UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

RODRIGO JANTSCH

Mitigação de ataques DDoS em redes baseadas em infraestruturas SDN/NFV

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação

Orientador: Prof. Dr. Alberto Egon Shaeffer-Filho

Porto Alegre
2016

AGRADECIMENTOS

Primeiramente, gostaria de agradecer ao Prof. Dr. Alberto Egon Schaeffer-Filho por todo auxílio, orientação e incentivo ao longo deste projeto. É difícil descrever tudo que eu aprendi durante esse período. Muito Obrigado.

Agradeço ao colega Pedro Faustini pela disponibilidade e pela ajuda com a plataforma utilizada neste trabalho.

Meus profundos agradecimentos a minha família e a minha namorada, Schylar, pelo amor e suporte de sempre. E a todos que diretamente ou indiretamente, fizeram parte da minha formação, o meu muito obrigado.

RESUMO

Ataques distribuídos de negação de serviço objetivam a disjunção de um serviço provido por um hospedeiro. Redes baseadas em infraestruturas SDN/NFV permitem o monitoramento e o gerenciamento da rede de uma forma logicamente centralizada, assim como a migração e a instanciação de funções sob demanda, o que possibilita maior flexibilidade e elasticidade no planejamento de novos mecanismos de defesa. Portanto, redes baseadas em infraestruturas SDN/NFV podem auxiliar na elaboração de estratégias de mitigação contra ataques DDoS. Neste trabalho, apresentamos uma revisão da literatura com o objetivo de enumerar as diferentes variações de ataques DDoS. Propomos uma taxonomia para mecanismos de mitigação contra ataques DDoS e apresentamos estratégias de mitigação baseadas em técnicas de *Rate-Limiting*, *Filtering* e *Reconfiguração*. Foram implementadas as estratégias de *Throttling*, que consiste na limitação de todo tráfego destinado a vítima, e *Firewalls Cooperativos*, que utiliza um sistema de detecção logicamente centralizado que auxilia na geração de assinaturas instaladas em filtros. Por meio dos experimentos realizados, verificamos que o protótipo apresentado é capaz de mitigar ataques DDoS de inundação ICMP com conjunto variável de agentes. Também concluímos que o tempo de resposta da estratégia de mitigação está diretamente associado à eficiência da estratégia de detecção utilizada.

Palavras-chave: Ataques DDoS. Estratégias de mitigação. Segurança. Redes baseadas em SDN/NFV.

DDoS Mitigation in SDN/NFV

ABSTRACT

Distributed Denial of Service (DDoS) attacks aim the disjunction of a service provided by a host. Networks based on SDN/NFV infrastructures allow monitoring and management of the network in a logically centralized way, as well as the migration and instantiation of network functions on demand, which enables flexibility and elasticity in planning new defense mechanisms. Therefore, networks based on SDN and NFV can assist the development of mitigation strategies agaisnt DDoS attacks. In this document, we present a bibliographic review with the purpose of numbering different DDoS attacks variations. We present a taxonomy to mitigation mechanisms against DDoS attacks and we depict mitigation strategies based on Rate-Limiting, Filtering, and Reconfiguration techniques. Two strategies were implemented: Throttling strategy, which consists of limiting all traffic destined to the victim, and Cooperative Firewalls strategy, which uses a centralized detection system that generates signatures which will be installed in filters. Through the performed experiments, we observed that the prototype is capable of mitigating ICMP flooding DDoS attacks with variable set of agents. We also concluded that the response time of the mitigation strategy is directly associated to the efficiency of the detection strategy used.

Keywords: DDoS Attacks, Mitigation Strategies, Security, Networks based on SDN/NFV.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

AFW	Assistant Firewall
CAPEX	Capital Expenditure
CDN	Content Delivery Network
DDoS	Distributed Denial-of-Service
DFW	Defender Firewall
IRC	Internet Relay Chat
ISP	Internet Service Provider
ICMP	Internet Control Message Protocol
LLDP	Link Layer Discovery Protocol
NFV	Network Function Virtualization
OPEX	Operational Expenditure
SDN	Software-Defined Networking
VM	Virtual Machines

SUMÁRIO

1 INTRODUÇÃO

A arquitetura da Internet foi originalmente projetada visando abertura e escalabilidade. Apesar de seu sucesso nesses quesitos, diversos problemas de segurança não foram originalmente previstos (PENG; LECKIE; RAMAMOHANARAO, 2007), como o *Distributed Denial of Service (DDoS)* que é um tipo de ataque que objetiva a disrupção de um serviço prestado por um hospedeiro, fazendo com que usuários não consigam acesso ao serviço oferecido. Com a extrema facilidade de acesso a ferramentas automatizadas que realizam a construção de redes de máquinas comprometidas e as tarefas requeridas pelo atacante, o número de ataques pertencentes a essa categoria tem aumentado significativamente.

Software-Defined Networking (SDN) é uma arquitetura emergente de redes que realiza a separação física do plano de dados do plano de controle, que é diretamente programável (ONF, 2012). *Network Function Virtualization (NFV)* transforma como operadores projetam suas redes ao implementar funções de redes, tais como firewalls, CDN e *message routers*, em softwares que são executados em ambientes virtualizados rodando em hardwares de uso genérico (ETSI, 2012). Com o surgimento de infraestruturas baseadas em SDN/NFV, tornou-se possível monitorar e gerenciar a rede de uma forma logicamente centralizada, migrar funções de rede, obter maior facilidade na modificação de funções de rede e uma gerência mais flexível das funções da rede comparado ao uso de dispositivos físicos dedicados (conhecidos como *middleboxes*). Consequentemente, redes baseadas em infraestruturas SDN e NFV permitem uma refatoração de técnicas de mitigação, melhorando os aspectos de flexibilidade e elasticidade dos mecanismos de defesa (FAYAZ et al., 2015). Esse trabalho investigará abordagens que possibilitem a melhora de tais aspectos.

1.1 Objetivos

O trabalho pretende atuar na abordagem de mecanismos de mitigação, que especificamente explorem características de SDN/NFV. Para a realização dessas tarefas, os seguintes items serão realizados:

- uma revisão da literatura com o objetivo de entender e enumerar os principais tipos de ataques, assim como entender as características fundamentais presentes na

cadeia de um ataque DDoS;

- o desenvolvimento de uma taxonomia referente aos mecanismos de mitigação de ataques DDoS que ilustra as principais técnicas e estratégias atualmente utilizadas;
- a elaboração de um protótipo que permitirá a realização de experimentos e análise de resultados das estratégias propostas;
- a comparação entre um subconjunto de estratégias avaliadas focando nas métricas de tempo de resposta da estratégia de mitigação e disponibilidade do serviço para usuários não maliciosos;

1.2 Contribuições

Para o desenvolvimento dos mecanismos de mitigação propostos nesses trabalho, os seguintes tópicos foram abordados:

- o levantamento bibliográfico dos principais ataques DDoS existentes e de sua cadeia;
- a pesquisa e a elaboração de uma taxonomia de mecanismos de mitigação em ataques DDoS baseada em técnicas de *Rate-Limiting*, *Filtering* e *Reconfiguração*;
- as estratégias de mitigação de *Throttling*, que com base na topologia da rede descoberta pela aplicação na camada superior ao controlador SDN aplica a técnica de Rate-Limiting aos dispositivos de encaminhamento com distância k da vítima, e *Firewalls Cooperativos*, que busca a aplicação de filtros na borda da rede com o auxílio do monitoramento executado por uma aplicação na camada superior do controlador SDN, foram desenvolvidas;
- o desenvolvimento do roteamento nível 3, utilizando o controlador POX, dos pacotes na rede e para o redirecionamento do fluxo identificado como malicioso;
- a análise de desempenho dos sistemas desenvolvidos utilizando as métricas de tempo de resposta, na estratégia de *Throttling*, e disponibilidade do serviço para usuários não maliciosos durante a ocorrência de um ataque;

1.3 Estrutura do Trabalho

No Capítulo 2, a cadeia de um ataque cibernético, a cadeia de um ataque DDoS e os critérios de classificação de ataques DDoS são apresentados. Também é introduzida a infraestrutura de *Software-Defined Networking* (SDN) e detalhes de funcionamento do protocolo OpenFlow. Por fim, as vantagens, as características de arquitetura e o relacionamento de Network Functions Virtualization (NFV) com o SDN são apresentadas. O Capítulo 3 descreve como as estratégias de mitigação baseadas em táticas de *Rate-Limiting*, *Filtering* e *Reconfiguração* podem ser utilizadas em redes com infraestruturas baseadas em SDN/NFV. No Capítulo 4, são implementadas as estratégias propostas para mitigação de ataques DDoS e é realizada a análise dos resultados obtidos. No Capítulo 5, são apresentados os trabalhos relacionados. Finalmente, o Capítulo 6 apresenta as conclusões atingidas através do desenvolvimento do trabalho, com o resumo das contribuições e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, será estabelecida a fundamentação teórica dos tópicos abordados nesse trabalho, com o objetivo de estimular o entendimento dos conceitos e tecnologias que serão utilizadas. A Seção 2.1 introduz os ataques distribuídos de negação de serviço - Distributed Denial-of-Service (DDoS), especificando suas etapas e classificações. Na Seção 2.2, detalhes sobre as infraestruturas de redes utilizadas para o desenvolvimento desse trabalho são abordados.

2.1 Ataques DDoS (Distributed Denial-of-Service)

Os ataques do tipo Denial of Service (DoS) podem ser descritos como uma forma de impossibilitar o fornecimento de serviço prestado por um hospedeiro, portanto esses ataques não possuem como objetivo destruir informação, mas sim, comprometer a disponibilidade dos recursos do sistema. Apesar dos ataques Distributed Denial-of-Service (DDoS) possuírem o mesmo objetivo dos ataques DoS, eles diferem na quantidade de máquinas necessárias para desempenhar o ataque, visto que em ataques DoS somente um atacante participa do ataque e em ataques DDoS múltiplas máquinas são tipicamente utilizadas (DOULIGERIS; MITROKOTSA, 2004). O primeiro ataque DDoS foi relatado em 1996, e desde então foram relatados ataques em grandes instituições tais como a webpage do FBI (2000), ClickBanck (2003), Facebook (2009), Twitter (2009) e Feedly (2014) (RADWARE, 2015). As motivações desses ataques podem ser políticas (um país em guerra pode efetuar ataques contra seu adversário),por prestígio da comunidade Hacker, por motivos pessoais (grande parte de ataques são contra computadores pessoais, provavelmente por vingança) ou por ganho material (danificar recursos do adversário ou chantagear companhias) (MIRKOVIC; REIHER, 2004). Em (IDEAS, 2016), é possível acompanhar ataques identificados como DDoS desde 2013. A evolução rápida desse tipo de ataque desde 1996 e o impacto que eles ocasionam na internet atualmente tornam o estudo dos tópicos relevante.

A Seção 2.1.1 apresenta a cadeia de um ataque cibernético. Na Seção 2.1.2, é descrito o modelo de cadeia de um ataque DDoS. Na Seção 2.1.3, os critérios de classificação definidos na taxonomia de Mirkovic e Reiher (2004) são introduzidos.

2.1.1 Cadeia de um Ataque Cibernético

No decorrer dessa Seção, serão introduzidos os elementos que compõem um ataque e será apresentado o modelo de cadeia um ataque cibernético. O estudo dessa modelagem é necessário para apresentação da modelagem desenvolvida especialmente para ataques DDoS, proposta na Seção 2.1.2.

Um ataque é composto por quatro elementos: o *atacante*, o *handler* ou *mestre*, o *agente* ou *zombie* e a *vítima*. O *atacante* é a máquina que efetivamente coordena o ataque. O *handler* é um hospedeiro infectado por um programa capaz de controlar múltiplos agentes. O *agente* é a máquina que realiza o ataque DoS. Dependendo da aplicação o agente pode recrutar novos agentes e, consequentemente, pode se tornar uma máquina mestre. A *vítima* é a máquina a ser atacada (DOULIGERIS; MITROKOTSA, 2004).

Uma *cadeia de um ataque cibernético* é um processo sistemático para atingir e envolver uma vítima para criar efeitos desejados pelo atacante. Os efeitos variam conforme o ataque realizado pelo atacante e podem violar a confidencialidade, a integridade e a disponibilidade de algum serviço prestado pela vítima. Esse processo é decrito como ”cadeia” pois qualquer deficiência em uma das etapas do processo irá interromper toda a cadeia do ataque e, por isso, é de extrema importância o seu estudo, visto que sabendo como romper essa cadeia é possível elaborar técnicas de defesa mais eficientes contra ataques cibernéticos. O modelo aqui apresentado foi proposto por Hutchins, Cloppert e Amin (2011) e caracteriza o processo em diferentes fases:

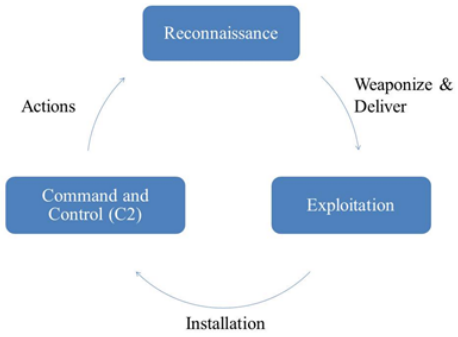
1. Reconhecimento (Reconnaissance): procura, identifica e seleciona alvos com possíveis vulnerabilidades. Normalmente são identificados como rastreadores de sites da internet que possuem como objetivo encontrar emails ou qualquer informação de identificação sobre usuários. A maneira com que os possíveis alvos são procurados, a forma como as vulnerabilidades são encontradas e se a busca é feita de forma manual ou automatizada caracterizam três classificações presentes na taxonomia definida por Mirkovic e Reiher (2004) que serão abordadas mais detalhadamente na seção subsequente.
2. Armamento (Weaponization): incorporação de um código malicioso em algum arquivo, que quando aberto por alguma aplicação da vítima, pode explorar as suas vulnerabilidades. O arquivo criado com o código invasor é denominado ”arma” (weapon). A arma possibilita procurar vulnerabilidades na máquina e disponibilizar alguma porta para o atacante estabelecer conexão e iniciar a exploração. Para a

arma ser ativada é preciso que ela seja executada por alguma aplicação da máquina vítima.

3. Entrega (Delivery): é a forma de entrega da arma para o alvo. Normalmente, são utilizados emails com a arma em anexo, downloads em websites e arquivos presentes em pendrives.
4. Exploração (Exploitation) : depois da arma ser entregue para a vítima, a vítima precisa ativá-la. Frequentemente, a arma é executada por aplicações ou pelo próprio sistema operacional nos quais foram encontradas vulnerabilidades pelo atacante.
5. Instalação (Installation): é efetuada a instalação de backdoors, que são métodos de ignorar autenticidade ou outros controles de segurança com o objetivo de de acessar o sistema e os dados contidos nele (WYSOPAL; ENG, 2007). No caso de ataques DDoS, os backdoors permitem a subversão da máquina.
6. Comando e Controle (Command and Control): as vítimas infectadas estabelecem uma conexão de Comando e Controle (C2). Após estabelecida a conexão, o atacante possui controle da máquina invadida. O objetivo de muitas armas é conseguir subverter o máximo de máquinas possíveis, com isso é possível estabelecer uma comunicação tanto com os mestres como com os próprios agentes. O tipo de comunicação caracteriza uma classificação da taxonomia de ataques DDoS definida por Mirkovic e Reiher (2004) e será abordada mais detalhadamente na Seção subsequente.
7. Ações e Objetivos (Actions on Objectives): somente após o estabelecimento da comunicação, os atacantes podem começar a exercer as ações para atingir o objetivo inicial do ataque. Diversas classificações presentes na taxonomia de Mirkovic e Reiher (2004), que serão abordadas na próxima Seção, ocorrem nessa etapa do ataque.

Autores como Douligeris e Mitrokotsa (2004) e Mirkovic e Reiher (2004) classificam o ataque com menos etapas. Ambos agrupam as etapas *Armamento*, *Entrega* e *Exploração* em uma etapa mais genérica denominada ”Exploração”. Douligeris e Mitrokotsa (2004) criam uma etapa denominada de ”Comunicação” que consiste na comunicação entre o atacante e os mestres com o intuito de identificar dados relativos à topologia do ataque, os agentes disponíveis, quando os ataques serão realizados e quando é necessário fazer a atualização do código malicioso.

Figura 2.1: Ciclo de Iteração de um Ataque



Fonte: (HARRIS BRYAN KONIKOFF; PETERSEN, 2013)

2.1.2 Cadeia de um Ataque DDoS

Segundo Harris Bryan Konikoff e Petersen (2013), a cadeia de um ataque DDoS é composta por duas interações sobre o ciclo apresentado na Figura 2.1. O ciclo começa com o *Reconhecimento* e continua conforme as fases da cadeia de um ataque cibernético. É importante ressaltar que, dependendo do tipo e da sofisticação do ataque, nem todas as fases são aplicáveis e precisam ser realizadas.

