FlowFence: Um Sistema para Prevenção de Ataques de Negação de Serviço usando Controle de Banda

Andrés Felipe Murillo Piedrahita, Diogo M. F. Mattos, Lyno Henrique G. Ferraz, Otto Carlos Muniz Bandeira Duarte *

[1]Grupo de Teleinformática e Automação (GTA)
Universidade Federal do Rio de Janeiro (UFRJ)
Caixa Postal 68504 – Rio de Janeiro – RJ – Brasil

{afmurillo, menezes, lyno, otto}@gta.ufrj.br

Resumo. *Os Ataques de Negação de Serviço (DoS) imitam o tráfego legítimo, dificultando sua detecção. Defesas convencionais para esses ataques não são escaláveis, demoram a reagir ou introduzem sobrecarga a cada pacote encaminhado na rede. Este artigo propõe FlowFence, um sistema rápido e leve de prevenção de ataques de negação de serviço através de controle de banda reativo de fluxos. Os roteadores usam uma aplicação em segundo plano para monitorar a ocupação de suas interfaces de saída e detectar DoS. O sistema utiliza um controlador para coordenar ás defesas em caso de um ataque. Em condições de ataque de negação de serviço, o controlador limita a taxa de transmissão dos fluxos ao longo do caminho para prevenir o desperdiço de recursos na rede. Um protótipo do sistema foi implementado e avaliado na plataforma de testes Future Internet Testbed with Security (FITS) e os resultados comprovam que a proposta garante aos usuários o uso equitativo de recursos sem introduzir sobrecarga na rede*

Abstract. *Denial of Service Attacks (DoS) imitates legitimate traffic, making harder its detection. Conventional Defenses for these attacks are not scalable, slow to react or introduce an overhead to each routed packet. In this paper, we present FlowFence, a lightweight and fast denial of service prevention system. Switches use a daemon to monitor the mean occupation of their interfaces to detect DoS. The system uses a controller to coordinate defense in case of an attack. In DoS conditions, the controller limits the flow transmission rate along the whole path to prevent users inanition and avoid wasting networks resources. A system prototype was implemented and evaluated in the Future Internet Testbed with Security (FITS) and the results prove that the proposal guarantees fair share of network resources without adding overhead in the network.*

1. Introdução

Os ataques de negação de serviço são uma ameaça para o funcionamento da Internet. Durante os últimos anos, o volume dos maiores ataques de negação de serviço continua crescendo, chegando a ataques de 100Gb/s em 2010 [Arbor Networks, 2012]

*Este trabalho foi realizado com recursos de: CNPq, CAPES, FAPERJ FINEP e FUNTTEL

e 300Gb/s em 2013 [The Huffington Post, 2013]. Tais volumes de tráfego podem facilmente comprometer a maioria dos enlaces, roteadores e serviços da Internet. Adicionalmente, as interrupções de serviço em infraestruturas críticas como Redes Elétricas Inteligentes podem ocasionar prejuízos econômicos e humanos enormes [Wenye e Zhuo, 2013]. A base dos ataques de negação de serviço é a assimetria existente entre os recursos disponíveis para gerar requisições fora da rede atacada e a incapacidade dos servidores para atender essas requisições. Ataques sofisticados aproveitam esses recursos para gerar inundações com tráfico muito similar ao tráfico legítimo, o que dificulta a detecção e prevenção do ataque.

Os ataques de negação de serviço usam um conjunto de máquinas espalhadas geograficamente para gerar requisições falsas ao seu alvo. O objetivo é conseguir uma concentração de requisições no alvo e nos últimos saltos dos enlaces a ele. Assim, a detecção perto da origem do ataque não é uma tarefa trivial, porque o número de requisições geradas por cada máquina pode não ser elevado. Algumas defesas contra negação de serviço usam mecanismos para detectar y bloquear tráfego no destino. Porém, o uso de defesas unicamente no objetivo, não evita o consumo de recursos na rede causado pelas inundações. Outro tipo de defesas, chamadas híbridas, combina a detecção do ataque perto do destino e mecanismos para bloquear o tráfego nos roteadores da rede. Logo, é possível evitar a concentração de requisições no alvo e o consumo de recursos na rede [Zargar et al., 2013]. Alguns desses mecanismos podem funcionar mesmo sem conseguir diferenciar os fluxos maliciosos dos legítimos e visam a evitar a inanição. Contudo, essas propostas usam mecanismos que atuam de forma distribuída, ou requer introduzir um cabeçalho adicional aos pacotes atravessando a rede, o que diminui seu desempenho.

Este trabalho propõe FlowFence, um sistema de prevenção de ataques de negação de serviço usando controle de banda, que não requer introduzir um cabeçalho adicional aos pacotes atravessando a rede. FlowFence possui uma arquitetura com roteadores e um controlador coordenando as ações de defesa à negação de serviço. Os roteadores da rede monitoram o estado de ocupação de suas interfaces de saída e quando detectam um congestionamento notificam ao controlador da rede. O controlador envia comandos de controle de banda para todos os roteadores que possuem fluxos encaminhados à interface congestionada. Em resposta aos comandos recebidos, os roteadores limitam a banda de suas interfaces de saída utilizando um sistema de filas. O controle de banda é utilizado para proporcionar uso equitativo dos recursos e diminuir os gargalhos na rede. Um protótipo do FlowFence foi implementado no *Future Internet Testbed with Security* (FITS) [1]. Os resultados da avaliação do protótipo evidenciam que o FlowFence garante o uso equitativo dos recursos da rede aos usuários. Os resultados também evidenciam que FlowFence evita a inanição dos usuários legítimos em presença de ataques de negação de serviço com grandes volumes de inundação.

O restante do artigo esta organizado da seguinte forma: A Seção 2 analisa os trabalhos relacionados em prevenção de DoS. A Seção 3 apresenta o modelo de atacante e os objetivos do sistema FlowFence. A Seção 4 descreve a arquitetura e o funcionamento do FlowFence. Os resultados da implementação do protótipo são apresentados e discutidos na Seção 5. A Seção 6 conclui o artigo.

