Estendendo o MENTORED *Testbed* para a execução de experimentos de cibersegurança multi-cluster e IoT

Khalil G. Q. de Santana[1], Bruno Henrique Meyer[2], Davi Daniel Gemmer[3], Marcos Schwarz[3], Michelle S. Wangham[1,3]

[1] Universidade do Vale do Itajaí - UNIVALI

[2]Universidade Federal do Paraná - UFPR

[3]Rede Nacional de Ensino e Pesquisa - RNP

Abstract. *IoT devices are widely used in many areas but often present critical vulnerabilities, such as weak passwords, exposed services, and outdated software. If compromised, they can form botnets and threaten systems on the Internet. To mitigate these weaknesses in an ethical and secure manner, a dedicated experimentation environment, a testbed, is needed to study attacks and develop security solutions. This work extends the MENTORED Testbed with the integration of IoT devices, creating a multi-cluster testbed for experiments on different architectures. Two attack scenarios were simulated to evaluate the developed extension, demonstrating its scalability and applicability in IoT security.*

Resumo. *Dispositivos IoT são amplamente utilizados em diversas áreas, mas frequentemente apresentam vulnerabilidades críticas, como senhas simples, serviços expostos e software desatualizado. Comprometidos, podem formar botnets e ameaçar sistemas na Internet. Para mitigar essas fraquezas de forma ética e segura, é necessário um ambiente para experimentação dedicado, um testbed, para o estudo de ataques e desenvolvimento de soluções de segurança. Este trabalho amplia o MENTORED Testbed com integração de dispositivos IoT, criando um testbed multi-cluster para experimentos em diferentes arquiteturas. Dois cenários de ataque foram simulados para avaliar a extensão desenvolvida, demonstrando sua escalabilidade e aplicabilidade na segurança IoT.*

1. Introdução

Dispositivos Internet das coisas (IoTs) são usados globalmente, contudo tais dispositivos ainda trazem consigo desafios de segurança e privacidade [Aqeel et al. 2022]. Em relação à segurança, o uso de esquemas de autenticação fracos ocupa o primeiro lugar dos riscos mais críticos [OWASP 2018]. Tais brechas na segurança possibilitam que *botnets* sejam formadas e lancem ataques de negação de serviço distribuídos (DDoS) contra serviços e aplicações na Internet que, de acordo com [Sritharan et al. 2022], são complexos de serem mitigados. Outra métrica relevante levantada são que ataques DDoS aumentaram em 31%, sendo lançados o equivalente à 44 mil ataques diários deste tipo [SentinelOne 2024]. *Botnets* têm utilizado dispositivos IoT e outros serviços comprometidos para ataques variados, como *spam* e campanhas de *phishing*. Segundo [Imperva 2023], 30% do tráfego da Web é gerado por *bots* maliciosos, evidenciando seu impacto negativo na Internet.

Para investigar soluções e estudar o comportamento de ataques de DDoS, *botnets* e protocolos de rede, pesquisadores utilizam ambientes controlados baseados em

simulações, porém, segundo [Siaterlis et al. 2012], o uso de simuladores pode não ser adequado para recriar comportamentos de redes complexas reais. O uso de um ambiente de experimentação de rede realístico, denominado *testbed*, pode ser mais adequado, possibilitando a coleta de dados (*datasets*) dos experimentos para serem estudados de maneira segura e isolada de sistemas reais críticos.

Na literatura, *testbeds* voltados a segurança em dispositivos IoTs como propostos em [Siboni et al. 2018] e [Al-Hawawreh and Sitnikova 2020], que apesar de oferecerem um ambiente realístico, sofrem de problemas de escalabilidade devido a sua arquitetura [Santana et al. 2024]. Em contra partida, um exemplo de arquitetura escalável é descrito no *testbed* EdgeNet [Şenel et al. 2021], o qual utiliza de camadas de abstração como *containers* e ferramentas de orquestração como Kubernetes para prover um *testbed* escalável e de fácil manutenção, contudo tal ambiente não conta com dispositivos IoT.

MENTORED *Testbed* [Meyer et al. 2022] é um ambiente para experimentação de segurança construído sobre a infraestrutura definida por software da RNP, chamada de Cluster Nacional. O projeto utiliza do Kubernetes para prover esta infraestrutura e engloba cenários realísticos, incluindo testes de alta-vazão como ataques DDoSs, um portal Web que permite a depuração dos componentes envolvidos. Contudo, este *testbed* está centrado em ambientes *AMD64* e não conta com uma ilha de dispositivos IoTs integrada ou suporte a redes sem fio até então. Este artigo descreve a evolução do MENTORED *Testbed*, incorporando tais aspectos e ainda assim, se mantendo escalável e seguro, para oferecer uma boa experiência a seus usuários, com facilidades para execução de experimentos no contexto da IoT e para geração de conjuntos de dados.

