AuthFlow: Um Mecanismo de Autenticação e Controle de Acesso para Redes Definidas por *Software*

Diogo Menezes Ferrazani Mattos e Otto Carlos Muniz Bandeira Duarte

[1]Grupo de Teleinformática e Automação
Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro – RJ – Brasil

Resumo. *As Redes Definidas por Software (Software Defined Network – SDN) desacoplam o controle do encaminhamento de dados, oferecendo alta programabilidade e uma visão global da rede. A adoção dessa tecnologia é crescente em redes empresariais, centros de dados e infraestruturas críticas como as redes elétricas inteligentes. No entanto, prover segurança nessas redes de nova geração é um desafio. Este artigo apresenta as principais ameaças de segurança às redes definidas por software e propõe o AuthFlow, um mecanismo de autenticação e controle de acesso de estações finais baseado na credencial da estação. O mecanismo proposto apresenta duas contribuições principais: i) autentica a estação diretamente na camada de enlace em uma rede OpenFlow, o que introduz uma baixa sobrecarga de controle e assegura um controle de acesso refinado; ii) usa a credencial de autenticação para realizar o controle de acesso de acordo com o nível de privilégio de cada estação, através da associação da credencial ao conjunto de fluxos pertencentes à estação. Um protótipo do mecanismo proposto foi implementado sobre o POX, controlador OpenFlow. Os resultados mostram que a proposta bloqueia o acesso de estações não autorizadas, mesmo no cenário em que uma estação tem a sua permissão de acesso revogada. Por fim, comprova-se que cada estação pode ter um nível diferente de acesso aos recursos da rede em função da sua credencial.*

Abstract. *Software Defined Networks are been widely adopted by enterprise networks. Providing security features in these next generation networks, however, is a challenge. In this paper, we present the main security threats in Software Defined Networks and we propose AuthFlow, an authentication and access control mechanism based on host credentials. The main contributions of the proposed mechanism are twofold: i) AuthFlow authenticates hosts directly at the data link layer in an OpenFlow network, which introduces a low overhead and ensures a fine-grained access control, ii) AuthFlow uses the authentication credential to perform access control according to the privilege level of each host, through the association of the host credential with the set of flows that belongs to the host. A prototype of the proposed mechanism was implemented over POX, an OpenFlow controller. The results show that the proposed mechanism blocks access from unauthorized hosts, even in the scenario where a host has its access authorization revoked. Finally, we show that each host can have different levels of access to network resources according to their authentication credential.*

1. Introdução

Prover segurança em redes é uma necessidade crescente em redes empresariais, em redes de centro de dados para computação em nuvem e em redes que constituem infraestruturas críticas como as redes elétricas inteligentes. As principais dificuldades para

Este trabalho foi realizado com recursos da CNPq, CAPES, FINEP, FUNTTEL e FAPERJ.

garantir um alto nível de segurança nas redes [Nayak et al., 2009, Kreutz et al., 2013] são a variedade de equipamentos de redes, como comutadores, roteadores, *middleboxes*, entre outros; e o fato de as estações finais que se conectam à rede nem sempre serem confiáveis e, até mesmo, poderem apresentar diversas vulnerabilidades. Assim, a implantação de políticas de segurança requer do operador da rede configurações manual dos equipamentos de acordo com o padrão e as funcionalidades de cada um. Além disso, as estações finais também devem ser autenticadas para garantir o acesso à rede somente às estações que apresentam credenciais válidas e autorizadas.

O paradigma de Redes Definidas por *Software* (*Software Defined Networks* – SDN) desacopla o controle do encaminhamento de dados, oferecendo alta programabilidade do controle e uma visão global da rede. A adoção desta tecnologia permite desenvolver, de forma logicamente centralizada, políticas integradas de segurança [Levin et al., 2012] e, assim, facilita a solução de problemas complexos de segurança em rede. A Interface de Programação de Aplicação (API) OpenFlow [McKeown et al., 2008] é a implementação de maior sucesso de uma rede definida por *software*. O controlador OpenFlow, implementando em *software*, centraliza o plano de controle. O plano de encaminhamento é executado por comutadores de alto desempenho compatíveis com o OpenFlow. Contudo, esse novo paradigma apresenta algumas limitações quanto à segurança da rede, pois um componente com comportamento malicioso pode comprometer o funcionamento de toda a rede, por exemplo, realizando um ataque de negação de serviço no controlador da rede. Dessa forma, o controle de acesso a essas redes é necessário para garantir a segurança. Tanto a autenticação das estações que têm acesso à rede, quanto o nível de privilégio atribuído a cada estação são essenciais para garantir a segurança da rede definida por *software*.

Este artigo propõe o AuthFlow, um mecanismo de autenticação e controle de acesso para redes definidas por *software*. O mecanismo AuthFlow apresenta duas contribuições principais: (i) a autenticação das estações finais diretamente na camada de enlace; e (ii) a associação das credenciais de acesso de uma estação aos fluxos pertencentes a essa estação. A autenticação da estação final na camada de enlace é realizada através do padrão IEEE 801.X que garante que as informações de autenticação sejam trocadas de forma padronizada entre a estação e o autenticador e, portanto, não requer qualquer alteração nas estações finais. O mecanismo de autenticação encapsula as mensagens no formato *Extensible Authentication Protocol* (EAP), o que permite a adoção de diferentes métodos de autenticação. A autenticação do mecanismo AuthFlow é direto na camada de enlace e, portanto, tem a vantagem prover uma baixa sobrecarga de controle quando comparada a uma autenticação na camada rede, ou na camada de aplicação, que dependem da atribuição de um IP à estação que está se autenticando e dependem da troca de informações da aplicação para a autenticação. Outra vantagem do mecanismo proposto é o provimento de um controle de acesso refinado, já que o AuthFlow permite definir políticas de acesso por fluxo para cada estação de acordo com a credencial da estação. Assim, o controle de quais serviços uma estação pode acessar passa a ser realizado de acordo com as suas credenciais e, não mais, com seus endereços IP ou MAC. O mecanismo AuthFlow é composto por uma aplicação que executa sobre o POX, o controlador OpenFlow, e possui mais dois componentes principais: o autenticador e o servidor RADIUS. O autenticador recebe as mensagens no padrão IEEE 802.1X e valida as credenciais da estação com o servidor RADIUS.

