O uso da ferramenta SDNPWN como forma de Pentest em uma rede definida por software

1º Ten QCO Lamartine de Oliveira Medeiros*

RESUMO

As Redes Definidas por Software (Redes SDN) apresentam características específicas quanto ao fato de segurança e ataques cibernéticos. Com o avanço das tecnologias e arquiteturas de redes, surgiu a necessidade de otimização do monitoramento e controle das redes em camadas dos níveis mais altos do modelo OSI, mais precisamente na camada de aplicação. Com essa nova abordagem surgiu novas formas de ataques cibernéticos testando a vulnerabilidade destas estruturas. No presente trabalho foram abordadas as características das redes SDN, o funcionamento do protocolo OpenFlow. Foram abordadas as principais vulnerabilidades e os problemas de segurança que ocorrem na infraestrutura SDN e quais componentes deste tipo de rede são atacados. Também foram levantados os principais projetos que visam a mitigação dos ataques. Por fim foi utilizado o software MININET para simular uma rede SDN utilizando o protocolo OpenFlow Floodlight e o framework SDNPWN que possui uma série de módulos para reconhecimento, gerenciamento, ataque e exploração de redes SDN. As simulações tiveram por objetivo verificar o comportamento e respostas de uma rede SDN simulada mediante os comandos realizados pelo framework.

Palavras-chave: Redes Definidas por Software. SDNPWN. Openflow.

The use of the SDNPWN tool as a form of attack in a software-defined network

ABSTRACT

Software Defined Networks (SDN Networks) have specific characteristics regarding security and cyber attacks. With the advancement of network technologies and architectures, the need for optimization of the monitoring and control of networks in layers of the highest levels of the OSI model, more precisely at the application layer. With this new approach emerged new forms of cyber attacks testing the vulnerability of these structures. In the present work the characteristics of the SDN networks were discussed, the operation of the OpenFlow protocol. The main vulnerabilities and security issues that occurred in the SDN infrastructure and which components of this type of network are attacked were addressed. The main projects aimed at mitigating attacks have also been raised. Finally, we used the MININET software to simulate an SDN network using the OpenFlow Floodlight protocol and the SDNPWN framework that has a series of modules for the recognition, management, attack and exploitation of SDN networks. The simulations were designed to verify the behavior and responses of a simulated SDN network using the commands performed by the framework.

Keywords: Software Defined Networks. SDNPWN. OpenFlow.

1 INTRODUÇÃO

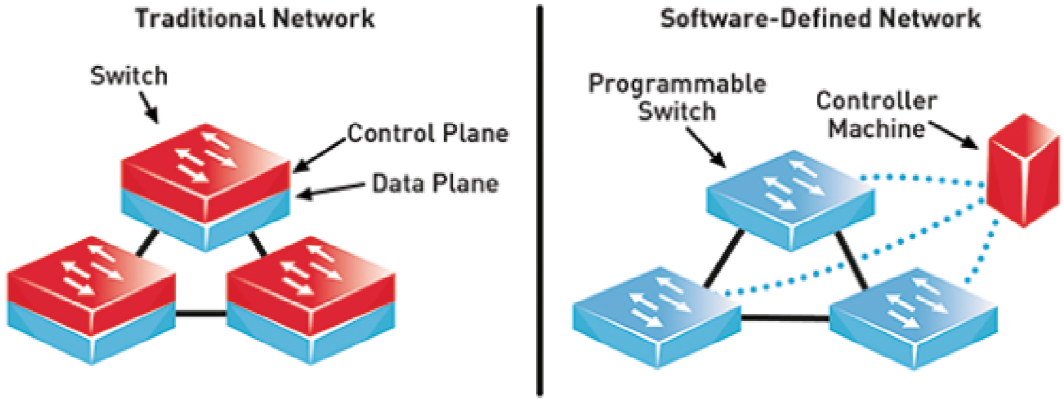
O sucesso das redes, sobretudo da internet é baseado em um princípio de distribuição de protocolos nos roteadores e switches permitindo que os pacotes se movam por toda a parte. Por conta disso o gerenciamento das redes tradicionais é considerada uma tarefa bastante desafiadora, com recursos bastante escassos de mecanismos de resposta e configurações para tarefas automatizadas. Aliado a isto, as redes IP existentes são verticalmente integradas ou seja: o plano de dados que encaminha o tráfego e o plano de controle que faz o encaminhamento são empacotados no mesmo dispositivo, dificultando as pesquisas de inovação, como fica evidenciado na lenta transição do IPv4 para o IPv6 (KREUTZ et al, 2015).

O uso das redes definidas por software (SDN) surge como uma proposta para viabilizar o controle e o gerenciamento das redes, tendo como principal objetivo resolver o problema da inflexibilidade das redes de computadores (GUEDES, 2012). A proposta é que as aplicações na rede: como roteamento inteligente, balanceamento de carga, controle de acesso e qualidade de serviço sejam implementadas para trabalhar em uma aplicação que utilize os recursos do Controlador SDN ao invés de trabalhar em cada roteador ou switch. Pode-se dizer que a rede definida por software é caracterizada pela existência de um sistema de controle, implementado em software, que pode controlar o mecanismo de encaminhamento dos elementos de comutação da rede por uma interface de programação bem definida (GUEDES, 2012).

Desta forma as promessas de agilidade, controle simplificado e programabilidade, que é a possibilidade de ajustar o comportamento de suas redes para oferecer suporte a novos serviços sem se preocupar com restrições de plataformas fechadas ou proprietárias, em tempo real são incentivos para a pesquisa e evolução das redes SDN, mas e quanto a segurança? Quais são as potenciais vulnerabilidades desta nova arquitetura? Quais os pontos frágeis e a mitigação que pode ser realizada?

