SECURITY ANALYSIS OF OPENID CONNECT PROTOCOL INTEGRATION ON CLOUDS BASED ON OPENSTACK USING AN EXTERNAL IDENTITY PROVIDER

Glauber Cassiano Batista, Charles Christian Miers
Universidade do Estado de Santa Catarina - UDESC
Departamento de Ciência da Computação, Campus universitário – Joinville - SC
glauber@colmeia.udesc.br, charles@udesc.br

Abstract. *Several Internet services require different pairs of usernames and passwords in order to authenticate their users. The password management process is time consuming and susceptible to security issues. Moreover, cloud computing solutions already run into related problems. Single Sign-On is a way to handle all the multiple accounts that a user may have and OpenID Connect has a great adoption as a Single Sign-On* (SSO) *solution. As well as OpenID Connect, OpenStack is growing as a cloud software due its large community and because its open source code. In this sense, our goal is to analyze the security aspects related to the usage of OpenID Connect, a single sign-on mechanism, in cloud computing solutions based on OpenStack using a external Identity Provider.*
Key words: *OpenID Connect, Cloud, Single Sign-On*

1. INTRODUÇÃO

Uma nuvem computacional é um modelo que permite acesso ubíquo, conveniente e sob demanda a um conjunto de recursos configuráveis que podem ser rapidamente provisionados e liberados com o mínimo de esforço [1]. O OpenStack[1] tem obtido destaque como solução de nuvem computacional [2], [3], no intuito de beneficiar e aumentar o aprendizado da comunidade, seja dentro de uma única organização ou na integração de várias organizações.

Vários sistemas em um mesmo departamento ou organização comumente utilizam serviços de autenticação distintos, o que resulta em dificuldades para o usuário, que deve lembrar de suas várias informações de autenticação [4]. Para cada ação de autenticação, é gerado um novo processo e requer a alocação de recursos do sistema, sem mencionar a possibilidade de comprometimento de informações do usuário. Assim como grande parte dos serviços na Internet, as nuvens computacionais se deparam com problemas similares e algumas já empregam mecanismos *Single Sign-On* (SSO), cuja principal é prover um identificador único ao usuário para que este possa autenticar-se em qualquer serviço que o suporte [5], [6], [7].

O uso de tecnologias de autenticação e autorização SSO centradas no usuário (*e.g.*, OpenID Connect) tem crescido nos últimos anos e proporciona uma possibilidade dinâmica e acessível para a solução do problema. Este trabalho tem como objetivo analisar a segurança do uso de tecnologias de autenticação e autorização centradas no usuário (*i.e.*, OpenID Connect) para disponibilização em serviços de nuvens baseadas no OpenStack.

2. OPENSTACK E KEYSTONE

O OpenStack é uma solução de nuvem que controla diversos conjuntos de recursos de processamento, armazenamento e rede de um *data center* [8]. O OpenStack é composto por diversos serviços com funções específicas, como armazenamento, rede, computação, BD, telemetria, orquestração e identidade.

O Keystone é o componente responsável por prover a autenticação e gerenciamento de identidade para cada serviço do OpenStack. Além da autenticação, o Keystone faz uma autorização de alto nível, transformando os atributos de autenticação em papéis, *Role-Based Access Control* (RBAC). Porém, a autorização ocorre de fato de forma descentralizada em cada

1 - http://www.openstack.org

módulo do OpenStack posteriormente, com base nos papéis e projetos do usuário [6].

Para utilizar SSO no Keystone é necessário configurar a extensão OS_FEDERATION com o plugin desejado (*e.g.*, OpenID Connect, SAML). No caso do OpenID Connect, são utilizadas *claims*, que requisitam e fornecem atributos dos usuários registrados. As *claims* do OpenID Connect são fornecidas através de um arquivo JSON assinado e cifrado utilizado posteriormente no mapeamento do usuário no projeto correto.

