Uma proposta de análise de desempenho para rastreamento de fluxos de trabalho usando SPIFFE/SPIRE

Milton P. Pagliusi [1], **Charles C. Miers**[1]

[1]Departamento de Ciência da Computação (DCC)
Universidade do Estado de Santa Catarina (UDESC)

milton.neto@edu.udesc.br,

charles.miers@udesc.br

Resumo. *A implementação de métodos seguros com o fim de estabelecer confiança entre entidades dentro de ambientes de nuvem é um problema discutido desde o início da adoção desta tecnologia por diferentes organizações. Este artigo propõe uma análise de desempenho de uma aplicação, utilizando SPIRE, entre o emprego de modos diferentes, durante execução das funções de delegação e uso de tokens de identidade estendidos.*

1. Introdução

Na crescente adoção de tecnologias para ambientes em nuvem por diferentes organizações, a tendência de ter aplicações divididas em escopos de menor tamanho e maior número é evidente, e.g., contêineres e microsserviços, e como consequência dessa tendência, mais pontos de vulnerabilidade em potencial [Feldman et al. 2020]. Diferentes abordagens foram implementadas para integrar mais segurança em ambientes de nuvem, muitas destas, como *Secure Production Identity Framework for Everyone* (SPIFFE), uma especificação para padronização de identidades entre microsserviços, e também uma abordagem alternativa para controle de acesso [Feldman et al. 2020]. O *SPIFFE Runtime Environment* (SPIRE) é uma implementação do SPIFFE, tem uma arquitetura sólida e visa oferecer uma solução de gerência de identidades de *workloads* [Falcão et al. 2022]. Contudo, a extensão do papel das credenciais é limitado em diferentes cenários, e.g., uma *workload* ao receber uma mensagem e queira verificar a identidade não apenas de qual *workload* está recebendo, mas também sob o nome de qual solicitante (e.g. usuário) esta *workload* está agindo. O uso de abordagens seguras é uma necessidade, mas também é necessário saber o impacto em desempenho a fim de conhecer os limiares operacionais e manter a aplicação final com desempenho satisfatório.

O objetivo deste artigo é analisar o desempenho do SPIRE usando o nosso modelo estendido de credenciais com modo de rastreamento para mensurar o impacto de latência e *payload* nas *workloads*. O artigo está organizado como segue. Fundamentação de SPIFFE / SPIRE, a extensão de seus documentos de identidade e seu modo de rastreamento na Seção 2. A proposta de análise do impacto no desempenho, assim como os critérios definidos para a análise na Seção 3.

2. Fundamentação

O SPIFFE é um projeto da *Cloud Native Computing Foundation* (CNCF), um conjunto de padrões para identificação de microsserviços em ambientes heterogêneos com o objetivo

de fornecer uma *framework* que ofereça, de forma otimizada, os processos e documentos necessários para que diferentes microsserviços, ou *workloads*, obtenham uma identidade criptográfica e uma validação para esta identidade [Feldman et al. 2020].

O SPIRE é a implementação de código aberto pronta para produção das especificações do *framework* SPIFFE, e é composto por dois componentes principais: (i) servidor, responsável pelo processo de autenticação de agentes, administração e emissão de diferentes documentos de identidade; e (ii) agente, responsável por delegar documentos de identidade para *workloads* e efetuar a atestação destas via uma API [Falcão et al. 2022].