[1]FITS é uma rede de testes interuniversitária desenvolvida a partir da parceria de instituições brasileiras e europeias. Site: http:/ www.gta.ufrj./fits

2. Detecção e Prevenção e ataque de Negação de Serviços

2.1. Ataques de Negação de Serviço

Existem dois métodos principais para lançar ataques de negação de serviço. O primeiro método consiste em enviar pacotes mal formados, que visam causar uma exceção na máquina de estados do protocolo ou da aplicação da vítima. Uma exceção de esse tipo ocasionaria a caída do serviço. Esse primeiro ataque e conhecido como ataque de vulnerabilidade. O segundo método, chamado ataques de inundação, consiste em criar uma quantidade enorme de requisições que visam consumir os recursos dos usuários legítimos. Entre os recursos que podem ser consumidos estão: largura de banda, capacidade de encaminhamento dos roteadores ou memória e processamento dos servidores. As inundações feitas no nível de rede/transporte visam a consumir a largura de banda disponível para os usuários legítimos criando pacotes TCP, UDP, ICMP ou DNS. Também podem ser feitas inundações que visam explorar características de determinados protocolos de transporte. O ataque SYN Flood explora a vulnerabilidade do processo de conexão com três trocas (*three way handshake*) do TCP. Esse ataque cria múltiplas requisições de abrir conexão no servidor, mas o atacante não completa o estabelecimento da conexão. De essa forma, o servidor é forçado a manter múltiplas conexões abertas ao mesmo tempo, consumindo sua memória. O modelo do atacante do FlowFence foca em ataques de negação de serviço por inundação no nível de rede/transporte.

2.2. Defesas contra Ataques de Negação de Serviço por Inundação

Para dificultar a detecção de um ataque distribuído de negação de serviço um atacante pode falsificar os campos dos pacotes enviados. Porém, esses pacotes têm que ser encaminhados por roteadores através de rotas determinadas. Em esse processo os roteadores podem marcar os pacotes e usar essas marcas para rastrear as fontes de um ataque de negação de serviço (*Denial of Service Attack* - DoS). Chen *et al.* propõem um sistema de prevenção de DoS baseado na marcação determinística de pacotes [Chen et al., 2007]. A diferencia da marcação probabilística, a marcação determinística marca todos os pacotes encaminhados por um roteador. Usando um Sistema de Detecção de Intrusão (IDS) (*Intrusion Detection System* - IDS) um host detecta um DoS e envia uma requisição ao seu roteador para marcar deterministicamente os pacotes pertencentes ao ataque. Os pacotes marcados são usados pelo IDS para construir uma árvore que tem como raiz a vítima e as folhas são cada fonte do ataque. A marcação determinística evita que um pacote com marca falsificada percorra todo o caminho do ataque, porque os roteadores no caminho remarcariam o pacote. Contudo, o sistema é vulnerável a ataques que enviem inundações em rajadas com uma duração menor do que o tempo de criação da árvore. Nesse caso, a fonte do ataque não é detectada. O sistema depende que os destinos finais sejam confiáveis, o que torna o sistema vulnerável a ataques de conspiração. Onde um destino final é um roteador podem falsificar requisições de marcação de pacotes, gerando falsos alarmes. FlowFence pode usar um IDS para identificar tráfego malicioso e enviar uma ordem para descartar esse tráfego, mas se não é possível identificar o tráfego malicioso, garante um uso equitativo entre todos os usuários, evitando a inanição.

Para evitar ataques em rajadas, Laufer *et al.* propõe, um sistema de rastreamento que reconstrói o caminho do ataque usando somente um pacote [Laufer et al., 2005] . O

sistema usa uma generalização de um filtro de Bloom para criar um *hash.* que inclui todas as assinaturas dos roteadores em um caminho. A generalização é usada para reduzir a probabilidade de falsos positivos. Um roteador pode dar capacidades, medidas em *tokens*, aos nós da rede para controlar o tráfego recebido. Os nós da rede utilizam certo número de *tokens* para enviar pacotes. Yang *et al.* propõem TVA, uma arquitetura que usa *tokens* para prevenir DoS [Yang et al., 2008]. Um IDS é utilizado para detectar tráfego malicioso e o roteador não renova os *tokens* dos nós enviando esse tráfego. Em TVA, os receptores devem armazenar informação de estado por cada fluxo e as fontes devem modelar o uso das capacidades para solicitar a renovação. Esse armazenamento de estado e modelagem de capacidades limita a escalabilidade da proposta. A ação de controle da TVA é a não renovação de *tokens*. Assim, um atacante identificado pode continuar criando inundações até que seus *tokens* acabem. FlowFence usa o controle de congestionamento, para evitar o uso de *tokens* que requerem processos de negociação e renovação.

NetFence [Liu et al., 2010] é uma arquitetura que cria uma malha fechada de controle de congestionamento para reduzir o impacto de um DoS. Nesta arquitetura, roteadores detectam congestionamento em suas interfaces de saída e enviam informação autenticada de retroalimentação aos roteadores pertos da fonte do ataque para reduzir o impacto do DoS. Para trocar essas informações de uma forma segura os autores propõem agregar um campo entre a camada IP e TCP [Liu et al., 2006]. Embora a proposta evite a inanição dos usuários legítimos, a arquitetura propõe a modificação da pilha TCP/IP, o que limita sua aplicabilidade. Além disso, a informação de retroalimentação está presente em cada pacote enviado e usa um *hash* criptográfico. O *hash* tem que ser processado para cada pacote enviado. FlowFence não requer modificações na pilha de protocolos e evita a sobrecarga da inclusão de um novo cabeçalho, com o uso de um controlador que se comunica com os comutadores da rede para controlar a banda.

Braga *et al.* propõem um IDS que funciona como uma aplicação de redes definidas por software [Braga et al., 2010]. O IDS esta baseado em um mapa auto organizado (*Self Organized Map - SOM*) que recebe as informações do controlador e identifica fluxos que pertencem a um DDoS. Os melhores resultados foram obtidos ao serem usadas quatro métricas dos fluxos na rede: média de pacotes por fluxo, média de bytes por fluxo, duração média do fluxo e porcentagem de fluxos pares. Os IDS baseados em classificadores são sensíveis aos dados usados para treinar o classificador e podem não detectar tipos de ataques desconhecidos ou que não foram considerados durante o treinamento. Nesse IDS, não foram considerados DDoS que realizam inundações em rajadas ou enviam pacotes grandes.

Mattos e Duarte propõem QFlow [Mattos e Duarte, 2012], um mecanismo para garantir qualidade de serviço em redes virtuais usando controle de recursos em redes OpenFlow. O mecanismo é baseado em Xen e OpenFlow, o controle de recursos de rede é feito usando um sistema de filas em OpenFlow. Cada máquina virtual configura seus parâmetros de QoS e eles são lidos por uma aplicação no hipervisor, essa aplicação associa esses parâmetros a um conjunto de filas para tentar garantir o mínimo solicitado por cada máquina virtual. Em caso de ainda dispor de recursos físicos disponíveis, os recursos são distribuídos de acordo aos parâmetros configurados por cada máquina virtual. FlowFence provê um mecanismo para detectar congestionamento nas interfaces de saída dos roteadores e brinda uso equitativo para todos os fluxos sendo encaminhados no

roteador.