Por definição, ataques DDoS requerem o controle de muitos agentes e mestres. Essa primeira interação consiste justamente na criação de uma rede de máquinas comprometidas, também chamada de *botnet*. Botnet é definida por Zhu et al. (2008) como uma coleção de bots que são executados de forma autônoma e automatizada. Os bots são executados em grupos de zombies que são controlados remotamente por atacantes. Um dos botnets mais populares atualmente segundo Milletary (2012) é o Citadel ZeuS. Esse botnet é uma atualização do botnet Zeus, porém como o código se tornou público, diversos autores obtiveram a oportunidade de desenvoler novas versões. As duas principais atualizações realizadas deram origem ao Citadel ZeuS e ao ICE IX. O ciclo de vida de um botnet é definido em três etapas segundo Hachem et al. (2011), sendo elas: *injeção e difusão*, *comando e controle (C2)* e *aplicação*. A etapa de injeção e difusão se assemelha as primeiras cinco fases do processo de cadeia do ataque.

A segunda interação consiste nas fases que o atacante precisa desempenhar para efetuar o ataque DDoS sendo que ele já possui acesso ao botnet construído na primeira interação. Nessa etapa, o atacante possui certa liberdade para o planejamento e execução do ataque. A fase de *Reconhecimento* consiste em encontrar vulnerabilidades no serviço que se deseja comprometer o funcionamento. Para tanto, é necessário ter o conhecimento de detalhes de configuração da rede da vítima. Esse *Reconhecimento* pode ser realizado

de forma automatizada utilizando ferramentas como Maltego, Nmap e Netcat. Na fase de *Armamento*, o atacante planeja o ataque. Os tipos de ataques que podem ser planejados são definidos por Mirkovic e Reiher (2004) como ataques semânticos e ataques de força bruta. Os ataques semânticos exploram uma designada característica ou um defeito de implementação de certo protocolo para consumir quantidade de recursos em excesso. Já os ataques de força bruta submetem uma grande quantidade de requisições com o objetivo de exaurir os recursos da vítima. Botnets como Dirt Jumper possuem a habilidade de desempenhar ataques de ambas as classificações. As fases de *Entrega*, *Exploração* e *Instalação* não são realizadas nessa interação, já que os zombies foram infectados na primeira iteração e não é necessário a entrega de nenhum código malicioso. Na fase de *Comando e Controle*, o atacante utiliza a comunicação estabelecida na primeira iteração para enviar comandos e configurações para os zombies. Essa fase se assemelha à etapa de *Comunicação* estabelecida por Douligeris e Mitrokotsa (2004). A última fase consiste no ataque sendo executado conforme foi modelado na fase anterior.

2.1.3 Critérios de Classificação

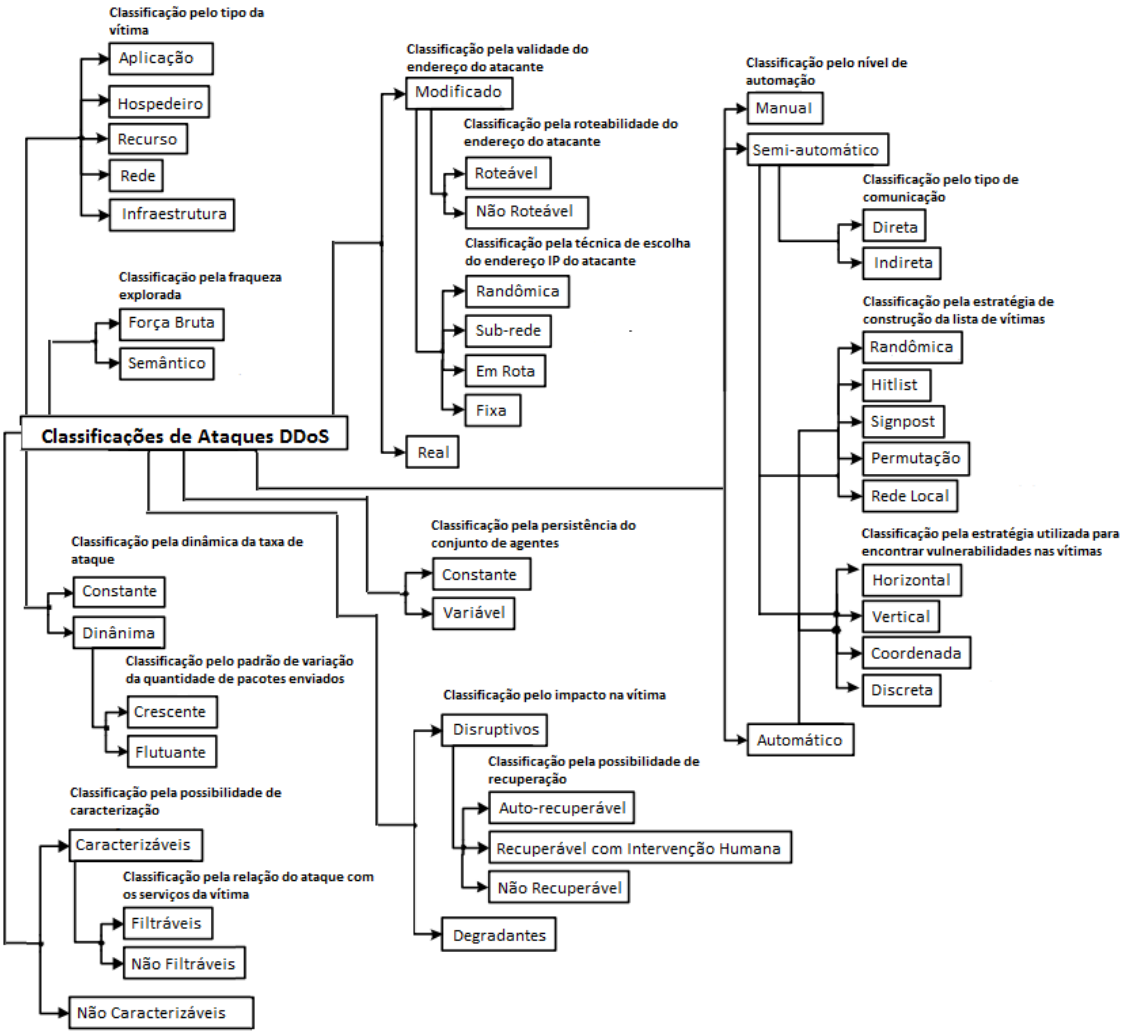
Todos os ataques podem ser classificados relacionando características de como cada uma das fases, listadas na Seção anterior, é realizada. Os critérios de classificação mostrados na Figura 2.2 foram propostos na taxonomia de Mirkovic e Reiher (2004) e serão discutidos ao longo dessa Seção.

2.1.3.1 Nível de Automação

Cada uma das fases descritas anteriormente pode ser classificada como manual, semi-automática e automática.

Os primeiros ataques DDoS podem ser classificados como manuais, visto que todas as fases eram realizadas pelo atacante de forma manual. Nos ataques DDoS semi-automáticos, as fases de *Reconhecimento* até a *Instalação* são automatizadas. Já a especificação do ataque é realizada de forma manual pelo atacante enviando comandos para os mestres. A forma como essa comunicação acontece pode ser direta ou indireta. Na comunicação direta, os agentes precisam saber a identidade dos mestres para estabelecer a comunicação e vice-versa. Por outro lado, a comunicação indireta é baseada em canais IRC (Internet Relay Chat). Esses canais garantem o anonimato do atacante por conter pa-

Figura 2.2: Classificações de Ataques DDoS



Fonte: (MIRKOVIC; REIHER, 2004) Tradução Livre

cotes de controle não diferenciáveis do fluxo normal da rede e por alternar entre os canais IRC frequentemente. Existem duas classificações referentes à como a fase de *Reconhecimento* é realizada: estratégia para construir lista de máquinas vulneráveis e estratégias para encontrar vulnerabilidade nas vítimas. As estratégias para construir a lista de máquinas vulneráveis podem ser: *randômica*, *hitlist*, *signpost*, *permutação* e *rede local*. A estratégia para achar vulnerabilidades em cada máquina dessa lista pode ser: *horizontal*, *vertical*, *coordenada* e *discreta*.

Na última classificação, todos as fases do ataque são automatizadas. Todos os detalhes do ataque são pré-programados no código de ataque e, como o atacante envia somente um comando para iniciar o ataque, é garantido uma menor exposição do atacante. Cada fase dos ataques pode ser realizada de forma diferente, por exemplo, a fase de *Reconhecimento* pode ser realizada de forma automatizada e todas as outras de forma manual ou vice-versa.

2.1.3.2 Fraqueza Explorada

Esses tipos de ataques exploram diferentes fraquezas para impossibilitar que a vítima forneça o serviço desejado. As duas classificações existentes são: *ataques semânticos* (vulnerabilidade) e *ataques de força bruta* (inundação). Os ataques semânticos consistem em aproveitar defeitos de implementação ou alguma aplicação instalada na vítima para consumir recursos em excesso. Por exemplo, em um ataque TCP SYN, a fraqueza explorada é a alocação substancial de espaço na fila de conexão após receber uma requisição de conexão. O ataque NAPTHA também explora características desse mesmo protocolo ao explorar vulnerabilidades de conexões estabelecidas e conexões finalizadas. (INSTITUTE, 2000). Por outro lado, os ataques de força bruta fundamentam-se em enviar grande quantidade de pacotes para esgotar a quantidade de recursos da vítima. Os ataques Smurfs são exemplos de ataques onde o atacante envia requisições para roteadores e esses enviam mensagens para a vítima. Para os roteadores mandarem os pacotes para a vítima, o atacante precisa substituir o seu endereço IP pelo endereço IP da vítima. As mensagens utilizadas nesse ataque são ICMP ECHO REQUEST (LAU et al., 2000). Outro ataque de inundação é o DNS request, que consiste em enviar grande quantidade dessas mensagens para sobrecarregar o servidor DNS e impossibilitar o processamento de todas as requisições recebidas (RADWARE, 2015).

2.1.3.3 Validade do Endereço do Atacante

O atacante pode tanto usar o seu endereço real, utilizado em situações em que são necessárias diversas trocas de mensagens entre agente e vítima, como mudar o endereço no cabeçalho dos seus pacotes para dificultar a descoberta de sua identidade. Quando essa técnica de mudar o endereço, também chamada de *Spoofing*, é utilizada, existem duas classificações correspondentes.

Quanto à técnica de escolha de endereço IP, o ataque pode ser classificado como *randômico*, no qual os 32-bits do endereço são sorteados aleatoriamente. É possível utilizar um endereço randômico da mesma subrede. Ao utilizar a técnica *em rota*, o atacante seleciona um endereço IP presente na rota do ataque, ou seja, entre o agente e a vítima. Outra técnica possível, é utilizar um endereço IP *fixo*. Essa técnica é utilizada frequentemente em ataques refletores onde o endereço do atacante é substituído pelo da vítima.

Outra possível classificação é quanto à possibilidade de roteamento do endereço. O endereço pode ser *roteável* e *não roteável*. Ataques que mudam o seu endereço para o endereço de outra máquina existente são definidos como roteáveis e é utilizado propositalmente em ataques refletores. Ataques em que o atacante seleciona endereço IP não utilizado ou reservado, são definidos como não-roteáveis.

2.1.3.4 Dinâmica da Taxa de Ataque

Cada agente pode enviar um fluxo de pacotes de forma *constante* ou *dinâmica* durante o ataque. A maioria dos ataques é realizada utilizando taxa constante pois apresentam ótimo custo-benefício, uma vez que o atacante pode realizar o ataque com o menor número de agentes possíveis necessários para o rompimento do serviço. Os ataques de taxa variável dificultam a detecção do ataque por parte da vítima e podem ser classificados conforme o padrão de variação da quantidade de pacotes enviados. Os ataques que apresentam um aumento gradual da quantidade de pacotes enviados e causam a exaustão lenta dos recursos da vítima são denominados *crescentes*. Como os recursos da vítima se esgotam lentamente, o ataque pode levar um tempo maior para ser detectado. Existem também ataques em que a quantidade de pacotes enviados é ajustada conforme o comportamento da vítima ou conforme configuração feita pelo atacante com o intuito de evitar a detecção por parte da vítima. Esses ataques são denominados *flutuantes*.

2.1.3.5 Possibilidade de Caracterização

Observando o conteúdo e cabeçalho dos pacotes, é possível caracterizar o ataque ou não. Os ataques *caracterizáveis* agem sobre protocolos específicos ou aplicações da vítima. Eles podem ser classificados como *filtráveis* (operam com pacotes malformados ou pacotes para serviços que não são críticos) e *não-filtráveis* (operam com pacotes bem formados que solicitam serviços críticos). TCP SYN, ICMP ECHO e DNS REQUEST são exemplos de ataques pertencentes à classe de ataques caracterizáveis. Os ataques *não caracterizáveis* procuram consumir os recursos da vítima utilizando diversos protocolos e aplicações. É importante ressaltar que a classificação dos ataques em caracterizáveis e não caracterizáveis depende fortemente da quantidade de recursos que podem ser dedicados para a classificação. Por exemplo, um ataque que utiliza um conjunto misturado de pacotes TCP SYN, TCP ACK, ICMP ECHO, ICMP ECHO REPLY provavelmente seria classificado como caracterizável, mas somente após esforço e tempo consideráveis.

2.1.3.6 Persistência do Conjunto de Agentes

Devido ao crescente número de ataques que variam o conjunto de máquinas que participam do ataque, foi estabelecida uma nova classificação por Mirkovic e Reiher (2004). Sendo assim, existem duas classificações quanto à persistência do conjunto de máquinas participantes de um ataque: *conjunto constante de agentes* e *conjunto variável de agentes*. Durante ataques com conjunto constante de agentes, todos os agentes recebem os mesmos comandos em um determinado tempo. Durante ataques com conjunto variável de agentes, o atacante divide o conjunto de agentes disp oníveis em vários grupos e então utiliza somente um dos grupos para efetuar o ataque em um determinado instante. O grupo que está efetuando o ataque é alternado frequentemente, com o intuito de dificultar a identificação da ocorrência do ataque pela vítima.

2.1.3.7 Tipo de Vítima

Os ataques podem ser destinados à *aplicações*, *hospedeiros*, *recursos*, *redes* e *infraestruturas*. Os ataques à *aplicação* desabilitam o uso da aplicação pelo cliente e possivelmente consomem recursos da vítima. Por exemplo, ataques de inundação HTTP possuem duas variações GET e POST , mas ambas possuem o objetivo de sobrecarregar o servidor web e impossibilitar o uso da aplicação (MYERS, 2015). Os ataques à *hospedeiro* desabilitam o acesso a máquina vítima ao sobrecarregar ou desabilitar o seu

mecanismo de comunicação ou ao fazer o hospedeiro parar de funcionar. Um exemplo desse ataque é o TCP SYN (RADWARE, 2015). Os ataques à *recurso* procuram selecionar como alvo algum recurso crítico do sistema, normalmente roteadores, servidores de DNS ou algum link essencial para a rede. Esses ataques são, geralmente, prevenidos replicando os serviços críticos. Os ataques à *rede* consomem a largura de banda de entrada da rede da vítima com pacotes cujo endereço de destino pode ser escolhido com base no espaço de endereçamento da rede da vítima. Os ataques à *infraestrutura* possuem como alvo algum serviço distribuído, tal como servidores DNS, protocolos de roteamento, servidores de certificação e etc, que são essenciais para o funcionamento da Internet.

2.1.3.8 Impacto na Vítima

O ataque pode ser caracterizado conforme o impacto na vítima. Os ataques podem ser *disruptivos* ou *degradantes*. Os ataques disruptivos possuem como objetivo negar completamente o serviço prestado pela vítima para seus clientes. É possível classificá-los em relação ao modo de recuperação que esses ataques possuem. Um ataque pode ser *auto-recuperável*, *recuperável com intervenção humanda* e *não recuperável*. Em ataques *auto-recuperáveis*, as vítimas se recuperam do ataque sem nenhuma intervenção humanda assim que mais nenhum pacote do atacante é recebido pela vítima. Em ataques que são *recuperáveis com intevenção humana*, é necessário a ação humana para disponibilizar o serviço novamente para seus clientes. Por exemplo, um ataque que causa o congelamento dos recursos da vítima pode ser recuperável com a reinicialização do sistema da vítima ou com uma mudança na sua configuração. Ataques *não recuperáveis* resultam em dano permanente no hardware da vítima. Um novo hardware precisa ser adquirido para substituir a peça atingida. Todos os ataques registrados podem ser classificados como *disruptivos*. Por outro lado, os ataques *degradantes* possuem como objetivo consumir parte dos recursos da vítima. Como esses ataques não geram perda total do serviço, eles podem demorar a ser identificados.

2.2 Infraestruturas de Redes

A Seção 2.2.1 aborda a motivação para criação das redes definidas por software, apresentando sua arquitetura e o protocolo *OpenFlow*. A Seção 2.2.2 apresenta a motivação para a criação de redes NFV e suas características. Na Seção 2.2.3, a relação entre

ambas as infraestruturas é descrita.

