas por parte do usuário que utiliza a plataforma de nuvem, seguindo as melhores práticas de segurança e menor privilégio. Consequentemente, tais usuários, muitas vezes por falta de experiência, ou também por falta de tempo, optam por criar políticas de acesso muito permissivas, que garantem acessos além dos necessários para a execução de suas atividades. Outra alternativa para o usuário iniciante é a utilização de políticas e funções gerenciadas pela AWS. Tais políticas são formuladas de maneira genérica, pois seu objetivo é atender a todos os requisitos de um serviço e facilitar o seu uso. No entanto, essas políticas não priorizam o fator segurança.

Diante da falta de uma análise de acessos visando à segurança dos dados, da infraestrutura na nuvem e do aumento do número de relatos sobre vazamentos de dados, torna-se o objetivo deste trabalho realizar a avaliação de políticas de acesso quanto a sua criticidade. Para isso, foi desenvolvida uma ferramenta de análise de risco para políticas de controle de acesso para o serviço AWS IAM, capaz de gerar informações relevantes para o profissional de Segurança da Informação, que podem evidenciar falhas de segurança, possíveis riscos de vazamento de dados, impactos e indisponibilidades em serviços hospedados na nuvem. Ela também fornece ao usuário recomendações, a fim de que se crie uma política mais restritiva e condizente com as boas práticas de segurança de controle de acesso.

Para alcançar o objetivo proposto, durante todo o projeto, a seguinte questão foi levada em consideração: “Como minimizar a possibilidade de um usuário acessar informações sensíveis e de alto risco, passíveis de prejuízos a nível organizacional?”. Sendo assim, a resposta para a pergunta foi: “Criando uma automação que avalia as permissões do usuário antes da concessão do acesso”. A metodologia adotada para conduzir a solução do problema consistiu em três etapas principais. Primeiramente, foi realizado o levantamento de todas as permissões existentes no serviço AWS IAM [7], e feita a análise de risco de cada uma delas. A análise de risco consistiu na identificação de ações de nível de escrita e leitura, capazes de visualizar, criar, alterar ou deletar algum recurso, regra ou credencial que possa causar impacto em aplicações hospedados na plataforma de nuvem. As ações consideradas de alto risco foram armazenadas em uma lista, que serviu como base para a análise das políticas de acesso. A segunda etapa consistiu no desenvolvimento da lógica do analisador de políticas, que envolveu a implementação de validações de formatação, tratamento de erros e análise de expressões regulares presentes nas políticas. E, por fim, foi realizada a discussão dos resultados e a aplicação da solução em cenários reais, analisando políticas de ferramentas conhecidas internacionalmente, a fim de se descobrir o risco relacionado às permissões e, com isso, criar estratégias para mitigá-lo.

Durante a fase de planejamento, foi realizada uma ampla busca por trabalhos e artigos similares ao tema proposto, porém nenhum foi encontrado. Portanto, devido à inexistência de documentos relacionados à análise crítica de políticas de acesso, foi necessário realizar uma pesquisa por produtos e serviços existentes no mercado da tecnologia, que oferecem soluções semelhantes à proposta neste trabalho. O módulo *AWS IAM Access Analyser* foi o produto encontrado que mais se assemelha a este tema, capaz de identificar falhas de formatação e, até o momento de escrita deste trabalho, apenas 2 ações

consideradas de risco para o ambiente da nuvem. O diferencial da ferramenta desenvolvida neste trabalho é a sua capacidade de identificar mais de 500 dessas ações consideradas de alto risco, críticas para o ambiente, entre as mais de 13 mil ações existentes na AWS para gestão da plataforma.

Os primeiros resultados indicam que a ferramenta mostra-se efetiva não só na identificação de permissões de alto risco, mas também nas recomendações para o usuário final. A solução foi precisa na detecção das ações críticas em todos os testes realizados, tendo sucesso também em casos onde apareceram expressões regulares ou erros de formatação. São realizadas simulações com modelos de políticas de diferentes níveis de complexidade, incluindo vários serviços da AWS. Para facilitar a compreensão pelo usuário, os resultados das análises são salvos em um diretório, onde é possível consultar todas as análises executadas previamente, identificar as permissões críticas, suas definições, e acessar o *link* da documentação oficial para os serviços AWS, cujo conteúdo descreve, de forma mais detalhada, as permissões descobertas.

Em síntese, conclui-se que a ferramenta contribui, não só para minimizar o tempo gasto por analistas de segurança dedicado à análise de políticas de acesso, como também ajuda a diminuir o risco de falha humana ao se avaliar tais políticas, ou seja, aumenta a precisão na identificação de ações de alto risco. Além disso, a ferramenta visa conscientizar o usuário sobre o princípio do menor privilégio (ter acesso somente ao necessário para realizar suas atividades) e minimizar o acesso de usuários a dados críticos para as organizações ou a permissões que possam gerar impactos e indisponibilidades em aplicações.

2. Fundamentação Teórica

Nesse capítulo, são apresentados os principais conceitos utilizados na concepção desta ferramenta - essenciais para a compreensão do trabalho. Os conceitos são percorridos em duas subseções. Inicialmente, é apresentada uma explicação sobre o conceito de computação em nuvem e plataformas que fornecem serviços de computação em nuvem. Posteriormente, é apresentado o serviço *AWS Identity and Access Management (IAM)*, seus recursos e funcionalidades importantes, o que representa a base para a implementação da ferramenta proposta neste trabalho.

2.1. Computação em nuvem

A expressão “computação em nuvem” foi empregada pela primeira vez em 2005, por Eric Schmidt, então CEO do Google [1]. Computação em nuvem é a disponibilidade de recursos computacionais pela Internet, é um conjunto de recursos virtuais facilmente utilizáveis e acessíveis, tais como *hardware*, *software*, plataformas de desenvolvimento e serviços. Esses recursos podem ser dinamicamente reconfigurados para se ajustarem a uma carga de trabalho (*workload*) variável, permitindo a otimização do seu uso. Esse conjunto de recursos é tipicamente explorado através de um modelo “pague-pelo-uso” (*pay-as-you-go*), com garantias oferecidas pelo provedor através de acordos de níveis de serviços [1].

A computação em nuvem visa substituir a infraestrutura de TI tradicional, que se baseia em servidores físicos gerenciados internamente por profissionais de TI. Com a computação em nuvem, não é necessário provisionar recursos em excesso para atender picos de atividades empresariais do futuro, pois uma de suas vantagens é a escalabilidade.

Provisiona-se somente a quantidade de recursos necessária e, através de poucos cliques, é possível aumentar ou diminuir instantaneamente a escala destes recursos para ajustar a capacidade de acordo com a evolução das necessidades empresariais [8].

Adicionalmente, um dos maiores benefícios da computação em nuvem é a habilidade de “pagar conforme o uso”. O cliente paga, somente, pelo poder computacional que consome. Portanto, o propósito da computação em nuvem é aumentar a eficiência no uso de recursos computacionais e, conseqüentemente, tornar a operação de TI mais econômica. Como consequência das vantagens apresentadas anteriormente, o conceito ganhou espaço rapidamente e, atualmente, está presente em grande parte dos sistemas existentes em todo o mundo.

2.1.1. Tipos de computação em nuvem

Os três principais tipos de computação em nuvem são: infraestrutura como serviço, plataforma como serviço e software como serviço. Cada tipo de computação em nuvem oferece diferentes níveis de controle, flexibilidade e gerenciamento para que se possa selecionar o conjunto adequado de serviços, a fim de atender demandas específicas [8].

- **Infraestrutura como Serviço (IaaS):** contém os componentes básicos da TI na nuvem. Normalmente, o IaaS oferece acesso a recursos de rede, computadores (virtuais ou em *hardware* dedicado) e espaço de armazenamento de dados. O IaaS oferece o mais alto nível de flexibilidade e controle de gerenciamento sobre os recursos de TI. Trata-se do tipo de computação que mais se assemelha aos recursos existentes de TI, já conhecidos por vários departamentos e desenvolvedores de TI. Serviços como *Amazon Web Services (AWS)*, *Microsoft Azure*, *Google Cloud Platform (GCP)*, *Oracle Cloud Infrastructure (OCI)*, entre outros, disponibilizam o modelo IaaS.
- **Plataforma como Serviço (PaaS):** disponibiliza à empresa uma plataforma completa, envolvendo *hardware*, *software* e infraestrutura para o desenvolvimento, implantação e gerenciamento de uma aplicação própria. É contratado um ambiente no qual o time de desenvolvimento tem à sua disposição uma infraestrutura completa, sem que sejam feitos investimentos vultosos, ou seja, não há preocupação com aquisição de recursos, planejamento de capacidade, manutenção de software, correções ou qualquer outro tipo de trabalho genérico repetitivo necessário para a execução dos aplicativos. Essa modalidade de serviço em nuvem é voltada para empresas que visam desenvolver, testar e implementar aplicações de forma mais barata. O serviço *AWS Elastic Beanstalk* é um exemplo interessante de PaaS, pois este serviço oferecido pela AWS é capaz de abstrair toda a infraestrutura de uma aplicação. Basta que o desenvolvedor da aplicação realize a implantação *deploy* da aplicação, que o serviço provisiona toda a infraestrutura necessária para que ela funcione propriamente, garantindo alta disponibilidade, escalabilidade, estabilidade e durabilidade.
- **Software como Serviço (SaaS):** oferece um produto completo, executado e gerenciado pelo provedor de serviços. Na maioria dos casos, quando as pessoas mencionam SaaS, estão se referindo a aplicativos de usuários finais (como *e-mail* baseado na *web*). Com uma oferta de SaaS, o cliente não precisa pensar a respeito

da manutenção do serviço ou no gerenciamento da infraestrutura subjacente. Sua única preocupação será com a forma de utilização desse *software* específico. Alguns exemplos de SaaS são o *Palo Alto Prisma Cloud* [9], o *Commvault* [10], o *Google Workspace* [11] e o *Microsoft 365* [12].

A ferramenta em questão neste trabalho será aplicada ao modelo de computação em nuvem IaaS, mais especificamente, na plataforma de computação em nuvem *Amazon Web Services* (AWS). É importante salientar esse aspecto, pois é no tipo IaaS que se encontra maior complexidade na gestão de permissões de acesso, devido ao grande número de funcionalidades oferecidas pelos provedores deste tipo de serviço. Além disso, a solução deste trabalho também pode ser aplicada na avaliação de risco de produtos SaaS (*Software como Serviço*), que oferecem serviços de implantação na plataforma AWS. É o caso das soluções de SaaS *Palo Alto Prisma Cloud* [9] e *Commvault* [10], citadas anteriormente.

2.2. Gerenciamento de identidade e acesso

O AWS IAM é o serviço capaz de realizar a gestão dos acessos aos recursos da AWS de forma segura. É utilizado para controlar quem é autenticado (fez login) e autorizado (tem permissões) para usar os recursos da plataforma [13]. O serviço foi projetado para atender os requisitos de acesso para os mais de 300 serviços existentes na AWS. Sua infraestrutura é, portanto, robusta o bastante para suportar e processar um grande número de requisições providas desses serviços.

Entre os vários elementos que constituem a infraestrutura do IAM, é importante destacar os elementos de identidade e de gerenciamento de acesso.