Por ser uma nova tecnologia, a fragilidade em termos de segurança das SDN é motivo para crescimento de ataques através da descoberta de seus serviços e dispositivos (BOMFIM, 2017).

Figura 1—Rede tradicional e rede definida por software



Fonte: Bonfim (2013).

2 REDES DEFINIDAS POR SOFTWARE (RDS)

As Redes de Computadores foram projetadas e construídas como um conjunto de dispositivos de hardware com propósitos distintos entre si. Estas servem para processar todo o tipo de informação, inclusive o controle da própria rede como o monitoramento de tráfego e roteamento (CENTENO, 2016). Assim as redes vêm se tornando parte de uma infraestrutura crítica, uma vez que essa tecnologia permeia todos os níveis da sociedade, pois sua utilização está nos lares, na forma das redes domésticas, na rotina de implementação de políticas públicas, na forma do governo eletrônico, na educação, onde a internet se tornou uma das fontes essenciais de informação. Por conta disso paradigmas surgiram ao longo do tempo, onde podemos verificar, conforme Nadeau (2013):

a) o poder de processamento focado na evolução dos servidores: por ser mais rentável comercialmente, as pesquisas para o aumento do poder de processamento procuraram focar mais em funcionalidades específicas dos servidores, como por exemplo a execução de aplicativos de servidores de e-mail, servidores de banco de dados e outras funcionalidades que pudessem ser utilizadas pelos usuários;

b) a era da computação elástica: capacidade de aumentar ou reduzir rapidamente os recursos de

armazenamento, memória e processamento para atender às exigências de forma dinâmica. O controle é feito por ferramentas de monitoramento de sistema, denominadas *hypervisor,* que ajustam a quantidade de recursos alocados à quantidade de recursos realmente necessários sem interromper as operações;

c) a virtualização das redes: o conceito das redes virtuais tomou força com as redes virtuais privadas (VPN) possibilitando o isolamento de uma rede de forma segura, isolando totalmente o seu tráfego;

d) os equipamentos de rede: os equipamentos de rede viraram "caixas-pretas", ou seja, implementações integradas baseadas em hardware proprietário (COSTA, 2013). Mesmo com a evolução dos *data centers*, os equipamentos de rede permaneceram parados em termos de inovação, ou seja, além do aumento constante de velocidade na interface, as comunicações de dados não evoluíram muito desde o advento do IP, o *Multi-Protocol Label Switching* (MPLS), que é uma tecnologia de encaminhamento de pacotes, baseada em rótulos, que atua entre as camadas 2 e 3 do modelo *Open System Interconnection* (OSI) e tecnologias móveis (NADEAU, 2013);

e) a calcificação das redes: a estrutura das redes tornou-se calcificada, sendo que todas as pesquisas realizadas, principalmente para a Internet são no nível de aplicação. Neste nível é onde se tem controle e não no núcleo da rede, que seriam as camadas que possuem as funções de transporte e roteamento e se tornaram muito dependentes da tecnologia de fabricantes como CISCO, 3COM entre outros (GUEDES et al, 2012).

Com a proposta das redes SDN surge a possibilidade de uma nova arquitetura de rede capaz de ser programada sob demanda, ou seja, redes programáveis (COSTA, 2013). A SDN propõe a separação do plano de controle e do plano de dados de uma rede fazendo com que a mesma se adapte facilmente às alterações (NADEAU, 2013).

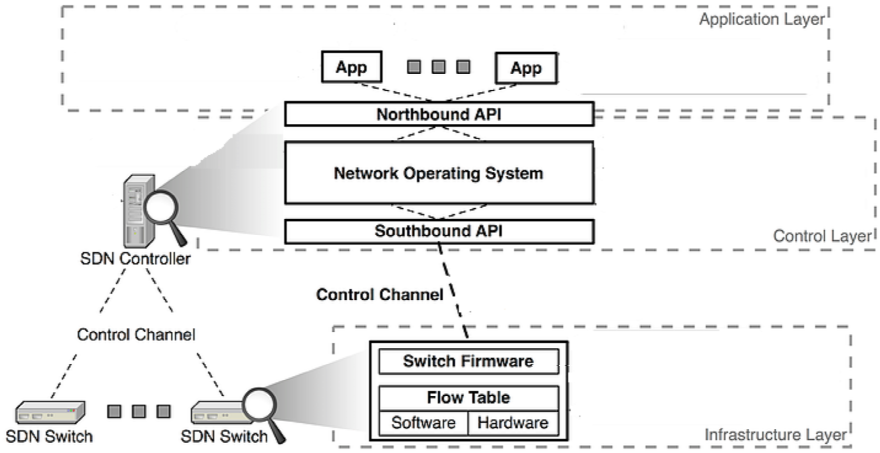
2.1 CARACTERÍSTICAS DAS REDES DEFINIDAS POR SOTFWARE

Em uma rede SDN existem três partes distintas separadas por camadas: aplicação, controlador e plano de dados. A camada de aplicação indica a parte que explora a dissociação do controle e do plano a fim de obter as metas específicas, como mecanismos de segurança ou soluções de gerenciamento de Internet. Esta é a camada em que aplicações e serviços definem o comportamento da rede (BONFIM, 2017).

O plano de dados lida com os pacotes de acordo com as instruções do Controlador. Normalmente, o plano de dados é o ponto final dos serviços do controlador e aplicações, não tendo apenas responsabilidade pelo roteamento ou descarte dos pacotes, mas também possui recursos para realização a classificação dos pacotes (BONFIM, 2017).

