Uma Proposta para a Autenticação de Estações de Trabalho em Redes Definidas por Software com Utilização de Certificados Auto-Assinados

Osiel O. Souza [1], **Jeferson C. Nobre** [2]

[1]Instituto de Informática – Universidade do Vale do Rio dos Sinos (UNISINOS)
Caixa Postal 15.064 – 91.501-970 – São Leopoldo – RS – Brazil

osielolivera@gmail.com

Abstract. *At the same time the architecture of the Software Defined Networking - SDN has to be promising, there are some security challenges to be overcome in the implementation of such technology. The need for authentication of network components becomes a key issue in SDN due to centralization of logic controllers. An attacker could host station malicious applications and launch attacks against the control plane, making the network vulnerable. This paper proposes the implementation of a public key infrastructure combined with the use of self-signed certificates as a possible solution to the authentication problem of the origin in a SDN OpenFlow.*

Resumo. *Ao mesmo tempo em que a arquitetura das redes definidas por software (Software Defined Networking - SDN) apresenta-se promissora, existem alguns desafios de segurança a serem transpostos na implementação de tal tecnologia. A necessidade de autenticação dos componentes da rede torna-se uma questão fundamental em SDN devido a centralização lógica dos controladores. Uma estação atacante poderia hospedar aplicações maliciosas e disparar ataques contra o plano de controle, tornando a rede vulnerável. Este trabalho propõe a implementação de uma infraestrutura de chaves públicas aliada a utilização de certificados auto-assinados como uma possível solução ao problema de autenticação da origem em uma SDN OpenFlow.*

1. Introdução

A abstração de rede proporcionada pela arquitetura SDN aliada a programabilidade e visão centralizada chamaram a atenção de profissionais de TI e pesquisadores do mundo todo, movendo rapidamente o conceito para uma realidade [Jain et al. 2013]. O paradigma SDN desacopla o controle de encaminhamento dos dados e estreita a interação entre aplicações, dispositivos e serviços de rede. A divisão entre o plano de controle e plano de dados permite que a tomada de decisões antes realizada nos dispositivos, seja efetuada em outro ponto da rede através de software [Guedes et al. 2012]. Os equipamentos tais como roteadores e comutadores estabelecem o plano de dados e são responsáveis pelo encaminhamento de pacotes. A inteligência para tomada de decisões está concentrada no plano de controle, executada por uma entidade de rede chamada de controlador [Kim and Feamster 2013]. Uma das primeiras abordagens que fundamentou o paradigma SDN foi a definição do protocolo OpenFlow [OpenFlow Specification-Version 2013]. O Openflow pode ser utilizado para prover diversas funcionalidades nas infraestruturas de

rede, como por exemplo a implementação gradual de recursos e funcionalidades relacionadas a segurança.

As redes de computadores, tanto as tradicionais quanto as SDN, necessitam de mecanismos de segurança. Um dos mecanismos de segurança aplicado as rede de computadores é a autenticação de portas, a qual restringe o acesso não autorizado de dispositivos a uma rede local [Barros and Foltran Junior 2008]. As estações finais que se conectam a uma rede de computadores nem sempre serão confiáveis, podendo apresentar inúmeras vulnerabilidades. O protocolo OpenFlow, apesar de largamente utilizado em SDN, não possui um mecanismo nativo para autenticação segura da origem. Por padrão, as estações finais se autenticam a uma rede OpenFlow através da validação de seus endereços MAC (*M*edia Access Control) ou IP (*I*nternet Protocol), com base na lógica de comutação de pacotes definida pelo administrador da topologia. Um atacante poderia, por exemplo, obter o endereço MAC ou IP de uma estação legítima e configurar esses dados em uma estação maliciosa. Posteriormente o atacante poderia realizar uma tentativa de autenticação na rede. Esse problema ao ocorrer em uma SDN, pode comprometer a segurança da topologia, devido a centralização lógica do controlador. Um atacante poderia explorar vulnerabilidades presentes no controlador e suas aplicações, ou disparar um ataque de negação de serviço (*D*enied of Service - DOS), provocando a indisponibilidade de recursos na rede.

