DmzVisor: um *Firewall* para a Segurança de Zona Desmilitarizada Corporativa em Redes Definidas por Software

Valson da S. Pereira[1], **Ygor Amaral B. L. de Sena**[1]

[1]Universidade Federal Rural de Pernambuco (UFRPE)
Caixa Postal 063 – 56.909-535 – Serra Talhada – PE – Brasil

valson.pereira@gmail.com, ygor.amaral@ufrpe.br

Abstract. *For the security of organizations that need to offer services to the external public, it is recommended to use a network protected by perimeter firewalls, known as the DeMilitarized Zone (DMZ), between the internal and external networks, in which it is located devices that hosting services accessed by the external public. However, the use of the DMZ may have a considerable financial cost for the organization. Therefore, through the Software Defined Networks (SDN) paradigm, it is possible to develop flexible and customized solutions that cover the whole network at a low-cost. Thus, this paper aims to propose the DmzVisor, which consists of a low-cost corporate SDN firewall developed with free open source software. The application in addition to filtering packets and isolating traffic between the local network and the corporate DMZ, also provides a web system for its management.*

Resumo. *Para a segurança das organizações que necessitam oferecer serviços ao público externo, recomenda-se a utilização de uma rede protegida por firewalls de perímetro, denominada de DeMilitarized Zone (DMZ), entre as redes interna e externa, na qual estão os dispositivos que hospedam serviços acessados pelo público exterior. Entretanto, o uso da DMZ pode ter um custo financeiro considerável para a organização. Diante disso, através do paradigma de Redes Definidas por Software (SDN) é possível desenvolver soluções flexíveis e personalizadas que abrangem toda a rede a um baixo custo. Dessa maneira, este artigo tem o objetivo de propor o DmzVisor, que consiste em um firewall SDN corporativo de baixo custo desenvolvido com software gratuito de código aberto. A aplicação além de filtrar pacotes e isolar o tráfego entre a rede local e a DMZ corporativa, também fornece um sistema web para seu gerenciamento.*

1. Introdução

O avanço na utilização da rede mundial de computadores nas últimas décadas fez a internet se tornar umas das principais ferramentas de comunicação ao redor do mundo. Milhões de usuários dos mais variados perfis estão conectados na web. Diante dessa realidade a utilização da internet passou a ser parte estratégica nos mais diversos ramos de negócio, portanto, tornando-se cada vez mais comum as empresas fornecerem serviços online para seus clientes.

Em virtude disso, para a segurança dessas organizações que necessitam oferecer serviços ao público externo, recomenda-se a utilização de uma rede de perímetro protegida por *firewalls*, denominada de *DeMilitarized Zone* (DMZ), entre as redes interna

e externa, na qual estão os dispositivos que hospedam serviços acessados pelo público exterior, de modo que, se algum dispositivo da DMZ esteja com a segurança comprometida, a rede interna que possui os dados mais sensíveis da organização permanece segura [Nakamura and de Geus 2007]. No entanto, de acordo com [Jiang et al. 2017], apesar de prover mais segurança em sistemas críticos, a utilização de mais de um *firewall*, como geralmente faz uso quando se trata de uma DMZ, pode tornar financeiramente mais caro e complexo o gerenciamento da rede.

Diante disso, através do paradigma de Redes Definidas por Software (SDN) é possível desenvolver soluções flexíveis e personalizadas que abrangem toda a rede, por meio da separação do plano de dados e plano de controle dos comutadores, centralizando a inteligência da rede em um controlador externo implementado via software [McKeown et al. 2008], que pode ser uma solução gratuita *open source*. Em decorrência desse fato, o *OpenFlow* se apresenta como um protocolo aberto amplamente utilizado que, possibilita a comunicação entre o plano de dados e o controlador externo em uma rede SDN [Braun and Menth 2014]. Uma das vantagens da utilização desse paradigma no desenvolvimento de uma aplicação de segurança de rede como o *firewall* é que, segundo [Guedes et al. 2012], "a visão global da rede oferecida pelo controlador SDN permite que regras de acesso sejam desenvolvidas com base em informações abrangentes e não apenas o que seria possível com o uso de um *firewall* em um enlace específico da rede".