Diante disso, este trabalho busca responder à pergunta de pequisa: ”Como evoluir o MENTORED *Testbed* para suportar experimentos com dispositivos IoT, oferecendo boa experiência de uso e garantindo realismo e larga escala?”. Para isso, foram formuladas sub-perguntas: (*i*) como refletir as limitações, heterogeneidade e escala dos dispositivos IoT no *testbed*? (*ii*) como preservar essas características sem comprometer manutenção, segurança e desempenho? (*iii*) como integrar duas ilhas com arquiteturas heterogêneas?

Para avaliar as extensões desenvolvidas no MENTORED *Testbed*, este artigo detalha dois experimentos de cibersegurança utilizando o protótipo desenvolvido, incluindo cenários mistos com dispositivos tradicionais e IoT em um único experimento. A viabilidade de utilizar as propostas desta pesquisa para gerar dados e apoiar estudos na área de cibersegurança foi avaliada por meio da elaboração de diferentes desafios de detecção de intrusão, desenvolvidos com base nos resultados dos experimentos realizados. Esses desafios serviram como base para testar e validar o modelo de detecção baseado em aprendizado de máquina, que foi aplicado em ambos os cenários propostos, cada um apresentando um nível distinto de complexidade. Os conjuntos de dados e detalhes técnicos para reproduzir os experimentos deste trabalho estão disponíveis para pesquisadores em um repositório público[1], visando possibilitar análises dos dados e a reprodutibilidade dos experimentos por terceiros.

Em resumo, este trabalho conta com as seguintes contribuições: (*i*) a extensão do MENTORED *Testbed* para execução de experimentos multi-arquitetura, incluindo dispositivos IoT e redes sem fio, (*ii*) a execução e análise de experimentos de cibersegurança

[1] https://github.com/mentoredtestbed/artigo-sbrc-2025

neste ambiente, (*iii*) conjunto de dados dos experimentos executados.

Este artigo está organizado da seguinte forma: a Seção 2 aborda os trabalhos relacionados; a Seção 3 detalha as abordagens para extensão do MENTORED *Testbed* e o protótipo implementado; a Seção 4 descreve os experimentos executados, os resultados obtidos, incluindo análises e os modelos de detecção; a Seção 5 discute as limitações do protótipo; e a Seção 6 conclui o trabalho com direções futuras.

2. Trabalhos Relacionados

Há poucos trabalhos na literatura sobre *testbeds* de cibersegurança que incluem dispositivos IoT. Um levantamento recente em [Santana et al. 2024] embasou a seleção de trabalhos relacionados neste artigo e a definição de requisitos como fidelidade, reprodutibilidade, heterogeneidade, escalabilidade, segurança e monitoramento. A Tabela 1 apresenta os trabalhos selecionados, que são comparados com o MENTORED *Testbed*.

Tabela 1. Comparação dos Trabalhos Relacionados

Trabalho	Propósito	Distribuído	Reproduzível	Escala (N# Dispositivos)
[1]	Análise de Vuln.	Não	Não detalhado	Incerto
[2]	Geração de Dataset	Não	Sim	17
[3]	Propósito Geral	Não	Não detalhado	42
[4]	Geração de Dataset	Não	Parcialmente	82
[5]	Propósito Geral	Não	Sim	140
Este Trabalho	Propósito Geral	Sim	Sim	320+

[1] - [Siboni et al. 2018]; [2] - [Moustafa 2021]; [3] - [Thom et al. 2021]; [4] - [Koroniotis et al. 2019]; [5] - [Sáez-de Cámara et al. 2023]

Entre os *testbeds* apresentados, o trabalho [1] descrito em [Siboni et al. 2018] se destaca pela diversidade de sensores e protocolos de rede em uso. O MENTORED *Testbed*, porém, diferencia-se como um *testbed* de propósito geral, não focado na análise de dispositivos IoT físicos. Além disso, o MENTORED *Testbed* possui como seus pilares a reprodutibilidade e escalabilidade, permitindo re-executar cenários com centenas de dispositivos representados a partir de arquivos de definição de experimentos.

O *testbed* [4] descrito em [Koroniotis et al. 2021] utiliza dispositivos IoT físicos e simulados para gerar um conjunto de dados por meio de um gêmeo digital de aeroporto, com dispositivos e protocolos IoT heterogêneos. Embora tenha escala considerável, tal *testbed* não é distribuído e nem monitora dados de aplicação. Em contraste, o MENTORED *Testbed* oferece maior escalabilidade e inclui dados de aplicações, embora tenha menor diversidade de protocolos de enlace.

