ISSN: 2175-1897

Testes de desempenho com o módulo de segurança L3-ARPSec em Redes Definidas por Software

Rogério Leão Santos de Oliveira[1]; Christiane Marie Schweitzer[2]; Ailton Akira Shinoda[3]; Ligia Rodrigues Prete[4]; Tiago Ribeiro Carneiro[5]

Resumo - O protocolo de resolução de endereços (ARP) é usado para mapear endereços IP a endereços MAC em redes locais. Este protocolo possui algumas vulnerabilidades de segurança e uma delas é ataque Man-in-the-Middle (MITM). O conceito de Redes Definidas por Software (SDNs) representa uma abordagem inovadora na área de redes de computadores, uma vez que propõe um novo modelo para o controle de repasse e roteamento dos pacotes de dados que navegam na Internet. Este trabalho apresenta resultados de testes de desempenho com o módulo L3-ARPSec, um conjunto de instruções escritas em linguagem de programação Python que propõe uma maneira de controlar a troca de mensagens ARP e também mitigar o ataque MITM em redes locais.

Palavras-chave: Redes Definidas por Software. *OpenFlow*. Envenenamento de cache ARP. MITM. Desempenho de rede.

Abstract - The Address Resolution Protocol (ARP) is used to map IP addresses to MAC addresses in local area networks. This protocol has some security vulnerabilities and one of them is the Man-in-the-Middle (MITM) attack. Software-Defined Networks (SDNs) represent an innovative approach in the area of computer networks, since they propose a new model to control forwarding and routing data packets that navigate the World Wide Web. This study presentes performance results of the module L3-ARPSec, a set of instructions written in the Python programming language that proposes a way to control the switching of ARP messages and also mitigates the MITM attack in local area networks.

Keywords: Software-Defined Network. OpenFlow. ARP cache poisoning. MITM. Network performance.

[1] Centro Paula Souza - Faculdade de Tecnologia de Jales – SP – Brasil - rogerio.leao@fatec.sp.gov.br
[2] Universidade Estadual Paulista “Julio de Mesquita Filho” – SP – Brasil - chris@mat.feis.unesp.br
[3] Universidade Estadual Paulista “Julio de Mesquita Filho” – SP – Brasil - shinoda@dee.feis.unesp.br
[4] Centro Paula Souza - Faculdade de Tecnologia de Jales – SP – Brasil - ligia.prete@fatec.sp.gov.br
[5] Centro Paula Souza - Faculdade de Tecnologia de Jales – SP – Brasil – tiago.carneiro01@fatec.sp.gov.br

1. Introdução

A Internet de hoje é sem dúvidas um dos mais importantes sistemas de engenharia criado pela humanidade, com centenas de dispositivos conectados, enlaces de comunicação e comutadores; centenas de milhares de usuários que se conectam esporadicamente por meio de telefones celulares e assistentes digitais pessoais (PDAs, Personal Digital Assistants); e dispositivos como sensores webcams, console para jogos, quadros de imagens, e até mesmo eletrodomésticos conectados à Internet (KUROSE; ROSS, 2010).

Com o intuito de sempre melhorar o desempenho e segurança da Internet, a comunidade de pesquisa em redes de computadores tem investido em inciativas que levem à implantação de redes com maiores recursos de programação, de forma que novas tecnologias e funcionalidades possam ser inseridas na rede de forma gradual e sem grande impacto financeiro. Exemplos de iniciativas desse tipo são as propostas de redes ativas (active networks) (TENNENHOUSE; WETHERALL, 2007), de testbeds como o PlanetLab (PETERSON; ROSCOE, 2006) e, mais recentemente, projeto GENI (TURNER, 2006) e (ELLIOTT; FALK, 2009). Redes ativas, apesar de seu potencial, tiveram pouca aceitação pela necessidade de alteração dos elementos de rede para permitir que se tornassem programáveis (GUEDES et al., 2012).

