Mecanismo Dinâmico para a Detecção de Anomalias em Rede Baseado no Fator de *Outlier* Local e MUD

Franklin A. M. Venceslau, José A. Suruagy Monteiro

[1]Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – 50.740-600 – Recife – PE – Brazil

{famv,suruagy}@cin.ufpe.br

Abstract. *We witness daily the occurrence of cyberspace attacks that are increasingly sophisticated and difficult to identify. Distributed Denial of Service (DDoS) attacks are characterized by the ability to make a system unavailable by interrupting services to legitimate users. This article presents a dynamic mechanism for detecting anomalies in the network based on the manufacturer usage description (MUD) associated with the LOF (local outlier factor) machine learning algorithm. Based on the theory of early warning signals (EWS), the results indicate that the mechanism can predict and detect anomalous traffic on the network with accuracy and specificity above 90%.*

Resumo. *Presenciamos diariamente a ocorrência de ataques ao ciberespaço cada vez mais sofisticados e de difícil identificação. Os ataques de negação de serviço distribuído (DDoS) possuem como característica a capacidade de tornar indisponível um sistema por meio da interrupção de serviços a usuários legítimos. Este artigo apresenta um mecanismo dinâmico para detecção de anomalias na rede com base na descrição de uso do fabricante (MUD) associado ao algoritmo de aprendizagem de máquina LOF (fator de outlier local). Fundamentado na teoria dos sinais de alerta precoce (EWS), os resultados indicam que o mecanismo consegue predizer e detectar tráfego anômalo na rede com precisão e especificidade acima de 90%.*

1. Introdução

Com a evolução constante da Internet e dos meios de comunicação, cada vez nos deparamos com ataques mais sofisticados que visam explorar vulnerabilidades em sistemas cibernéticos. Ataques de negação de serviço distribuído (*Distributed Denial of Service – DDoS*) são uma ameaça crescente, caracterizados pelo esforço de sobrecarregar redes, servidores ou serviços com tráfego excessivo, comprometendo sua disponibilidade e integridade. Estes ataques evoluíram em complexidade e volume, desafiando as estratégias convencionais de defesa em cibersegurança [Griffioen and Doerr 2020].

Por tratar-se de uma ameaça distribuída e de difícil identificação, faz-se necessário que mecanismos de defesa atuem de modo eficiente e ao mesmo tempo ágeis o suficiente para tomar decisões em tempo hábil a fim de minimizar prejuízos ocasionados pela indisponibilidade dos serviços. Citamos como exemplo, um dos maiores ataques DDoS já registrados, ocorrido em 2016, que afetou a disponibilidade de serviços *online* essenciais à indústria 4.0 [Woolf 2016].

Ataques DDoS geram um alto tráfego de requisições simultâneas com o intuito de tornar indisponível os servidores ou nós da rede. Diversas soluções existentes para a predição e identificação possuem limitações quando a tratativa foca em analisar ataques DDoS desconhecidos, a exemplo dos ataques *zero-day*. Nestes ataques, os quais continuamente evoluem em sofisticação e magnitude, as estratégias tradicionais de mitigação, como listas de controle de acesso (*Access Control List – ACLs*) e sistemas de detecção de intrusão baseados em assinaturas (*Signature-based Intrusion Detection Systems – IDS*), enfrentam limitações significativas. No caso dos sistemas IDS, estes dependem de assinaturas conhecidas de ataques, tornando-os ineficazes contra novas variantes de ataques DDoS que não correspondam às assinaturas preexistentes.

Por exemplo, o ataque DDoS direcionado ao GitHub em 2018, que foi um dos maiores registrados na época, representou 1,35 terabits por segundo de tráfego utilizando uma metodologia que não requer *botnet* e que destacou a necessidade de métodos de detecção e mitigação mais adaptáveis e dinâmicos [Maciel 2018]. Já no caso das ACLs, apesar de serem mais vulneráveis a uma ação maliciosa mais ágil, ainda assim, a janela de tempo necessária para uma ação de reconfiguração da política de segurança pode ser consideravelmente alta, tornando o ambiente de rede mais vulnerável e suscetível a ampliação da superfície de ataque. Enquanto as ACLs dependem de regras estáticas, que podem ser inadequadas para responder rapidamente a ameaças emergentes, esta abordagem oferece uma capacidade de resposta mais dinâmica. Como resultado, a janela de vulnerabilidade é reduzida, limitando a ampliação da superfície de ataque e fortalecendo a segurança geral da rede.

Este artigo apresenta um mecanismo dinâmico que atua holisticamente na detecção de sinais que contrariam o comportamento normal da rede, antes que o atacante atinja estágios mais avançados em sua ação maliciosa. Tomando-se como base a teoria dos sinais de alerta precoce (*Early Warning Signals – EWS*), este mecanismo realiza a identificação sem rótulos a partir de oscilações de comportamento nos estados da rede, mensurados através de indicativos estatísticos. Estes indicadores são conhecidos na literatura científica e apoiam a tomada de decisão, no nosso caso, a partir da sinalização prévia fornecida pela descrição de uso do fabricante (*Manufacturer Usage Description – MUD)*. Deste modo, definimos três momentos de atuação do mecanismo na rede: (*i*) captura dos dados e preparação para análise, (*ii*) sinalização prévia da suspeita de ataque e (*iii*) emissão de alertas sugerindo a ocorrência de uma ação maliciosa na rede.

Para a validação do mecanismo proposto, duas bases de dados foram utilizadas como referência. Ambas as bases contam com tráfego benigno e tráfego oriundo de emulação de ataques DDoS. Os referidos dados analisados apresentam ocorrência de DDoS a partir de ataques nas camadas de rede e de transporte. O conjunto de dados conhecido como *IoT Network Intrusion Dataset* consiste em 42 arquivos de pacotes de rede brutos (*Packet Capture – PCAP*) capturados em diferentes momentos. Os pacotes de ataques são, em geral, provenientes de simulação usando ferramentas como o Nmap (Mapeador de Rede) [Kang et al. 2019]. A segunda base de dados é igualmente composta pelo tráfego benigno e maligno, além de caracterizar o tráfego e fazer uma extração de assinaturas para categorias de ataques igualmente definidos para as camadas de rede e de transporte [Zangrandi et al. 2022].