O trabalho [2] de [Moustafa 2021] descreve um *testbed* voltado à geração de conjuntos de dados, utilizando dispositivos como *smart-TV*, *smartphones*, máquinas virtuais, serviços vulneráveis e auxiliares (DNS, DHCP, etc.). Similar ao MENTORED *Testbed*, captura registros de rede para análise com aprendizado de máquina. No entanto, diferencia-se por seu foco exclusivo na criação de conjuntos de dados e por ser geograficamente centralizado.

O trabalho [3] de [Thom et al. 2021] apresenta um *testbed* de propósito geral que modela uma pequena cidade, incluindo dispositivos IoT físicos e virtuais, redes com e sem fio, e protocolos industriais. Apesar da diversidade, representa apenas 42 dispositivos, centralizados geograficamente, e não detalha um método para reprodutibilidade, aspecto que o diferencia do MENTORED *Testbed*.

O Gotham *Testbed* [5][Sáez-de Cámara et al. 2023] é um ambiente para experimentação de propósito geral baseado na extensão do GNS3 [Grossmann and Duponchelle 2008], utilizando *scripts* para construir dispositivos representados por máquinas virtuais ou *containers*. Similar ao MENTORED *Testbed*, usa *containers* para escalabilidade e permite configurar latência, vazão e *jitter* na rede. No entanto, não é geograficamente distribuído e não inclui enlaces sem fio como o MENTORED *Testbed*.

A solução proposta neste artigo apresenta um *testbed* distribuído com nós espalhados pelo Brasil, utilizando a infraestrutura do Cluster Nacional RNP e uma ilha IoT. Diferentemente de outros trabalhos, suporta mais de 320 dispositivos por experimento, com 232 CPUs lógicos[2] e 976 GB de RAM, sendo 40 CPUs e 40 GB dedicados a dispositivos IoT. Expansões futuras incluem a integração entre a ilha IoT, o Cluster Nacional e a ativação de uma segunda ilha IoT. O MENTORED *Testbed* prioriza a reprodutibilidade e flexibilidade, oferecendo arquivos de definição de experimentos parametrizáveis (tipos de ataque, duração, dentre outros) e os dados gerados.

3. Solução multi-cluster para o MENTORED *Testbed*

Diante dos desafios dos dispositivos IoT e da infraestrutura do MENTORED *Testbed*, em relação aos provedores de recursos como o Cluster Nacional RNP, foram analisadas diferentes formas de integrar dispositivos IoT ao *testbed*, conforme detalhado a seguir:

1. Abordagem de Cérebro Único: os dispositivos IoT são diretamente integrados ao *cluster* Kubernetes existente, simplificando o gerenciamento com um único ponto de contato. Contudo, há desvantagens nesta abordagem, como o tratamento de ciclos de atualização distintos entre os recursos IoT e não IoT e a necessidade de pilha de software otimizada e distinta, como utilização de distribuições Kubernetes mais leves para dispositivos IoT.
2. Abordagem de Cérebro Dividido: os recursos computacionais tradicionais e IoT são mantidos em *clusters* distintos, simplificando o atendimento dos requisitos e ciclos de vida específicos. Porém, essa abordagem exige a reescrita significativa do MENTORED *Testbed*, para que o fluxo de execução de experimentos seja capaz de tratar múltiplos *clusters* Kubernetes em um mesmo experimento.
3. Abordagem de Cérebro Dividido com Abstração: combina os benefícios das abordagens anteriores, evitando suas desvantagens, ao introduzir uma camada de abstração entre o MENTORED *Testbed* e os provedores de recursos (1). Essa camada conecta múltiplos *clusters* Kubernetes físicos em um único *cluster* lógico, permitindo que recursos sejam instanciados de forma unificada. Contudo, a desvantagem está na necessidade de encontrar ou desenvolver essa camada de abstração e integrá-la ao MENTORED *Testbed*.