Existem algumas propostas para autenticação de portas em SDN. Uma delas é o mecanismo de autenticação chamado AuthFlow, que realiza autenticação das estações finais diretamente na camada enlace e associa suas credenciais a porta do comutador na qual a estação está conectada [Mattos et al. 2014]. Apesar de prover uma importante contribuição para segurança SDN, o mecanismo AuthFlow realiza apenas a autenticação de portas em uma SDN OpenFlow, não garante que a origem é realmente que diz ser. Outra proposta sugere que a estação final receba um endereço IP temporário e seja direcionada para um sítio Web, onde terá as credenciais de acesso validadas [Casado et al. 2007]. Tal proposta autentica as estações na rede através de um usuário e senha conhecido pelo controlador, porém não implementa autenticação mútua ou negociação de um seguro método de criptografia entre as partes.

O presente artigo propõe a implementanção de uma infraestrutura de chaves públicas aliada a utilização de certificados auto-assinados como possível solução ao problema de autenticação da origem em uma SDN OpenFlow. Para isso, um mecanismo de autenticação de portas que utilize o padrão 802.1x em conjunto com certificados digitais deverá ser desenvolvido. Esse mecanismo será integrado a um controlador OpenFlow de modo a assegurar autenticação segura para as estações finais.

A estrutura deste trabalho encontrasse no seguinte formato: na seção 2 é apresentada uma introdução ao paradigma SDN e funcionamento do protocolo OpenFlow; A seção 3 detalha os principais desafios de segurança em SDN e apresenta o problema de autenticação da origem em uma SDN OpenFlow; A seção 4 detalha a proposta de um mecanismo de autenticação e seus aspectos técnicos; A seção 5 sugere uma discussão sobre trabalhos relacionados; Por fim, a seção 6 conclui o artigo e apresenta as considerações finais.

2. O Paradigma das Redes Definidas por Software

O paradigma SDN desacopla o controle de encaminhamento dos dados e estreita a interação entre aplicações, dispositivos e serviços de rede, sejam estes reais ou virtualizados. A inteligência para tomada de decisões está concentrada no controlador de rede, que por meio de interfaces de programação, realiza a mediação entre aplicações que desejam comunicar-se com elementos da rede e elementos da rede que desejam transmitir informações na topologia [Nadeau and Gray 2013]. Uma SDN possibilita a seu administrador o desenvolvimento de uma lógica centralizada, implementação gradual de recursos e visão global da rede, consolidando as ações em um único ponto de controle. A rede torna-se independente dos fabricantes de equipamentos, pois os dispositivos são responsáveis apenas por executar as decisões previamente definidas pelo controlador [Guedes et al. 2012].

Uma SDN é caracterizada pela presença de um software de controle, o qual através de uma interface de programação supervisiona o encaminhamento de pacotes realizado pelos elementos de comutação. O software de controle recebe dos comutadores uma interface de programação que o permite inspecionar, alterar e definir entradas na tabela de roteamento dos elementos de comutação [Nadeau and Gray 2013]. O controlador possibilita ainda a utilização de um divisor de visões, permitindo que pacotes que se identifiquem com um determinado padrão, sejam associados a múltiplos comportamentos e executados em diferentes controladores de uma mesma rede física [Guedes et al. 2012]. Na Figura 1, podemos vislumbrar a arquitetura de uma SDN em suas principais camadas e interfaces de comunicação. Em uma SDN, interfaces *Southbound* são usadas para a comunicação entre o controlador e o plano de dados, possibilitando alterações na rede em tempo real. Tais interfaces podem ser de código aberto ou proprietárias e constituem uma camada de abstração para o controle e gerenciamento da topologia.

