Aplicação de SDN no gerenciamento de perfis de usuário em dispositivos de rede

Christiano M. Costa[1], Inácio C. Alves[1], Jean M. S. Maciel[1], Wellington Albano[1]

[1]Departamento de Computação
Instituto Federal do Ceará – Campus Maracanaú
Av. Parque Central S/N - Distrito Industrial I, Maracanaú, CE

christianomachado10@gmail.com, {inacioalves,jeanmdsm,wellington}@ifce.edu.br

Resumo. *As Redes Definidas por Software (Software Defined Network - SDN) fazem a divisão do plano de controle e do plano de encaminhamento, permitindo que haja uma visão global da rede, proporcionando uma rede programável de alto nível. Isso pode ser muito útil para permitir que o controle de acesso à rede seja gerenciado associando os fluxos dos dispositivos a perfis de usuários. Este artigo propõe uma ferramenta de gerenciamento de perfis de usuário para a autorização de fluxos de dispositivos utilizando redes definidas por software. Essa ferramenta foi implementada sobre o controlador OpenFlow POX. No artigo descrevemos os experimentos realizados e os resultados obtidos, que mostram a eficácia e a facilidade de utilização do mecanismo.*

Abstract. *Software Defined Networks (SDN) isolate the control plane from the data plane, allowing a global view of the network and providing a high level programmable network. This can be very useful for managing network access control by associating device flows to user profiles. This paper proposes a tool for managing user profiles by authorizing device flows using software-defined networks. This tool has been implemented on the OpenFlow controller POX. In this paper we describe the experiments performed and the results obtained, which show the efficacy and the ease of use of the mechanism.*

1. Introdução

No âmbito de redes de computadores, a segurança dos dados sempre foi um grande desafio. Um dos problemas relacionados à segurança está nos privilégios recebidos pelos usuários participantes da rede, que podem estar associados ao dispositivo utilizado. Mesmo em redes mais simples encontra-se uma grande quantidade de dispositivos utilizando algum recurso da rede, sendo necessário que os administradores tenham políticas de segurança de fácil implementação.

Soluções existentes utilizam autenticação e autorização do usuário para a concessão de privilégios, como é feito com o uso de *login* e senha e com a gerência de permissões dos diversos arquivos relacionados aos usuários e grupos em sistemas Linux. A autenticação pode ser feita também com a utilização de um servidor de autenticação, de acordo com o padrão 802.1x. Em Redes Definidas por *Software* (*Software Defined Networks* - SDN) [Azodolmolky 2013, McKeown et al. 2008], o plano de controle de decisão é removido do dispositivo de comutação e levado para um controlador logicamente centralizado, trazendo, assim, alta programabilidade na rede. Com a utilização de SDN

é possível gerenciar a autorização de fluxos de dispositivos. Daí, privilégios relativos à participação na rede podem ser atribuídos a dispositivos que se encaixem em perfis definidos pelo administrador.

O objetivo deste trabalho é demonstrar o uso de ferramentas SDN para o controle de acesso de dispositivos de acordo com o perfil do usuário, de forma a facilitar a gerência da rede. Através da utilização de um *switch* SDN e de um controlador, os fluxos são controlados baseando-se em um banco de dados que relaciona dispositivos a perfis de usuário. Através de uma interface gráfica, mesmo com pouco conhecimento de gerenciamento de redes será possível gerenciar a autorização de usuários, inicialmente vinculando um dispositivo a esse usuário e, em seguida, atribuindo privilégios para os fluxos dos dispositivos desse usuário. Neste artigo será apresentada uma prova de conceito do controle de acesso através de um controlador SDN.

O artigo está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados. A Seção 3 descreve os materiais e metódos e o cenário utilizado. A Seção 4 apresenta os experimentos realizados com a ferramenta desenvolvida. Em seguida, mostramos os resultados obtidos na Seção 5 e as conclusões na Seção 6.