Dessa maneira, este artigo tem como objetivo propor o *DmzVisor*, que consiste em um *firewall* SDN corporativo de baixo custo implementado com software gratuito de código aberto. A aplicação além de filtrar pacotes e isolar o tráfego entre a rede local e a DMZ corporativa em redes definidas por software, também fornece um sistema web com interface de fácil gerenciamento.

2. Trabalhos Relacionados

O *firewall* exposto por [Badotra and Singh 2018], consiste numa aplicação SDN desenvolvida no controlador Ryu, que provê segurança para as camadas de transporte e aplicação, por meio da filtragem de pacotes. O sistema é capaz de bloquear o acesso de pacotes TCP, ICMP e web nestas camadas, no qual, a regra pode ser aplicada em um único *host* ou na rede inteira. A avaliação dessa proposta consistiu no bloqueio de determinados sites da web e do protocolo ICMP. Ao contrário desta aplicação, o *software* apresentado neste trabalho fornece segurança às camadas de rede e transporte. Além do mais, o *DmzVisor* também permite regras para máquinas específicas (servidores da DMZ) ou para toda a rede (LAN).

Em [Kumar and Srinath 2016], é proposto um *firewall* SDN dedicado para redes *full mesh* utilizando o controlador POX e *OpenFlow* 1.0. Esse tipo de rede pode ser complexo para escolher um ponto central para posicionar um *firewall* tradicional, tendo em vista que, todos os dispositivos do núcleo da rede estão conectados ponto a ponto. Entretanto os desenvolvedores dessa solução se beneficiaram com a visão geral da rede proporcionada pelo paradigma SDN, para implementar o software.

A proposta de [Zerkane et al. 2016] oferece um *firewall OpenFlow* reativo de conexões de estado. Esse *firewall* busca tratar conexões TCP, por meio de monitoramento do seu estado, para mitigar ataques do tipo DoS de inundação de TCP SYN. A aplicação foi desenvolvida fazendo uso do controlador Ryu e fornecendo ao usuário uma interface

gráfica. O trabalho apenas avaliou o estado de conexão de um servidor HTTP, por meio de uma topologia composta por 3 clientes, no qual, cada cliente realizava diversas consultas simultâneas ao servidor web. No entanto, diferentemente deste *firewall* de conexão de estado, o *DmzVisor* tem a responsabilidade de aplicar regras de segurança em duas redes distintas e, além disso, mantê-las separadas uma da outra.

A proposta deste trabalho é oferecer certa comodidade ao administrador de rede através de um *firewall* amigável com regras de fácil gerenciamento para as camadas de rede e transporte, e além disso, prover também toda a segurança que segrega as redes LAN e DMZ pelo tipo dos *hosts* conectados, mantendo o isolamento entre ambas.

3. DmzVisor: SDN *Firewall*

A proposta foi implementada utilizando o ***Ryu***[1], um controlador SDN simples de código aberto que traz consigo toda a praticidade e robustez da linguagem Python, juntamente com o ***OpenFlow* 1.3**, que é uma versão estável do protocolo que suporta os requisitos necessários para a implementação desta aplicação. A arquitetura do *DmzVisor* juntamente com os módulos implementados no controlador Ryu podem ser observados na Figura 1.