O uso de mecanismos de preço e mercado é uma abordagem recente para prevenir DoS. Neste modelo todos os usuários compartilhando um recurso expõem à rede o que poderiam pagar por seu uso. Um preço é calculado e todos os usuários, legítimos e maliciosos, pagam pelo que usam. Vulimiri *et al.* propõem um modelo analítico do máximo dano que um usuário malicioso pode fazer [Vulimiri et al., 2012]. Neste cenário. No pior caso, onde um usuário malicioso controla uma *botnet* de M usuários com uma quantidade fixa de recursos, um usuário pode obter um rendimento de: B/B+M, em que B é a quantidade de recursos que um usuário legítimo deseja gastar para usar o recurso compartilhado. Esse rendimento é superior às abordagens de uso equitativo, onde um usuário pode obter máximo 1/B+M. No entanto, as estratégias de preço requerem uma análise adicional para fazê-las realizáveis em redes reais.

3. Objetivos do Sistema FlowFence

3.1. Modelo do Atacante

Ataques de inundação: O atacante gera ataques de inundação de pacotes que visam esgotar recursos da rede, como capacidade dos enlaces ou de encaminhamento dos roteadores. No modelo, não são considerados atacantes que geram negação de serviço por exploração de vulnerabilidades ou mensagens com valores inesperados no cabeçalho. O atacante não consegue comprometer ao controlador ou os comutadores.

Tráfego malicioso idêntico ao legítimo: Assume-se que o atacante pode gerar inundações com tráfego que imita tráfego legítimo e por tanto, não é possível diferenciar os fluxos maliciosos dos legítimos.

3.2. Objetivos

Uso equitativo de recursos: Na ausência de identificação de tráfego malicioso, a proposta garante o uso equitativo dos recursos da rede para os usuários, o que reduz o impacto do ataque de negação de serviço. Algumas propostas visam punir fluxos que apresentam grande consumo de banda, para garantir recursos para os fluxos pequenos [Mahajan et al., 2002]. Mas na ausência de um IDS, é possível que alguns de esses fluxos grandes sejam de usuários legítimos.

Não requer modificações nos protocolos usados: O sistema não deve requer modificações nas pilhas de protocolos usadas nas redes dos usuários. O sistema não deve incluir a sobrecarga de um cabeçalho nos pacotes encaminhados na rede.

Rápida resposta: O sistema deve reagir rapidamente às condições de negação de serviço detectadas.

4. Arquitetura

O sistema FlowFence consiste de roteadores que monitoram suas interfaces de saída para detectar condições de congestionamento e um controlador que mantém informações sobre o estado da rede, para enviar comandos de controle de banda a todos os roteadores com fluxos ao enlace congestionado. Os roteadores ao receber comandos do controlador criam filas para cada fluxo sendo encaminhado. Nesse caso, um fluxo é definido como o conjunto de mensagens que possuem a mesma IP de origem. O roteador

divide equitativamente a banda disponível na interface de saída entre as filas criadas e cada fluxo ocupa uma fila. Quando termina o ataque, o roteador reporta o novo estado ao controlador e termina o controle de banda. Para prevenir que um atacante gere falsas alarmas a comunicação entre o FlowFence e o controlador usa um canal seguro. A arquitetura do FlowFence é apresentada na Figura 1. Os componentes do funcionamento do FlowFence são apresentados na Figura 2

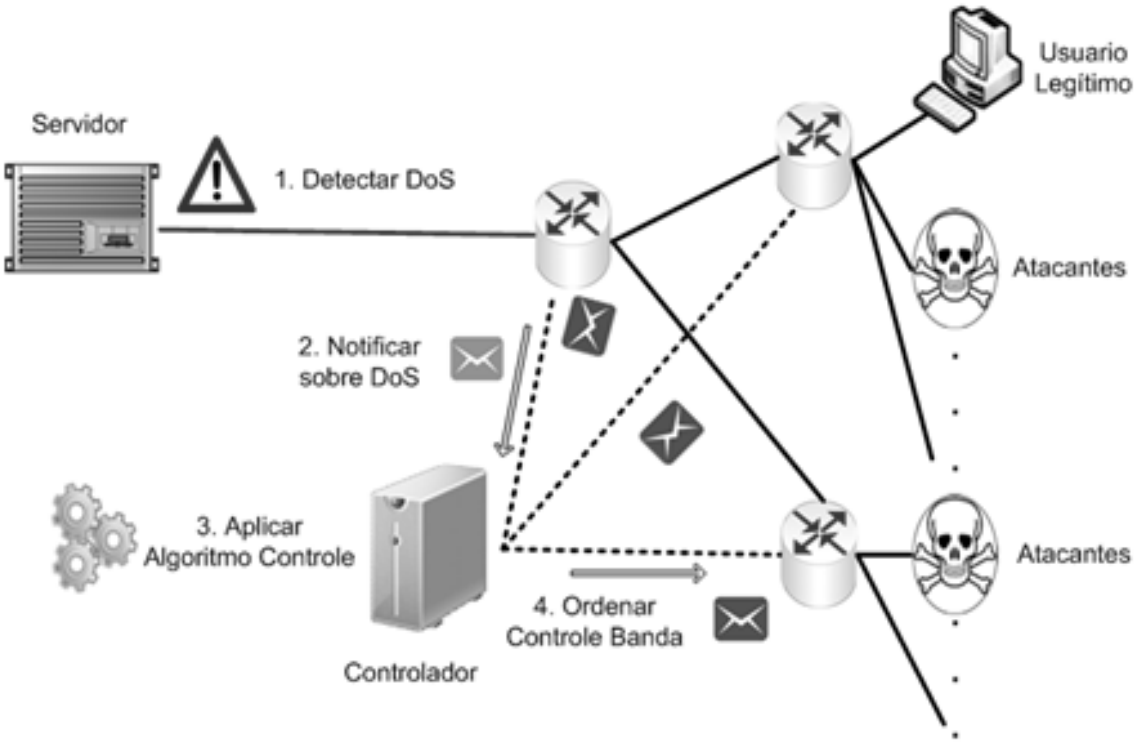

Figura 1. Arquitetura de FlowFence. 1) roteadores monitoram o uso de suas interfaces de saída. 2) Ao detectar uma condição de negação de serviço, notificam ao controlador. 3) O controlador executa a aplicação de controle de congestionamento. 4) O controlador envia ordens para que os roteadores controlem a banda de suas interfaces de saída. O objetivo da aplicação de controle e garantir uso equitativo aos usuários legítimos ou maliciosos dos recursos da rede

