Mapeamentos Sintático e Semântico em Federações de Políticas de Autorização para Nuvens Heterogêneas

Ioram S. Sette[1], Marcus R. Xavier[2], Carlos A. G. Ferraz[2], David W. Chadwick[3]

[1]Centro de Estudos e Sistemas Avançados do Recife (CESAR)

[2]Centro de Informática - Universidade Federal de Pernambuco (UFPE)

[3]University of Kent at Canterbury (UKC)

ioram.sette@cesar.org.br, {mrxl,cagf}@cin.ufpe.br, d.w.chadwick@kent.ac.uk

Resumo. *Os principais provedores de computação em nuvem, como Amazon, Google e Microsoft, utilizam tecnologias proprietárias para oferecer seus serviços. Usuários têm o desejo de usar múltiplos provedores para aumentar a disponibilidade de seus serviços, reduzir custos, não depender de um único fornecedor, dentre outros motivos. Federações de Políticas de Autorização (APFs) permitem a definição de uma política de controle de acesso única que pode ser aplicada em múltiplos provedores heterogêneos. Para isso, essas políticas, descritas na Forma Normal Disjuntiva (DNF) e semântica definida por uma ontologia, devem ser mapeadas para o contexto local de cada usuário e nuvem. Este trabalho discute os processos de mapeamento sintático e semântico das políticas e a métrica Nível de Equivalência Semântica (LSE) utilizada para medir sua efetividade.*

Abstract. *Leading Cloud Service Providers (CSPs), like Amazon, Google and Microsoft, use proprietary technologies to offer services to their customers. Users desire to use multiple CSPs to increase the availability of their services, to save money, and to avoid vendor lock-in, among other reasons. Authorisation Policy Federations (APFs) allow a single access control policy to be applied across multiple heterogeneous providers. Global policies defined in Disjunctive Normal Form (DNF) and ontology terms are mapped to local tenant and cloud specific formats. This paper discusses the semantic and syntactic policy mapping processes, and the Level of Semantic Equivalence (LSE) metric used for measuring their effectiveness.*

1. Introdução

O uso de computação em nuvens se consolida cada vez mais, mundialmente e também no Brasil. Estudo recente mostra uma queda no investimento em *data centers* e nuvens privadas e um aumento em nuvens públicas, híbridas e ambientes multi-nuvem, devido principalmente ao menor custo de propriedade e maior segurança [TI inside online 2018]. Dentre os principais motivos para adoção de ambientes multi-nuvem estão o maior alcance e disponibilidade dos recursos replicados em *data centers* em nível global, obter o menor custo devido à concorrência, e evitar o *vendor lock-in* [Opara-Martins et al. 2016], gerado pela dependência de implementação de tecnologia de um único provedor de nuvem [Toosi et al. 2014]. Mas, para isso, os usuários precisam lidar com ambientes heterogêneos, pois grandes provedores de nuvem, como Amazon, Google e Microsoft, utilizam seus padrões e protocolos proprietários.

A segurança dos dados é uma das maiores preocupações das empresas para adoção de nuvens públicas, uma vez que as informações passam a ser armazenadas em ambientes terceirizados. Mecanismos de segurança, como controle de acesso, fornecidos pelos principais provedores e plataformas de nuvem, aumentaram a confiabilidade destes ambientes por parte dos usuários. No entanto, em ambientes multi-nuvem heterogêneos, gerenciar estes diversos mecanismos nas diferentes nuvens torna-se um problema.

Este trabalho discute soluções para que usuários de nuvens heterogêneas de Infraestrutura como Serviço – *Infrastructure as a Service (IaaS)* tenham uma experiência de controle de acesso homogênea e, consequentemente, mais simples e fácil de ser trabalhada. Mecanismos de controle de acesso englobam a autenticação e autorização dos usuários. A autenticação visa a garantir que o usuário é quem ele alega ser. A autorização decide se a ação solicitada pelo usuário a um recurso será permitida ou negada, baseada em regras e políticas definidas. Esta pesquisa apresenta uma solução para controle de acesso homogêneo em ambientes multi-nuvem heterogêneos, chamada de Federação de Políticas de Autorização - Authorisation Policy Federation (APF), onde uma política de autorização global é definida e administrada de forma centralizada e concretizada em políticas nos formatos locais das plataformas de nuvem existentes. A tradução das políticas é feita através dos mapeamentos semântico e sintático, principal contribuição deste artigo.

O restante deste texto está estruturado da seguinte forma: a seção 2 discute o estado da arte referente a controle de acesso e ambientes multi-nuvem heterogêneos. A solução APF e seus principais módulos são apresentados na seção 3. Na seção 4, exemplos de uso da solução, detalhes de implementação e resultados obtidos são discutidos. Por fim, as conclusões são apresentadas na seção 5.

2. Estado da Arte

Usuários que usam diferentes provedores de serviço de nuvem encontram muitos entraves relacionados à heterogeneidade dos mesmos. Um exemplo é o gerenciamento de políticas de controle de acesso em diferentes formatos e padrões que se torna uma tarefa árdua para os administradores de serviços em nuvem.

2.1. Multi-nuvens Heterogêneas

O uso de serviços em múltiplas nuvens evita a dependência de um único fornecedor, o chamado *Vendor Lock-in*, e aumenta a confiabilidade e continuidade dos serviços oferecidos [Yousif 2017], entre outras razões. Um exemplo real é a empresa Netflix, que utiliza a Amazon Web Services (AWS) para transcodificação de filmes digitais e para seu engenho de recomendações, mas possui uma cópia de segurança de todos os seus dados na nuvem da Google, precavendo-se de uma indisponibilidade da AWS [Zeck and Bouroudjian 2017].