As principais propostas para prover segurança às redes definidas por *software* buscam desenvolver módulos de segurança no controlador que facilitam o desenvol-

vimento de novas aplicações seguras [Porras et al., 2012, Shin et al., 2013]. Por outro lado, outras propostas de autenticação de estações finais em SDN consideram que a autenticação deve ser feita somente após a estação receber um endereço IP temporário e ser redirecionada a um sítio *Web*, onde deve apresentar suas credenciais [Nayak et al., 2009, Casado et al., 2007]. Contudo, essas propostas estão sujeitas a ataques de falsificação de endereço, além de introduzirem uma maior sobrecarga de controle quando comparadas ao AuthFlow. Outras propostas descrevem algumas ameaças de segurança das redes definidas por *software* e indicam possíveis direções para solucionar essas ameaças [Kreutz et al., 2013, Heller et al., 2012]. Considerando as principais ameaças às redes definidas por *software*, um protótipo do mecanismo AuthFlow foi desenvolvido e avaliado no ambiente de experimentação *Future Internet Testbed with Security* (FITS) [Guimarães et al., 2013][1]. A eficiência do mecanismo de controle de acesso proposto é evidenciada nos experimentos que mostram o bloqueio de estações ao tentar usar a rede, tanto no caso em que a estação não está autenticada, quanto no caso de a estação ter sua autenticação revogada. Os resultados da avaliação do protótipo mostram ainda que as estações finais têm visões diferentes da rede, liberando ou bloqueando acesso a determinados serviços, dependendo do nível de privilégio que cada estação tem de acordo com a sua credencial de acesso.

O restante do artigo está organizado da seguinte forma. A Seção 2 discute os principais conceitos do paradigma de Redes Definidas por *Software* e suas limitações de segurança. O mecanismo AuthFlow proposto é apresentado na Seção 3. A Seção 4 apresenta os resultados experimentais da avaliação do mecanismo proposto. Os trabalhos relacionados são discutidos na Seção 5. A Seção 6 conclui o artigo.

2. O Paradigma Redes Definidas por *Software*

O paradigma de Redes Definidas por Software (*Software Defined Networking* - SDN) [Casado et al., 2012] se baseia na separação das funções de controle, plano de controle, das funções de encaminhamento de quadros, plano de dados. A ideia chave da separação é prover maior flexibilidade às funções de controle enquanto o *hardware* especializado para comutar quadros a alta velocidade permanece inalterado. Logo, a SDN oferece uma alta programabilidade das funções de controle da rede em um comutador com alto desempenho de encaminhamento de quadros. O operador pode definir de maneira simples os fluxos e as ações sobre os fluxos através de uma interface de programa de aplicação [Guedes et al., 2012].

A Figura 1 ilustra uma rede definida por *software* com controle centralizado, separado dos elementos comutadores responsáveis pelo encaminhamento de pacotes. O controle da rede é executado por um *software* de propósito geral, denominado controlador de rede, sobre o qual se desenvolvem aplicações com propósitos específicos para o controle da rede. O controlador se comunica com os comutadores e, então, possui uma visão unificada de todo o estado da rede. Assim, uma das principais vantagens da abordagem SDN é a formação de uma visão global, unificada, do controle da rede facilitando a tomada de decisões sobre sua operação. A visão global centralizada torna a programação da rede mais fácil e simplifica a representação de problemas [Guedes et al., 2012]. O OpenFlow define que os elementos de encaminhamento ofereçam uma interface de programação de aplicação (*Application Programming Interface* - API) que permita um nó controlador centralizado estender as ações de controle e de acesso sobre a tabela utilizada pelos

[1]O FITS é uma rede de testes interuniversitária desenvolvida a partir da parceria de instituições brasileiras e europeias. Maiores informações em http://www.gta.ufrj.br/fits/.

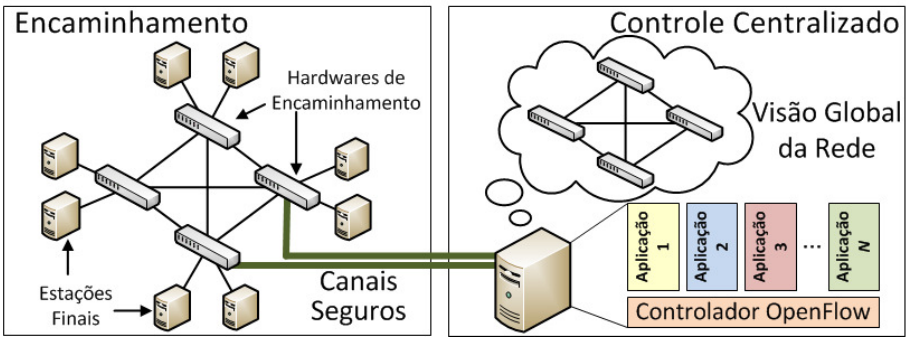


Figura 1. Rede Definida por *Software* separa as funções de encaminhamento e controle. O *hardware* de encaminhamento é controlado por aplicações centralizadas que têm uma visão global da rede.

componentes de encaminhamento para determinar o próximo destino de cada pacote encaminhado.

2.1. Ameaças de Segurança em Redes Definidas por *Software*

A visão global centralizada das redes definidas por *software* permite que a lógica de controle das aplicações de segurança seja mais completa e integrada do que as atuais [Shin et al., 2013] e, portanto, simplifica o tratamento de problemas complexos de segurança em rede. As aplicações de segurança em rede se valem do controlador centralizado para implementarem lógicas de definição de fluxos baseadas em estados e, também, segurança baseada em fluxos como, por exemplo, algoritmos de detecção de intrusão ou de anomalias. Contudo, a criação de aplicações de segurança em SDN é um desafio, pois a própria segurança de uma rede definida por *software* ainda é questionável [Kreutz et al., 2013]. Os desafios de segurança de uma rede definida por *software* se dividem em três categorias: negação de serviço, ausência de confiança entre componentes e vulnerabilidades de componentes.

Negação de Serviço pode ocorrer tanto no plano de dados quanto no plano de controle. No plano de dados, uma estação maliciosa que gere fluxos falsos pode exaurir tanto os recursos de banda, quanto os recursos de memória, ou tabela de fluxos, dos comutadores da rede. A negação de serviço no plano de controle pode ser alcançada em dois pontos distintos da rede: no controlador e na comunicação do controlador com os comutadores. É possível exaurir a capacidade de processamento do controlador de rede ao se enviar uma grande quantidade de pacotes com diferentes cabeçalhos. Isto acontece, pois todo pacote é analisado e um pacote com cabeçalho que não corresponde a nenhum fluxo já definido deve ser enviado ao controlador de rede. Assim, em um cenário em que um comutador envia uma quantidade atípica de novos cabeçalhos de pacotes para o controlador, este pode ter seus recursos de processamento exauridos e não ser capaz de responder a pedidos de novos fluxos em tempo hábil. Da mesma forma, a negação de serviço pode ser obtida quando o enlace de conexão entre o controlador e os comutadores na rede é intencionalmente congestionado. Caso não haja redundância ou banda suficiente no enlace que conecta os comutadores ao controlador, um comutador malicioso pode gerar tráfego suficiente para sobrecarregar esse enlace e, consequentemente, impedir a comunicação do controlador com os demais comutadores. A autenticação de estações e dos comutadores, usando *Secure Socket Layer* (SSL) e infraestrutura de chaves públicas (*Public Key Infrastructure* – PKI), é capaz de evitar esse tipo de ameaça, pois somente nós autorizados têm

acesso à rede e, no caso de identificação de um comportamento malicioso, a autenticação do nó pode ser revogada e o nó, expulso da rede.