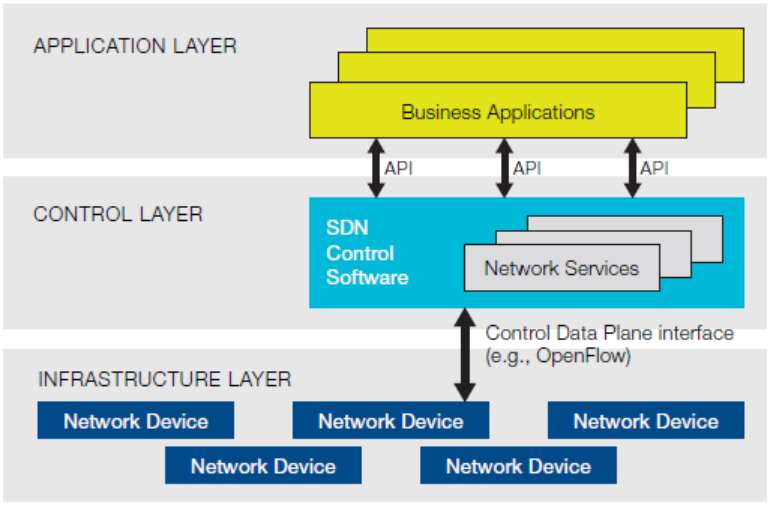
2.2.1 Redes Definidas por Software

Redes tradicionais podem ser caracterizadas por apresentar plano de controle e de dados unificado, um sistema distribuído de controle dos equipamentos da rede e um uso de uma única infraestrutura de rede física. O plano de controle é responsável pela configuração do dispositivo e pela programação dos caminhos do fluxo de dados . Quando esses caminhos já foram determinados, eles são estabelecidos no plano de dados. O encaminhamento dos pacotes no hardware é baseado nessa informação proveniente do plano de controle. Uma vez estabelecida as regras de encaminhamento dos fluxos em cada dispositivo, só é possível estabelecer mudanças nessa regras reconfigurando os dispositivos. *Redes Definidas por Software* (Software Defined Networks, ou SDN) constituem um novo paradigma para o desenvolvimento de pesquisas em redes de computadores que vem ganhando a atenção de grande parte da comunidade acadêmica e da indústria da área de redes de computadores por justamente proporcionar maior flexibilidade e programabilidade ao separar o plano de controle do plano de dados.

A ONF (Open Networking Foundation), que é uma organização sem fins lucrativos dedicada ao desenvolvimento, padronização e comercialização de SDN, decreve redes definidas por software como: ”Na arquitetura do SDN, os planos de controle e de dados são separados, o controle da rede é logicamente centralizado, e a camada de infraestrutura é abstraída das aplicações” (ONF, 2012). A arquitetura de redes baseadas em SDN, como é apresentada na Figura 2.3, é divida em três camadas: a *Camada de Infraestrutura*, a *Camada de Controle* e a *Camada de Aplicação*.

- Camada de Infraestrutura: É composta principalmente por dispositivos de encaminhamento, incluindo switches físicos e switches virtuais. Esses switches são acessíveis através de protocolos de comunicação, tais como Nettle, OpenFlowJ e OpenFaucet para dispositívos virtuais e IBM RackSwitch G8264 e Juniper Junos MX-Series para dispositívos físicos. Essa camada também pode ser chamada de *plano de dados*.
- Camada de Controle: Consiste em um conjunto de controladores SDN que supervisionam e o controlam o encaminhamento. Um controlador SDN pode se comunicar com outro controlador, utilizando as interfaces de comunicação chamadas de *East-*

Figura 2.3: Arquitetura de Redes SDN



Fonte: (ONF, 2012)

bound e *Westbound*, e com a Camada de Infraestrutura utilizando a interface de comunicação denominada *Southbound*. Essa camada também pode ser chamada de *plano de controle*. OpenFlow, ForCES e PCEP são exemplos de implementações da interface Southbound. Na próxima Seção, o protocolo OpenFlow será abordado mais detalhadamente.

- Camada de Aplicação: Consiste em aplicações dos usuários finais. Virtualização da rede, rede unificada de monitoramento e análise e aplicações de segurança são exemplos de aplicações pertencentes à essa camada. A comunicação entre essa camada e a Camada de Controle é realizada utilizando a interface denominada *Northbound*.

SDN apresenta diversas características, como separação do plano de dados do plano de controle, controlador logicamente centralizado com visão global da rede, programabilidade da rede por aplicações externas, análise de tráfego baseado em software e atualização dinâmica das regras de encaminhamento, que oferecem diversos benefícios para a defesa contra ataques DDoS. A separação do plano de controle do de dados permite à pesquisadores e administradores experimentarem técnicas de defesas em ambientes reais e maior flexibilidade para configurar, gerenciar e otimizar os recursos da rede, facilitando a experimentação de novos protocolos e mecanismos de defesa. Essa separação também garante abstração entre as camadas de controle e de infraestrutura, o que elimina possíveis restrições impostas pelos fabricates dos dispositivos. O controlador com visão global da rede possibilita a construção de políticas de segurança e o monitoramento ou

análise de padrões de tráfego. Como o controlador é logicamente centralizado, torna-se possível eliminar temporariamente hospedeiros comprometidos e autenticar hospedeiros utilizando RADIOUS (Remote Authentication Dial In User Service). A programabilidade da rede por aplicações externas possibilita a utilização de sistemas existentes de detecção de intrusões (Intrusion detection systems, ou IDSs) e sistemas de prevenção de intrusões (Intrusion prevention systems, ou IPSs). Já a análise de tráfego baseada em software permite o uso de diversos algoritmos que utilizam inteligência artificial e banco de dados. A atualização dinâmica das regras de encaminhamento é de grande utilidade para a parte de resposta à um ataque DDoS, visto que, baseada na análise do tráfego, atualizações ou novas regras de encaminhamento podem ser distribuídas pela rede para descarte dos pacotes maliciosos sem atraso.

2.2.1.1 OpenFlow

O OpenFlow é a primeira interface padronizada projetada especificamente para SDN e consiste no protocolo de comunicação entre a Camada de Infraestrutura e a Camada de Controle.

Para que uma rede programável com OpenFlow exista, equipamentos habilitados que possam alterar suas tabelas de encaminhamento conforme as decisões de um controlador em software que é conectado a eles por um canal de comunicação seguro, são necessários.

Os componentes da arquitetura OpenFlow, apresentados na Figura 2.4, podem ser divididos em quatro diferentes tipos:

- Tabela de Fluxos (Flow Tables): Cada registro na tabela de fluxos do hardware de rede é composto por regras, ações e estatísticas. A regra é formada pela definição dos valores de um ou mais campos do cabeçalho do pacote, é por meio dela que é determinado o fluxo. As ações então ficam associadas ao fluxo e vão determinar como os pacotes devem ser processados, para onde vão ser encaminhados ou se serão descartados. As estatísticas consistem de contadores utilizados para manter estatísticas de utilização e para remover fluxos inativos ou que não existam mais. Logo, os registros da tabela de fluxos são interpretados pelo hardware como decisões do plano de controle em software.
- Protocolo OpenFlow: É o protocolo para a comunicação entre o switch e o controlador de rede para que haja a troca de mensagens por um canal seguro.

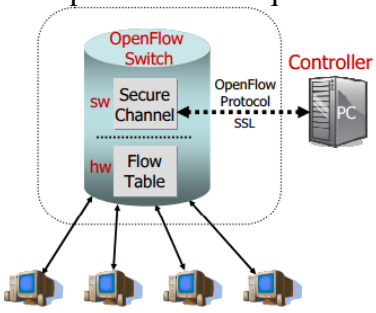
- Controlador: O controlador é responsável por tomar as decisões adicionando e removendo entradas na tabela de fluxos de acordo com uma política de encaminhamento. O controlador é uma camada de abstração da infraestrutura física que tem como objetivo facilitar o desenvolvimento de aplicações e serviços que gerenciam as entradas de fluxo na rede.
- Canal Seguro: Este elemento não só garante que uma rede SDN não sofra ataques de elementos mal intencionados como garante também que a troca de informa- ções entre os comutadores e controladores da rede sejam confiáveis com baixa taxa de erros. De acordo com as especificações do projeto do OpenFlow, o uso de SSL/TLS é recomendado com o objetivo de garantir confidencialidade em toda comunicação.

Quando um pacote qualquer chega ao switch, diferentes ações podem ser tomadas conforme características de cada fluxo. Segundo McKeown et al. (2008), as seguintes ações podem ser executadas:

- Se o fluxo não possui nenhuma entrada corresponde na tabela de fluxos do switch, os pacotes são encapsulados e enviados através de uma comunicação segura para o controlador. Por apresentar uma visão global da rede, o controlador é responsável por definir se é preciso adicionar ou modificar alguma regra das tabelas de fluxos dos switches pertencentes à rede. Após a inserção ou modificação das entradas necessárias, os pacotes pertencentes à aquele fluxo serão encaminhadas conforme a nova regra.
- Se o fluxo possuí registro correspondente na tabela de fluxos do switch, os pacotes são encaminhados para a porta de saída especificada na tabela de fluxos. Isso possibilita os pacotes serem roteados pela rede.
- Pacotes podem ser descartados pelo switch. Normalmente, utilizado como forma de segurança, visto que fluxos indesejados podem ser identificados pelo controlador.

Embora cada dispositivos físico apresente especificações diferentes, o OpenFlow explora um conjunto de informações presentes em todos eles, o que garante a compatibilidade do protocolo com dispositivos presentes nas redes tradicionais. Com isso, pesquisadores são beneficiados, visto que experimentos podem ser realizados em paralelo à redes existentes, já que com as regras da tabela, é possível tratar cada fluxo independentemente. E garante à administradores a possibilidade da utilização de ambos os modelos ao mesmo tempo, possibilitando o aumento da utilização do OpenFlow conforme a necessidade da rede.

Figura 2.4: Componentes da Arquitetura OpenFlow



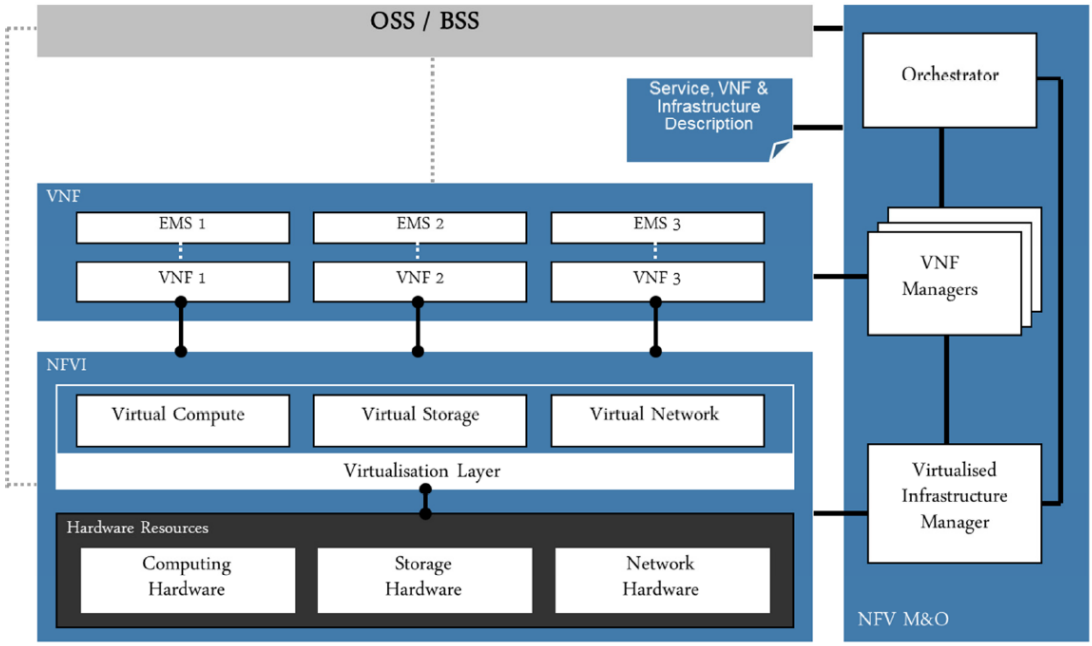
Fonte: (MCKEOWN et al., 2008)

Detalhes e especificações técnicas da arquitetura SDN e do protocolo OpenFlow podem ser encontradas na documentação oficial disponibilizada pela Open Networking Foundation (ONF).

2.2.2 Virtualização de Função de Rede

As redes de computadores são constituídas por uma vasta quantidade de hardware com diferentes características. Para o desenvolvimento de um novo serviço de rede é necessário outra grande quantidade de hardwares específicos e também é preciso encontrar espaço para acomodar e prover energia para cada novo dispositivo utilizado. Esse processo é de extrema dificuldade, considerando o custo da energia e a dificuldade para projetar e integrar dispositívos de diferentes provedores. A virtualização de funções da rede, também denominada *Network Functions Virtualization (NFV)*, oferece uma alternativa para solucionar esses problemas. A virtualização de funções da rede transforma como operadores projetam suas redes ao implementar funções de redes, tais como firewalls, CDN e message routers, em softwares que são executados em ambientes virtualizados em hardwares de uso genérico. A grande vantagem de utilizar dispositivos virtuais é a possibilidade de os instanciar por demanda sem a necessidade de instalação de outro equipamento. Isso reduz o custo ao não necessitar novos equipamentos e ao reduzir o consumo de energia por serviços que são menos necessários para o momento. Por exemplo, com base no tráfego é possível instanciar um ou mais message routers. Surge uma nova possibilidade de desenvolver e testar esses serviços em uma mesma infraestrutura, o que garante teste e intregração mais eficientes, reduzindo custo de desenvolvimento e tempo para disponibilização no mercado. A separação do software do hardware possibilita a evolução independente do software em relação ao hardware, e vice-versa. Essa separação

Figura 2.5: Arquitetura de Redes NFV



Fonte: (ETSI, 2013)

e a diminuição do tempo para comercialização estimulam o aparecimento de novos provedores de serviço, o que aumenta a quantidade de serviços disponíveis (ETSI, 2012). Os benefícios do uso de NFV estabelecem um avanço significativo em redes móveis e redes fixas, porém existem dificuldades específicas a cada caso de aplicação, como é relatado em (ETSI, 2013).

A arquitetura do NFV é ilustrada na Figura 2.5 e é composta por três blocos principais: *NFV Infrastructure (NFVI)*, *Virtualized Network Function (VNF)* e *NFV MANO (Management and Orchestration ou M&O)*. A NFVI fornece os recursos virtuais que são necessários para a execução de funções virtualizadas de redes (VNF). Dentre os recursos estão hardware, componentes de aceleração e uma camada de software que virtualiza e abstrai o hardware sendo utilizado. O bloco VNF é a implementação da função de rede que é capaz de ser executada pelo NFVI. Isso pode ser acompanhado pelo Element Management System (EMS) que é capaz de administrar um único VNF. O VNF é o elemento que corresponde aos nodos das redes de hoje, o que com NFV é esperado ser provido como pura implementação de software. A NFV MANO abrange a orquestração e administração do ciclo de vida de recursos físicos e/ou de recursos de software que auxiliam a virtualização da infraestrutura e a gerência de VNFs. A NFV MANO interage com OSS/BSS, o que permite a integração do NFV à redes já existentes e gerenciadas. A interação entre esses blocos se dá através de pontos de referência definidos com o intuito de

que diversas entidades possam fornecer serviços específicos independentes (ETSI, 2013).

2.2.3 Relação Complementar entre SDN/NFV

A virtualização de funções de rede é um serviço complementar à noção de SDN, mas não dependente. NFV pode ser implementada sem ser relacionada com SDN, embora os dois conceitos combinados possam pontencializar grande ganho de CAPEX, OPEX, espaço, consumo de energia e rapidez de inovação. Os dois conceitos representam um caminho em direção a um hardware mais genérico e a um software mais livre, onde o controle e gerência centralizados pelo SDN podem ser alcançados, em parte, através das funções virtualizadas provenientes de NFV. A aplicação de ambos conceitos resulta em serviços mais flexíveis em relação a monitoramento, gerência, análise de tráfego e controle de carga. Sendo assim, NFV é capaz de suportar SDN ao fornecer a infraestrutura sobre a qual o software SDN irá ser executado. Além do mais, os objetivos de NFV se assemelham aos das redes baseadas em SDN ao utilizar padrões comuns de hardware.

Apesar da existência de inúmeros benefícios da utilização de redes baseadas em infraestruturas SDN/NFV, existem diversos desafios que precisam ser investigados para garantir a adoção de ambos os conceitos. Dentre os desafios de investigação, está a necessidade de analisar e propor mecanismos de mitigação de ataques DDoS em redes baseadas em SDN/NFV (Seção 3.0 e Seção 4.0).