2. Trabalhos relacionados

Um dos avanços tecnológicos na área de redes foi a criação do paradigma de Redes Definidas por Software (*Software Defined Networking - SDN*) [Azodolmolky 2013]. O paradigma SDN traz como ideia chave a separação das funções de controle e de encaminhamento. Esta divisão permite maior flexibilidade às funções de controle, enquanto o *hardware* de encaminhamento não sofre alteração. Dessa forma, surge o conceito de redes programáveis, que permite que o plano de controle da rede seja facilmente configurado. A divisão dos planos de controle e de encaminhamento é ilustrada na Figura 1. O controlador, logicamente centralizado, é separado dos *hardwares* destinados à comutação de pacotes. O comutador consulta o controlador para receber informações quanto ao encaminhamento ou descarte desses pacotes. A programabilidade é realizada através de *softwares* que executam sobre o controlador e que permitem que o operador possa definir, de maneira simples, as ações que serão tomadas, baseando-se nos fluxos recebidos.

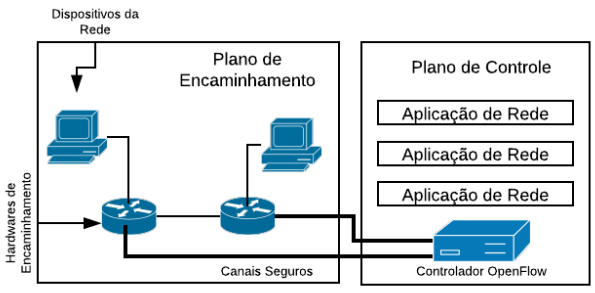


Figura 1. Separação dos planos de controle e de encaminhamento proposta pelo paradigma de Redes Definidas por *Software*

O mecanismo AuthFlow [Mattos and Duarte 2014] apresenta uma proposta de autenticação, adotando o padrão IEEE 802.1X e o *Extensible Authentication Protocol*

(EAP). Este mecanismo é composto por três nós essenciais, o controlador OpenFlow, o autenticador e o servidor RADIUS. A aplicação que executa sobre o controlador OpenFlow é a responsável por bloquear ou autorizar uma estação suplicante, dependendo do resultado da autenticação dessa estação no autenticador. A autorização ou bloqueio da estação é feita no nível do enlace, devido ao padrão IEEE 802.1X. No presente artigo, em vez de utilizar autenticação baseada nas estações, realizamos a autorização de fluxos baseada em perfis de usuário, em que cada perfil pode possuir um ou mais dispositivos associados.

Role-Based Access Control Models (RBAC) [Sandhu et al. 1996] trata de modelos de controle de acesso baseado em papéis. Neste modelo, os usuários são associados a papéis, que por sua vez possuem permissões. Assim, são definidos três tipos de conjuntos, a saber: os de usuários, os de papéis e os de permissões. Cada papel pode possuir um subconjunto das permissões e cada usuário pode ser associado a um ou mais papéis. Se um usuário não está associado com papel algum, então este usuário não possui autorização.

Em [Kamboj and Raj 2016] o modelo RBAC é usado em uma rede OpenFlow. No entanto, cada dispositivo de usuário na rede é tratato como um papel. Neste caso, a proposta muito se assemelha a um modelo que utiliza lista de controle de acesso (*Access Control List, ACL*) de modo que as permissões são definidas diretamente para o dispositivo em si.

Neste artigo não temos como objetivo a autenticação do dispositivo ou do usuário, mas realizar a autorização para a liberação dos fluxos dos dispositivos de acordo com o perfil do usuário. Assim, um usuário pode possuir diversos dispositivos e todos estarão sujeitos às mesmas políticas de autorização.