Ausência de Confiança entre Componentes da Rede compromete uma Rede Definida por *Software*, pois as aplicações executadas sobre o controlador podem ter comportamentos maliciosos. Nesse caso, o controlador deve ser capaz de identificar quais são as aplicações confiáveis e quais são maliciosas. Assim, uma possível medida para aumentar a segurança nas aplicações executadas em uma SDN é o uso de mecanismos de atestação de aplicações e o estabelecimento de cadeias de confiança por atestação. Algumas propostas para prover segurança em SDN consideram o uso de um núcleo de segurança no próprio controlador para garantir a execução segura de aplicações, sem que uma aplicação interfira em outras ou execute ações proibidas na rede [Shin et al., 2013, Porras et al., 2012]. Por outro lado, a ausência de confiança também afeta o registro de ações que ocorreram na rede (*log*), já que o registro de ações, quando é feito, não apresenta nenhuma garantia de que a ação ocorreu e de que a aplicação que o registrou não tinha comportamento malicioso. Uma possível solução é a adoção de registro de ações por aplicação, assinados por cada aplicação, que por sua vez sejam atestadas e assinadas por desenvolvedores.

Vulnerabilidade de Componentes não é um desafio de segurança exclusivo desse novo paradigma de rede, mas torna-se mais crítico, pois uma vulnerabilidade em um nó controlador torna toda a rede vulnerável. Assim, são três as possíveis fontes de vulnerabilidades: comutadores, controlador e estações de gerenciamento. Uma vulnerabilidade em um comutador pode permitir que um atacante, que ganhe acesso a um comutador, exerça um ataque contra o plano de controle, a exemplo da falsificação de mensagens de outros comutadores para exaurir os recursos do controlador. Uma vulnerabilidade no controlador permite que um atacante altere o plano de controle ou, até mesmo, execute uma nova aplicação de controle da rede. Uma vulnerabilidade em uma estação de gerenciamento permite que o atacante exerça configurações no plano de controle diferentes das corretas. Medidas de prevenção a este tipo de ataque são a atestação das aplicações de controle, o uso de protocolos de certificação dupla entre estações de controle e aplicações e, por fim, a replicação das aplicações de controle para a tolerância a falhas e a intrusão.

Entre alguns dos principais desafios em segurança para as redes definidas por *software*, é possível destacar três características necessárias a essas redes: escalável, responsiva e disponível [Nayak et al., 2009]. Para prover tais características há o desafio da localização de controladores [Heller et al., 2012]. Nesse sentido, a localização e a quantidade de controladores replicados necessários a uma rede devem respeitar os requisitos de segurança, escala, disponibilidade e tempo de resposta da rede.

A autenticação, autorização e controle de acesso são primitivas essenciais em uma Rede Definida por *Software*. Essas primitivas, somadas à atestação e replicação de controladores, são a base de uma rede segura em que componentes maliciosos, sejam por vulnerabilidades em *software*, sejam pelo comportamento nocivo à rede, podem ser identificados e isolados do funcionamento normal da rede [Nagahama et al., 2012, Nayak et al., 2009, Shin et al., 2013].

3. O Mecanismo AuthFlow

A ideia principal do mecanismo AuthFlow é realizar a autenticação usando protocolos da camada de enlace, fazendo o mapeamento da identidade usada na autenticação em fluxos criados por uma dada estação autenticada. Para tanto, o mecanismo proposto usa o padrão IEEE 802.1X e o *Extensible Authentication Protocol* (EAP). O EAP encap-

sula as trocas de mensagens de autenticação entre a estação suplicante[2] e um servidor de autenticação RADIUS. O autenticador empregado no mecanismo AuthFlow é um processo que se comunica como uma aplicação OpenFlow, que executa sobre o controlador POX. A aplicação aceita ou bloqueia o tráfego de rede da estação suplicante, dependendo do resultado da autenticação entre o suplicante e o autenticador.

O mecanismo AuthFlow adota o padrão IEEE 802.1X, pois esse padrão especifica a autenticação diretamente na camada de enlace e, por ser um padrão bastante adotado, não requer modificações nas estações finais para a autenticação na rede. Assim, quando uma estação compatível com o padrão IEEE 802.1X inicia, ela também inicia a fase de autenticação através do envio da mensagem de `início` do IEEE 802.1X para o endereço reservado MAC *multicast* (`01:80:C2:00:00:03`), com o tipo Ethernet definido em `0x888E`. Dessa forma, o procedimento de autenticação de uma estação não depende de nenhum conhecimento prévio acerca da rede, nem mesmo da tradução de um endereço IP em um endereço MAC. Esse procedimento evita que a estação receba um IP temporário para só recebrer um IP definitivo dependendo do resultado da autenticação. O uso do padrão IEEE 802.1X facilita muito o processo de autenticação.

A seguir, discute-se a arquitetura e o funcionamento do mecanismo AuthFlow. O estudo de caso considerado é usar mecanismo proposto para a autenticação de roteadores virtuais em uma infraestrutura de rede pluralista híbrida Xen e OpenFlow. Contudo, a proposta não se limita a esse estudo de caso e pode ser usada, sem qualquer alteração, na autenticação de estações finais em uma rede OpenFlow. No estudo de caso considerado, as estações componentes da rede OpenFlow são máquinas virtuais que se comportam tanto como estações finais quanto como roteadores.

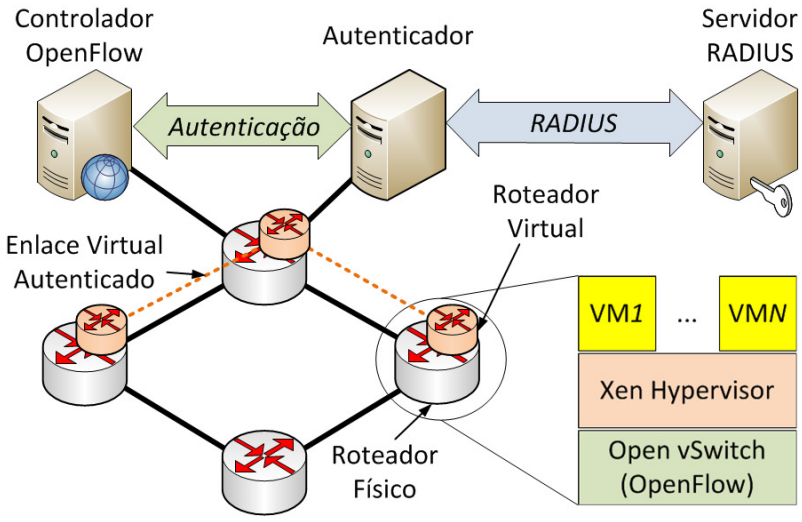


Figura 2. A arquitetura do mecanismo AuthFlow é composta por três nós essenciais: o controlador OpenFlow, o Autenticador e o Servidor RADIUS. Uma máquina virtual ao se autenticar na rede, envia um pacote IEEE 802.1X com a autenticação EAP. O Autenticador usa o conteúdo EAP do pacote IEEE 802.1X para autenticar o roteador virtual com o servidor RADIUS e comunica o resultado da operação ao controlador OpenFlow.

