FlowFence: Um Sistema de Defesa contra Ataques de Negação de Serviço para Redes Definidas por Software*

Andrés Felipe Murillo Piedrahita[1], Sandra Rueda[1], Diogo Menezes Ferrazani Mattos[2] e Otto Carlos Muniz Bandeira Duarte[2]

[1]Systems and Computing Engineering Department – School of Engineering
Universidad de los Andes – Colômbia

[2]Grupo de Teleinformática e Automação
Universidade Federal do Rio de Janeiro (UFRJ) – Rio de Janeiro – Brasil

Resumo. *Este artigo propõe o FlowFence, um sistema eficiente e de reação rápida para a detecção e a mitigação de ataques de negação de serviço em Redes Definidas por Software. A mitigação da inanição dos usuários legítimos da rede ocorre através da alocação de uma banda média para os fluxos, enquanto os fluxos superiores à média são penalizados com o acesso a uma banda menor. A penalização de diminuição de banda é exponencial à diferença entre o valor médio e o valor atual de uso de banda do fluxo. Um protótipo do sistema foi implementado e avaliado no Future Internet Testbed with Security (FITS).*

Abstract. *In this paper, we propose FlowFence, a lightweight and fast denial of service detection and mitigation system for Software Defined Networking. The mitigation procedure for network-user starvation allocates an average bandwidth, while flows exceeding the mean are penalized with a lower bandwidth assignment. The penalization is exponential to the difference between the fair limit and the current bandwidth usage. A system prototype is implemented and evaluated in the Future Internet Testbed with Security (FITS).*

1. Introdução

O volume de tráfego de Ataques de Negação de Serviço (*Denial of Service* - DoS) vem apresentando um padrão crescente, chegando aos 100 Gb/s em 2010 e 400 Gb/s em 2014. Esse tráfego compromete os principais enlaces, roteadores e serviços da Internet. Ataques sofisticados imitam o tráfego legítimo, dificultando a detecção e a prevenção. A detecção baseada na origem não é uma tarefa trivial em ataques distribuídos, pois o número de requisições gerado por cada atacante pode ser muito baixo. A detecção baseada no destino é mais fácil que a baseada na origem, mas não evita o consumo de recursos de rede da origem até o destino. A detecção híbrida combina a detecção perto do destino com mecanismos para bloquear o tráfego nos roteadores perto da origem do ataque. Desta forma, é possível reduzir a concentração de requisições falsas na vítima e, ao mesmo tempo, reduzir o consumo de recursos na rede [Zargar et al. 2013]. Muitos mecanismos híbridos sofisticados agem de forma distribuída ou requerem cabeçalhos adicionais nos pacotes da rede, o que é lento e provocam sobrecarga.

Este artigo propõe o FlowFence, um sistema de prevenção de congestionamento baseado em Redes Definidas por Software (SDN) para mitigar ataques de negação de

*Este trabalho foi realizado com recursos da CNPq, CAPES, FAPERJ e Colciencias.

serviço por inundação. O FlowFence aplica um simples controle de banda para reduzir o impacto de ataques sem a necessidade de adição de novos campos no cabeçalho dos pacotes. O FlowFence se baseia em um controlador SDN que monitora o nível de ocupação das interfaces e atua nos roteadores direcionando pacotes para filas com largura de banda limitada, quando um excesso de tráfego é detectado em uma interface. Quando um congestionamento é detectado, o roteador notifica o controlador e o controlador envia comandos aos roteadores para limitar o uso de banda nas interfaces congestionadas. Os fluxos com consumo de banda maior que o uso equitativo são punidos com menor banda designada, aplicando uma redução exponencial à diferença entre o uso de banda atual e o uso equitativo. Um protótipo do FlowFence foi implementado no *Future Internet Testbed with Security - FITS* [Moraes et al. 2014]. O protótipo foi avaliado e os resultados mostram que o FlowFence evita a inanição dos usuários legítimos em presença de ataques de negação de serviço com altos volumes de inundação de pacotes.