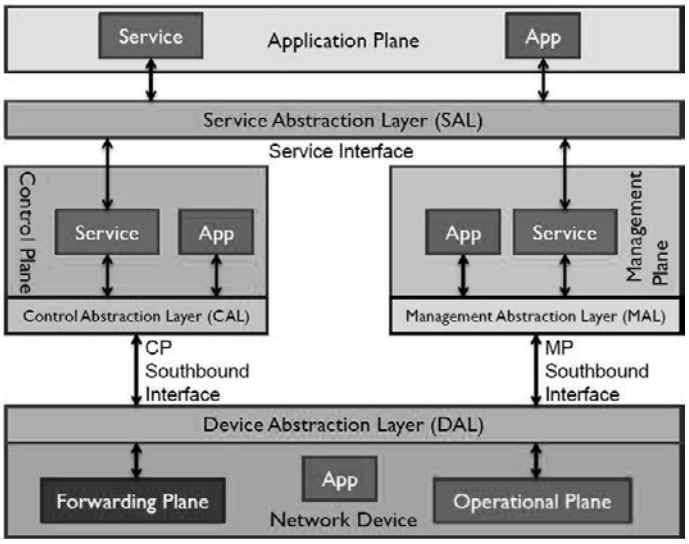

Figura 1. Arquitetura SDN Modelo em Camadas
[Haleplidis et al. 2014]

De uma forma geral, o plano de controle é responsável por estabelecer um conjunto de dados utilizado para a criação de uma tabela de encaminhamento. Essa tabela é utilizada pelo plano de dados para encaminhar o tráfego no destino correto

[Guedes et al. 2012]. O conjunto de dados construído a partir da topologia da rede é chamado de *Routing Information Base - RIB*. Muitas vezes a RIB é mantida consistente através da troca de informações entre controladores de uma mesma rede. As tabelas de encaminhamento *Forwarding Information Base - FIB* contém as informações necessárias sobre os fluxos de dados e as ações necessárias para cada um deles, como por exemplo, sair por determinada porta de um comutador. A FIB é criada contanto que o RIB seja considerado consistente e estável [Nadeau and Gray 2013].

O plano de dados é responsável pela comutação e repasse dos datagramas na rede. Tal plano opera em nível de link coletando os datagramas entrantes via inúmeros meios físicos, tais como fibra óptica, cabeado ou sem fio [Nadeau and Gray 2013]. Ao receber os datagramas, o plano de dados realiza uma verificação de integridade. Caso o datagrama seja considerado consistente, este é processado pelo plano de dados de acordo com uma consulta realizada na FIB(recebida do plano de controle). Sendo o destinatário conhecido e presente na FIB, o datagrama será encaminhado de acordo com os parâmetros estipulados para a entrega do dado. Se o destinatário for considerado desconhecido pelo plano de dados, este encaminha os datagramas para o plano de controle, que executa novamente o processo de criação da RIB e consequentemente a geração de uma nova FIB [Rothenberg et al. 2010].

2.1. O Protocolo OpenFlow

Umas das formas de implementar interfaces *S*outhbound é através do protocolo OpenFlow. O OpenFlow foi desenvolvido a partir de um projeto na rede de estudos da universidade de Stanford. O propósito inicial era o desenvolvimento de novos protocolos que pudessem auxiliar em pesquisas e experimentos [McKeown et al. 2008]. Para atingir o objetivo a universidade precisou criar seu ambiente de experimentação a partir do zero. Mediante a criação do núcleo da rede, pesquisadores sugeriram que o OpenFlow poderia substituir as camadas 2 e 3 do modelo TCP/IP em equipamentos de rede comerciais. Em 2011, um consórcio sem fins lucrativos chamado*(Open Networking Foundation - ONF)* foi constituído por um grupo de prestadores de serviços com objetivo de comercializar, padronizar e promover a utilização do OpenFlow em redes produtivas [Nadeau and Gray 2013].

Os elementos chave do protocolo OpenFlow tornaram-se parte da definição comum de SDN. Tais elementos são a separação do plano de controle e plano de dados. O OpenFlow estabelece um protocolo padrão, que age como mediador entre o controlador e os equipamentos da rede, por meio de uma API moderna e extensível [Rothenberg et al. 2010]. É importante salientar que o OpenFlow é um conjunto de protocolos e uma API. Não devemos categorizar o OpenFlow como um produto por si só, ou uma única característica de um produto, pois o controlador não executa nada sem uma API para determinar as regras como os pacotes devem ser comutados [Nadeau and Gray 2013].

