Detecção de *malware* em ambientes IoT habilitados por SDN

Cristian H. M. Souza[1], **Carlos H. Arima**[1]

[1]Centro Estadual de Educação Tecnológica Paula Souza (CEETEPS)
São Paulo-SP, Brasil

cristianmsbr@gmail.com, carlos.arima@cpspos.sp.gov.br

Abstract. *Malicious software remain one of the main challenges for the security of computer systems. The growth of the Internet of Things (IoT) technological paradigm has raised several concerns regarding the security of devices connected to the Internet, especially in industrial environments, where compromising or malfunctioning such devices can cause damage to the physical environment and put human lives at risk. This work proposes a hybrid approach for detecting malicious artifacts in SDN-enabled IoT environments. The solution combines the use of YARA rules and machine learning for classifying malicious artifacts based on network traffic analysis. The implemented Random Forest algorithm achieved an accuracy of 99.33% on the test dataset. When evaluated against real malicious programs, the proposal achieved a detection rate of 98.44% and an average processing time of 0.0217s.*

Resumo. *Programas maliciosos continuam sendo um dos principais desafios para a segurança dos sistemas computacionais. O crescimento do paradigma tecnológico da Internet das Coisas tem gerado diversas preocupações a respeito da segurança dos dispositivos conectados à Internet, especialmente em ambientes industriais, onde o comprometimento ou mau funcionamento de tais aparelhos pode ocasionar danos ao ambiente físico e colocar vidas humanas em risco. Este trabalho propõe o uma ferramenta híbrida para detecção de artefatos maliciosos em ambientes IoT habilitados por SDN. A solução combina o uso de regras YARA e machine learning para classificação de artefatos maliciosos a partir da análise do tráfego da rede. O algoritmo Random Forest implementado obteve uma acurácia de 99.33% no conjunto de dados de teste. Ao ser avaliada contra programas maliciosos reais, a ferramenta obteve uma taxa de detecção de 98.44% e um tempo de processamento médio de 0.0217s.*

1. Introdução

Malwares continuam sendo um dos principais desafios à segurança dos sistemas computacionais. O advento do paradigma IoT foi acompanhado pelo aumento do número de programas maliciosos com foco nas arquiteturas ARM (do inglês *Advanced RISC Machine*) e MIPS (do inglês *Microprocessor without Interlocked Pipeline Stages*). Um exemplo disso é a utilização da *botnet* Mirai [Kumar and Chandavarkar 2023], cujo objetivo é infectar e controlar dispositivos embarcados, como câmeras IP e roteadores, para execução de ataques de negação de serviço distribuídos (DDoS, do inglês *Distributed Denial of Service*).

O uso de *machine learning* tem se mostrado efetivo na detecção genérica de artefatos maliciosos a nível de rede [Gaurav et al. 2023]. Para isso, um modelo é treinado

com base em uma grande quantidade de dados a respeito de ameaças conhecidas. Uma vez treinado, ele é capaz de realizar previsões assertivas ao lidar com novas informações. Isso permite a detecção e classificação efetiva das ameaças com base na análise de tráfego, minimizando o risco de comprometimentos.

Pesquisas recentes elevam o potencial das Redes Definidas por Software (SDN, do inglês *Software-Defined Networking*) por meio da construção de planos de dados programáveis [Hauser et al. 2023]. Essa abordagem possibilita a reconfiguração sistemática das etapas de processamento de baixo nível aplicadas aos pacotes da rede, reduzindo a complexidade e aprimorando a utilização dos recursos dos *switches*. Como destacado por [Liatifis et al. 2023], a próxima geração de SDN (NG-SDN, do inglês *Next-Generation SDN*) é caracterizada pelo uso de interfaces e *hardware* abertos, contrariamente às redes tradicionais, que adotam padrões proprietários e fechados. Para isso, tecnologias como a linguagem P4 estão sendo desenvolvidas para habilitar o desenvolvimento de soluções inovadoras que podem ser implantadas diretamente em *switches* programáveis [Peter et al. 2022].

