IX FICE
Feira de Iniciação Científica e Extensão

DETECÇÃO E MITIGAÇÃO ANTECIPADA DE ATAQUES EM REDE DEFINIDA POR SOFTWARE

Um estudo de caso com ataque DoS

Fernando L. Moro[1]; Alexandre Amaral[2]; Ana Paula Amaral[3]; Rodrigo R. Nogueira[4]

RESUMO

Devido as demandas impostas pelo rápido crescimento da Internet, novos dispositivos foram incorporados na rede, tornando-a heterogênea. Consequentemente, o gerenciamento de rede e a detecção e mitigação de ataques tornaram-se complexos. Visando mitigar estes problemas e propor novos serviços, surgiu a rede definida por software. Contudo, um ataque direcionado pode causar a indisponibilidade da infraestrutura e serviços deste paradigma, devido a sua forma de funcionamento. Neste trabalho é proposto uma mecanismo formado por três módulos, responsável por detectar e mitigar um ataque DoS em seu estágio inicial em uma SDN. Para este fim, foi considerado três aspectos principais para o desenvolvimento da solução e realizado um estudo de caso para validá-la.

Palavras-chave: Detecção de anomalias. Mitigação de ataques. Rede definida por software. Segurança da informação.

INTRODUÇÃO

Os relatórios digitais da *We Are Social* e *Hootsuite* em 2018, mostraram que do total 7.59 bilhões de pessoas no mundo, 4.02 bilhões estão conectadas em rede (KEMP, 2018). Estes dados demonstram a importância da infraestrutura de rede e a dependência dos serviços prestados por ela para a sociedade. Para acompanhar o crescimento e as demandas da Internet, novos dispositivos de rede produzidos por diferentes fabricantes com variados sistemas operacionais e interfaces proprietárias foram incorporados, tornando-a heterogênea (PANDIKUMAR *et al.*, 2017).

O gerenciamento de rede, assim como, as tarefas de detecção e mitigação de ataques se tornaram uma tarefa complexa em virtude das inúmeras configurações que cada dispositivo de rede possui. Mediante a falta de homogeneidade e a necessidade de novas soluções, surgiu a rede definida por

1 Cursando Bacharelado em Sistemas de Informação. Instituto Federal Catarinense – Campus Camboriú. fernandoluizmoro@gmail.com.
2 Doutor em Engenharia Elétrica. Instituto Federal Catarinense – Campus Camboriú. alexandre.amaral@ifc.edu.br.
3 Doutora em Ciência da Computação. Instituto Federal Catarinense – Campus Camboriú. ana.amaral@ifc.edu.br.
4 Mestre em Ciência da Computação. Instituto Federal Catarinense – Campus Camboriú. rodrigo.nogueira@ifc.edu.br.

software (*Software-Defined Networking* – SDN) que promove um controle melhor e mais simplificado sobre os recursos da rede (MOUSAVI, 2014).

Uma das principais vantagens da arquitetura SDN é o controlador, um elemento central responsável pelas funções de controle. Outrora, esta funcionalidade era definida pelos fabricantes e implementada nos dispositivos de rede com as funções de encaminhamento, impedindo assim, a criação de novos protocolos, políticas e soluções. Por meio desta separação, o controle da rede pode ser programável e a sua infraestrutura abstraída para as aplicações e os serviços de rede (OPEN NETWORKING FOUNDATION, 2018).

Em uma SDN tradicional se o controlador falhar toda a rede poderá ficar indisponível e devido a este fator, ele se torna um ponto de falha e alvo de ataques (DACIER *et al.*, 2017)(QIAO *et al.*, 2015). Em detrimento de um ataque DoS (*Denial-of-Service*) do tipo *flooding* por exemplo, a comunicação realizada pelo protocolo OpenFlow entre a camada de controle e a de infraestrutura será sobrecarregada afetando todo o funcionamento da rede. Neste cenário se torna crucial o desenvolvimento de uma estratégia para detectar o ataque nos primeiros fluxos, a fim de permitir a execução de contramedidas pela gerência para minimizar o impacto na rede (CARVALHO *et al.*, 2016).