O plano de controle (Controlador) é responsável pelos protocolos e pela tomada de decisões que resultam na tabela de encaminhamentos, controlando dessa forma como o plano de dados vai encaminhar seus conteúdos (COSTA, 2013).

Figura 2 -Infraestrutura SDN e principais pontos de ataque



Fonte: SDN SECURITY, (2016) adaptado pelo autor.

A comunicação entre o plano de controle e o plano de dados, ou seja, entre o *controller* e seus dispositivos de rede ocorre através da *Southbound Interface* (SBI). Para que ocorra essa comunicação são necessários 06 itens que o Controlador precisa saber:

a) os dispositivos que estão na rede;

b) o que cada dispositivo é capaz de fazer;

c) quais portas e interfaces que tem o dispositivo;

d) o atual estado de cada porta;

e) topologia;

f) configuração dos dispositivos (BONFIM, 2017).

As alterações necessárias serão realizadas entre as aplicações (APIs) do Controlador através da *Northbound Interface* (NBI). Assim, a SBI coleta as informações, encaminha ao Controlador, que por sua vez passa para a NBI, que coleta essas informações e envia para a API ou aplicações de software que envia de volta instruções para alteração dos pacotes, quando necessário via SBI para o plano de controle dos equipamentos (BONFIM, 2017).

2.2 CONTROLADOR

Em linhas gerais, o Controlador é um sistema de software ou uma coleção de sistemas que juntos proveem:

a) gerenciamento do estado da rede, com informações de configurações, topologia aprendida e informações da sessão de controle;

b) um modelo de dados de alto nível que captura os relacionamentos entre os recursos gerenciados, políticas e outros serviços providos pelo controlador;

c) uma interface de programação de aplicativos (API) moderna com o objetivo de facilitar a interação entre o Controlador e as aplicações;

d) uma sessão segura de controle do *Transmission Control Protocol* (TCP) entre o controlador e os agentes associados nos elementos de rede;

e) um protocolo baseado em padrões para provisionamento de rede orientada a aplicativos de acordo com o estado dos elementos da rede;

f) mecanismo de descoberta de rede, topologia e serviço; um sistema de descoberta de rotas e potencialmente outros serviços centrados na rede ou informações de recursos centrados (NADEAU, 2013);

Para os softwares de *switches*/roteadores o controlador SDN é uma crítica interface de gerenciamento, o qual fornecem serviços de provisionamento e descoberta de redes, sendo responsáveis pelo estado associado das entidades da rede (NADEAU, 2013).

2.3 PROTOCOLO *OPENFLOW*

Originalmente concebido como um protocolo para experimentos acadêmicos, evoluiu a ponto de ser utilizado em substituição aos protocolos de camada 2 e 3 completamente em *switches* comerciais e roteadores (NADEAU, 2013). A proposta do *OpenFlow* promove a criação das redes SDN com a utilização de elementos comuns como *switches* e roteadores, pontos de acesso ou computadores pessoais (COSTA, 2013).

Umas das vantagens de se utilizar a arquitetura *OpenFlow* é a flexibilidade que ela oferece para se programar de forma independente, o tratamento de cada fluxo da rede e como ele dever ser ou não encaminhado. O *OpenFlow* determina como o fluxo dever ser definido, quais serão as ações que podem ser realizadas por pacote do fluxo e quais protocolos de comunicação devem ser utilizados entre o controlador e os comutadores para realizar as definições de fluxo e ação (COSTA, 2013).

3 ASPECTOS DE SEGURANÇA E VULNERABILIDADES DAS REDES SDN

Considerada uma tecnologia recente, a segurança das redes SDN tornou-se uma questão prioritária a ser resolvida, necessitando de um ambiente simples, escalável e eficiente. Como inicialmente, segurança não foi considerado como parte do desenvolvimento das SDN, cada camada possui implicações e requerimentos que necessitam serem avaliadas como questões de segurança. A rede necessita de um *framework* robusto que garanta a direção correta do controlador. Apesar de que a segurança deveria ser construída como parte da arquitetura SDN, ela seria entregue como um serviço para prover privacidade e integridade de todos os recursos conectados (BONFIM, 2017).

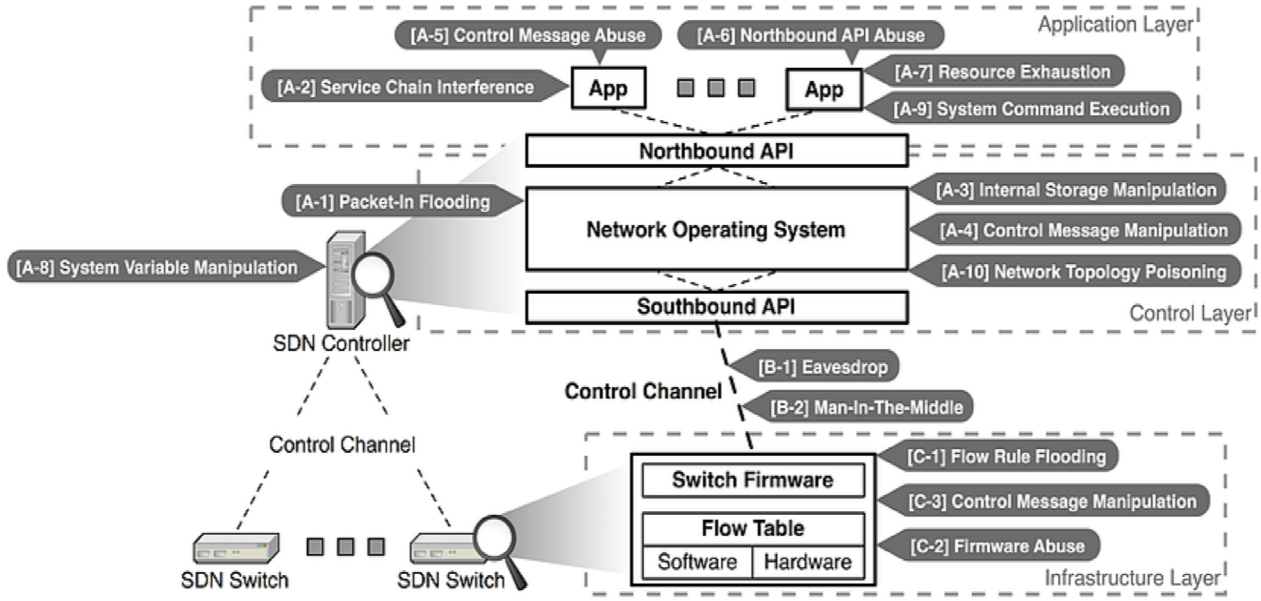
Em 2013 os estudantes Seungwon Shin e Guofei Gu publicaram um artigo sobre segurança em Redes SDN chamado "Atacando redes definidas