Inspirado pelo poder e flexibilidade dos *switches* programáveis, este trabalho propõe uma abordagem híbrida para detecção de artefatos maliciosos em ambientes IoT habilitados pela tecnologia SDN. A abordagem é composta pela detecção de assinaturas maliciosas e pela classificação do tráfego por meio de *machine learning*. A solução desenvolvida se apresenta como uma ferramenta de grande potencial no combate a ameaças de *malware* em IoT devido a sua capacidade de identificar artefatos que ainda estão sendo transferidos para o alvo, além de detectar fluxos maliciosos gerados por *malwares* em execução e prontamente isolar os dispositivos comprometidos.

Diferentemente das abordagens tradicionais, a solução é acoplada diretamente aos *switches* da rede, o que reduz a ocorrência de pontos únicos de falha e otimiza a utilização dos recursos presentes na infraestrutura. A ferramenta é avaliada em um ambiente controlado e emulado por meio da ferramenta Mininet, juntamente com o auxílio de *switches* virtualizados. São utilizados exemplares de *malwares* reais para validar a acurácia da solução. Os resultados obtidos evidenciam que a proposta obteve uma taxa de detecção real de 98.44% e tempo de processamento médio de 0.0217s, demonstrando que soluções integradas aos *switches* da rede se apresentam como abordagens rápidas, efetivas e de baixo custo para o combate a programas maliciosos.

O restante deste artigo está estruturado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados, visando evidenciar o diferencial do presente estudo; a Seção 3 expõe os detalhes da arquitetura proposta; a Seção 4 apresenta os resultados da prova de conceito implementada; e, por fim, a Seção 5 apresenta as considerações finais e delineia apontamentos de trabalhos futuros.

2. Trabalhos relacionados

A detecção de *malwares* em ambientes IoT é de fundamental importância para ambientes corporativos e domésticos, visto que o comprometimento de tais dispositivos pode acarretar em danos físicos e na exposição de dados sensíveis. Diante disso, esta seção aborda os principais esforços de pesquisa propostos pela comunidade científica no âmbito da identificação de atividades maliciosas em ambientes IoT por meio de *machine learning*.

O trabalho de [Maeda et al. 2019] propõe um mecanismo baseado em *deep lear-*

ning para detecção de tráfego malicioso originado por máquinas infectadas por *botnets* em redes SDN. Para isso, os autores empregam uma rede neural do tipo MLP. Após a classificação, o mecanismo isola as máquinas infectadas e bloqueia conexões externas. O modelo foi treinado utilizando tráfego malicioso do *dataset* CTU-13, bem como tráfego legítimo do *dataset* ISOT. Os autores destacam que a acurácia da solução é de 99.2%. Entretanto, o artigo destaca que não foram conduzidos experimentos em terminais que estavam efetivamente infectados com *bots*, o que pode afetar a generalização dos resultados.

Em [Khan and Akhunzada 2021], os autores oferecem um mecanismo híbrido para detecção de *malwares* que exploram vulnerabilidades em equipamentos IoT, mais especificamente em aparelhos *Internet of Medical Things* (IoMT). A solução apresentada combina *deep learning* para classificação e detecção de *malwares* e utiliza a arquitetura SDN para facilitar a aquisição e coleta do tráfego da rede. A principal contribuição deste trabalho está no aumento da eficácia na detecção de *malwares* devido ao uso de dois modelos de classificação: *Convolution Neural Network* (CNN) e *Long Short Term Memory* (LSTM). Os autores destacam que esse esquema apresenta melhores resultados quando comparado a outras estratégias de classificação. O *dataset* utilizado para treinar o modelo foi composto por 128 amostras maliciosas e 1.089 artefatos benignos, que alcançou uma acurácia de 99.83%. Embora os autores destaquem que a solução proposta tenha uma baixa complexidade, nenhuma avaliação do *overhead* causado na rede é realizada.

O estudo de [Muthanna et al. 2022] propõe um mecanismo para detecção de intrusões em ambientes IoT via *deep learning*. A ferramenta proposta analisa o tráfego da rede e utiliza um classificador denominado *Cuda Long Short Term Memory Gated Recurrent Unit* (cuLSTMGRU), que consiste em uma variante do algoritmo LSTM, para identificar atividades maliciosas. O modelo é treinado e avaliado utilizando o *dataset* CICIDS2017, alcançando uma acurácia de 99.23%. Como trabalho futuro, os autores pretendem utilizar uma *blockchain* para melhorar a eficiência do sistema de detecção.