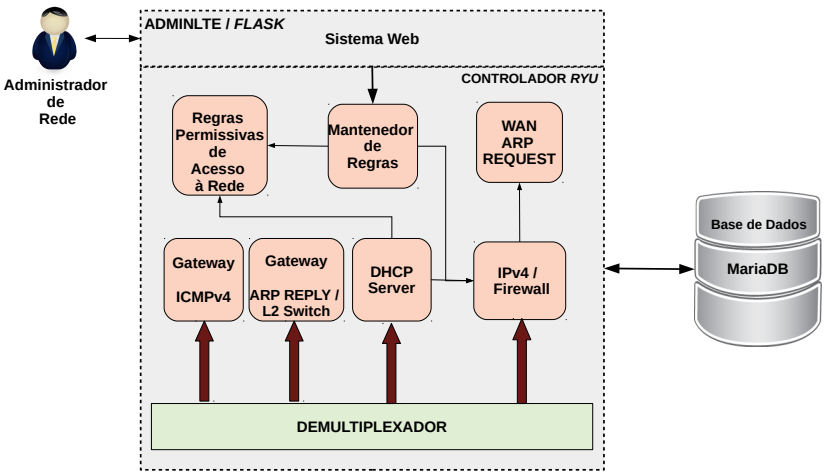

Figura 1. Arquitetura do *DmzVisor*.

O **demultiplexador** tem o importante papel de receber os fluxos oriundos do *switch* para o controlador, análisando o protocolo do fluxo e encaminhando para o módulo correto fazer o devido tratamento. Já o módulo ***Gateway* ICMPv4** apenas responde mensagens ICMP ECHO REQUEST (ping) destinadas aos *gateways* da rede. Sendo assim, quando um fluxo ICMPv4 chega ao Demultiplexador ele analisa se o destino é o *gateway* da rede e encaminha o fluxo para este módulo, no qual o próprio controlador responderá. Caso o destino seja outro *host* em uma outra rede, o fluxo é enviado para o módulo IPv4 / *Firewall*.

O módulo **DHCP Server** é o responsável por fazer o isolamento e distribuir os devidos endereços IPs para os dispositivos na LAN e DMZ, de acordo com o perfil de cada *host* conectado na rede, se for um *host* da LAN, o DHCP entregará um IP apropriado

[1]*Ryu*: `https://github.com/osrg/ryu`

as configurações da LAN, da mesma forma, se for um *host* da DMZ, será entregue um IP apropriado a essa rede. Quem define o perfil de cada *host* conectado é o administrador de rede.

A princípio para que todo *host* ingresse na rede via DHCP de uma forma segura, foi necessário alterar o comportamento mais utilizado em uma rede *OpenFlow*. O fluxo *table-miss flow entry*, muito comum nas soluções SDN com o propósito de encaminhar para o controlador todos os fluxos de entrada até então desconhecidos na tabela de fluxos do plano de dados, foi substituído por um fluxo com a maior prioridade, que redireciona para o controlador todos os pacotes DHCP. Com a retirada do *table-miss flow entry*, o plano de dados passa a descartar automaticamente todos os pacotes de fluxos que não estão instalados na tabela do comutador *OpenFlow*. Desse modo, tem-se a possibilidade de oferecer mais segurança, pois, o *switch* somente tomará decisões baseado em fluxos confiáveis previamente instalados pelo controlador, sendo assim, rejeitando de maneira nativa os fluxos desconhecidos na rede. Consequentemente, enquanto um *host* ingressante não se identificar através do DHCP, nenhum pacote vindo desse *host* será comutado ou processado, dificultando também as tentativas maliciosas de entrar na rede através de configurações manuais.

Durante a troca de mensagens entre o *host* e o controlador, o módulo DHCP armazena no SGDB as informações do dispositivo contidas nas mensagens DHCP, como: o endereço MAC; o endereço IP que o módulo entregou ao dispositivo e também o número da porta na qual ele está conectado no *switch*. A partir do momento que o *host* obtém endereço IP por meio do módulo DHCP, as próximas conexões DHCP ou reinício desse dispositivo na rede, assim como também todas as regras de segurança, inclusive as regras de *firewall*, serão baseadas nos dados que foram mapeados para o banco. Portanto, todas as regras do *DmzVisor* são compostas por fluxos de tríplices atributos (porta do *switch*, MAC e IP) de origem e de destino, com exceção do fluxo DHCP, pois nessa situação o IP do dispositivo não é informado no cabeçalho de origem. Diante disso, uma simples mudança de IP que pode ser até um endereço da mesma rede ou das outras adjacentes, feita manualmente pelo usuário da máquina ou por algum ataque *hacker*, terá como consequência o isolamento total deste dispositivo da rede, pois apenas o administrador tem a possibilidade de mudar as configurações de rede via interface gráfica.