2.3. Elementos de identidade

2.3.1. Usuários

Um usuário do *AWS Identity and Access Management* (IAM) é um recurso que se cria na AWS para representar a pessoa ou a aplicação que o utilizará para interagir com a AWS. Um usuário na AWS possui um nome e credenciais de autenticação [14]. Basicamente, o usuário pode se autenticar de duas maneiras: utilizando uma senha de console ou chaves de acesso. A senha de console é utilizada para se acessar a console da AWS, a interface *web* de interação com a plataforma. Já as chaves de acesso são utilizadas para se acessar a AWS, via acesso programático, utilizando o terminal do computador ou SDKs (*Software Development Kits*). Cada usuário pode ter, no máximo, duas chaves de acesso por vez.

O IAM também oferece ao usuário a possibilidade de se adicionar uma camada extra de segurança, com a autenticação multifator (MFA) [15]. A configuração do MFA é recomendada para todos os usuários do IAM, com o intuito de adicionar mais uma camada de proteção aos recursos da AWS.

2.3.2. Grupos de usuários

Um grupo de usuários do IAM é um conjunto de usuários do IAM. Os grupos de usuários permitem especificar permissões para vários usuários simultaneamente, o que pode facilitar o gerenciamento das permissões para os mesmos [16]. Por exemplo,

existe um grupo de nome *Admins* e nesse grupo são adicionadas todas as permissões de que usuários administradores normalmente precisam. Qualquer usuário inserido dentro desse grupo herdaria as permissões do próprio grupo. Caso um novo usuário ingresse na organização e precise de privilégios de administrador, basta adicioná-lo no grupo *Admins* e ele terá as permissões necessárias para executar suas atividades. Como se observa, o uso de grupos de usuários facilita a atribuição de permissões a vários usuários.

2.3.3. Funções

Uma função do IAM é uma identidade que pode ser criada em sua conta AWS, com políticas de permissões específicas. Uma função do IAM assemelha-se a um usuário do IAM, no sentido de que é uma identidade da AWS com políticas de acesso, que determinam o que a identidade pode ou não fazer na AWS. No entanto, ao invés de ser exclusivamente associada a uma pessoa, uma função destina-se a ser assumida por qualquer indivíduo que dela necessitar. Além disso, uma função não tem credenciais de longo prazo padrão, como uma senha ou chaves de acesso, associadas a ela. Em vez disso, quando se assume uma função, ela fornecerá credenciais de segurança temporárias para sua sessão de função [17]. Recomenda-se que toda aplicação hospedada na AWS faça uso de funções do IAM, em vez de usuários, justamente pelo fato de não haver risco envolvendo vazamento de senhas, chaves de acesso ou credenciais de autenticação. Isso é possível pelo fato da função possuir uma relação de confiança com o serviço ou identidade que a utiliza [18]. Esta relação de confiança é personalizável e deve ser elaborada com muita atenção, seguindo o conceito de menor privilégio.

Em vez de se utilizar senhas ou credenciais para realizar a autenticação, a função utiliza a política de relação de confiança para este fim. Esta é a área onde será feita a validação condicional no momento da autenticação, como exibe a Figura 2. Somente o que estiver contido nesta política de relação de confiança, poderá se autenticar e fazer uso da função [19]. Logo, pode-se concluir que este elemento garante a segurança da função, e restringe seu acesso a somente as identidades realmente permitidas, oferecendo, assim, maior segurança ao procedimento de autenticação. É possível restringir o acesso de várias maneiras, especificando algumas das informações a seguir [20]:

- Nome do usuário;
- Nome da função;
- IP de origem do usuário ou aplicação;
- Faixa de IP de uma rede;
- Autenticação realizada utilizando duplo fator de autenticação (MFA);
- Nome do serviço AWS;
- Nome do recurso AWS;
- Número da conta AWS;
- Código da Organização AWS;
- Código externo, para acessos de identidades de outras contas AWS;
- Período de acesso organizado por data;
- *Tags*.

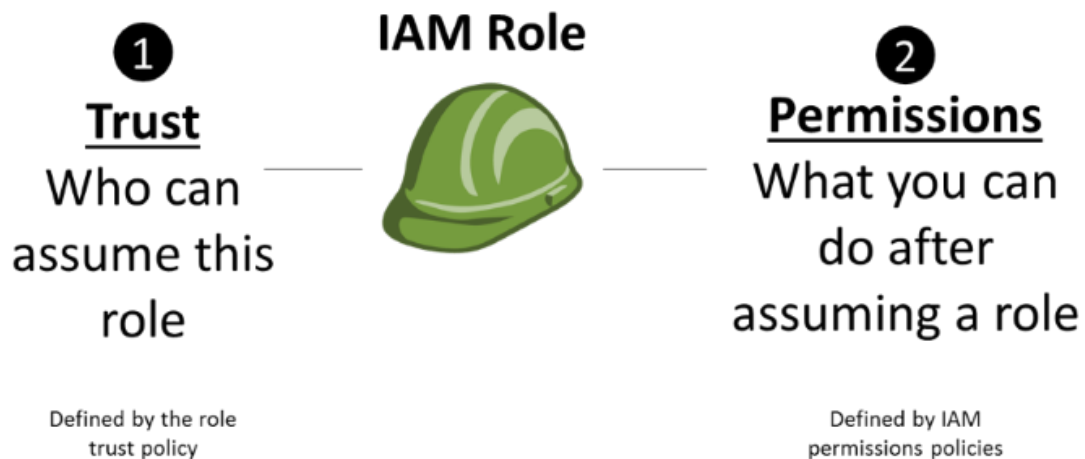


Figura 2. Relação de Confiança de uma Função do IAM

As funções podem ser usadas pelas seguintes entidades:

- Um usuário do IAM na mesma conta da AWS que a função;
- Um usuário do IAM em uma conta da AWS diferente da conta da função;
- Uma outra função na mesma conta da AWS que a função;
- Uma outra função em uma conta da AWS diferente da conta da função;
- Um serviço da Web oferecido pela AWS, como o Amazon Elastic Compute Cloud (Amazon EC2);
- Um usuário externo autenticado por um serviço de provedor de identidade (IdP).

Sendo assim, pode-se notar que o uso de funções é versátil e bastante útil em cenários onde serviços, usuários ou outras funções precisam realizar alguma atividade na plataforma AWS. Por exemplo, é possível conceder aos usuários de uma conta AWS acesso temporário a recursos que normalmente não têm, ou conceder aos usuários, em uma conta da AWS acesso a recursos em outra conta. Elas também são úteis quando se deseja conceder a colaboradores terceiros acesso a uma conta AWS, para que possam realizar uma atividade de auditoria em seus recursos.

2.4. Elementos de gerenciamento de acesso

Usuários, grupos de usuários e funções são criados inicialmente sem nenhuma permissão, até que lhes seja atribuída uma política de acesso. A gestão de acesso na AWS é realizada criando-se políticas e anexando-as às identidades do IAM. Uma política é um objeto na AWS que, quando associado a uma identidade ou a um recurso, define suas permissões, ou seja, o que um usuário ou função está autorizado a realizar na plataforma AWS. Logo, a AWS avalia essas políticas quando uma entidade de segurança do IAM (usuário ou função) faz uma solicitação. As ações nas políticas determinam se a solicitação será permitida ou negada. As políticas são armazenadas na AWS como documentos JSON (*JavaScript Object Notation*) [21], conforme exemplifica a Figura 3. Este formato foi determinado por ser considerado de fácil interpretação e manipulação.


```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Resource": "arn:aws:ec2:*:*:instance/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Owner": "${aws:username}"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:DescribeInstances",
      "Resource": "*"
    }
  ]
}

```

Figura 3. Exemplo de política do IAM

2.4.1. Estrutura de documento de política JSON

Como evidenciado na Figura 3, os documentos de política do IAM são elaborados em formato JSON. Esses documentos incluem dois elementos básicos: um cabeçalho, constituído por informações opcionais da política na parte superior do documento, e uma ou mais instruções individuais, chamadas *Statements*. Cada *Statement* inclui informações sobre uma única permissão, ou sobre um conjunto de permissões. Caso uma política possua várias declarações, a AWS aplicará um "OU" (*OR*) lógico a todas as declarações no momento de avaliação.

A instrução é composta pelos seguintes elementos:

- *Version*: versão do idioma da política. É aconselhável que esse valor permaneça sempre o mesmo para todas as políticas. A versão mais recente é a *2012-10-17*.
- *Statement*: elemento em que está localizado o conjunto das instruções. É o principal elemento de uma política.
- *Sid* (Opcional): serve para inserir uma descrição à instrução, com o objetivo de diferenciar as instruções umas das outras.
- *Effect*: o efeito que aquela instrução terá na política. Pode ter somente dois valores: *Allow* ou *Deny*, que servem para indicar se a política permite ou nega tal acesso.
- *Principal* (obrigatório apenas em algumas circunstâncias): especifica o agente que executa a ação. Pode ser um usuário, uma função ou um serviço da AWS. Para políticas do IAM, esse elemento não se faz necessário, pois a entidade principal fica implícita para o usuário ou função que utiliza a política.
- *Action*: lista de ações que a política permite ou nega acesso. As ações podem ser do tipo listagem, leitura, escrita ou "tagueamento".

- *Resource*: campo em que se especificam os recursos que poderão ser acessados ou terão seu acesso negado, dependendo do valor do elemento *Effect*. Um recurso pode ser um banco de dados, uma máquina virtual, um disco de um servidor, uma função do IAM, entre outros.
- *Condition* (opcional): as circunstâncias sob as quais a política concede a permissão. É possível criar expressões com operadores de condição para fazer a correspondência de chaves e valores de condição na política com os valores e chaves no contexto da solicitação. Por exemplo, é possível usar *Conditions* para restringir o IP de onde um acesso é originado, ou para impedir que um usuário faça alterações somente em recursos que possuem uma *tag* específica, entre outros cenários.

2.4.2. Expressões Regulares em políticas do IAM

Para os elementos *Principal*, *Action*, *Resource* e *Condition*, é possível fazer uso de expressões regulares, de modo a facilitar a construção das políticas de permissões, pois permite adicionar um ou mais itens em apenas uma linha [22]. Como mostra a Figura 3, o elemento *Resource* é um exemplo de uso de expressão regular. No lugar onde se especificaria a região, a conta AWS e o *ID* da instância do serviço *EC2*, faz-se uso do sinal asterisco ("*"), que significa zero ou mais ocorrências de caracteres. Logo, pode-se ler a primeira instrução da política da seguinte forma: Permissão de iniciar e parar qualquer instância do serviço *EC2*, independentemente de sua região, conta AWS ou nome, na condição de que o recurso possua uma *Tag* de chave "Owner" e seu valor seja o nome do usuário que executa a ação. A segunda instrução é mais simples de ser interpretada e pode ser lida da seguinte forma: Permissão de descrever todas as instâncias do serviço *EC2*, sem exceções.

A Figura 4 exibe outro exemplo de uso de expressão regular, de modo a simplificar a política. Pode-se identificar que a primeira instrução inclui ações básicas para navegação entre os recursos, as quais permitirão que o usuário veja que recursos estão presentes naquela conta AWS, bem como suas características. Isso só é possível devido à generalização do elemento *Resource*, que faz uso do sinal de asterisco ("*"). A segunda instrução já mostra uma única ação, seguida do elemento *Resource*, indicando que o usuário poderá listar os itens presentes apenas naquele recurso especificado na política. Por fim, a terceira instrução mostra um caso de expressão regular eficiente, colocando-se o sinal de asterisco ("*") ao iniciar e ao finalizar a ação. Essa instrução permite que o usuário realize todos os tipos de manipulação de objetos que estejam contidos no recurso de nome *bucket-name*, podendo ser uma ação de leitura, escrita, deleção ou edição do objeto. Existem, ao todo, 76 ações diferentes que contêm a palavra *Object*, seja no seu início, meio ou fim. A instrução expressa em somente uma linha é equivalente a todas as 76 ações.