As demais seções deste artigo estão organizadas da seguinte forma: a Seção 2

descreve os trabalhos relacionados que associam mecanismos de detecção de intrusão a técnicas de aprendizagem de máquina; a Seção 3 trata sobre detecção de anomalias; na Seção 4 explicamos a proposta de mecanismo; a Seção 5 trata da avaliação preliminar de desempenho, e a Seção 6 conclui e apresenta os trabalhos futuros.

2. Trabalhos Relacionados

Observamos um crescente número de propostas de soluções para detecção e mitigação de ataques DDoS. Em [Jia et al. 2020] os autores propõem um mecanismo de defesa contra ataques DDoS em redes IoT (*Internet of Things)*. *FlowGuard* é apresentado como uma solução para mitigar esses ataques, operando na borda da rede. Este mecanismo utiliza uma abordagem de defesa em várias camadas: primeiro, ele emprega um sistema de detecção de anomalias baseado em aprendizado de máquina para identificar tráfego malicioso em tempo real; em seguida, ele usa um modelo de decisão para avaliar a ameaça e determinar a resposta adequada. Os autores propõem um novo algoritmo de detecção baseado em variações de tráfego a partir de dois modelos de aprendizado de máquina para identificação e classificação de DDoS. Os resultados indicam uma alta precisão de identificação e a precisão da classificação da rede neural convolucional de até 99,9%.

Em [Mirdula and Roopa 2023] temos uma solução baseada em algoritmos de aprendizado profundo para reforçar a segurança em redes IoT nos ambientes de edifícios inteligentes. A abordagem utiliza um IDS que integra os conceitos de MUD, gêmeos digitais e informações de comportamento do usuário obtidas por meio de aprendizado profundo. O modelo usa políticas MUD para proteger dispositivos contra ataques, como *BotNet*, DDoS, *Crypto-jacking* e *Ransomware*. Utiliza-se um modelo de aprendizado de máquina para classificar perfis MUD, que são então usados para treinar o modelo de aprendizado profundo. O modelo proposto é baseado em um *framework* MUD-DL (*Manufacturer Usage Description – Deep Learning*). Os resultados obtidos mostram que o *framework* proposto melhora a segurança cibernética e enfrenta desafios de aplicativos da Indústria 4.0, tendo o desempenho comparado com métodos de ponta em um novo *framework* de teste de capacidade de provisão de segurança.

Em [Morgese Zangrandi et al. 2022] os autores apresentam o MUDscope, um método que aproveita a especificação MUD para analisar assinaturas de ataques para múltiplos dispositivos. O tráfego de rede específico do dispositivo que não corresponde às regras do perfil MUD é considerado anômalo e registrado em arquivos de *log* de rede. Esses dados são então convertidos em formato de fluxo de pacotes para análise. A abordagem adotada para agrupar fluxos pertencentes a tráfego malicioso similar é baseada na análise de características de fluxo de rede, como *bytes* por pacote, *flags* de sinalização de estado TCP e portas de destino. Utiliza-se o algoritmo de clusterização HDBSCAN para agrupar fluxos anômalos semelhantes observados em uma janela de tempo, o que é adequado para a natureza dinâmica e desconhecida do tráfego de rede. A assinatura de um evento anômalo no MUDscope é definida pela evolução de *clusters* em *feeds* MRT (*MUD-Rejected Traffic*), representada por uma matriz com colunas indicando características da anomalia. Para detectar ameaças semelhantes em dispositivos IoT, o MUDscope compara essas assinaturas usando o coeficiente de correlação de *Pearson*, observando mudanças similares nos valores das características em diferentes janelas de tempo.

Em [Mazhar et al. 2021] observamos um sistema de detecção e prevenção de

intrusões em tempo real para redes IoT, denominado R-IDPS. O R-IDPS é desenvolvido como uma aplicação SDN (*Software Defined Network*) e opera em uma arquitetura cliente-servidor distribuída. O sistema cria um perfil base do tráfego de rede de IoT sob condições normais e utiliza esses dados para detectar anomalias. Para tal, emprega um modelo de aprendizado de máquina baseado em Máquina de Vetores de Suporte (SVM) para identificar ataques de inundação ICMP e SYN TCP. Para avaliação, foi configurado um ambiente de teste usando o *Mininet WiFi* sobre o sistema operacional *Ubuntu*. A avaliação do desempenho mostrou precisões de detecção de ataques entre 97% e 99%, sem falsos positivos.

Já em [Gonçalves et al. 2019] o artigo apresenta um Sistema de Prevenção de Intrusões (IPS) para redes IoT integradas com SDN. A proposta visa implementar funcionalidades de *firewalls* e IPS em uma arquitetura distribuída que suporta instâncias IoT, permitindo a identificação de comportamentos anômalos por parte de dispositivos IoT para bloquear ataques o mais próximo possível de suas fontes. A arquitetura é composta por dois componentes: um módulo de detecção baseado no IDS *Snort* e um módulo de proteção formado pelo *Controller SystemAPI* e *SnortAPI*, responsável por implementar as medidas de segurança na rede. Os resultados obtidos mostram que a arquitetura proposta é eficaz no enfrentamento do tráfego malicioso em redes IoT. A conclusão destaca que o IPS desenvolvido é capaz de configurar medidas de proteção na infraestrutura de rede com base na identificação de atacantes pelo IDS.

O estudo de [Pelloso et al. 2018] apresenta o STARK, um sistema autoadaptável para previsão de ataques DDoS fundamentado na teoria da metaestabilidade. O STARK fornece aprendizado estatístico não supervisionado e identifica a iminência de ataques DDoS sem a necessidade de conhecimento prévio ou supervisão. O mecanismo envolve três etapas principais: medições e preparação dos dados, predição dos ataques e emissão de alertas. Os resultados obtidos demonstram que o STARK é capaz de prever ataques DDoS com antecedência, variando de minutos a horas antes do início da sobrecarga gerada pelo ataque. Por exemplo, em um dos casos testados o sistema identificou um ataque DDoS com 23 minutos de antecedência. Os indicadores mostraram uma queda na taxa de retorno e um aumento nas curvas de autocorrelação, coeficiente de variação e assimetria, indicando uma forte instabilidade na rede e a aproximação de um ataque DDoS.

