CAPÍTULO 13

FUZDETECT: SISTEMA DE DETECÇÃO E CLASSIFICAÇÃO DE ATAQUES DE NEGAÇÃO DE SERVIÇO

Data de aceite: 03/04/2023

Ariane Ventura
Programa de Pós Graduação em Informática (PPGI), Universidade Federal da Paraíba
João Pessoa-PB, Brasil

Vivek Nigam
Programa de Pós Graduação em Informática (PPGI), Universidade Federal da Paraíba
João Pessoa-PB, Brasil

Iguatemi Fonseca
Programa de Pós Graduação em Informática (PPGI), Universidade Federal da Paraíba
João Pessoa-PB, Brasil

RESUMO: Ataques de negação de serviço podem variar de acordo com sua natureza, porém todos eles têm o objetivo de impedir que o servidor alvo receba novas solicitações de usuários legítimos. Este artigo propõe o FuzDetect, um novo sistema que não apenas alerta quando um ataque está acontecendo, mas também o classifica. O sistema FuzDetect recolhe dados de um Rede Definida por Software e em seguida os repassa para um sistema de classificação com Lógica Fuzzy. Tal classificação é capaz de se adaptar à rede dinamicamente, com o uso da Otimização por Enxame de Partículas. Nossos resultados, com diferentes tipos de tráfego, demonstraram a eficiência do método proposto.
PALAVRAS-CHAVE: SDN, DDoS, Lógica Fuzzy, PSO.

1 | INTRODUÇÃO

Ataques de negação de serviço, quando executados com várias máquinas agindo como atacantes, de forma distribuída, são reconhecidos como Ataques de Negação de Serviço Distribuídos (DDoS} - Distributed Denial of Service Attacks). Conforme o Cert. BR(1), estes ataques podem ser divididos em: **ataques de exaustão**, cujo objetivo é saturar a capacidade dos recursos computacionais (memória, processamento e outros) e **ataques volumétricos** que tentam exaurir a banda disponível do alvo. Este artigo propõe o FuzDetect cujas contribuições são: **detectar e categorizar ataques**, acima mencionados, por meio da Lógica *Fuzzy*. Em Lógica *Fuzzy*,

os valores entre o intervalo [0,1] são usados para representar graus de associação, diferentemente da lógica de conjuntos clássica. Seu principal benefício é aproximar-se ao comportamento de sistemas onde as relações numéricas são complexas (2). Uma outra contribuição é a **adaptação do classificador *fuzzy* à rede**, uma vez que cada rede possui suas particularidades de tráfego e isto é obtido por meio da Otimização por Enxame de Partículas (PSO - *Particle Swarm Optimization*). PSO tem sido usado com sucesso em outras aplicações onde os problemas são de alta complexidade (3). E por último, **coletar dados da rede de forma leve** (dados de entrada do classificador), sem inspeção de pacotes, por meio de fluxos de uma Rede Definida por *Software* (SDN - *Software Defined Networking*) (4).

Em Mondal *et al.* (5) é apresentado um esquema de detecção com Lógica *Fuzzy*, porém a quantidade de regras é alta (aumentando o tempo computacional), a quantidade de parâmetros analisados é baixa, facilitando falsos negativos e nenhum mecanismo é utilizado para que o sistema pudesse se adaptar à rede. Dickerson *et al.* (6*)* propõe um sistema de detecção de ataques DDoS, entretanto o mesmo utiliza inspeção de pacotes, além disso, os ataques detectados não são especificados, tampouco é apresentado algum mecanismo que torne o classificador adaptado à rede analisada.

2 | FUZDETECT - FUNCIONAMENTO

Nesta etapa foi montado um ambiente virtual com variações de tráfego similar a uma rede real. A partir deste tráfego, variáveis foram coletadas (por meio de fluxos SDN), estabelecidas e utilizadas como entrada pelo sistema.

a) ***Estabelecimento das variáveis de entrada* - Variância de portas (VP) e *bytes* (VB):** durante os ataques, a dispersão em função da média foi alterada, tanto quanto aos *bytes* como quanto a média de portas. **Porcentagem de *bytes* em portas (PBP):** foi identificado, ao longo dos experimentos, que existem variações na relação entre a média de portas e a média de *bytes*, de acordo com o tipo de tráfego. **Mediana de *bytes* (MB) e pacotes (MP):** as motivações no uso destas encontram-se no trabalho de Braga *et al.* (7).

b) ***Sistema de classificação - Lógica Fuzzy:*** Na Tabela I, as variáveis de entrada representam as variáveis mencionadas anteriormente e os seus respectivos conjuntos: A - alto, B - baixo e M - médio e as variáveis de saída são: AV - ataque volumétrico, AE - ataque de exaustão e TNM - tráfego não malicioso, estas últimas também com seus respectivos conjuntos. Seguem as regras usadas no sistema *Fuzzy* aqui proposto, conforme Tabela II. A conversão das siglas é a mesma da Tabela I e o arranjo destas regras foi feito de forma empírica, a partir de observações acerca do tráfego. De acordo com a Tabela II, na primeira linha, é definido que se **MP** é **baixa** e **VP** é **alta** ou **MB** é **alta** e **PBP** é **alta**, então, **AV: alta** chance de acontecer, **AE:** chance **baixa** e para **TNM: média** chance (siglas mencionadas anteriormente). No processo de saída das regras (*defuzzyficação*), é gerada uma

saída numérica, informando assim, as chances dos tipos de tráfego mencionados estarem ocorrendo.