2.5. Simulador de políticas do IAM

Considerando os elementos que constituem as instruções e o número de ações existentes, é possível elaborar políticas com diferentes níveis de complexidade, capazes de limitar o acesso de usuários e funções com precisão. É possível, por exemplo, elaborar uma política que permita que o usuário tenha permissão de alterar a sua própria senha,

```

example-policy-s3.json
1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Sid": "ConsoleAccess",
6        "Effect": "Allow",
7        "Action": [
8          "s3:GetAccountPublicAccessBlock",
9          "s3:GetBucket*",
10         "s3:Get*Configuration",
11         "s3:ListAllMyBuckets"
12       ],
13       "Resource": "*"
14     },
15     {
16       "Sid": "ListObjectsInBucket",
17       "Effect": "Allow",
18       "Action": "s3:ListBucket",
19       "Resource": ["arn:aws:s3:::bucket-name"]
20     },
21     {
22       "Sid": "AllObjectActions",
23       "Effect": "Allow",
24       "Action": "s3:*Object*",
25       "Resource": ["arn:aws:s3:::bucket-name/*"]
26     }
27   ]
28 }

```

Figura 4. Política do IAM para o serviço S3

porém não a dos outros usuários. Também é possível limitar o acesso a recursos por meio do nome, ou por meio de *tags* de identificação, endereços de IP, região, conta AWS, entre outras possibilidades.

A forma mais simples de se criar uma política do IAM é por meio do próprio editor de políticas, presente na Console da AWS [23]. Para se ter acesso a esta funcionalidade, é necessário estar autenticado em uma conta AWS, a partir de um usuário ou função do IAM. O editor de políticas guia o usuário durante a formulação da política, permitindo que ele escolha o serviço que precisa de acesso, selecione as ações necessárias, especifique os recursos que precisa de acesso, e até permite criar condições para as instruções. Ele também permite que o usuário pesquise pela definição das ações, direcionando diretamente para a documentação oficial da AWS, onde terá todos os detalhes sobre aquela ação, como a sintaxe de resposta do comando, casos de uso, exemplos de código, entre outros. Ao concluir a construção da política, o editor entrega ao usuário a política completa, em formato JSON, para que o usuário possa editá-la, caso queira dar um toque final, e realizar algum ajuste na política, para que fique mais refinada.

Caso o usuário não possua acesso a uma Console AWS, existe outra alternativa para se gerar uma política do IAM. O *AWS Policy Generator* [24] é a solução da AWS para geração de uma política do IAM sem fazer uso da Console da AWS. Assim como o editor de políticas explicado anteriormente, ele permite que o usuário selecione os serviços, ações, recursos e condições que deseja incluir na sua política. No entanto, não é possível

encontrar a definição das ações por esta ferramenta, sendo necessário, assim, que o usuário pesquise por conta própria, na documentação oficial da AWS [7], sobre as ações que atenderão sua necessidade. Em contrapartida, esta ferramenta traz uma vantagem, pois permite que o usuário gere não só políticas do IAM, como também políticas de outros serviços AWS que requerem o uso de políticas para restrição de acessos, como S3, SNS, SQS e VPC.

Foi realizado um levantamento de todos os serviços AWS e as ações existentes para cada um deles, incluindo suas descrições, nível de acesso, tipos de recursos suportados, entre outras informações. Até o presente momento de desenvolvimento deste trabalho, foram identificados 340 (trezentos e quarenta) serviços da AWS e aproximadamente 13.000 (treze mil) diferentes ações ao todo, o que mostra a alta granularidade e complexidade existente na elaboração de políticas. Sendo assim, a compreensão do serviço *AWS Identity and Access Management* é complexa e demanda certo estudo por parte do usuário, para que entenda todos os seus elementos e construa um ambiente na nuvem com acessos seguros, seguindo o conceito de menor privilégio.

3. Trabalhos Relacionados

Esta seção apresenta trabalhos relacionados a este estudo, focados no acesso não autorizado a informações sensíveis na nuvem. Serão citados dois trabalhos: um envolvendo a prevenção de acesso não autorizado por meio de um algoritmo de criptografia, e outro que aprofunda no rastreamento de acessos não autorizados por usuários na nuvem; além da apresentação do módulo *AWS IAM Access Analyzer*, uma utilidade bastante inteligente para apontar sugestões ao usuário sobre melhorias na construção de sua política do IAM. Posteriormente, será explicada a proposta deste trabalho e o motivo de esse apresentar um conceito diferente dos outros trabalhos.

Kumar e Nirmala [25] mostram em seu estudo uma solução que visa aumentar o nível de segurança e controle de acesso durante a manipulação de informações em bancos de dados distribuídos. Afirmam-se que a autenticação por usuário e senha não é o bastante para se garantir a proteção da informação, e que possíveis vazamentos de credenciais poderiam ser catastróficos para uma organização. Para acessar as informações armazenadas na nuvem, o estudo propõe a implementação de um algoritmo de autenticação, baseado em *blowfish calculation*, e o gerenciamento de chaves de criptografia, capaz de realizar um embaralhamento de chaves de criptografia juntamente com o *User ID* do usuário que está fazendo a solicitação, a fim de garantir que somente o usuário solicitante terá acesso àquelas informações.

O algoritmo de autenticação por criptografia mostra-se mais eficiente que o conhecido algoritmo *Advanced Encryption Standard* (AES). Quando pensado em larga escala, o algoritmo, em comparação com técnicas existentes, seria capaz de otimizar consideravelmente o tempo de execução, tanto para criptografia quanto para descryptografia de dados.

Já no trabalho realizado por Mehta e Saini [3], a solução proposta visa a detecção de acessos a servidores por usuários não autorizados, por razões de comprometimento de dados ou a presença de possíveis invasores. Foram implementados módulos, a fim de satisfazer todas as etapas de implementação da solução. O módulo de corrupção e modificação não autorizada de dados é o módulo mais importante da solução, encarregado

de identificar qual servidor possui inconsistências de dados e qual é a gravidade do incidente. No diagrama de atividades evidenciado no estudo, existe um usuário administrador responsável pela gestão de acessos, pelo bloqueio e desbloqueio de endereços IP com permissão para se conectar ao servidor e, não menos importante, pela detecção de atacantes e possíveis invasores.

Por fim, o estudo conclui que a segurança aplicada no armazenamento de dados na nuvem ainda encara muitos desafios. A pesquisa também expressa a possibilidade de futuras pesquisas relacionadas a este estudo, no sentido de automatizar a verificação dos acessos de usuários, reduzindo, assim, a necessidade de ação humana e, consequentemente, melhorando-os em precisão e eficiência. No entanto, os estudos relatados não focam na questão da análise de políticas de acesso atribuídas a usuários, que é de fato o objetivo deste trabalho, mas sim em outras maneiras de se proteger um ambiente através de criptografia e embaralhamento de informação.

O módulo *IAM Access Analyzer* [26] é um módulo do serviço AWS IAM que, entre outras funções, realiza a análise da política que está sendo construída pelo usuário através do simulador de políticas na Console da AWS [23]. Ele verifica a formatação da política, em tempo real, para o usuário, sendo inclusive capaz de identificar algumas falhas de segurança, como a referência equivocada para uma conta da AWS ou região, e até o uso de ações consideradas de risco [27]. No entanto, de todas as verificações que o *IAM Access Analyzer* é capaz de fazer, ele só possui a habilidade de identificar 2 ações do IAM consideradas de risco, que são a *iam:PassRole* e a *iam:CreateServiceLinkedRole* [28]. Ambas são, de fato, críticas e devem ser utilizadas com responsabilidade e cuidado. A solução proposta por este trabalho, no entanto, vai além do que é proposto pelo *IAM Access Analyzer*, já que é capaz de identificar mais de 500 (quinhentas) ações do IAM classificadas como críticas.

Outra funcionalidade interessante do módulo *IAM Access Analyzer* é a sua capacidade de gerar políticas de permissão refinadas com base na atividade de acesso capturada em seus *logs* de chamadas de API, executadas na AWS por usuários e funções. A partir das descobertas feitas pelo módulo, é possível gerar políticas que concedem apenas as permissões necessárias para operar a aplicação. De forma sucinta, a partir da integração do *IAM Access Analyzer* ao serviço *AWS Cloudtrail*, um serviço de auditoria, oferecido pela AWS, que captura todos os eventos de ações realizados por um usuário, uma função ou um produto AWS, o módulo analisa os eventos interceptados pelo *Cloudtrail* e oferece ao usuário informações relevantes para a segurança do ambiente da nuvem, que permitem que o usuário identifique oportunidades para restringir suas permissões [29]. Algumas dessas informações são:

- Os eventos mais recentes realizados por um usuário, função ou produto AWS;
- O endereço IP do qual o evento foi realizado, o agente e a data da ação;
- Se um recurso está sendo compartilhado para fora da sua zona de confiança;
- Recomendações de remoção de ações que existem em uma política, mas que não estão sendo utilizadas ultimamente.

Apesar da existência de trabalhos que discutem a proteção do ambiente da nuvem contra acessos não autorizados, ainda há uma grande dificuldade em se encontrar uma solução universal, que se aplique a todas as plataformas de computação em nuvem. Isso

se deve, principalmente, à alta complexidade dos serviços de gestão de acessos e suas diferenças. O presente trabalho se diferencia dos demais pelo seu foco nas permissões dos usuários, nas suas políticas de permissão. Devido ao grande número de serviços oferecidos pela plataforma AWS, o nível de granularidade e refinamento envolvido na análise de políticas de acesso torna-se bastante elevado. Sendo assim, o intuito deste trabalho é trazer clareza para o usuário que utiliza a plataforma AWS, para que tenha maior facilidade na manipulação de suas próprias permissões. Na próxima seção, é apresentada a metodologia aplicada para o desenvolvimento da solução proposta.

4. Metodologia

A metodologia utilizada para o desenvolvimento deste trabalho consiste em 5 etapas, como descrita na Figura 5.

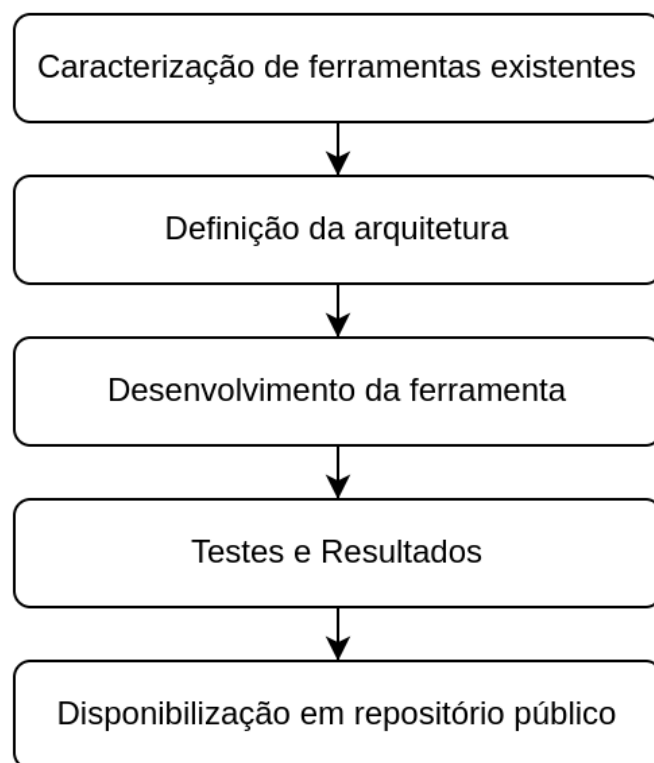


Figura 5. Metodologia utilizada

A primeira etapa do trabalho consistiu na pesquisa e caracterização de ferramentas já existentes, que exercem ações semelhantes ao tema proposto. Como apresentado na Seção 3, concluiu-se que há trabalhos acadêmicos que mostram soluções eficientes para a proteção da informação, porém tais trabalhos não possuem a mesma abordagem proposta neste estudo.