3. OPENID CONNECT

O OpenID Connect é a terceira geração da tecnologia de autenticação OpenID[2] que opera com o protocolo de autorização OAuth 2.0[3] e fluxo de mensagens diretas REST/JSON, utilizando também SSL/TLS para a cifragem e lidar com questões de comunicação segura [9].

Para explicar como o usuário interage com o OpenID Connect, um cenário fictício é utilizado. O usuário deseja acessar a nuvem exemplo.com, nesse cenário, denominado Provedor de Serviço (SP). Em vez de preencher o formulário de cadastro, o usuário fornece um identificador (e.g., uma URL) que representa sua identidade. A partir deste momento, o SP faz o processo de descoberta para verificar a propriedade do usuário sobre o identificador. O usuário deve então se autenticar no Provedor de Identidades (IdP) utilizando as suas credenciais (e.g., par de usuário e senha) e autorizar o acesso do SP às suas informações de identidade. O IdP redireciona o usuário através do UA (e.g., navegador) para o SP, que fornece uma nova conta ao usuário, caso não possua uma. O fluxo descrito pode ser observado na Figura 1.

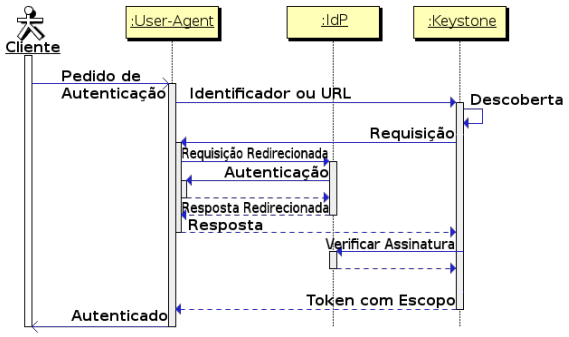

Figura 1: Autenticação OpenStack e OpenID Connect

Na Figura 1, é possível observar que existe um ator e três objetos: *User-Agent* (UA), IdP e o Keystone, nesse caso representando o SP. Ao fim do processo de autenticação o usuário recebe um *token* sem escopo que será trocado por um *token* com escopo, que define o projeto do usuário.

4. TRABALHOS RELACIONADOS

Nos últimos anos os mecanismos SSO têm sido muito debatidos e diversas soluções surgiram. Dessa forma, trabalhos sobre a segurança e implementação são comumente encontrados na literatura [10], [11], [12].

Delft *et. al* [10] realizaram uma análise de segurança do protocolo OpenID 2.0 sob duas perspectivas: do usuário e técnica. Vulnerabilidades de segurança e privacidade foram levantadas, como o registro das atividades do usuário no IdP, ataques *Cross-Site Scripting* (XSS), ataques *Man-in-the-Middle, phishing* e reciclagem de identificadores OpenID. Os autores também relataram que um terço das vulnerabilidades podem ser resolvidas com baixo custo de implementação.

Yang *et. al* [11] realizaram uma análise de segurança do protocolo OAuth 2 com quatro módulos de ataque, um cenário local e outro remoto. Foi constatado que o cenário local não oferece o grau de segurança adequado, uma vez que permite a reutilização dos códigos de segurança. Já o cenário remoto passou no teste, uma vez que oferecia canais cifrados com TLS e descartava os *tokens* de acesso já utilizados.

Por fim, Li *et. al* [12] analisaram a segurança da implementação do OpenID Connect do Google. Foi constatado que

2 - http://openid.net
3 - http://oauth.net/2/

grande parte das vulnerabilidades são oriundas da não entendimento dos desenvolvedores que utilizam o Google para autenticar seus usuários. Também foi relatado o problema da "concessão de autorização automática", que permite ataques *Cross-Site Request Forgery* (CSRF).

Ainda que as análises citadas ofereçam informações importantes a serem consideradas na escolha de um sistema SSO, não foram encontradas análises de segurança da utilização do OpenID Connect em nuvens computacionais, principalmente àquelas baseadas em OpenStack.