Um dos principais desafios para o uso de ambientes multi-nuvem é a heterogeneidade. Cada provedor define seus próprios formatos e protocolos, dificultando a interoperabilidade com outros serviços similares, com o objetivo de fidelizar os usuários em suas plataformas. *Middleware* para ambientes multi-nuvem heterogêneos têm o desafio de tornar a experiência de uso e gerenciamento destes ambientes transparente e homogênea, tanto para usuários quanto para desenvolvedores. São tarefas destes serviços gerenciar

recursos de diferentes domínios administrativos, gerenciar a localização dos dados e sua latência, implementar mecanismos de autenticação e autorização, analisar e combinar bilhetagem em diversos formatos e níveis de serviço, dentre outras [Bittencourt et al. 2017].

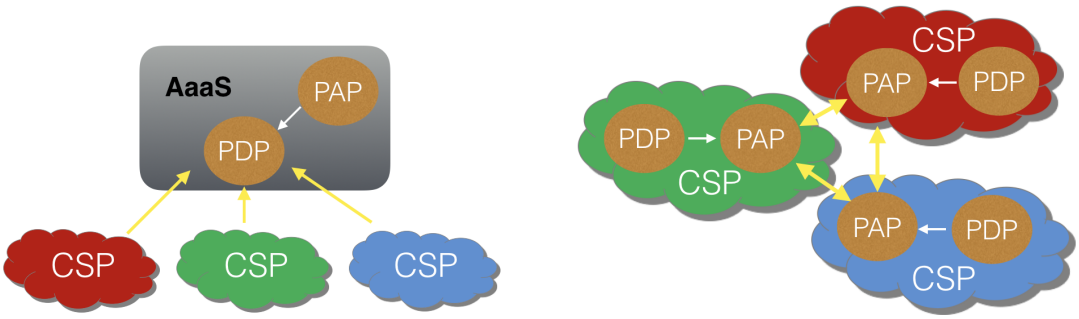
2.2. Controle de Acesso em Multi-nuvens Heterogêneas

Mecanismos de controle de acesso são responsáveis pela autenticação e autorização dos usuários em sistemas. Num ambiente multi-nuvem heterogêneo, o usuário autentica-se em cada um dos provedores, que possuem atributos diferentes sobre ele. Diferentes mecanismos de autorização com políticas divergentes trazem uma experiência desuniforme.

Para uma melhor experiência relacionada à autenticação, o usuário deve autenticar-se uma única vez e obter acesso a múltiplas nuvens. Além disso, é desejável que todos os serviços de nuvem recebam os mesmos atributos referentes aos usuários, identificando-os de uma forma homogênea. Federações de identidade atendem a estes requisitos através do mecanismo de *Single Sign-On (SSO)* [Chadwick et al. 2014] [Sette 2016].

Uma experiência uniforme de autorização deve garantir que uma única política de segurança seja utilizada em todas as contas dos usuários nas diversas nuvens. Na literatura, duas abordagens para autorização em ambientes multi-nuvem foram encontradas, a centralizada e a distribuída.

Na abordagem centralizada, ilustrada na Figura 1(a), um serviço de autorização externo às nuvens é utilizado de forma análoga aos provedores de identidade numa federação de identidades [Tang et al. 2013] [Bernabe et al. 2014]. O serviço de autorização é composto por um ponto de administração de políticas – *Policy Administration Point (PAP)* e um ponto de decisão de políticas – *Policy Decision Point (PDP)* [Moore 2003] centralizados. Este modelo, também referenciado por autorização como serviço – *Authorisation as a Service (AaaS)*, tem pontos positivos como possuir uma política comum única, administrada num ponto central, aplicada a serviços heterogêneos. Porém, o PDP centralizado é um ponto único de falhas e pode apresentar baixo desempenho devido a latência de rede, uma vez que o serviço encontra-se externo às nuvens. Um baixo desempenho do PDP preocupa principalmente devido à alta quantidade de consultas realizadas ao mesmo - pelo menos uma a cada requisição de acesso.



(a) Abordagem Centralizada (b) Abordagem Distribuída

Figura 1. Abordagens centralizada (a) e distribuída (b) para autorização em ambientes multi-nuvem

No modelo distribuído, os engenhos de autorização (PAP e PDP) de cada nuvem sincronizam as políticas, conforme mostra a Figura 1(b). As soluções distribuídas estudadas [Ngo et al. 2016] [Almutairi et al. 2012] também utilizam uma política comum, porém administrada de forma descentralizada, podendo gerar conflitos caso sejam alteradas simultaneamente. Este modelo não apresenta ponto único de falhas e possui bom desempenho, uma vez que as regras são validadas localmente nas nuvens. Os sistemas estudados, nas duas abordagens, necessitam ser integradas aos PDPs das nuvens, o que é uma restrição forte quando falamos de grandes provedores de IaaS.

A solução proposta neste trabalho apresenta uma abordagem mista, onde uma política comum única é definida e administrada através de um PAP centralizado, e traduzida para os formatos e semânticas locais de cada provedor de nuvem, que utilizam seus próprios PDPs para realizar a autorização de forma distribuída.

3. Federação de Políticas de Autorização

Uma federação de políticas de autorização permite que um ou mais clientes com contas em nuvens heterogêneas consigam utilizar políticas de segurança equivalentes em cada uma delas. Por exemplo, um cliente com contas na AWS, Google Cloud e numa nuvem privada OpenStack pode definir uma política de segurança através de um PAP centralizado, com sintaxe e semântica próprios, e esta política será traduzida para equivalentes locais de cada uma delas, respectivamente. Desta forma, usuários destas contas, representados através dos mesmos atributos atestados por emissores de credenciais verificáveis ou provedores de identidade, terão a mesma experiência de autorização ao acessá-las.