por software: uma primeira viabilidade de estudo" (SHIN; GU, 2013) onde realizaram diversos laboratórios com a arquitetura SDN sendo levantados vários problemas de segurança, que segundo os autores não poderiam ser ignorados. O objetivo era dar o ponta pé para diversos estudos na área. Logo após este estudo, ainda em 2013 foram criados os sítios eletrônicos http://www.sdnsecurity.org (SDN SECURITY, 2017) e http:// www.openflowsec.org (OPEN FLOW SEC, 2017) com o objetivo de estimular pesquisadores a divulgarem seus trabalhos sobre segurança em redes SDN. A comunidade do sdnsecurity.org preocupa-se com as falhas de segurança na arquitetura SDN e a comunidade do openflowsec.org é voltada mais para as falhas de segurança no protocolo *OpenFlow* (SDN SECURITY, 2017).

Os colaboradores da comunidade sdnsecurity.org criaram o Projeto GENOMA (SDN SECURITY, 2017) com o objetivo de focar as vulnerabilidades do ambiente de redes SDN, e visa sistematizar ou caracterizar as vulnerabilidades existentes, além disso, encontrar novas vulnerabilidades que não foram relatadas, procurando desenvolver de forma séria, cada vez mais ambientes seguros de redes SDN (SDN SECURITY, 2017). Segundo este estudo os ataques SDN foram divididos em três categorias: a) plano de controle específico: que inclui todos os casos de ataques contra controle SDN e camada de aplicação; b) canal de controle específico: que visam todos os ataques a interface, como por exemplo o *OpenFlow;* e c) plano de dados específico: que estuda os ataques em dispositivos que suportam funções SDN (SDN SECURITY, 2017).

Baseado nisso foram levantados os principais tipos de ataques conforme demonstrado na figura 2, onde são observadas quais ameaças agem dentro da estrutura SDN e quais dispositivos são atacados.

Figura 3 - Infraestrutura SDN com seus principais pontos de ataque e tipos de ataques



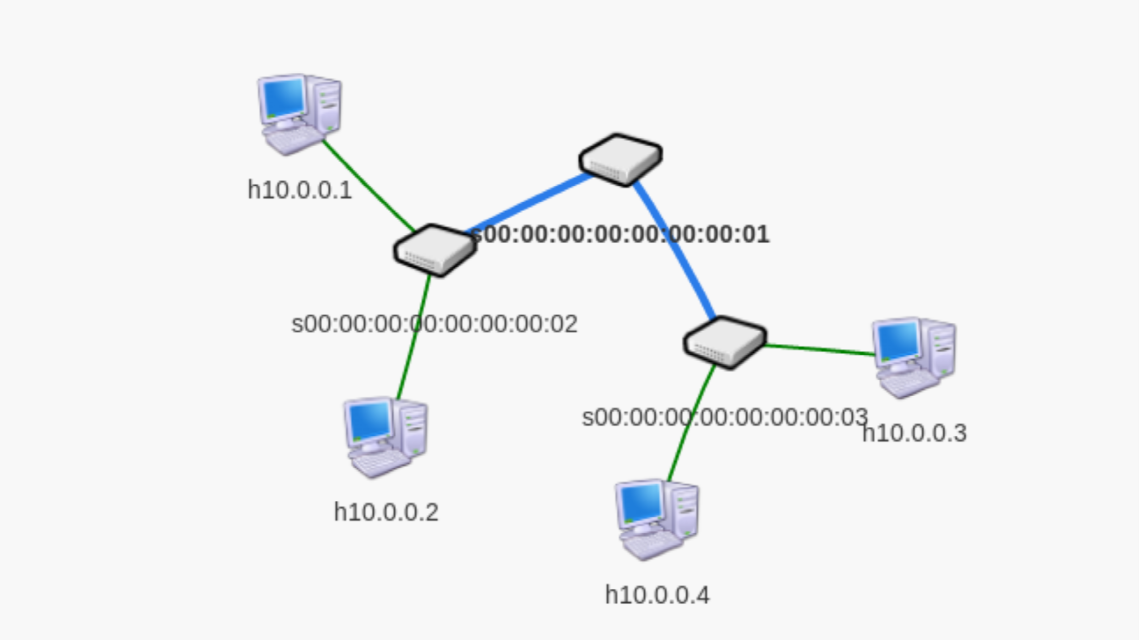
Fonte: SDN Security (2013).

Com o objetivo de mitigar essas vulnerabilidades foram criados uma série de projetos de pesquisa em segurança de redes SDN que são mantidos pelos pesquisadores e colaboradores do sdnsecurity.org.