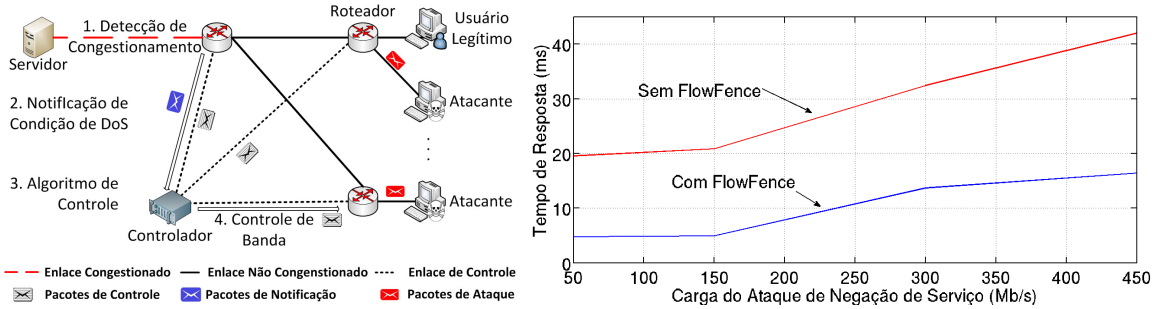
Lim *et al.* propõem um mecanismo SDN para bloquear ataques de negação de serviço distribuídos usando-se barreiras de custo computacional alto, como CAPTCHAs [Lim et al. 2014]. Contudo, essa proposta modifica a pilha convencional TCP/IP. Braga *et al.* propõem um Sistema de Detecção de Intrusão baseado em Mapas Auto Organizados (*Self Organizing Maps* - SOM) [Braga et al. 2010]. Mattos e Duarte propõem o XenFlow um mecanismo para garantir Qualidade de Serviço em redes virtuais [Mattos e Duarte 2014], do qual o FlowFence usa o mecanismo de administração de banda por fila.

2. O Sistema FlowFence Proposto

Considera-se que os atacantes são usuários que geram fluxos que podem afetar o desempenho da rede, devido ao uso malicioso ou devido a tráfegos não esperados. É assumido que o atacante não compromete o controlador ou os roteadores.

O controlador SDN mantém conexões seguras com os roteadores usando interfaces com o plano de controle. Se um roteador detecta uma condição de congestionamento em uma de suas interfaces, o roteador envia uma mensagem ao controlador que verifica a topologia da rede e solicita estatísticas de cada roteador no caminho dos fluxos que passam pela interface congestionada. A centralização lógica do controle em SDN possibilita a operação e, também, permite a reação rápida do controlador que se comunica com cada roteador no caminho congestionado. A ideia fundamental é realizar o controle de congestionamento da origem até o destino do fluxo, evitando uso desnecessário dos enlaces do caminho. Os roteadores respondem enviando suas estatísticas de fluxos. Depois de receber as estatísticas de fluxos de um roteador com uma interface congestionada, o controlador classifica os fluxos. Os fluxos que usam menos banda do que o uso equitativo da capacidade da interface são considerados bem comportados. Os fluxos com um uso de banda maior do que o equitativo são considerados como mal comportados. Fluxos mal comportados são punidos. Desta forma, se $bw_{ri} > C_t/n$, então $fluxo_i$ é mal comportado, onde bw_{ri} é a banda do $fluxo_i$, C_t é a capacidade do enlace e n é o número de fluxos compartilhando o enlace. Portanto, a estratégia de uso equitativo beneficia os fluxos de pouca largura de banda, pois esses causam um baixo impacto no uso de recursos da rede. Uma classificação mais detalhada requer um Sistema de Detecção de Intrusão, o que incrementaria a complexidade do FlowFence e, portanto, fora do escopo deste trabalho.

Uma vez que os fluxos são classificados, o controlador envia ao roteador um co-



(a) Procedimento da mitigação da condição de congestionamento.

(b) Resultados de tempo de resposta de uma requisição de conteúdo usando o `Httperf`.

Figura 1. O sistema FlowFence. (a) Cenário de teste. 1) o roteador detecta a condição de congestionamento em uma interface e 2) notifica o controlador. 3) O controlador calcula a divisão justa de banda para cada fluxo e 4) aplica o controle de banda em todos os roteadores. (b) Resultado de tempo de resposta de uma requisição de conteúdo na condição de ataque. Um cliente legítimo recebe uma resposta até quatro vezes mais rápido do que receberia sem o FlowFence.

mando para criar uma fila para cada fluxo e atribui um valor de banda independente a cada fila. Os fluxos bem comportados recebem a banda de acordo com

$$bw_i = bw_{ri} + (bw_{extra}/n_{good}),$$

onde bw_{i} é a banda usada pelo $fluxo_i$, bw_{extra} é a banda restante no enlace após a atribuição de banda, e n_{good} é o número total de fluxos bem comportados.