A arquitetura do mecanismo AuthFlow é composta de máquinas físicas hospedando máquinas virtuais, comutadores OpenFlow por *software*, um controlador POX, um

[2]Os nomes suplicante, autenticador e servidor de autenticação são definidos pelo padrão IEEE 802.1X.

autenticador e um servidor de autenticação RADIUS, conforme mostrado na Figura 2. As máquinas físicas e virtuais agem como roteadores e, então, são denominadas respectivamente roteadores físicos e roteadores virtuais. Os roteadores físicos são os nós com o sistema de virtualização Xen que hospedam os roteadores virtuais. O encaminhamento dos pacotes entre os roteadores físicos e virtuais é realizado por um comutador por *software* compatível com a API OpenFlow. No caso do mecanismo proposto, o comutador OpenFlow é um Open vSwitch[3] instanciado em cada roteador físico. O modelo de virtualização adotado no mecanismo proposto é o modelo híbrido Xen e OpenFlow usado no sistema XenFlow [Mattos et al., 2011, Mattos et al., 2013]. O controlador POX executa uma aplicação para manipular o encaminhamento de todos os pacotes, em especial, os pacotes[4] do padrão IEEE 802.1X. Os pacotes IEEE 802.1X são encaminhados diretamente para o autenticador. Autenticador é um cliente RADIUS que implementa o padrão IEEE 802.1X e repassa o conteúdo EAP para o RADIUS. O autenticador foi desenvolvido como uma versão adaptada do `hostapd`[5], que é um autenticador usado em redes sem fio. O `hostapd` foi modificado para informar à aplicação POX sobre a autenticação das redes virtuais. Assim, ao realizar a autenticação de uma rede, o `hostapd` envia uma confirmação de autenticação para o POX através de um canal seguro, criptografado e autenticado usando o esquema de distribuição de chaves públicas (*Public Key Infrastructure* – PKI) e o padrão SSL 3.0 (*Secure Socket Layer*). O servidor de autenticação é um servidor RADIUS que extrai as informações de autenticação do encapsuladas pelo EAP e valida as credenciais apresentadas pelos roteadores virtuais. Como o EAP permite o uso de diversos métodos de autenticação diferentes, o método adotado foi o MS-CHAP v2 [Zorn, 2000] que autentica o roteador virtual em uma base de dados usando como credenciais nome do usuário e senha. Para tanto, foi usada uma base de dados *Lightweight Directory Access Protocol* (LDAP). Contudo, o mecanismo de autenticação e a base de dados a serem usados não são essenciais para a descrição do mecanismo proposto, pois não interagem diretamente com a rede OpenFlow.

O mecanismo de autenticação AuthFlow funciona da seguinte maneira. Um roteador virtual envia um pedido de autenticação, no padrão IEEE 802.1X, e o controlador POX redireciona para o Autenticador. Em seguida, o Autenticador responde e a estação suplicante envia suas credenciais. O Autenticador verifica as credenciais de estação suplicante com o servidor RADIUS, executando o método de autenticação definido no EAP. Se as credenciais estão corretas, o Autenticador envia uma mensagem de `sucesso` para a estação suplicante e envia uma mensagem de autorização e confirmação de autenticação para o POX, através do canal seguro SSL, identificando a estação suplicante e que a autenticação foi bem-sucedida. Após o estágio de autenticação, o controlador POX permite que a estação suplicante acesse os recursos da rede. Em caso de revogação das credenciais da estação suplicante, o Autenticador comunica ao POX, que imediatamente suspende o acesso da estação à rede.

O controle de acesso do mecanismo proposto funciona através da liberação e bloqueio de enlaces. Assim, ao iniciar uma rede OpenFlow que empregue o AuthFlow, todos os enlaces da rede estão bloqueados para a comunicação, inclusive os enlaces que interconectam os nós comutadores OpenFlow, ou seja, os enlaces pertencente ao núcleo da rede. Nesse caso, esses enlaces não necessitam realizar o processo de autenticação para ter o seu tráfego liberado. Para tanto, o mecanismo AuthFlow realiza a descoberta da topologia

[3]http://www.openvswitch.org/.

[4]A nomenclatura de pacote foi usada, pois é mais genérica e a API OpenFlow tem acesso é às camada de enlace Ethernet até a camada de transporte.

[5] http://hostap.epitest.fi/hostapd/.

do núcleo da rede através de pacotes *Link Layer Discovery Protocol* (LLDP) encaminhados enlace a enlace. Como o controlador gera e verifica cada pacote LLDP transmitido no núcleo da rede, o controlador é capaz de identificar quais enlaces estão entre comutadores OpenFlow e quais são enlaces finais ligados a estações finais. Os pacotes LLDP gerados pelo controlador são marcados unicamente para evitar ataques de repetição (*replay*) ou de falsificação (*spoofing*).

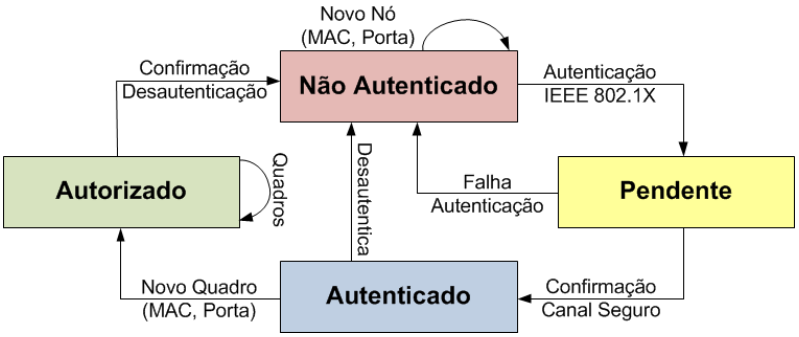


Figura 3. O controle de acesso do mecanismo AuthFlow ocorre em quatro estados. i) Não autenticado, quando a estação não iniciou o processo de autenticação; ii) Pendente, enquanto o processo de autenticação ocorre; iii) Autenticado, quando a estação já finalizou a autenticação com sucesso; iv) Autorizado, quando a estação está autorizada a usar a rede.