4 ESTUDO DE CASO

Por serem consideradas estruturas críticas e devido ao alto valor dos equipamentos em ambientes reais, para viabilizar a situação optou -se pela utilização do *MININET* (MININET.ORG, 2017) (www.mininet.org) que é uma ferramenta que permite emular os componentes típicos de uma rede SDN (CENTENO, 2016).

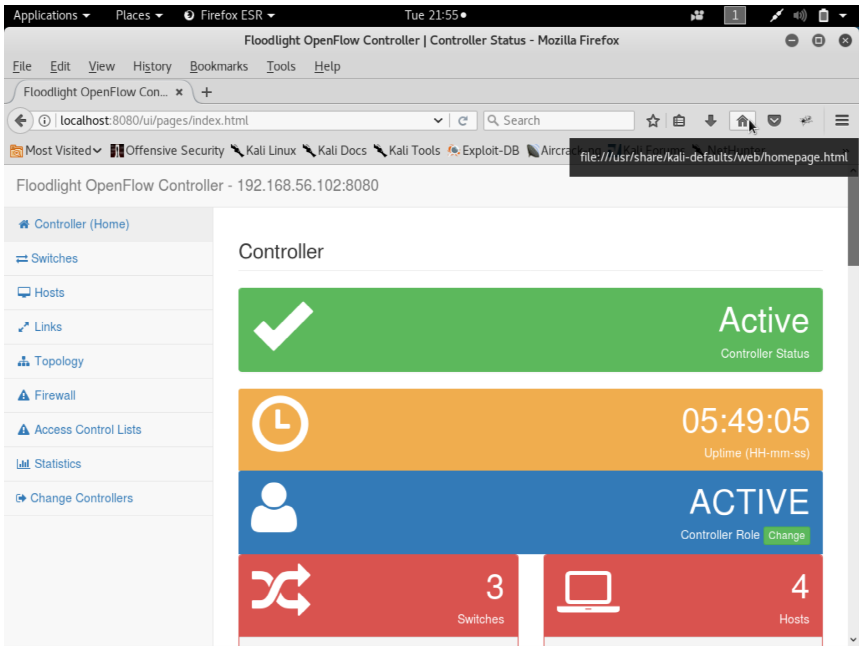
Figura 4 - Topologia da Rede Definida por Software criada pelo MININET



Fonte: o autor (2018).

O simulador *MININET* permite uma rápida simulação de uma grande infraestrutura virtual de rede, executando em um *kernel* real, com a utilização de apenas um computador. O *MININET* cria uma rede virtual *OpenFlow* com um controlador, *switches*, *hosts* e *links,* permitindo também desenvolver topologia personalizadas utilizando *scripts* em *python* (MARCHESAN; MEDINA, 2015). Para a realização dos testes foi utilizado o protocolo *OpenFlow Floodlight*, sendo escolhido por conta do carregamento de módulos extensíveis da linguagem java, como por exemplo o controle via *web* (CENTENO, 2016).

Figura 5 - Módulo web do Controlador Floodlight



Fonte: o autor (2018).

4.1 FERRAMENTA DE TESTE DE PENETRAÇÃO PARA REDES DEFINIDAS POR SOFTWARE

A ferramenta chamada Software Defined Networking Pown (SDNPWN) é um kit de ferramentas de teste de penetração para redes definidas por software (SDN), sendo que possui uma série de módulos a serem utilizados para experimentos e ataques a redes SDN. Pode-se dizer que o SDNPWN é um conjunto de ferramentas que inclui módulos para reconhecimento, outros para ataque e para explorar vulnerabilidades em controladores SDN. Escrito em python 3, o SDNPWN visa habilitar os ataques genéricos de SDN a serem realizados, ao mesmo tempo que permite vulnerabilidades específicas a serem exploradas. O SDNPWN possui 17 módulos; sendo 03 módulos de reconhecimento, 04 módulos de gerenciamento e 10 módulos de ataque e exploração. Cada módulo contém o código usado para uma função específica de ataque ou exploração, com outros módulos atuando como bibliotecas (SMITH, 2016).

4.1.1 Reconhecimento

As capacidades de reconhecimento do sdnpwn estão separadas em três módulos; arpmon, sdn-detect e controller-detect. Todos esses módulos tem o objetivo de identificar os componentes de uma rede SDN. O módulo arpmon pode ser usado para imprimir informações do tráfego ARP capturado em uma determinada interface. O módulo sdn-detect é usado para verificar se uma rede é ou não uma SDN. O módulo controller-detect é usado para identificar o controlador na rede, essa identificação é realizada monitorando o tráfego do protocolo de descoberta de camada de ligação (LLPD), ou enumerando a interface norte do controlador (SMITH, 2016).