3. Detecção de Anomalias

Nesta seção apresentamos conceitos teóricos relevantes para compreensão da proposta do mecanismo de detecção de anomalias.

3.1. Descrição de uso do fabricante – MUD

MUD é um padrão técnico e um *framework* concebido para melhorar a segurança em redes IoT. Padronizado na RFC 8520 [Dunbar et al. 2019], o MUD fornece um meio para que os fabricantes de dispositivos IoT descrevam o comportamento de comunicação esperado entre o seu dispositivo e a rede. Essa descrição é expressa em um formato JSON (*JavaScript Object Notation*) e é conhecida como arquivo MUD. Este perfil especifica quais tipos de comunicação de rede são esperados e permitidos para um determinado dispositivo IoT. Isso inclui informações sobre quais portas e protocolos o dispositivo deve usar e com quais *endpoints* específicos ele pode se comunicar. O objetivo é restringir o

dispositivo a operar dentro de seu escopo pretendido, limitando o impacto de dispositivos comprometidos [Mirdula and Roopa 2023].

No contexto deste estudo, limitamos a comunicação do dispositivo a *endpoints* e serviços conhecidos e legítimos, fazendo com que o MUD reduza a superfície de ataque de dispositivos IoT, tornando-os menos suscetíveis a serem explorados para atividades maliciosas. Com a adoção deste mecanismo, administradores de rede implementam políticas de segurança mais facilmente, automatizando a configuração de ACLs e regras de *firewall* baseadas nas descrições fornecidas pelos fabricantes. Com as especificações de uso normal do dispositivo claramente definidas, qualquer desvio do comportamento esperado pode ser rapidamente identificado como potencialmente suspeito ou anômalo.

O MUDscope [Morgese Zangrandi et al. 2022] é uma ferramenta projetada para auxiliar na implementação e gestão de arquivos MUD. Seu objetivo principal é analisar e interpretar o tráfego de rede para gerar e refinar arquivos MUD e assim definir o comportamento esperado de dispositivos IoT. Inicialmente ele analisa o tráfego de rede para entender como os dispositivos IoT estão se comunicando e com quem. Essa análise ajuda a identificar padrões de comunicação legítimos, que podem ser usados para estabelecer regras de segurança de rede específicas para cada dispositivo.

Trazemos no trecho de código abaixo um recorte que representa o formato e a ”aparência”de um arquivo MUD. Analisando a representação em formato JSON, observamos que uma câmera de segurança inteligente deve se comunicar especialmente com um serviço na nuvem e aceitar conexões de um aplicativo móvel. Além disso, ele incorpora uma funcionalidade de detecção de anomalias, onde qualquer tráfego que não se encaixe nessas categorias pode ser sinalizado como potencialmente anômalo, levantando a um alerta de alta prioridade. Tal funcionalidade é útil para identificar atividades suspeitas, como tentativas de acesso não autorizado ao dispositivo.

```
{
  "ietf-mud:mud": {
    "mud-version": 1,
    "mud-url": "https://manufacturer.com/mud/v1/security-
       camera",
    "manufacturer": "SecureCam Inc.",
    "model": "SC-2000",
    "from-device-policy": {
      "access-lists": {
        "access-list": [{
          "name": "outbound-cloud-service"
        }]
      }
    },
    "to-device-policy": {
      "access-lists": {
        "access-list": [{
          "name": "inbound-mobile-app"
        }]
      }
```

```
    },
    "anomaly-detection": {
      "unexpected-traffic": {
        "alert-level": "high",
        "description": "Traffic not conforming to the
           defined policies"
      }
    }
  }
}
```

3.2. Teoria dos sinais de alerta precoce – EWS

Os alertas precoces representam um campo de pesquisa que foca na identificação de indicadores preditivos de transições críticas em sistemas complexos. O fundamento desta teoria reside na premissa de que muitos sistemas naturais e artificiais exibem mudanças sutis em seu comportamento antes de uma transição significativa ou um evento crítico [Yan and Zhang 2013]. EWS são identificados como mudanças nas propriedades estatísticas de um sistema que sinalizam a aproximação de um ponto crítico ou de uma transição de fase. Estes sinais são frequentemente modelados matematicamente e identificados através de técnicas de análise estatística.

À medida que um sistema se aproxima de um ponto crítico, frequentemente observa-se um aumento na autocorrelação temporal. Isto é, as observações sucessivas tornam-se mais previsivelmente semelhantes [Sapienza et al. 2017]. No contexto deste estudo, utilizamos o seu embasamento estatístico, como por exemplo, a avaliação da autocorrelação temporal dos dados e a análise de variância para analisar os dados sinalizados pelo MUD como potencialmente anômalos.

Realizamos uma análise preliminar, conforme observado pela Figura 1, a qual apresenta dois gráficos (a) e (b): o primeiro ilustra a autocorrelação temporal do tráfego de rede durante um ataque DDoS. No eixo X, temos a representação do *lag*, que consiste em um termo estatístico usado em análises de séries temporais para referência ao intervalo de tempo ou ao deslocamento entre observações na série. Por exemplo, um *lag* de 1 pode significar um deslocamento de uma unidade de tempo (um segundo, uma hora, um dia, dependendo da granularidade dos dados) [Tang et al. 2018]. Em um gráfico de autocorrelação, cada ponto no eixo X representa o grau de correlação do tráfego de rede com ele mesmo, deslocado por esse intervalo de *lag*. Portanto, na escala de tempo que consideramos em nosso estudo, 10s (segundos), por exemplo, compara o tráfego no momento atual com o tráfego de 10 unidades de tempo atrás. No eixo Y, a autocorrelação, também conhecida como correlação serial, é uma medida da relação linear entre os valores de uma série temporal e os valores da mesma série deslocados pelo *lag*. No contexto deste estudo, utilizamos para identificar padrões repetitivos ou dependências temporais em dados de séries temporais, como o tráfego de rede analisado.

