Sistema Conteinerizado de Simulação para Algoritmos de Detecção de Ataques em Redes Definidas por Software

Matheus B. Pivetta[1], **Thiago dos S. Cavali**[2], **Keiko V. O. Fonseca**[2], **Mauro Sergio P. Fonseca**[2]

[1] Departamento Acadêmico de Eletrônica
Universidade Tecnológica Federal do Paraná (UTFPR)
Campo Mourão, PR – Brasil

[2]Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial
Universidade Tecnológica Federal do Paraná (UTFPR)
Curitiba, PR – Brasil

matheuspivetta@alunos.utfpr.edu.br
{thiagocavali,keiko,maurofonseca}@utfpr.edu.br

Abstract. *Various types of attacks pose threats to Software Defined Networks (SDN). The application of Machine Learning (ML) methods has emerged as a promising approach to detecting attacks, however, a performance evaluation of these methods is essential. This research proposes the development of a container-based simulator to evaluate the performance of ML algorithms in detecting attacks on SDN. The system was tested through a proof of concept, and the results obtained were consistent with those reported in the literature. Furthermore, the system proved to be lightweight, easily adaptable, and versatile for use in diverse attack scenarios and network topologies.*

Resumo. *Diferentes tipos de ataque representam uma grande ameaça às Redes Definidas por Software (SDN). Nesse contexto a utilização de métodos de aprendizado de máquina (ML) mostra-se uma boa alternativa para detecção desses ataques, porém se faz necessária uma avaliação do desempenho desses algoritmos. Assim, esse trabalho propõe o desenvolvimento de uma ferramenta de simulação baseada em contêineres para avaliação do desempenho de algoritmos de ML para detecção de ataques em SDN. O sistema proposto foi testado com uma prova de conceito e os resultados obtidos foram compatíveis com a literatura. Além disso, o sistema se mostrou leve, facilmente adaptável e versátil para utilização nos mais diversos cenários de ataque e topologias de rede.*

1. Introdução

Em redes convencionais o fluxo de dados e o tratamento desse fluxo ocorrem no mesmo dispositivo, o que leva a desafios ao lidar com muitos dispositivos conectados, apresentando limitações de escalabilidade, mudanças nas políticas, dependência do fabricante, entre outros [Alhijawi et al. 2022]. Nas Redes Definidas por Software (SDN), o controle do fluxo de dados é desvinculado dos dispositivos que efetuam o tráfego, sendo o controle da rede centralizado, permitindo a administração de pacotes em tempo real e a resolução eficiente dos desafios inerentes ao paradigma convencional [Aslam et al. 2022].

As SDN não estão isentas de ataques, assim como as redes convencionais. Ataques de negação de serviço (DoS) e DoS distribuída (DDoS) representam uma

ameaça particularmente danosa para o controlador de uma SDN. Dado que o controle é centralizado, a saturação do controlador pode resultar na queda completa da rede [Alhijawi et al. 2022]. Nesse contexto, a literatura nos últimos anos explora o aprendizado de máquina (ML) como uma ferramenta para identificação de ataques a redes, já que sua utilização possibilita o desenvolvimento de modelos de reconhecimento de padrões [Yungaicela-Naula et al. 2022]. O modelo pode ser pré-treinado com um conjunto de dados de fluxos de pacotes de rede, classificados e rotulados como legítimos ou maliciosos, e posteriormente implementado em conjunto com o controlador para avaliar em tempo real a legitimidade de um fluxo [Aslam et al. 2022].

Diante do cenário exposto, o uso de ferramentas que permitam testar diferentes formas de aprimorar a segurança das SDN se faz importante. Espera-se que essas ferramentas sejam adaptáveis, com fácil configuração e que utilizem recursos de hardware acessíveis aos pesquisadores de segurança de todos os níveis. Nesse contexto, esse estudo visa apresentar uma ferramenta de simulação, baseada em contêineres, que permita avaliar modelos de ML para detecção de ataques em ambientes SDN. Será demonstrada a aplicação da ferramenta mediante uma prova de conceito, a fim de mostrar que um algoritmo de ML conectado a aplicação é capaz de realizar a detecção de um ataque a uma SDN obtendo, até o momento, resultados compatíveis com os presentes na literatura.