É proposto neste trabalho uma solução para detectar um ataque DoS em seu estágio inicial e aplicar ações de reparo para evitar a sobrecarga do controlador e consequentemente da infraestrutura da rede. Foram desenvolvidos três módulos principais fundamentados em três principais pontos. A escolha dos fluxos IP como fonte de dados que objetiva minimizar o volume de dados coletados, processados e armazenados. O ajuste do parâmetro entropico do detector que possibilita controlar a sensibilidade da detecção e a execução automatizada de uma ação para interromper o ataque.

PROCEDIMENTOS METODOLÓGICOS

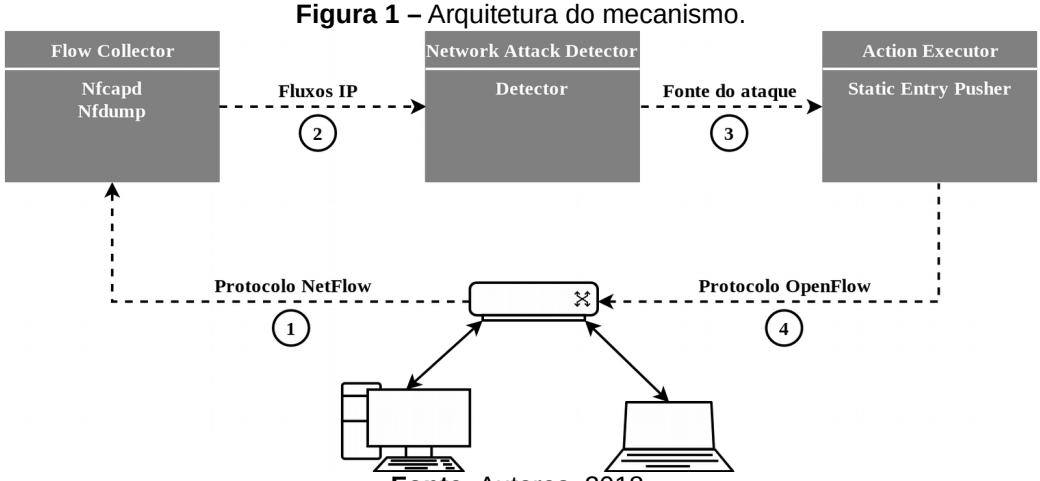
Este trabalho foi desenvolvido no Instituto Federal Catarinense – Campus Camboriú vinculado ao projeto de pesquisa Redução da intervenção humana no processo de diagnóstico e solução de anomalias em redes definidas por software – Parte 1, aprovado no edital nº 007/GDG/IFC-CAM/2017. Inicialmente foi realizada uma pesquisa bibliográfica sobre o funcionamento de uma rede definida por software e sobre as áreas de detecção e mitigação de anomalias de rede. Após, foram pesquisados trabalhos relacionados a estes temas e foi avaliado como o detector

desenvolvido por Amaral *et al.* (2017) seria implementado em um ambiente SDN. Foram pesquisadas ferramentas para a simulação de uma SDN, execução de ataques reais e coleta e processamento de dados.

Para validar a proposta foi realizado um estudo de caso com ataques DoS do tipo TCP *flooding* em uma rede definida por software virtual simulada através da ferramenta Mininet. A topologia de rede desenvolvida contém 5 *switches* virtuais habilitados com a tecnologia OpenFlow, 9 *hosts* e o controlador Floodlight. O ambiente criado foi executado em uma máquina virtual com sistema operacional Ubuntu 16.04 em um computador Intel i7-4510U de 2.00GHz, com 8 GB de memória e 1 TB de disco rígido.

Foi desenvolvido uma solução dividida em três módulos principais que são acionados ao término de cada etapa conforme apresentado na Figura 1.

Figura 1 – Arquitetura do mecanismo.



Fonte: Autores, 2018.

Primeiramente os *switches* OpenFlow foram habilitados para exportarem os fluxos IP através do protocolo Netflow v9, responsável por coletar e gerar a fonte de dados contendo as propriedades como data, hora, duração, protocolo, IP e porta de origem e destino, quantidade de pacotes, quantidade de bytes e quantidade de fluxos. Os dados exportados são enviados para uma máquina específica onde o primeiro módulo denominado *Flow Collector* (FC) realiza a coleta por meio da ferramenta Nfcapd. Porém, os dados gerados por esta ferramenta estão em um formato ilegível para serem processados pelo detector e para convertê-los, foi utilizado o Nfdump.