Variáveis de Entrada					Variáveis de Saída		
VP	VB	MB	MP	PBP	AV	AE	TNM
B	B	B	B	B	B	B	B
M	M	-	-	-	M	M	M
A	A	A	A	A	A	A	A

Tabela I - VARIÁVEIS DE ENTRADA E DE SAÍDA E SEUS RESPECTIVOS CONJUNTOS

c) ***Otimização do FuzDetect – PSO:*** O PSO é utilizado a fim de tornar o classificador adaptado à rede, por meio de ajustes nos conjuntos *fuzzy* e isto é obtido por meio de tráfego previamente identificado. Quanto mais acertos na classificação do tráfego, por conjuntos testados, maior a chance destes serem usados no classificador. Para mais detalhes acerca do PSO no FuzDetect, consultar (12) (Seção 4.0.5 e 5.3.1).

Regras	Tráfego
((MP – B AND VP - A) OR (MB – A AND PBP - A))	(AV - A) (AE - B) (TNM - M)
((VP - A AND VB - B) OR (MP - B AND PBP -B))	(AV - M) (AE - A) (TNM - B)
(((VP - B OR VP - M) AND (MP - A)) OR (MB - A AND MP - A))	(AV - B) (AE - M) (TNM - A)

Tabela II - REGRAS *FUZZY* DO FUZDETECT

3 I EXPERIMENTOS, RESULTADOS E CONSIDERAÇÕES FINAIS

Os *hosts* e *switches* virtuais que simularam clientes legítimos e atacantes foram executados no Mininet. Para a simulação de ataques volumétricos, utilizou-se o *dnsdrdos (*8) e o *Hping3* (9). Já para a execução de ataques de exaustão do serviço, o *Hping3* também. Por último, para simular conexões de clientes legítimos, foi utilizado o *Siege* (10) e um *script* HTTP foi criado usando a biblioteca Python, Urllib3 (11) (nas referências destas ferramentas são apresentados detalhes sobre a configuração e parametrização das mesmas, respectivamente). Foram executados dois experimentos diferentes na topologia apresentada na Fig. 1. De acordo com a mesma, no experimento I, durante os ataques volumétricos (fluxo identificado por um redirecionador não pontilhado), o tráfego de ataque

se originou em S1, onde o atacante acessou servidores DNS com alto fator de amplificação e retornou a resposta ao alvo, no *switch* S5. Já no experimento II, durante os ataques por exaustão (redirecionador com linha pontilhada), o ataque se originou em S5 e possuiu como alvo um servidor *web* em S0. Demais *switches* da rede geraram exclusivamente tráfego legítimo (linha sem redirecionador e sem estar pontilhada), exceto os *switches* S0, S1, S4 e S5 que simultaneamente receberam e repassaram tráfego legítimo e de ataque, por experimento.