O controle de acesso no mecanismo AuthFlow é executado de acordo com o diagrama de estados apresentado na Figura 3. Na figura, um nó da rede é sempre representado por uma tupla `(MAC, porta)`. Essa tupla identifica que uma estação com um dado endereço `MAC` está conectada na `porta` do comutador. A figura sintetiza o processo de autenticação, mostrando que a autenticação ocorre no sentido de controlar o acesso de um MAC através de uma porta do comutador. Dessa forma, um nó, ao ingressar na rede, está inicialmente no estado `não autenticado` e, assim, todo tráfego gerado ou destinado a esse nó é bloqueado, exceto pelo tráfego com tipo Ethernet `0x888E` (IEEE 802.1X). Esse tráfego é encaminhado da estação para o Autenticador através de fluxos *multicast* e, no sentido contrário, é encaminhado em fluxos *unicast*, já que o Autenticador aprende o endereço MAC da estação suplicante após o recebimento do primeiro pacote IEEE 802.1X. Assim que a estação inicia o procedimento de autenticação, enviando a mensagem de `início`, a estação é movida para o estado de `pendente`, estado em que todo o tráfego da estação continua bloqueado, mas a estação está no aguardo de uma confirmação do Autenticador para o POX de que sua autenticação foi bem sucedida e quais as credenciais foram usadas na autenticação. Assim que há a confirmação de que a autenticação foi bem sucedida, o POX move a estação para o estado `autenticado`. Nesse estado, o POX confere as permissões de acesso a recursos da rede que a estação possui, de acordo com suas credenciais, e libera o acesso da estação à rede. No entanto, quando há tráfego para a estação, o POX confere se o trafego para a estação está de acordo com as políticas referentes às credenciais usadas pela estação para acessar a rede. Caso as políticas estejam de acordo com o uso da rede, a estação é movida para estado `autorizado` e acessa os recursos da rede de acordo com seus privilégios.

Tendo em vista o controle de acesso empregado, a liberação ou bloqueio do tráfego de uma estação final é realizado de acordo com as credenciais apresentadas pela estação. A autenticação da tupla `(MAC, porta)` está relacionada com a identidade da estação. Dessa forma, é possível fazer o mapeamento dos fluxos de uma dada estação para a sua

identidade. O mapeamento ocorre no seguinte sentido. Se um fluxo OpenFlow apresenta entre suas características o endereço MAC de origem (`dl_src`) e a porta de entrada no comutador (`in_port`) iguais aos que estão na tupla de autenticação, a credencial de autenticação da estação é atribuída ao fluxo. Assim, a decisão de encaminhamento desse fluxo pode tomar como parâmetro, também, a credencial da estação e, portanto, o controle de acesso à rede pode ser mais refinado. De forma análoga, se um fluxo OpenFlow apresenta entre suas características o endereço MAC de destino (`dl_dst`) e a porta de saída do comutador (`output`) iguais aos que estão na tupla de autenticação, a credencial de autenticação da estação é também atribuída ao fluxo. Assim, o mecanismo AuthFlow também controla os fluxos destinados a uma estação de acordo com a sua identidade. Portanto, no AuthFlow, as políticas de controle de acesso podem definir regras tanto de saída quanto de entrada de pacotes para as estações finais de acordo com sua identidade.

4. Os Resultados Experimentais

O protótipo do mecanismo AuthFlow foi implementado em uma ilha do *Future Internet Testbed with Security* (FITS) [Guimarães et al., 2013]. O protótipo utiliza o hipervisor Xen 4.1.4 para prover os domínios virtuais que agem como estações finais acessando uma rede OpenFlow que, por sua vez, é implementada através do comutador programável Open vSwitch 1.2.2. O Open vSwitch [Pfaff et al., 2009] é configurado para ser controlado pelo POX[6], o controlador OpenFlow utilizado. A aplicação que realiza o controle de acesso das estações finais à rede e o controle do encaminhamento de pacotes na rede OpenFlow foi desenvolvida em Python e executa sobre o POX. O autenticador usado no protótipo é uma versão modificada do `hostapd`, para criar o canal seguro e informar ao controlador POX quando há uma autenticação ou perda da autenticação de uma estação final. O servidor RADIUS empregado no protótipo é o FreeRADIUS v2.1.12[7]. Como prova de conceito, o método de autenticação testado no protótipo foi o EAP-MSCHAP v2 [Zorn, 2000].

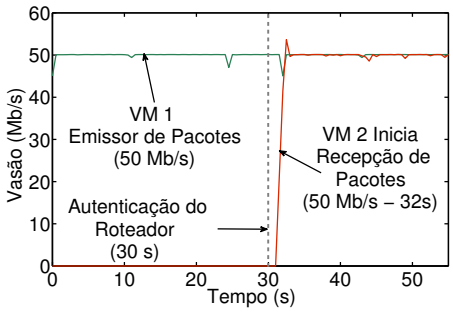
As ferramentas `Iperf`, `Nmap` e `Tcpdump`[8] foram usadas para realizar as medidas de avaliação de desempenho do protótipo. Quatro computadores pessoais compõem o cenário dos experimentos. Todos executam o protótipo do mecanismo AuthFlow. Nos computadores pessoais foram instanciadas quatro máquinas virtuais que agem como roteadores, enviam e recebem pacotes, dependendo de cada experimento. Todos os computadores possuem processadores Intel Core 2 Quad 2.4 GHz, 3 GB de memória RAM e executam o Debian Linux 3.2.0-4-amd64. Cada computador possui, no mínimo, 2 interfaces de rede sendo que todas são configuradas para funcionarem a 100 Mb/s, para garantir homogeneidade, uma vez que havia também interfaces de 1 Gb/s. As máquinas virtuais são configuradas com uma CPU virtual, 128 MB de memória RAM e executa o Debian Linux 3.2.0-4-amd64. As máquinas virtuais executam os protocolos de roteamento através da plataforma XORP [Handley et al., 2003].

O primeiro experimento avalia a eficácia do mecanismo AuthFlow em bloquear tráfegos não autorizados. O encaminhamento de pacotes de um fluxo só é liberado após a autenticação, caso contrário, os pacotes são descartados. O cenário é simples, a máquina virtual 1 (VM1) envia pacotes destinados à máquina virtual 2 (VM2) através de um roteador virtual. Assume-se que as máquinas virtuais 1 e 2 foram previamente autenticadas

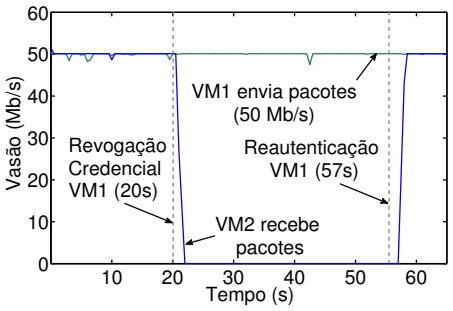
[6]O controlador POX utilizado nos experimentos é uma adaptação do controlador usado na rede de testes FITS para dar suporte ao mecanismo AuthFlow.

[7]http://freeradius.org/.

[8]http://iperf.sourceforge.com/, http://www.nmap.org/ e http://www.tcpdump.org/.