Detecção de Condição de Negação de Serviço: Os ataques de negação de serviço por inundação geram altas taxas de ocupação das interfaces de saída dos roteadores. Em muitos casos os roteadores possuem enlaces bem provisionados e em condições normais a ocupação media da interface é menor ao 95% [Liu et al., 2010]. Os roteadores em FlowFence periodicamente tomam amostras da quantidade de bytes transmitida por segundo e utilizam uma janela deslizante exponencial para medir a ocupação media do canal. Quando a ocupação do canal supera o 95% da capacidade total, uma notificação de congestionamento é enviada ao controlador. O uso de uma janela deslizante permite evitar alarmes falsos por possíveis picos de curta duração que podem ser condições normais da rede. Foi utilizada uma janela exponencial, por dar um peso maior às últimas amostras tomadas, detectando rapidamente uma condição de DoS.

Notificação ao Controlador: O controlador da rede mantém conexões seguras com os roteadores utilizando uma interface diferente da utilizada para o encaminhamento

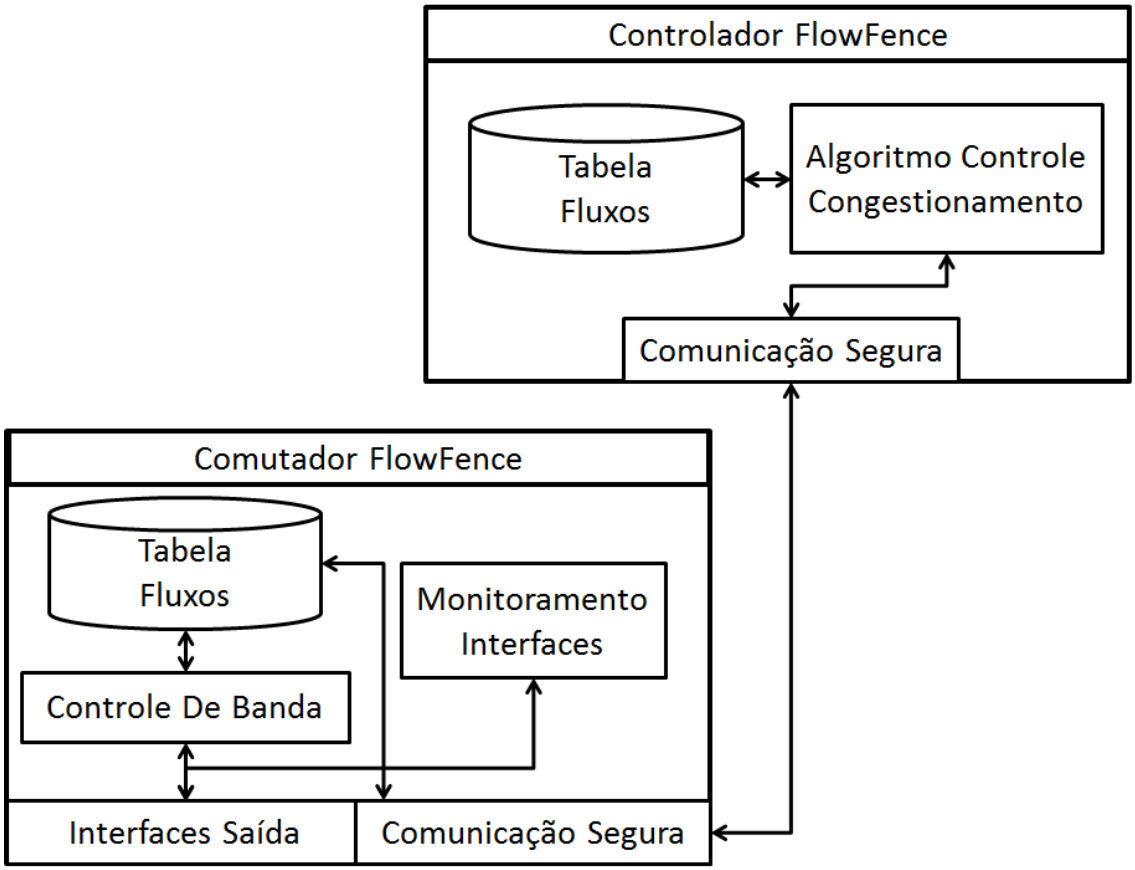

Figura 2. Componentes do FlowFence. O roteador possui um módulo para monitorar as interfaces e reportar congestionamentos ao controlador usando um canal de comunicação seguro. O controlador usa as informações da tabela de fluxos e um algoritmo de controle de congestionamento para encaminhar os fluxos às filas de controle de banda no roteador.

dos dados dos usuários. Quando uma condição de DoS é detectada o roteador envia uma mensagem que contêm: O identificador do roteador, o endereço de rede do enlace congestionado e uma marcação de tempo. A marcação de tempo é usada para evitar possíveis ataques de repetição.

Algoritmo de Controle: Quando o controlador recebe uma notificação de congestionamento, verifica em sua tabela de fluxos, quais roteadores possuem fluxos que usam o enlace congestionado e envia um comando para encaminhar os fluxos às filas de controle de banda criadas pelo roteador. O controle de banda brinda a mesma quantidade de banda para cada fluxo encaminhado.

Controle de Banda: Um roteador cria filas para cada um dos fluxos encaminhados e da uma fatia igual da banda total para cada fila. A banda é controlada até que a ocupação da interface de saída é menor ao 80 % da capacidade total da interface. Quando a ocupação do canal é menor que 80% uma nova notificação é enviada ao controlador e o controle banda é terminado.