Dentro deste contexto surgiu um novo paradigma chamado Redes Definidas por Software (Software Defined Network - SDN). Uma estrutura que tem o objetivo de garantir o desempenho atual alcançado, no repasse e roteamento dos pacotes de dados, pois preserva a atual estrutura com os roteadores dedicados fazendo seu trabalho de retransmissão, mas que ao mesmo tempo delega a política de como isso será feito para um novo componente, chamado de controlador de rede.

Este novo modelo de rede controlada por aplicações gerencia a transferência de pacotes na rede, mas não interfere nos atuais protocolos das camadas, como ARP, IP, TCP, UDP e HTTP. Desta maneira, problemas já reconhecidos que afetam a estrutura atual também ocorrem nas redes definidas por software. Um destes ataques é o Man-in-the-Middle (MITM), em que um atacante usando mensagens ARP fraudulentas se posiciona no meio de uma comunicação. Depois de configurado e estabelecido o ataque, o fraudador consegue visualizar todas mensagens trocadas pelas vítimas.

Este trabalho tem como foco avaliar o desempenho do módulo de programação para controladores SDN chamado L3-ARPSec. Proposto por Oliveira (2015), este módulo tem o objetivo de controlar as mensagens ARP e também mitigar o ataque de MITM em redes locais.

As seções seguintes estão organizadas da seguinte maneira: Na seção 2 são explicados alguns aspectos das redes SDN, a estrutura do protocolo ARP e como o ataque MITM ocorre. Na seção 3, o módulo L3-ARPSec é suscintamente

apresentado. Na seção 4 os testes de desempenho são executados e os resultados são discutidos a cada teste. Finalmente na seção 5, as considerações finais e sugestões para trabalhos futuros são apresentadas.

2. Referencial Teórico

2.1. Redes Definidas por Software

A arquitetura de rede SDN é formada por três elementos: controladores, elementos de comutação programáveis e o protocolo padrão de comunicação entre os controladores e os elementos de rede.

Os controladores, também conhecidos como sistemas operacionais de rede oferecem um ambiente de programação onde o desenvolvedor ou administrador pode ter acesso aos eventos gerados por uma interface de rede que siga um protocolo padrão como o *OpenFlow* e podem também gerar comandos para controlar a infraestrutura de repasse e roteamento dos pacotes (OLIVEIRA et al., 2014).

Esta nova estrutura de controle da rede permite um gerenciamento mais efetivo e independente. Os controladores podem implementar lógicas de monitoramento do tráfego mais sofisticadas, por exemplo, uma solução que ofereça novas abstrações para os usuários, dando a cada um, a visão de que suas máquinas estão ligadas a um switch único e privado, independente dos demais.

Os comutadores e roteadores de rede, anteriormente independentes e autônomos, passam a ser configurados pelos controladores de rede que podem implementar as políticas de repasse, roteamento e de segurança baseadas em níveis de abstração maiores que o modelo atual pois foram definidas anteriormente pelos administradores de rede nos controladores.

2.2. O Protocolo ARP e o ataque MITM

Todos os dispositivos conectados a uma rede TCP/IP são identificados na camada de rede por seu endereço IP de 32 bits e na camada de enlace por seu endereço MAC de 48 bits. Em uma comunicação, quando a camada de rede recebe um pacote das camadas superiores a fim de enviá-lo a um determinado endereço IP, ela verifica se este endereço está na mesma rede local do emissor. Se estiver, o pacote deverá ser entregue a camada de enlace que o envia a interface física apropriada. Para que isto ocorra é necessário que o emissor também já conheça o endereço MAC do destinatário.

Este gerenciamento é feito pelo protocolo de resolução de endereços (ARP), que automaticamente mapeia endereços IP para endereços MAC usando

mensagens de requisições e resposta ARP. Cada host na rede possui este mapeamento dentro de uma tabela temporária chamada cache ARP. Quando necessário qualquer host pode procurar em sua própria tabela ARP para encontrar mapeamentos de endereços.