O documento está organizado da seguinte forma: a Seção 2 faz uma breve revisão da literatura. A Seção 3 apresenta o sistema proposto, sua estrutura e detalhes de implementação. A Seção 4 apresenta a prova de conceito realizada e mostra os resultados parciais obtidos. Por fim, a Seção 5 apresenta as conclusões obtidas até o momento e aponta os próximos passos no desenvolvimento do sistema.

2. Revisão da Literatura

O paradigma de SDN é baseado na separação entre o processamento e o encaminhamento dos fluxos de dados [Nunes et al. 2014]. Para tal, os *switches* se comunicam com um controlador via um protocolo específico, o controlador faz todas as decisões de encaminhamento e alimenta uma tabela de regras de fluxo presente nos *switches* que direciona os pacotes. A centralização do controle simplifica a implementação de políticas de rede, tornando a rede mais eficiente e adaptável [Chouikik et al. 2022].

Apesar das vantagens, as SDN também introduzem desafios de segurança. A centralização do controle representa uma vulnerabilidade, já que falhas de segurança que comprometam o controlador afetam diretamente a rede como um todo [Haas et al. 2021]. Além disso, as SDN estão suscetíveis a ataques como manipulação da rede, das aplicações e desvio de tráfego [Pradhan and Mathew 2020]. As SDN também podem ser alvos de ataques DoS e DDoS que podem ter como alvo os *switches* e também o controlador, sendo que no controlador o objetivo é a saturação da banda no canal de controle ou dos recursos de processamento [Haas et al. 2021].

Diversas soluções foram apresentadas na literatura para identificação dos ataques que podem ocorrer em uma SDN. Trabalhos como [Aslam et al. 2022], [Bithi et al. 2024] e [Garba et al. 2024] utilizam algoritmos como *Random Forest* (RF), XGBoost, *Support Vector Machine*, *Decision Tree* e *K-Nearest Neighbors* para detecção de ataques (principalmente do tipo DDoS) nas redes. Os resultados obtidos, em geral, apresentam um índice próximo a 100% de detecção dos ataques, sendo que um dos principais fatores

Tabela 1. Comparação dos trabalhos analisados em termos das ferramentas utilizadas, ambiente de simulação de rede, foco do trabalho e presença de módulos adaptáveis (MA).

Trabalho	Ferramentas	Ambiente	Foco	MA
[Abdullah Mohammed and Ibrahim Aldabbagh 2023]	Docker	Mininet	Controladores	X
[Kaihara et al. 2022]	Docker	Máquinas Virtuais	Desempenho	✓
[Dias et al. 2023]	Docker+Kubernetes	Mininet	Segurança	✓
[Aslam et al. 2022]	-	Mininet	Segurança	✓
[Garba et al. 2024]	-	Dispositivos físicos	Segurança	X
Esse trabalho	Docker	Mininet	Segurança	✓

para a qualidade dos resultados obtidos é o *dataset* utilizado no treinamento desses algoritmos. Trabalhos como [Sharafaldin et al. 2019] e [Elsayed et al. 2020] apresentam diferentes *datasets* que podem ser utilizados para o treinamento dos modelos.