O trabalho de [Chang et al. 2022] apresenta uma ferramenta para detecção de *malwares* por meio de *switches* programáveis e *deep learning*. Para isso, uma *Convolutional Neural Network* (CNN) é empregada para classificação de tráfego. Os autores afirmam que o modelo, treinado com o *dataset* IoT-23, obteve uma acurácia de 99%, acompanhada por um tempo de processamento significativamente reduzido quando comparado com ferramentas tradicionais de detecção de intrusões. Entretanto, é importante observar que a solução inspeciona apenas os primeiros 40 *bytes* do *payload* de cada pacote, o que pode limitar sua eficácia, especialmente no caso de *malwares* que empregam técnicas de evasão, como a fragmentação do *payload* em múltiplos pacotes ou *junk code*. Ademais, as características utilizadas para a classificação não são especificadas no estudo.

[Chaganti et al. 2023] define uma abordagem para detecção de intrusões em ambientes IoT com base na análise e classificação de tráfego. Os autores fazem uso do classificador LSTM para detecção de atividades suspeitas. O modelo foi treinado utilizando os *datasets* SDNIoT e SDN-NF-TJ, alcançando uma acurácia de 97.1%. Entretanto, a degradação imposta pela ferramenta na performance da rede não foi considerada durante as avaliações realizadas. Como trabalho futuro, os autores pretendem validar se o modelo proposto é capaz de resistir a ataques adversariais.

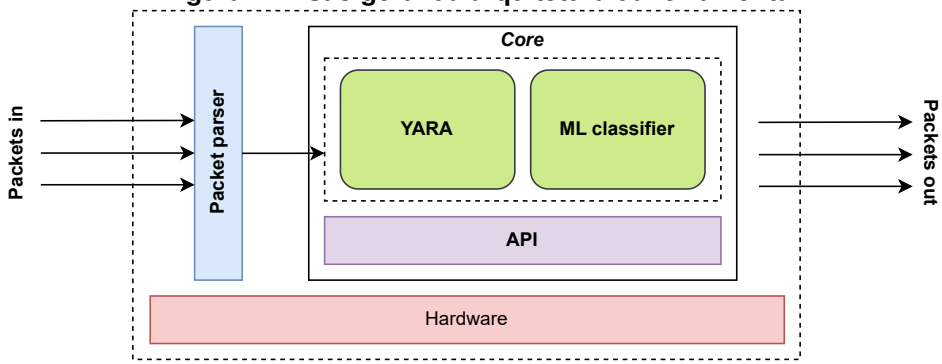
A proposta mais similar ao presente trabalho é a desenvolvida por [Chang et al. 2022]. Entretanto, a solução aqui proposta utiliza mais características do tráfego para identificação de *malwares*, melhorando a sua confiabilidade e diminuindo o número de falsos positivos. Outra distinção é a utilização de uma abordagem híbrida, composta por um módulo YARA responsável pela detecção rápida de padrões maliciosos e um classificador *Random Forest* para identificação de ameaças desconhecidas. Além disso, a ferramenta é capaz de isolar as máquinas infectadas do restante da rede, atuando como um mecanismo de contenção para evitar a propagação de ameaças. Por fim, a solução é avaliada considerando amostras de *malware* reais, que não estão presentes no conjunto de dados de treinamento ou de teste do modelo de *machine learning*.

3. Proposta

A arquitetura proposta tem como requisito a detecção agnóstica de artefatos maliciosos em ambientes IoT. Para se atingir esse objetivo, a solução é acoplada diretamente aos comutadores presentes na rede. Isso permite que a identificação de tráfego malicioso seja realizada independentemente das características específicas dos dispositivos conectados à infraestrutura. É importante notar que essa abordagem também reduz a sobrecarga no controlador da rede e evita pontos únicos de falha.

O procedimento adotado pela solução consiste na análise do tráfego da rede. Durante a análise, o *payload* do pacote é extraído para comparação com assinaturas de artefatos maliciosos específicos para dispositivos IoT e classificação via *machine learning*. Como exposto na Figura 1, os pacotes que chegam ao *switch* são pré-processados e enviados para detecção baseada em regras YARA. Essa abordagem possibilita a rápida detecção de artefatos que estão trafegando na rede. Em caso negativo, o módulo de classificação via *machine learning* é acionado, o que permite a identificação eficaz de ameaças desconhecidas ou que não possuem regras definidas.