A segunda etapa tratou da definição da arquitetura e das tecnologias necessárias para o completo desenvolvimento do projeto. Foram mapeados todos os elementos necessários, como a base de dados, o analisador e o histórico de análises.

A terceira etapa foi dedicada ao desenvolvimento da ferramenta. Durante sua implementação, foi aplicada a arquitetura anteriormente definida.

Na quarta etapa, foi realizada a avaliação da ferramenta quanto a sua eficiência, levando em consideração taxas de acerto, estabilidade e tratamento de erros. Por fim, na quinta etapa, a ferramenta foi disponibilizada na plataforma GitHub, a fim de torná-la acessível publicamente, em formato de código aberto.

5. Desenvolvimento do Trabalho

O capítulo atual visa descrever, detalhadamente, todas as etapas de implementação do projeto. Dividido em etapas de desenvolvimento, ele começa pela definição da arquitetura do projeto, seguida da construção da base de dados, composta pela tabela de ações e a lista de ações críticas. Com a base de dados completa, foi possível seguir para desenvolvimento do analisador de políticas. Posteriormente, explicar-se-á o histórico de análises, em que se pode consultar o resultado de todas as análises realizadas e, por fim, é citado o perfil do GitHub, no qual está publicado o código fonte do projeto.

5.1. Definição da arquitetura

A definição da arquitetura, exibida na Figura 6, consiste no planejamento dos elementos que compõem a ferramenta. O analisador de políticas é o elemento central da ferramenta, onde está implementada toda a lógica de programação, capaz de avaliar uma política do IAM e detectar suas ações críticas. Ele faz uso da base de dados para identificar as ações críticas contidas na política analisada. Feita a análise da política, o analisador salva os resultados da análise dentro do diretório de histórico de análises. Todos os componentes da ferramenta estão armazenados em um repositório público no GitHub [30], disponível para qualquer pessoa que queira utilizá-los.

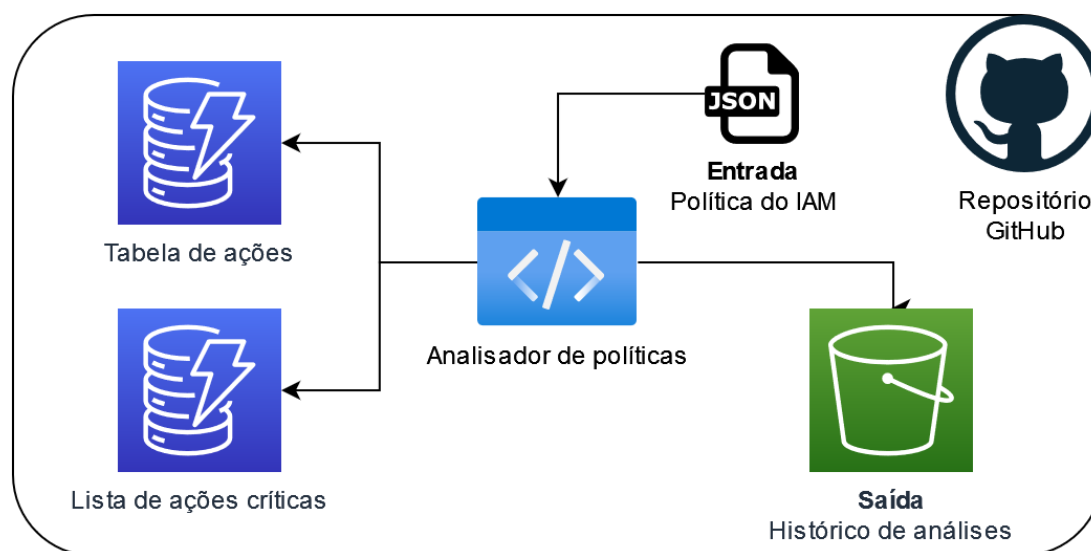


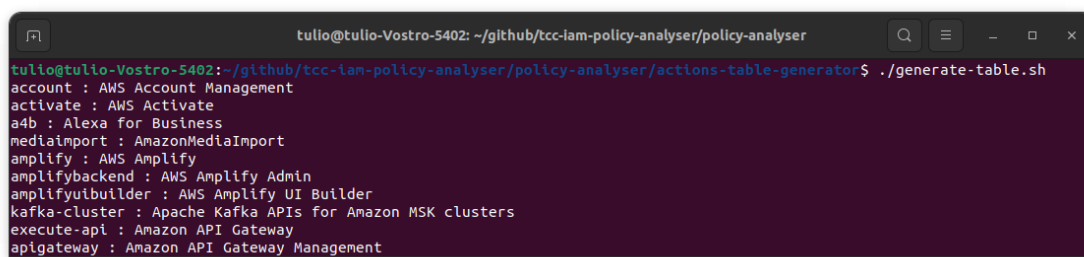
Figura 6. Arquitetura da ferramenta

A base de dados consiste em uma tabela de ações do IAM, contendo todos os serviços da AWS e suas ações, além da descrição e outros detalhes; e uma lista de ações

críticas, selecionadas e classificadas manualmente a partir da leitura da tabela de ações. Os critérios usados para classificar as ações como críticas estão descritos na seção 5.3. Já o histórico de análises consiste em um diretório, que armazena os resultados de todas as análises de política realizadas pela ferramenta. As análises são registradas em sub-pastas, cujo nome é a marca de tempo em que foram executadas.

5.2. Tabela de ações

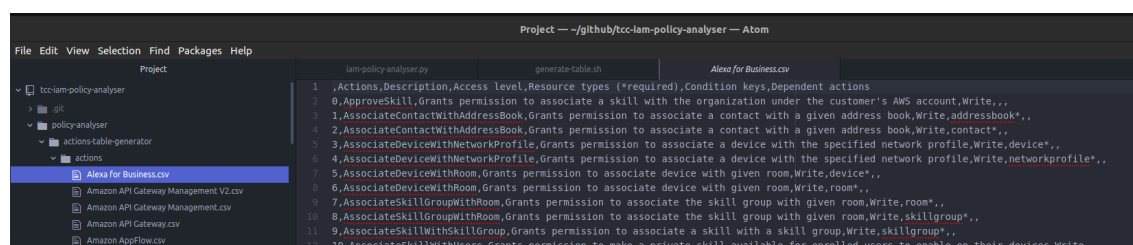
A primeira etapa do desenvolvimento deste trabalho consiste na geração da base de dados que será utilizada na análise das políticas de permissão. Todas as informações são coletadas diretamente da documentação oficial da AWS, na seção de ações, recursos e chaves de condição dos serviços da AWS [7]. Essa página lista todos os serviços oferecidos pela AWS, bem como suas devidas ações, tipos de recursos e chaves de condição suportadas por cada uma das ações. A Figura 7 exhibe o modo como se executa a atualização da tabela de ações, utilizando-se o terminal para acessar o diretório *policy-analyser/actions-table-generator/* e executando o arquivo *generate-table.sh*.



```
tulio@tulio-Vostro-5402: ~/github/tcc-iam-policy-analyser/policy-analyser
tulio@tulio-Vostro-5402:~/github/tcc-iam-policy-analyser/policy-analyser/actions-table-generator$ ./generate-table.sh
account : AWS Account Management
activate : AWS Activate
a4b : Alexa for Business
mediainport : AmazonMediaImport
amplify : AWS Amplify
amplifybackend : AWS Amplify Admin
amplifyuibuilder : AWS Amplify UI Builder
kafka-cluster : Apache Kafka APIs for Amazon MSK clusters
execute-api : Amazon API Gateway
apigateway : Amazon API Gateway Management
```

Figura 7. Comando para gerar nova tabela de ações

A geração da tabela consiste no levantamento de todos os nomes dos serviços AWS [7], bem como as URLs de cada um deles. O resultado da busca é armazenado em um arquivo chamado *service_urls.json*, em formato chave-valor, sendo chave, o nome do serviço, e valor, a URL que direciona para a tabela de ações do serviço. Em seguida, o código gerador da tabela de ações acessa cada uma das URLs e busca por duas informações principais: o prefixo de serviço e a tabela de ações do serviço. O prefixo de serviço é a sigla universal utilizada pelo serviço para definir suas ações no momento de definição de uma política. O prefixo de serviço é armazenado em um arquivo chamado *service_alias.json*, também em formato chave-valor, sendo chave, o nome do serviço, e valor, o prefixo do serviço. Já as tabelas de ações dos serviços são armazenadas em arquivos no formato CSV, no diretório *policy-analyser/actions-table-generator/actions/*, cujo nomes são compostos pelo nome do serviço (*Nome Do Serviço.csv*), como é mostrado na Figura 8.



File	Content
actions-table-generator/actions/Alexa for Business.csv	1 ,Actions,Description,Access level,Resource types (*required),Condition keys,Dependent actions 2 0,ApproveSkill,Grants permission to associate a skill with the organization under the customer's AWS account,Write,... 3 1,AssociateContactWithAddressBook,Grants permission to associate a contact with a given address book,Write,addressbook*,... 4 2,AssociateContactWithAddressBook,Grants permission to associate a contact with a given address book,Write,contact*,... 5 3,AssociateDeviceWithNetworkProfile,Grants permission to associate a device with the specified network profile,Write,device*,... 6 4,AssociateDeviceWithNetworkProfile,Grants permission to associate a device with the specified network profile,Write,networkprofile*,... 7 5,AssociateDeviceWithRoom,Grants permission to associate device with given room,Write,device*,... 8 6,AssociateDeviceWithRoom,Grants permission to associate device with given room,Write,room*,... 9 7,AssociateSkillGroupWithRoom,Grants permission to associate the skill group with given room,Write,room*,... 10 8,AssociateSkillGroupWithRoom,Grants permission to associate the skill group with given room,Write,skillgroup*,... 11 9,AssociateSkillWithSkillGroup,Grants permission to associate a skill with a skill group,Write,skillgroup*,... 12 10,AssociateSkillWithUsers,Grants permission to make a private skill available for enrolled users to enable on their devices,Write,...
actions-table-generator/actions/Amazon API Gateway Management V2.csv	
actions-table-generator/actions/Amazon API Gateway Management.csv	
actions-table-generator/actions/Amazon API Gateway.csv	
actions-table-generator/actions/Amazon AppFlow.csv	

Figura 8. Tabela de ações por serviço

Por fim, executa-se um código agregador de informações, que fará uso do arquivo *service_alias.json* para finalizar a geração da tabela de ações. O agregador percorre todos os arquivos de ações, dentro do diretório *policy-analyser/actions-table-generator/actions/*, coletando as informações das colunas *Actions*, *Description*, *Access level* e *Resource types*, e agrupa-as, juntamente com o prefixo do serviço, em um único arquivo, chamado *iam-actions-table.csv*. A Figura 9 exibe o resultado da geração da tabela de ações, contendo todos os serviços da AWS, suas ações correspondentes, além de informações sobre cada uma delas. Todos os arquivos gerados nesta etapa serão utilizados para a análise de políticas de permissão.