No OpenFlow, consultas as tabelas de encaminhamento são realizadas nos próprios dispositivos de rede, garantindo o desempenho do tráfego. Decisões de como cada pacote deve ser tratado na topologia não são mais tarefas dos equipamentos de rede e sim do controlador [McKeown et al. 2008]. Interfaces programáticas tem o domínio sobre os dispositivos de rede para a implementação de novas funcionalidades e aplicação

instantânea de regras. Os comutadores recebem uma tabela de fluxo interno e possuem uma interface padronizada para adicionar ou remover entradas conforme a rede vai sendo programada [Nadeau and Gray 2013]. Atualmente o protocolo OpenFlow está dividido em duas partes:

- *Wire Protocol:* Um protocolo utilizado para o estabelecimento de sessões de controle, define uma estrutura para troca de mensagens de modificações no fluxo e realiza a coleta de dados estatísticos. Define ainda a estrutura básica do switch como, por exemplo, o conjunto de portas e tabelas nos dispositivos.
- *Of-Config:* É um protocolo de configuração e gerenciamento, seu funcionamento foi baseado no protocolo NETCONF e sua principal função é alocação de portas físicas de um switch para um controlador em especial. Além disso, é utilizado para definição de parâmetros de alta disponibilidade (ativo / passivo) e ações que o plano de dados deve tomar em caso de falha no controlador.

O protocolo Wire introduz o conceito de substituição de estado efêmero, ou seja, as entradas de fluxo não são armazenadas de forma permanente nos equipamentos. O conceito de estado efêmero dispensa o uso de dispositivos com alto poder de processamento uma vez que os equipamentos são responsáveis apenas pela comutação de pacotes [Rothenberg et al. 2010].

Em uma entrada de fluxo OpenFlow, todo o cabeçalho do pacote está disponível para modificações [McKeown et al. 2008]. Através da utilização do protocolo OpenFlow o controlador pode adicionar, atualizar e excluir entradas nas tabelas de fluxo, tanto de forma proativa como de forma reativa. Na forma proativa, o controlador popula previamente as flow tables podendo descartar fluxos não previstos. Na forma reativa, o primeiro pacote do fluxo aciona o controlador para inserir uma entrada na flow table. O modelo reativo apresenta um uso eficiente, porém gera maior fluxo de mensagens entre controlador e os switches [OpenFlow Specification-Version 2013].

3. Desafios de Segurança em SDN

Ao mesmo tempo em que a arquitetura SDN apresenta-se promissora, existem alguns desafios de segurança a serem transpostos na implementação de tal tecnologia. A centralização do plano de controle traz inúmeras vantagens, como programabilidade, lógica centralizada e visão global da rede [Feamster et al. 2014]. Tais atributos representam significativos benefícios, porém aumentam a exposição do controlador e suas aplicações a ataques de DOS e interceptação de fluxos. Em uma rede OpenFlow, todo pacote é analisado, estando um pacote com o cabeçalho não associado aos fluxos existentes, é enviado para inspeção do controlador. Caso algum comutador da topologia envie uma quantidade incomum de novos cabeçalhos de pacotes para o controlador, tal entidade poderia ter seus recursos de processamento esgotados[Kreutz et al. 2013].

Outra questão pertinente refere-se a falta de uma relação de confiança entre os elementos de rede e os controladores. Por exemplo, as aplicações que rodam no controlador podem apresentar um comportamento malicioso, o controlador deveria ser capaz de identificar ações anômalas, porém, não o faz nativamente [Kreutz et al. 2013]. Tal relação de confiança também deveria existir entre os controladores e comutadores da rede, pois uma vulnerabilidade em um comutador poderia permitir acesso indevido a dados da topologia. Não menos importante é a forma como as estações finais se autenticam em uma SDN, pois

podem hospedar códigos maliciosos destinados a exploração de vulnerabilidades presentes no controlador [Porras et al. 2012].