O envenenamento de cache ARP acontece quando um atacante maliciosamente modifica o mapeamento de um endereço IP para seu próprio endereço MAC na tabela ARP de outros hosts (PHILIP, 2007). Esta técnica, também chamada de fraude ARP (ARP *spoofing*), ocorre com o envio de respostas ARP para todos os hosts em uma rede local de forma indiscriminada e mesmo sem requisições anteriores.

Ao fraudar as tabelas ARP de dois comunicantes, o atacante se instala no meio da comunicação e todo tráfego recebido por ele de alguma das vítimas poderá ser visualizado e imediatamente repassado ao destino de forma discreta. Desta maneira as vítimas conseguem se comunicar normalmente e, portanto, podem nem perceber que são vítimas do conhecido ataque MITM.

3. O módulo controlador L3-ARPSec

O módulo L3-ARPSec é um conjunto de instruções escritas em linguagem de programação Python que é executado no controlador. Ele propõe uma maneira de controlar a troca de mensagens ARP e ao mesmo tempo combater o ataque MITM em redes locais.

Ao ser executado no controlador, o módulo L3-ARPSec instancia duas tabelas virtuais ARP que são alimentadas com informações de endereços IP e ARP de todos os dispositivos comunicantes da rede. Com as tabelas cheias de valores, algoritmos de verificação periódicos vasculham por tentativas de ataques de duas maneiras diferentes: A primeira é através de uma função de detecção de inundação de ARPReply, onde o atacante é descoberto pois está enviando respostas ARP com alta frequência na rede e em uma situação normal isso não ocorre; A segunda forma de detecção do ataque é a verificação das tabelas virtuais ARP. Neste caso o atacante é identificado pois as tabelas virtuais conterão dois ou mais endereços IP mapeados para um único endereço MAC.

A punição frente a descoberta do ataque em ambos os casos é o bloqueio do endereço MAC na rede pelo tempo inicial de dois minutos. Após este período, o computador atacante pode voltar a se comunicar na rede, porém se reincidir no ataque a punição será adaptativa, e o tempo de bloqueio será multiplicado pelo número de tentativas, resultado assim em um período cada vez maior de punição.

Mais detalhes sobre o funcionamento do módulo, bem como testes de funcionamento são descritos no trabalho de Oliveira (2015), restando a este artigo

apenas o foco de relatar e discutir os resultados de testes de desempenho deste módulo.

A análise de desempenho do módulo e sua comparação com uma rede de estrutura tradicional, se faz necessário pois de nada adianta possuir um mecanismo que garanta a segurança frente a um determinado ataque, mas que não consiga suportar desempenho mínimo e suficiente nas operações básicas de repasse e roteamento de pacotes na rede. Se o módulo obtiver desempenho negativo frente a demanda da rede, os administradores poderão ficar desencorajados em utilizá-lo.

4. Testes de desempenho e resultados

4.1. Desempenho Utilizando uma Rede Tradicional

Para a execução deste teste, foi criada uma rede local semelhante a estrutura de um laboratório de informática no modelo tradicional.

O equipamento adotado para os testes foi o switch de marca TP-Link, modelo TL-WR1043ND que contem cinco portas de conexão do tipo ethernet cabeada e ainda permite comunicação de dispositivos sem fio. Inicialmente, nenhuma alteração foi feita no sistema operacional do aparelho, desta forma ele repassa os pacotes de uma porta a outra no modelo tradicional, de forma independente. Todos os microcomputadores possuem com processadores Intel® Core™ i5-2400 CPU @ 3.10GHz, 4 GB de memória RAM, disco rígido de 300 GB e executam Sistema operacional Windows 7 64 bits. Os cabos permitem a transferência de até cem Megabits por segundo entre todos os nós da rede.