[2]Incluindo hyperthreading

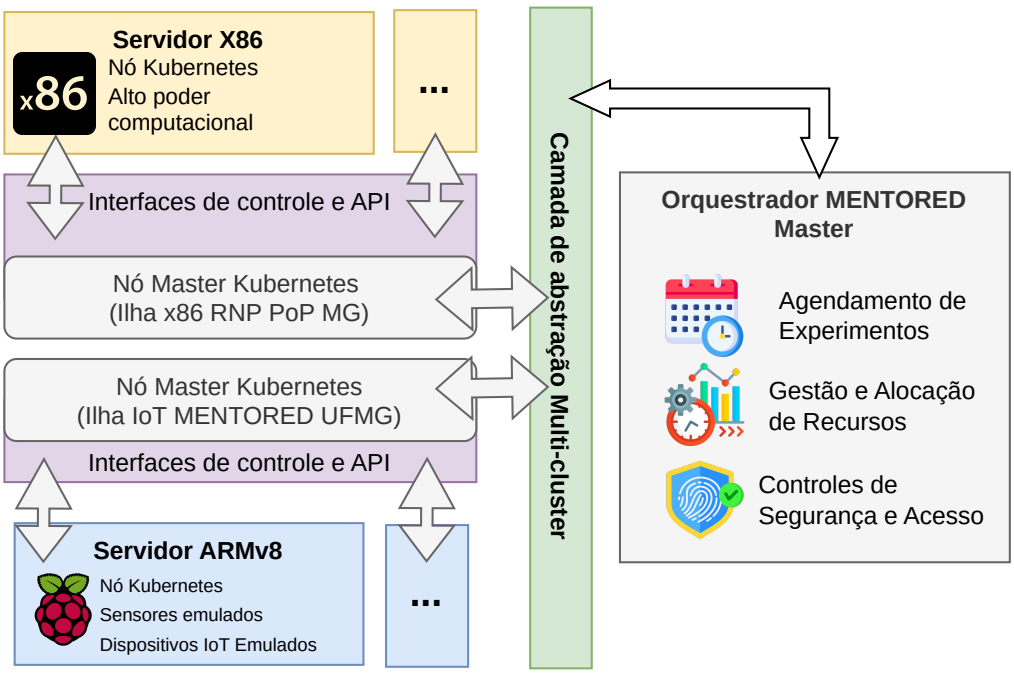


Figura 1. Abordagem de Cérebro Dividido com Abstração

Dentre as abordagens disponíveis, optou-se pela Abordagem de Cérebro Dividido com Abstração. Foram consideradas ferramentas como Admiralty [The Admiralty Authors 2024] e Karmada [Karmada Authors 2025] para atuar como a camada de abstração, sendo realizados testes em ambas para identificar a solução mais adequada. Nos testes, o Admiralty destacou-se pela simplicidade de implantação, porém esta ferramenta abstrai os nós físicos de cada *cluster* como um único nó virtual, dificultando o posicionamento de dispositivos. Isto é, seria necessário um novo mecanismo para que o MENTORED *Testbed* identifique cada nó representado por cada nó virtual. Além disso, a execução de comandos em *containers* via API Kubernetes não estava funcional no momento da avaliação. Desta maneira, o não cumprimento deste requisito pelo Admiralty inviabilizou seu uso como uma camada de abstração no MENTORED *Testbed*.

A ferramenta Karmada foi avaliada quanto à sua adequação à finalidade proposta. Funcionalidades como o *proxy* global e o registro de recursos foram consideradas essenciais e implementadas, mas exigiram várias adaptações no MENTORED *Testbed*. Entre elas, a migração de *Pods* (menor unidade de execução no Kubernetes) para *Deployments* (um componente de mais alto nível que controla *Pods*), já que o Karmada não suporta diretamente instanciar *Pods* em um *cluster* alvo. Além disso, foram avaliados os impactos em recursos customizados adicionados ao Kubernetes, por meio de *plugins* como Multus.

3.1. Protótipo desenvolvido

Como prova de conceito foi desenvolvido o protótipo ilustrado na Figura 2, com duas ilhas de recursos computacionais: a Ilha IoT MENTORED UFMG e a Ilha x86 PoP RNP MG, contendo dispositivos IoT e servidores x86 tradicionais, respectivamente. O MENTORED MASTER é capaz de instanciar experimentos que utilizam ambas as ilhas simul-

taneamente, enquanto o caminho de dados da experimentação é contido em uma VLAN dedicada que interconecta ambas as ilhas.