(a) Liberação de tráfego pelo roteador situado entre as máquinas virtuais 1 e 2 (VM1 e VM2) após a sua autenticação que ocorre em 30 s.



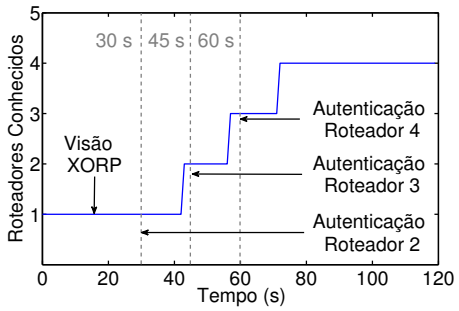
(b) Bloqueio do tráfego quando em 20 s a autenticação da máquina virtual 2 é revogada e liberação do tráfego quando a autenticação é restabelecida ao fim do teste.

Figura 4. Bloqueio da função encaminhamento de tráfego por falta de autenticação e por revogação de credencial. Máquina virtual 1 (VM1) envia pacotes para a máquina virtual 2 (VM2). a) Autenticação do roteador entre VM1 e VM2. b)Revogação da autenticação da VM1.

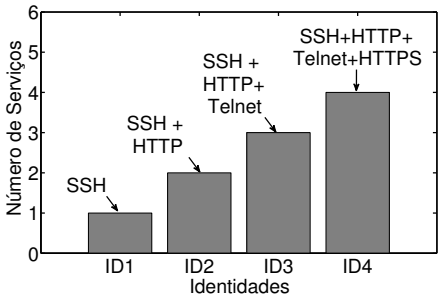
na rede e o roteador virtual que as interconecta não está autenticado. A VM1 gera um fluxo de pacotes UDP de tamanho 1472 B de conteúdo a uma taxa constante de 50 Mb/s. Como o roteador não está autenticado, o fluxo não chega à VM2. Após 30 s, o roteador se autentica na rede, como mostrado na Figura 4(a), e o fluxo UDP é recebido pela VM2. A Figura 4(a) evidencia que há um atraso, da ordem de 2 s a 2,5 s entre o início do processo de autenticação do roteador e a efetiva liberação do acesso à rede. Esse atraso é devido ao processo de autenticação do padrão IEEE 802.1X somado ao tempo de instanciação do fluxo OpenFlow. Esse atraso ocorre somente no momento em que o roteador, entre VM1 e VM2, ingressa na rede.

O segundo experimento evidencia a eficácia do mecanismo de revogação de credencial do AuthFlow, mostrado na Figura 4(b). O cenário consiste de uma máquina virtual, VM1, que se comunica diretamente com outra máquina virtual, VM2, não há roteadores entre elas. Mais uma vez assume-se que inicialmente ambas as máquinas virtuais estão autenticadas. Após 20 s, a autenticação da VM2 é revogada e, então, o acesso à rede da VM 2 é bloqueado, tanto para o envio quanto para a recepção de pacotes. Observa-se que o atraso para o bloqueio da atividade da VM2 na rede é menor que 1 s. Após 57 s, a autenticação da VM2 é restabelecida e a VM2 volta a receber os pacotes. O procedimento de restabelecimento da autenticação ocorre com um atraso de aproximadamente 2 s, assim como a autenticação de uma nova estação. Deve ser ressaltada a efetividade do mecanismo proposto AuthFlow correspondente à ação de liberação e de bloqueio de tráfego através do encaminhamento e do descarte de pacotes, respectivamente, associada ao procedimento de autenticação e revogação de credenciais. Assim, o AuthFlow se constitui em um forte aliado na defesa contra ataques de negação de serviço devido a sua efetividade na ação de liberar e bloquear fluxos em redes definidas por *software*, condicionadas ao processo de autenticação.

Os experimentos seguintes demonstram a visão da rede para as estações autenticadas, ou seja, quais estações da rede uma estação autenticada alcança e quais os serviços da rede essa estação acessa. A Figura 5(a) mostra a visão da rede segundo um roteador virtual executando o protocolo de roteamento de estado de enlace OSPF (*Open Shortest Path First*). A topologia considerada é um anel, conectando quatro roteadores virtuais. O experimento consiste em verificar a base de dados do OSPF ao longo do tempo e, então,



(a) Quantidade de vizinhos um roteador virtual percebe na rede de acordo com o número de outros roteadores autenticados.



(b) Número de serviços providos pela rede, de acordo com a credencial usada pela estação final ao se autenticar na rede.

Figura 5. Visão da rede a partir de um roteador virtual (a) e de uma estação final (b). Cada identidade usada só tem acesso a um determinado conjunto de serviços na rede.

identificar quantos roteadores vizinhos o roteador observado conhece. No início do experimento, somente o roteador observado está autenticado na rede. Após 30 s, autentica-se o segundo roteador. Em 45 s, autentica-se o terceiro e, finalmente em 60 s, autentica-se o quarto roteador. Vale ressaltar que o atraso entre a autenticação e a descoberta de cada novo roteador, indicado na Figura 5(a), é devido ao tratamento de pacotes de *brodcast/multicast* adotado na rede OpenFlow. Para evitar a sobrecarga na rede, a cada inundação de pacotes, é inserida uma regra nas tabelas de fluxos para realizar o descarte de pacotes com a mesma característica do pacote inundado por 5 s.

O quarto experimento procura demonstrar uma das principais vantagens oferecidas pelo mecanismo AuthFlow, que consiste no uso da credencial de autenticação como forma de realizar o encaminhamento de fluxos. A ideia chave é a autenticação prover uma "identificação" dos fluxos correspondentes aos serviços que são autorizado para a estação. Assim, a autenticação pelo mecanismo AuthFlow possibilita a liberação dos fluxos correspondentes aos serviços que foram liberados, bloqueando todos os demais fluxos. O experimento consiste em uma estação solicitante, autenticada com uma das quatro identidades possíveis (ID1, ID2, ID3 ou ID4), acessar outra estação fornecedora de serviços na rede. Cada identidade permite o acesso a um determinado número de serviços na rede (um, dois, três ou quatro serviços, respectivamente). Para tanto, a estação solicitante executa uma varredura de portas (`nmap`) na estação fornecedora de serviços. No cenário de testes, a estação solicitante é a mesma, para as quatro identidades, mantendo o mesmo endereço IP e MAC durante todo o experimento. A única modificação no cenário de testes é a autenticação da estação solicitante com outra identidade a cada teste. A Figura 5(b) mostra os serviços que a estação solicitante consegue acessar na estação fornecedora de serviços na rede. Assim, é possível observar que, ao estar autenticada com uma identidade, a estação só consegue acessar os serviços liberados para aquela identidade. Vale ressaltar que a varredura de portas retorna que as portas que não têm serviços liberados são filtradas, o que mostra que o bloqueio dos demais serviços é realizado pelo descarte dos pacotes `SYN`, o que, de fato, ocorre pelas regras instaladas pelo controlador POX ao verificar que uma estação não tem o devido nível de privilégio para acessar um serviço.