No gráfico a) representando pela Figura 1, a partir do instante 500 no eixo X, observamos um comportamento de baixa autocorrelação, indicando tráfego normal. Com o início do ataque DDoS, observa-se um pico significativo na autocorrelação, refletindo um aumento na similaridade dos dados de tráfego ao longo do tempo, o que é típico durante

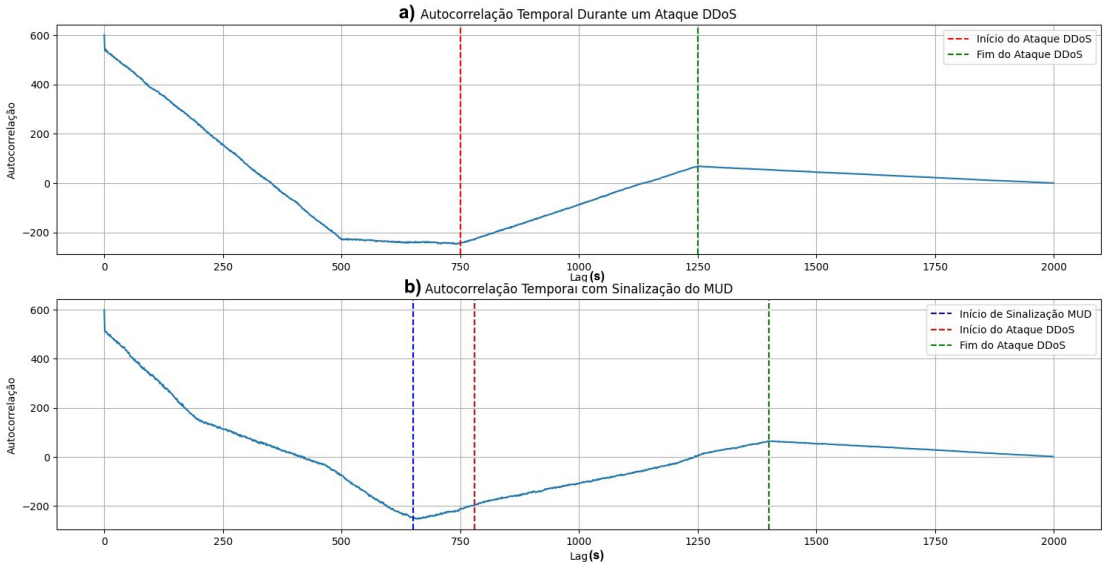

Figura 1. Autocorrelação Temporal do Tráfego de Rede

um ataque DDoS devido ao volume massivo e consistente de tráfego. No gráfico b), temos a autocorrelação temporal com a sinalização do MUD. Antes do ataque DDoS, há um período em que o MUD começa a sinalizar pacotes suspeitos. Durante este período, pode-se observar uma leve alteração na autocorrelação, não tão pronunciada quanto durante o ataque DDoS, mas significativa o suficiente para indicar uma mudança no padrão do tráfego de rede. O pico de autocorrelação durante o ataque DDoS é semelhante ao observado no primeiro gráfico. A inclusão da sinalização do MUD adiciona uma nova dimensão ao gráfico, permitindo a visualização de um estágio preliminar de atividade suspeita antes do pico do ataque DDoS.

Como pode-se observar, a principal diferença entre os gráficos é a inclusão da fase de sinalização do MUD no gráfico b). Isso permite identificar uma fase de alerta antes do ataque DDoS. Em a) temos uma representação mais direta do impacto de um ataque DDoS na autocorrelação do tráfego de rede, enquanto que em b) fornecemos uma melhor compreensão sobre as fases de detecção e alerta que antecedem um ataque DDoS, pois destaca a importância de mecanismos de alerta precoce, como o MUD, na detecção de potenciais anomalias.

3.3. Fator de *outlier* local – LOF

O LOF [Breunig et al. 2000] mede a densidade local de um ponto em relação aos seus vizinhos. Isso é útil para detectar anomalias em ambientes onde a densidade de pontos, ou seja, o tráfego de rede, varia significativamente. No domínio que investigamos, o LOF é eficaz na identificação de *outliers* que podem ser perdidos em métodos globais. Ele pode detectar anomalias que são incomuns em seu contexto local, mesmo que sejam semelhantes a outros pontos em uma escala global. Um dos benefícios que esperamos obter é a sinalização do LOF a pequenas mudanças na proximidade de pontos de dados, o que pode ser relevante para detectar ataques sutis ou emergentes em redes IoT.

Observamos na Figura 2, pela representação da simulação realizada, que o arquivo MUD após estabelecer as regras de comunicação esperadas para cada dispositivo,

complementa a segurança identificando desvios dessas regras, mesmo que sutis através do LOF. Os pontos em azul representam o tráfego normal, enquanto que os pontos vermelhos representam o tráfego potencialmente anômalo. Ainda de acordo com a figura, visualizamos a representação de duas características consideradas na análise em questão, sendo elas: o comprimento dos pacotes, no eixo X e a volumetria dos pacotes, no eixo Y.