ServiceName	Alias	Permission	Action	Description	AccessLevel	Resource
ServiceName	Alias	Permission	Action	Description	AccessLevel	Resource
AWS Account Management	account	DeleteAlternateContact	account:DeleteAlternateContact	Grants permission to delete the alternate contacts for an account	Write	True
AWS Account Management	account	DisableRegion	account:DisableRegion	Grants permission to disable use of a Region	Write	False
AWS Account Management	account	EnableRegion	account:EnableRegion	Grants permission to enable use of a Region	Write	False
AWS Account Management	account	GetAlternateContact	account:GetAlternateContact	Grants permission to retrieve the alternate contacts for an account	Read	True
AWS Account Management	account	GetContactInformation	account:GetContactInformation	Grants permission to retrieve the primary contact information for an account	Read	True
AWS Account Management	account	ListRegions	account>ListRegions	Grants permission to list the available Regions	List	False
AWS Account Management	account	PutAlternateContact	account:PutAlternateContact	Grants permission to modify the alternate contacts for an account	Write	True
AWS Account Management	account	PutContactInformation	account:PutContactInformation	Grants permission to update the primary contact information for an account	Write	True
AWS Activate	activate	CreateForm	activate:CreateForm	Grants permission to submit an Activate application form	Write	False
AWS Activate	activate	GetAccountContact	activate:GetAccountContact	Grants permission to get the AWS account contact information	Read	False
AWS Activate	activate	GetContentInfo	activate:GetContentInfo	Grants permission to get Activate tech posts and offer information	Read	False
AWS Activate	activate	GetCosts	activate:GetCosts	Grants permission to get the AWS cost information	Read	False
AWS Activate	activate	GetCredits	activate:GetCredits	Grants permission to get the AWS credit information	Read	False
AWS Activate	activate	GetMemberInfo	activate:GetMemberInfo	Grants permission to get the Activate member information	Read	False
AWS Activate	activate	GetProgram	activate:GetProgram	Grants permission to get an Activate program	Read	False
AWS Activate	activate	PutMemberInfo	activate:PutMemberInfo	Grants permission to create or update the Activate member information	Write	False
Alexa for Business	alexa	ApproveSkill	alexa:ApproveSkill	Grants permission to associate a skill with the organization under the customer's AWS account	Write	False
Alexa for Business	alexa	AssociateContactWithAddressBook	alexa:AssociateContactWithAddressBook	Grants permission to associate a contact with a given address book	Write	True
Alexa for Business	alexa	AssociateDeviceWithNetworkProfile	alexa:AssociateDeviceWithNetworkProfile	Grants permission to associate a device with the specified network profile	Write	True
Alexa for Business	alexa	AssociateSkillGroupWithRoom	alexa:AssociateSkillGroupWithRoom	Grants permission to associate device with given room	Write	True
Alexa for Business	alexa	AssociateSkillGroupWithRoom	alexa:AssociateSkillGroupWithRoom	Grants permission to associate the skill group with given room	Write	True
Alexa for Business	alexa	AssociateSkillWithSkillGroup	alexa:AssociateSkillWithSkillGroup	Grants permission to associate a skill with a skill group	Write	True

Figura 9. Tabela de ações finalizada

Um fato importante a ser destacado é a contínua evolução dos serviços oferecidos pela AWS. Serviços já existentes lançam novas funcionalidades e novos serviços são lançados pela empresa a todo momento. Com isso, novas ações são constantemente disponibilizadas na documentação de ações da AWS [7]. Portanto, é essencial que a tabela de ações seja atualizada com frequência, para que as novas ações sejam inseridas na fonte de dados. Durante a execução deste trabalho, foram identificados mais de 16 novos serviços e aproximadamente 600 novas ações adicionadas na documentação da AWS.

5.3. Lista de ações críticas

A fim de completar a criação da base de dados, torna-se necessário gerar também a lista de ações críticas. Trata-se da lista de ações que possuem certo risco ao ser atribuída para algum usuário ou função. Tendo em vista que não existe uma automação para seleção dessas ações, as ações foram selecionadas manualmente, com base na tabela de ações gerada na seção anterior. Consequentemente, o trabalho de geração da lista envolve grande esforço manual do desenvolvedor, uma vez que é preciso avaliar as ações individualmente e julgá-las como críticas ou não.

O julgamento das ações quanto à sua criticidade foi realizado conforme os seguintes critérios:

- Possibilidade de visualizar, criar, modificar ou deletar informações sensíveis e confidenciais, como bancos de dados que contém dados pessoais (PII - *Personal Identifiable Information*), informações sobre transações bancárias, repositórios contendo código fonte de aplicações e qualquer outra informação categorizada como restrita;

- Possibilidade de criar, alterar ou deletar recursos e regras de configuração que possam causar impacto em aplicações e serviços;
- Possibilidade de visualizar, criar, alterar ou deletar credenciais de acesso, chaves de criptografia e configurações de usuários e funções, que permitam acesso não autorizado a servidores, APIs, aplicações, bancos de dados, entre outros.

Caso a ação se encaixe em algum dos critérios selecionados, ela é adicionada à lista de ações críticas. Ao todo, foram selecionadas 553 ações críticas, até o momento de desenvolvimento deste trabalho. No entanto, assim como a tabela de ações, a lista de ações críticas deve ser atualizada, conforme a evolução dos serviços e lançamento de novos serviços pela AWS. A ação manual para geração e atualização desta lista de ações críticas pode ser considerada a parte mais desafiadora do trabalho, por ser uma etapa que depende de um julgamento subjetivo do desenvolvedor. É importante ressaltar também que não houve o interesse em criar uma categorização das ações em níveis de criticidade. Tal implementação demandaria a reestruturação desta base de dados, além da adaptação da lógica do analisador de políticas, para poder manipular a lista de ações críticas corretamente.

5.4. Analisador de políticas

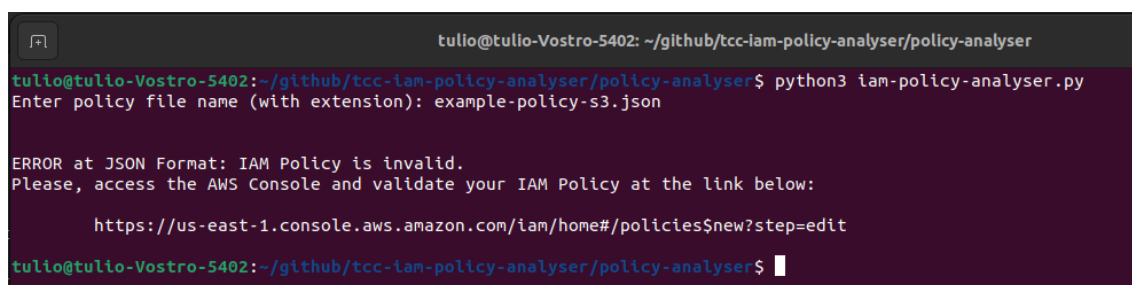
O analisador de políticas é a etapa em que se realiza a avaliação crítica das políticas de permissões, fazendo uso da base de dados, gerada anteriormente, para tal. Em resumo, o analisador recebe como entrada a política a ser analisada, em formato JSON, e entrega, como saída, a política analisada, as ações críticas identificadas na política, bem como suas descrições e *links* para se consultar a documentação oficial da AWS. O analisador de políticas não só avalia as ações presentes na política, mas também realiza várias validações durante a sua execução, como por exemplo, quanto à estrutura do documento JSON e os elementos que compõem uma política do IAM (*Statement*, *Effect*, *Action*, *Resource*, etc).

O analisador enumera os *Statements* e analisa todos eles, avaliando ação por ação. Para cada ação, a lista de ações críticas é consultada, a fim de verificar se a ação avaliada está presente na lista. Caso a ação que está sendo avaliada seja encontrada na lista de ações críticas, ela é incluída em uma lista de resultados, juntamente com informações que trazem detalhes sobre ela. As informações são coletadas da tabela de ações e dos demais arquivos gerados na sub-seção 5.2:

- Número do *Statement*;
- Nome da ação;
- Nível de acesso;
- Se a especificação de recurso se faz necessária ou não;
- Descrição da ação;
- *Link* para documentação oficial do serviço.

A lógica implementada no analisador de políticas inclui o uso de expressões regulares, para identificação de ações. Elas podem aparecer de várias formas, então o analisador está configurado para identificar essas expressões regulares e realizar o tratamento adequado. As maiores ocorrências são do uso do símbolo asterisco ("*"), que significa zero ou mais ocorrências de caracteres. O uso de expressões regulares pode tornar o resultado da análise muito amplo, uma vez que uma única ação seguida de asterisco pode corresponder a várias ações.

Caso o analisador encontre inconsistências na estrutura da política de permissões, será exibida uma mensagem de erro ao usuário. A mensagem de erro, como mostra a Figura 10, direciona o usuário para o simulador de políticas do IAM, que lhe mostrará as falhas encontradas na política e como corrigi-las. As falhas de inconsistência mais comuns ocorrem na formatação do arquivo em JSON (a falta de uma única vírgula pode gerar esta falha) e na falta de algum elemento essencial para uma política de IAM (como *Statement*, *Effect*, *Action*, *Resource*, etc). Para que os resultados sejam satisfatórios, o usuário precisará corrigir os erros da política antes de executar o analisador novamente.

A terminal window with a dark background and light-colored text. The prompt is 'tulio@tulio-Vostro-5402: ~/github/tcc-iam-policy-analyser/policy-analyser'. The user has run 'python3 iam-policy-analyser.py' and entered 'example-policy-s3.json'. The output shows an error: 'ERROR at JSON Format: IAM Policy is invalid. Please, access the AWS Console and validate your IAM Policy at the link below: https://us-east-1.console.aws.amazon.com/iam/home#/policies\$new?step=edit'. The prompt is now 'tulio@tulio-Vostro-5402:~/github/tcc-iam-policy-analyser/policy-analyser\$' with a cursor.

```
tulio@tulio-Vostro-5402: ~/github/tcc-iam-policy-analyser/policy-analyser
tulio@tulio-Vostro-5402:~/github/tcc-iam-policy-analyser/policy-analyser$ python3 iam-policy-analyser.py
Enter policy file name (with extension): example-policy-s3.json

ERROR at JSON Format: IAM Policy is invalid.
Please, access the AWS Console and validate your IAM Policy at the link below:

https://us-east-1.console.aws.amazon.com/iam/home#/policies$new?step=edit

tulio@tulio-Vostro-5402:~/github/tcc-iam-policy-analyser/policy-analyser$
```

Figura 10. Mensagem de erro

Posteriormente, na seção 6, são apresentados os testes executados para avaliar a eficiência e utilidade da ferramenta. Será possível encontrar a Figura 12, que exemplifica uma execução realizada com sucesso do analisador de políticas, onde a estrutura do arquivo JSON foi validada, os elementos básicos estavam presentes e o arquivo prosseguiu para análise.

5.5. Histórico de análises

A fim de mostrar os resultados com mais clareza, o analisador realiza o armazenamento de todas as suas políticas de permissão analisadas. Em uma pasta chamada *history*, os resultados das políticas analisadas são armazenados em sub-pastas, cujo nome é a marca de tempo em que aquela análise fora executada. O resultado exibe sempre três arquivos: uma cópia da política analisada e dois arquivos contendo o mesmo resultado da análise de ações, porém em formatos diferentes. Um arquivo está em formato CSV (caso o usuário queira visualizar o resultado em formato de planilha), e o outro em formato JSON (caso prefira visualizar em formato de texto, de forma organizada). A Figura 11 exibe os três arquivos gerados como resultado da análise da política de permissões.