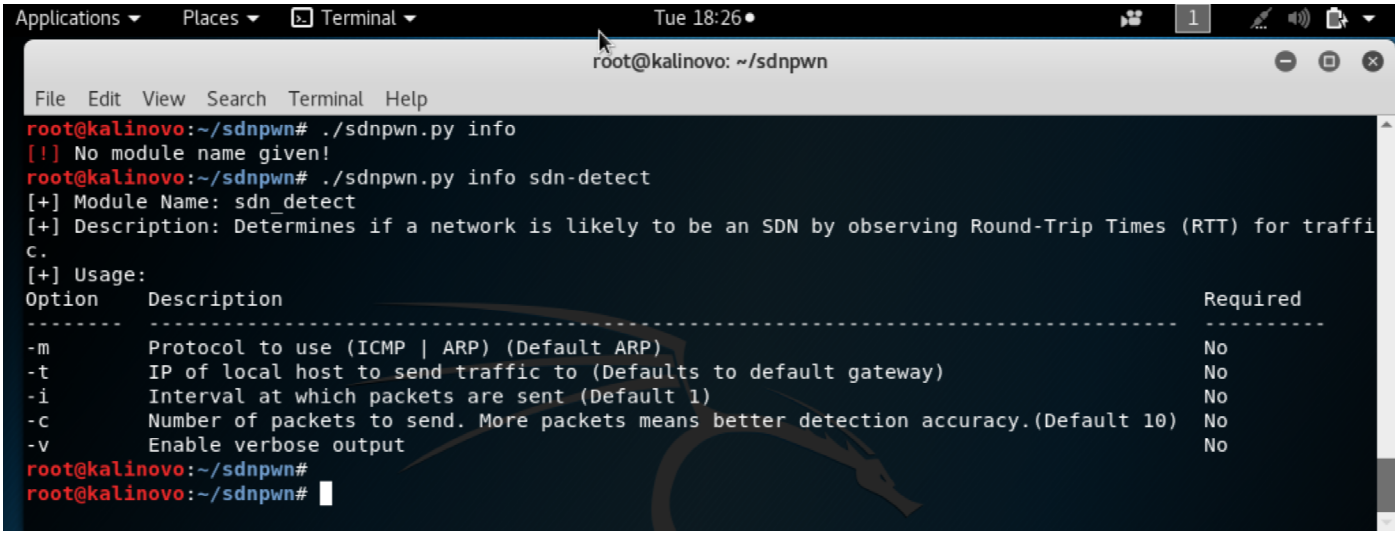
4.1.2 Gerenciamento

Existem quatro módulos de gerenciamento que servem para gerenciar as principais informações sobre a ferramenta e sobre o uso de cada módulo, sendo eles o módulo help que auxilia no uso da ferramenta, o módulo mods lista todos os módulos executáveis que estão na ferramenta, o módulo info mostra a descrição e as opções disponíveis para um módulo específico da ferramenta e o módulo system que é acessado por uma infraestrutura de linguagem comum (Common Languace Infraestructure – CLI), que define um ambiente que permite a utilização de multiplas linguagens de alto nível em diferentes plataformas, mas sem a necessidade de serem reescritas para uma arquitetura específica.

4.1.3 Ataque

Existem ainda os módulos que são utilizados para ataque e exploração de vulnerabilidades da rede SDN e principalmente do controlador SDN. O lfa-relay, lfa-scapy e lldp-replay são usados para executar o Link Fabrication Attack (LFA). O módulo host-location-hijack é usado para executar o sequestro de localização de host. O módulo of-switch é usado para se conectar a um controlador usando o OpenFlow com uma versão de switch personalizada. Isso é útil para reunir padrões de fluxo, testar a segurança do canal de controle e explorar as vulnerabilidades no controlador através da configuração de switch personalizada (SMITH, 2016).

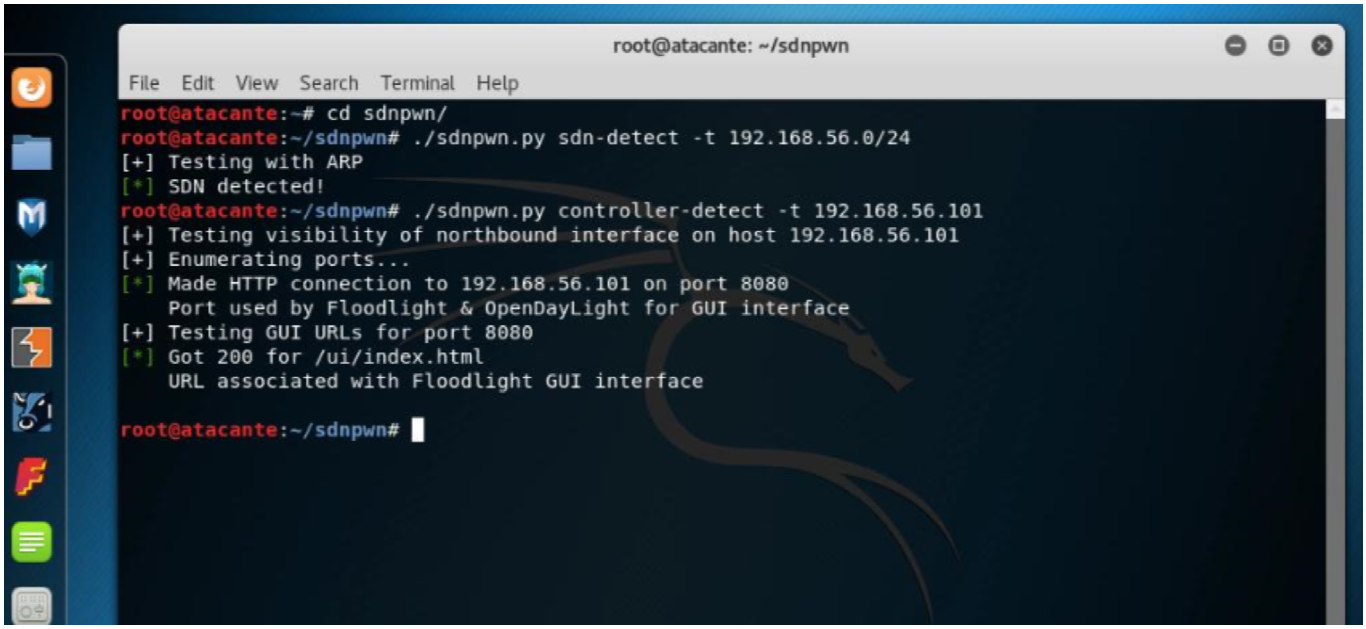
Figura 6 - Exemplo de uso do módulo de informações