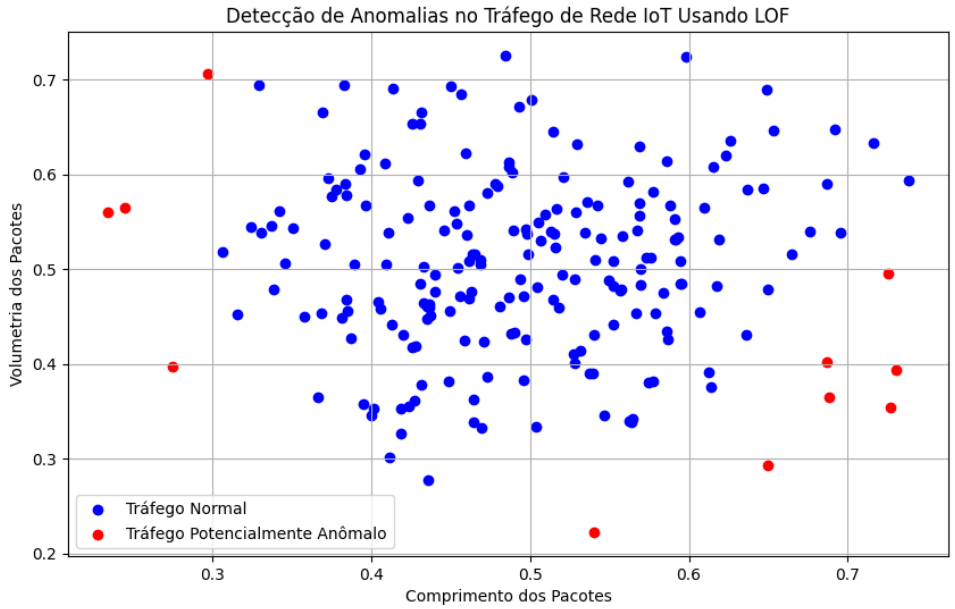

Figura 2. Representação dos *outliers* com uso do LOF

4. Proposta de Mecanismo Dinâmico

Nesta seção detalhamos o mecanismo proposto, que tem como objetivo a detecção de comportamentos anômalos na rede a partir do MUD, associado a técnicas de aprendizagem de máquina com o foco em sinalizar ao *firewall* de rede sobre a necessidade de realizar uma ação dinâmica para reconfiguração do seu perfil de segurança, com base no comportamento do dispositivo.

4.1. Posicionamento na rede

A Figura 3 ilustra a sugestão de posicionamento do mecanismo, além de exibir as etapas do seu funcionamento. Consideramos que a decisão sobre o melhor posicionamento de mecanismos de segurança em uma rede, depende dos requisitos do ambiente que se deseja monitorar, além das características intrínsecas ao contexto da aplicação. Este estudo sugere a atuação do mecanismo entre o roteador de borda, que é o ponto na rede onde o tráfego de entrada e saída é monitorado, sendo este o local onde os dados de tráfego são inicialmente capturados e também onde as regras derivadas da análise são aplicadas para bloquear ou limitar o tráfego anômalo; e o sistema de *firewall* com implementação em um servidor de aplicação em nuvem, porém sem desconsiderarmos a possibilidade de utilização em uma solução dedicada em *hardware* ou podendo ainda ser implementado junto ao *firewall* por meio de API específica. Ainda na Figura 3, visualizamos a DMZ (zona desmilitarizada) que consiste na representação de uma área da rede projetada para adicionar uma camada adicional de segurança, separando a rede interna da Internet.

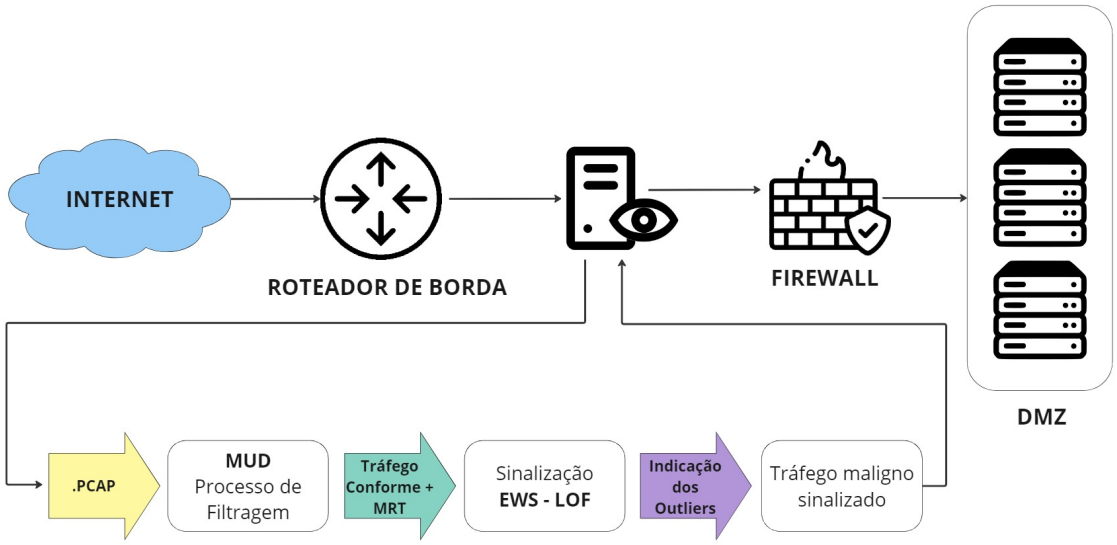

Figura 3. Posicionamento do mecanismo na rede

4.2. Planejamento do mecanismo

Nosso sistema atua a partir dos seguintes papéis: (*i*) captura e preparação de dados, (*ii*) sinalização MUD-EWS, e (*iii*) cálculo dos indicadores com base no LOF.

O processo inicia com a captura do tráfego de rede em arquivos PCAP, que registram o tráfego de dados que passa através de um ponto de rede, em nosso contexto, o roteador de borda. A preparação dos dados envolve a filtragem e preparação dos pacotes contendo características (*features*) específicas do tráfego, no nosso caso, consideramos comprimento e volumetria de pacotes. O arquivo MUD, que descreve o comportamento de rede esperado para dispositivos específicos, é usado para identificar o tráfego que não corresponde ao padrão esperado. Esse passo considera os sinais de alerta precoce, que usa as informações do MUD para sinalizar tráfego potencialmente anômalo ou suspeito. O EWS pode gerar alertas ou marcar o tráfego para uma análise mais detalhada.

Através da análise do tráfego, o sistema identifica um perfil de comportamento normal com base no MUD e um perfil de alerta precoce com base no EWS [Yan and Zhang 2013]. Estes sinais referem-se a um conjunto de indicadores ou padrões que são utilizados para detectar sinais de uma mudança iminente em um sistema. O objetivo do EWS, neste contexto, é permitir uma intervenção precoce para evitar ou mitigar resultados negativos através da análise de padrões, tendências e outros indicadores que historicamente precederam eventos significativos.