A implementação do FlowFence atribui banda aos fluxos como se segue. Primeiro, ocorre a atribuição de banda aos fluxos bem comportados. Depois, atribui-se a banda aos fluxos classificados como mal comportados, os quais são punidos de forma exponencial ao seu excesso. Finalmente, distribui a largura de banda restante entre os fluxos bem comportados. Os fluxos classificados como mal comportados, recebem banda por

$$bw_i = bw_r/n_{bad} - (1 - e^{-(bw_{ri} - (C_t/n))}) * \alpha * bw_{ri},$$

onde bw_{r} é a banda restante no enlace após a atribuição de banda aos fluxos bem comportados, n_{bad} é o número de fluxos mal comportados, C_{t} é a capacidade total da interface e α é uma constante que o administrador de rede fixa para determinar a agressividade da punição. Se $\alpha = 0$, nenhuma punição é aplicada e, se $\alpha = 1$, a punição máxima é aplicada aos fluxos mal comportados.

3. Avaliação e Resultados

Um protótipo do FlowFence foi implementado como uma aplicação no controlador POX[1] e uma aplicação Python executada pelos roteadores. Os roteadores executam Open vSwitch[2]. A aplicação Python monitora o uso das interfaces, administra a comunicação com o controlador e aplica o controle de banda. Todos os experimentos foram realizados no *Future Internet Testbed with Security* (FITS) e foram controlados com o arcabouço MAGI. FITS é uma rede de testes interuniversitária desenvolvida por universidades brasileiras e europeias [Moraes et al. 2014]. MAGI é um arcabouço, desenvolvido

[1]POX é um controlador para Redes Definidas por Software (http://www.noxrepo.org/pox/about-pox/).
[2]Open vSwitch é um comutador implementado em software (http://www.openvswitch.org/).

por DeterLab[3], para criar experimentos controláveis e replicáveis. Nos experimentos do FlowFence, cada nó foi implementado como uma máquina virtual e duas máquinas físicas hospedaram as máquinas virtuais. A topologia *dumbbell*, apresentada na Figura 1(a), foi usada nos experimentos. O enlace ao servidor foi limitado a uma capacidade de 50 Mb/s.

HttpPerf foi usado para medir o tempo de resposta para um cliente quando solicita conteúdo HTTP durante um ataque. Nos experimentos, o tamanho do conteúdo solicitado foi de 1 kB, a uma taxa de 10 requisições por segundo, com um total de 100 requisições. O experimento avalia a efetividade de FlowFence na mitigação de condições de congestionamento devidas a ataques de inundação. Os experimentos foram executados durante 60 segundos. Cada inunda o enlace com uma carga de 50 Mb/s.

A Figura 1(b) apresenta o tempo médio de resposta para as 100 requisições, nos cenários com e sem FlowFence. Os resultados mostram que o tempo de resposta é maior quando não é usada a defesa. O tempo de resposta cresce quando o número de atacantes e o volume da inundação aumentam. Quando o volume é 450 Mb/s, o tempo de resposta é de aproximadamente 40 ms, no cenário sem o FlowFence. Contudo, com o FlowFence, o tempo de resposta é 62% menor, ficando em aproximadamente 15 ms.

4. Conclusão

Este artigo propôs o FlowFence, um sistema de mitigação de negação de serviço para Redes Definidas por Software. FlowFence identifica condições de congestionamento, monitorando o nível de uso das interfaces de saída dos roteadores da rede. FlowFence rapidamente reage ao cenário de congestionamento usando controle de banda em cada fluxo da interface congestionada. FlowFence não requer cabeçalhos adicionais na pilha TCP/IP e não necessita de sistemas de detecção de intrusão. Os resultados mostram que o FlowFence reduz o tempo de resposta das requisições na rede em até 62%.

Referências

[Braga et al. 2010] 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)*, páginas 408–415, Denver, USA.

[Lim et al. 2014] Lim, S., Ha, J., Kim, H., Kim, Y. e Yang, S. (2014). A SDN-oriented DDoS Blocking Scheme for Botnet-based Attacks. Em *2014 Sixth International Conference on Ubiquitous and Future Networks*, China.

[Mattos e Duarte 2014] Mattos, D. M. F. e Duarte, O. C. M. B. (2014). XenFlow: Seamless migration primitive and quality of service for virtual networks. Em *IEEE Global Communications Conference (GLOBECOM 2014)*.

[Moraes et al. 2014] Moraes, I. M., Mattos, D. M. F., Ferraz, L. H. G., Campista, M. E. M., Rubinstein, M. G., Costa, L. H. M., de Amorim, M. D., Velloso, P. B., Duarte, O. C. M. B. e Pujolle, G. (2014). FITS: A Flexible Virtual Network Testbed Architecture. *Computer Networks*, 63:221 – 237.

[Zargar et al. 2013] 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*, 15(4):2046–2069.

[3]DETER é uma rede de testes para segurança (https://www.deterlab.net/).