Existem duas linhas de pensamento sobre a segurança em redes SDN, a primeira diz que melhorias podem ser desenvolvidas explorando a capacidade de programação e visão global nativa de tal arquitetura [Porras et al. 2012]. Por exemplo, a centralização lógica do plano de controle torna possível um monitoramento de segurança altamente reativo, viabilizando a implementação de métodos para detecção de anomalias, os quais poderiam gerar dados a partir de uma visão global da topologia e enviá-los para o controlador. Com base nas informações recebidas via análise, uma política de segurança poderia ser constantemente atualizada e propagada na rede [Scott-Hayward et al. 2013]. A segunda linha de pensamento diz que os atributos de programabilidade e centralização expõe a rede a uma nova gama de ataques, em virtude das diversas possibilidades de customização. Segundo [Kreutz et al. 2013] a criação de aplicações de segurança em uma SDN torna-se um desafio uma vez que a própria segurança de uma SDN é questionável, visto que questões como a autenticação de dispositivos não possui padrão estabelecido. Sem um mecanismo seguro de autenticação, um *host* atacante poderia alcançar o controlador e explorar vulnerabilidades presentes em tal entidade [Scott-Hayward et al. 2013].

3.1. Problema de Autenticação de Portas em SDN

A necessidade de autenticação dos elementos de rede não é exclusividade de uma SDN, tão pouco um novo paradigma, redes convencionais sem separação do plano de dados e controle apresentam o mesmo tipo de problema [Porras et al. 2012]. A questão peculiar em uma SDN é o aumento da criticidade desse fato, pois estando um nó da rede comprometido, alvos como o controlador e estação de gerenciamento podem ser alcançados, tornando a rede vulnerável [Kreutz et al. 2013]. Um controlador OpenFlow deveria ser capaz de identificar aplicações maliciosas sob sua gestão, porém não executa essa tarefa por si só. Para detecção de fluxos maliciosos é necessário o desenvolvimento de um módulo de segurança no próprio controlador [Rothenberg et al. 2010]. Tal fato reforça a necessidade de um mecanismo para autenticação da origem, pois caso uma estação maliciosa conseguisse se autenticar, poderia enviar fluxos ao controlador sem ser detectada.

O protocolo OpenFlow provê programabilidade a rede, porém, não possui um mecanismo nativo para autenticação de portas [Rothenberg et al. 2010]. Por padrão, as estações finais se autenticam a uma rede OpenFlow através da validação de seus endereços MAC (*M*edia Access Control) ou IP (*I*nternet Protocol), com base na lógica de comutação de pacotes definida pelo administrador da topologia. Um atacante poderia obter o endereço MAC ou IP de uma estação legítima e configurar esses dados em uma estação invasora. Posteriormente, o atacante realizaria uma tentativa de autenticação na rede. Caso o acesso fosse bem sucedido, a estação maliciosa poderia explorar aplicações vulneráveis no plano de controle ou disparar um ataque de DOS contra o controlador, consumir recursos de link, memória e processamento dos comutadores, gerando indisponibilidade de serviços na topologia.

Nativamente, o OpenFlow implementa uma conexão protegida através do protocolo TLS entre os comutadores e os controladores da rede. O TLS não garante a autenticação, o TLS garante a privacidade dos dados, pois estes são criptografados [McKeown et al. 2008]. Para haver autenticação entre controlador e comutador, ou en-

tre os *hosts* e o controlador OpenFlow, os dispositivos de rede deveriam ter sua origem certificada.

4. Proposta para Autenticação de Portas em SDN

Em um ambiente de rede onde o meio físico é compartilhado ou aberto, como nas redes sem fio e redes cabeadas, a necessidade de confiança nas estações finais torna-se um aspecto fundamental na topologia [Congdon et al. 2003]. Uma forma encontrada para resolução desse problema foi o desenvolvimento de protocolos para autenticação de portas, provendo controle de acesso a uma rede computacional. Através da autenticação de portas é possível aprimorar a segurança do ambiente tecnológico e aplicar políticas de controle de acesso a todos os dispositivos que precisam acessar uma determinada rede [Barros and Foltran Junior 2008]. O padrão 802.1X é amplamente utilizado em mecanismos para autenticação de portas, tal padrão provê autenticação entre clientes de rede e os equipamentos nos quais estão conectados.