3 MITIGAÇÃO DE ATAQUES DDOS EM SDN/NFV

Técnicas que visam lidar com ataques DDoS podem ser divididas em três categorias: *Prevenção*, *Detecção* e *Mitigação*. A prevenção procura evitar a execução de ataques DDoS com o uso de filtros estáticos, desabilitando serviços ou mudando IP da vítima. A detecção realiza o monitoramento que abstrai dados relevantes do tráfego. Posteriormente, a análise desses dados é realizada e caso seja detectado alguma anormalidade, dados característicos do ataque são encaminhados para o mecanismo de mitigação. O sistema de mitigação é responsável por tentar minimizar o impacto do ataque DDoS em clientes legítimos que acessam o serviço. Mirkovic e Reiher (2004) divide as técnicas de mitigação em identificação de agentes, reconfiguração, rate-limiting e filtering. Já Shameli-Sendi et al. (2015) apresenta mecanismos de defesa usando uma categorização baseada em filtering e rate-limiting. A identificação de agentes, categorizada por Mirkovic e Reiher (2004) como técnica de mitigação, é classificada por Specht e Lee (2004) e Douligeris e Mitrokotsa (2004) como técnica utilizada em conjunto com filtering, rate-limiting ou reconfiguração. Portanto, a categorização adotada nesse trabalho consiste em estratégias de mitigação baseadas em técnicas de *Rate-Limiting*, *Filtering* e *Reconfiguração* contra ataques DDoS em redes baseadas em SDN/NFV. As estratégias apresentadas partem do pressuposto que uma técnica de detecção foi previamente empregada. Por exemplo, Silva et al. (2016) apresenta um framework, chamado *ATLANTIC*, que utiliza SDN para combinar o uso da teoria da informação para calcular desvios na entropia das tabelas de encaminhamento com um conjunto de algoritmos de machine learning para classificar os fluxos. Ou ainda, Wang et al. (2015) que propõe uma arquitetura para detecção baseada em anomalias e mitigação utilizando SDN.

Um mecanismo de mitigação contra ataques DDoS pode ser caracterizado por dois conceitos: a *tática* e a *estratégia*. Segundo Shameli-Sendi et al. (2015), uma tática de mitigação consiste na abordagem local implementada em um componente da rede, ou seja, é uma ação que um componente da rede pode executar para reduzir o impacto do ataque na rede. Já a estratégia de mitigação é o método global que faz o uso de táticas de mitigação em diferentes locais da rede com o intuito de minimizar a eficácia do ataque.

Neste capítulo, as táticas de mitigação e suas respectivas estratégias desenvolvidas para o problema proposto serão apresentadas. A Seção 3.1 descreve como as estratégias baseadas em Rate-Limiting podem ser utilizadas. A Seção 3.2 modela estratégias baseadas em Filtering em redes baseadas em infraestruturas SDN/NFV. A Seção 3.3 apresenta

estratégias baseadas em Reconfiguração adaptadas para redes baseadas em infraestruturas SDN/NFV.

3.1 Estratégias baseadas em Rate-Limiting

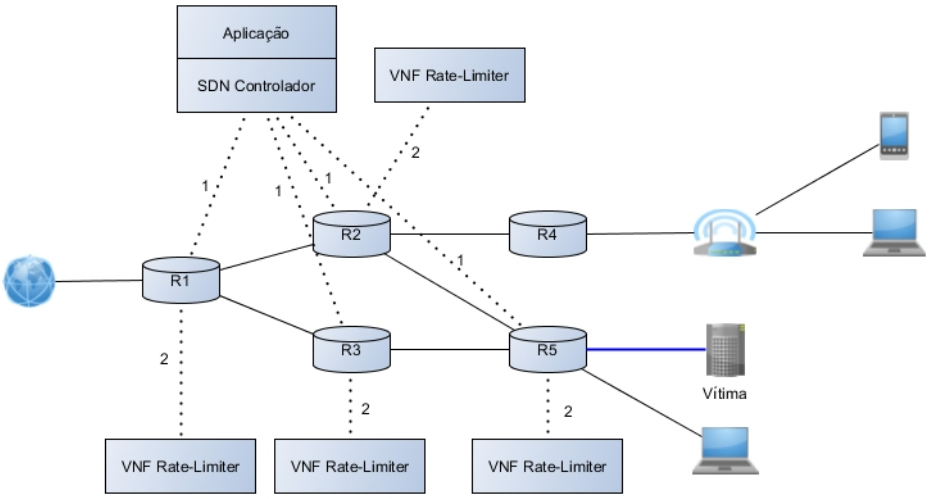
Essa é a forma mais popular para mitigar ataques DDoS, tendo em vista que ela consiste em descartar uma fração dos pacotes de ataques que foram caracterizados como maliciosos pelo mecanismo de detecção. Isso é realizado com o objetivo de continuar a prover o serviço, pelo menos, para uma parte dos clientes. Existem duas formas de implementar o Rate-Limiting: *Flow Rate-Limiting* e *Aggreagate Rate-Limiting*. Flow Rate-Limiting considera fluxos individuais e, portanto, o tráfego é limitado de acordo com um valor padrão determinado para cada um dos fluxos (CHEN; LONGSTAFF; CARLEY, 2004). Já o Agreggate Rate-Limiting é baseado na restrição da largura de banda do tráfego agregado. O tráfego pode ser agregado conforme características em comum de vários fluxos, tais como: protocolo de transporte, endereço IP de origem ou destino e protocolo utilizado pela aplicação.

As Seções 3.1.1 e 3.1.2 apresentam detalhes das estratégias de Pushback e Throttling, respectivamente.

3.1.1 Pushback

A proposta inicial de implementação de Pushback utiliza Agreggate Rate-Limiting e é uma estratégia de defesa colaborativa, na qual o congestionamento é detectado pelo roteador e são coletadas informações referentes aos pacotes descartados (SHAMELI-SENDI et al., 2015). Essas informações auxiliam no estabelecimento de assinaturas referentes ao ataque. Após isso, pacotes que correspondem àquela assinatura são limitados pelo rate-limiter. Mensagens de pushbacks são enviadas para os roteadores de upstream. Os roteadores aplicam as regras do rate-limiter, respondem ao roteador que inicialmente sinalizou a ocorrência do ataque e propagam mensagens para os seus roteadores de upstream. A limitação do ataque é alcançada impondo regras de limitação de tráfego o mais próximo dos roteadores de borda, porém pacotes que apresentam características em comum com pacotes maliciosos também podem ser descartados. Segundo Ioannidis e Bellovin (2002), cada roteador possui uma quantidade de assinaturas, um rate-limiter e um

Figura 3.1: Pushback: Ordem de Execução



Fonte: Autor

pushbackd (pushback daemon). Cada pacote é verificado com base nas assinaturas presentes no roteador e, caso a assinatura do pacote esteja presente na lista do roteador, o rate-limiter descarta ou encaminha para a fila de saída conforme o seu fator de limitação. Todos os pacotes descartados pelo rate-limiter são encaminhados para o *pushbackd*. O daemon periodicamente atualiza a taxa de limitação do rate-limiter, informa os daemons upstream para atualização das taxas e das assinaturas presentes em cada roteador.

A estratégia de mitigação pode ser aplicada em redes baseadas em infraestruturas SDN/NFV separando o pushbackd e o rate-limiter do roteador. Na Figura 3.1, uma rede composta por cinco roteadores e que está sendo vítima de um ataque DDoS é apresentada. O algoritmo de detecção, implementado por uma aplicação presente no nível superior ao controlador SDN, identificou que o link R5-V está congestionado e estabeleceu assinaturas para a sua mitigação. O controlador SDN, por possuir uma visão global da rede, redireciona diretamente o tráfego malicioso de todos roteadores upstream da vítima (1) - ou seja, o tráfego dos roteadores R1, R2, R3 e R5 - para *VNFs Rate-Limiter* que serão responsáveis por tratar os pacotes pertencentes à assinatura estabelecida pelo algoritmo de detecção (2). Todos os pacotes descartados pelas VNFs são armazenados para o acesso da aplicação e cálculo dinâmico da nova taxa de limitação imposta às VNFs.

Sathyanarayana (2011) apresenta a implementação da estratégia em redes baseadas em SDN utilizando filtragem que é implementada no próprio roteador. Contudo, a utilização de funções de rede virtualizadas proporcionam elasticidade em relação ao tamanho do tráfego do ataque ao permitir que VNFs possam ser instanciadas por demanda (ETSI, 2012). Com isso, se o tráfego aumentar significativamente, VNFs poderão conti-

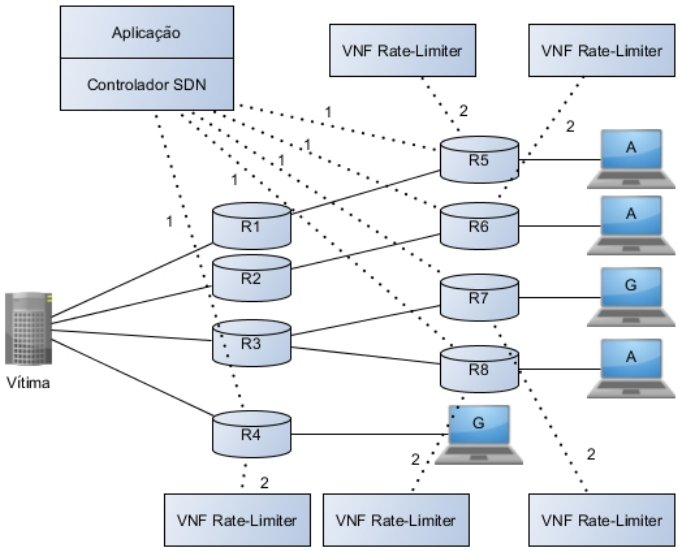
nuar a prover serviço, o que não acontece em redes tradicionais.

3.1.2 Throttling

A estratégia consiste em um servidor, que está sendo vítima de um ataque, instalar fatores de limitação de tráfego, também conhecidos como *rate throttle*, em um conjunto de roteadores upstream a uma determinada quantidade de *k* saltos de distância ou a uma distância menor que *k* mas diretamente conectado a um host. Ao serem instalados esses fatores de limitação, todo o tráfego que passa pelo roteador com destino à vítima é limitado, ou seja, é utilizado Aggreagate Rate-Limiting com base no destino do pacote. Sendo assim, não existe diferenciação de tráfego malicioso e tráfego legítimo, portanto, tanto pacotes maliciosos podem ser encaminhados para o servidor quanto pacotes legítimos podem ser descartados. Diferencia-se da estratégia de pushback por impor limite de tráfego em todos os roteadores a uma determinada distância, uma vez que no pushback o tráfego em todos roteadores upstream é limitado. Outro ponto importante é que a estratégia limita todo o tráfego, ao passo que no pushback somente o tráfego que possui determinada assinatura é limitado. Essa estratégia distribui a capacidade do servidor de uma forma max-min justa entre os roteadores. Isso significa que somente fluxos que não obedecem suas limitações são punidos. O objetivo principal é fazer a vítima operar conforme a sua capacidade. Para isso ser alcançado, o cálculo do fator de limitação é atualizado regularmente conforme valor mínimo de utilização da vítima, L_s, e sua capacidade máxima, U_s, e é enviado para os roteadores através de um sinal de redução da taxa, α , ou de aumento de taxa, β.

Em redes baseadas em infraestruturas SDN/NFV, essa estratégia pode ser implementada utilizando uma aplicação que monitore o congestionamento da rede, aplicando rate-limiting em VNFs e redirecionando o tráfego de todos os roteadores com *k* saltos de distância da vítima para essas VNFs. Na Figura 3.2, uma rede a qual a vítima está sofrendo ataque DDoS é apresentada e são sinalizados os roteadores nos quais é necessário a instalação do throttle quando *k* é igual a 2 - i.e, R4, R5, R6, R7 e R8. A aplicação, apresentada na Figura 3.2, monitora a utilização dos hospedeiros na rede. Se algum sistema final estiver operando acima da sua capacidade máxima (U_s) ou abaixo do seu valor mínimo (L_s), a aplicação gera um novo fator de limitação (α ou β) que é utilizado para atualizar a configuração do VNF Rate-Limiter e, se necessário, comunica a aplicação que devem ser instaladas throttles em roteadores com *k* saltos de distância da vítima. O

Figura 3.2: Throttling: Ordem de Execução



Fonte: Autor

controlador SDN é responsável por modificiar a tabela de encaminhamento dos dispositivos de encaminhamento envolvidos (1) com o intuito de redirecionar do tráfego para a VNF Rate-Limiter (2) toda vez que uma nova throttle é necessária, assim como remover o redirecionamento quando a utilização da vítima for menor que o seu limite mínimo (L_s).

3.2 Estratégias baseadas em Filtering

Filtering consiste em descartar pacotes de acordo com alguma característica específica do pacote. Essas características normalmente consistem em endereços de origem ou destino, porta utilizada ou protocolo e são providas pelo Intrusion Protection System (IPS). A filtragem pode ser feita conforme a 5-tupla (IP origem, porta origem, IP destino, porta destino e protocolo), conforme indicações predefinidas que são características de um tráfego malicioso, ou conforme desvios de características do tráfego em relação a um comportamento preestabelecido. Diversos mecanismos de defesa adotam filtros estáticos para prevenção de ataques DDoS. Porém, como apresentado na Seção 2.1.3.6, existem diversos ataques que variam o conjunto de agentes pertencentes ao ataque e, por isso, as estratégias baseadas em filtragem utilizam filtros dinâmicos.

As próximas Seções (3.2.1 e 3.2.2) apresentam modelos das estratégias de Firewalls Cooperativos e Honeypots, respectivamente.

3.2.1 Firewalls Cooperativos

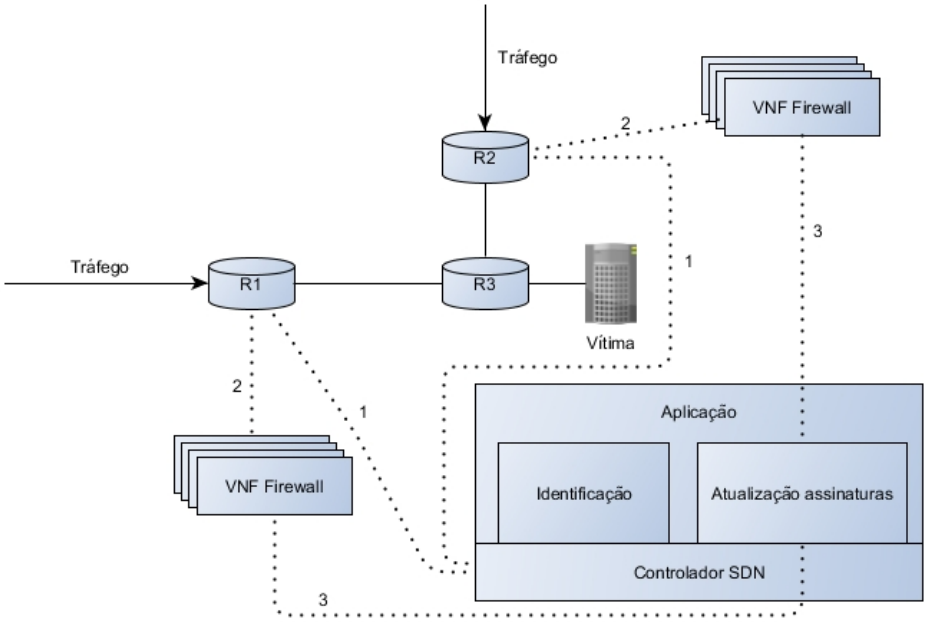
Diversas estratégias de mitigação que utilizam firewalls existem. A estratégia proposta por (ZARGAR; JOSHI, 2013) é baseada em dois componentes responsáveis por executar o monitoramento e a detecção do ataque de forma colaborativa, e distribuir técnicas de mitigação para roteadores presentes em diversas ISPs através de um servidor central presente em cada ISP. . A estratégia abordada tem como base o sistema proposto por El-Soudani e Eissa (2003), o qual é composto por dois tipos de firewall: Defender Firewalls (DFWs) e Assistant Firewalls (AFWs). Os DFWs são conectados à borda da rede e são responsáveis por verificar cada pacote entrante na rede. Se for detectado algum ataque, o DFW envia uma mensagem em broadcast com uma regra de filtragem para os AFWs, que são responsáveis por mitigar o ataque o mais próximo possível da origem.

A técnica de filtragem com a utilização de firewalls em redes baseadas em infraestruturas SDN/NFV utiliza funções virtualizadas de rede para a implementação dos AFWs e o conceito de Redes Definidas por Software para a implementação de firewalls reativos. A ordem de execução da estratégia de mitigação contra ataques DDoS, como mostrada na Figura 3.3, começa com a identificação do ataque realizada pela aplicação. A identificação se assemelha à implementação dos algoritmos de detecção que são executados pelos DFWs, diferenciando-se por ser executada não na borda da rede, mas sim no núcleo. O controlador SDN é responsável por modificar a tabela de encaminhamento dos roteadores de borda (1) para redirecionar o tráfego para as VNFs Firewall instanciadas por demanda (2). Isso caracteriza um sistema de filtragem reativo, uma vez que a filtragem só é executada após a identificação do ataque. A VNF Firewall mantém uma lista de IPs legítimos de origem e regras para filtragem de ataques conhecidos, assim como regras que fizeram o algoritmo de detecção de ataque habilitar a VNF Firewall. Todos pacotes entrantes na rede são verificadas pelo VNF Firewall e, conforme a sua caracterização podem ser descartados ou encaminhados para o seu destino original. O sistema de identificação do ataque, que é executando frequentemente, com base nos dados obtidos pode gerar novas regras de filtragem que devem ser instaladas na VNF Firewall (3). Portanto, o processo de atualização das regras de filtragem se difere da proposta do sistema de El-Soudani e Eissa (2003), visto que, as mensagens broadcast não são enviadas pelo DFWs, mas pela aplicação.