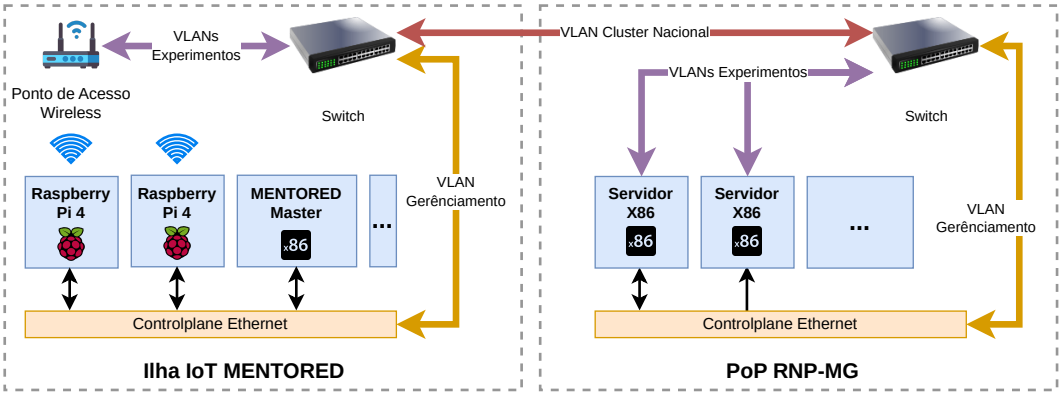


Figura 2. Topologia de rede física para inter-conexão das ilhas

O protótipo utilizou a distribuição Kubernetes RKE 2 (v1.28.14+rke2r1) [SUSE Rancher 2025] com os componentes Cilium v1.16.1 e Multus v4.1.0. Nos dispositivos IoT, foi usado Ubuntu 24.04, enquanto a ilha x86 utilizou a versão 20.04. Estes componentes foram selecionados a partir de uma prospecção de tecnologias, a qual buscou encontrar os componentes compatíveis com a arquitetura AArch64 dos dispositivos IoT assim como Wi-Fi, e a partir disto foram selecionados os componentes que ofereceram o melhor desempenho em testes. Por fim, *playbooks* Ansible são utilizados para instalar todos os componentes mencionados, assim como ferramentas de monitoramento, simplificando a implantação e manutenção do ambiente.

Quanto ao caminho de dados utilizados para a experimentação, foi utilizado uma VLAN da interface de rede do hospedeiro na ilha x86, enquanto na ilha IoT é utilizado a interface Wi-Fi no modo IPvlan *Layer* 2, assim como são realizados ajustes no hospedeiro para permitir a comunicação entre dispositivos (modo promíscuo). É importante notar que os dispositivos IoT se conectam a um roteador Wi-Fi Huawei AX3, utilizando a frequência 5GHz para um melhor desempenho e menor interferência com redes existentes. Tal roteador é configurado no modo ponto de acesso *bridge*, isto é, com as funcionalidades de roteamento e NAT desabilitadas, formando um único domínio de *broadcast* entre sua WAN (VLAN de experimentação) e sua WLAN (dispositivos IoT).

4. Estudo de caso e resultados experimentais

Esta seção tem como objetivo apresentar os cenários experimentais definidos, detalhando suas respectivas entidades, comportamentos esperados e a sequência cronológica das etapas de cada cenário. Esses experimentos visam validar a hipótese de que a arquitetura proposta neste trabalho é útil para a pesquisa experimental em cibersegurança. Além disso, demonstram a viabilidade da integração dessa arquitetura com o MENTORED *Testbed*, possibilitando a definição de experimentos confiáveis, automatizados e reprodutíveis, facilitando a síntese e análise dos resultados.

A Seção 4.1 detalha dois experimentos projetados para comparar cenários de ataques cibernéticos, um de menor e outro de maior complexidade. Os critérios para coleta

de dados durante a execução dos experimentos são apresentados na Seção 4.2, enquanto as análises correspondentes encontram-se descritas na Seção 4.3. Por fim, a Seção 4.4 compara os resultados dos cenários, destacando sua aplicação na criação de desafios de detecção de intrusão e na avaliação de desempenho de modelos baseados em inteligência artificial, seguindo métodos similares aos de [Koroniotis et al. 2021].

4.1. Definição de cenários e experimentos

O cenário 1 tem duração de 300 segundos, durante os quais dispositivos realizam requisições HTTP/1.1 periódicas a um servidor Web Apache, registrando latência e erros. Aos 60 segundos, um ataque Slowloris é iniciado contra o servidor, encerrando-se aos 240 segundos. A Figura 3 ilustra a distribuição das entidades: o servidor Web está em um nó x86 (whx-mg), dez clientes HTTP em um dispositivo IoT (rpi-1-ufmg) e o atacante em outro dispositivo IoT (rpi-8-ufmg).

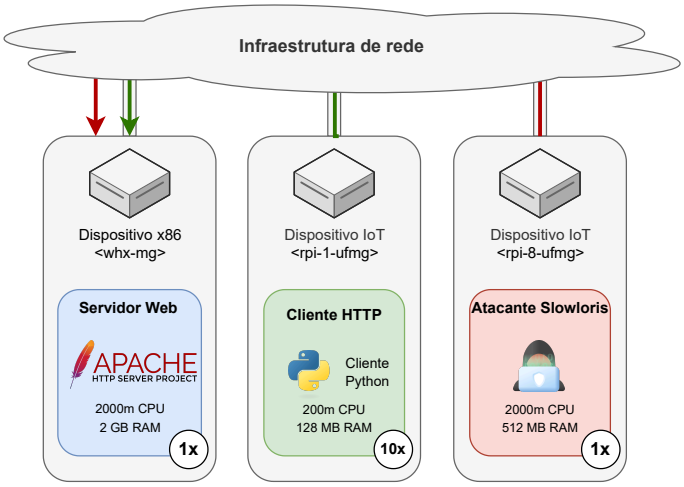


Figura 3. Cenário 1 - Slowloris

Similarmente ao cenário anterior, o cenário 2 também dura 300 segundos, com um servidor Web e clientes HTTP ativos durante todo o experimento. Contudo, conforme ilustrado pela Figura 5, há diferenças notáveis na escala, distribuição das entidades representadas, assim como no tipo do ataque realizado. Especificamente, são representados 320 clientes HTTP distribuídos em 8 nós IoT. Além disso, quatro nós com servidores OpenSSH vulneráveis são incluídos no cenário. É importante notar que, neste cenário, o atacante realiza um ataque de força bruta contra os nós IoT vulneráveis e, após comprometer esses nós, lança um ataque ao servidor Web a partir deles utilizando a ferramenta *hping3*. Por fim, a linha do tempo de ambos os cenários é ilustrada em Figura 4.