A arquitetura da solução está representada na Figura 2. O módulo de gerenciamento de políticas de autorização federadas - *Federated Authorisation Policy Management Service (FAPManS)* é formado por um PAP centralizado e um sincronizador. O PAP permite a criação e o gerenciamento de políticas através de interfaces gráfica (GUI) e de aplicação (API). Cada membro da APF deve utilizar um agente para se registrar no sincronizador através de um mecanismo de *publish/subscribe*. Sempre que uma nova versão da política for definida (passo 1), o sincronizador notifica os agentes registrados/interessados naquela política (passo 2), e estes recuperam as atualizações do PAP (passo 3). Os adaptadores são módulos responsáveis pela tradução das políticas. Os agentes utilizam os adaptadores para converter as políticas no formato e semântica da APF para os equivalentes locais (passo 4) e atualizar as políticas no PAP local da nuvem (passo 5), tornando-as disponíveis para o PDP (passo 6).

Todo o processo de administração, sincronismo e tradução de políticas ocorre em "tempo de administração" (*offline*). Isso significa que, embora o desempenho destas atividades seja importante (ex. o sincronismo das políticas deve ser feito rapidamente para as novas regras passarem a vigorar), ele não é tão crítico como o "tempo de autorização" (*online*), quando usuários enviam dezenas ou centenas de requisições de acesso ao sistema e as regras devem ser avaliadas para cada uma das requisições. Por este motivo, uma análise de desempenho da tradução de políticas foi considerada fora do escopo da pesquisa.

Nesta seção, os principais componentes da solução APF são detalhados: o PAP do FAPManS (3.1), que armazena políticas globais, e os mapeamentos semântico (3.2) e sintático (3.3), realizados pelos adaptadores, concretizando as políticas nos formatos

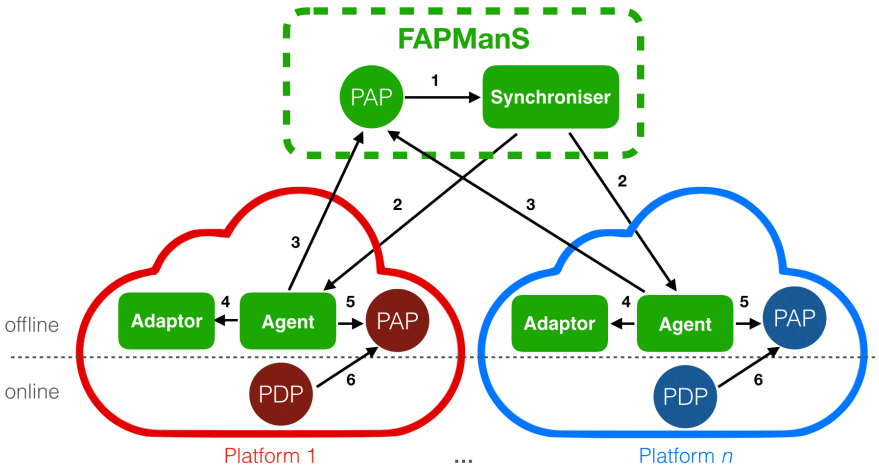


Figura 2. Arquitetura de Federação de Políticas de Autorização (APF)

nativos aceitos pelas plataformas de nuvem.

3.1. FAPManS

As políticas de acesso comuns são armazenadas no FAPManS na forma normal disjuntiva – *Disjunctive Normal Form (DNF)*. Políticas de autorização podem ser descritas através de expressões lógicas representadas em diversos formatos, como eXtensible Access Control Mark-up Language (XACML) ou nos padrões proprietários da AWS, OpenStack e Google Cloud. Uma forma normal foi adotada nesta solução com objetivo de facilitar a comparação entre tais políticas heterogêneas, por exemplo, identificando regras idênticas escritas de formas diferentes.

Uma política representada em DNF é composta por regras combinadas com o operador lógico "ou", significando que basta que apenas uma regra seja validada para que o acesso seja concedido. Se nenhuma regra for validada, o acesso deve ser negado por padrão. Portanto, esta forma normal serve bem para representar políticas de autorização. A Figura 3 mostra um exemplo de uma **política** em DNF composta por três **regras** combinadas pelo operador "ou". Uma dessas regras é composta por 5 **condições** combinadas pelo operador "e". Portanto, um usuário só será autorizado se possuir atributos *"role"* = *"manager"* e *"name"* $\neq$ "Ioram", requisitando realizar a "ação" de "criar" um "recurso tipo" "VM" com "disco" $\leq$ 3GB.

Os termos utilizados nas políticas da APF são definidos numa ontologia. Neste trabalho, foi definida uma ontologia com os termos utilizados por nuvens IaaS. No entanto, a solução de APFs também pode ser adaptada para outros modelos como Platform as a Service (PaaS) e Software as a Service (SaaS). A ontologia deve conter termos que representem conceitos comuns nas diversas plataformas de nuvens. Por exemplo, máquinas virtuais - *Virtual Machine (VM)* contendo discos virtuais - *Virtual Hard Disk (VHD)* e interfaces de rede virtuais - *Virtual Network Interface (VNI)* que são conceitos comuns para o serviço de computação em nuvem, enquanto usuários estão presentes no serviço de gestão de identidade e acesso - *Identity and Access Management (IAM)*.