A realização de filtragem pelo próprio dispositivo de encaminhamento foi considerada, sendo assim, todas as regras de filtragem deveriam ser instaladas no próprio

dispositivo de encaminhamento. Contudo, se houvessem muitas regras de filtragem instaladas no dispositivo de encaminhamento, o desempenho do encaminhamento poderia ser afetado. A utilização de funções de rede virtualizadas para a implementação dessa estratégia proporciona elasticidade, ao possibilitar aumentar o número de VNFs por demanda.

Figura 3.3: Firewalls: Ordem de execução



Fonte: Autor

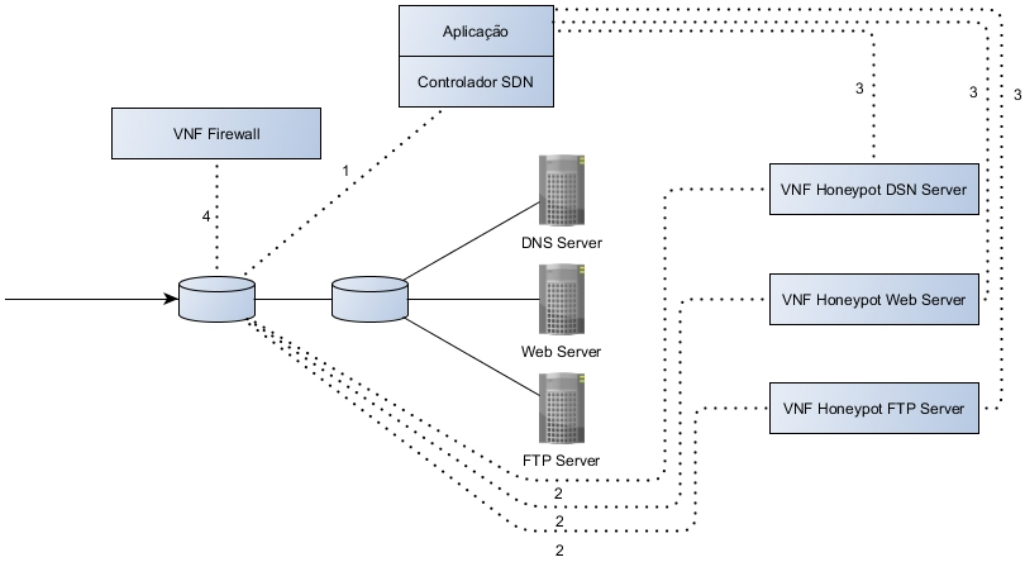
3.2.2 Honeypots

Um honeypot pode ser definido como uma armadilha para o atacante, que procura reproduzir algumas ou todas atividades realizadas de um sistema real e registrar as atividades do atacante (JOSHI; SARDANA, 2011). Por exemplo, se é identificado que estão sendo enviados pacotes maliciosos para um web server, os pacotes são direcionados para o honeypot processar. O honeypot interage com o atacante de uma forma similar ao web server real, ou seja, a resposta que o atacante obtém deve ser igual à resposta do serviço real. Para isso, segundo Weiler (2002), três problemas devem ser solucionados: a) o ataque deve ser detectável, b) os pacotes devem ser redirecionados para o honeypot e c) o honeypot deve ser capaz de simular a infraestrutura da rede da instituição, pelo menos as partes conhecidas pelo atacante.

Com o uso de redes baseadas em infraestruturas em SDN/NFV, o primeiro problema é solucionado utilizando uma aplicação no nível superior ao controlador SDN que

permite a detecção do ataque e possui um mapeamento de cada honeypot para o seu serviço. O controlador SDN utiliza esse mapeamento para redirecionar o tráfego para o honeypot correspondente. A solução para o terceiro problema é alcançada utilizando uma função de rede virtualizada para cada seviço que pode ser acessado pelo atacante.

Figura 3.4: Honeypots: Fluxo entre VNFs



Fonte: Autor

Na Figura 3.4, a ordem de execução da estratégia é apresentada. Primeiramente, a aplicação é responsável por detectar a ocorrência de um ataque e sinalizar para o controlador SDN para qual VNF o fluxo pertencente àquela assinatura deve ser encaminhado. O controlador SDN modifica a tabela de encaminhamento do roteador de borda (1) que de acordo com o ataque identificado redireciona o tráfego para a correta VNF Honeypot (2). Cada Honeypot é responsável em prover um serviço, do ponto de vista do atacante, exatamente igual ao serviço original. A VNF Honeypot correspondente coleta informações do atacante ao mesmo tempo que fornece o serviço. Como todos pacotes maliciosos são redirecionados para as Honeypots, é assegurado que a rede não virtualizada está protegida contra os ataques. Após serem coletadas informações suficientes a respeito da origem do atacante pela Honeypot, a VNF sinaliza a aplicação (3) que redireciona o tráfego para a VNF Firewall (4) que é responsável por descartar pacotes pertences ao ataque caracterizados pelas informações obtidas pela VNF Honeypot. Como as honeypots são implementadas como VNFs ao invés de servidores físicos, a solução se torna mais econômica e proporciona menor tempo de resposta em caso de comprometimento de VMs.

3.3 Estratégias baseadas em Reconfiguração

A estratégia de reconfiguração muda a topologia, o mecanismo de defesa da vítima ou da rede intermediária, com o intuito de evitar ataques DDoS (MIRKOVIC, 2003). Dependendo do que for reconfigurado, essa estratégia de mitigação pode ser classificada em três categorias: *Reconfiguração de Serviço*, *Reconfiguração de Rede* e *Reconfiguração de Defesa* (SHAMELI-SENDI et al., 2015). A primeira estratégia proposta (Seção 3.3.1) pertence a um sub-tópico da Reconfigurção de Serviço, denominado *clonagem de serviço*. E a segunda estratégia apresentada (Seção 3.3.2) pertence à classificação de Reconfiguração de Rede. A Reconfiguração de Defesa aborda estratégias nas quais a capacidade do mecanismo de defesa é alterada sob demanda, e todas as estratégias aqui apresentadas possuem essa característica.

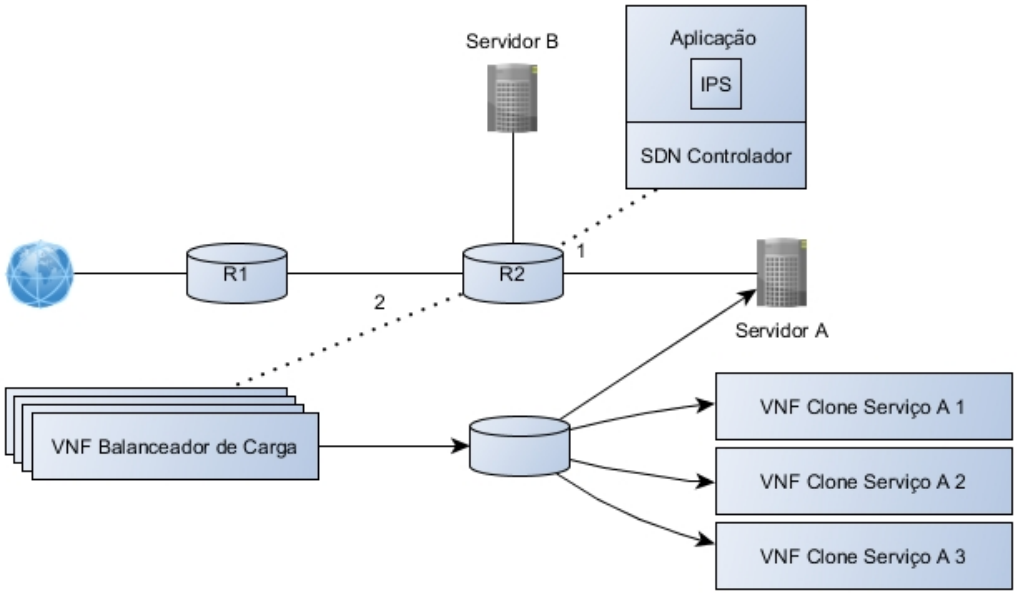
3.3.1 Reconfiguração de Serviço

Yau et al. (2005) classificam ataques DDoS como um problema de gerenciamento de recursos. Pesquisadores demonstram que o ponto principal do combate à ataques DDoS é a competição por recursos: o vencedor é o lado que possui mais recursos. Seguindo esse conceito, existem estratégias, tais como apresentadas em (YAN; EARLY; ANDERSON, 2000) e (YU et al., 2014), em que o único objetivo é aumentar os recursos da vítima sob demanda, uma vez que, grande parte dos ataques são compostos de bots alugados por determinado tempo. Yau et al. (2005) descreve uma estratégia para mitigar ataques DDoS em redes tradicionais que consiste na instalação de servidores capazes de executar código, chamados de *XenoServers*, em ISPs espalhadas pela internet. A comercialização do uso desses XenoServers é realizada para algum provedor de serviço. A qualidade do serviço é monitorada e, se é identificado deterioração de serviço devido ao aumento da demanda, o serviço é replicado para outros XenosServers que podem estar em diferentes ISPs e que também são responsáveis por gerenciar os recursos. Com isso, sites e serviços que estão sob ataques DDoS podem, em segundos, adquirir mais recursos com o objetivo de continuar a prover o seu serviço.

Em redes baseadas em infraestruturas SDN/NFV, não somente funções que implementam as táticas de mitigação, isto é, filtros e limitadores, podem ser instanciadas sob demanda, mas também os próprios sistemas que estão sendo sobrecarregados. A Figura 3.5 apresenta a ordem de execução e o caminho percorrido pelo tráfego na estratégia pro-

posta em uma rede composta por dois roteadores e dois servidores, A e B, que fornecem os serviços A e B, respectivamente.

Figura 3.5: Reconfiguração de Serviço: Ordem de execução



Fonte: Autor

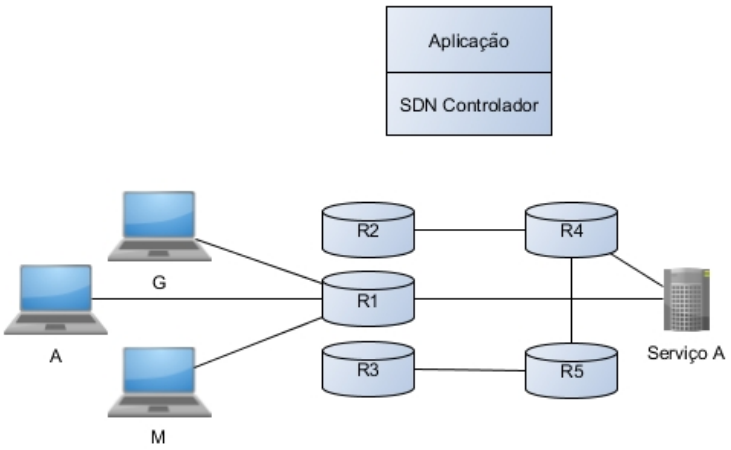
Após o Intrusion Protection System (IPS) identificar a sobrecarga de algum serviço presente na rede, o controlador SDN atualiza a tabela de encaminhamento do servidor mais próximo da vítima (1). Em seguida, todo o tráfego destinado à vítima é redirecionado para a VNF Balanceador de Carga (2) que é responsável por distribuir o tráfego para várias funções virtualizadas. Essas funções possuem a mesma funcionalidade do serviço prestado pela vítima. Em caso de aumento de tráfego, novas réplicas são instanciadas. Diferentemente de Honeypots, a replicação do serviço não é utilizada para monitoramento.

3.3.2 Reconfiguração de Rede

Nessa estratégia, a topologia da rede é reconfigurada para evitar que o ataque cumpra seu objetivo, fornecendo rotas alternativas de roteamento através da disponibilidade de gateways variados. Ao contrário da *Reconfiguração de Serviço*, na qual recursos novos são alocados, aqui a topologia da rede é alterada. Cai (2008) propõe uma estratégia de defesa, chamada de *SCOLD*, que é composta por um coordenador e diversos *proxy* servers. Quando um ataque é identificado pelo Intrusion Detection System (IDS) da vítima, uma notificação é enviada para o SCOLD Coordenador. O Coordenador envia mensagens

para habilitação de determinados *proxy* servers. Os *proxy* servers são responsáveis por atualizar os DNS servers dos clientes legítimos. Com isso, rotas indiretas são habilitadas e todos pacotes enviados pelo cliente para a vítima passarão por algum *proxy* server. Os *proxy* servers são encarregados de analisar o tráfego e direcionar os pacotes para gateways alternativos ao principal. O fluxo malicioso que passa pelos *proxy* servers pode ser limitado ou descartado utilizando filtros ou limitadores de tráfego. Assim, o objetivo desse mecanismo de defesa pode ser alcançado aumentado a vazão da rede utilizando rotas alternativas, e filtrar ou limitar o tráfego malicioso nos *proxy* servers.

Figura 3.6: Exemplo de Rede

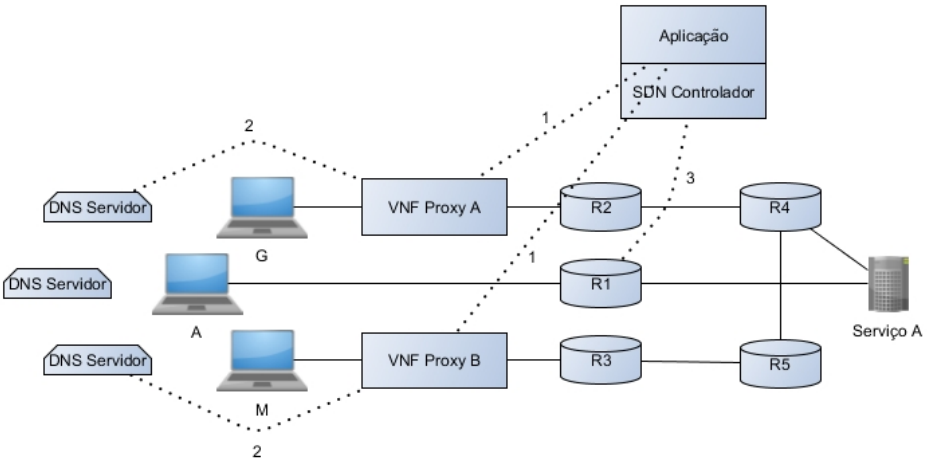


Fonte: Autor

Na Figura 3.6, é apresentado um ataque DDoS, cujo tráfego ingressa na rede através do roteador R1, o qual recebe tráfego de *G*, *A* e *M*. Sendo que G representa tráfego legítimo, A representa tráfego malicioso e M representa mistura de tráfego malicioso e legítimo. Na Figura 3.7, a ordem de execução para a estratégia implementada em redes baseadas em infraestruturas SDN/NFV é apresentada. O IDS identifica que o ataque está sendo realizado e VNFs Proxy são instanciados (1). Cada uma dessas VNFs são vinculadas a gateways alternativos. Quando os proxies são instanciados, é enviado mensagem para o servidor DNS atualizar o endereço da vítima (2). Com isso, todo o tráfego legítimo para o serviço é redirecionado para diversos gateways, criando diferentes caminhos para os tráfegos. Pode-se aplicar, ainda, filtros ou limitadores em cada VNF Proxy com o objetivo de descartar ou limitar o fluxo do tráfego malicioso destinado a vítima. Já que tráfego malicioso que foi identificado pelo IPS não foi redirecionado, todos os pacotes chegam ao mesmo gateway e, portanto, o controlador SDN pode instalar regras na tabela de encaminhamento do roteador para eliminar todos os pacotes recebidos de origem externa

(3).

Figura 3.7: Reconfiguração de Rede: Ordem de execução



Fonte: Autor

4 IMPLEMENTAÇÃO E AVALIAÇÃO

As estratégias propostas no capítulo anterior visam estabelecer mecanismos de mitigação que são ativados pela aplicação que detecta o ataque e identifica assinaturas de possíveis atacantes. A partir dessas estratégias, foi realizada a implementação de um protótipo. O objetivo da implementação do protótipo é utilizá-lo na execução de experimentos, avaliar sua capacidade de manter a continuidade do serviço no decorrer dos ataques DDoS e avaliar o tempo de resposta relacionado à estratégia de mitigação.

A Seção 4.1 apresenta detalhes de desenvolvimento do protótipo, assim como detalhes do ambiente de emulação utilizado. Por fim, a Seção 4.2 descreve os experimentos realizados para a avaliação do protótipo e analisa os resultados obtidos.