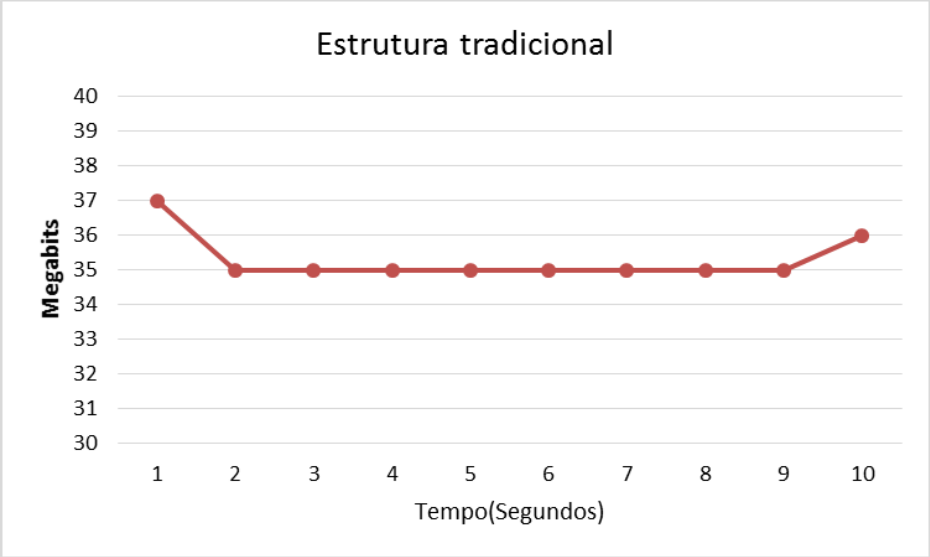
Todos os computadores foram configurados manualmente com endereços IP na faixa de 192.168.0.0/24 e para não influenciar nos resultados dos testes, foram desativados quaisquer outros programas que utilizassem fluxo de rede.

A ferramenta utilizada para medir a velocidade da troca de bits entre os dispositivos foi a IPERF (BLUM, 2003), composta de um aplicativo servidor e um aplicativo cliente, que após configurados os parâmetros básicos, como, a quantidade de dados que se deseja passar de um dispositivo a outro, a porta de comunicação e o tipo de conexão (TCP ou UDP), o aplicativo cliente começa a enviar os bits pela rede para o aplicativo servidor e ambos registram a velocidade destes fluxos.

O aplicativo cliente do IPERF foi instalado e configurado no host A e o aplicativo servidor do IPERF no host B. A porta definida para a comunicação foi a de número 5001, o protocolo de transporte escolhido foi o TCP com um número de dez repetições, uma a cada segundo. Iniciada a transferência dos dados, o

aplicativo registrou a velocidade da transmissão dos pacotes que atravessaram o switch e o resultado está ilustrado na Figura 1.

Figura 1 – Velocidade de transmissão na estrutura tradicional.



Fonte: Elaboração do autor.

Percebe-se que a transmissão dos dados se deu de maneira uniforme, em torno de 35 megabits por segundo e sem falhas.

4.2. Desempenho Utilizando uma Rede OpenFlow e o Módulo L3-ARPSec

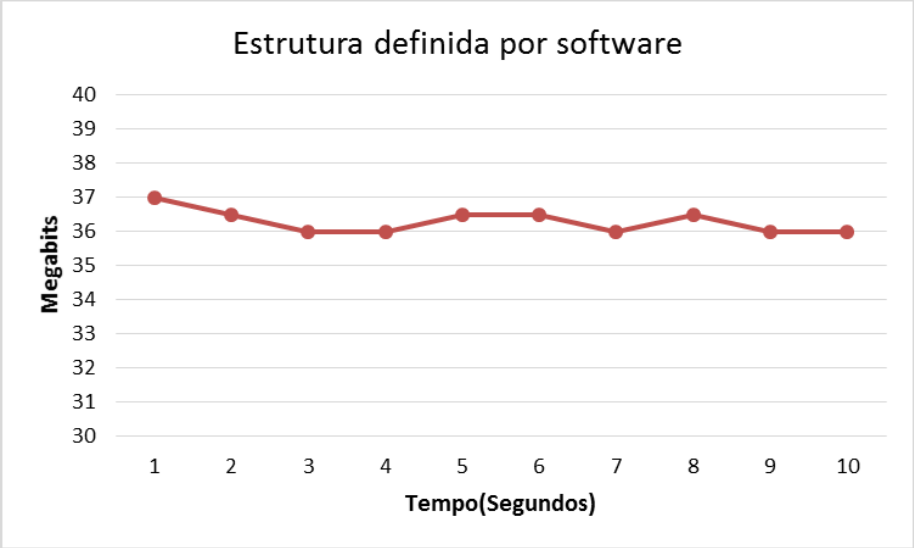
A estrutura para este segundo teste foi configurada de forma similar a utilizada no teste anterior, com exceção de que o *switch* TP-Link teve seu respectivo sistema operacional alterado para suportar o protocolo *OpenFlow* e também foi adicionado a rede, o controlador com o módulo L3-ARPSec implementado. Este controlador foi conectado ao switch através de uma porta específica e separada das demais conexões, certificando assim a impossibilidade de ataque diretamente ao mesmo. Uma observação importante é que nesta estrutura, o repasse dos pacotes é gerenciado pelo controlador.