Vale ressaltar que a consulta da documentação oficial da AWS é de extrema importância para a interpretação dos resultados das análises. Para cada ação identificada como crítica pelo analisador, é escrito no arquivo de resultados o número da instrução onde se encontra a ação, o nome da ação, seu nível de acesso, se aquela ação requer a especificação do recurso ou não, a descrição da ação e o *link* para a documentação oficial do serviço. A documentação apresenta detalhadamente a definição de cada ação, para cada serviço da AWS. Portanto, a documentação oficial da AWS é utilizada como complemento da ferramenta, para que todas as análises sejam examinadas com precisão.

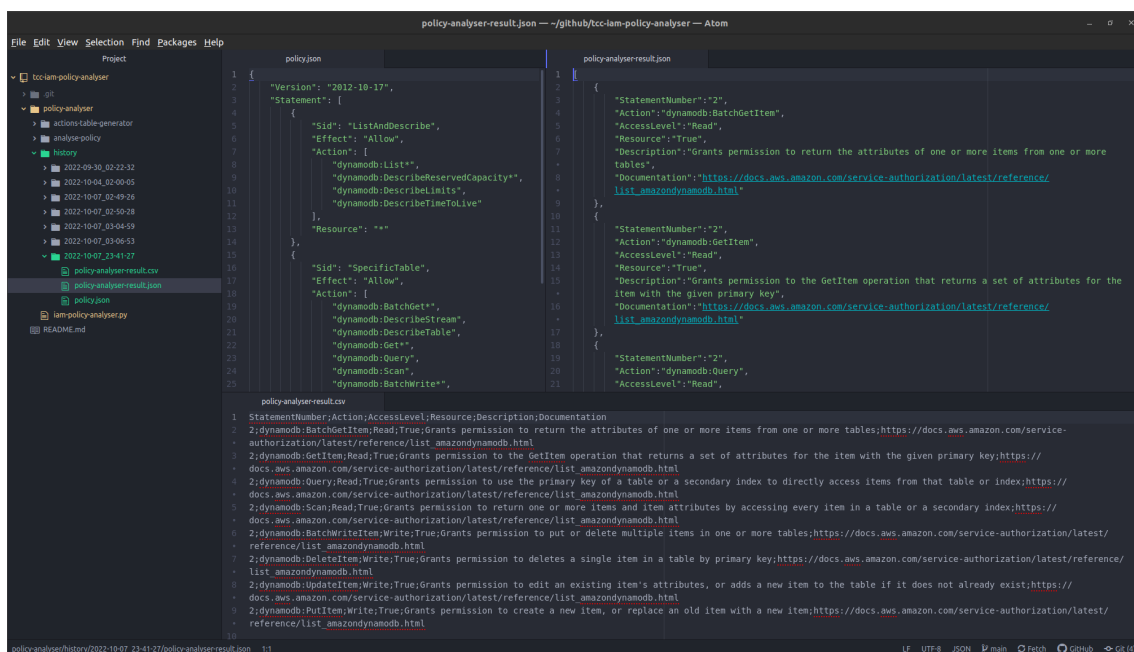


Figura 11. Histórico de resultados das análises

5.6. Disponibilização em repositório público

O código fonte do trabalho está disponível para acesso ao público, no repositório GitHub [30]. Todos os tópicos explicados nesta seção estão evidenciados com clareza, a fim de garantir organização de código. Os comentários, variáveis e as funções dentro do código do trabalho estão implementados fazendo uso a língua inglesa, com o objetivo de facilitar a interpretação de desenvolvedores de qualquer lugar do mundo, que tenham interesse em contribuir para o projeto.

A base de dados (tabela de ações e lista de ações críticas) já estará criada dentro do repositório, para facilitar o uso do analisador de políticas pelo usuário. Ele poderá utilizar a tabela de ações como fonte de consulta, a fim de formular sua política. Porém, é sugerido que o usuário realize a geração da tabela de ações antes de executar o analisador de políticas, para que a tabela esteja atualizada em relação aos serviços e novas funcionalidades lançados recentemente pela AWS.

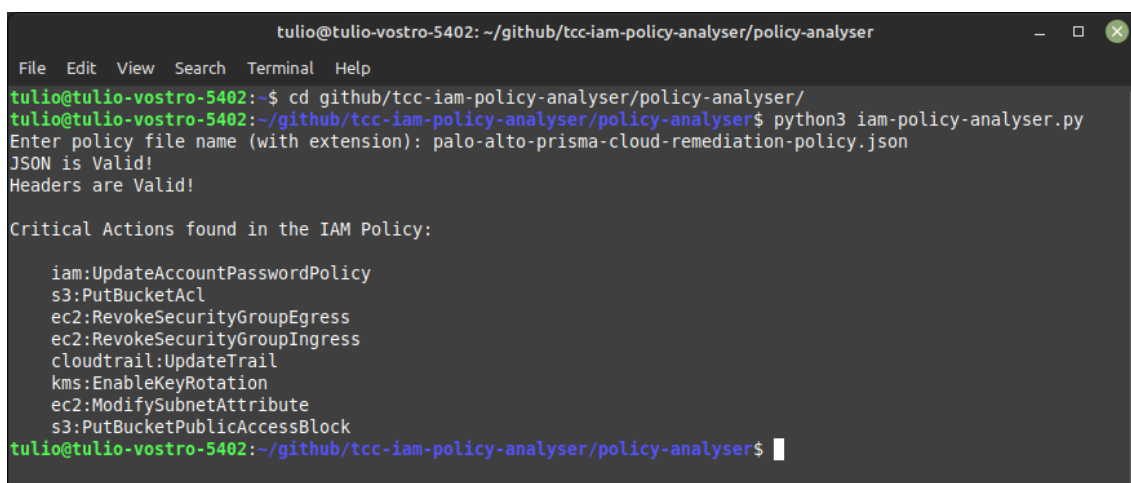
6. Testes e Resultados

Nesta seção, são exibidas demonstrações de uso da ferramenta, com modelos de políticas utilizadas comumente por aplicações, bem como os desafios encarados durante o desenvolvimento do trabalho. Também são mostrados casos de uso da ferramenta em cenários reais, quando uma empresa adquire um serviço, e precisa integrá-lo ao ambiente de nuvem que possui. Tem-se, dessa forma, uma visão geral da evolução do trabalho e sua aplicabilidade em situações de trabalho.

6.1. Testes com modelos de Políticas

Para que se execute a análise de uma política do IAM, é necessário clonar o repositório do GitHub para um diretório local no computador do usuário e seguir as instruções

contidas no arquivo *README.md* para instalar todos os componentes e bibliotecas necessárias para a perfeita execução do analisador de políticas. Em seguida, o usuário deve salvar a política a ser analisada, em formato JSON, no diretório */tcc-iam-policy-analyser/policy-analyser/analyse-policy* e, por fim, utilizando o terminal, navegar até o diretório */tcc-iam-policy-analyser/policy-analyser/* e executar o comando *python3 iam-policy-analyser.py*. Assim que o sistema estiver em execução, o usuário deve digitar o nome do arquivo em formato JSON que deseja analisar, que serve como entrada para o sistema. Dessa forma, o processo é iniciado, começando pelas validações de formatação, para garantir que a política é válida e está no formato aceito pelo AWS IAM. Em seguida, a análise da política é executada e armazenada no diretório de histórico de análises. A Figura 12 ilustra as etapas de validação e as ações detectadas, sendo mostradas em tempo de execução pela ferramenta.



```
tulio@tulio-vostro-5402: ~/github/tcc-iam-policy-analyser/policy-analyser
File Edit View Search Terminal Help
tulio@tulio-vostro-5402:~$ cd github/tcc-iam-policy-analyser/policy-analyser/
tulio@tulio-vostro-5402:~/github/tcc-iam-policy-analyser/policy-analyser$ python3 iam-policy-analyser.py
Enter policy file name (with extension): palo-alto-prisma-cloud-remediation-policy.json
JSON is Valid!
Headers are Valid!

Critical Actions found in the IAM Policy:

iam:UpdateAccountPasswordPolicy
s3:PutBucketAcl
ec2:RevokeSecurityGroupEgress
ec2:RevokeSecurityGroupIngress
cloudtrail:UpdateTrail
kms:EnableKeyRotation
ec2:ModifySubnetAttribute
s3:PutBucketPublicAccessBlock
tulio@tulio-vostro-5402:~/github/tcc-iam-policy-analyser/policy-analyser$
```

Figura 12. Demonstração de execução da ferramenta

A fim de testar o funcionamento do analisador de políticas, foram criadas duas políticas com base em dois serviços muito utilizados no mundo AWS, voltados para o armazenamento de dados, que são os serviços S3 e DynamoDB. O serviço S3 é um serviço de armazenamento de arquivos, cujo foco é garantir a durabilidade, a disponibilidade e a segurança do arquivo armazenado. Os recursos usados pelo serviço S3 para armazenamento de arquivos são chamados *buckets*. Já o serviço DynamoDB é um serviço de banco de dados NoSQL do tipo chave-valor, que segue o conceito de banco de dados *serverless* (sem servidor). Ele é projetado para suportar aplicações de alta performance, com respostas a nível de milissegundos.

A política para o serviço DynamoDB, mostrada na Figura 13, evidencia um caso de uso do serviço DynamoDB. A política permite a visualização de todas as tabelas DynamoDB existentes, bem como seus atributos e especificações, além do acesso a uma tabela de nome *MyTable*, que dá a liberdade ao usuário ou à função do IAM, de realizar a criação da tabela e fazer a manipulação total de seus dados, permitindo leituras, escritas, deleções de itens, consultas e escaneamentos dentro daquela tabela em específico.

```

example-policy-dynamodb.json
example-policy-s3.json

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "ListAndDescribe",
6       "Effect": "Allow",
7       "Action": [
8         "dynamodb:List*",
9         "dynamodb:DescribeReservedCapacity*",
10        "dynamodb:DescribeLimits",
11        "dynamodb:DescribeTimeToLive",
12        "dynamodb:DescribeStream",
13        "dynamodb:DescribeTable"
14      ],
15      "Resource": "*"
16    },
17    {
18      "Sid": "ManipulateTable",
19      "Effect": "Allow",
20      "Action": [
21        "dynamodb:BatchGet*",
22        "dynamodb:Get*",
23        "dynamodb:Query",
24        "dynamodb:Scan",
25        "dynamodb:BatchWrite*",
26        "dynamodb:CreateTable",
27        "dynamodb>Delete*",
28        "dynamodb:Update*",
29        "dynamodb:PutItem"
30      ],
31      "Resource": "arn:aws:dynamodb:*:*:table/MyTable"
32    }
33  ]
34 }

```

Figura 13. Política para o serviço DynamoDB

O resultado da análise da política para o serviço DynamoDB detectou ao todo 8 ações críticas, como mostrado anteriormente, na Figura 11. Todas as ações identificadas pelo analisador envolvem, como previsto, o acesso, seja de leitura ou escrita, aos dados contidos no recurso. Tais ações devem ser selecionadas com cautela no momento de elaboração da política, uma vez que as informações contidas na tabela podem ser sensíveis, confidenciais.

A política para o serviço S3, já mostrada anteriormente na Figura 4, exemplifica também uma situação de manipulação de arquivos. É possível identificar pela instrução número 1, que a política permite a leitura das configurações de todos os *buckets*. Nas instruções números 2 e 3, há a especificação do *bucket* a ser acessado, que se chama *bucket-name*, portanto o acesso fica restrito àquele *bucket* somente. No entanto, na instrução 3, há o uso de expressão regular ao se especificar a ação, o que é um ponto de atenção. Utiliza-se 2 sinais de asterisco ("*") entre a palavra *Object*, o que representa todas as ações que possuem o nome *Object* em seu nome, seja como prefixo, sufixo ou no meio do nome da ação.