4.1 Implementação do Protótipo

Para avaliar as propostas apresentadas nesse trabalho, estratégias que utilizam técnicas de mitigação distintas foram implementadas: *Throttling* e *Firewalls Cooperativos*.

A Seção 4.1.1 apresenta detalhes do ambiente de emulação utilizado. A Seção 4.1.2 apresenta detalhes de implementação da estratégia *Throttling*. Na Seção 4.1.3, são descritos os detalhes presentes na implementação da estratégia *Firewalls Cooperativos*.

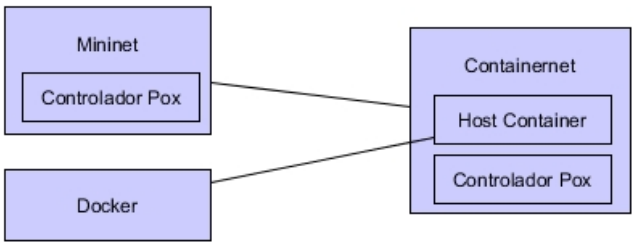
4.1.1 Ambiente de Emulação

Para a construção do protótipo avaliado, o ambiente de emulação foi contruído utilizando as ferramentas: *Mininet*, *Docker* e *Containernet*. Na Figura 4.1, a estruturação do ambiente de emulação é apresentada. *Mininet* [1] é uma ferramenta de emulação que possibilita a criação de uma rede virtual OpenFlow - com controlador, switches, hospedeiros e links - em uma única máquina real ou máquina virtual. *Docker* [2] é uma aplicação que permite a criação de containers que adicionam um software a um ambiente Linux que possui tudo que precisa para ser executado. Como os containers são construídos sobre o sistema Linux, qualquer ferramenta necessária para medição de consumo de CPU, ou até mesmo para verificação de tráfego recebido em determinada interface pode ser instalada. Isso garante que o software sempre será executado da mesma forma, independentemente

[1] http://mininet.org/
[2] https://www.docker.com/

do seu ambiente. O *Containernet* [3] é uma extensão do Mininet que possibilita instanciar containers Dockers em hospedeiros. Com isso, as VNFs foram implementadas em containers Docker e Controlador SDN utilizado foi o *Controlador POX* [4].

Figura 4.1: Ferramentas do Ambiente de Emulação



Fonte: Autor

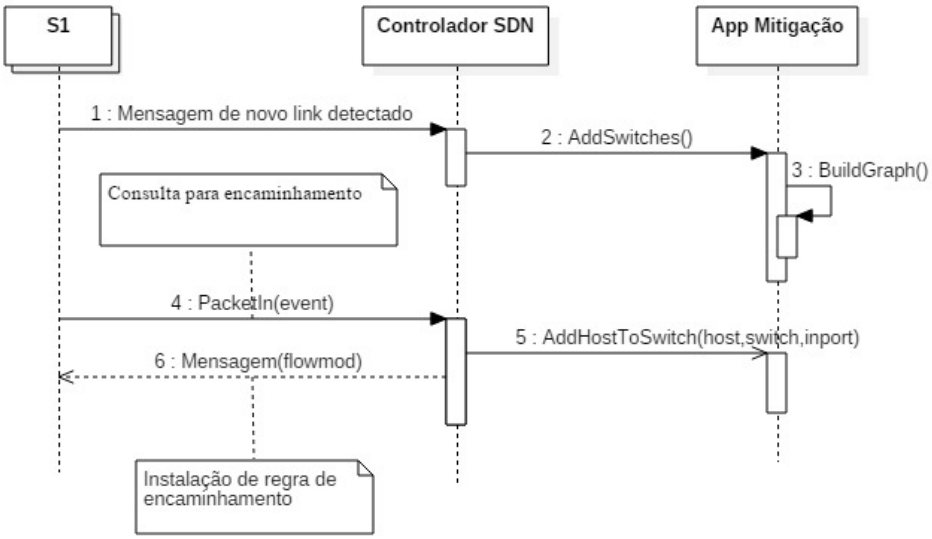
4.1.2 Estratégia de Throttling

A implementação dessa estratégia requer a modificação da tabela de encaminhamento de dispositivos a uma distância de k saltos da vítima. Para isso, é necessário o conhecimento de toda a topologia da rede para cálculo da distância. Na Figura 4.2, o diagrama de sequência referente a tarefa de construção do grafo da topologia é apresentado. Essa tarefa é realizada utilizando o componente *openflow.discovery* em conjunto com *host_tracker*. O *openflow.discovery* gera mensagens Link Layer Discovery Protocol (LLDP) para todo link removido e inserido na rede. Já o *host_tracker* permite o mapeamento de MAC/IP de hosts e o conhecimento de qual dispositivo de encaminhamento o host está conectado. Ambos componentes são apresentados como componentes internos ao controlador SDN apresentado na Figura 4.2. Na inicialização do controlador e dos switches Open vSwitch, cada switch envia uma mensagem de reconhecimento para o Controlador SDN identificando os link que ele possui conhecimento (1). O Controlador SDN sinaliza para a aplicação que realiza a mitigação a necessidade de inserção dos switches no grafo que representa a topologia da rede (2). Caso seja o primeiro link detectado, a aplicação instancia a estrutura de dados referente à rede (3). Como cada pacote que o switch não sabe encaminhar inicialmente é submetido ao Controlador SDN (4), é possível saber a qual switch o host que enviou o pacote está conectado. Com isso, O Controlador SDN notifica a aplicação que realiza a mitigação da inserção do host no grafo (5). En-

[3] https://github.com/mpeuster/containernet
[4] https://github.com/noxrepo/pox

quanto isso, o Controlador SDN pode atualizar a tabela de encaminhamento do switch (6).

Figura 4.2: Diagrama de Sequência: Construção da Topologia



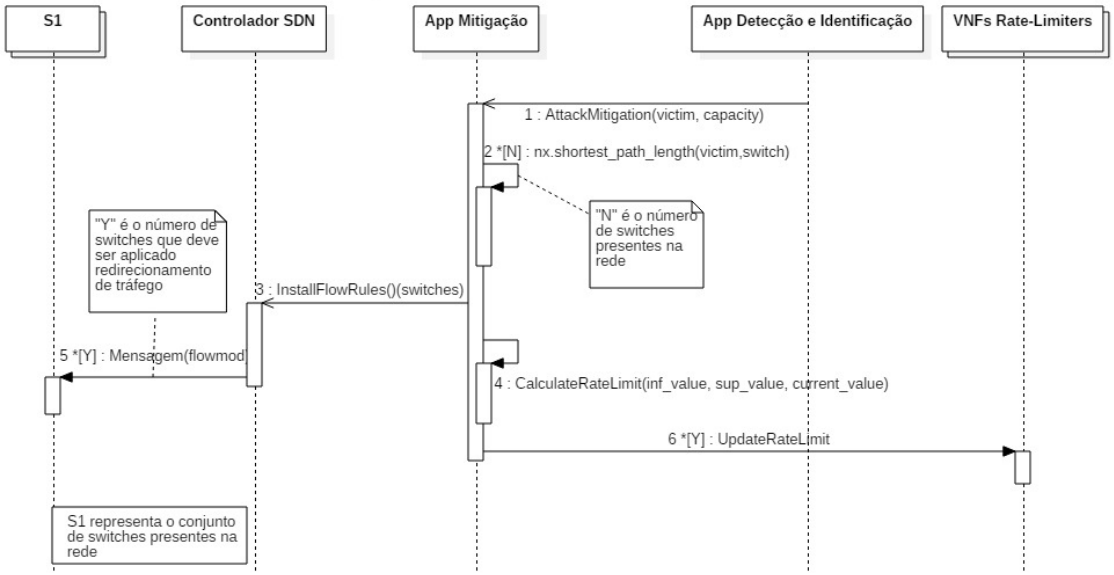
Fonte: Autor

Para a representação da rede e cálculo de distância entre dois nodos foi utililizado o pacote *NetworkX*. Na Figura 4.3, o diagrama de sequência para a instalação das regras de encaminhamento nos switches a uma distância de k saltos é apresentado. Após a detecção e identificação das assinaturas (1), a aplicação que realiza a mitigação calcula o menor caminho de distância entre a vítima e cada nodo do grafo, isto é, cada switch da rede (2). A aplicação informa ao Controlador SDN a quais switches devem ser aplicados o redirecionamento de tráfego (3). De forma paralela, a taxa de limitação é calculada com base nos valores de utilização da vítima que são provenientes da aplicação de detecção (4) e as tabelas de encaminhamento dos switches calculados são atualizadas (5). Por fim, mensagens contendo a taxa de limitação são enviadas para todas as VNFs necessárias (6).

Para a implementação do Rate-Limiter foram estudadas três diferentes formas:

1. *Filas no roteador*: são criadas diversas filas, que são vinculadas a portas específicas, com diferentes prioridades. Não foi encontrada documentação sobre mudança de taxas de limitação por meio de mensagens Openflow utilizando controlador POX.
2. *Meter Table no roteador*: uma *meter* mede a taxa dos pacotes atribuídos a ele e habilita o controle da taxa desses pacotes. A utilização de meters possibilita a implementação de Rate-Limiting dinâmico direto no dispositivo de encaminhamento, visto que é possível configurar a tabela com mensagens Openflow 1.3 (NADA, a).
3. *Função Virtualizada de Rede*: redirecionamento do tráfego para VNFs responsáveis

Figura 4.3: Diagrama de Sequência: Instalação de Regras de Encaminhamento



Fonte: Autor

por limitar o tráfego. Essa é a forma implementada devido à incompatibilidade do controlador POX com Openflow versão 1.3.

As regras de limitação da VNF são configuradas utilizando SSH e iptables (com parâmetros de *–hashlimit*). Com base nas utilizações obtidas da aplicação de monitoramento e detecção, uma nova taxa de limitação é estabelecida em todas as VNFs. Segundo Yau et al. (2005), dois algoritmos para o cálculo da taxa de limitação existem. Um deles distribui a taxa de limitação para todos os dispositivos de encaminhamento de distância k de forma igual e o outro envia uma taxa que deve ser somada a cada *throttle* com o objetivo de penalizar somente os dispositivos de encaminhamento de distância k que apresentam comportamento anormal.

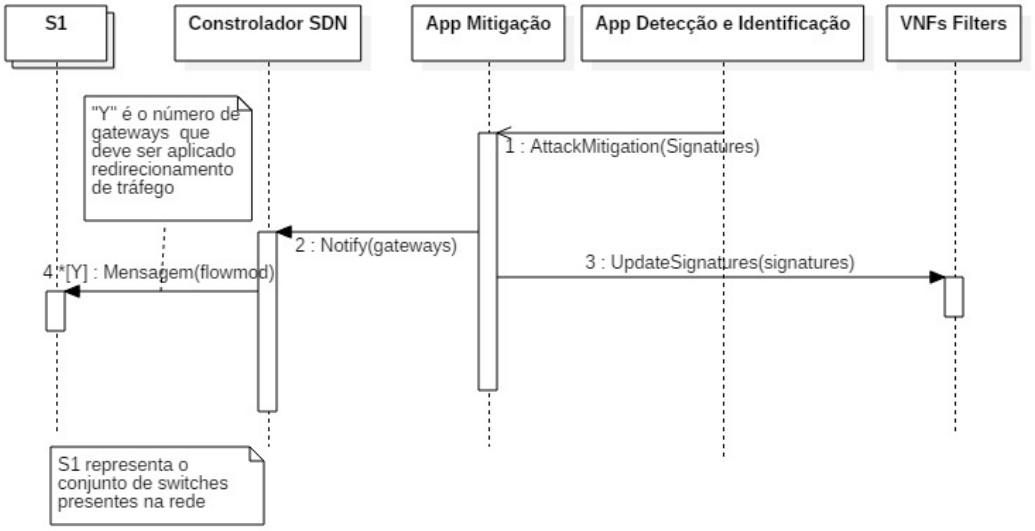
4.1.3 Estratégia de Firewalls Cooperativos

A implementação dessa estratégia requer alteração da flowtable de todos dispositivos de encaminhamento de borda. A Figura 4.4 apresenta o diagrama de sequência da inicialização do mecanismo de mitigação utilizado para essa estratégia. A aplicação responsável por realizar a detecção e identificação do ataque fornece uma lista de IPs de origem que devem ser filtrados (1). Essa lista é alimentada frequentemente pelo algoritmo de detecção. A aplicação sinaliza ao Controlador SDN que as tabelas de encaminhamento dos gateways precisam ser modificadas (2). O Controlador SDN envia mensagens para

atualização das flowtables (4). Por fim, a aplicação responsável por realizar a mitigação atualiza as VNFs Filter com as assinaturas geradas pela aplicação de detecção e identificação (3). A atualização das assinaturas de filtragem é realizada frequentemente. Com base nas utilizações obtidas da aplicação de detecção e identificação, novas regras de filtragem podem ser geradas e instaladas em todas as VNFs.

Nos experimentos realizados, a identificação do ataque é simulada por um trigger disparado após determinado tempo. Quando o trigger é disparado, as tabelas de encaminhamento dos gateways são alteradas para o redirecionamento do tráfego. Os gateways são predeterminados na execução do controlador. Com isso, é instanciada uma nova VNF para cada gateway. Após o disparo do trigger, uma função que instala as regras de filtragem em cada VNF é iniciada. As regras de filtragem de cada VNF são configuradas utilizando SSH e iptables.

Figura 4.4: Diagrama de Sequência: Inicialização do Mecanismo de Mitigação



Fonte: Autor

4.2 Avaliação Experimental

Os experimentos para avaliação do funcionamento das estratégias implementadas foram realizados através de simulações de ataques DDoS, utilizando a ferramenta *hping3* [5]. Os ataques efetuados na simulação foram ataques DDoS de inundação ICMP REQUEST com IP spoofing. Todos os pacotes dos atacantes são destinados à vítima,

[5]https://linux.die.net/man/8/hping3

porém como todo fluxo que o dispositivo de encaminhamento não sabe encaminhar é enviado para o controlador, acontece um ataque DDoS no controlador SDN. Os dados de tráfego, gerados durante as simulações, são obtidos acessando as tabelas de encaminhamento de cada dispositivo e acessando as regras de filtragem presentes em cada VNF. Os experimentos utilizaram uma combinação entre tráfego malicioso e não malicioso para verificação da continuidade do fornecimento do serviço após o início da mitigação. Para a experimentação da estratégia de *Firewalls Cooperativos*, a avaliação do tempo de resposta em relação ao tempo de detecção foi analisado. Já na estratégia de *Throttling* foi analisado o cálculo implementado da taxa de limitação, assim como a variação da relação entre pacotes não maliciosos e pacotes maliciosos.

Na Seção 4.2.1, a topologia da rede utilizada para o experimento é apresentada. A Seção 4.2.2 analisa os resultados obtidos.

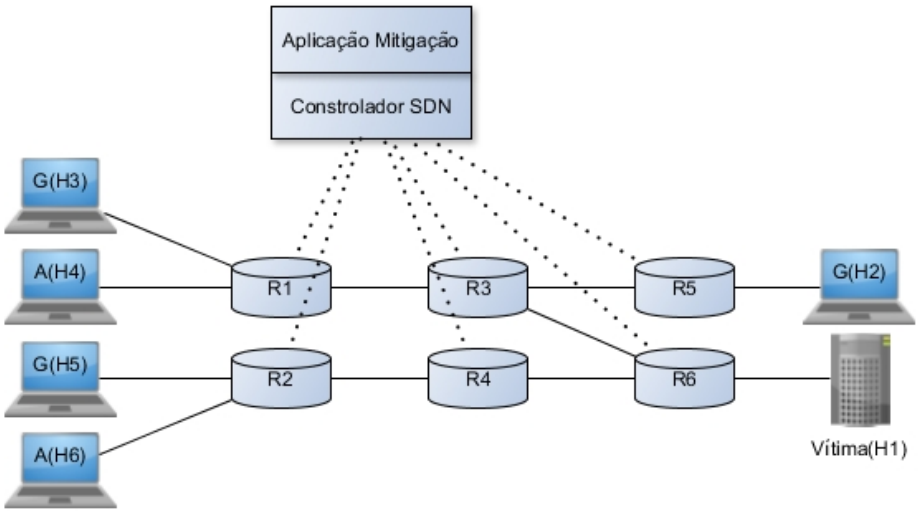
4.2.1 Topologia da rede avaliada

A Figura 4.5 apresenta a topologia utililizada para a realização dos experimentos desse trabalho. Uma rede com 6 dispositivos de encaminhamento e 6 hospedeiros é criada. As máquinas H4 e H6 conectadas aos roteadores R1 e R2, respectivamente, são responsáveis pela realização do ataque à vítima. As duas máquinas realizam o ataque com endereços IP de origem modificados e não apresentam limite de taxa para geração dos pacotes. Os hospedeiros H3, H5 e H2 realizam requisições simultaneamente com o intuito de simular o acesso ao serviço da vítima por parte de usuários não maliciosos. A escolha da topologia descrita é baseada na redundância de caminhos, uma vez que é um mecanismo muito empregado para tolerância a falhas (MENDES, 2010), na possibilidade de experimentação de diferentes valores de k na estratégia de *Throttling* e na possibilidade de utilização de filtros cujas assinaturas podem ser instaladas em mais de uma VNF, visto que existem mais de um gateway.