O padrão IEEE 802.1X possibilita o acesso autenticado em redes Ethernet, Token Ring e redes sem fio padrão 802.11, também oferece suporte ao protocolo RADIUS *(remote authentication dial in user service support)*. O padrão IEEE 802.1X define porta como sendo um ponto de conexão à rede, podendo ser uma porta física, em redes cabeadas ou uma porta lógica, quando existe associação entre um dispositivo sem fio e o ponto de acesso [Congdon et al. 2003]. Um Servidor de autenticação RADIUS pode autenticar cada estação conectada a uma porta antes que esta possa acessar qualquer serviço oferecido pela rede. Até que o cliente esteja devidamente autenticado, o controle de acesso 802.1X habilita somente o tráfego do protocolo EAP *Estensible Authentication Protocol* na porta onde a estação estiver conectada. Caso a autenticação obtenha sucesso, o tráfego de pacotes será integralmente permitido [Barros and Foltran Junior 2008]. A utilização do protocolo EAP torna possível a independência quanto a mecanismos de autenticação limitados apenas por senha, como por exemplo, o protocolo de autenticação PPP (*P*oint-to-Point Protocol). O EAP apresenta uma alternativa para interligação de redes devido a sua capacidade de adaptação a novos mecanismos de autenticação e pode, por exemplo, ser utilizado em conjunto com o protocolo TLS para implementações onde sejam utilizados certificados digitais [Congdon et al. 2003]. O EAP-TLS usa certificados padrão X.509 para verificar a identidade do usuário, aplicação ou estação de trabalho. Suporta autenticação mútua, onde o cliente deve confiar no certificado de um servidor e o servidor deve confiar no certificado do cliente. Através da utilização do EAP-TLS, também pode ser definido um algoritmo de criptografia antes da transmissão dos dados.

Um modo de prover autenticação, integridade e confidencialidade através da utilização do protocolo EAP-TLS, é a implementação de uma infraestrutura de chaves públicas. Tal infraestrutura é uma composição de segurança cujos serviços são executados e entregues utilizando conceitos e técnicas de criptografia assimétrica [Adams and Lloyd 2003]. Isso significa que para a encriptação de uma mensagem serão necessárias uma chave pública e uma chave privada. As chaves públicas não devem ser mantidas em segredo, porém devem ser protegidas contra falsificação. As chaves privadas entretanto, devem ser mantidas permanentemente em segredo [Buchmann 2002]. As infraestruturas de chaves públicas são responsáveis pela distribuição e gerenciamento de tais chaves. Podemos dizer ainda que tais infraestruturas foram desenvolvidas para autenticar e identificar usuários e serviços, garantindo que as informações trocadas estejam

disponíveis apenas as entidades autorizadas, assegurando que se uma entidade realizar uma ação, não poderá negar que a realizou [Adams and Lloyd 2003].

O presente artigo propõe o desenvolvimento de um mecanismo de autenticação para estações finais em uma SDN OpenFlow através da implementação de uma infraestrutura de chaves públicas. A Figura 2 ilustra uma proposta de arquitetura para autenticação de estações de trabalho em uma SDN OpenFlow através da utilização de certificados auto-assinados.