A Figura 4 explica como os termos da ontologia são definidos e classificados. No cenário 4(a), uma APF é composta por nuvens de duas plataformas distintas. Elementos

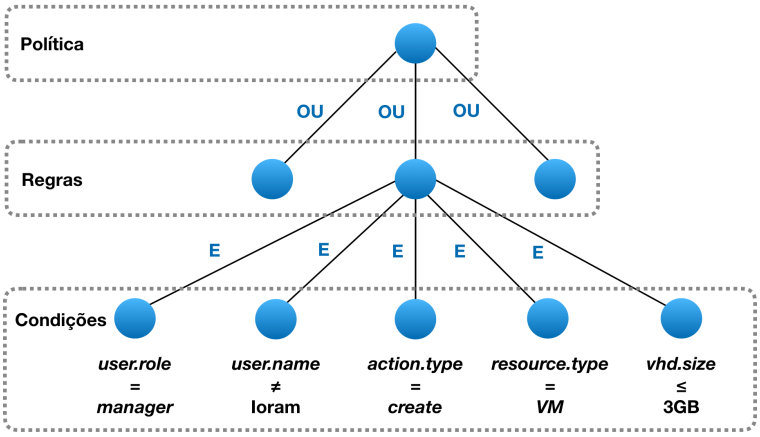
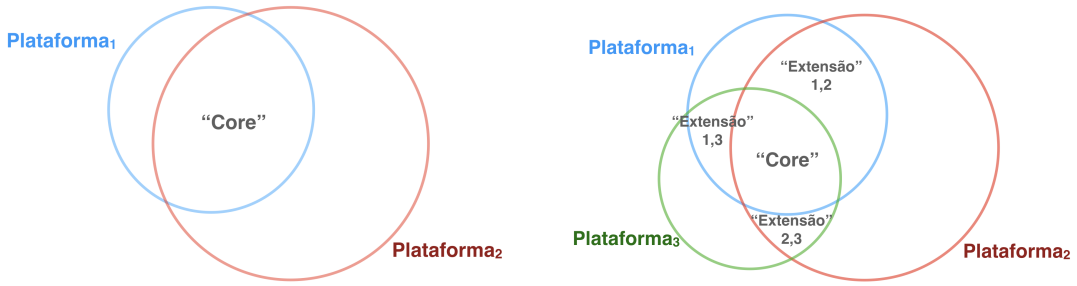


Figura 3. Política de Autorização em uma Forma Normal Disjuntiva (DNF)

comuns a estas plataformas formam a ontologia (*core*) – $Plataforma_1 \cap Plataforma_2$. Estes elementos são termos que formam condições e compreendem atributos do sujeito (usuário ou sistema que realiza o acesso), da ação pretendida, do recurso acessado e do ambiente, operadores (ex. de igualdade ou de pertinência) e valores esperados para os atributos. No cenário 4(b), uma terceira plataforma se une à APF. Os elementos comuns às três formam o *core* – $Plataforma_1 \cap Plataforma_2 \cap Plataforma_3$, enquanto elementos comuns a apenas duas das plataformas formam grupos chamados de *extensões*. Uma política composta de regras apenas com elementos do *core* pode ser traduzida para todas as nuvens da federação. Porém, se houver regras com elementos das extensões, apenas as plataformas que definirem estes elementos poderão aplicá-las. É possível ainda definir regras na APF com elementos de uma plataforma específica. Neste caso, os elementos não precisam estar na ontologia, e as regras serão aplicadas apenas a nuvens desta plataforma.



(a) 2 plataformas (b) 3 plataformas

Figura 4. Dinâmica de ontologia para APFs: de (a) para (b).

Como exemplos de termos, considerando as plataformas OpenStack, AWS e Google Cloud Provider (GCP), temos:

1. Core: operador igual (=), usuário, VM, VHD, VNI
2. Extensão OpenStack/AWS: operador diferente ($\neq$), grupos de usuários
3. Específico AWS: operador "menor que" ($<$) e "maior que" ($>$)

O FAPManS provê uma interface (API) que permite os administradores realizar funções de criar, ler, atualizar e apagar (CRUD) em políticas e regras da APF. Ele também

provê uma interface gráfica (GUI), onde administradores podem editar visualmente as políticas e regras [Sette 2016].

3.2. Mapeamento Semântico

A função de tradução de regras realizada pelos adaptadores pode ser decomposta em duas fases: o mapeamento semântico e o mapeamento sintático. O mapeamento semântico é responsável por traduzir políticas em DNF, onde os termos das condições são descritos por elementos da ontologia, para políticas ainda em DNF, mas com termos correspondentes das plataformas de nuvem. Por exemplo, na ontologia representamos a permissão de criar VMs pelas condições `resource.type = "vm"; action.type = "create"`. Numa tradução

- para OpenStack, os termos seriam traduzidos para `service="compute"; action="create"`,
- para AWS seriam `service="ec2"; action="RunInstances"` e
- para GCP seriam `service="compute"; resource="instances"; action="create"`.

É possível que condições sejam mescladas ou quebradas nas traduções, como acontece na ação `"RunInstances"` da AWS, que combina as semânticas da ação **criar** e do tipo de recurso **instância** que representa VMs. Já na plataforma OpenStack, a ação **criar** associada ao serviço **computação** pressupõe implicitamente que o tipo do recurso é uma VM.

3.3. Mapeamento Sintático

O mapeamento sintático traduz as políticas em DNF para formatos locais das plataformas e vice-versa. Seguindo o exemplo anterior, as condições em DNF já com termos da nuvens locais são traduzidas para suas respectivas nuvens, conforme a Tabela 1.