5. Implementação e Avaliação

O protótipo de FlowFence foi implementado em uma ilha do *Future Internet Testbed with Security* (FITS) [Guimarães et al., 2013]. FITS [2] é um testbed de código aberto baseado em OpenFlow e no hipervisor Xen [Mattos et al., 2012]. Uma rede virtual foi criada com máquinas virtuais gerenciadas pelo hipervisor Xen 4.1.4. Os roteadores da rede utilizam o roteador programável Open vSwitch[3], que provê o mecanismo para implementar as filas para o controle de banda. O encaminhamento dos fluxos para as filas é controlado por OpenFlow. O controlador da rede utiliza o controlador de OpenFlow POX. Cada roteador virtual executa uma aplicação cliente FlowFence encarregada de monitorar as interfaces de saída, notificar ao controlador sobre o estado do congestionamento e aplicar o controle de banda. O controlador da rede executa a aplicação FlowFence que mantém informações sobre o estado da rede, recebe as notificações de congestionamento e modifica os fluxos dos roteadores para encaminhá-los nas filas de controle de banda.

Para criar os ataques de negação de serviço foram importadas ao FITS as ferramentas oferecidas pelo tesbted DETER [Mirkovic et al., 2010]. DETER é um testbed de segurança fundado pela Agência Nacional de Segurança e com participação das Universidades de Berkeley e South Califórnia. DETER oferece a ferramenta de controle de experimentos chamada MAGI. MAGI [4] é um sistema de gerenciamento que permite controlar experimentos, através de procedimentos, eventos e grupos de nós. Um procedimento é um *script* que define o experimento realizado. No procedimento são especificados os nós participantes do experimento e os grupos aos que pertencem. Nós no mesmo grupo executam o mesmo código. Esse código corresponde às ferramentas de negação de serviço desenvolvidas pelo equipe do DETER. Finalmente, a execução de código é controlada utilizando os eventos que servem como gatilhos para começar a executar o código. Nos testes feitos ao protótipo foi criado um procedimento onde um conjunto de máquinas virtuais pertence a um grupo de atacantes. São configurados os parâmetros dos ataques e todos os nós iniciam e terminam seus ataques ao mesmo tempo.

As ferramentas `Iperf`[5], `Httperf`[6] foram usadas para realizar as medidas de avaliação de desempenho do protótipo. Um servidor foi utilizado para hospedar as máquinas virtuais usadas nos testes. O servidor possui um processador Intel(R) Xeon(R) CPU X5690 3.47GHz com 16 núcleos e 48GB de memória RAM e executa Debian Linux 3.2.0-4-amd64. As máquinas virtuais são configuradas com um CPU virtual, 256MB de ARAM e executam Debian Linux 3.2.0-4-amd64, a banda das interfaces virtuais foi limitada a 100Mb/s.

Foi utilizada uma topologia Dumbbell apresentada na Figura 3, a topologia está composta por 2 roteadores, 1 controlador, 10 dispositivos finais e 1 servidor. Os 10 dispositivos finais pertencem a uma mesma LAN e se conectam a um dos roteadores. O servidor está conectado ao segundo roteador e os dois roteadores são conectados usando uma interface virtual de 100Mb/s. Uma rede separada é usada para comunicar os roteadores com

[2]FITS é uma rede de testes interuniversitária desenvolvida a partir da parceria de instituições brasileiras e europeias. Maiores informações em http://www.gta.ufrj./fits

[3]http://www.openvswitch.org/.

[4]Mais informações nas características do MAGI: http://montage.deterlab.net/montage

[5]http://iperf.sourceforge.com/

[6]http://www.hpl.hp.com/research/linux/httperf/

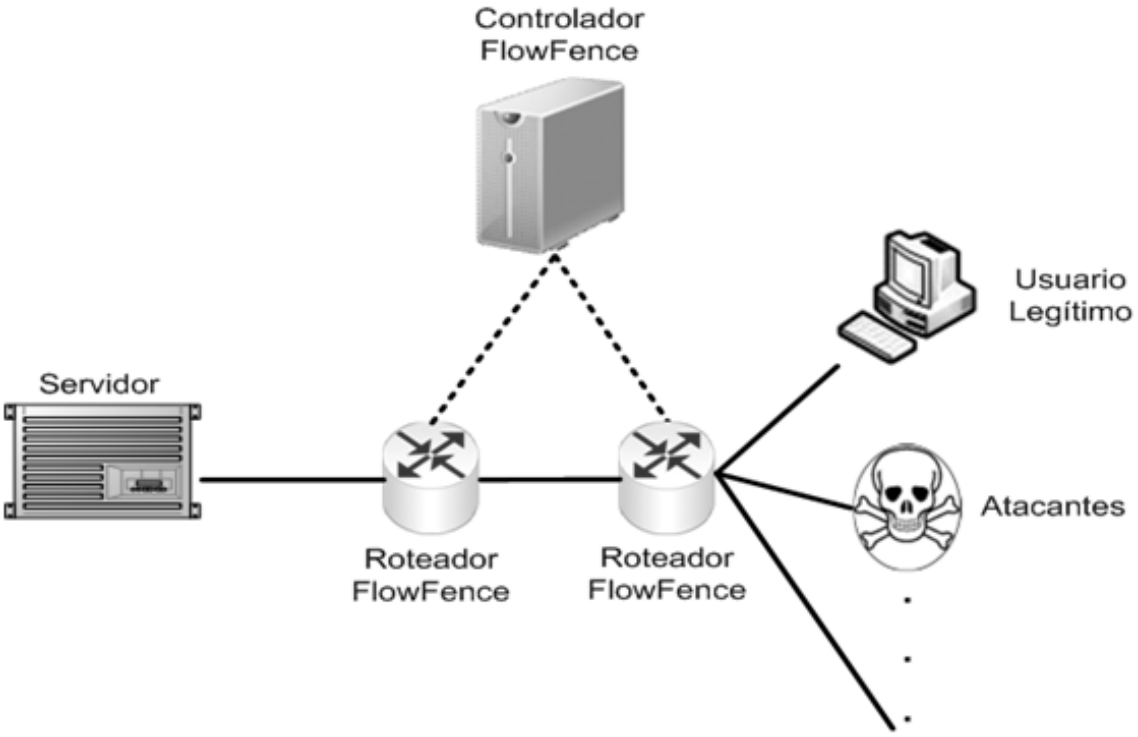

Figura 3. Topologia utilizada durante os testes. Um grupo de atacantes criam inundações ao servidor e um usuário legítimo realiza requisições ao servidor. Os enlaces entre os roteadores e o roteador e o servidor são congestionados e o FlowFence atua controlando banda em esses enlaces.

o controlador. Dos 10 dispositivos finais, 1 é usado como cliente do servidor e o número de atacantes é alterado segundo os testes a continuação. Tanto o cliente como os atacantes encaminham suas requisições e inundações ao servidor.