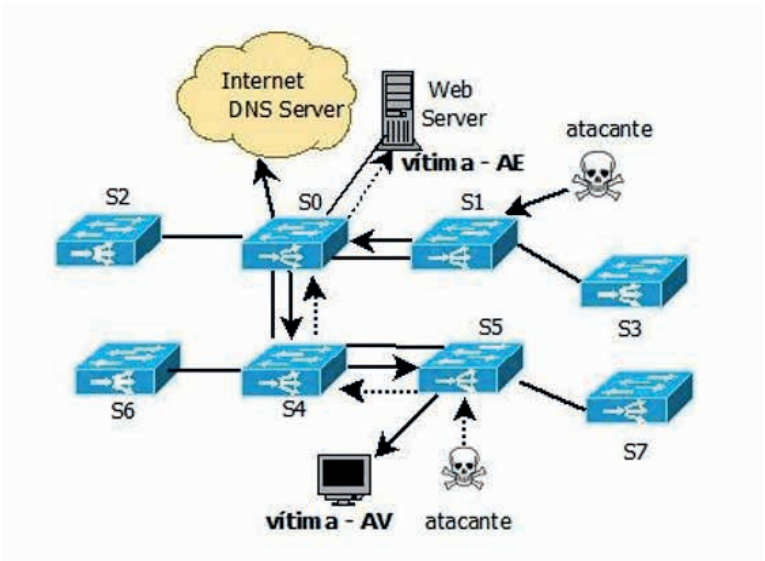

Figura 1. Topologia I

Na Tabela III, os resultados estão separados por experimento, podendo ser AV - ataques volumétricos ou AE - ataques de exaustão e por *switch*, de acordo com a Fig. 1. Para cada experimento / *switch* o valor da variável de saída esperada é destacado em negrito (devendo o valor desta ser maior que as demais, em todos os casos). Por exemplo, na Tabela III, no experimento I (AV), foi esperado que no *switch* S0 o tráfego de maior predominância identificado tenha sido o de ataque volumétrico e foi, com 64,5% de chance de ocorrer, já no experimento II (AE), esperou-se que em S1 o tráfego não malicioso tenha sido o de maior predominância e assim ocorreu (com 77% de chance de acontecer).

Neste trabalho, falsos positivos corresponderam a classificação errônea, em que em um *switch* trafegando unicamente dados legítimos, o classificador identifique este como de ataque. Já para falsos negativos, qualquer *switch* com tráfego de ataque, mas que não tenha sido considerado como tal. O FuzDetect foi preciso em todas as classificações, para todos os casos a saída em negrito sempre foi maior que as demais, indicando assim a precisão na identificação do tráfego.

experimento I - AV			experimento II - AE			Switch
AV	AE	TNM	AV	AE	TNM	
64,5%	31,8%	53%	24,7%	**74,7%**	50%	S0
74,7%	24,7%	24,7%	30%	16%	**77%**	S1
57%	42,9%	**59,2%**	25%	50%	**75%**	S2
50%	50%	**59,2%**	30%	18%	**78%**	S3
74,4%	24,7%	24,1%	20,1%	**79,2%**	50%	S4
73%	22%	53,7%	16,6%	**82,6%**	50%	S5
56,1%	43,5%	**62%**	25%	50%	**80%**	S6
44,4%	36,1%	**63,3%**	28,3%	50%	**75%**	S7

Tabela III - RESULTADOS - TOPOLOGIA I - EXPERIMENTO I E II

REFERÊNCIAS

(1) CertBR - Ataques Distribuídos de Negação de Serviço (DDoS) - 2016. Disponível em: https://www.cert.br/docs/whitepapers/ddos. Acessado em: 1 de jul. 2018.

(2) T. J. Ross. Fuzzy Logic: With Engineering Applications. Third Edition. West Sussex, United Kingdom: WILEY, 2010.

(3) K. Chia-Nan and W. Chia-Ju. A PSO-Tuning Method for Design of Fuzzy PID Controllers. Journal of Vibration and Control, Los Angeles, CA, USA, v. 14, n. 3, p. 375-395, Marc, 2008.

(4) T. D. Nadeau e K. Gray. SDN: Software Defined Networks: An Authoritative Review of Network Programmability Technologies. First Edition. Gravenstein Highway, Sebastopol: "O'Reilly Media, Inc.", 2013.

(5) Mondal, H. Shekhar and Hasan, Md Tariq and Hossain, M. Bellal and Rahaman, M. Ekhlasur and Hasan e Rabita. Enhancing Secure Cloud Computing Environment by Detecting DDoS Attack Using Fuzzy Logic. Electrical Information and Communication Technology (EICT), 2017 3rd International Conference on, Khulna, Bangladesh, pages 1--4. IEEE, Dec, 2017.

(6) J. E. Dickerson e J. A. Dickerson. Fuzzy Network Profiling for Intrusion Detection. Fuzzy Information Processing Society 2000. NAFIPS. 19th International Conference of the North American, Atlanta, GA, USA, pages 301–306. IEEE, Jul, 2000.

(7) R. Braga, E. Mota e A. Passito. Lightweight DDoS Flooding Attack Detection using Nox/Openflow. Local Computer Networks (LCN) 2010, IEEE 35th Conference on, Denver, CO, USA, pages 408–415. IEEE, Oct, 2010.

(8) DDoS Amplification Attacks - 2018. Disponível em: https://www.noction.com/blog/ddos-amplification-attacks. Acessado em: 19 de Ago. 2018.

(9) Hping usage examples - 2009. Disponível em: https://www.rationallyparanoid.com/articles/hping.html. Acessado em: 19 de Ago. 2018.

(10) Load Testing Web Servers - 2015. Disponível em: https://www.linode.com/docs/tools-reference/tools/load-testing-with-siege/. Acessado em: 19 de Agp. 2018.

(11) Urllib3 Documentation - 2018. Disponível em: https://urllib3.readthedocs.io/en/1\.5/\#connectionpool. Acessado em: 19 de Ago. 2018.

(12) A. Falcão, \emph{FuzDetect: Sistema de Detecção e Classificação de Ataques de Negação de Serviço}. 2018. 91f. Qualificação de Mestrado - UFPB, PB, 2018. https://www.dropbox.com/s/91bl63tl9ppg36o/doc.pdf?dl=0