Tabela 1. Exemplo de Mapeamento Sintático

	DNF	Sintaxe Local
Ontologia	resource.type = “vm”, action.type = “create”	–
OpenStack	service = “compute”, action = “create”	“compute:create”: “”
AWS	service = “ec2”, action = “RunInstances”	“Effect”: “Allow”, “Action”: “ec2:RunInstances”, “Resource”: “*”
GCP	service = “compute”, resource = “instances”, action = “create”	compute.instances.create

Observe que na AWS as regras possuem efeitos explícitos (ex. “Effect”: “Allow”), enquanto que nas demais, inclusive na APF (Ontologia), as regras possuem um efeito de

permissão (*allow*) implícito, e a negação (*deny*) implícita se dá pela não validação das regras de uma política. Regras cujo efeito é *allow* possuem o mesmo significado que as políticas com o *allow* explícito. Já as regras com efeito *deny* devem prevalecer às regras com efeito *allow*, sendo avaliadas anteriormente. Por exemplo, se há uma regra que proíbe o acesso de leitura a arquivos com rótulo "confidencial" para usuários que não são administradores, mesmo que exista uma outra regra permitindo tal acesso, ele deve ser negado.

Para representar regras com efeito explícito em DNF, a expressão $dnf(A \wedge \neg D)$ é usada, sendo A a expressão lógica que representa regras com efeito *allow* e D com efeito *deny*. A expressão resultante garante que caso haja alguma condição C no conjunto de regras de efeito *allow* que contradiga qualquer regra com efeito *deny*, a regra combinada irá sempre ser avaliada como negação, pois ela conterá a expressão $C \wedge \neg C$ que é avaliada para $false$. No entanto, a arquitetura da solução permite que, após a combinação das regras pela expressão acima e sua conversão para DNF, novas regras sejam acrescentadas à política. Estas novas regras podem por sua vez contradizer as regras de efeito *deny*, uma vez que estas não serão mais combinadas. Esta é uma limitação da solução, que atualmente não preserva a semântica de regras com efeito *deny*. Para resolver esta limitação, a política APF deve manter duas estruturas separadas com regras de efeito *allow* (atual) e *deny*. Desta forma, sempre que uma regra com efeito *allow* for inserida, as duas estruturas devem ser combinadas gerando a política global, e a semântica do *deny* será mantida.

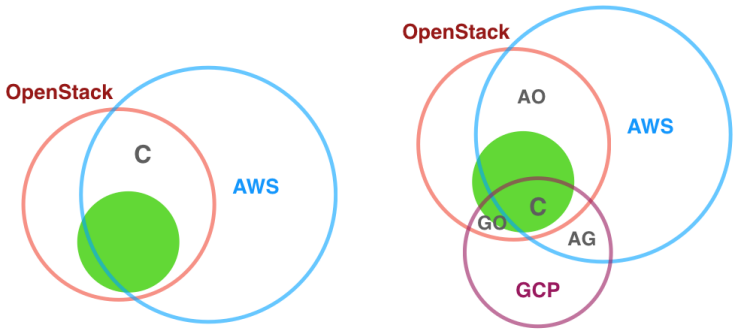
4. Implementação e Resultados

Três adaptadores foram implementados para as plataformas de nuvem OpenStack, AWS e GCP. Nesta seção, demonstraremos o uso da solução de APF através de um exemplo utilizando estes adaptadores, discutiremos sobre a implementação destes e os resultados obtidos.

4.1. Exemplo de Uso

Em trabalho anterior [Sette et al. 2017], foram implementados adaptadores para as plataformas OpenStack e AWS. Uma APF foi definida a partir de uma política OpenStack real, traduzida para o formato APF global, isto é, no formato DNF com termos definidos numa ontologia formada pelos conceitos comuns às duas nuvens. Este artifício foi usado uma vez que a GUI do FAPManS não foi implementada. A Figura 5(a) descreve o cenário onde a política OpenStack (círculo verde) é composta por um subconjunto de elementos permitidos para a nuvem OpenStack (círculo vermelho). A maior parte dos elementos desta política faz parte do "core" (C) da ontologia, enquanto uma parte menor é mantida em termos do OpenStack. As regras definidas no "core" da ontologia foram traduzidas para o formato AWS.

Neste trabalho, um novo adaptador foi desenvolvido, para a nuvem GCP, representada na Figura 5(b) por um novo círculo (em roxo). Com a diminuição do *core* da ontologia ($OpenStack \cap AWS \cap GCP$), parte das regras da política – na *extensão* AWS/OpenStack (AO) – não podem ser traduzidas para GCP. A ontologia anterior pode ser ampliada com duas novas extensões: AG, contendo elementos comuns às plataformas AWS e GCP, e GO, com elementos comuns a GCP e OpenStack. Porém, isto não foi contemplado no escopo deste trabalho.



(a) OpenStack e AWS (b) OpenStack, AWS e GCP

Figura 5. Ontologias para APFs: de (a) para (b).

Observe o fluxo exibido na Figura 6. Inicialmente, o adaptador OpenStack traduz a política local para uma global em dois passos: (1) mapeamento sintático gerando uma política intermediária em DNF com semântica OpenStack e (2) mapeamento semântico traduzindo os termos para a ontologia. O adaptador AWS recebe a política global gerada pelo passo 2 e traduz para o formato local também em dois passos: (3) mapeamento semântico traduzindo os termos para AWS e mantendo o formato DNF e (4) mapeamento sintático traduzindo de DNF para o formato de políticas AWS. Por fim, o adaptador GCP realiza os mapeamentos semântico (5) e sintático (6) gerando uma política GCP equivalente.