O primeiro teste avalia a eficácia do protótipo em assegurar o uso equitativo dos recursos da rede para o cliente. Um cliente utiliza `Iperf` para avaliar a banda disponível até o servidor é ao mesmo tempo um número variável de atacantes inicia uma inundação. Cada atacante gera uma inundação de 300 Mb/s, usando pacotes UDP de 1024 bytes destinadas à porta 80 do servidor. O `Iperf` foi executado durante 30 segundos. A Figura 4 mostra os resultados do teste. É possível perceber que o cliente, mesmo durante uma inundação que supera várias vezes a capacidade dos enlaces, consegue atender uma taxa muito perto da taxa esperada.

O segundo teste avalia a capacidade do FlowFence para evitar a inanição dos usuários legítimos na presencia de um DoS. Neste teste o cliente usa a aplicação `Httperf` para testar o tempo de resposta de um servidor HTTP. O cliente tenta realizar conexões a uma taxa de 100 conexões por segundo, durante 30 segundos. O parâmetro *timeout* da tentativa de conexão foi estabelecido em 7 segundos. Durante os 30 segundos os atacantes realizam uma inundação com pacotes UDP de 1024 bytes na porta 80, que é a porta onde o servidor escuta as requisições HTTP. Cada atacante cria 40000 pacotes por segundo, gerando um fluxo de 320Mb/s, e o número de atacantes foi variado a cada teste até alcançar um volumo de inundação de 3 Gb/s. A Figura 5 mostra o resultado do teste. Sem o uso de FlowFence o cliente não consegue estabelecer uma conexão para volumes

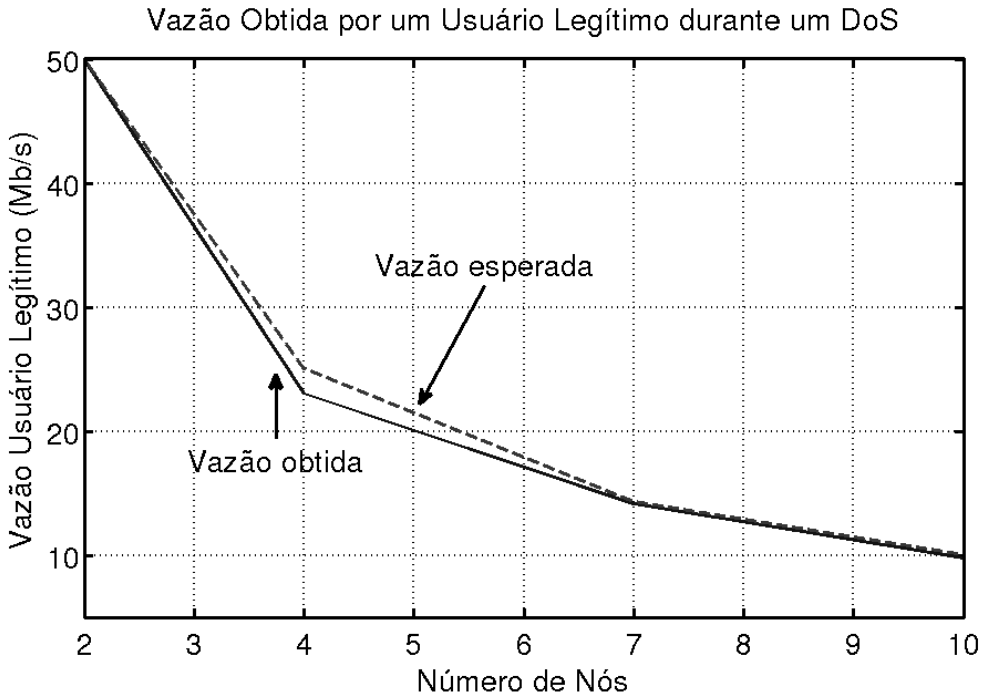

Figura 4. Vazão de um usuário legítimo durante um ataque de negação de serviço. A vazão de um usuário legítimo diminui proporcionalmente com a quantidade de usuários, transmitindo uma taxa de 40 000 pacotes por segundo aumenta. A diminuição proporcional é uma consequência do uso equitativo dos recursos disponíveis da rede estabelecidos pelo algoritmo de controle utilizado.

maiores aos 1500 Mb/s, já que o tempo de tentativa de conexão excede os 7 segundos. Com FlowFence, os resultados mostram um crescimento baixo no tempo de resposta do servidor com o incremento no número de clientes. O tempo de resposta do servidor é de 1 segundo com um volume de inundação de 3 Gb/s, o que comprova que a inanição é evitada.

O terceiro teste apresenta o tempo de reação do FlowFence, entendido como o tempo que tarda o sistema em dar um uso equitativo do enlace a um usuário legítimo depois do inicio de uma inundação. Novamente, foi utilizado o `Iperf` para medir a banda disponível para um usuário legítimo, quando 9 atacantes geram uma inundação de 3Gb/s. A duração do `Httperf` foi de 30 segundos e da inundação de 35 segundos. A Figura 6 apresenta os resultados do teste. O usuário legítimo perde sua banda disponível no segundo 4, e 4 segundos depois FlowFence consegue dar uma fatia equitativa da banda a ele.

6. Análise proposta

A segurança do mecanismo FlowFence é proporcionada pela comunicação segura entre os roteadores e o controlador, que é feita com uma rede distinta à rede utilizada para encaminhar pacotes dos usuários. Desta forma, um atacante não consegue falsificar uma notificação de congestionamento que poderia prejudicar aos usuários legítimos. Se um roteador fosse comprometido, ele poderia notificar ao controlador de um possível DoS, o que dispararia erroneamente o mecanismo de controle de banda em vários roteadores. Se