As configurações de porta, endereço IP, protocolo de transporte e demais parâmetros foram mantidos iguais ao teste anterior. A transferência dos pacotes foi registrada e está ilustrada na Figura 2.

Nota-se que a velocidade de transmissão dos pacotes também se manteve uniforme em um número de 36,5 megabits por segundo.

Uma informação relevante é que cada switch possui sua própria capacidade de repasse de pacotes, sendo definida pelo poder de processamento do hardware e pela eficiência do software configurado. Como o objetivo foi comparar o desempenho no repasse, nos dois testes realizados o mesmo hardware foi utilizado, ficando a diferença apenas em relação aos softwares implantados.

Figura 2 – Velocidade de transmissão na estrutura SDN.



Fonte: Elaboração do autor.

4.3. Desempenho e Funcionalidade com vários usuários

Neste último teste, o objetivo foi ativar e testar o módulo proposto neste trabalho em um laboratório de informática com um número mínimo de vinte usuários simultâneos. Como mostrado na Figura 3, vinte computadores foram conectados a um switch tradicional e configurados para acessar a internet exclusivamente através do servidor de acesso. Este servidor de acesso à internet foi conectado a um switch *OpenFlow*, que por sua vez estava conectado a outros dois computadores e também ao switch tradicional.

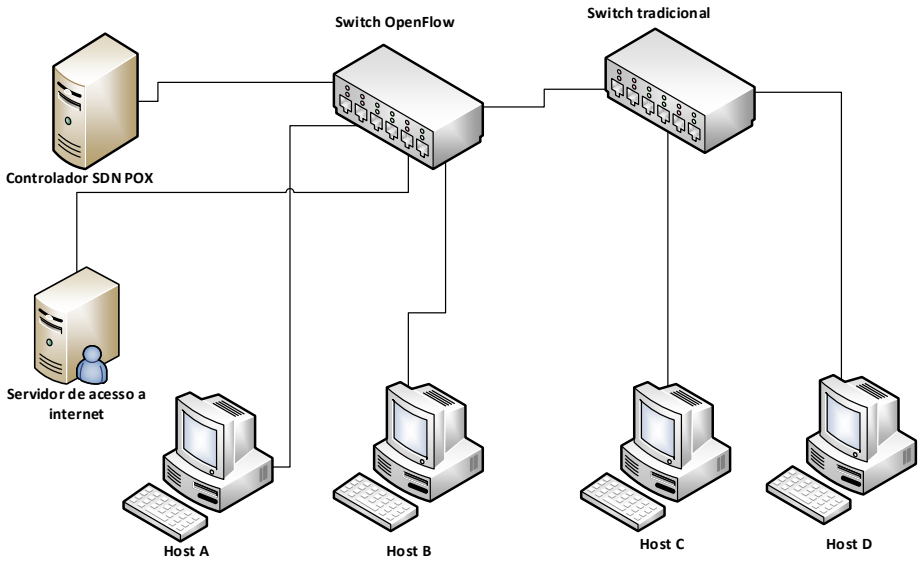
O controlador de rede implementado com o modulo L3-ARPSec foi conectado ao mesmo switch *OpenFlow* a fim de gerenciar o tráfego na rede e bloquear possíveis tentativas de ataques ao protocolo ARP.