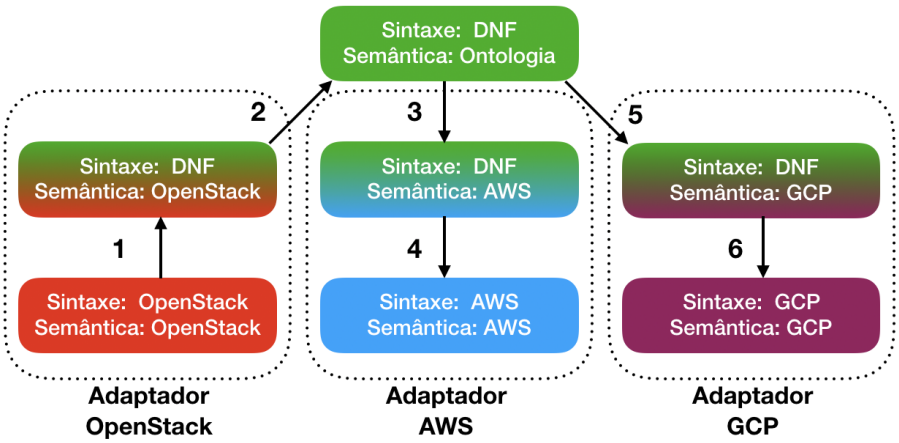


Figura 6. Exemplo de uso da solução.

A política inicial do Nova (serviço de computação da plataforma OpenStack) foi definida conforme o Código 1. Ela é composta por 13 regras, descritas nas linhas 5 a 17. As linhas iniciais (2 e 3) contêm apelidos para expressões mais complexas, usados nas regras para evitar repetições.

O adaptador OpenStack realiza o mapeamento sintático (vide passo 1 da Figura 6), traduzindo cada uma das regras para DNF. Tomando a linha 14 como exemplo,

1. a regra é expandida para `"compute:delete": "role:admin or project_id:%(project_id)s"` através da substituição do apelido "admin_or_owner" (linha 2);

```
{
    "admin_or_owner":  "role:admin or project_id:%(project_id)s",
    "default": "rule:admin_or_owner",

    "compute:create": "",
    "compute:get": "",
    "compute:update": "",
    "compute:start": "rule:admin_or_owner",
    "compute:stop": "rule:admin_or_owner",
    "compute:attach_volume": "",
    "compute:detach_volume": "",
    "compute:attach_interface": "",
    "compute:detach_interface": "",
    "compute:delete": "rule:admin_or_owner",
    "network:get": "",
    "network:create": "",
    "network:delete": ""
}
```

Código 1. Exemplo de Política do OpenStack Nova

2. em seguida, condições, no formato "<atributo> <operador> <valor>", combinadas por operadores lógicos "e" ($\wedge$), "ou" ($\vee$) e "não" ($\neg$) são extraídas.

O resultado é a expressão $C_1 \wedge C_2 \wedge (C_3 \vee C_4)$, onde: $C_1 : service = \text{“}compute\text{”}$, $C_2 : action = \text{“}delete\text{”}$, $C_3 : role = \text{“}admin\text{”}$ e $C_4 : project\_id = \text{“}\%(project\_id)s\text{”}$. Os termos "service" e "action", bem como o operador de igualdade ($=$), foram inferidos pelo adaptador através de seu conhecimento do formato de políticas OpenStack. Por fim, o operador converte a expressão para DNF obtendo o resultado $(C_1 \wedge C_2 \wedge C_3) \vee (C_1 \wedge C_2 \wedge C_4)$, ou seja, duas regras DNF foram geradas a partir de uma única regra OpenStack. Como toda expressão lógica pode ser transformada na forma normal, o mapeamento sintático deve sempre traduzir 100% das regras.

Para gerar a regra global da APF, o adaptador realiza o mapeamento semântico, convertendo os termos das condições para a ontologia (passo 2). Desta vez, algum termo (atributo, operador ou valor) pode não encontrar um equivalente e, portanto, não ser traduzido. Usando o exemplo anterior, duas regras são geradas:
(1) $resource.service = \text{“}compute\text{”} \wedge resource.type = \text{“}vm\text{”} \wedge action.type = \text{“}delete\text{”} \wedge user.role = \text{“}admin\text{”}$ e
(2) $resource.service = \text{“}compute\text{”} \wedge resource.type = \text{“}vm\text{”} \wedge action.type = \text{“}delete\text{”} \wedge resource.tenant.id = \text{“\$}(user.tenant.id)\text{”}$.

Neste exemplo, a condição "service=compute" é transformada em duas condições "resource.service=compute" e "resource.type=vm". Novamente, o adaptador usa seu conhecimento sobre políticas OpenStack e infere o tipo do recurso através do nome do serviço. A condição $resource.tenant.id = \$(user.tenant.id)$ compara se a "conta" que contém o recurso é a mesma "conta" que o usuário obteve o acesso. Desta forma, o valor da condição é avaliado dinamicamente no momento da validação da regra e armazenado na forma de uma variável.

Para traduzir a política global no formato AWS (passo 3), o adaptador AWS inicialmente realiza o mapeamento semântico, mantendo o formato DNF. O mapeamento semântico traduz a primeira regra da seguinte forma: $resource\_service =$

$\text{“}ec2\text{”} \wedge action\_service = \text{“}ec2\text{”} \wedge action = \text{“}TerminateInstances\text{”} \wedge principal\_role = \text{“}admin\text{”} \wedge principal\_account = \text{“}123456\text{”}$. O adaptador AWS infere algumas informações, como o serviço da ação e a conta em que o usuário está mapeado. Em seguida, o adaptador realiza o mapeamento sintático (passo 4), transformando as informações para o formato AWS, obtendo o resultado mostrado no Código 2.

```
{
    "Action" : "ec2:TerminateInstances",
    "Effect" : "Allow",
    "Principal" : {"AWS" : "arn:aws:iam::123456:role/admin"},
    "Resource" : "arn::ec2:::*"
}
```