O segundo módulo denominado *Network Attack Detector* (NAD) possui o objetivo de detectar e classificar o ataque através da análise dos fluxos IP coletados pelo primeiro módulo. Uma das principais caraterísticas deste módulo, consiste no ajuste do parâmetro entrópico da entropia de Tsallis, que possibilita controlar a

sensibilidade do detector permitindo a identificação do ataque em uma curta janela de tempo e com poucos fluxos gerados. Este módulo foi desenvolvido em um trabalho anterior (AMARAL *et al.*, 2017) e os detalhes de implementação não serão abordados neste trabalho. O NAD retorna para o próximo módulo o tipo de ataque detectado, o IP atacante e o IP do dispositivo alvo.

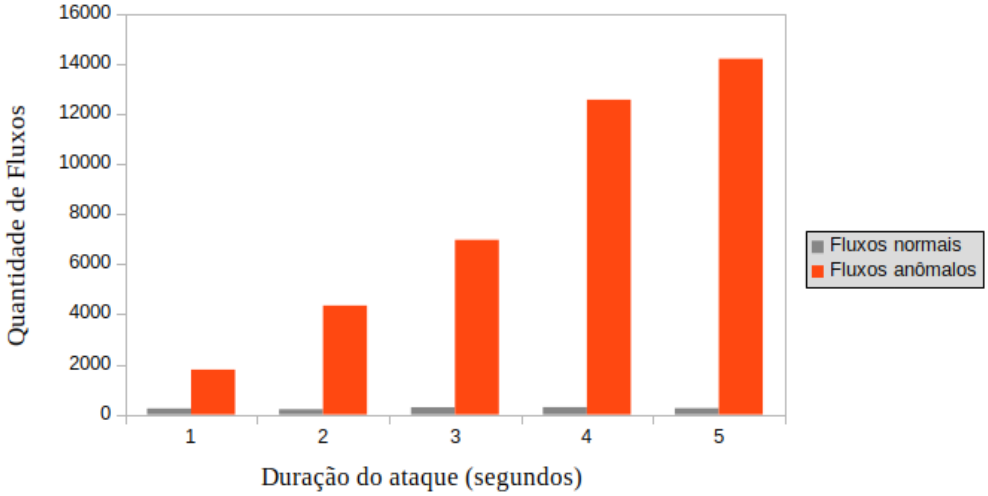
O último módulo chamado *Action Executor* (AE) é responsável pela aplicação de ações automatizadas de reparo. O AE baseia-se no módulo *Static Entry Pusher* (SEP) do controlador Floodlight, que permite adicionar, remover e consultar as regras de fluxos nos *switches* OpenFlow utilizando uma API REST (*Representational State Transfer Application Programing Interface*) (IZARD, 2017). Desta maneira, ao receber os dados de entrada, o AE insere uma regra na tabela de fluxo do *switch* OpenFlow onde originou-se o ataque com o objetivo de bloquear todos os fluxos gerados pelo atacante para o dispositivo alvo. Este procedimento representa um dos benefícios oriundos do paradigma SDN que possibilita o acesso e o controle dos dispositivos de rede de forma homogênea.

RESULTADOS E DISCUSSÃO

Com o objetivo de detectar a ocorrência do ataque nos primeiros fluxos foi optado pela utilização da janela temporal de 60 segundos (BRAUCKHOFF *et al.,* 2012). Contudo, observou-se que um ataque de rede pode ocupar toda uma janela, assim como, apenas uma fração dela. Caso o número de fluxos gerados pelo ataque seja demasiadamente pequeno, uma nova janela de tempo terá que ser aguardada postergando a detecção e consequentemente sobrecarregando o controlador.

Para contornar esta questão e avaliar a sensibilidade do detector foi executado cinco vezes um ataque DoS do tipo TCP *flooding* com duração variando entre 1 a 5 segundos dentro da janela temporal. Foi configurado o envio de 100 pacotes por segundo com porta de origem aleatória para uma porta de destino fixa do dispositivo alvo. Foram testados os valores entrópicos do detector que variaram entre -1.3 a 1.3 e o valor ideal encontrado que permitiu a detecção com a menor duração do ataque foi de 1.0. A Figura 2 mostra a quantidade de fluxos normais e anômalos detectados pelo módulo NAD em cada duração do ataque.