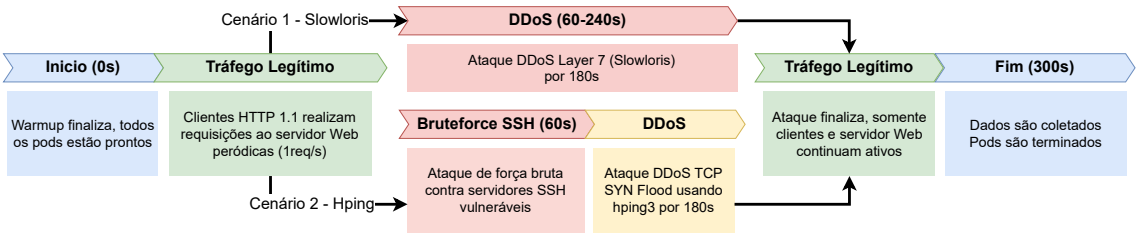


Figura 4. Linha do tempo dos cenários 1 e 2

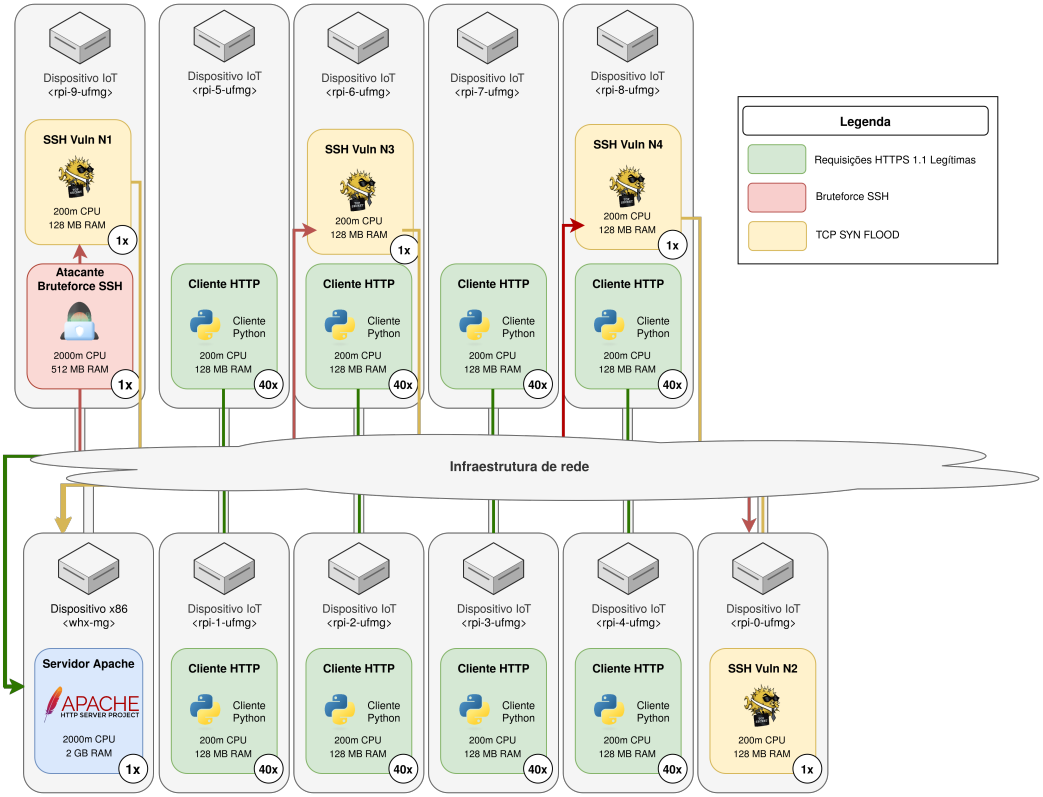


Figura 5. Cenário 2 - Hping

4.2. Conjunto de dados coletados

Similarmente ao descrito em [Meyer et al. 2024], diferentes tipos de informações são coletadas de cada entidade modelada nos dois cenários analisados. Um resumo desses dados coletados está descrito a seguir, enquanto a listagem completa está disponível no repositório do conjunto de dados.