Código 2. Resultado do mapeamento sintático para AWS

Para traduzir a política global para o formato GCP (passo 5), o mapeamento semântico é realizado (semelhante ao passo 3), substituindo os termos da ontologia por aqueles específicos da GCP. Realizado o mapeamento semântico, a regra segue o seguinte formato: $service = \text{“}compute\text{”} \wedge resource = \text{“}instances\text{”} \wedge action = \text{“}delete\text{”} \wedge role = \text{“}admin\text{”}$. Após o adaptador GCP substituir os termos da ontologia por termos específicos da GCP durante o mapeamento semântico, é realizado o mapeamento sintático (passo 6), cujo resultado pode ser conferido no Código 3.

```
{
    "role" : "admin",
    "permissions" : ["compute.instances.delete"]
}
```

Código 3. Resultado do mapeamento sintático para GCP

4.2. Equivalência Semântica

Após uma tradução da política, seja do formato global para um local ou vice-versa, algumas regras podem não obter sucesso na tradução. Como vimos, no mapeamento sintático, 100% das regras são traduzidas, uma vez que expressões lógicas sempre podem ser transformadas em formas normais. Entretanto, no mapeamento semântico, alguns motivos podem levar a uma não-tradução. Por exemplo, a falta de um equivalente semântico, na ontologia ou formato local, para um elemento (atributo, operador ou valor) de uma condição, ou a falta de uma regra de mapeamento no adaptador, podem fazer um mapeamento falhar.

A métrica Nível de Equivalência Semântica – *Level of Semantic Equivalence (LSE)* foi proposta neste trabalho para medir, em termos quantitativos, a razão das regras que foram mapeadas com sucesso após uma tradução. Ela é definida pela expressão:

$$L_{SE} = \frac{Regras_{Traduzidas}}{Regras_{Total}}$$

É importante considerar que as quantidades de regras traduzidas e regras totais devem ser medidas em políticas no formato DNF, uma vez que em formatos locais é

possível que regras sejam mescladas, distorcendo os valores. Isso é possível, pois sempre que uma tradução é realizada por um adaptador, um formato intermediário em DNF com termos locais é produzido. Portanto, numa tradução de política global, em DNF, para local, inicialmente é realizado o mapeamento semântico gerando um formato intermediário em DNF com termos locais. A quantidade de regras traduzidas é medida neste formato, antes do mapeamento sintático ser realizado. A quantidade de regras totais é medida na regra original, também em DNF. Numa tradução de política local para global, o primeiro mapeamento é o sintático, traduzindo a regra para DNF com termos locais. A quantidade total de regras a serem traduzidas é medida neste formato intermediário. Posteriormente, o mapeamento semântico é realizado, e a quantidade de regras traduzidas é medida na política global resultante.

Um LSE alto (próximo de 1) significa que a maioria das regras de uma política pôde ser traduzida após um mapeamento, porém pode não indicar qualitativamente que a tradução teve êxito, pois as poucas regras que não puderam ser traduzidas podem ser cruciais para que a política seja efetiva. Portanto, uma análise qualitativa deve ser feita por um especialista após a tradução, a fim de identificar o motivo das falhas, e eventuais soluções. Três situações podem ser consideradas. Na primeira, a regra pode ser substituída por equivalentes. Por exemplo, se o operador $>$ não for suportado, a condição $idade > 18$ pode ser substituída pela regra $idade = 19 \vee idade = 20 \vee ... \vee idade = 120$. Na segunda, a regra não pode ser substituída, mas não há perda semântica real. Por exemplo, a regra pode fazer sentido apenas num contexto específico, não aplicando-se no contexto global da APF. Na terceira, a regra não pode ser substituída havendo perda semântica.

No caso das regras locais serem traduzidas para uma global, caso haja um histórico do uso das regras, é possível definir uma nova métrica considerando pesos maiores para regras mais frequentemente utilizadas. Esta evolução da métrica LSE pode trazer mais valor para o administrador avaliar o sucesso da tradução de uma política. Embora, ainda assim, pode existir uma regra muito importante e pouco usada que necessite de atenção.

A Tabela 2 apresenta os resultados obtidos após a tradução da política OpenStack definida no Código 1 para um formato global, e desta para os formatos locais da AWS e GCP.

Tabela 2. Medidas do Nível de Equivalência Semântica (*LSE*)

Passo	Origem	Destino	Regras	Intermediário	Traduzidas	LSE
1	`OS`	`DNF/OS`	13	16	–	–
2	`DNF/OS`	`Global`	–	16	13	81.3%
3	`Global`	`DNF/AWS`	13	13	–	–
4	`DNF/AWS`	`AWS`	–	13	13	100.0%
5	`Global`	`DNF/GCP`	13	13	–	–
6	`DNF/GCP`	`GCP`	–	13	8	61.5%

4.3. Discussão

As nuvens estudadas, OpenStack, AWS e GCP, apresentam sistemas de autorização bem distintos. A AWS apresenta um controle de acesso baseado em atributos – *Attribute-Based Access Control (ABAC)*, com um nível de granularidade bastante fina. É possível

definir condições bastante elaboradas com os atributos, utilizando diversos operadores de comparação ($<$, $>$, $\leq$, $\geq$), strings (LIKE), listas e conjuntos ($\subset$, $\supset$, $\in$), dentre outros. As plataformas OpenStack e GCP utilizam um sistema de controle de acesso baseado em papéis – *Role-Based Access Control (RBAC)*, porém com algumas diferenças. O OpenStack associa permissões, compostas pelo tipo de serviço (ex. computação, armazenamento, rede), tipo do recurso (muitas vezes de forma implícita) e a ação a ser executada sobre o recurso, a regras de acesso. Estas regras devem usar operadores de igualdade ($=$, $\neq$) e podem comparar atributos do usuário, como "papel" ou mesmo o "id do usuário", "id do projeto" a valores constantes ou variáveis, que podem incluir atributos do recurso. Tanto OpenStack como AWS permitem o uso de variáveis para comparar valores definidos em tempo de decisão. As políticas GCP são mais restritas e as permissões, compostas de tipo do serviço, tipo do recurso e ação, são mapeadas diretamente aos papéis. Regras mais elaboradas não são permitidas, sendo o operador de igualdade ($=$) o único possível.