O resultado da análise da política, evidenciado na Figura 14, revelou 16 ações críticas contidas na política. Todas as ações detectadas como críticas foram provenientes

da instrução 3, em que se encontra o uso de dois sinais de expressão regular. Isso mostra como o uso de expressões regulares pode ser perigoso, pois várias outras ações podem ser derivadas de uma única ação com expressão regular.

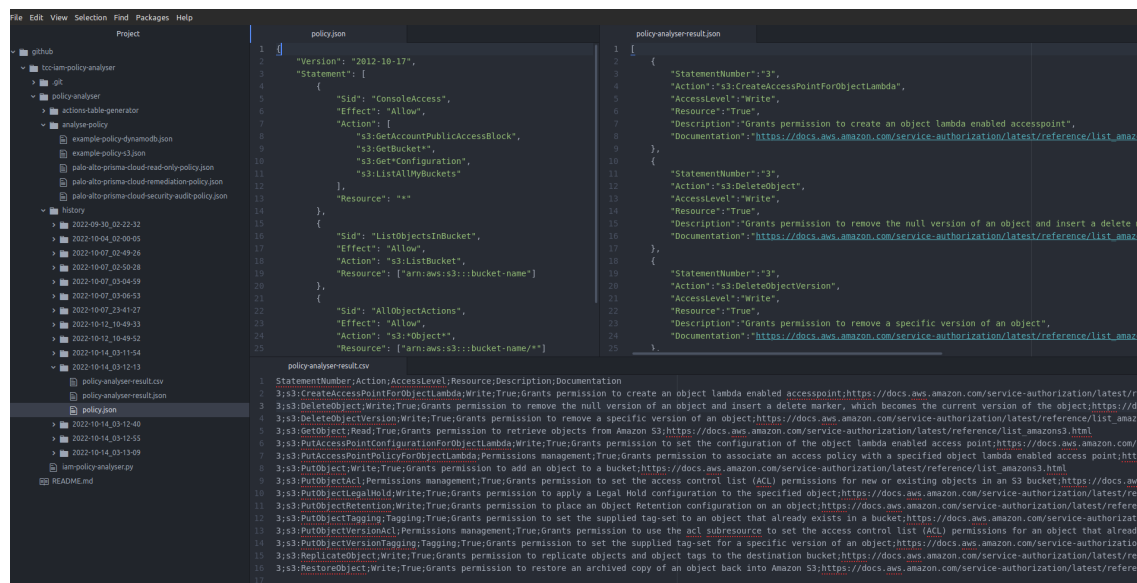


Figura 14. Resultado da análise da política para o serviço S3

6.2. Casos de Uso

O analisador de políticas tem potencial para detectar possíveis tentativas de acesso a dados sensíveis no mundo da tecnologia, bem como acesso a ações de alto risco. Serviços de tecnologia, consultores e fornecedores precisam de acesso ao ambiente de nuvem de uma empresa, para que certa atividade seja feita e, para isso, se faz necessário o uso de um usuário ou uma função do IAM com permissões para realizá-la. No entanto, tais solicitações de acesso podem incluir algumas ações críticas, cuja liberação se tornaria um risco para a empresa, uma vez que o serviço ou pessoa teria acesso a dados sensíveis de clientes da empresa, informações de contratos ou permissão para realizar alterações de configuração em recursos críticos para a instituição.

O serviço *Prisma Cloud* [9], fornecido pela empresa *Palo Alto Networks*, é útil para exemplificar a situação. Esta ferramenta é mundialmente conhecida por muitas empresas, por sua proposta de garantir a segurança da infraestrutura de nuvem do cliente. Ela tem a capacidade de detectar e corrigir falhas de configuração, anomalias e dados expostos publicamente. Tendo isso em vista, a documentação[31] do *Prisma Cloud* fornece ao cliente um modelo para se criar uma função do IAM e adicionar várias políticas de permissão a ela, para que o serviço tenha acesso à sua infraestrutura de nuvem e possa realizar o escaneamento do ambiente[32] e possível correção de recursos configurados inadequadamente. Este modelo foi analisado pela ferramenta desenvolvida neste trabalho, a fim de testar sua eficiência. O resultado expôs o total de 50 ações críticas, presentes na política de acesso sugerida pelo serviço. As ações solicitadas pelo fabricante envolvem o envio de comandos a servidores, a criação e deleção de regras de rede, acessos a arquivos e credenciais criptografados, visualização do código fonte de aplicações, entre outros.

A solução *Commvault*[10] também serve como exemplo de aplicação do analisador de políticas no cenário corporativo. Seu principal objetivo é gerenciar os *backups e restores* de todo e qualquer tipo de dado ou aplicação, tanto no ambiente da nuvem, quanto no ambiente *on-premises*. Ou seja, o *Commvault* é responsável por realizar cópias de segurança de bancos de dados e de servidores inteiros, a fim de armazená-las como forma de contingência em cenários de recuperação de desastres. Embora sua documentação[33] seja bastante completa, as permissões por ela sugeridas, para que a solução funcione corretamente, se mostram abrangentes. O analisador de políticas do IAM foi capaz de detectar 29 (vinte e nove) ações críticas presentes nas políticas requeridas pela documentação. Dentre elas, ações que permitem a alteração de itens em bancos de dados, a criação e deleção de máquinas virtuais e bancos de dados, a configuração de rede de todo e qualquer recurso, a visualização de credenciais de autenticação, a atribuição de uma IAM Role com permissões administrativas a qualquer máquina virtual, entre outras permissões. Para se ter uma noção de risco, com tais privilégios, um colaborador mal intencionado, que possua acesso ao serviço *Commvault*, seria capaz de iniciar uma máquina virtual com sistema operacional *Linux*, liberado para ser acessado através da internet, e atribuir a este servidor uma IAM Role com permissão administrativa. Então, o atacante, por meio de um computador externo, utilizaria o IP público do servidor para acessá-lo, e assim, teria permissão total em toda a conta AWS onde está o servidor. Dessa forma, ele poderia acessar e roubar qualquer informação da conta AWS que seja de seu interesse, como bancos de dados, credenciais e arquivos sigilosos, além de também causar indisponibilidades nos serviços críticos da companhia.

Esses exemplos mostram a importância de se analisar uma política de acesso antes de liberá-la para um serviço ou pessoa. Garantir o acesso a ações críticas pode implicar no roubo de credenciais, vazamentos de informações confidenciais e, inclusive, extorsões por recompensas em dinheiro. Consequentemente, a empresa, além de perder sua credibilidade, pode vir a pagar multas ou, em casos piores, até vir à falência. Por esses motivos, é inquestionável a necessidade de se realizar uma análise prévia de permissões, para que tais situações possam ser evitadas. Um profissional de tecnologia, ao utilizar a ferramenta, terá capacidade de criar reforços de segurança em seu ambiente, para que o serviço ou o fornecedor não tenha acesso aos recursos que são considerados críticos. Ou, em casos mais severos, poderá seguir o conceito de menor privilégio e, antes da criação da função do IAM para o serviço, remover da política, aquelas ações críticas que representam um risco operacional para a organização.

6.3. Desafios Durante o Desenvolvimento

Os maiores desafios da implementação do analisador de políticas do IAM foram a implementação e manutenção da base de dados. A base de dados foi extraída diretamente da documentação oficial da AWS [7] e, por esse motivo, sua implementação envolveu a manipulação de páginas HTML e diversos redirecionamentos, para se extrair apenas o conteúdo necessário das páginas de documentação.

Como complemento, devido à constante evolução da plataforma AWS e sua gama de serviços, a tabela de ações precisa ser constantemente atualizada. Consequentemente, a lista de ações críticas também é afetada pelo mesmo motivo. Logo, à medida que novos serviços são disponibilizados e novas ações são criadas, a tabela de ações precisa ser sincronizada com a documentação, e as novas ações devem ser selecionadas manualmente

quanto à sua criticidade. Sendo assim, a manutenção da base de dados torna-se o maior desafio, a longo prazo, para o bom funcionamento deste trabalho. De fato, esta limitação existe devido à limitação do escopo deste trabalho, mas através de melhorias futuras na implementação da ferramenta, pode-se alcançar melhor eficiência na atualização da base de dados, fazendo-se uso de automações.

A etapa de tratamento de expressões regulares também foi desafiadora. Foram utilizadas bibliotecas específicas da linguagem *python* para realizar a formatação das ações, antes que elas fossem direcionadas para avaliação. A lógica para formatação das ações levou tempo considerável para ser implementada, uma vez que ela deveria se aplicar a todos os cenários possíveis.

7. Conclusão e Trabalhos Futuros

O objetivo deste trabalho foi apresentar uma solução que facilite a identificação de permissões consideradas não autorizadas, críticas e de alto risco para o negócio de uma empresa. Essa implementação foi proveniente da problemática envolvendo a concessão de permissões excessivas e, em alguns momentos, desnecessárias, responsável por vazamentos de dados, fraudes bancárias, ataques cibernéticos, entre outros cenários. Assim sendo, além de apresentar a ferramenta em questão, tornou-se o objetivo secundário deste trabalho promover a importância do uso consciente do serviço AWS IAM para realizar a gestão de acessos em plataformas de computação em nuvem.

Os resultados indicaram que a identificação prévia de tais acessos permite que o profissional de Segurança da Informação execute a ação de bloqueio antes mesmo que o serviço ou usuário tenha o seu acesso concedido. Tal ação prova a aplicabilidade da solução na prevenção contra vazamentos de informações confidenciais e possíveis multas relacionadas a esse fato. Embora haja planos de implementação futuros para a ferramenta (que serão apresentados na próxima seção), a versão publicada mostra-se capaz de entregar resultados precisos e de fácil interpretação pelo usuário.

De forma sucinta, a solução mostra-se relevante na medida em que contribui, em termos científicos e técnicos, para o setor de Segurança da Informação. Em relação à contribuição científica, a solução poderá ser incorporada à plataforma AWS, tornando-se uma funcionalidade do serviço AWS IAM. Assim, como o módulo *IAM Access Analyser* [26], a solução deste trabalho poderia se tornar um novo módulo do serviço IAM, capaz de classificar, em tempo real e por níveis de criticidade, as políticas do IAM, evidenciando, assim, aquelas que possuem altos privilégios, capazes de causar algum impacto ao negócio. Logo, a implementação incentiva a pesquisa voltada para o setor de gestão de acessos no ambiente da nuvem, com o objetivo de trazer maior visibilidade para o profissional que atua no setor. Já em relação à contribuição técnica, seu valor é notado por sua aplicação em cenários reais, onde se deseja avaliar a criticidade de uma política de permissões e evitar um possível impacto para a organização.

7.1. Futuras implementações

Deseja-se implementar melhorias neste projeto, com o objetivo de facilitar a experiência do usuário ao fazer uso da ferramenta, tornando o seu uso mais intuitivo e automatizado. Dessa forma, pretende-se transformar este projeto em um *website*, com hospedagem na própria plataforma AWS. A arquitetura planejada para esta implementação

futura é detalhada na Figura 15 e consiste no uso do serviço DynamoDB, para armazenamento da base de dados (tabela de ações e lista de ações críticas), do serviço S3, para armazenamento do histórico de análises, e do serviço AWS Lambda, para armazenamento dos códigos fonte da aplicação (mais especificamente, para o gerador da base de dados e o analisador de políticas). Os códigos da aplicação ficarão armazenados em funções do serviço Lambda, que farão uso de funções do IAM, com permissões semelhantes às demonstradas neste trabalho, para se ter acesso às tabelas do serviço *DynamoDB* e aos *buckets* do serviço S3.