Figura 1. Visão geral da arquitetura da ferramenta



3.1. Packet parser

Este componente é responsável por tratar os pacotes antes do processamento por parte do núcleo da ferramenta. O módulo atua similarmente a um *sniffer* de rede, o que possibilita a visualização completa das informações presentes em cada pacote. O principal objetivo é extrair as informações relevantes para a identificação por assinaturas e classificação via *machine learning*.

3.2. Core

O núcleo (*core*) da ferramenta é composto pelos principais componentes da arquitetura, responsáveis por efetivamente identificar atividades maliciosas na rede, além de prover uma API (*Application Programming Interface*) para gerenciamento remoto da ferramenta. A seguir, cada componente é descrito em detalhes.

3.2.1. YARA

As regras da abordagem proposta são definidas via YARA, uma solução amplamente utilizada por ferramentas de segurança. Isso possibilita a reutilização e adaptação de definições existentes, além de permitir que a ferramenta seja ajustada para detectar padrões de *malwares* com foco em diferentes sistemas operacionais.

A estrutura básica de uma regra para inserção na ferramenta é composta por um cabeçalho (1), uma meta descrição (2), uma seção de *strings* (3) e condições (4). O cabeçalho contém o nome da regra a ser definida e *tags* opcionais, que podem ser utilizadas para aplicação de filtros. A seção de meta descrição define informações úteis para o entendimento e documentação da regra. Já a seção de *strings* contém as sequências textuais e hexadecimais que serão buscadas nos *bytes* analisados. Por fim, a seção de condições estabelece quais requisitos a solução deve considerar para categorização do conteúdo analisado. É importante notar que, a depender da família de *malware* catalogada, módulos extras podem ser carregados para melhorar a precisão da regra, como os módulos PE (*Portable Executable*) e ELF (*Executable and Linkable Format*).

3.2.2. Machine learning classifier

O módulo de *machine learning* é responsável por analisar os *bytes* dos pacotes se nenhuma regra YARA puder identificá-los como maliciosos. Para isso, o algoritmo *Random Forest* foi treinado utilizando o *dataset* IoT-23 [Garcia et al. 2020] e a biblioteca *scikit-learn*[1]. A escolha deste algoritmo deve-se à sua utilização de uma série de árvores de decisão para melhorar a precisão do modelo e reduzir o *overfitting*, tornando-o um dos algoritmos mais comumente empregados na resolução de problemas de classificação e regressão. Dessa forma, é possível classificar várias famílias de *malware* independentemente da plataforma alvo.

O IoT-23 é um *dataset* rotulado que contém tráfego malicioso e benigno gerado por dispositivos IoT. Das 23 características presentes no conjunto de dados, as colunas `ts`, `uid` e `tunnel_parents` foram removidas, pois não representavam dados relevantes para o treinamento do modelo. A técnica de *feature encoding* foi aplicada às colunas `id.orig_h`, `id.resp_h`, `proto`, `service`, `conn_state` e `history`. Linhas com dados ausentes nas colunas `duration`, `orig_bytes` ou `resp_bytes` tiveram os valores ausentes definidos como a média da respectiva coluna. Por fim, a técnica de *feature scaling* foi empregada para melhorar o desempenho do modelo e evitar problemas de *overfitting*.

[1]https://scikit-learn.org

Para alcançar um modelo aceitável, 70% do *dataset* foi utilizado para treinamento e 30% para teste. Durante o treinamento, foram utilizados dois *jobs* paralelos, visando acelerar o processo, e o número de árvores no modelo foi configurado para ser 100. Esta configuração permite a fácil replicação dos resultados a partir do mesmo conjunto de dados. O modelo foi capaz de alcançar uma acurácia de 99.33%. Entretanto, é importante notar que essa acurácia se refere ao conjunto de dados de teste. Portanto, como destacado na Seção 4, uma validação cruzada é necessária para avaliar o modelo em cenários compostos por *malwares* reais.