Nosso mecanismo se concentra na criação de um fluxo de trabalho abrangente e dinâmico para a detecção de anomalias na rede. Inicialmente, o tráfego de entrada passa por uma fase de filtragem, onde o MUD, estabelece parâmetros claros para o comportamento esperado dos dispositivos conectados à rede. Durante esta fase, o tráfego que não corresponde às definições do MUD é categorizado como MRT (*Malicious Rejected Traffic*), evidenciando atividades potencialmente maliciosas. Em seguida, o tráfego filtrado, incluindo o MRT, é submetido ao processo de detecção de anomalias, onde a combinação do EWS e do algoritmo LOF entra em atuação. Este sistema analisa o tráfego, identi-

ficando desvios sutis dos padrões normais. Os dados identificados como *outliers* pelo LOF, apoiados pelo alerta precoce do EWS, são então marcados para uma análise mais detalhada, possibilitando uma resposta e ação mais rápidas.

O *sniffer* grava os dados de captura em um arquivo no formato PCAP. Este arquivo contém informações detalhadas sobre cada pacote capturado, incluindo cabeçalhos de protocolo, *timestamps* e *payload*.

Posteriormente implementamos a biblioteca MUDscope, esta ferramenta permite a utilização dos perfis MUD para cada dispositivo monitorado. Filtramos o tráfego de rede com base no perfil MUD, e chamamos o tráfego rejeitado por MUD (MRT), onde, diferentemente do proposto em [Zangrandi et al. 2022] aqui o tráfego será sinalizado como potencialmente anômalo para sequencialmente passar pelas tratativas dos mecanismos de EWS e aprendizagem de máquina. A partir dos pacotes capturados são identificados os fluxos correspondentes, ou seja, há uma conversão de dados de pacotes para dados de fluxos de pacotes, sendo a análise individualizada por dispositivo e por janela de tempo. Outra distinção desta proposta em relação a [Zangrandi et al. 2022] é que utilizamos o algoritmo de detecção de *outliers* LOF ao invés do mecanismos proposto pelos autores, o HDBSCAN, para analisar e obter grupos de tráfego semelhantes.

4.3. Modelagem do mecanismo proposto

Vejamos a seguir o modelo de base considerando o tráfego esperado:

A partir do conjunto de dados do tráfego de rede X com n observações, que representam o comportamento normal do tráfego de rede esperado para um dispositivo, conforme descrito pelo MUD, o tráfego normal pode ser modelado como uma distribuição de probabilidade $P(X)$, como uma distribuição normal $\mathcal{N}(\mu, \sigma^2)$, onde μ é a média e σ^2 é a variância do tráfego de rede esperado. Para a detecção de desvios, definimos uma variável aleatória Y que representa o tráfego de rede observado. A probabilidade de Y desviar significativamente de X pode ser calculada usando a função densidade de probabilidade de $P(X)$. Seja α um limiar definido para a detecção de anomalias, por exemplo, o 95º percentil da distribuição $P(X)$. O EWS atua como um classificador binário que sinaliza uma anomalia se $P(Y > \alpha)$ for verdadeiro. A eficácia do EWS pode ser medida pela sua sensibilidade, representada pelo número de verdadeiros positivos e especificidade, representada pelo número de verdadeiros negativos. Sensibilidade é a probabilidade do EWS sinalizar corretamente uma anomalia quando uma ocorre:

$$\text{Sensibilidade} = P(\text{Sinalização}|Y > \alpha)$$

Já a especificidade é a probabilidade do EWS não sinalizar uma anomalia quando o tráfego é normal:

$$\text{Especificidade} = P(\text{Não Sinalização}|Y \leq \alpha)$$

No papel (*iii*) o algoritmo LOF é aplicado para detectar anomalias no tráfego de rede. Através do cálculo da densidade local de cada ponto, ou seja, as amostras do tráfego, comparamos com a densidade dos seus vizinhos para identificar pontos que são considerados *outliers*. Seja $D = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ o conjunto de amostras de tráfego de rede, onde cada $\mathbf{x}_i$ é um vetor de características. Cada amostra $\mathbf{x}_i$ possui um rótulo

verdadeiro y_i, onde $y_i = 1$ se a amostra for normal e $y_i = -1$ se for anômala. O LOF atribui a cada amostra um *score* de *outlier* s_i, com base na densidade local. Temos então um limiar t tal que se $s_i > t$, $\mathbf{x}_i$ é classificado como anômalo ($\hat{y}_i = -1$), caso contrário, é classificado como normal ($\hat{y}_i = 1$). A partir disso definimos a matriz de confusão C onde:

- C_{11} é o número de Verdadeiros Positivos (TP): $\sum_{i=1}^{n} \mathbb{I}(y_i = 1 \wedge \hat{y}_i = 1)$
- C_{10} é o número de Falsos Positivos (FP): $\sum_{i=1}^{n} \mathbb{I}(y_i = -1 \wedge \hat{y}_i = 1)$
- C_{01} é o número de Falsos Negativos (FN): $\sum_{i=1}^{n} \mathbb{I}(y_i = 1 \wedge \hat{y}_i = -1)$
- C_{00} é o número de Verdadeiros Negativos (TN): $\sum_{i=1}^{n} \mathbb{I}(y_i = -1 \wedge \hat{y}_i = -1)$

Aqui, $\mathbb{I}$ é a função indicadora que retorna 1 se a condição for verdadeira e 0, caso contrário.

5. Avaliação Preliminar de Desempenho

Nesta seção é apresentada uma avaliação preliminar de desempenho do mecanismo proposto na Seção 4, a fim de avaliar a sua eficácia na detecção de tráfego malicioso e o seu custo computacional.

5.1. Métricas

Utilizamos cinco métricas para analisar a eficiência do mecanismo, além do custo computacional correspondente.