Fonte: o autor (2018)

Como exemplo de invasão foi utilizado o módulo of-gen da ferramenta. Através dos parâmetros utilizados com esse módulo foi possivel demonstrar a exploração da vulnerabilidade OpenDayLight CVE-2017-1000357. Esta vulnerabilidade afetou a versão 3.3 do protocolo FloodLight (lítio-SR3), utilizada no teste, onde o invasor que possua acesso ao canal do controlador SDN e desde que o TLS não esteja habilitado para conexões de switch, consegue inundar uma grande quantidade de mensagens do OpenFlow Hello a uma alta taxa, sendo usado o módulo of-gen do SDNPWN. Com esse módulo o controlador SDN de um endereço IP conhecido, na porta 6653 será inundado por pacotes a uma taxa de 1 a .0001 segundos até que 100.000 mensagens tenham sido enviadas (SMITH, 2016). Primeiramente foram realizadas as detecções da rede SDN e do protocolo utilizado.

Figura 7 - Identificando a rede SDN e seu controlador

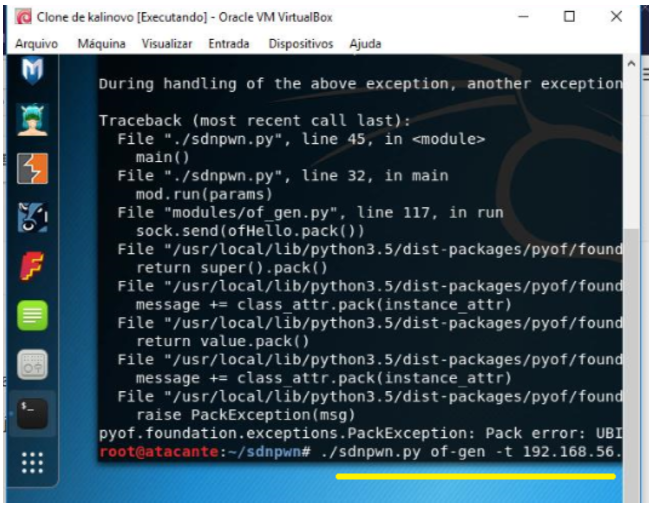


Fonte: o autor (2018)

Logo após utilizando o módulo of-gen realizou-se a inundação de pacotes conforme a Figura 7.

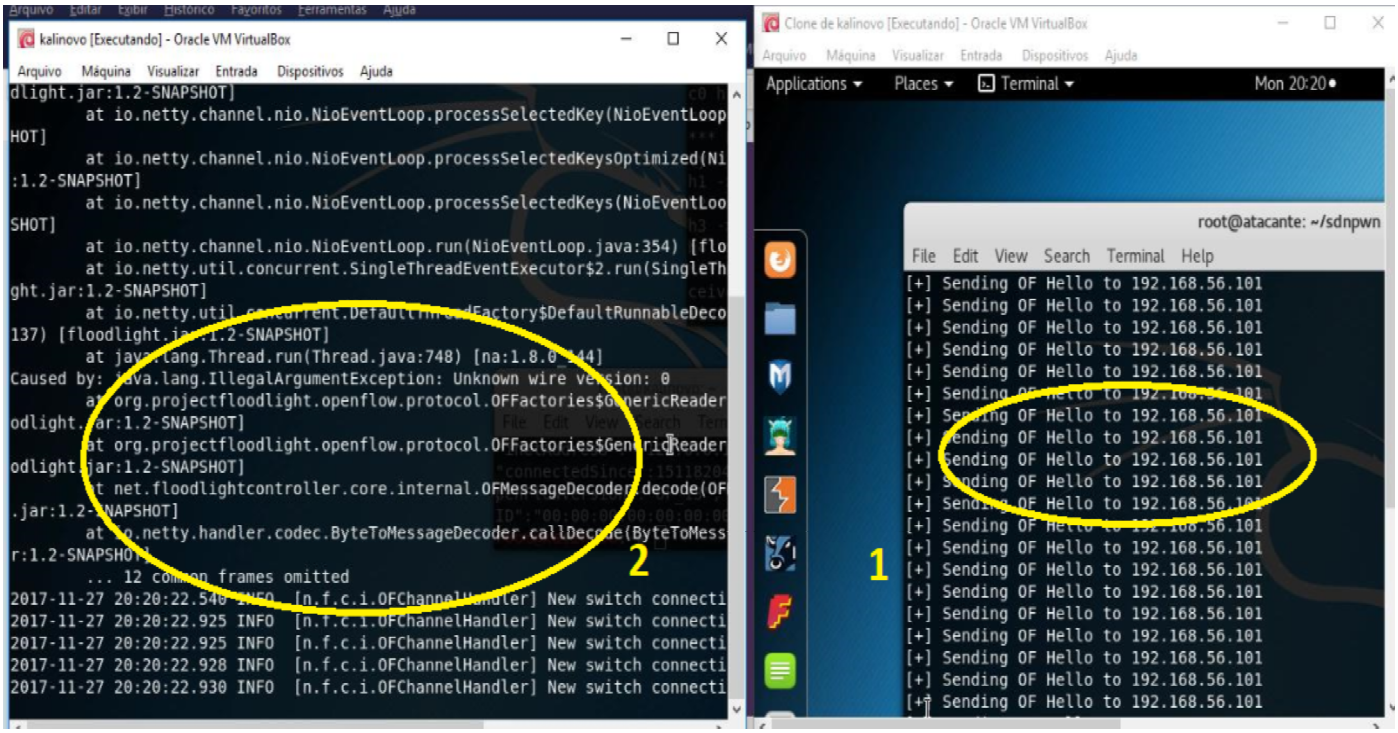
Na Figura 8 verificou-se a dificuldade do controlador tentar restabelecer o serviço e não conseguindo.