As **regras permissivas de acesso à rede** são responsáveis por instalar regras que possibilitam a inspeção pelo controlador, dos pacotes sem rota definida dos novos *hosts*, logo após a obtenção de endereço IP via DHCP, a partir disso, o controlador poderá criar as devidas rotas. Sem essas regras permissivas os *hosts* ingressantes permaneceriam isolados. Para não ocorrer essa situação é obrigatório que toda máquina ingresse na rede por intermédio de um endereço atribuído pelo módulo DHCP.

O módulo ***Gateway* ARP REPLY / L2 *Switch*** atende às requisições ARP destinadas aos *gateways* e faz a comutação de nível 2, ou seja, a comunicação entre os *hosts* da mesma rede. Ao receber o pacote ARP REQUEST do demultiplexador, o *gateway* ARP REPLY / L2 *Switch* examina se o pacote é destinado ao *gateway* ou a outro *host* da mesma rede. Caso seja para o *gateway*, o próprio controlador, constrói um pacote ARP REPLY e responde ao *host*. Já no segundo caso, o módulo analisa se o dispositivo de destino realmente está mapeado na base de dados. Comprovando que o *host* de destino está devidamente mapeado, o módulo instala regras de comutação de pacotes de nível 2,

ou seja, regras que permitem máquinas confiáveis da mesma rede se comunicarem sem restrições.

O módulo **IPv4 / *Firewall*** é o principal módulo do controlador, com a finalidade de instalar as regras de roteamento IPv4 nas redes LAN e DMZ com base nas regras de *firewall* definidas pelo administrador. As regras impostas para a rede LAN são de maneira generalizada, ou seja, todos os *hosts* da LAN possuem as mesmas regras de *firewall*. Em contrapartida, para a rede DMZ, o usuário pode definir regras de filtragem específicas para cada uma das máquinas, isso porque, cada dispositivo da DMZ pode oferecer diferentes serviços e em razão dessa situação, a aplicação oferece essa possibilidade da instalação de regras de acordo com o serviço disponibilizado pelo *host* da zona desmilitarizada.

As regras de *firewall* do *DmzVisor* que consistem na filtragem de pacotes da camada de rede e transporte, suporta os seguintes protocolos: **IPv4**, **ICMP (ECHO REQUEST e ECHO REPLY)**, **TCP** e **UDP**. A proposta também fornece um formato de regras em alto nível, no qual, o usuário não necessita possuir conhecimentos específicos em *OpenFlow* para gerenciar o *firewall*. O módulo IPv4 / *Firewall* se encarrega de traduzir essas regras de alto nível em fluxos da tabela do plano de dados. As regras possuem o seguinte formato: **direção** – se as conexões para aplicação das regras são de entrada ou saída; **prioridade** – ordem de aplicabilidade da regra; **protocolo** – indica em qual protocolo a regra deve ser aplicada; **porta TCP/UDP** – número da porta de conexão TCP ou UDP. Se não for definido o número da porta a regra será aplicada em toda a extensão do protocolo. Caso a regra seja para outros protocolos (ICMP ou IPv4) o campo da porta deve ser ignorado; **Ação** – A atuação que o plano de dados deve ter com relação à regra, ou seja, se encaminha o pacote ou descarta.

O módulo **Mantenedor de regras** é uma *thread* dedicada que faz a comunicação com a aplicação web. O mantenedor de regras permanece à escuta de requisições do sistema web recebendo comandos tais como: adicionar, editar, excluir regras e as aplica com o auxílio de outros módulos. Por fim o módulo **WAN ARP REQUEST** tem como função enviar exclusivamente mensagem ARP REQUEST para obter endereços MAC de elementos da rede que estejam conectados através da porta WAN.