5. AMBIENTE DE TESTES

O ambiente utilizado para os experimentos é formado por três servidores: um nó controlador, um nó de rede e um nó de computação, executando RDO OpenStack Liberty sobre GNU/Linux CentOS 7. Os usuários se autenticam através do protocolo OpenID Connect, com IdP do Google. O nó controlador da nuvem possui duas interfaces de rede: uma para acesso externo e outra para gerenciamento. O nó de rede possui três interfaces: uma com saída para a Internet para as máquinas virtuais, uma para gerenciamento e outra para o tráfego interno das máquinas virtuais. O nó de computação possui duas interfaces de rede: uma para gerenciamento e outra para o tráfego interno das máquinas virtuais. A Figura 2 ilustra a arquitetura da nuvem OpenStack utilizada nos testes. A rede de gerenciamento possui o endereço 10.0.0.0/24 e a rede de tráfego interno o endereço 172.16.10.0/24.

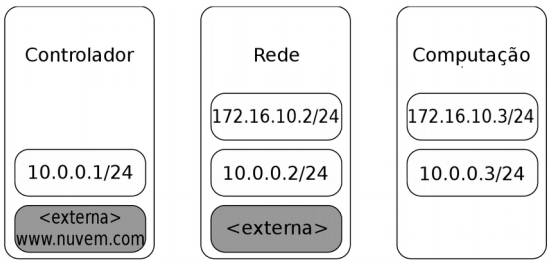

Figura 2: Ambiente de testes OpenStack

controlador e todos os serviços se autenticam pela URL http://10.0.0.1:5000. As requisições internas e administrativas da API são realizadas através da rede de gerenciamento. Já as requisições públicas da API são realizadas pela rede externa. Para habilitar o uso do OpenID Connect é necessário instalar e configurar seu módulo, também são necessárias as credenciais[4] do Google para autenticar o usuário. Além de outras configurações no OpenStack para exibir a nova opção. Para análise do tráfego, foi utilizada a ferramenta `tcpdump`[5] com escuta na interface externa do controlador. As coletas foram realizadas dez vezes a fim de assegurar que o fluxo de requisições segue um mesmo padrão, tanto quanto a natureza dos dados como das partes envolvidas.

6. ANÁLISE DE SEGURANÇA

Para a análise de segurança, foram levados em conta os seguintes critérios:

- Cifragem dos dados;
- Utilização de um IdP Externo; e
- Acesso administrativo não autorizado ou a outros projetos no Openstack.

Os critérios foram definidos de acordo com as vulnerabilidades levantadas nos trabalhos relacionados, na Seção 4. Para o critério Cifragem dos Dados, foi realizada uma escuta entre todos os canais de comunicação: OpenStack (SP) e IdP; SP e UA; e entre UA e IdP. Foi possível observar que o OpenStack já utiliza soluções contra algumas vulnerabilidades, como o CSRF, mesmo que não utilize TLS para proteger o canal SP – UA. Por não utilizar TLS nesse canal, os dados do usuário podem ser interceptados. Também foi possível observar que não existe tráfego direto entre o SP e o IdP, pois este tráfego é redirecionado pelo UA. Já o canal UA – IdP é protegido com TLS e não apresentou potenciais problemas. Também foi encontrada uma falha no SP, com análise do código, e permite que um atacante use o *id_token* de outro usuário, uma vez que não o verifica na autenticação.

A utilização de um IdP externo tem impacto maior na privacidade, uma vez que os IdPs podem rastrear a atividade do

4 -https://console.cloud.google.com/apis/credentials
5 -http://www.tcpdump.org

usuário, como é o caso do Google. Portanto é necessário avaliar a política de privacidade e verificar se não fere a política interna da organização.

O acesso administrativo não autorizado, ou a outros projetos, pode ser resolvido através do mapeamento do usuário no OpenStack. Mesmo que o Google não recicle os identificadores OpenID, é possível que outro IdP o faça. Dessa forma o usuário deve ser mapeado de acordo com outras informações, como e-mail e nome, além do identificador.