Um fato relevante é que nenhum dos usuários foi avisado do teste, desta forma eles não seriam influenciados psicologicamente na coleta dos resultados.

O teste foi realizado pelo período de uma hora ininterrupta e durante todo o tempo a rede foi monitorada. Ao final, os usuários foram questionados sobre a

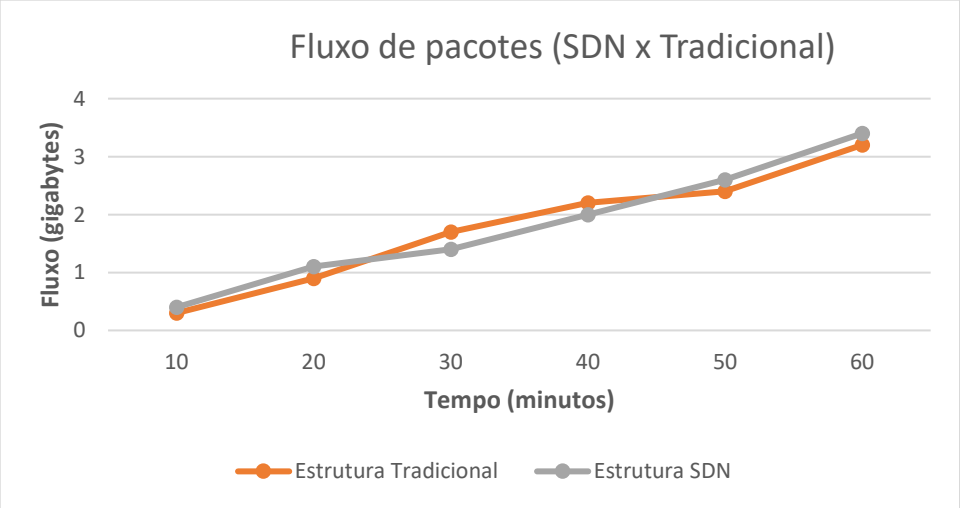
qualidade na utilização do acesso à internet e também sobre possíveis problemas de conexão de rede ocorridos no período. Não foi registrada nenhuma queixa e todos relataram que o acesso ao conteúdo da internet ocorreu de forma tranquila e comum, como em dias anteriores.

Figura 3 – Estrutura tradicional e SDN.



Fonte: Elaboração do autor.

Figura 4 – Fluxo de pacotes registrado no teste.



Fonte: Elaboração do autor.

O monitoramento da rede também revelou que nenhuma alteração significativa de fluxo ou perda de pacotes ocorreu no período do teste. É possível visualizar na Figura 4 a comparação do fluxo de pacotes no dia do teste e também o de um dia anterior. O monitoramento do dia anterior, foi realizado pelo mesmo período de uma hora, no mesmo laboratório e pela mesma turma de usuários. A única diferença está no fato de que a rede estava estruturada na forma tradicional, não havia o controlador e nem o switch *OpenFlow*.

Nota-se que o fluxo de informações durante os sessenta minutos com a rede de estrutura tradicional e a rede definida por software, ficaram em torno de três gigabits.

5. Considerações finais e sugestões para trabalhos futuros

Os testes realizados para validação do desempenho do módulo L3-ARPSec demonstraram não haver diferenças significativas no uso da rede definida por software se comparada a rede de estrutura tradicional. Este fato foi importante para atestar que mesmo possuindo o mecanismo de segurança já mencionado, o módulo proposto não perdeu desempenho e executou todas as funcionalidades básicas de repasse e roteamento dos pacotes na rede.

Uma observação relevante está na possibilidade de um switch *OpenFlow* se conectar a um switch tradicional, formando uma rede hibrida. Entende-se assim, que as redes tradicionais podem ser migradas para o novo modelo de redes definidas por software, de maneira gradativa e sem significativo impacto estrutural.