3. Materiais e métodos

Nesta seção apresentaremos as ferramentas utilizadas para o desenvolvimento do trabalho, assim como o cenário que foi montado. As ferramentas de software foram controlador POX [NOXRepo 2012], Python[Rossum 1991], MySQL[MySQL 1995], Open vSwitch [OpenVSwitch 2009], o emulador Mininet [Mininet 2013] e o protocolo OpenFlow [D. Marschke et al 2015], utilizados em conjunto para permitir que o administrador gerencie o controle de privilégios na rede SDN de um ou mais perfis de usuários de forma prática e fácil.

O *testbed*, descrito mais adiante, foi montado com a placa HCIPC M503-1 LAN-HCM52L26B, que faz as vezes de comutador, e com três computadores para as funções de estações e de controlador. A placa HCIPC M503-1 LAN-HCM52L26B foi utilizada para hospedar o comutador programável Open vSwitch. Seu processador é Intel Atom D525 Dual core 1.8 GHz, com 4 GB de memória RAM e seis portas LAN 10/100/1000 Mbps.

3.1. Cenários

No primeiro cenário foram instanciadas na mesma máquina física, com processador Intel Core i5, 8 GB de memória RAM e sistema operacional Linux Ubuntu 17.10 64 bits, e com auxílio do Mininet, quatro hosts virtuais e também um comutador virtual Open vSwitch. No segundo cenário foram utilizadas três máquinas físicas com processador Intel Core i5, 8 GB de memória RAM cada, duas com Linux Ubuntu 16.04 LTS virtualizado

e uma com Linux Ubuntu 17.10 64 bits. Também foi utilizada a placa HCIPC M503-1 LAN-HCM52L26B com Open vSwitch instanciado. Em ambos os cenários o controlador OpenFlow POX e o banco de dados utilizados foram instanciados na máquina física com Linux Ubuntu 17.10 LTS.

De modo geral, a ferramenta proposta funciona da seguinte maneira. Quando um novo dispositivo ingressa na rede OpenFlow, implementada por um comutador virtual Open vSwitch 2.8.1, o controlador POX[1] é notificado. No primeiro momento, as informações do dispositivo são coletadas e avaliadas através de uma base de dados. Caso o dispositivo esteja associado a um perfil de usuário, o controlador POX verifica se tal usuário possui privilégios na rede, dado o horário atual do acesso. Caso contrário, o dispositivo é associado a um perfil padrão de usuário, que não possui privilégios. O fluxograma mostrado na Figura 2 representa as etapas necessárias para que o dispositivo participe da rede.

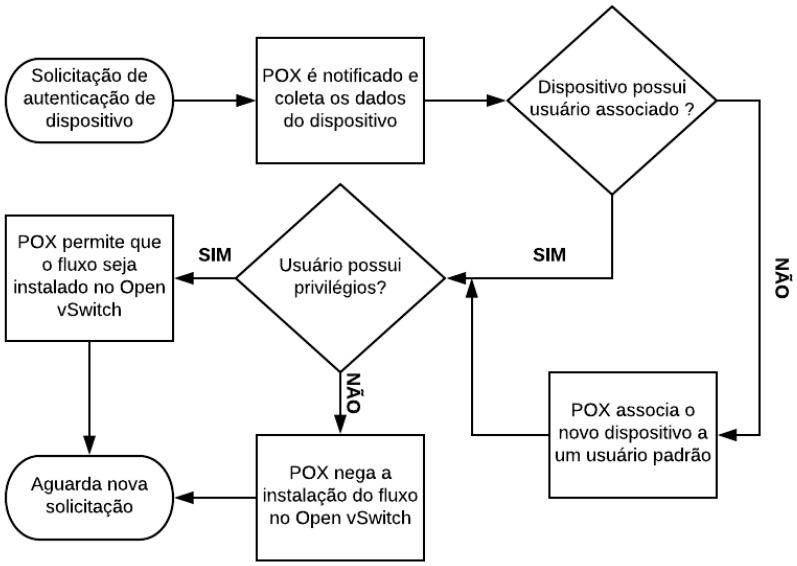


Figura 2. Fluxograma de ações tomadas pelo controlador POX

A chegada de um novo dispositivo requer que este seja associado a um perfil de usuário. Essa tarefa é realizada por uma aplicação desenvolvida para o gerenciamento do controle de acesso à rede, simples e de fácil entendimento do administrador. As telas principais são mostradas nas Figuras 3 e 4.