5. Os Trabahos Relacionados

A segurança de redes definidas por *software*, em especial a segurança de redes OpenFlow, é um tema bastante discutido atualmente. Há propostas para o desen-

volvimento de aplicações de segurança sobre a infraestrutura de rede OpenFlow, como também há outras que visam garantir a segurança da infraestrutura. Contudo, garantir a autenticação, o controle de acesso, a escalabilidade, o baixo tempo de resposta, a confidencialidade e a disponibilidade em SDN continua a ser um desafio [Kreutz et al., 2013].

Kreutz *et al.* apresentam uma classificação dos principais vetores de ataque a uma rede definida por *software* e possíveis contramedidas para se proteger desses ataques [Kreutz et al., 2013]. Kreutz *et al.* restringem-se a ataques contra a resiliência e confiabilidade da rede. Visando garantir a confidencialidade e a disponibilidade de redes definidas por *software*, o sistema QFlow [Mattos e Duarte, 2012, Mattos et al., 2013] se baseia em um sistema híbrido Xen e OpenFlow para prover o isolamento de recursos e de comunicação entre redes virtuais sobre uma infraestrutura SDN. O sistema adota o encaminhamento de pacotes por filas para garantir a reserva de banda para cada rede virtual e marca os pacotes de cada rede com um marcador de VLAN, para multiplexar a qual rede virtual um pacote pertence. No QFlow, no entanto, não há mecanismos de autenticação ou controle de acesso entre máquinas virtuais e a infraestrutura de rede. Assim, o AuthFlow é complementar ao QFlow, pois assegura segurança ao controle de acesso.

A rede UPV/EHU [Matias et al., 2011], uma rede OpenFlow de testes europeia, também adota uma proposta de autenticação baseada no padrão IEEE 802.1X. Contudo, tal proposta não considera o uso das credenciais de autenticação do nó na rede para a definição de novos fluxos. O diferencial da proposta AuthFlow é realizar o mapeamento das credencias de autenticação para os fluxos, assim, a qualquer tempo, é possível identificar os nós que estão gerando ou recebendo um tráfego em um dado comutador e, se necessário, revogar sua autenticação. A proposta AuthFlow fornece ainda a primitiva de se definir regras de encaminhamento no controlador da rede de acordo com a identidade das estações, o que é um diferencial em relação às outras propostas.

Guenane *et al.* propõem um mecanismo de autenticação de redes virtuais usando EAP-TLS implementado em cartões inteligentes (*smart cards*) [Guenane et al., 2012]. A proposta consiste em garantir o acesso de máquinas virtuais e de clientes das redes virtuais a cartões inteligentes, que implementam o protocolo TLS e encapsulam as mensagens em EAP. As mensagens encapsuladas EAP são enviadas para um servidor RADIUS que autentica os componentes da rede virtual, assim como os clientes da rede virtual, através da autenticação mútua provida pelos certificados, assinados por uma Autoridade Certificadora, apresentados durante a negociação TLS. No entanto, essa proposta não define como seria o mecanismo de controle de acesso dos nós à rede e como a autenticação é usada para autorizar o acesso do cliente aos recursos da rede. O mecanismo AuthFlow proposto nesse artigo pode ser usado conjuntamente com esta proposta de autenticação com cartões inteligentes, uma vez que os dados de autorização são encapsulados em EAP. Assim, o AuthFlow controlaria o acesso dos roteadores virtuais aos recursos da rede.

Resonance [Nayak et al., 2009] e Ethane [Casado et al., 2007] são outras propostas que visam a autenticação de nós em uma rede definida por *software*. Ambas defendem que a autenticação do nó na rede deve ser feita por através de portal *Web* em que o usuário deve apresentar as suas credenciais. Essa abordagem apresenta uma restrição básica que é a necessidade de o nó, que está acessando a rede, ter um navegador *Web* instalado. Esse requisito é bem limitante, quando se consideram ambientes formados por redes virtuais compostas por máquinas virtuais extremamente leves que não possuem nem interface gráfica. Outra desvantagem desse método de autenticação é a limitação ao modelo de autenticação por usuário e senha, enquanto o modelo de autenticação adotado pelo AuthFlow baseia-se no encapsulamento EAP, assim, qualquer que seja o método de

autenticação escolhido, se for compatível com EAP, é trivialmente suportado pelo AuthFlow. Como citado anteriormente, até mesmo métodos de autenticação robustos baseados em microcontroladores seguros são possíveis com o mecanismo AuthFlow. Outra vantagem do AuthFlow em relação a essas propostas é a autenticação diretamente na Camada 2, assim não há a necessidade de um nó adquirir um endereço IP para depois se autenticar, como ocorre no Resonance ou no Ethane. No AuthFlow, assim que uma estação entra na rede, ela inicia sua autenticação de acordo com o padrão IEEE 802.1X diretamente na camada de enlace, autenticando o seu endereço MAC na porta do comutador em que está conectado. Esse procedimento impede que um nó use um endereço MAC falsificado, ao contrário de propostas, como Resonance e Ethane, que não visam impedir a falsificação do endereço MAC.

As propostas FRESCO [Shin et al., 2013] e FortNOX [Porras et al., 2012] definem um conjunto de primitivas de segurança para redes OpenFlow. A proposta FortNOX defende a criação de um núcleo seguro de execução de aplicações sobre um controlador da rede OpenFlow. Esse núcleo seguro impede que uma aplicação execute ações que interfiram nas políticas de controle de outra aplicação. A proposta FortNOX defende o fatiamento da rede entre aplicações sobre um mesmo controlador, o que gera um controle mais fino dos privilégios e do domínio de controle de cada aplicação do que o previsto pelo FlowVisor [Sherwood et al., 2009]. A proposta do FlowVisor, por sua vez, fatia a rede entre diversos controladores, contudo, não prevê uma política de segurança entre controladores para que as ações de um controlador não afete os demais. Seguindo a ideia do núcleo seguro de execução de aplicações, a proposta FRESCO define um conjunto de primitivas e uma linguagem modular para o desenvolvimento de aplicações de segurança para a rede OpenFlow. Essas propostas relacionam-se com o AuthFlow no sentido de que a proposta deste artigo pode ser usada como um módulo seguro do FRESCO, por exemplo, para permitir o uso de uma nova primitiva de segurança, a primitiva de autenticação. Com a primitiva de autenticação é possível identificar a quem pertence cada conjunto de fluxos definidos na rede.