7. CONSIDERAÇÕES E TRABALHOS FUTUROS

A análise realizada através dos critérios estabelecidos reforça as afirmações que as nuvens computacionais, quanto a segurança, herdam os aspectos de segurança das tecnologias que empregam e somam-se novos aspectos oriundos da sua integração com seu software de integração. Neste sentido, percebe-se que tecnologias, como TLS, trazem toda uma herança de questões de segurança e atenção quanto a vulnerabilidades. Contudo, o uso de IdPs externos, ao mesmo tempo que terceiriza a segurança de parte do processo de autenticação, também inclui novos aspectos.

Embora exista a versão mais nova do OpenStack, o Mitaka, suas APIs não se mostraram compatíveis, inicialmente, com o OpenID Connect e carecem de um estudo mais aprofundado para implementação e para verificar se as vulnerabilidades encontradas ainda persistem.

Por fim, entende-se que a análise aqui descrita cobre apenas uma parte de vários aspectos de segurança previstos em normas e procedimentos de segurança [13][14].

REFERÊNCIAS

[1] P. Mell e T. Grance, “The NIST definition of cloud computing”, 2011.

[2] S. Bonner, C. Pulley, I. Kureshi, V. Holmes, J. Brennan, e Y. James, “Using OpenStack to improve student experience in an H.E. environment”, in *Science and Information Conference (SAI), 2013*, 2013, p. 888–893.

[3] M. R. Abid, I. F. Fihri, H. Mousannif, M. Bakhouya, C. El Amrani, M. Aissaoui, M. D. El Ouadghiri, A. Haqiq, A. Hayar, e M. Essaaidi, “MarUnivCloud: Towards a Moroccan inter-University Cloud”, in *2014 Second World Conference on Complex Systems (WCCS)*, 2014.

[4] X. You e Y. Zhu, “Research and design of Web Single Sign-On scheme”, in *2012 IEEE Symposium on Robotics and Applications (ISRA)*, 2012, p. 383–386.

[5] D. W. Chadwick, K. Siu, C. Lee, Y. Fouillat, e D. Germonville, “Adding Federated Identity Management to OpenStack”, *J. Grid Comput.*, vol. 12, nº 1, p. 3–27, dez. 2013.

[6] I. S. Sette e C. A. G. Ferraz, “Integrating Cloud Platforms to Identity Federations”, in *2014 Brazilian Symposium on Computer Networks and Distributed Systems (SBRC)*, 2014.

[7] M. Urueña, A. Muñoz, e D. Larrabeiti, “Analysis of privacy vulnerabilities in single sign-on mechanisms for multimedia websites”, *Multimed. Tools Appl.*, vol. 68, nº 1, 2014.

[8] R. H. Khan, J. Ylitalo, e A. S. Ahmed, “OpenID authentication as a service in OpenStack”, in *7th International Conference on Information Assurance and Security (IAS)*, 2011.

[9] L. Lynch, “Inside the Identity Management Game”, *IEEE Internet Comput.*, vol. 15, nº 5, p. 78–82, set. 2011.

[10] B. van Delft e M. Oostdijk, “A Security Analysis of OpenID”, in *Policies and Research in Identity Management*, E. de Leeuw, S. Fischer-Hübner, e L. Fritsch, Orgs. Springer Berlin Heidelberg, 2010, p. 73–84.

[11] F. Yang e S. Manoharan, “A security analysis of the OAuth protocol”, in *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, 2013, p. 271–276.

[12] W. Li e C. J. Mitchell, “Analysing the Security of Google’s implementation of OpenID Connect”, *ArXiv150801707 Cs*, ago. 2015.

[13] C. Alliance, “Security guidance for critical areas of focus in cloud computing v3. 0”, *Cloud Secur. Alliance*, 2011.

[14] C. Miers, M. S. Jr, T. Carvalho, G. Koslovski, M. Rojas, B. Rodrigues, and L. H. Iwaya, “Análise de Segurança para Soluções de Computação em Nuvem”, SBRC 2014 Minicursos, 2014.