A utilização de contêineres para simulação de aspectos relacionados à SDN também é uma tendência devido à facilidade que sua utilização traz para o uso de diferentes ferramentas e por facilitar também a escalabilidade dos sistemas [Kaihara et al. 2022]. Em [Abdullah Mohammed and Ibrahim Aldabbagh 2023], por exemplo, os autores utilizam uma simulação de SDN baseada em contêineres para verificar o comportamento da rede quando utilizado um *cluster* de controladores. Em [Kaihara et al. 2022] e [Dias et al. 2023] os autores utilizam-se de contêineres como uma forma de facilitar a simulação das SDN e seus aspectos de segurança, sendo que nenhum desses trabalhos oferece, simultaneamente, a utilização somente da plataforma Docker[1] e a praticidade da utilização do Mininet[2] como plataforma de simulação. A Tabela 1 resume os trabalhos verificados destacando as características de cada um em comparação ao presente trabalho.

Tendo em vista os diversos modelos de ML, os diversos parâmetros e *features* para treinamento desses modelos e também as facilidades oferecidas por um ambiente conteinerizado, um ambiente baseado em módulos e contêineres surge como uma solução para simulação e análise de aspectos fundamentais de segurança em SDN.

3. Metodologia

A proposta deste estudo é a construção de uma plataforma de testes baseada em contêineres Docker para testes de modelos de detecção de ataques DDoS em SDN. Através da simulação da SDN utilizando Mininet e de ataques em um ambiente controlado visa-se validar a eficácia de modelos de ML na identificação dos ataques.

A Figura 1 representa a topologia da plataforma em contêineres, sendo cada bloco um contêiner, e a sequência representa a ordem de execução. Para a implementação foi usada a ferramenta *Docker Compose* que permite a criação de aplicações com múltiplos contêineres. No contêiner Mininet[3] é feita a simulação da rede SDN usando Mininet.

[1]https://www.docker.com
[2]https://mininet.org/
[3]https://github.com/iwaseyusuke/docker-mininet

Os processos de coleta e geração de tráfego foram executados em *threads* diferentes e a captura dos pacotes é feita usando a ferramenta tshark[4].

Figura 1. Estrutura dos contêineres para o sistema proposto

Após a captura, o contêiner Wireshark[5] faz a ordenação dos pacotes para que não ocorrerem erros na etapa de processamento. Na etapa seguinte é utilizada uma versão em contêiner do CICflowmeter [Engelen et al. 2021] para obter e extrair os parâmetros necessários para o modelo de classificação. A extração ocorre em 2 contêineres para dividir o processamento dos fluxos legítimos e maliciosos.

O contêiner Sensor executa o modelo de classificação. Esse sensor é previamente treinado com qualquer *dataset* que possua dados relacionados a ataques DDoS. Com os dados processados nas etapas anteriores os fluxos são classificados entre legítimos e maliciosos e comparados com as reais classificações para se obter o desempenho do modelo, podendo as métricas de avaliação serem definidas pelo próprio usuário do sistema.

4. Prova de conceito e estudo de caso

Como prova de conceito, foi utilizado o algoritmo RF para execução do sensor. O treinamento do modelo foi feito com o *dataset* CIC-DDoS2019 [Sharafaldin et al. 2019] para ataques DDoS do tipo *TCP-SYN Flood*. A escolha do algoritmo, ataque, *dataset*, e métricas de avaliação se baseia nos trabalhos da literatura citados na Seção 2.

Devido a limitações de hardware, optou-se pela utilização de 1% do *dataset* com entradas escolhidas aleatoriamente, resultando em um *dataset* balanceado com 86636 fluxos. Desse dataset, 75% foi utilizado para treinamento e 25% para validação, sendo que após 10 treinamentos obteve-se um score R^2 de 93,35% com 95% de nível de confiança.

A topologia de rede representada na Figura 2 foi simulada no Mininet usando um *script* desenvolvido na linguagem Python para esse fim. A partir do *host* 1 realizou-se um o ataque utilizando a ferramenta hping3[6] direcionado ao *host* 2, para realizar a captura de pacotes foi utilizado espelhamento de porta para o *host* 3. Os fluxos oriundos desse ataque foram organizados pelo CICflowmeter e analisados pelo sensor juntamente com pacotes de fluxos legítimos obtidos no *dataset* TII-SSRC-23 [Herzalla et al. 2023].