[1]Alguns scripts do controlador POX foram levemente modificados para interagir com o banco de dados MySQL.

(a) (b)

Figura 3. Telas da aplicação de gerência da rede: (a) Registro de um novo perfil de usuário; (b) Busca por um novo perfil criado

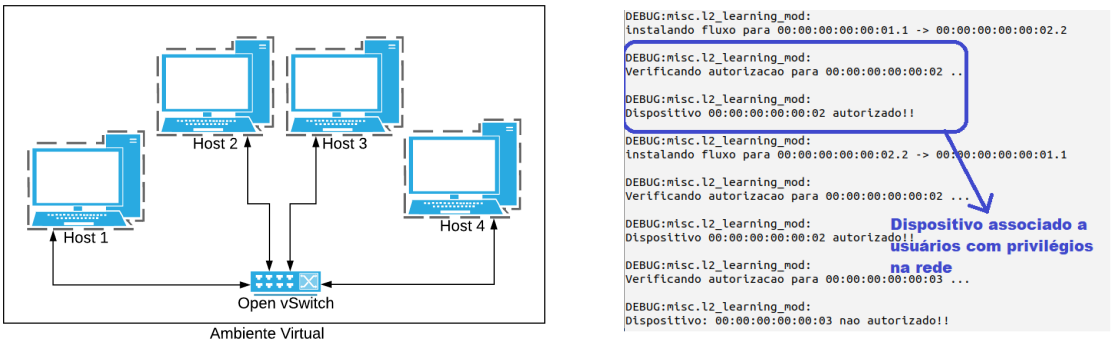
(a) (b)

Figura 4. Telas da aplicação de gerência da rede: (a) Modificação dos privilégios de acesso do usuário; (b) Associação de um dispositivo a um perfil de usuário

4. Experimentos realizados

Os experimentos realizados consistem em testar a eficácia e a facilidade do gerenciamento de perfis de usuários em dispositivos de rede. Dessa forma, o administrador pode conceder privilégios a um determinado perfil que poderá ser associado a diversos dispositivos. Cada fluxo gerado por um dispositivo é analisado pelo controlador. O encaminhamento de pacotes deste fluxo é liberado se, e somente se, o usuário ao qual este dispositivo gerador do fluxo está associado possuir privilégios na rede, dado o momento da solicitação. Caso contrário, os pacotes serão descartados.

O primeiro cenário foi montado da seguinte forma. Quatro *hosts* virtuais e um comutador virtual foram instanciados em um computador Linux Ubuntu 17.10 LTS 64 bits, os quais chamaremos de H1, H2, H3, H4 e SW, respectivamente. H1 se comunica diretamente com H4, enquanto que H2 se comunica com H3. Para geração de tráfego foi utilizada a ferramenta `iperf`, gerando fluxos de comunicação entre os *hosts*. Estes passam por SW que, por sua vez, faz as devidas solicitações ao controlador POX, este instanciado na mesma máquina física. O controlador, que possui uma aplicação de consulta ao SGBD MySQL, decide se autoriza ou não a instalação do fluxo referente ao dispositivo requisitante. O esquema estrutural descrito é mostrado na Figura 5.



(a) (b)

Figura 5. Encaminhamento de tráfego liberado entre os hosts virtuais (H1, H2, H3 e H4): (a) Visão esquemática do ambiente de teste; (b) Saída do controlador POX durante a comunicação entre os hosts.

Os *hosts* virtuais foram associados a perfis de usuários distintos de forma que, inicialmente, todos possuíam privilégios de acesso à rede. Portanto, os pedidos de instalação de fluxo são aceitos. Com o passar do tempo de comunicação, H3 teve seu privilégio removido por um intervalo de tempo, portanto não poderia mais acessar a rede e se comunicar com H4 durante esse período. Os tráfegos gerados foram capturados e analisados com ajuda das ferramentas `tcpdump` e Wireshark [Combs 2018].