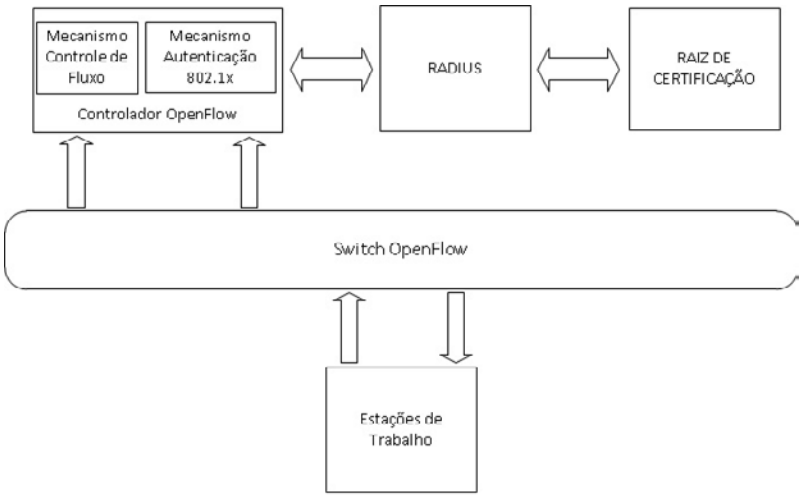

Figura 2. Arquitetura proposta

O mecanismo proposto irá adotar o padrão 802.1x para realizar a troca de mensagens com as estações de trabalho na camada de enlace. As estações deverão enviar um pacote 802.1x com o método de autenticação EAP-TLS ao controlador. O mecanismo de autenticação deverá realizar a troca de mensagens 802.1x com a estação e receber um pacote com conteúdo EAP proveniente da estação suplicante. Posteriormente o mecanismo de autenticação deverá encaminhar a requisição para o servidor RADIUS, que por sua vez, irá consultar a raiz de certificação e identificar se a estação possui um certificado válido. A raiz de certificação é responsável pela emissão, gerenciamento e revogação de certificados para a SDN. A estrutura de um certificado é basicamente um nome previamente comprovado, uma chave pública e uma assinatura digital. A assinatura é geralmente realizada pela autoridade certificadora, que atesta a autenticidade do certificado através de sua chave privada. Os certificados auto-assinados são produzidos de uma forma que é possível identificar se um atacante substituiu a chave pública, nome, ou qualquer outra informação. Um formato de certificado amplamente utilizado é o padrão X.509 v3, atualmente descrito na RFC5280 [Dang et al. 2010]. Caso as credenciais estejam corretas, a estação de trabalho recebe uma mensagem de sucesso e o processo de autenticação é concluído. Por fim, as tabelas de encaminhamento serão criadas pelo mecanismo de controle de fluxo e repassadas ao comutador que passará a tratar os pacotes oriundos dos dispositivos previamente autenticados.

5. Trabalhos Relacionados

Existem algumas propostas que buscam prover segurança a uma SDN, uma delas é o FortNOX, um mecanismo que verifica os fluxos gerados em uma rede OpenFlow que pos-

sua um controlador NOX. Tal mecanismo identifica contradições causadas por eventuais inserções de fluxos maliciosos. Mesmo relevante a segurança de uma SDN OpenFlow, o FortNOX é um mecanismo de autorização, não autenticando os elementos da rede, age após a estação estar autenticada na rede concedendo acesso com base nos fluxos dos dispositivos [Porras et al. 2012].

Certas propostas estão diretamente relacionadas a autenticação de estações finais em uma SDN, uma delas é o AuthFlow. O AuthFlow é um mecanismo de autenticação de portas e controle de acesso para redes OpenFlow. A ideia principal do AuthFlow é realizar a autenticação utilizando-se de protocolos da camada de enlace, fazendo o mapeamento da identidade usada na autenticação, em fluxos criados por uma dada estação autenticada. A troca de informações oriundas do processo de autenticação é feita por meio do padrão IEEE 802.1X e o protocolo EAP, o qual encapsula a troca de mensagens de autenticação entre a estação suplicante e um servidor de autenticação RADIUS [Mattos et al. 2014]. Apesar de oferecer um considerável avanço na segurança de uma SDN o AuthFlow não garante que a estação final é realmente quem diz ser. Um atacante com acesso físico ao comutador poderia identificar a porta na qual a estação final está conectada, copiar o endereço MAC e realizar uma tentativa de acesso à rede.

Outra proposta para autenticação de estações finais é a concessão de um endereço IP temporário e o posterior redirecionamento dos *h*osts para um sítio Web. Dessa forma, as estações que pretendem acessar a rede seriam isoladas e deveriam apresentar suas credenciais de acesso antes de serem autenticadas [Casado et al. 2007]. Tal proposta tem como requisito a presença de um navegar Web instalado nas estações finais, apresentando limitação para estações que não possuam interface gráfica instalada. Outra questão a ser analisada é o método de autenticação, que fica limitado a utilização de usuário e senha.