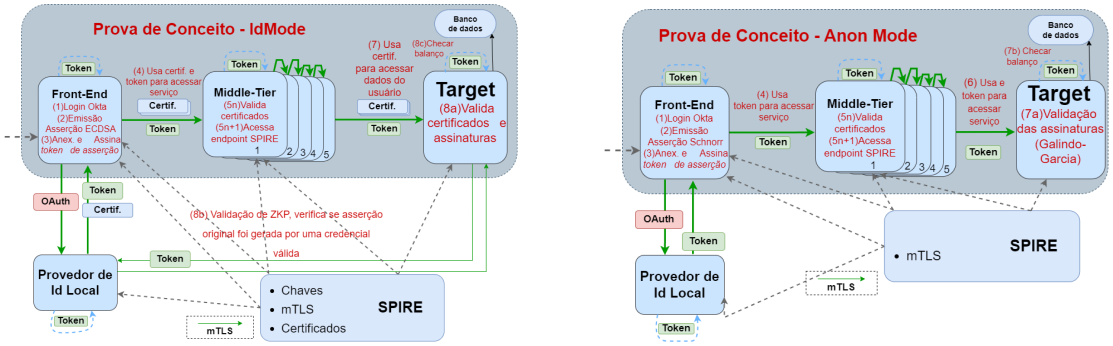
A especificação SPIFFE define a *framework* e documentos para emissão de identidades e autenticação entre microsserviços [Feldman et al. 2020], mas a extensão dos papéis destes documentos são limitados pelo seu contexto, e.g., a ausência de suporte a autenticações delegadas ou transitividade em documentos de identidade. Dentre esses documentos: (i) O SPIFFE ID, uma *string* que representa o nome da identidade de uma *workload*; e (ii) O SVID, ou *SPIFFE Verifiable Identity Document*, um documento criptográfico verificável utilizado para provar a autenticidade de uma identidade. Uma proposta para solucionar estas limitações foi a adesão de um novo documento chamado *Delegated Assertion SVID* (DA-SVID), um *JSON Web Token* (JWT) [IETF 2022] assinado por uma *workload* arbitrária e a construção de um novo modelo de asserções e *tokens* com a possibilidade de um rastreamento do fluxo pelo qual este *token* foi utilizado. As asserções com assinatura foram implementadas baseadas na necessidade de um *token* confiável que pudesse ser assinado por diferentes entidades, i.e., um *token* que ofereça suporte para assinaturas distribuídas e a capacidade de agregar diferentes assinaturas. O objetivo do modelo é oferecer a capacidade destas asserções carregar informações anexada ao documento, informações pertinentes ao processo de autenticação, etc. Essas propriedades são a base para este modelo de *token* poder ser rastreável. A implementação deste novo modelo foi chamado de modelo aninhado, consistindo de uma asserção originalmente composta por uma assinatura e *payload*, no qual um novo *token* será criado toda vez que novas informações ou *claims* forem adicionadas. Assim, a asserção original vai ser anexada com uma nova *payload* e o conjunto inteiro é assinado, sem a possibilidade de qualquer modificação ser feita sem invalidar a assinatura.

O rastreamento de fluxo do percurso no qual o *token* foi utilizado pode ser identificado pelo salto entre diferentes *workloads* utilizando o elo entre duas *claims* inseridas no *token*. Uma é a *claim* emissor, identificando qual *workload* o *token* é proveniente e quem assinou e a outra é *claim* audiência, identificando o destino do *token* para qual *workload* está direcionada, e nesta *workload* destino, a autoridade para utilizá-la e anexar novas informações a este *token*. A análise objetiva comparar o consumo computacional no desempenho de uma aplicação, implementada em uma PoC, durante a utilização do modelo aninhado em conjunto com os *tokens* e asserções assinadas em contraste com o comportamento da aplicação sem a utilização da extensão de credenciais.

3. Proposta e critérios

A proposta desta pesquisa é coletar dados de desempenho dentro do caso de uso proposto, que se refere a delegação de asserções com assinatura, com o modelo aninhado como foi explicado brevemente anteriormente, dentro um de ambiente SPIRE. O cenário para este caso de uso é permitir que uma *workload* possa receber, de uma outra *workload*,

um *token* de asserção com diferentes *claims* anexadas, aderir novas informações, e.g., sua próxima audiência, e assiná-lo, criando dessa forma um novo *token* pronto para ser encaminhado ao próximo destino. Além disso, seja possível verificar tanto a identidade da *workload* remetente como o nome de qual solicitante aquela instância estaria agindo. Com o intuito de simular o comportamento esperado destas asserções e o modelo aninhado em um contexto de aplicação real, uma aplicação foi desenvolvida uma PoC.



(a) *ID-Mode* (b) *Anon Mode*

Figura 1. Prova de conceito: requisição utilizando o modo *anon* e modo *ID*.

As Figuras 1(a) e 1(b) representam a prova de conceito desenvolvida com a finalidade de simular uma aplicação bancária, em dois modos diferentes, com o objetivo de permitir a emissão de DA-SVID a utilização do novo modelo aninhado de asserções. Na Figura 1(a), o fluxo do processo começa com o solicitante fornecendo um *token* OAuth (passo 1), por sua vez o *Local IdP* recebe este *token* do *Front End*, e emite um *token* relacionado ao OAuth (passo 2), toda *workload* em diante, que utilizar o *token* em uma requisição, terá a obrigação de anexar *claims* e assiná-lo, da forma como o modelo aninhado propõe, anexando o *trust bundle* e seu certificado (passos 3 e 4), na conclusão, o *Target* recebe o *token* e *trust bundle* utilizando os certificados em uma validação sequencial de assinaturas (passo 5). Nesta prova de conceito, cada componente é uma *workload* autenticada com um documento SPIFFE-ID em execução em um contêiner de Docker.