O segundo cenário foi montado em um *testbed* com uma pequena rede formada por cinco computadores reais, computador A (PCA), computador B (PCB) e computador C (PCC), um comutador com Open vSwitch 2.8.1 e um servidor DHCP. O PCC contém o controlador POX e a aplicação de gerência do administrador da rede.

Inicialmente, os computadores não foram associados a nenhum perfil de usuário. Portanto, o controlador POX, PCC, registrou-os com o perfil de usuário padrão no banco de dados. Neste experimento, o perfil de usuário padrão não possuía nenhum privilégio, de forma que os dispositivos foram incapacitados, em um primeiro momento, de trafegar pacotes na rede. Num segundo momento, os PCA e PCB foram associados a perfis de usuários distintos. O perfil associado ao PCA possuía, como privilégio, o acesso permitido de segunda-feira a quarta-feira. Para o perfil associado ao PCB, foi concedido privilégio para o acesso à rede de quarta-feira a sábado. Assim, os pacotes gerados pelo PCA com destino ao PCB, na segunda-feira e terça-feira, foram descartados, pois o PCB não possuía privilégios para utilização da rede nesse dia. O mesmo resultado foi obtido com pacotes gerados pelo PCB ao PCA de quinta-feira a sábado. Apenas na quarta-feira, quando ambas as máquinas tinham seus fluxos autorizados, a comunicação entre esses dispositivos era possível.

5. Resultados

Quando um novo dispositivo conecta-se à rede, ele é associado, automaticamente, a um perfil de usuário padrão, que é definido na criação da rede. O administrador tem total poder para criar, modificar, alterar e remover um perfil de usuário. Ele também pode associar a qualquer dispositivo, registrado no banco de dados, um perfil de usuário adequado. O controlador OpenFlow baseia-se nos privilégios dos perfis de usuários para autorizar um determinado fluxo. O dispositivo originador do pacote só poderá ter seu

fluxo instalado caso o perfil ao qual está associado possua privilégios mínimos definidos pelo administrador. Caso contrário, seu pacote é automaticamente descartado.

A Figura 6 mostra o comportamento do tráfego durante a comunicação entre os dispositivos. Na Figura 6a, podemos ver o tráfego entre dois dispositivos cujos fluxos estão sempre autorizados. Na Figura 6b, observamos o fluxo entre dois dispositivos em uma situação onde o fluxo de um deles teve a autorização removida durante um intervalo de tempo.

Nos experimentos realizados, percebeu-se que, devido à maneira que o algoritmo de *learning switch* foi implementado no controlador, durante o processo de bloqueio ou desbloqueio de um usuário, a instalação ou remoção dos fluxos sofre um pequeno atraso, como pode ser visto pela linha tracejada na Figura 6b. No entanto, este atraso não compromete o funcionamento da rede.