Numa ontologia para estas três plataformas, o *core* é limitado pela interseção dos três modelos, sendo bastante reduzido principalmente pelas restrições do formato GCP. A existência de um *core* mínimo ou nulo é considerada uma limitação deste trabalho, pois restringe ou impossibilita uma política global. Para que isso não ocorra, as plataformas devem suportar um conjunto mínimo de elementos definidos para constar no *core*. Portanto, se uma plataforma é responsável por reduzir demais o core, deve ser considerado não adicioná-la à federação. *Extensões* da ontologia também podem ser consideradas para mitigar este problema. Elas permitem que regras com elementos fora do *core* possam ser escritas, e sejam aplicadas a um subconjunto das nuvens. Porém, este recurso aumenta a complexidade da política, uma vez que o administrador precisa ter ciência de para quais nuvens/plataformas a regra irá ser aplicada.

5. Conclusões

Administradores de segurança que gerenciam políticas em múltiplas nuvens heterogêneas têm um árduo trabalho de mantê-las atualizadas em cada uma de suas contas. Este trabalho apresenta uma solução, chamada "Federação de Políticas de Autorização" (APF), que permite a definição de uma única política de autorização, de forma centralizada, traduzida para cada uma das nuvens. As políticas globais da APF são definidas na forma normal disjuntiva DNF e os termos definidos por uma ontologia.

A tradução de políticas entre formatos diferentes pode ser realizada em duas etapas. O mapeamento sintático é responsável por traduzir a estrutura das políticas de formatos distintos. Políticas são expressões lógicas e a conversão para DNF permite que elas sejam facilmente comparadas e transformadas. O mapeamento semântico é responsável por traduzir os termos das condições das políticas, como atributos, operadores e valores, em seus equivalentes semânticos. Termos que não fazem sentido no contexto de pelo menos uma nuvem possivelmente não poderão ser traduzidos para esta(s). É papel do administrador de segurança analisar as regras que falharam no processo de tradução, para que os mesmos possam criar regras alternativas que possam ser traduzidas. O nível de equivalência semântica (LSE) mede de forma quantitativa a razão de regras que foram traduzidas, indicando aos administradores o trabalho que terão para analisar as falhas de forma qualitativa.

Neste trabalho, uma APF para nuvens de IaaS nas plataformas OpenStack, AWS

e GCP foi criada. Uma ontologia para as três nuvens foi definida, e a tradução de uma política inicialmente escrita para a plataforma OpenStack foi traduzida para AWS e GCP com LSEs de 100% e 61.5%, respectivamente. Como trabalhos futuros, podemos listar: a definição de APF para nuvens PaaS e SaaS, e a definição de sub-federações, onde parte dos membros pode definir um grupo de regras válidas apenas para eles.

Referências

Almutairi, A., Sarfraz, M., Basalamah, S., Aref, W., and Ghafoor, A. (2012). A distributed access control architecture for cloud computing. *Software, IEEE*, 29(2):36–44.

Bernabe, J. B., Perez, J. M. M., Calero, J. M. A., Clemente, F. J. G., Perez, G. M., and Skarmeta, A. F. G. (2014). Semantic-aware multi-tenancy authorization system for cloud architectures. *Future Generation Computer Systems*, 32(C):154–167.

Bittencourt, L. F., Calheiros, R. N., and Lee, C. (2017). Middleware for multicloud. *IEEE Cloud Computing*, 4(4):22–25.

Chadwick, D. W., Siu, K., Lee, C., Fouillat, Y., and Germonville, D. (2014). Adding federated identity management to openstack. *Journal of Grid Computing*, 12(1):3–27.

Moore, B. (2003). Policy Core Information Model (PCIM) Extensions. RFC 3460 (Proposed Standard).

Ngo, C., Demchenko, Y., and de Laat, C. (2016). Multi-tenant attribute-based access control for cloud infrastructure services. *Journal of Information Security and Applications*, 27-28:65 – 84. Special Issues on Security and Privacy in Cloud Computing.

Opara-Martins, J., Sahandi, R., and Tian, F. (2016). Critical analysis of vendor lock-in and its impact on cloud computing migration: a business perspective. *Journal of Cloud Computing*, 5(1):4.

Sette, I. S. (2016). *Access Control in IaaS Multi-Cloud Heterogeneous Environments*. PhD thesis, Universidade Federal de Pernambuco, Recife, PE, Brazil.

Sette, I. S., Chadwick, D. W., and Ferraz, C. A. G. (2017). Authorization policy federation in heterogeneous multicloud environments. *IEEE Cloud Computing*, 4(4):38–47.

Tang, B., Sandhu, R., and Li, Q. (2013). Multi-tenancy authorization models for collaborative cloud services. In *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, pages 132–138.

TI inside online (2018). Estudo mostra retração no mercado de cloud e data center no brasil. http://tiinside.com.br/tiinside/services/24/11/2018/estudo-mostra-retracao-no-mercado-de-cloud-e-data-center-no-brasil/. Online; postado em 24/11/2018; acessado em 10/12/2018.

Toosi, A. N., Calheiros, R. N., and Buyya, R. (2014). Interconnected cloud computing environments: Challenges, taxonomy, and survey. *ACM Comput. Surv.*, 47(1):7:1–7:47.

Yousif, M. (2017). Multiclouds in an enterprise – a love-hate relationship. *IEEE Cloud Computing*, 4(4):4–5.

Zeck, A. and Bouroudjian, J. (2017). Real-world experience with a multicloud exchange. *IEEE Cloud Computing*, 4(4):6–11.