O *Local IdP* é um componente confiável que tem a permissão para emitir *tokens* de identidade estendida, quando o *Front End* da aplicação tem uma requisição do usuário para acesso de dados, ele estabelece uma conexão *Mutual Transport Layer Security* (mTLS) com o *Local IdP*, enviando o *token* OAuth do usuário solicitante. Por sua vez, o *Local IdP* recebe o *token* OAuth, emite um DA-SVID, assina este *token* com uma chave privada, e o devolve ao *Front End*. O componente *Middle Tier*, são explicitamente, 5 simulações diferentes de aplicações em nuvens de forma genérica, e.g., um balanceador de cargas, com o fim de demonstrar a transitividade do *token* DA-SVID. Este componente, ao receber requisições que utilizam DA-SVID, valida sua assinatura e expiração, e após isso, o documento passa por todos as simulações até atingir o componente *Target*, onde uma validação completa é efetuada, com intuito de permitir o acesso a dados.

Como proposta, um *benchmark* do consumo computacional da aplicação da Figura 1, com captura de informações como: (i) consumo de memória; (ii) consumo de processador; (iii) tráfego de rede (volume trafegado e conexões); (iv) latência; e (v) *payload* gerado pela adoção do modelo aninhado de *tokens*.

Para o plano de testes, os seguintes cenários são definidos:

1. Cenário básico: comunicação utilizando o modo *ID*.
2. Cenário básico: comunicação utilizando o modo *anon*.
3. Cenário expandido: comunicação utilizando o modo *ID* adicionando o custo da comunicação com as diferentes *workloads* de *Middle Tier*.
4. Cenário expandido: comunicação utilizando o modo *anon* adicionando o custo da comunicação com as diferentes *workloads* de *Middle Tier*.

Os cenários descritos são executados para permitir a comparação entre o desempenho da aplicação, em ambos os modos, ao utilizar diferentes *tokens* para permitir o rastreamento dos fluxos de trabalho.

4. Considerações

Um *framework* para controle e emissão de identidades universais assim como a capacidade de operá-la em ambientes heterogêneos surge como uma solução para a crescente necessidade de aderir segurança e estabelecer confiabilidade dentro de infraestruturas baseadas em ambientes de nuvem. A especificação SPIFFE padroniza e explora, de maneira flexível, processos de identificação, atestação e autenticação de *workloads*. Logo, uma análise do desempenho durante o funcionamento da aplicação de prova de conceito (PoC) é relevante para poder mensurar, o *payload* durante o emprego do *token* DA-SVID no modo *ID* em comparação com o tamanho de *payload* durante o emprego do *token* de asserção com assinatura, e seu modelo aninhado no modo *Anon*, assim como também compreender como esta diferença de tamanho influencia a latência da comunicação entre *workloads*.

A implementação de novos modelos de documentos resulta em uma maior expressividade das credenciais em comparação com o contexto original, anteriormente limitado a processos de autenticação: (i) o documento DA-SVID, utilizado na aplicação como um elo entre um usuário solicitante e uma *workload* de *Front End*, traz o conceito de identidade transitiva para dentro do ambiente SPIFFE, e é utilizado no modo *ID* da aplicação de prova de conceito; e (ii) o modelo aninhado, um *token* que carrega informações em anexo, e que permite a uma *workload* emitir um novo *token* utilizando uma asserção original anexada a *claims* pertinentes, i.e., um esquema que suporta assinaturas distribuídas, sendo utilizado no modo *ID* e no modo *anon* da aplicação de prova de conceito.

Agradecimentos: Os autores agradecem o apoio da HPE, USP, LabP2D/UDESC e a FAPESC.

Referências

Falcão, E., Silva, M., Luz, A., and Brito, A. (2022). Supporting confidential workloads in spire. In *2022 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 186–193.

Feldman, D., Fox, E., Gilman, E., Haken, I., Kautz, F., Khan, U., Lambrecht, M., Lum, B., Fayó, A. M., Nesterov, E., Vega, A., and Wardrop, M. (2020). Solving the bottom turtle — a spiffe way to establish trust in your infrastructure via universal identity.

IETF (2022). Json web token (jwt). https://datatracker.ietf.org/doc/html/rfc7519. Acesso em: 9 Mai. 2022.