- Precisão: Prec $= \frac{C_{11}}{C_{11}+C_{10}}$
- *Recall* ou Sensibilidade: Sens $= \frac{C_{11}}{C_{11}+C_{01}}$
- Especificidade: Esp $= \frac{C_{00}}{C_{00}+C_{10}}$
- F1-score: $F1 = 2 \times \frac{\text{Prec} \times \text{Sens}}{\text{Prec} + \text{Sens}}$
- Taxa de Falsos Positivos: FPR $= \frac{C_{10}}{C_{10}+C_{00}}$

A precisão avalia a proporção de identificações corretas de anomalias em relação a todas as identificações de anomalias corretas e incorretas; *recall* ou sensibilidade mede a proporção de anomalias reais que foram corretamente identificadas pelo sistema; a especificidade avalia a capacidade do sistema em identificar o tráfego normal; F1-score combina precisão e *recall* em uma única métrica e a taxa de falsos positivos que mede a frequência com que o tráfego normal é incorretamente classificado como anômalo.

5.2. Fatores

No estudo realizado para um mesmo conjunto de dados foram obtidas as métricas de eficácia e de custo computacional para o mecanismo com a atuação isolada do LOF e a atuação do mesmo com o apoio da sinalização prévia do MUD e EWS.

5.3. Observações durante a execução dos testes

Os pacotes analisados variaram em tamanho, com uma média de 800 bytes, mas oscilando significativamente entre 60 e 1230 bytes, especialmente durante a fase de sinalização de possível ataque através do MUD. Durante a janela de tempo considerada, foram analisados os fluxos com períodos de alta densidade de tráfego intercalados com períodos mais calmos. Os testes foram conduzidos em uma janela de aproximadamente 8 horas, proporcionando uma visão abrangente do comportamento da rede ao longo do tempo. Observamos oscilações no comportamento do tráfego, tanto na fase de treinamento quanto na de

teste, com variações no volume e no tamanho dos pacotes, refletindo a dinâmica real da rede. A autocorrelação temporal aumentou significativamente, indicando uma tendência do tamanho dos pacotes de dados em permanecer com valores em torno de 1100 bytes, especialmente durante os períodos de tráfego anômalo. Essa elevada autocorrelação é um indicativo de padrões de tráfego consistentes, que podem ser um indicativo de atividade maliciosa.

5.4. Descrição do ambiente de teste

Com a finalidade de testar o mecanismo proposto neste estudo, de acordo com o que ilustramos na Figura 3, desenvolvemos um ambiente para execução dos testes e simulações propostos a partir de máquinas virtuais Linux, utilizando a distribuição Ubuntu sem interface gráfica. A nível de *hardware* utilizamos duas máquinas com 2 GB de memória RAM, ambas com um núcleo de processador e uma máquina com 8 GB de memória RAM com dois núcleos de processamento. Neste ambiente de teste, utilizamos a topologia proposta na Figura 3, onde a máquina com 8 GB de memória RAM representa a interface NAT (*Network Address Translation*) do sistema virtualizado com a rede real. Utilizamos o *tcpdump* para captura do tráfego de rede e a etapa de análise de dados foi processada igualmente na máquina virtual mais potente com a utilização das bibliotecas *scikit-learn* para implementação do LOF, incluindo as etapas de definição do número de vizinhos, o *Pandas* para manipulação dos dados e o *Matplotlib* para a visualização dos dados de forma gráfica.

5.5. Resultados obtidos para a eficácia do mecanismo

A Tabela 1 exibe uma comparação entre a atuação isolada do LOF (sem MUD-EWS) e a atuação do mesmo com o apoio da sinalização prévia do MUD e EWS. Observamos um aumento em todas as métricas de desempenho comparadas, fato este que evidencia a eficiência do sistema proposto quando utilizamos os mecanismos combinados.

Tabela 1. Avaliação preliminar da eficácia do mecanismo

Métrica	Sem MUD-EWS	Com MUD-EWS
Precisão (Prec)	0.82	0.91
Recall / Sensibilidade (Sens)	0.79	0.90
Especificidade (Esp)	0.85	0.92
F1-score	0.81	0.89
Taxa de Falsos Positivos (FPR)	0.13	0.10

5.6. Resultados obtidos para o custo computacional

A Figura 4 ilustra uma comparação do consumo de recursos computacionais, considerando processamento de CPU e memória RAM a partir de simulações com uso do LOF isolado, e das estratégias combinadas de MUD-EWS e LOF. Os testes foram executados na mesma janela de tempo considerada na Seção 5.3 e mostram uma diferença de consumo bastante sutil a partir da implementação do mecanismo combinado, o que reforça o potencial da proposta deste estudo quando comparamos o aumento dos valores obtidos a partir das métricas de desempenho analisadas em relação ao pequeno aumento do custo

computacional. Enfatizamos ainda, que o ambiente de testes foi composto por máquinas com configuração bastante modesta (2 GB de RAM e um núcleo de processamento), o que enfatiza ainda mais a possibilidade de utilização do mecanismo proposto em cenários distintos com severas limitações de recursos.