6. Conclusão

Pela observação dos aspectos analisados nesse artigo, entende-se que a segurança de uma SDN necessita de mecanismos eficientes para autenticação e controle de acesso. O protocolo OpenFlow fez com que o paradigma SDN evoluísse rapidamente de um conceito para uma realidade adotada por diversas empresas. Tal evolução trouxe consigo importantes benefícios como a separação eficiente entre os planos de dados e controle, porém apresentou alguns desafios de segurança a serem transpostos. A centralização lógica dos controladores e a criticidade de suas aplicações nos fazem entender quão importante é a autenticação da origem em uma arquitetura SDN. A falta de um mecanismo seguro para autenticação dos dispositivos finais poderia permitir que uma estação maliciosa disparasse ataques contra o controlador e provocasse indisponibilidade de recursos na rede. O desenvolvimento de um mecanismo de autenticação com base na utilização de certificados auto-assinados poderia garantir que uma estação final é realmente quem diz ser, dificultando o acesso de estações maliciosas na topologia.

Referências

Adams, C. and Lloyd, S. (2003). *Understanding PKI: concepts, standards, and deployment considerations*. Addison-Wesley Professional.

Barros, L. G. and Foltran Junior, D. C. (2008). Autenticação ieee 802.1 x em redes de computadores utilizando tls e eap.

Buchmann, J. A. (2002). Introdução à criptografia. *São Paulo: Berkeley Brasil*.

Casado, M., Freedman, M. J., Pettit, J., Luo, J., McKeown, N., and Shenker, S. (2007). Ethane: taking control of the enterprise. In *ACM SIGCOMM Computer Communication Review*, volume 37, pages 1–12. ACM.

Congdon, P., Aboba, B., Smith, A., Zorn, G., and Roese, J. (2003). Ieee 802.1 x remote authentication dial in user service (radius) usage guidelines. *RFC3580, September*.

Dang, Q., Polk, T., and Brown, D. R. (2010). Internet x. 509 public key infrastructure: Additional algorithms and identifiers for dsa and ecdsa.

Feamster, N., Rexford, J., and Zegura, E. (2014). The road to sdn: an intellectual history of programmable networks. *ACM SIGCOMM Computer Communication Review*, 44(2):87–98.

Guedes, D., Vieira, L., Vieira, M., Rodrigues, H., and 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*, 30(4):160–210.

Haleplidis, E., Denazis, S., Pentikousis, K., Salim, J. H., Meyer, D., and Koufopavlou, O. (2014). Sdn layers and architecture terminology. *Internet Engineering Task Force, Internet Draft, Aug*.

Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M., et al. (2013). B4: Experience with a globally-deployed software defined wan. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pages 3–14. ACM.

Kim, H. and Feamster, N. (2013). Improving network management with software defined networking. *Communications Magazine, IEEE*, 51(2):114–119.

Kreutz, D., Ramos, F., and Verissimo, P. (2013). Towards secure and dependable software-defined networks. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pages 55–60. ACM.

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74.

Nadeau, T. D. and Gray, K. (2013). *SDN: Software Defined Networks*. "O'Reilly Media, Inc.".

OpenFlow Specification-Version, O. S. (2013). 1.4. 0.

Porras, P., Shin, S., Yegneswaran, V., Fong, M., Tyson, M., and Gu, G. (2012). A security enforcement kernel for openflow networks. In *Proceedings of the first workshop on Hot topics in software defined networks*, pages 121–126. ACM.

Rothenberg, C. E., Nascimento, M. R., Salvador, M. R., and Magalhães, M. F. (2010). Openflow e redes definidas por software: um novo paradigma de controle e inovação em redes de pacotes. *Cad. CPqD Tecnologia, Campinas*, 7(1):65–76.

Scott-Hayward, S., O'Callaghan, G., and Sezer, S. (2013). Sdn security: A survey. In *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for*, pages 1–7. IEEE.