4.2.2 Análise dos Resultados

Nos experimentos relacionadas à implementação de *Firewalls Cooperativos*, os filtros foram aplicados nos dispositivos de encaminhamento R1 e R2 da rede apresentada na Figura 4.5. Após o início da efetuação do ataque, existe uma sobrecarga no dispositivo

Figura 4.5: Topologia da rede avaliada



Fonte: Autor

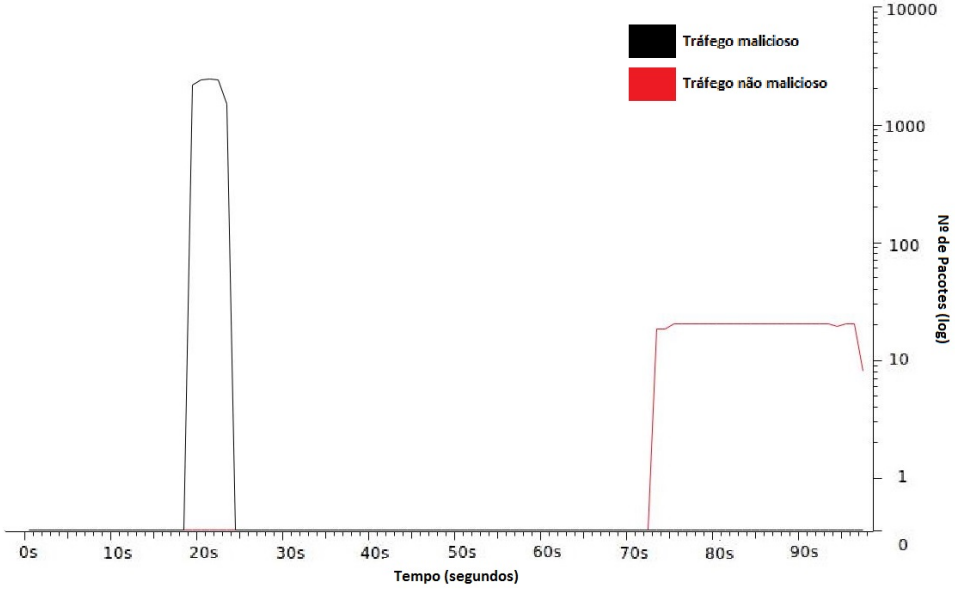
R6. Porém, como cada fluxo que o dispositivo não sabe encaminhar é direcionado ao controlador, acontece um ataque DDoS ao controlador.

Três experimentos diferentes foram realizados. Cada qual variando o tempo de identificação do ataque DDoS, e consequentemente, do início do mecanismo de mitigação. O tráfego presente no link R6-Vítima com variações de tempo para detecção do ataque de 5 segundos, 10 segundos e 15 segundos são apresentados nas Figuras 4.6, 4.7 e 4.8, respectivamente. Primeiramente, é possível observar que aumentando o tempo para a detecção do ataque, o tempo para a resposta da rede também aumenta. Como apresentado na Figura 4.10, se o algoritmo de detecção demora 5 segundos para habilitar o algoritmo de mitigação, a rede demora 49 segundos para normalizar a prestação do serviço. Se o algoritmo de detecção demora 10 segundos para a habilitação da estratégia de mitigação, o tempo de resposta da rede aumenta para 120 segundos.

O atraso da resposta da rede é devido ao excesso de carga no controlador. Durante esse período a única medida adotada no sistema é a utilização de filtros na borda da rede para impossibilitar que o tráfego malicioso entre na rede. Porém, os hospedeiros só poderão acessar o serviço novamente após a normalização do funcionamento do controlador. Isso pode ser observado analisando o desempenho dos filtros, uma vez que o tráfego não malicioso ingressa na rede, como apresentado na Figura 4.9. Porém, a resposta não é obtida pelos hosts. Por isso, quanto maior o tempo de detecção do ataque, maior é o número de pacotes maliciosos que sobrecarregam a rede e o controlador.

A implementação da estratégia de *Throttling* é construída baseando-se na verifi-

Figura 4.6: Detecção 5 segundos: tráfego na interface R6-Vítima

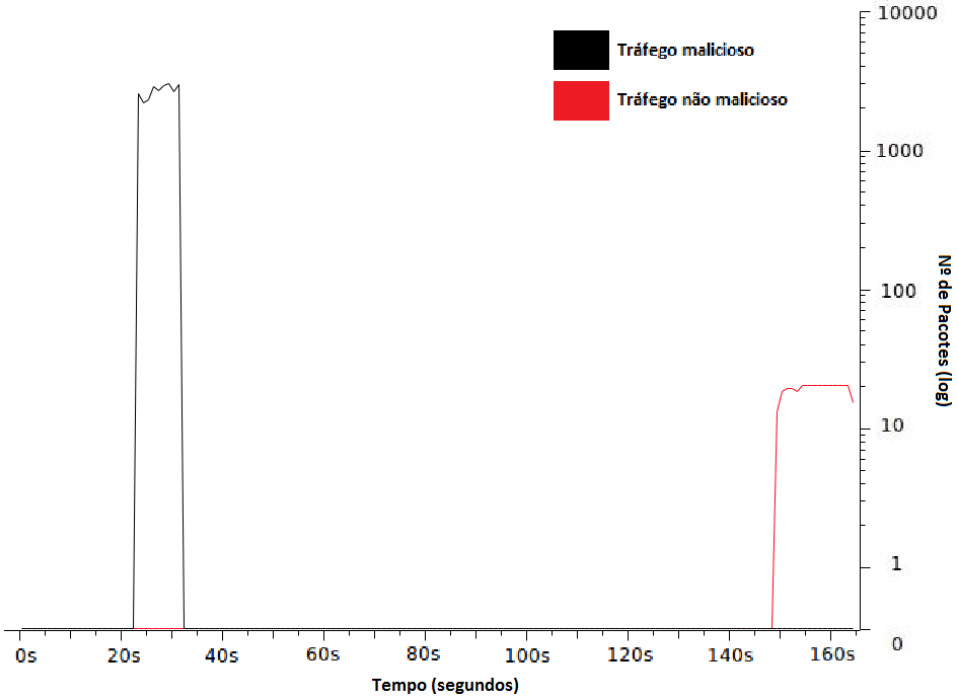


Fonte: Autor

cação constante da utilização dos hosts. Cada host possui um intervalo de utilização [L_s pps , U_s pps] que garante o funcionamento sem perda de serviço. Baseado na utilização de cada host que é fornecida pela aplicação de identificação, a aplicação responsável por realizar a mitigação calcula a taxa de limitação (T_L). Para o cálculo da taxa de limitação duas constantes são utilizadas: α, que representa o fator redutivo e β, que representa o fator aditivo (YAU et al., 2005). Com isso, quando a utilização da vítima está superior ao seu limite máximo (U_s), a taxa de limitação é multiplicada pelo fator de limitação α e quando a utilização da vítima está inferior ao seu limite mínimo (L_s), a taxa de limitação é multiplicada pelo fator β e somada à taxa atual ($T_L = T_L + (T_L * \beta)$). Já nos experimentos relacionados a essa estratégia, os valores que representam as utilizações da vítima foram simulados utilizando valores pré-configurados.

Partindo do princípio que o intervalo de utilização da vítima para o funcionamento sem perda de serviço seja de [310 pps, 345 pps], e que a cada iteração da aplicação de identificação os valores de utilizações da vítima possuem os valores de 250, 350, 300 e 343 pacotes por segundo. As constantes para decremento, α, e incremento, β, da taxa de limitação apresentam valores de 0.5 e 0.4, respectivamente. O tempo de recálculo da taxa de limitação é de 25 segundos. Como a utilização da vítima a cada iteração da aplicação de detecção é simulada, a velocidade de convergência não foi avaliada. Nos experimentos relacionados à implementação de *Throttling*, a mesma sobrecarga no controlador descrita anteriormente acontece. Porém, com base na topologia da rede, o controlador SDN cal-

Figura 4.7: Detecção 10 segundos: tráfego na interface R6-Vítima

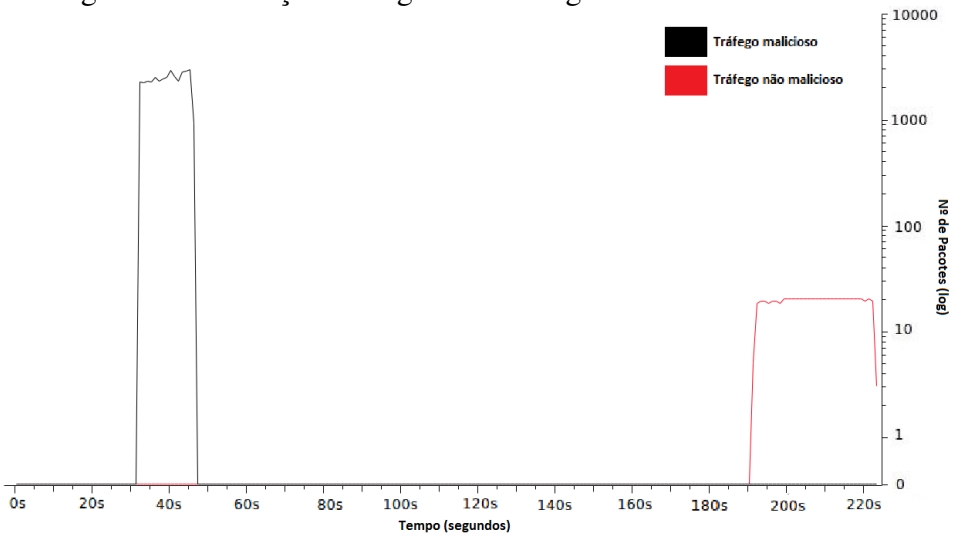


Fonte: Autor

cula os dispositivos que devem redirecionar o tráfego para a aplicação da limitação de tráfego. Na topologia da Figura 4.5, esses dispositivos são os roteadores R1, R2 e R5. Na Figura 4.11, é possível observar a variação da taxa de limitação no decorrer do ataque. Por volta dos 25 segundos, o ataque sobrecarrega o controlador e a mitigação é iniciada. Quando o mecanismo de mitigação é ativado, o valor de utilização (250 pps) é enviado para vítima. Como o valor não está no intervalo [310 pps, 345 pps], a taxa de limitação é multiplicada por α (250 * 0.5 = 125 pps). A próxima iteração do mecanismo de detecção sinaliza ao mecanismo de mitigação a quantidade de pacotes (350) por segundo recebidos pela vítima. Como o valor é inferior ao limite mínimo presente no intervalo [310 pps, 345 pps], a constante β é somada a taxa de limitação (125+(125*0,4) = 175 pps). A próxima utilização da vítima recebida (300) não está contida no intervalo [310 pps, 345 pps], sendo assim a nova taxa de limitação é de 87,5 pps. A próxima utilização recebida permite a soma da constante aditiva, fazendo a próxima iteração estabilizar a variação da taxa de limitação.

Por fim, uma determinada quantidade de pacotes não maliciosos é enviada para a vítima. Três diferentes experimentos foram realizados. Com base na quantidade de pacotes ICMP REPLY recebidos pelos clientes, foi estabelecida a comparação apresentada na Figura 4.12. O objetivo de ambos experimentos é medir o impacto da taxa de geração dos pacotes não maliciosos no serviço disponível na estratégia de *Throttling*, e

Figura 4.8: Detecção 15 segundos: tráfego na interface R6-Vítima



Fonte: Autor

Figura 4.9: Utilização dos Filtros S1 e S2

```
 Filtro S1
 560 47040 ACCEPT     all  --  any    any     10.0.0.3             anywhere
   0     0 ACCEPT     all  --  any    any     10.0.0.5             anywhere
323K 9037K DROP       all  --  any    any     anywhere             anywhere

 Filtro S2
   0     0 ACCEPT     all  --  any    any     10.0.0.3             anywhere
 676 56784 ACCEPT     all  --  any    any     10.0.0.5             anywhere
301K 8419K DROP       all  --  any    any     anywhere             anywhere
```

Fonte: Autor

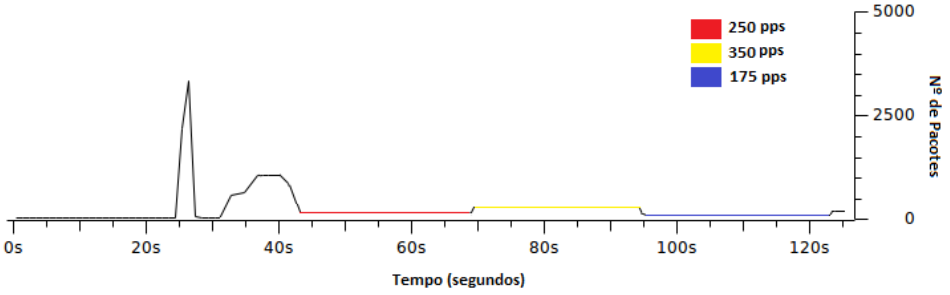
comparar o desempenho obtido com a estratégia de *Firewalls Cooperativos*. Em ambos os experimentos, 1000 mensagens de ICMP REQUEST são geradas pelos clientes. A taxa de geração dos pacotes não maliciosos no primeiro experimento é de 100 pacotes por segundo, no segundo é de 1000 pacotes por segundo e no terceiro é de 10000 pacotes por segundo. A estratégia de *Throttling* não diferencia tráfego malicioso (T_M) de tráfego não malicioso(T_{N_M}), com isso tanto tráfego malicioso pode ser encaminhado para a vítima quanto tráfego não malicioso pode ser descartado pelo Rate-Limiter. Analisando a Figura 4.12 é possível observar que quanto maior a relação T_{N_M} / T_M, maior será a probabilidade da VNF Rate-Limiter não descartar os pacotes não maliciosos. O que não acontece na estratégia de *Firewalls Cooperativos*, uma vez que a aplicação de detecção gera assinaturas que são instaladas nos filtros. Já que as assinaturas não variam nos experimentos, os filtros descartam todos os pacotes maliciosos, gerando assim um melhor desempenho.

Figura 4.10: Atraso associado ao algoritmo de detecção

Tempo para Detecção	Tempo de resposta da rede
5 segundos	49 segundos
10 segundos	120 segundos
15 segundos	145 segundos

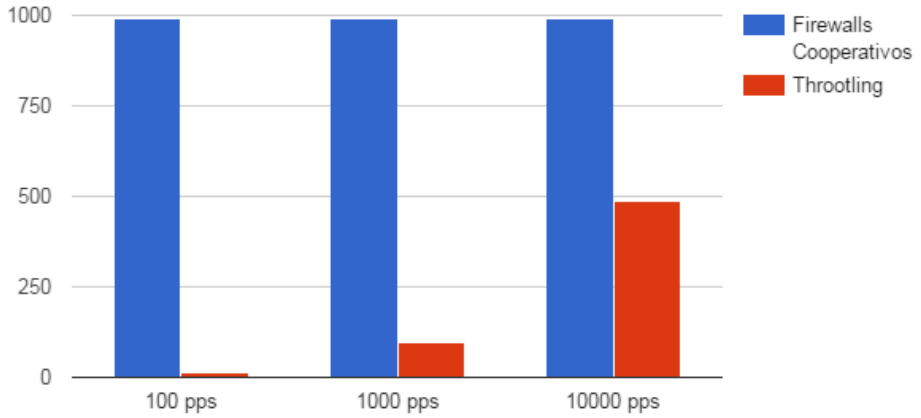
Fonte: Autor

Figura 4.11: Impacto da adaptação da taxa de limitação verifaca na interface R6-Vítima



Fonte: Autor

Figura 4.12: Comparação da quantidade de pacotes ICMP REPLY recebidos



Fonte: Autor

5 TRABALHOS RELACIONADOS

Para a apresentação dos trabalhos relacionados, a pesquisa foi dividida em três diferentes áreas de estudo, portanto trabalhos sobre mitigação em redes tradicionais são abordados na Seção 5.1, trabalhos sobre mitigação em redes baseadas em infraestruturas SDN são introduzidos na Seção 5.2 e trabalhos sobre mitigação em redes baseadas em infraestruturas NFV são tratados na Seção 5.3.