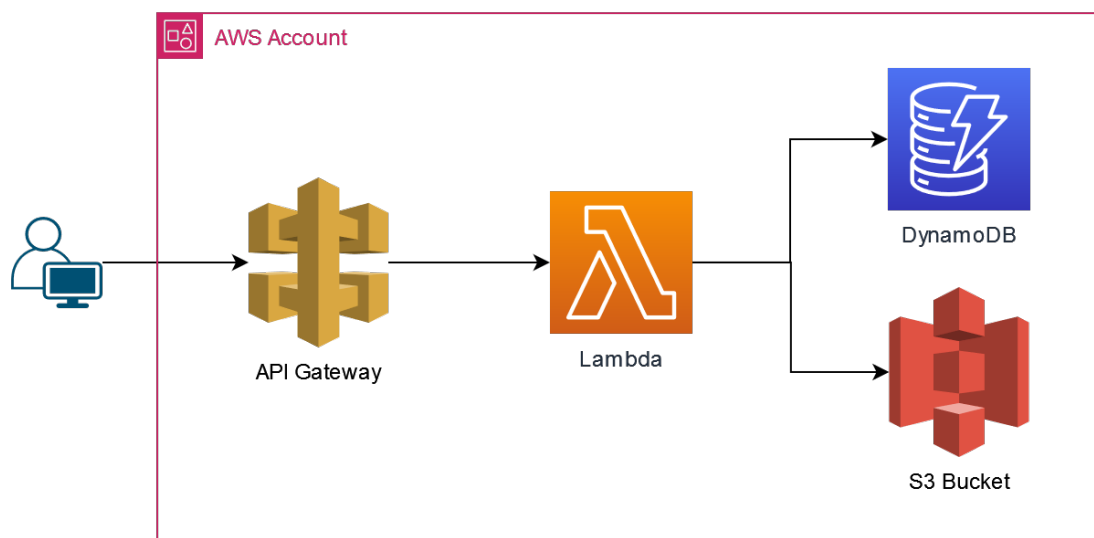


Figura 15. Arquitetura de Trabalhos Futuros

Para sanar o problema de manutenção constante da base de dados, será implementada uma rotina de atualização da base de dados, fazendo uso também do serviço AWS Lambda. A rotina poderá ser customizada, portanto será possível definir execuções diárias, para que a tabela de ações se mantenha atualizada, com os mais novos serviços e novidades. Todavia, a atualização da lista de ações críticas ainda permanecerá sendo realizada manualmente, uma vez que a AWS não fornece esse tipo de informação em sua documentação. Em contrapartida, planeja-se implementar, na interface web, uma área onde o usuário da ferramenta possa inserir sugestões de ações críticas, para que sejam avaliadas posteriormente pelo desenvolvedor da aplicação.

Para o desenvolvimento do *frontend*, pretende-se utilizar o framework *Django*. A interface permitirá que o usuário consulte a tabela de ações a qualquer momento, crie filtros, pesquise por palavras-chave e até realize *download* da tabela em formato de planilha, caso queira. Haverá um campo para que o usuário insira a política que será analisada e o resultado da análise aparecerá na tela para o usuário, ao lado da política inserida por ele. Será possível, também, ter acesso ao histórico de análises executadas, ordenadas por tempo. As análises que tiverem sido executadas há mais de 30 dias, serão apagadas do histórico, para que o custo de armazenamento seja controlado com eficiência.

Referências

- [1] Manoel Veras. *Computação em Nuvem - Nova Arquitetura de TI*. 1st. Brasport, 2015.
- [2] Ishaq Azhar Mohammed. “CLOUD IDENTITY AND ACCESS MANAGEMENT – A MODEL PROPOSAL”. Em: *INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY [IJIERT]* 06.10 (2019). DOI: https://www.researchgate.net/profile/Ishaq-Azhar-Mohammed/publication/353887567_CLOUD_IDENTITY_AND_ACCESS_MANAGEMENT_-_A_MODEL_PROPOSAL_/links/61169d070c2bfa282a41f553/CLOUD-IDENTITY-AND-ACCESS-MANAGEMENT-A-MODEL-PROPOSAL.pdf.
- [3] Swati Mehta e Jitender Saini. “Tracking Down Unauthorized Access by Users in Cloud”. Em: *International Research Journal of Engineering and Technology (IRJET)* 03.08 (2016). DOI: <https://www.irjet.net/archives/V3/i8/IRJET-V3I894.pdf>.
- [4] Henrique Bueno. *AWS na prática — Modelo de responsabilidade compartilhada*. URL: <https://blog.estabilis.com.br/aws-na-pratica-modelo-de-responsabilidade-compartilhada/>. Acesso em 23 Nov. 2021.
- [5] Emerson Alecrim. *Dados sigilosos de 10 mil brasileiros vazam de empresa financeira*. URL: <https://tecnoblog.net/418460/vazamento-dados-10-mil-clientes-prisma-corretora/>. (acessado: 23.11.2021).
- [6] Guy Fawkes. *Report: 10,000s of Brazilians Exposed to Fraud in Massive Data Breach*. URL: <https://www.vpnmentor.com/blog/report-prisma-promotora-leak/>. (acessado: 23.11.2021).
- [7] AWS Documentation. *Actions, resources, and condition keys for AWS services*. URL: https://docs.aws.amazon.com/service-authorization/latest/reference/reference_policies_actions-resources-contextkeys.html. Acesso em 01 Fev. 2022.
- [8] AWS Documentation. *O que é a computação em nuvem?* URL: <https://aws.amazon.com/pt/what-is-cloud-computing/>. Acesso em 9 Fev. 2022.
- [9] Palo Alto Networks. *Prisma Cloud*. URL: <https://docs.paloaltonetworks.com/prisma/prisma-cloud>. Acesso em 31 Out. 2022.
- [10] Commvault. *Enterprise Data Protection Solutions - SaaS Software & Cloud Backup*. URL: <https://www.commvault.com>. Acesso em 15 Nov. 2022.
- [11] Google. *Apps Para Empresas e Ferramentas Colaborativas — Google Workspace*. URL: <https://workspace.google.com/intl/pt-BR>. Acesso em 21 Nov. 2022.
- [12] Microsoft. *Microsoft 365 - Assinatura Para Aplicativos do Office — Microsoft 365*. URL: <https://www.microsoft.com/pt-br/microsoft-365>. Acesso em 21 Nov. 2022.
- [13] AWS Documentation. *O que é o IAM?* URL: https://docs.aws.amazon.com/pt_br/IAM/latest/UserGuide/introduction.html. Acesso em 9 Fev. 2022.
- [14] AWS Documentation. *Usuários do IAM*. URL: https://docs.aws.amazon.com/pt_br/IAM/latest/UserGuide/id_users.html. Acesso em 9 Fev. 2022.
- [15] AWS Documentation. *Uso de autenticação multifator (MFA) na AWS*. URL: https://docs.aws.amazon.com/pt_br/IAM/latest/UserGuide/id_credentials_mfa.html. Acesso em 9 Fev. 2022.
- [16] AWS Documentation. *Grupos de usuários do IAM*. URL: https://docs.aws.amazon.com/pt_br/IAM/latest/UserGuide/id_groups.html. Acesso em 9 Fev. 2022.
- [17] AWS Documentation. *Funções do IAM*. URL: https://docs.aws.amazon.com/pt_br/IAM/latest/UserGuide/id_roles.html. Acesso em 9 Fev. 2022.

- [18] AWS Documentation. *Termos e conceitos das funções*. URL: https://docs.aws.amazon.com/pt_br/IAM/latest/UserGuide/id_roles_terms-and-concepts.html. Acesso em 14 Dez. 2022.
- [19] AWS Security Blog. *Now Create and Manage AWS IAM Roles More Easily with the Updated IAM Console*. URL: <https://aws.amazon.com/pt/blogs/security/now-create-and-manage-aws-iam-roles-more-easily-with-the-updated-iam-console/>. Acesso em 15 Dez. 2022.
- [20] AWS Security Blog. *How to use trust policies with IAM roles*. URL: <https://aws.amazon.com/pt/blogs/security/how-to-use-trust-policies-with-iam-roles>. Acesso em 15 Dez. 2022.
- [21] AWS Documentation. *Políticas e permissões no IAM*. URL: https://docs.aws.amazon.com/pt_br/IAM/latest/UserGuide/access_policies.html. Acesso em 9 Fev. 2022.
- [22] AWS Documentation. *Gramática da linguagem das políticas de JSON do IAM*. URL: https://docs.aws.amazon.com/pt_br/IAM/latest/UserGuide/reference_policies_grammar.html. Acesso em 14 Out. 2022.
- [23] AWS Documentation. *Simulador de políticas do IAM*. URL: [https://us-east-1.console.aws.amazon.com/iam/home#/policies%5C\\$new?step=edit](https://us-east-1.console.aws.amazon.com/iam/home#/policies%5C$new?step=edit). Acesso em 24 Out. 2022.
- [24] AWS Documentation. *Gerador de políticas do IAM*. URL: <https://awspolicygen.s3.amazonaws.com/policygen.html>. Acesso em 24 Out. 2022.
- [25] S. Naveen Kumar e K. Nirmala. “Cloud security: to prevent unauthorized access using an efficient key management authentication algorithm”. Em: *International Journal of Engineering & Technology (IJET)* 07.1.1 (2018). DOI: <https://www.sciencepubco.com/index.php/ijet/article/view/10787/3969>.
- [26] AWS Documentation. *Usar o AWS Identity and Access Management Access Analyzer*. URL: https://docs.aws.amazon.com/pt_br/IAM/latest/UserGuide/what-is-access-analyzer.html. Acesso em 22 Nov. 2022.
- [27] AWS Documentation. *Referência de verificação de política do Access Analyzer*. URL: https://docs.aws.amazon.com/pt_br/IAM/latest/UserGuide/access-analyzer-reference-policy-checks.html. Acesso em 22 Nov. 2022.
- [28] Amazon Web Services. *Demo: Use IAM Access Analyzer Policy Validation to Set Secure and Functional Policies*. URL: <https://www.youtube.com/watch?v=XhOOOfNfOkS4>. Acesso em 22 Nov. 2022.
- [29] Amazon Web Services. *Registrar em log chamadas de API do IAM Access Analyzer com o AWS CloudTrail*. URL: https://docs.aws.amazon.com/pt_br/IAM/latest/UserGuide/logging-using-cloudtrail.html. Acesso em 15 Dez. 2022.
- [30] Tulio Bittar. *Código fonte do projeto de TCC IAM Policy Analyser*. URL: <https://github.com/TulioBittar/tcc-iam-policy-analyser>. Acesso em 05 Out. 2022.
- [31] Palo Alto Networks. *Add an AWS Cloud Account on Prisma Cloud*. URL: <https://docs.paloaltonetworks.com/prisma/prisma-cloud/prisma-cloud-admin/connect-your-cloud-platform-to-prisma-cloud/onboard-your-aws-account/add-aws-cloud-account-to-prisma-cloud>. Acesso em 14 Out. 2022.
- [32] Palo Alto Networks. *Read-Write (Limited)*. URL: <https://redlock-public.s3.amazonaws.com/cft/rl-dlp-read-and-write.template>. Acesso em 31 Out. 2022.
- [33] Commvault. *Amazon Web Services User Permissions for Backups and Restores*. URL: https://documentation.commvault.com/disaster_recovery/30960_amazon_

[web_services_user_permissions_for_backups_and_restores.html](#). Acesso em 15 Nov. 2022.