Figura 2 – Quantidade de fluxos detectados em cada duração do ataque.



Fonte: Autores, 2018.

Para mensurar a taxa de sucesso e fracasso obtido na detecção foram utilizados dois indicadores apresentados em Bhuyan *et al.* (2014), o TPR (*True Positive Rate*) e o FPR (*False Positive Rate*). Com base nestas métricas, o NAD apresentou para os cinco ataques realizados utilizando um valor entrópico $q = 1$ uma TPR de 100%, enquanto que FPR, foi de 0%. Em cada um dos experimentos foram geradas as informações necessárias para a execução de contramedidas ao ataque. De forma simples, mas eficaz, o IP atacante foi bloqueado com sucesso através da regra inserida diretamente no *switch* OpenFlow pelo AE. Em todos os casos, o dispositivo alvo continuou a comunicação com os demais *hosts* sendo bloqueado somente a comunicação unidirecional proveniente da máquina atacante para o alvo.

CONCLUSÕES

Foi apresentado neste trabalho uma solução para detectar e mitigar um ataque DoS do tipo TCP *flooding* antecipadamente, ou seja, em seu estágio inicial. Através da utilização dos fluxos IP e da janela temporal de 60 segundos, foi possível obter uma maior escalabilidade no processo de coleta dos dados e detecção do ataque. Através dos ajustes da sensibilidade do detector foi possível identificar em 60 segundos um ataque com a duração variando entre 1 a 5 segundos. Somado a isto, a solução apresentada executou de forma automatizada uma ação de bloqueio para impedir a continuidade do ataque, preservando assim, a disponibilidade do controlador e da infraestrutura SDN. Como trabalhos futuros, está em desenvolvimento um mecanismo baseado em *machine learning* com o objetivo de inferir a ocorrência de ataques utilizando dados e padrões históricos.

REFERÊNCIAS

AMARAL, A. A. et al. Deep IP flow inspection to detect beyond network anomalies. **Computer Communications**, v. 98, p. 80-96, 2017.

BHUYAN, M. H.; Bhattacharyya, D. K; Kalita, J. K. Network Anomaly Detection: Methods, Systems and Tools. **IEEE Communications Survey & Tutorial**, vol. 16, no. 1, p. 303-336, 2014.

BRAUCKHOFF, D. et al. Anomaly Extraction in Backbone Networks Using Association Rules. **IEEE/ACM Transactions on Networking**, v. 20, n. 6, p. 1788-1799, 2012.

CARVALHO, L. F. et al. Unsupervised learning clustering and self-organized agents applied to help network management. **Expert Systems with Applications**, v. 54, p. 29-47, 2016.

DACIER M. C. et al. Security Challenges and Opportunities of Software-Defined Networking. **IEEE Security & Privacy**, v. 15, n. 2, p. 96-100, 2017.

IZARD, R. **How to add a REST API to a Module.** 2017. Disponível em: <https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/pages/15040589/How+to+add+a+REST+API+to+a+Module>. Acesso em: jul. 2018.

KEMP, S. **Digital in 2018:** World's internet users pass the 4 billion mark. 2018. Disponível em: <https://wearesocial.com/blog/2018/01/global-digital-report-2018>. Acesso em: jul. 2018.

MOUSAVI., S. M. **Early Detection of DDoS Attacks in Software Defined Networks Controller.** Thesis (Master), Ottawa, Ontario: Carleton University, 2014.

OPEN NETWORKING FOUNDATION. **SDN Overview.** 2018. Disponível em: <https://www.opennetworking.org/sdn-definition/>. Acesso em: jul. 2018.

PANDIKUMAR, T.; ATKILT F.; HASSEN, A. Early Detection of DDoS Attacks in a Multi-Controller Based SDN. **International Journal of Engineering Science and Computing**, v. 7, n. 6, p. 13422-13429, 2017.

QIAO, Y. et al. Software-defined networking and Distributed Denial-of-Service attacks ind cloud computing environments: a survey, some research issues, and challenges. **IEEE Communications Surveys & Tutorials**, v. 18, n. 1, p. 602-622, 2015.