5.1 Mitigação em Redes Tradicionais

Em (GARG, 2011), são apresentadas técnicas de mitigação para ataques DDoS comumente encontrados na literatura. Técnicas de identificação de pacotes maliciosos como *HCF probabilístico* e *HCF simples* e de mitigação de ataque DDoS como *Source Router Preferential dropping*, *Pushback* e *Path Fingerprint* são analisadas em relação à tempo para identificação, quantidade de pacotes maliciosos identificados e descartados. Já Rai e Challa (2016) classificam as técnicas de mitigação dividindo-as em relação aos três níveis do modelo OSI em que os ataques podem ocorrer: 3 (Rede), 4 (Transporte) e 7 (Aplicação). Rai e Challa (2016) exibem as limitações e vulnerabilidades dos mecanismos de defesa devido à dificuldade de identificação da ocorrência de um ataque DDoS. Douligeris e Mitrokotsa (2004) apresentam uma classificação de ataques DDoS e de mecanismos de defesa. Os mecanismos de defesa apresentados são classificados conforme a atividade, isto é, se a técnica é utilizada para prevenir, detectar, responder ou tolerar o ataque e continuar provendo o serviço, e conforme a localização onde o mecanismo de defesa é executado. Com isso, vantagens e desvantagens de cada técnica de defesa são analisadas. As classificações abordadas por Mirkovic e Reiher (2004) se assemelham às categorias propostas por Douligeris e Mitrokotsa (2004), mas diferem-se por agrupar as classificações de detecção, resposta e tolerância na mesma atividade denominada *reativa*. Em Ioannidis e Bellovin (2002), um mecanismo de defesa utilizando a arquitetura de *pushback* é implementado. A técnica de mitigação consiste nos roteadores identificarem o congestionamento e limitarem a quantidade de pacotes repassados para a interface de saída. Se os pacotes forem descartados, eles são redirecionados para o daemon, que coordena os padrões de configuração de cada roteador em relação à limitação dos pacotes encaminhados.

5.2 Mitigação em Redes SDN

Em (SAHAY et al., 2015), as vantagens de se usar SDN para mitigar ataques DDoS são evidenciadas ao ser proposta uma ferramenta que possibilita os usuários terem acesso aos serviços de mitigação DDoS providos por ISPs. Quando os usuários pedem acesso aos serviços, ISPs podem mudar a identificação do tráfego anômalo e redirecionar esse tráfego para dispositivos físicos dedicados, enquanto a análise é realizada no usuário. Yan e Yu (2015) abordam a relação entre ataques distribuídos de negação de serviço com SDN, discorre sobre os benefícios adquiridos pelo uso de SDN em mitigação de ataques DDoS e sumariza os novos ataques DDoS introduzidos pelo uso desse conceito e suas possíveis soluções. Em (Y MR.JUSTIN GOPINATH M.TECH, 2015), é apresentada a implementação de uma técnica de mitigação que consiste em identificar e filtrar os pacotes maliciosos na camada de aplicação e também é implementado controle de carga usando algoritmo baseado em round-robin. A eficácia de ambas as implementações é analisada. Dao et al. (2015) apresenta uma técnica de filtragem baseada no endereço IP origem para mitigar ataques DDoS utilizando SDN. Marcas são adicionadas aos pacotes que são identificados como maliciosos. Essa identificação é realizada comparando o número de conexões estabelecidas com aquele endereço e o número de pacotes enviados por conexão com valores limites já estabelecidos. Wang et al. (2015) propõe o sistema de mitigação denominado *DaMask* que é baseado em dois módulos: DaMask-D, responsável por realizar um identificação estatística do ataque, e DaMask-M, responsável por responder ao ataque, geralmente, com descartes dos pacotes pertencentes ao ataque.

5.3 Mitigação em Redes NFV

Fung e McCormick (2015) propõem o uso de NFV para mitigar um ataque DDoS através do direcionamento do tráfego de entrada em dois túneis com políticas distintas. O desempenho e a robustez da proposta em relação à política de tráfego estática e dinâmica são analisadas.

(ALLIANCE, 2016) descreve brevemente a implementação da técnica de mitigação de filtragem em redes SDN/NFV para ataques de inundação. Fayaz et al. (2015) identifica novas oportunidades para melhorar os mecanismos de defesa atuais contra ataques DDoS utilizando SDN/NFV e implementa um sistema de gerência elástico e flexível de defesa contra esses ataques demonstrando os benefícios do uso desses dois conceitos

no contexto de mitigação de ataques DDoS. O mecanismo de gerência de defesa proposto é composto por três elementos: a estratégia de detecção de tráfego anômalo, o gerenciador de recursos e a orquestração da rede. Após identificado que a rede está sofrendo um ataque DDoS, estimativas de volume de cada ataque são passadas para o gerenciador de recursos. O gerenciador de recursos possui uma biblioteca de políticas de defesas que configura quais ações devem ser executadas e a ordem em que elas devem ser chamdas. Ao receber os detalhes do tráfego, ele encaminha para a orquestração da rede as ações que devem ser tomadas, ou seja, quais máquinas virtuais devem ser acionadas e quando. O objetivo de Fayaz et al. (2015) não é avaliar técnicas de mitigação para esses ataques, mas sim criar o sistema de gerenciamento de defesa que utiliza essas técnicas de mitigação. Esse trabalho procura justamente abordar e implementar técnicas de mitigação que posam ser utilizadas em um ambiente similar ao usado por Fayaz et al. (2015).

6 CONCLUSÕES E TRABALHOS FUTUROS

A utilização de redes baseadas em infraestruturas SDN/NFV possibilita a elaboração de estratégias de mitigação mais complexas e eficientes unindo os benefícios de facilidade de monitoramento e gerência, presentes em SDN, com a possibilidade de instanciação de funções de rede conforme o tráfego, presente em NFV. O trabalho investiga e propõe estratégias de mitigação DDoS em redes baseadas em SDN e NFV. As estratégias propostas visam utilizar a elasticidade, proveniente de NFV, e a flexibilidade para redirecionamento de tráfego, proveniente de SDN. Por isso, é de extrema importância o estudo de como diferentes estratégias de mitigação podem ser implementadas e utilizadas para a construção de mecanismos de defesa mais robustos e eficientes.

Na Seção 6.1, são apresentadas as contribuições deste trabalho. As propostas para trabalhos futuros são descritas na Seção 6.2.

6.1 Resumo de Contribuições

Para a realização deste trabalho, uma revisão na literatura para entender e enumerar os diferentes tipos de ataques DDoS foi realizada. Com o intuito de estruturar o estudo e a apresentação das estratégias de mitigação propostas, uma taxonomia que ilustra as principais técnicas e estratégias para mitigação existentes foi desenvolvida. A taxonomia fundamenta uma divisão das estratégias de mitigação conforme as possíveis técnicas de mitigação de *Rate-Limiting*, *Filtering* e *Reconfiguração*. A ferramenta utilizada para a simulação de um ataque DDoS foi a *hping3*, a qual é capaz de realizar *IP Spoofing* e possibilitar a alteração de protocolos de ataque. Foi contruído um protótipo para validação das propostas apresentadas. O protótipo apresenta a implementação das estratégias de *Throttling* e *Firewalls Cooperativos*, e foi implementado em um ambiente que possibilita a construção de redes virtuais, denominado *Mininet*. As funções de rede virtualizadas foram implementadas através de containers *Docker* e a integração das funções de rede à rede virtual foi realizada através da plataforma *Containernet*. A estratégia de *Throttling* implementa a descoberta da topologia com o intuito de aplicar regras de redirecionamento de tráfego para funções virtualizadas responsáveis pelo descarte de uma quantidade de pacotes destinados a vítima. Já a estratégia de *Firewalls Cooperativos* implementa filtros na borda da rede. As regras de filtragem presentes nos filtros são atualizadas pela aplicação executada nas camadas superiores ao controlador SDN. Com isso, a cooperatividade entre

os filtros é garantida através do IPS executado na aplicação.

As estratégias foram avaliadas através de experimentos que simulam a execução de um ataque DDoS de inundação ICMP REQUEST. Como nenhum algoritmo de detecção foi implementado, tanto as regras de limitação de tráfego quando as regras de filtragem variam conforme dados pré-configurados. Os resultados dos experimentos demonstram que o tempo para normalização da rede está diretamente associado ao tempo necessário para ativação do mecanismo de mitigação. Os resultados também demonstram que as estratégias propostas são capazes de mitigar ataques DDoS e, no caso de *Firewalls Co-operativos*, de possibilitar a continuidade do fornecimento do serviço para usuários não maliciosos.

6.2 Trabalhos Futuros

Como trabalhos futuros, propomos a avaliação das estratégias restantes propostas nesse trabalho e a investigação da possibilidade de introdução de novos ataques direcionados especificamente a redes baseadas em SDN/NFV. Outro ponto interessante a ser pesquisado e implementado é a utilização de taxas de limitação aplicadas diretamente nos dispositivos de encaminhamento ao invés de redirecionar para uma VNF particular. Propomos a implementação de um sistema de orquestração de mecanismos de defesa, o qual possibilite a utilização de mecanismos para ataques específicos como é proposto por Fayaz et al. (2015). Propomos também a integração das estratégias de mitigação a mecanismos de detecção de ataques já existentes, como aquele implementado por Silva et al. (2016).

REFERÊNCIAS

CLOUD Security Alliance.

ALLIANCE, C. S. **SECURITY POSITION PAPER Network Function Virtualization**. 2016. Unpublished thesis.

CAI, Y. A ddos defense mechanism with topology reconfiguration. **Journal of Engineering; Computing and Architecture**, Citeseer, v. 23, p. 65–78, 2008.

CHEN, L.-C.; LONGSTAFF, T. A.; CARLEY, K. M. Characterization of defense mechanisms against distributed denial of service attacks. **Computers & Security**, Elsevier, v. 23, n. 8, p. 665–678, 2004.

DAO, N.-N. et al. A feasible method to combat against ddos attack in sdn network. In: **2015 International Conference on Information Networking (ICOIN)**. [S.l.: s.n.], 2015. p. 309–311. ISSN 1550-445X.

DOULIGERIS, C.; MITROKOTSA, A. {DDoS} attacks and defense mechanisms: classification and state-of-the-art. **Computer Networks**, v. 44, n. 5, p. 643 – 666, 2004. ISSN 1389-1286.

EL-SOUDANI, M. M. S.; EISSA, M. A. Cooperative defense firewall protocol. In: ____. **Security and Privacy in the Age of Uncertainty: IFIP TC11 18th International Conference on Information Security (SEC2003) May 26–28, 2003, Athens, Greece**. Boston, MA: Springer US, 2003. p. 373–384. ISBN 978-0-387-35691-4.

ETSI. **Network Functions Virtualisation Use Cases**. [S.l.], 2013. 1–50 p.

ETSI, N. Network functions virtualisation: An introduction, benefits, enablers, challenges and call for action. issue 1. 2012.

ETSI, N. Network functions virtualisation: Network operator perspectives on industry progress. 2013.

FAYAZ, S. K. et al. Bohatei: Flexible and elastic ddos defense. **CoRR**, abs/1506.08501, 2015.

FUNG, C. J.; MCCORMICK, B. Vguard: A distributed denial of service attack mitigation method using network function virtualization. In: TORTONESI, M. et al. (Ed.). **CNSM**. [S.l.]: IEEE Computer Society, 2015. p. 64–70. ISBN 978-3-9018-8277-7.

GARG, D. Ddos mitigation techniques-a survey. 2011.

HACHEM, N. et al. Botnets: lifecycle and taxonomy. In: IEEE. **Network and Information Systems Security (SAR-SSI), 2011 Conference on**. [S.l.], 2011. p. 1–8.

HARRIS BRYAN KONIKOFF, E.; PETERSEN, P. Breaking the ddos attack chain. CMU-ISR-MITS-2, Pittsburgh, PA 15213, 2013.

HUTCHINS, E. M.; CLOPPERT, M. J.; AMIN, R. M. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. **Leading Issues in Information Warfare & Security Research**, v. 1, p. 80, 2011.

IDEAS, A. N. G. **Digital Attack Map Top daily DDoS attacks worldwide**. 2016. Available online at <http://www.digitalattackmap.com/#anim=1&color=0&country=ALL&list=0&time=17066&view=map>.

INSTITUTE, S. Naptha: A new type of denial of service attack. 2000.

IOANNIDIS, J.; BELLOVIN, S. M. Implementing pushback: Router-based defense against ddos attacks. In: **NDSS**. [S.l.: s.n.], 2002.

JOSHI, R.; SARDANA, A. **Honeypots: a new paradigm to information security**. [S.l.]: CRC Press, 2011.

LAU, F. et al. Distributed denial of service attacks. In: **Systems, Man, and Cybernetics, 2000 IEEE International Conference on**. [S.l.: s.n.], 2000. v. 3, p. 2275–2280 vol.3. ISSN 1062-922X.

MCKEOWN, N. et al. Openflow: enabling innovation in campus networks. **ACM SIGCOMM Computer Communication Review**, ACM, v. 38, n. 2, p. 69–74, 2008.

MENDES, R. A. **Redundância e Disponibilidade características no SYSTEM302**. 2010. Available online at <http://www.smar.com/PDFS/SYSTEM302/SYSTEM302_REDUNDANCIA_DISPONIBILIDADE.pdf>.

MILLETARY, J. **Citadel Trojan Malware Analysis**. 2012. [Online; Accessed: 2016-08-29]. Available from Internet: <http://botnetlegalnotice.com/citadel/files/Patel_Decl_Ex20.pdf>.

MIRKOVIC, J. **D-WARD: source-end defense against distributed denial-of-service attacks**. Thesis (PhD) — University of California Los Angeles, 2003.

MIRKOVIC, J.; REIHER, P. A taxonomy of ddos attack and ddos defense mechanisms. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 34, n. 2, p. 39–53, abr. 2004. ISSN 0146-4833.

MYERS, S. O. M. L. Guide to ddos attacks. n. 12, 2015.

ONF. **Software-Defined Networking: The New Norm for Networks**. [S.l.], 2012.

PENG, T.; LECKIE, C.; RAMAMOHANARAO, K. Survey of network-based defense mechanisms countering the dos and ddos problems. **ACM Computing Surveys (CSUR)**, ACM, v. 39, n. 1, p. 3, 2007.

RADWARE. **DDoS Handbook Radware**. 2015. [Online; Accessed: 2016-08-29]. Available from Internet: <https://security.radware.com/uploadedfiles/resources_and_content/ddos_handbook/ddos_handbook.pdf>.

RAI, A.; CHALLA, R. K. Survey on recent ddos mitigation techniques and comparative analysis. 2016.

SAHAY, R. et al. Towards autonomic DDoS mitigation using Software Defined Networking. In: **SENT 2015 : NDSS Workshop on Security of Emerging Networking Technologies**. San Diego, Ca, United States: Internet society, 2015. p. .

SATHYANARAYANA, S. M. **Software Defined Network Defense**. Thesis (PhD) — University of Pennsylvania, 2011.

SHAMELI-SENDI, A. et al. Taxonomy of distributed denial of service mitigation approaches for cloud computing. **J. Netw. Comput. Appl.**, Academic Press Ltd., London, UK, UK, v. 58, n. C, p. 165–179, dec. 2015. ISSN 1084-8045.

SILVA, A. S. da et al. Atlantic: A framework for anomaly traffic detection, classification, and mitigation in sdn. In: **NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium**. [S.l.: s.n.], 2016. p. 27–35.

SPECHT, S. M.; LEE, R. B. Distributed denial of service: Taxonomies of attacks, tools, and countermeasures. In: **ISCA PDCS**. [S.l.: s.n.], 2004. p. 543–550.

WANG, B. et al. Ddos attack protection in the era of cloud computing and software-defined networking. **Comput. Netw.**, Elsevier North-Holland, Inc., New York, NY, USA, v. 81, n. C, p. 308–319, abr. 2015. ISSN 1389-1286.

WEILER, N. Honeypots for distributed denial-of-service attacks. In: IEEE. **Enabling Technologies: Infrastructure for Collaborative Enterprises, 2002. WET ICE 2002. Proceedings. Eleventh IEEE International Workshops on**. [S.l.], 2002. p. 109–114.

WYSOPAL, C.; ENG, C. Static detection of application backdoors. **Black Hat**, 2007.

Y MR.JUSTIN GOPINATH M.TECH, G. N. Ddos mitigation using software defined network. 2015.

YAN, J.; EARLY, S.; ANDERSON, R. The xenoservice-a distributed defeat for distributed denial of service. In: **Proceedings of ISW**. [S.l.: s.n.], 2000. v. 2000.

YAN, Q.; YU, F. R. Distributed denial of service attacks in software-defined networking with cloud computing. **IEEE Communications Magazine**, v. 53, n. 4, p. 52–59, 2015.

YAU, D. K. Y. et al. Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles. **IEEE/ACM Trans. Netw.**, IEEE Press, Piscataway, NJ, USA, v. 13, n. 1, p. 29–42, feb. 2005. ISSN 1063-6692.

YU, S. et al. Can we beat ddos attacks in clouds? **IEEE Transactions on Parallel and Distributed Systems**, v. 25, n. 9, p. 2245–2254, Sept 2014. ISSN 1045-9219.

ZARGAR, S. T.; JOSHI, J. Dicodefense: distributed collaborative defense against ddos flooding attacks. In: **IEEE symposium on security and privacy**. [S.l.: s.n.], 2013.

ZHU, Z. et al. Botnet research survey. **2014 IEEE 38th Annual Computer Software and Applications Conference**, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 967–972, 2008. ISSN 0730-3157.