Figura 8 - Aplicando o módulo *of-gen*



Fonte: o autor (2018)

Figura 9 - Inundação de pacotes sobre o controlador utilizando o módulo *of-gen*



Fonte: o autor (2018).

E por fim o resultado do ataque utilizando o módulo *of-gen,* conforme figura acima.

A partir da descoberta desta vulnerabilidade foram criados patchs de correção a este problema. Como mitigação do problema recomenda-se que, em caso de uso de uma versão mais antiga, seja a mesma atualizada. (SMITH, 2016).

5 CONCLUSÃO

O presente trabalho buscou apresentar um exemplo de vulnerabilidade e as possíveis falhas de segurança, ressaltando que por ser uma tecnologia recente durante a elaboração do projeto de arquitetura, não houve um cuidado específico referente a segurança das Redes Definidas por Software. Por conta disso, viu-se a necessidade de criação de uma equipe de colaboradores através do SDNSECURITY.ORG, o qual procura criar projetos referentes as vulnerabilidades específicas, procurando resolver os problemas de segurança encontrados.

Através da utilização do simulador SDN MININET juntamente com o controlador Floodlight, foi implementada a utilização do framework SDNPWN.

Este framework foi desenvolvido em python e possui uma série de módulos que possibilitam verificar o reconhecimento, gerenciamento e ataque em redes SDN.

As Redes Definidas por Software por serem uma tecnologia recente, embora tenham a projeção de grandes avanços para o desenvolvimento de novos serviços e soluções, ainda carecem de muitas pesquisas, especialmente na área de segurança.

Caso não seja possível a atualização, a exposição dos serviços OpenFlow do controlador devem ser limitadas a dispositivos confiáveis. Se o OpenFlow não estiver sendo utilizado como um protocolo Southbound, os serviços OpenFlow devem ser desativados (SMITH, 2016).

Por fim, como forma de sugerir novos trabalhos, pode-se citar o desenvolvimento de novos módulos com o objetivo de fazer parte da ferramenta SDNPWN, agregando novos conhecimentos, contribuindo tanto na formação técnica quanto cientifica dos interessados no assunto.

REFERÊNCIAS BIBLIOGRÁFICAS

BOMFIM, Leonardo Henrique da Silva. **Um serviço para anonimização em redes definidas por software.** Sergipe: UFS, 2017. Disponível em < https://bdtd.ufs.br/bitstream/tede/3767/2/LEONARDO_HENRIQUE_SILVA_BOMFIM.pdf >. Acesso em: 31 ago. 2017.

CENTENO, Paulo Vieira. **Uma análise de segurança das redes definidas por software sobre o protocolo OpenFlow**. Florianópolis: UFSC, 2016. Disponível em : < https://repositorio.ufsc.br/bitstream/handle/123456789/171402/monografia_tcc_paulo_centeno.pdf?sequence=1&isAllowed=y > . Acesso em: 3 set. 2017.

COSTA, Lucas Rodrigues. **OpenFlow e o paradigma das redes definidas por software**. Brasília: UNB, 2013. Disponível em < http://bdm.unb.br/handle/10483/5674 >. Acesso em: 31 ago. 2017.

GUEDES, Dorgival et al. **Redes Definidas por Software**: uma abordagem sistêmica para o desenvolvimento de pesquisas em redes de computadores. In: XXX SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, p. 160-210, 2012. Disponível em: < http://homepages.dcc.ufmg.br/~mmvieira/cc/papers/minicurso-sdn.pdf >. Acesso em: 2 set. 2017.

KREUTZ, Diego et al. Software-defined networking: A comprehensive survey. **Proceedings of the IEEE**. [S.l.]. v. 130, p. 10–76, 2015. Disponível em: < https://pdfs.semanticscholar.org/d8bd/4c1e92420200bd29cb1a233bd81eb3c28bba.pdf >. Acesso em: 2 set. 2017.

MARCHESAN, Gabriel; MEDINA, Roseclea Duarte. **Simulando cenários para redes definidas por software. 2015.** UFSM. Disponível em < http://eati.info/eati/2015/assets/anais/Longos/L20.pdf>. Acesso em: 2 nov. 2017.

MININET.ORG. Disponível em < https://www.mininet.org/>. Acesso em: 7 nov. 2017.

NADEAU, Thomas D.; GRAY, Ken**. SDN:** Software Defined Networks. California. O' Reilly Media, 2013.

OPENFLOWSEC.ORG. Disponível em < https://openflowsec.org/>. Acesso em: 7 nov. 2017. [**SDNSECURITY.ORG**. Disponível em < https://sdnsecurity.org/>. Acesso em: 7 nov. 2017.

SHIN, Seungwon; GU, Guofei. **Attacking Software-Defined Networks**: a first feasibility study. 2013. Disponível em:<http://conferences.sigcomm.org/sigcomm/2013/papers/hotsdn/p165.pdf >. Acesso em: 2 nov. 2017.

SMITH, Dylan. **SDNPWN:** pratical software-defined network security. 2016. [S.l.] Disponível em < https://sdnpwn.net/>. Acesso em: 7 nov. 2017.

*Artigo realizado a partir do trabalho de conclusão do Curso de Especialização em Guerra Cibernética para Oficiais do Centro de Instrução de Guerra Eletrônica (CIGE) em 2017 pelo 1º Tenente Quadro Complementar de Oficiais Lamartine de Oliveira Medeiros do Exército Brasileiro. Email: lamartine.medeiros@eb.mil.br