3.2.3. API

A API da ferramenta fornece recursos para o gerenciamento remoto da solução. Para isso, é empregado o padrão *Representational State Transfer* (REST). Isso permite consultar o estado da solução e gerenciá-la remotamente a partir de qualquer ferramenta capaz de interagir com o HTTP. Vale notar que a API retorna informações no formato *JavaScript Object Notation* (JSON), um padrão de troca de dados entre sistemas que é independente da linguagem de programação. Os dados retornados pela API incluem informações sobre os dispositivos infectados, momento da detecção, família do *malware* e *hashes*. Isso permite que os operadores de rede manipulem facilmente os dados, possibilitando que as informações sejam utilizadas como entrada para outras soluções de segurança presentes na rede, como sistemas de detecção e prevenção de intrusões.

4. Prova de conceito

Esta seção aborda a validação da arquitetura proposta. A solução foi avaliada em uma plataforma de teste emulada por meio da ferramenta Mininet[2] e com o uso do *software* BMv2 P4[3]. A topologia utilizada para os testes consistiu em 2 dispositivos ARM emulados (H1 e H2) via QEMU[4], 1 *switch* programável emulado e habilitado com o a solução, e 1 controlador SDN. As avaliações foram conduzidas em uma máquina com processador Intel Core i7-11800H 2.30GHz (8 vCPUs), 32GB de RAM e sistema operacional Ubuntu Server 22.04.3 LTS 64-bit (Kernel Linux 6.2).

Para avaliar a capacidade de generalização do modelo e a eficácia da abordagem híbrida, foram utilizados 3030 artefatos maliciosos reais. Os artefatos foram obtidos da plataforma Malware Bazaar[5], uma solução amplamente utilizada na comunidade de segurança cibernética para análise automatizada e compartilhamento de *malware*, por meio de um *script* que consulta sua API pública. As *tags* "`iot`", "`arm`", "`mirai`", "`gafgyt`"e "`hajime`"foram utilizadas para buscar binários maliciosos que afetam dispositivos IoT. Os 3030 artefatos obtidos são divididos nas seguintes famílias: Mirai (2220), Gafgyt (754) e Hajime (56). Após o *download*, eles foram extraídos e enviados para o H1.

As regras YARA utilizadas na implementação foram obtidas do repositório de código aberto `Yara-Rules/rules`[6]. Este repositório contém regras desenvolvidas

[2]http://mininet.org
[3]https://github.com/p4lang/behavioral-model
[4]https://www.qemu.org
[5]https://bazaar.abuse.ch
[6]https://github.com/Yara-Rules/rules

Figura 2. Detecções por família

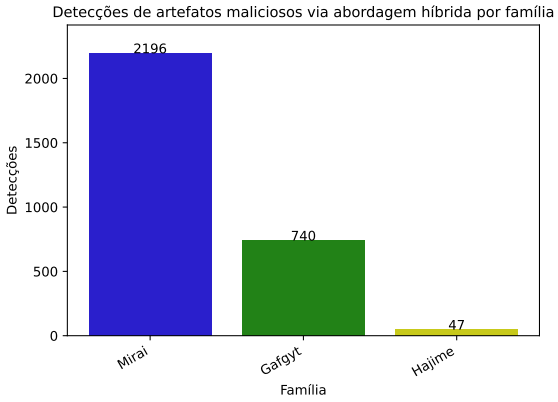


Figura 3. Tempo para classificação

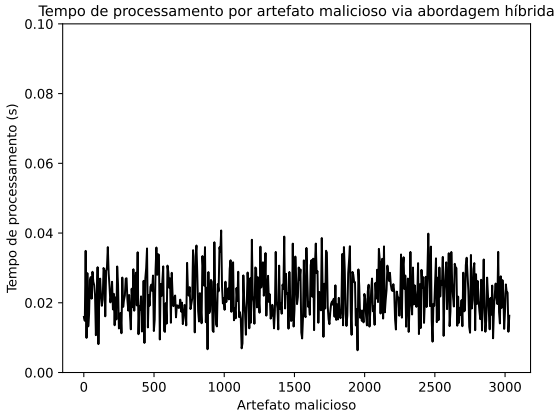


Tabela 1. Artefatos detectados

Família	Quantidade	Detectados	Acurácia
Mirai	2220	2196	98.91%
Gafgyt	754	740	98.14%
Hajime	56	47	83.92%
Total	**3030**	**2983**	**98.44%**

por profissionais de segurança da informação após a identificação e análise de ataques e artefatos maliciosos. Para maximizar o desempenho da ferramenta, foram selecionadas regras de categorias relevantes para ambientes IoT, incluindo: `Anti-debug/Anti-VM`, `CVE Rules`, `Exploit kits`, `Malware`, `Packers` e `Malware mobile`.