3.1. Mudança de um *Host* para outra Rede

A mudança de rede é uma ação privativa e explícita do administrador de rede que, consiste em mudar um determinado *host* que está na LAN para a DMZ, ou vice e versa. Esse gerenciamento ocorre via uma interface web amigável. O usuário informa que deseja mudar o dispositivo da rede. Após isso, é necessário que a conexão do *host* seja reinicializada, enquanto isso não acontece, o *host* ficará sem conectividade, evitando o risco na segurança de continuar tendo acesso a sua antiga rede. Após o reinício da conexão o módulo DHCP fornece um endereço IP para nova rede e atualiza todas as regras no SGBD e no plano de dados de acordo com o novo endereço de rede atribuído.

3.2. Sistema Web para Gerência do *Firewall*

O sistema web foi desenvolvido utilizando o template administrativo **AdminLTE**[2] no *front-end* e o microframework Python **Flask**[3] no *back-end*, ambos *open source*. Por intermédio da interface gráfica o usuário pode realizar operações básicas com as regras,

[2]AdminLTE: `https://adminlte.io`

[3]Flask: `https://github.com/pallets/flask`

como por exemplo, adicionar, editar ou excluir uma regra. A Figura 2 apresenta regras padrão que já oferecem um bom nível de segurança para a rede LAN, que consiste em: bloquear conexões de entrada TCP (somente pacotes com a flag SYN, permitindo a passagem das demais flags TCP) e UDP (respectivamente regras 01 e 02), permitir o envio de requisições e recebimento de respostas de pacotes ICMP (regras 03 e 04) e encaminhar IPv4 para fora da rede (regra 05).

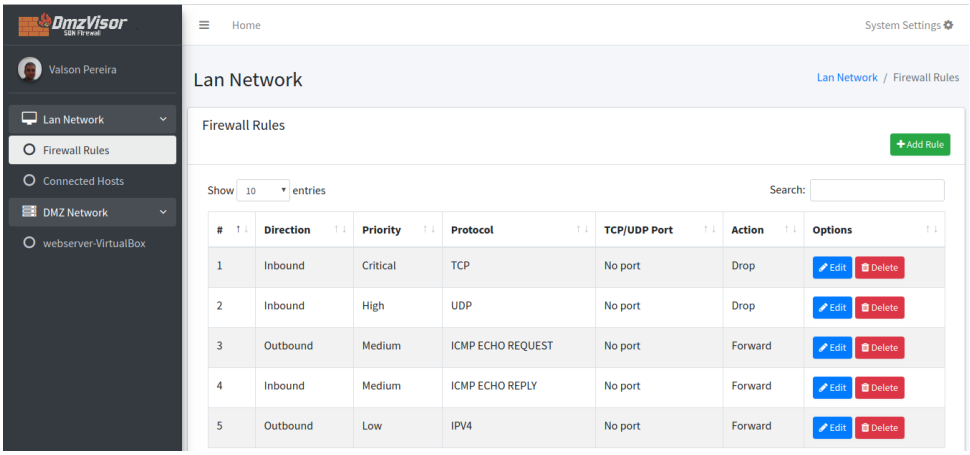

Figura 2. Tela principal do *DmzVisor* com regras padrão de *firewall* para rede LAN.

O sistema ainda apresenta na aba ***Connected Hosts*** as configurações de todas as máquinas conectadas na rede LAN que foram mapeadas pelo módulo DHCP do controlador. Nesta aba o usuário pode mover uma máquina para a DMZ, editar a porta do *switch* no qual o dispositivo está conectado, ou ainda excluir o *host* do comutador, liberando a porta para receber uma nova máquina. Já em ***DMZ Network*** estão os servidores da rede DMZ identificados pelo seu *hostname* mapeado pelo DHCP, onde são apresentadas as regras e configurações de rede de cada servidor, a exemplo da *LAN Network*.

4. Avaliação da Proposta e Resultados

A avaliação da proposta foi composta por 02 cenários utilizando 6 máquinas virtualizadas pelo VirtualBox, conectadas em um único *switch OpenFlow* virtual, o *Open vSwitch*. Além da avaliação do isolamento de redes em ambos cenários, no primeiro cenário foram avaliadas as principais regras de *firewall* padrão da rede LAN (regras 01 e 05 descritas na Figura 2). Para o segundo cenário foi avaliado a segurança de um servidor web na DMZ que apenas possui regras para permitir o acesso aos seus serviços na porta TCP/80. O DHCP distribui IP's da sub-rede 10.0.0.0/24 para dispositivos da LAN e IP's da sub-rede 192.168.0.0/24 para dispositivos da DMZ. Também foi utilizada uma máquina virtual numa terceira rede, externa a LAN e DMZ, simbolizando a internet nos experimentos (IP: 172.16.0.2/16).