Os testes foram realizados de maneira a garantir um nível de confiança de 95% nos resultados obtidos e mostram que a plataforma funcionou adequadamente para a avaliação do desempenho do modelo, obtendo um *recall* de 99,38% na identificação de um ataque. Optou-se pelo *recall* como parâmetro de avaliação pelo fato de essa métrica ser a que melhor demonstra a influência dos falsos negativos no resultado do classificador. No cenário considerado, um falso negativo representa um fluxo malicioso erroneamente classificado como legítimo, sendo esse o pior caso possível no contexto de segurança em

[4]https://tshark.dev
[5]https://github.com/linuxserver/docker-wireshark
[6]https://www.kali.org/tools/hping3

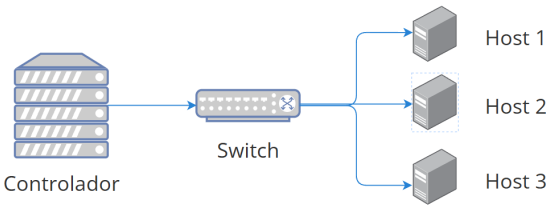

Figura 2. Topologia da rede simulada no Mininet para a prova de conceito

redes. Vale ressaltar que o resultado obtido foi compatível com o obtido em trabalhos como [Aslam et al. 2022], e [Bithi et al. 2024] sendo eles, respectivamente, 99% e 100%.

A plataforma de testes desenvolvida está disponível no repositório do Github `MatheusPivetta/dockerMininetWorkflow` e se mostrou, até o momento, facilmente adaptável em diversos aspectos. A utilização do Mininet permite que as mais diversas topologias de rede sejam utilizadas pelo usuário, além de permitir a utilização de diferentes ferramentas para simulação de ataques. O contêiner do sensor, por sua vez, pode conter qualquer algoritmo de ML e ser treinado com qualquer *dataset* disponível na literatura, permitindo diferentes análises por parte do usuário.

O ambiente computacional utilizado para execução do sistema foi uma máquina virtual com 16 GB de memória e 30 GB de armazenamento com um processador i7-12700h, demonstrando que o sistema pode ser executado em computadores pessoais, sendo o treinamento do sensor a parte mais custosa em termos de hardware. A utilização do Docker facilita a implementação, reduzindo o tempo de configuração dos componentes utilizados. Sendo assim, pode-se dizer que o sistema apresentado é uma ferramenta útil e promissora para avaliação de diferentes cenários de segurança em SDN.

5. Conclusões preliminares e próximos passos

Esse trabalho apresentou os passos desenvolvidos até o momento para a implementação de um simulador que permite a avaliação de algoritmos de ML em diferentes cenários de segurança em uma SDN. Esse simulador é baseado em contêineres para cada uma de suas partes e, por isso, se mostrou altamente versátil para testes de diferentes sensores e cenários de rede. A operação do sistema foi demonstrada com uma prova de conceito com a qual foi possível observar que os resultados obtidos foram compatíveis com a literatura, evidenciando o potencial do sistema proposto para detecção das ameaças às SDN.

Para a continuação do trabalho pretende-se melhorar a interface para interação do usuário com as diferentes partes do sistema usando interfaces gráficas ou linhas de comando específicas. Outro ponto de melhoria são questões internas do sistema, como a forma de obtenção e processamento dos fluxos de pacotes, bem como análise do uso de recursos computacionais. Pretende-se também utilizar o sistema para testes de bloqueio de ataques, permitindo que o usuário compare diferentes controladores e algoritmos de ML na detecção de fluxos maliciosos e mitigação de um ataque em tempo real.

As melhorias citadas são apenas algumas das mapeadas até o momento. Ao longo do desenvolvimento será possível verificar outras demandas de melhoria contínua até que o sistema esteja completo para utilização em salas de aula, laboratórios de pesquisa, entre outras aplicações nas quais simuladores auxiliam no processo de desenvolvimento.

6. Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Código de Financiamento 001

Referências

Abdullah Mohammed, G. and Ibrahim Aldabbagh, O. A. (2023). A comparative evaluation of the performance of sdn controllers (onos) using docker container. *Technium: Romanian Journal of Applied Sciences and Technology*, 8:25–31.

Alhijawi, B., Almajali, S., Elgala, H., Bany Salameh, H., and Ayyash, M. (2022). A survey on dos/ddos mitigation techniques in sdns: Classification, comparison, solutions, testing tools and datasets. *Computers and Electrical Engineering*, 99:107706.

Aslam, N., Srivastava, S., and Gore, M. M. (2022). Onos flood defender: An intelligent approach to mitigate ddos attack in sdn. *Transactions on Emerging Telecommunications Technologies*, 33(9):e4534.

Bithi, M., Hossain, M. A., Ahmed, M. K., Sultana, R., Ahammad, I., and Islam, M. S. (2024). Enhanced ddos detection in software defined networking using ensemble-based machine learning. In *2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, pages 1032–1037. IEEE.

Chouikik, M., Ouaissa, M., Ouaissa, M., Boulouard, Z., and Kissi, M. (2022). Software-defined networking security: A comprehensive review. *Big Data Analytics and Computational Intelligence for Cybersecurity*, pages 91–108.

Dias, V., Silva, M., Gomes, M., Oliveira, L., Abreu, D., Ferreira, J., and Abelém, A. (2023). Detecção de ataques ddos em redes sdn utilizando aprendizado de máquina: Uma abordagem em microsserviços. In *Anais Estendidos do XXIII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pages 141–152, Porto Alegre, RS, Brasil. SBC.

Elsayed, M. S., Le-Khac, N.-A., and Jurcut, A. D. (2020). Insdn: A novel sdn intrusion dataset. *IEEE access*, 8:165263–165284.

Engelen, G., Rimmer, V., and Joosen, W. (2021). Troubleshooting an intrusion detection dataset: the cicids2017 case study. In *2021 IEEE Security and Privacy Workshops (SPW)*, pages 7–12. IEEE.

Garba, U. H., Toosi, A. N., Pasha, M. F., and Khan, S. (2024). Sdn-based detection and mitigation of ddos attacks on smart homes. *Computer Communications*, 221:29–41.

Haas, Z. J., Culver, T. L., and Sarac, K. (2021). Vulnerability challenges of software defined networking. *IEEE Communications Magazine*, 59(7):88–93.

Herzalla, D., Lunardi, W. T., and Andreoni, M. (2023). Tii-ssrc-23 dataset: Typological exploration of diverse traffic patterns for intrusion detection. *IEEE Access*, 11:118577–118594.

Kaihara, A., Bondan, L., Gondim, J., Rodrigues, G., Marotta, M., and Rodrigues, G. (2022). Lst: Testbed emulado leve para redes sdn aplicado ao contexto de segurança.

In *Anais Estendidos do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 41–48, Porto Alegre, RS, Brasil. SBC.

Nunes, B. A. A., Mendonca, M., Nguyen, X.-n., Obraczka, K., and Turletti, T. (2014). A Survey of Software-Defined Networking : Past , Present , and Future of Programmable Networks. *IEEE Commun. Surv. Tutor., 16 (3), 1617-1634.*, 16(3):1617–1634.

Pradhan, A. and Mathew, R. (2020). Solutions to vulnerabilities and threats in software defined networking (sdn). *Procedia Computer Science*, 171:2581–2589.

Sharafaldin, I., Lashkari, A. H., Hakak, S., and Ghorbani, A. A. (2019). Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In *2019 International Carnahan Conference on Security Technology (ICCST)*, pages 1–8.

Yungaicela-Naula, N. M., Vargas-Rosales, C., Pérez-Díaz, J. A., and Zareei, M. (2022). Towards security automation in software defined networks. *Computer Communications*, 183:64–82.