Como apresentado na Figura 2 e na Tabela 1, ao tentar enviar artefatos maliciosos de H1 para H2, a ferramenta alcançou uma acurácia de detecção de 98.44% ao adotar a abordagem híbrida, com a família Mirai tendo o maior número de detecções (2196), seguida pela família Gafgyt (740) e Hajime (47). Isso evidencia que a combinação de assinaturas e *machine learning* melhora a capacidade de identificação de programas maliciosos. O tempo médio para classificação de artefatos foi de 0.0217s, como exposto na Figura 3. O menor tempo registrado para classificação via abordagem híbrida foi de 0.0006s, e o maior foi de 0.0347s.

5. Conclusão

Este trabalho introduz uma solução para detecção de artefatos maliciosos em ambientes IoT habilitados por SDN. A arquitetura da ferramenta proposta se diferencia do estado da arte por fazer o uso de uma abordagem híbrida, capaz de realizar a rápida identificação de programas maliciosos via regras YARA e a classificação deles por meio do algoritmo de *machine learning Random Forest*. Com o auxílio de uma API, os operadores da rede podem facilmente alterar e gerenciar diferentes atributos da ferramenta.

Os resultados obtidos evidenciam que a abordagem híbrida proposta foi capaz de atingir uma acurácia de 98.44% e um tempo de processamento médio de 0.0217s. Como trabalhos futuros, pretende-se implementar um módulo que possibilite a análise

dos artefatos detectados, bem como avaliar a ferramenta em *switches* SDN reais, com o objetivo de determinar a eficiência da solução ao ser executada em um *hardware* dedicado.

Referências

Chaganti, R., Suliman, W., Ravi, V., and Dua, A. (2023). Deep learning approach for sdn-enabled intrusion detection system in iot networks. *Information*, 14(1):41.

Chang, H.-F., Wang, M. I.-C., Hung, C.-H., and Wen, C. H.-P. (2022). Enabling malware detection with machine learning on programmable switch. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pages 1–5. IEEE.

Garcia, S., Parmisano, A., and Erquiaga, M. J. (2020). IoT-23: A labeled dataset with malicious and benign IoT network traffic. More details here https://www.stratosphereips.org /datasets-iot23.

Gaurav, A., Gupta, B. B., and Panigrahi, P. K. (2023). A comprehensive survey on machine learning approaches for malware detection in iot-based enterprise information system. *Enterprise Information Systems*, 17(3):2023764.

Hauser, F., Häberle, M., Merling, D., Lindner, S., Gurevich, V., Zeiger, F., Frank, R., and Menth, M. (2023). A survey on data plane programming with p4: Fundamentals, advances, and applied research. *Journal of Network and Computer Applications*, 212:103561.

Khan, S. and Akhunzada, A. (2021). A hybrid dl-driven intelligent sdn-enabled malware detection framework for internet of medical things (iomt). *Computer Communications*, 170:209–216.

Kumar, S. and Chandavarkar, B. (2023). Analysis of mirai malware and its components. In *Machine Learning, Image Processing, Network Security and Data Sciences: Select Proceedings of 3rd International Conference on MIND 2021*, pages 851–861. Springer.

Liatifis, A., Sarigiannidis, P., Argyriou, V., and Lagkas, T. (2023). Advancing sdn from openflow to p4: A survey. *ACM Computing Surveys*, 55(9):1–37.

Maeda, S., Kanai, A., Tanimoto, S., Hatashima, T., and Ohkubo, K. (2019). A botnet detection method on sdn using deep learning. In *2019 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6. IEEE.

Muthanna, M. S. A., Alkanhel, R., Muthanna, A., Rafiq, A., and Abdullah, W. A. M. (2022). Towards sdn-enabled, intelligent intrusion detection system for internet of things (iot). *IEEE Access*, 10:22756–22768.

Peter, L. S., Kobo, H., and Srivastava, V. M. (2022). A comparative review analysis of openflow and p4 protocols based on software defined networks. *Data Intelligence and Cognitive Informatics: Proceedings of ICDICI 2022*, pages 699–711.