Para a avaliação do isolamento das redes foram alterados de um *host* da LAN, o endereço MAC com auxílio da ferramenta GNU Mac Changer, como também o endereço IP foi modificado manualmente para um endereço IP válido da DMZ. Porém como demonstra a Figura 3, ainda conectados no mesmo *switch*, não há nenhum tipo de comunicação entre o *host* da DMZ (à esquerda da imagem) e o *host* da rede LAN com o MAC e IP alterado (à direita da imagem), Isso se deve ao fato dele ter violado as regras de

segurança da rede, pois no plano de dados não há rotas para ele com esse endereço MAC e endereço IP do tipo DMZ. Vale ressaltar que o mesmo experimento foi realizado na rede DMZ com intuito de tentar passar uma máquina dessa rede para LAN manualmente, porém foram obtidos os mesmos resultados do primeiro cenário.

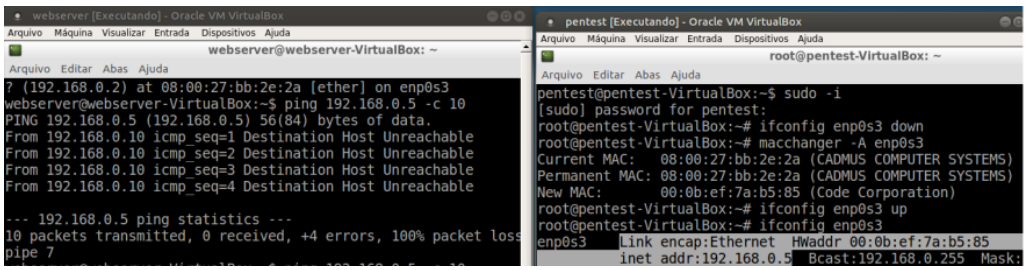

Figura 3. Conectividade sem sucesso entre servidor web da DMZ e *host* LAN malicioso.

A Figura 4 demonstra o escaneamento de portas feito pela a ferramenta Hping, a partir de um *host* externo em direção a um *host* na LAN. Apesar de ter várias portas abertas (esquerda da imagem), o escaneamento não encontra nenhuma porta aberta aceitando a inicialização de conexões TCP, o que prova a eficácia da regra 01 (TCP) da rede LAN. Pode-se observar também na Figura 5 que o *host* na LAN pode acessar serviços do *host* externo por meio da regra 05 (IPv4), ou seja, nada influencia no acesso da rede LAN à máquina externa o fato do *host* externo não conseguir visualizar as portas abertas de dispositivos LAN.

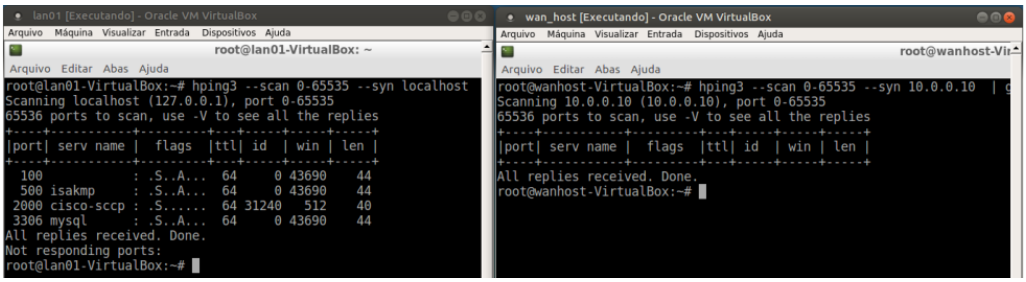

Figura 4. *Host* LAN com portas TCP abertas e escaneamento feito pelo wanhost (internet).