Para todas as entidades, são coletados registros detalhados da inicialização e termino de cada processo, assim como os endereços IP atribuídos. O servidor web armazena o tráfego de rede capturado, assim como *logs* de acesso e erros. Os clientes registram o tempo de início das requisições e informações sobre latência ou falhas de conexão. O atacante tem registros temporais dos ataques Slowloris (cenário 1) e SSH (cenário 2). Por fim, no cenário 2, os nós IoT vulneráveis armazenam logs do ataque Hping3.

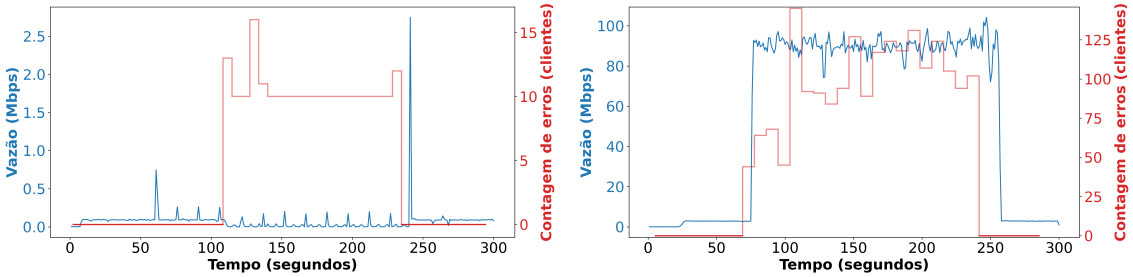
4.3. Resultados

Após a execução dos experimentos, os dados coletados foram analisados utilizando ferramentas próprias desenvolvidas em C++ e Python 3, utilizando bibliotecas como *libpcap*, *numpy* e *matplotlib*. Essas ferramentas são executadas em *containers* para simplificar sua execução em múltiplos ambientes e garantir a reprodutibilidade das análises. A Tabela 2 apresenta a latência média (em milissegundos) em cada período, assim como mostra o número de erros, que corresponde às requisições legítimas não concluídas feitas pelos clientes ao servidor Web. Adicionalmente, a Figura 6a e Figura 6b ilustram a vazão no servidor e contagem de erros dos clientes nos cenários 1 e 2, respectivamente.

Como esperado, o cenário 1 apresenta baixa vazão durante toda sua execução (Fi-

Cenário	Pré-ataque		Durante ataque		Pós-ataque	
	Latência	Erros	Latência	Erros	Latência	Erros
Cenário 1 - Slowloris	13	0	84	212	11	0
Cenario 2 - Hping3	20	0	631	1950	18	15

Tabela 2. Latência média das respostas (ms) e quantidade de erros por período registrados em cada cenário



(a) Cenário 1 **(b) Cenário 2**

Figura 6. Tráfego de rede

gura 6a). No entanto, observa-se uma queda na vazão durante o período de ataque. O ataque teve início aos 60 segundos do experimento, conforme confirmado pela chegada dos pacotes do atacante neste período e pelos registros de eventos do experimento. Contudo, a negação do serviço tornou-se mais efetiva a partir dos 110 segundos, quando mais clientes começaram a enfrentar erros e latências elevadas ao tentar acessar o servidor Web. Em Tabela 3, observa-se que, após o fim do ataque, aos 240 segundos, o acesso dos clientes ao servidor é normalizado.

No cenário 2, observa-se um comportamento significativamente diferente em relação à vazão. O atacante obtêm exito no seu *bruteforce* SSH aos 77, 76, 75 e 76 segundos, respectivamente, em cada nó vulnerável. Imediatamente, tais nós iniciam ataques ao servidor Web com a ferramenta hping3, causando um aumento significativo na vazão do tráfego de rede observada no servidor Apache (Figura 6b). Durante o ataque, a latência média dos clientes foi 31 vezes maior, com 1950 erros registrados. No período pós-ataque, estas métricas foram normalizadas, embora uma pequena taxa de erros remanescentes tenha sido observada, devido ao início dos ataques hping3 pouco após os 60 segundos. Por fim, nota-se um atraso no início das requisições dos clientes, aspecto discutido na Seção 5.

4.4. Detecção de intrusão

Os resultados dos cenários 1 e 2 descritos na Seção 4.2 foram processados para criar desafios de detecção de intrusão, com base nos métodos apresentados por [Koroniotis et al. 2021]. A ferramenta de análise de tráfego de rede OpenArgus[3] foi utilizada para processar os arquivos *pcap* gerados nos experimentos, convertendo-os em arquivos *csv* com os dados de tráfego de rede, em um formato tabular, na qual cada linha representa um fluxo de rede e as colunas representam as características e métricas desse fluxo, como endereços IP de origem e destino, portas de origem e destino, protocolo,

[3]https://openargus.org

tamanho do pacote, entre outros.