Nas seções iniciais deste trabalho foram brevemente apresentadas as características e funcionalidades das redes definidas por software. Também foi apresentado o protocolo ARP, explicado seu funcionamento e importância em uma rede local de computadores e ainda foram demonstrados os ataques de segurança ao protocolo ARP, principalmente o ataque MITM que pode comprometer tanto redes locais tradicionais quanto redes locais definidas por software.

A possibilidade de programação no controlador SDN cria oportunidade para o desenvolvimento de técnicas e funcionalidades de gerenciamento dessas redes. Na seção 3 foi apresentado o módulo L3-ARPSec, que controla a troca de mensagens ARP na rede e mitiga o ataque MITM. Posteriormente na seção 4 foram feitos testes de desempenho deste módulo e as discussões a respeito foram detalhadas.

Por fim, conclui-se que o módulo proposto no trabalho de Oliveira (2015) atingiu seus objetivos e representa uma linha de pesquisa vasta a ser explorada e melhorada ainda mais. Trabalhos futuros podem testar este módulo em redes com um número maior de elementos, executar outros testes de desempenho e reescreverem este módulo em outras linguagens de programação que suportem

outros controladores SDN. Outros cenários também podem ser criados a fim de testar o módulo ainda mais e até mesmo ajustes no código-fonte do módulo podem ser efetuados.

Como todo software, o modulo L3-ARPSec pode evoluir e ser aperfeiçoado sempre que se desejar. Novas ideias e funcionalidades poderão ser agregadas ao projeto e contribuir para criar redes com melhor desempenho e mais seguras.

Referências

BLUM, R. Network Performance Toolkit: Using Open Source Testing Tools. Wiley, 2003.

ELLIOTT, C.; FALK, A. An update on the geni project. SIGCOMM Comput. Commun. Rev., 39(3):28–34, 2009.

GUEDES, D; VIEIRA, L. F. M; VIEIRA, M. M; RODRIGUES, H; NUNES, R. V. Redes Definidas por Software: uma abordagem sistêmica para o desenvolvimento de pesquisas em Redes de Computadores. Minicursos - Livro Texto do XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. Porto Alegre: SBC, v. p. 161-212, 2012.

KUROSE, J. F.; ROSS, K. W. Redes de computadores e a Internet: uma abordagem top down. Pearson, 2010.

OLIVEIRA, R. L. S.; SCHWEITZER, C. M.; SHINODA, A. A.; PRETE, L. R. Using Mininet for emulation and prototyping Software-Defined Networks. In 2014 IEEE Colombian Conference on Communications and Computing (COLCOM), pages 1-6, Bogotá, Colombia, 2014.

OLIVEIRA, R. L. S.; SHINODA, A. A.; SCHWEITZER, C. M.; IOPE. R. L.; PRETE, L. R. L3-ARPSec – A Secure Openflow Network Controller Module to control and protect the Address Resolution Protocol. XXXIII Simpósio Brasileiro De Telecomunicações – (SBrT2015), pages 158-162, Juiz de Fora - MG, Brasil, 2015.

PETERSON, L.; ROSCOE, T. The design principles of planetlab. SIGOPS Oper. Syst. Rev., 40(1):11–16, 2006.

PHILIP, R. “Securing Wireless Networks from ARP Cache Poisoning”, 2007. Master's Projects. Paper 131. Disponível em: <http:// http://scholarworks.sjsu.edu/etd_projects/131>. Acesso em: 26 ago. 2014.

TENNENHOUSE, D. L.; WETHERALL, D. J. Towards an active network architecture. SIGCOMM Comput. Commun. Rev., 37(5):81–94, 2007.

TURNER, J. S. A proposed architecture for the geni backbone platform. In Proceedings of the 2006 ACM/IEEE symposium on Architecture for networking and communications systems, ANCS ’06, pages 1–10, New York, NY, USA. ACM, 2006.