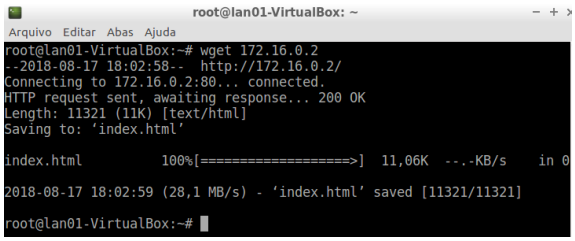

Figura 5. *Host* LAN acessando serviço da rede externa.

Semelhantemente à rede LAN foi realizado o escaneamento de portas TCP no servidor web a partir do *host* externo que simboliza a internet, o resultado pode ser visto na Figura 6. O *host* externo a DMZ conseguiu visualizar apenas a porta permitida explicitamente pela regra do *firewall*, ou seja, a porta TCP/80 a qual consequentemente lhe permite o acesso ao serviço.

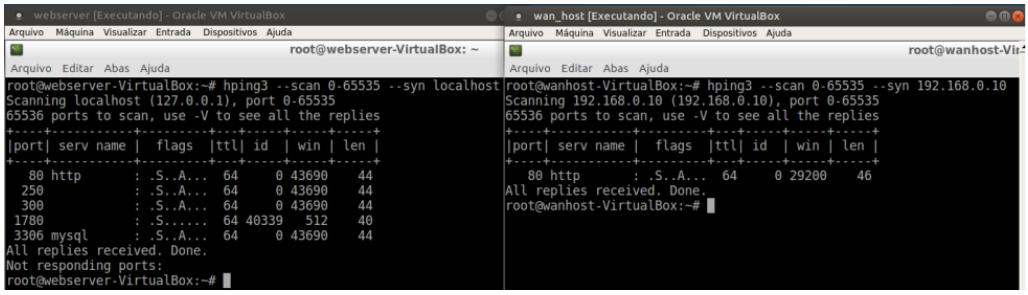

Figura 6. Portas abertas no servidor web e o escaneamento realizado pelo *host* externo.

5. Considerações Finais

O *DmzVisor*, *firewall* proposto neste trabalho, apresentou-se como uma solução de baixo custo desenvolvida com ferramentas gratuitas de código aberto, capaz de isolar as redes da organização e de mantê-las seguras por meio da filtragem de pacotes das camadas de transporte e de rede. Como trabalhos futuros, pretende-se realizar experimentos reais, avaliação de desempenho e oferecer suporte do *firewall* à IPv6.

Referências

Badotra, S. and Singh, J. (2018). Creating firewall in transport layer and application layer using software defined networking. In Saini, H. S., Sayal, R., Govardhan, A., and Buyya, R., editors, *Innovations in Computer Science and Engineering*, pages 95–103, Singapore. Springer Singapore.

Braun, W. and Menth, M. (2014). Software-defined networking using openflow: Protocols, applications and architectural design choices. *Future Internet*, 6(2):302–336.

Guedes, D., Vieira, L., Vieira, M., Rodrigues, H., and Nunes, R. V. (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.

Jiang, N., Lin, H., Yin, Z., and Xi, C. (2017). Research of paired industrial firewalls in defense-in-depth architecture of integrated manufacturing or production system. In *Information and Automation (ICIA), 2017 IEEE International Conference on*. IEEE.

Kumar, A. and Srinath, N. (2016). Implementing a firewall functionality for mesh networks using sdn controller. In *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, pages 168–173, Bangalore. IEEE, IEEE.

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.

Nakamura, E. T. and de Geus, P. L. (2007). *Segurança de redes em ambientes cooperativos*. Novatec Editora, São Paulo.

Zerkane, S., Espes, D., Le Parc, P., and Cuppens, F. (2016). Software defined networking reactive stateful firewall. In Hoepman, J.-H. and Katzenbeisser, S., editors, *ICT Systems Security and Privacy Protection*, pages 119–132, Cham. Springer International Publishing.