6. Conclusão

A segurança de redes empresariais depende de mecanismos de controle de acesso e de autenticação eficientes. Com a crescente adoção de redes definidas por *software* (SDN) por redes empresarias, o desafio de prover segurança às SDN tornou-se ainda mais fundamental. Esse artigo propõe o AuthFlow, um mecanismo de autenticação e controle o acesso à infraestrutura de um rede definida por *software* OpenFlow, baseado no padrão IEEE 802.1X e no servidor de autenticação RADIUS. O mecanismo AuthFlow proposto implementa a autenticação através de uma base dados LDAP com RADIUS. A proposta, no entanto, é extensível a outros métodos de autenticação, como o EAP-TLS, que autentica os nós com base em certificados X.509. Os resultados mostram que o mecanismo de autenticação proposto impede que estações não autorizadas acessem recursos da rede, mesmo quando já autenticadas e, após, perdem seus privilégios. Os resultados mostram ainda que o mecanismo proposto é mais eficiente que as demais propostas, já que introduz menor sobrecarga de controle, e permite a definição de políticas de controle de acesso por fluxo de acordo com as credenciais de acesso de cada estação.

Como trabalhos futuros, pretende-se implantar o mecanismo AuthFlow no *Future Internet Testbed* (FITS), como seu mecanismo de autenticação e controle de acesso padrão, e estender o AuthFlow para novos métodos de autenticação, tal como o EAP-TLS que permite o uso de certificados assinados como credencial de acesso.

7. Referências

[Casado et al., 2007] Casado, M., Freedman, M., Pettit, J., Luo, J., McKeown, N. e Shenker, S. (2007). Ethane: Taking control of the enterprise. *ACM SIGCOMM Computer Communication Review*, 37(4):1–12.

[Casado et al., 2012] Casado, M., Koponen, T., Shenker, S. e Tootoonchian, A. (2012). Fabric: a retrospective on evolving SDN. Em *Proceedings of the first workshop on Hot topics in software defined networks*, HotSDN '12, p. 85–90, New York, NY, USA. ACM.

[Guedes et al., 2012] Guedes, D., Vieira, L., Vieira, M., Rodrigues, H. e Nunes, R. (2012). Redes Definidas por Software: uma abordagem sistêmica para o desenvolvimento de pesquisas em Redes de Computadores. *Minicursos do Simpósio Brasileiro de Redes de Computadores-SBRC 2012*, p. 160–210.

[Guenane et al., 2012] Guenane, F., Samet, N., Pujolle, G. e Urien, P. (2012). A strong authentication for virtual networks using eap-tls smart cards. Em *Global Information Infrastructure and Networking Symposium (GIIS), 2012*, p. 1–6.

[Guimarães et al., 2013] Guimarães, P. H., Ferraz, L., Torres, J. V., Mattos, D., Murillo, A., Lopez, M. A., Alvarenga, I., Rodrigues, C. e Duarte, O. C. M. B. (2013). Experimenting Content-Centric networks in the future internet testbed environment. Em *IEEE International Conference on Communications 2013: IEEE ICC'13 - Workshop on Cloud Convergence: challenges for future infrastructures and services (WCC 2013) (ICC'13 - IEEE ICC'13 - Workshop WCC)*, p. 1398–1402, Budapest, Hungary.

[Handley et al., 2003] Handley, M., Hodson, O. e Kohler, E. (2003). XORP: An open platform for network research. *ACM SIGCOMM Computer Communication Review*, 33(1):53–57.

[Heller et al., 2012] Heller, B., Sherwood, R. e McKeown, N. (2012). The controller placement problem. Em *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, HotSDN '12, p. 7–12, New York, NY, USA. ACM.

[Kreutz et al., 2013] Kreutz, D., Ramos, F. M. e Verissimo, P. (2013). Towards secure and dependable software-defined networks. Em *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, HotSDN '13, p. 55–60, New York, NY, USA. ACM.

[Levin et al., 2012] Levin, D., Wundsam, A., Heller, B., Handigol, N. e Feldmann, A. (2012). Logically centralized?: state distribution trade-offs in software defined networks. Em *Proceedings of the First workshop on Hot topics in software defined networks*, HotSDN '12, Helsinki, Finland. ACM.

[Matias et al., 2011] Matias, J., Jacob, E., Toledo, N. e Astorga, J. (2011). Towards neutrality in access networks: A nando deployment with openflow. Em *ACCESS 2011, The Second International Conference on Access Networks*, p. 7–12, Luxembourg City, Luxembourg.

[Mattos et al., 2011] Mattos, D., Fernandes, N. C. e Duarte, O. C. M. B. (2011). XenFlow: Um sistema de processamento de fluxos robusto e eficiente para migração em redes virtuais. Em *XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC'2011*.

[Mattos et al., 2013] Mattos, D., Ferraz, L. e Duarte, O. C. M. B. (2013). Um mecanismo para isolamento seguro de redes virtuais usando a abordagem híbrida xen e openflow. Em *SBSeg 2013 - XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, Manaus - Brazil.

[Mattos e Duarte, 2012] Mattos, D. M. F. e Duarte, O. C. M. B. (2012). QFlow: Um sistema com garantia de isolamento e oferta de qualidade de serviço para redes virtualizadas. Em *XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos - SBRC'2012*.

[McKeown et al., 2008] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S. e Turner, J. (2008). OpenFlow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev., 2008*.

[Nagahama et al., 2012] Nagahama, F. Y., Granville, L., Farias, F., Cerqueira, E., Aguiar, E., Gaspary, L. e Abelém, A. (2012). IPSFlow – uma proposta de sistema de prevenção de intrusão baseado no framework openflow.

[Nayak et al., 2009] Nayak, A. K., Reimers, A., Feamster, N. e Clark, R. (2009). Resonance: Dynamic access control for enterprise networks. Em *Proceedings of the 1st ACM Workshop on Research on Enterprise Networking*, WREN '09, p. 11–18, New York, NY, USA. ACM.

[Pfaff et al., 2009] Pfaff, B., Pettit, J., Koponen, T., Amidon, K., Casado, M. e Shenker, S. (2009). Extending networking into the virtualization layer. *Proc. HotNets*.

[Porras et al., 2012] Porras, P., Shin, S., Yegneswaran, V., Fong, M., Tyson, M. e Gu, G. (2012). A security enforcement kernel for openflow networks. Em *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, HotSDN '12, p. 121–126, New York, NY, USA. ACM.

[Sherwood et al., 2009] Sherwood, R., Gibb, G., Yap, K., Appenzeller, G., Casado, M., McKeown, N. e Parulkar, G. (2009). Flowvisor: A network virtualization layer. Relatório técnico, Tech. Rep. OPENFLOW-TR-2009-01, OpenFlow Consortium.

[Shin et al., 2013] Shin, S., Porras, P., Yegneswaran, V., Fong, M., Gu, G. e Tyson, M. (2013). Fresco: Modular composable security services for software-defined networks. Em *Proceedings of Network and Distributed Security Symposium*.

[Zorn, 2000] Zorn, G. (2000). Microsoft PPP CHAP Extensions, Version 2. RFC 2759 (Informational).