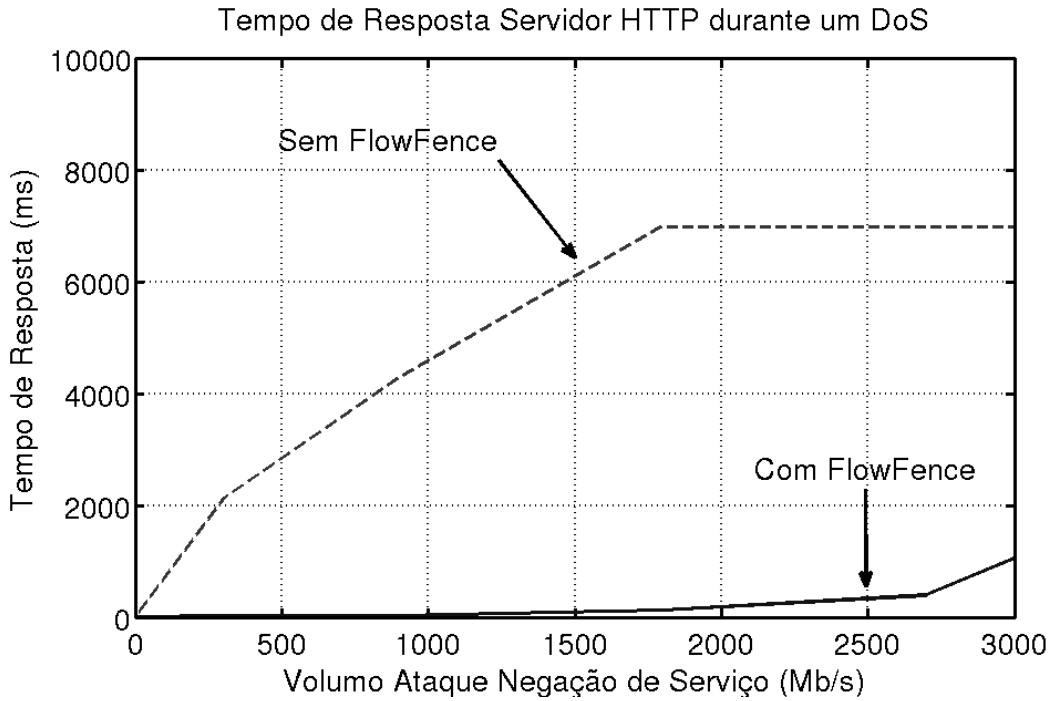

Figura 5. Tempo de resposta de um servidor HTTP durante um ataque de negação de serviço. Com volumes de inundação maiores a 1.5Gb/s o servidor não consegue responder as requisições durante o máximo tempo estabelecido (7 segundos). Com FlowFence, o tempo de resposta é incrementado proporcionalmente até um volumo de 3Gb/s

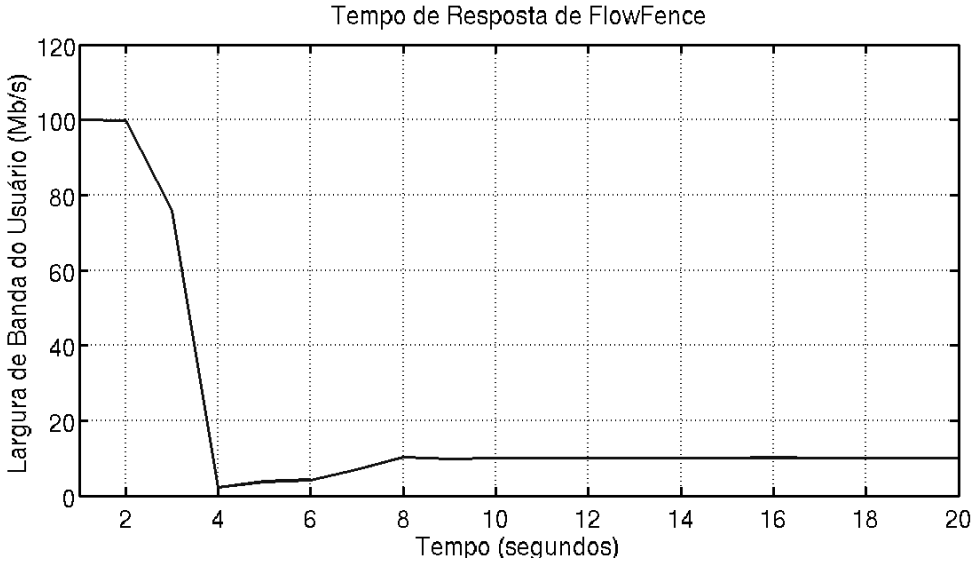

Figura 6. Tempo que tarda o FlowFence em limitar a banda das interfaces congestionadas. `Httperf` foi usado para medir a banda disponível para um usuário, durante uma inundação, através do tempo. A inundação começa no segundo 2 do teste. No segundo 4 a interface de saída é congestionada e depois de 4 segundos o FlowFence consegue dar a banda adequada para o usuário legítimo.

o controlador fosse comprometido, toda a rede estaria comprometida.

O sistema FlowFence pode oferecer, na ausência de um sistema de Detecção de Intrusão, uso equitativo dos recursos da rede por parte de usuários maliciosos ou legítimos. Se um IDS fosse utilizado, o FlowFence poderia evitar a inanição dos usuários legítimos, quando o IDS detecta o fluxo malicioso e controlador ordena seu bloqueio. No entanto, o FlowFence não consegue limitar efetivamente um ataque de negação de serviço distribuído, porque um atacante com a capacidade de criar centenas de fluxos maliciosos, levaria ao usuário a usar uma banda muito pequena da rede, o que poderia ocasionar que muitas transações não fossem terminadas com sucesso.

Mesmo em inundações com um alto volume de tráfego, FlowFence consegue reagir em poucos segundos e evitar a inanição dos usuários legítimos. Uma proposta que usasse um mecanismo de controle ponto a ponto, com comunicação entre os roteadores poderia tardar mais tempo até que o controle de banda chegue perto ao origem do ataque. A arquitetura de comunicação entre os roteadores e o controlador evita a necessidade de incluir informação de retroalimentação para o controle de banda, incluir essa informação implicaria uma modificação dos protocolos convencionais utilizados nas redes de dados. Adicionalmente, essa informação deve estar autenticada para evitar sua falsificação, o que representaria realizar operações criptográficas salto, a salto, para um determinado conjunto de pacotes.

Uma arquitetura com um controlador com informações sobre o estado da rede permite a implantação de algoritmos de controle fim-a-fim para prevenir ataques de negação de serviço e outras situações de congestionamento, FlowFence, usa um algoritmo para dar uso equitativo a todos os fluxos, mas alternativas podem ser utilizadas [Ghobadi et al., 2012]

Para a implementação do sistema FlowFence, é necessária a inclusão de um controlador na rede, e de enlaces adicionais que comuniquem ao controlador com os roteadores, adicionalmente, os roteadores devem ser modificados para implementar o monitoramento, as notificações e o controle de banda. Se forem usados roteadores programáveis como o OpenvSwitch, essas modificações são simples de implementar.