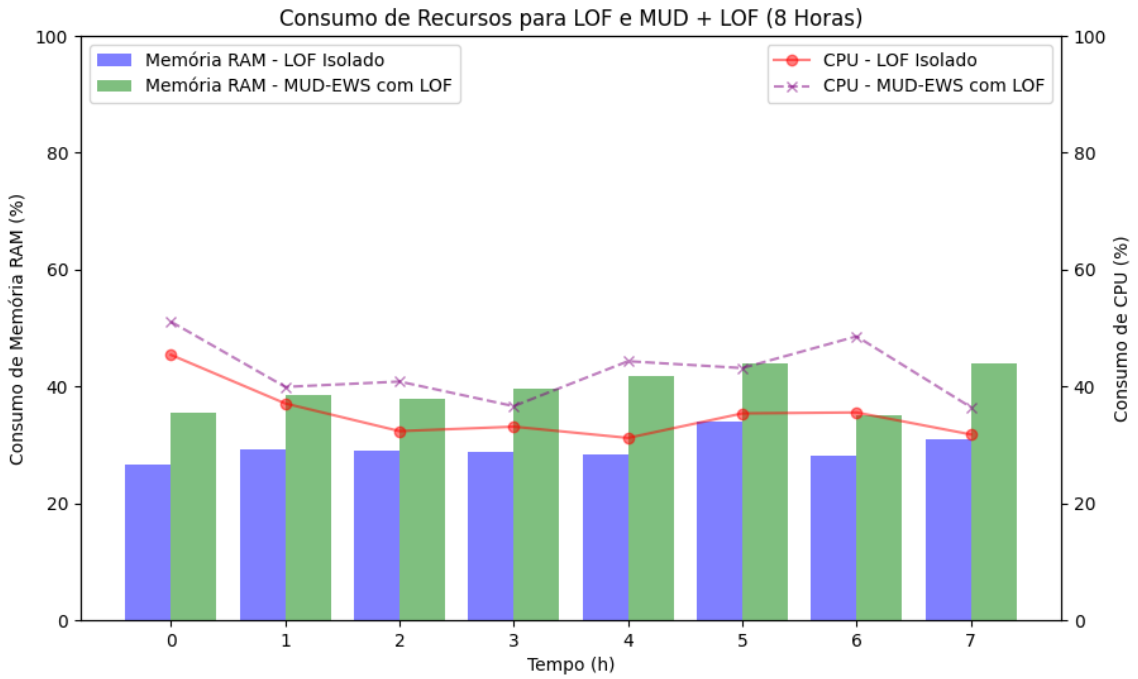

Figura 4. Comparação do consumo de recursos

6. Conclusão

Este artigo apresenta um mecanismo dinâmico que realiza a detecção de anomalias na rede a partir da sinalização do comportamento esperado do dispositivo. Após detectada uma ação potencialmente maliciosa, alertas são enviados ao mecanismo de aprendizagem de máquina que busca sinalizar corretamente a identificação do ataque como estratégia para mitigação de ações maliciosas no ambiente de rede. Atingimos taxas de desempenho satisfatórias a partir da comparação de cinco métricas distintas analisadas e comparamos o custo de processamento computacional a partir de estratégias isoladas e combinadas para detecção de tráfego anômalo, mostrando, em todos os casos, uma alteração discreta no consumo de recursos.

Em trabalhos futuros, pretendemos implementar a solução em posicionamentos distintos na rede, testando e medindo a sua eficiência em ativos de segurança, tais como em *firewalls*, cujas APIs permitam este tipo de implementação. Pretendemos também realizar novos testes aumentando a quantidade de *features* analisadas e testar em cenários com inserção de *delays* entre a transmissão de dados a fim de atingir níveis ainda mais altos de confiabilidade para outros ambientes aderentes à proposta.

Agradecimentos

Este trabalho foi financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq (Proc. 162441/2021-5), e pela Fundação de Amparo à Pesquisa do Estado de São Paulo – FAPESP (Proc. 2018/23098-0).

Referências

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: Identifying density-based local outliers. In *ACM SIGMOD Record*.

Dunbar, L., Lear, E., Droms, R., and Romascanu, D. (2019). Manufacturer usage description specification. RFC 8520, RFC Editor.

Gonçalves, D., Kfouri, G., Dutra, B., Alencastro, J., Filho, F., Martins, L., Albuquerque, R., and de Sousa Junior, R. (2019). Arquitetura de IPS para redes IoT sobrepostas em SDN. In *Anais do XIX SBSeg*, pages 309–322.

Griffioen, H. and Doerr, C. (2020). Examining mirai's battle over the internet of things. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 743–756.

Jia, Y., Zhong, F., Alrawais, A., Gong, B., and Cheng, X. (2020). FlowGuard: An intelligent edge defense mechanism against IoT DDoS attacks. *IEEE Internet of Things Journal*, 7(10):9552–9562.

Kang, H., Ahn, D. H., Lee, G. M., Yoo, J. D., Park, K. H., and Kim, H. K. (2019). IoT network intrusion dataset. `https://dx.doi.org/10.21227/q70p-q449`.

Maciel, R. d. S. (2018). Avaliação do impacto de ataques ddos e malware: uma abordagem baseada em árvore de ataque. Master's thesis, Universidade Federal de Pernambuco.

Mazhar, N., Salleh, R., Zeeshan, M., Hameed, M. M., and Khan, N. (2021). R-IDPS: Real time SDN based IDPS system for IoT security. In *IEEE HONET 2021*, pages 71–76.

Mirdula and Roopa (2023). MUD enabled deep learning framework for anomaly detection in IoT integrated smart building. *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, 5:100186.

Morgese Zangrandi, L., van Ede, T., Booij, T., Sciancalepore, S., Allodi, L., and Continella, A. (2022). Stepping out of the MUD: Contextual thr information for IoT devices with manufacturer-provided behavior profiles. In *Proceedings of ACSAC '22*. ACM.

Pelloso, M., Vergütz, A., Santos, A., and Nogueira, M. (2018). Um sistema autoadaptável para predição de ataques DDoS fundado na teoria da metaestabilidade. In *Anais do XXXVI SBRC*, pages 726–739, Porto Alegre, RS, Brasil. SBC.

Sapienza, A., Bessi, A., Damodaran, S., Shakarian, P., Lerman, K., and Ferrara, E. (2017). Early warnings of cyber threats in online discussions. In *2017 Annual Computer Security Applications Conference (ICDMW)*, pages 667–674.

Tang, M., Alazab, M., Luo, Y., and Donlon, M. (2018). Disclosure of cyber security vulnerabilities: time series modelling. *International Journal of Electronic Security and Digital Forensics*, 10(3):255–275.

Woolf, N. (2016). DDoS attack that disrupted internet was largest of its kind in history, experts say. *The Guardian*, 26.

Yan, X. and Zhang, J. Y. (2013). Early detection of cyber security threats using structured behavior modeling. *ACM Transactions on Information and System Security*, 5(10).

Zangrandi, L., van Ede, T., Booij, T., Sciancalepore, S., Allodi, L., and Continella, A. (2022). MUDscope dataset. `https://doi.org/10.5281/zenodo.7182597`.