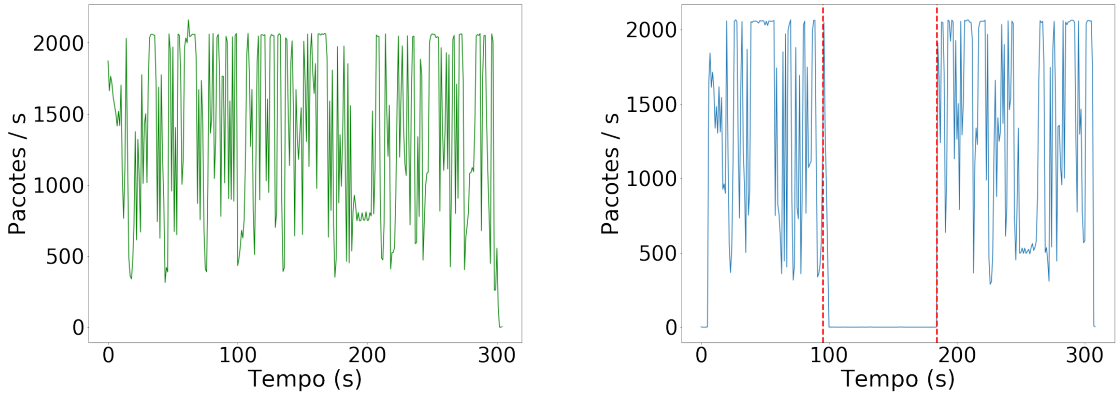


Figura 6. Gráfico de tráfego entre os dispositivos: (a) dispositivos autorizados durante todo o processo; (b) um dos dispositivos é bloqueado pelo administrador por um período de tempo.

6. Conclusão

O paradigma de redes definidas por *software* (SDN) apresenta muitos benefícios, principalmente em relação à utilização de redes programáveis. As características de redes SDN permitem que diversas aplicações sejam criadas para facilitar e melhorar o gerenciamento da rede, de modo que um usuário leigo possa gerenciar sua rede sem maiores dificuldades. A ferramenta apresentada neste artigo traz uma aplicação para a autorização de dispositivos, em redes SDN, utilizando perfis de usuários. A autorização dos fluxos dos usuários pode ser definida pelo gerente da rede, associando-os a perfis que podem ser criados, modificados ou deletados.

Os resultados obtidos mostram que a ferramenta pode ser utilizada na autorização do uso da rede por dispositivos de usuários associados a determinados perfis, enquanto que dispositivos associados a perfis sem privilégios terão o acesso à rede bloqueado. Como trabalhos futuros, pretende-se aplicar o mecanismo a redes sem fio, bem como implantar o mesmo em redes domésticas e aprimorar o gerenciamento dos perfis de usuário.

Agradecimentos

Os autores agradecem o programa PIBIC/IFCE/CNPq pela concessão de bolsa de estudo que possibilitou o desenvolvimento do projeto que originou o presente artigo.

Referências

Azodolmolky, S. (2013). *Software Defined Networking with OpenFlow*. Packet Publishing Ltd.

Combs, G. (2018). Disponível em < https://www.wireshark.org/ >. Acesso em: 10 de abril de 2018.

D. Marschke et al, J. Doyle, P. M. (2015). *SDN: Anatomy of Openflow*. Lulu Publishing Service rev.

Kamboj, P. and Raj, G. (2016). Analysis of role-based access control in software-defined networking. In Pant, M., Deep, K., Bansal, J. C., Nagar, A., and Das, K. N., editors, *Proceedings of Fifth International Conference on Soft Computing for Problem Solving*, pages 687–697, Singapore. Springer Singapore.

Mattos and Duarte (2014). AuthFlow Um Mecanismo de Autenticação e Controle de Acesso para Redes Definidas por Software. 32o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, p. 661–674.

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74.

Mininet (2013). Disponível em < http://mininet.org/ >. Acesso em: 10 de abril de 2018.

MySQL (1995). Sistema de Gerenciamento de Banco de Dados MySQL. Disponível em <https://dev.mysql.com/doc/refman/5.7/en/>. Acesso em: 9 de novembro de 2017.

NOXRepo (2012). Nox repo. Disponível em <http://www.noxrepo.org>. Acesso em: 15 de dezembro de 2017.

OpenVSwitch (2009). Disponível em < http://www.openvswitch.org/ >. Acesso em: 9 de novembro de 2017.

Rossum, G. V. (1991). Python. Disponível em <https://www.python.org/>. Acesso em: 12 de setembro de 2017.

Sandhu et al. (1996). Sandhu ', R. S., Coyne, E. J., Feinstein, H. L., Youman, C. E. and Sandhu, R. (1996). Role-Based Access Control Models. IEEE Computer, v. 29, n. 2, p. 38–47.