Os sistemas que utilizam arquiteturas com controle centralizado possuem desafios de escalabilidade que devem ser abordados para garantir seu funcionamento [Yeganeh et al., 2013]. Algumas propostas visam a brindar escalabilidade em esse tipo de sistemas. Yu *et al.* propõem o DIFANE, um sistema de redes definidas por software com escalabilidade [Yu et al., 2010]. DIFANE cria uma arquitetura hierárquica entre os controladores, estabelecendo um controlador principal e controladores de autoridade. Cada controlador de autoridade gerencia uma fatia da rede e suas políticas são estabelecidas pelo controlador principal. De essa forma, as requisições para criar novos fluxos são distribuídas entre os controladores de autoridade. Koponen *et al.* propõem o sistema ONIX para redes definidas por software [Koponen et al., 2010]. ONIX, modela uma rede como uma base de dados global com entradas sobre os roteadores, o estados dos fluxos e suas ações de encaminhamento. Diversos controladores da rede podem consultar e modificar essa base de dados, alterando o encaminhamento dos pacotes. ONIX oferece ferramentas para que as aplicações criadas para controlar a rede, garantam a consistência na base de dados.

7. Conclusões

Neste trabalho foi apresentado FlowFence, um sistema de prevenção de ataques de negação de serviço usando comunicação segura entre roteadores e controlador e o controle de banda. O sistema proposto identifica condições de negação de serviço monitorando a ocupação das interfaces de saída dos roteadores e um controlador para implementar uma aplicação de controle que garante uso equitativo de recursos a todos os fluxos encaminhados a um enlace congestionado. Os resultados mostram que na ausência de um sistema de detecção de intrusão, FlowFence garante o uso equitativo aos usuários durante um DoS, evitando a inanição. Como trabalho futuro, pretende-se usar OpenFlow para utilizar algoritmos mais robustos de controle de congestionamento fim-a-fim, assim como estudar as vulnerabilidades da comunicação entre roteadores e controladores em OpenFlow na presença de ataques distribuídos de negação de serviço.

Referências

Arbor Networks (2012). *Worldwide Infrastructure Security Report: 2012 Report*. Arbor Networks, http://pages.arbornetworks.com/rs/arbor/images/WISR2012. Acessado em novembro de 2013.

Braga, R., Mota, E. e Passito, A. (2010). Lightweight DDoS flooding attack detection using NOX/openflow. Em *IEEE 35th Conference on Local Computer Networks (LCN), 2010*, p. 408–415.

Chen, R., Park, J.-M. e Marchany, R. (2007). A divide-and-conquer strategy for thwarting distributed denial-of-service attacks. *IEEE Transactions on Parallel and Distributed Systems*, 18(5):577–588.

Ghobadi, M., Yeganeh, S. H. e Ganjali, Y. (2012). Rethinking end-to-end congestion control in software-defined networks. Em *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, p. 61–66. ACM.

Guimarães, P. H. V., Ferraz, L. H. G., Torres, J. V., Mattos, D. M. F., Murillo P., A. F., Andreoni, M., Alvarenga, I. D., Rodrigues, C. S. C. e Duarte, O. C. M. B. (2013). Experimenting content-centric networks in the future internet testbed environment. *IEEE International Conference on Communications (ICC)-Workshop on Cloud Convergence*.

Koponen, T., Casado, M., Gude, N., Stribling, J., Poutievski, L., Zhu, M., Ramanathan, R., Iwata, Y., Inoue, H., Hama, T. e Shenker, S. (2010). Onix: A distributed control platform for large-scale production networks. Em *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation*, OSDI'10, p. 1–6. USENIX Association.

Laufer, R. P., Velloso, P. B. e Duarte, O. C. M. B. (2005). Um novo sistema de rastreamento de pacotes ip contra ataques de negação de serviço. *XXIII Simpósio Brasileiro de Redes de Computadores - SBRC 2005*.

Liu, X., Yang, X., Wetherall, D. e Anderson, T. (2006). Efficient and secure source authentication with packet passports. Em *Proceedings of the 2Nd Conference on Steps to Reducing Unwanted Traffic on the Internet - Volume 2*, p. 2–2. USENIX Association.

Liu, X., Yang, X. e Xia, Y. (2010). NetFence: preventing internet denial of service from inside out. Em *Proceedings of the ACM SIGCOMM 2010 conference*, SIGCOMM '10, p. 255–266. ACM.

Mahajan, R., Bellovin, S. M., Floyd, S., Ioannidis, J., Paxson, V. e Shenker, S. (2002). Controlling high bandwidth aggregates in the network. *ACM SIGCOMM Computer Communication Review*, 32(3):62–73.

Mattos, D. M. F. e Duarte, O. C. M. B. (2012). QFlow: Um sistema com garantia de isolamento e oferta de qualidade de serviço para redes virtualizadas. *XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos-SBRC*.

Mattos, D. M. F., Mauricio, L. H., Cardoso, L. P., Alvarenga, I. D., Ferraz, L. H. G. e Duarte, O. C. M. B. (2012). Uma rede de testes interuniversitária a com técnicas de virtualizacao híbridas. *Salão de Ferramentas do XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos-SBRC*.

Mirkovic, J., Benzel, T., Faber, T., Braden, R., Wroclawski, J. e Schwab, S. (2010). The DETER Project: Advancing the Science of Cyber Security Experimentation and Test. Em *IEEE International Conference on Technologies for Homeland Security (HST)*.

The Huffington Post (2013). *Spamhaus Hit With 'Largest Publicly Announced DDoS Attack' Ever, Affecting Internet Users Worldwide*. The Huffington Post Inc, http://www.huffingtonpost.com/2013/03/27/spamhaus-cyber-attack_n_2963632.htm. Acessado em outubro de 2013.

Vulimiri, A., Agha, G. A., Godfrey, P. B. e Lakshminarayanan, K. (2012). How well can congestion pricing neutralize denial of service attacks? Em *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '12, p. 137–150. ACM.

Wenye, W. e Zhuo, L. (2013). Cyber security in the smart grid: Survey and challenges. *Computer Networks*, 57(5):1344–1371. Elsevier.

Yang, X., Wetherall, D. e Anderson, T. (2008). TVA: A DoS-limiting network architecture. *IEEE/ACM Transactions on Networking*, 16(6):1267–1280.

Yeganeh, S., Tootoonchian, A. e Ganjali, Y. (2013). On scalability of software-defined networking. *IEEE Communications Magazine*, 51(2):136–141.

Yu, M., Rexford, J., Freedman, M. J. e Wang, J. (2010). Scalable flow-based networking with difane. Em *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, p. 351–362. ACM.

Zargar, S., Joshi, J. e Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Communications Surveys Tutorials*, PP(99):1–24.