Uma inovação tecnológica do MENTORED *Testbed* é a rotulação simplificada de dados. Isso é feito com precisão por meio dos arquivos de metadados gerados nos experimentos, que registram ações e comandos executados por cada nó da topologia. Assim, cada fluxo de rede pode ser associado a um comando, possibilitando a criação de desafios de detecção de intrusão baseados em ataques ou ações legítimas.

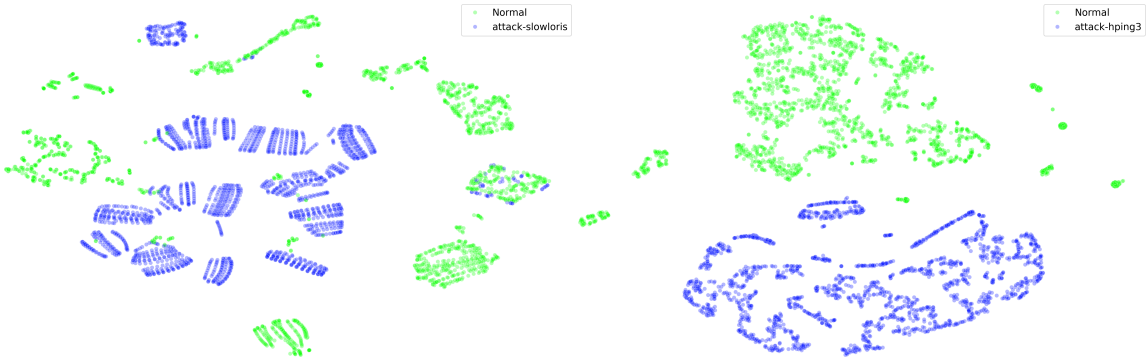
Nos desafios de detecção de intrusão, os modelos de detecção são treinados para identificar se um fluxo de rede é malicioso ou benigno. Para isso, utilizamos a tabela que descreve os fluxos de rede e seus rótulos (base de dados). Neste trabalho, os cenários 1 e 2 foram utilizados para criar duas bases de dados, utilizadas no treinamento e teste de modelos de detecção baseados em aprendizado de máquina.

As bases de dados dos cenários 1 e 2 foram particionadas em 10 partes usando validação cruzada *K-Fold*, onde 9 partes são utilizadas para treinamento e 1 parte para teste, repetindo o processo 10 vezes [Airola et al. 2009]. A métrica de avaliação utilizada foi a AUC (*Area Under the Curve*), que é uma métrica comum para avaliar a qualidade de modelos de classificação binária, que varia de 0 a 1, onde 1 representa um modelo perfeito e 0.5 um modelo aleatório. Também foi realizada uma segunda análise onde 50% dos dados foram separados para treinamento, e diferentes proporções dos dados restantes foram utilizadas para teste, permitindo avaliar o impacto da quantidade de dados de treinamento na qualidade do modelo. Cada experimento foi repetido 10 vezes para garantir a reprodutibilidade dos resultados. Para permitir uma comparação justa, um subconjunto foi extraído aleatoriamente de cada conjunto de dados, com mesmo rótulo para garantir que ambos os cenários tenham o mesmo número de amostras para treinar e avaliar os modelos de detecção.

O modelo de classificação utilizado foi baseado na biblioteca EvalML, uma ferramenta de AutoML que automatiza a seleção de algoritmos e a otimização de hiperparâmetros. Para interpretação e visualização dos dados, foi utilizado o algoritmo t-SNE [Van der Maaten and Hinton 2008], que é uma técnica de redução de dimensionalidade que permite visualizar dados de alta dimensionalidade em um espaço bidimensional, preservando suas relações.

4.4.1. Análise dos resultados de detecção

As bases de dados dos cenários 1 e 2 definidas para o desafio de intrusão de detecção foram utilizadas para criar a Figura 7, que mostra a visualização dos dados em um espaço bidimensional utilizando o algoritmo t-SNE. Cada ponto representa um fluxo de rede, sendo os azuis maliciosos e os verdes benignos. Observa-se que, para ambos os cenários, os fluxos maliciosos e benignos estão bem separados, indicando a viabilidade de um modelo de detecção de intrusão com base nas características dos fluxos de rede. Contudo, no Cenário 1, há interseções entre fluxos maliciosos e benignos, possivelmente devido à baixa intensidade do ataque Slowloris (Cenário 1), que gera pouco tráfego e dificulta a detecção. Por outro lado, o ataque SYN Flood utilizado no Cenário 2 é um ataque de alta intensidade, que gera um grande volume de tráfego, o que facilita a detecção do ataque, características ilustradas nas figuras 6a e 6b.



(a) Cenário 1 (b) Cenário 2

Figura 7. Visualização dos fluxos de rede usando o algoritmo t-SNE

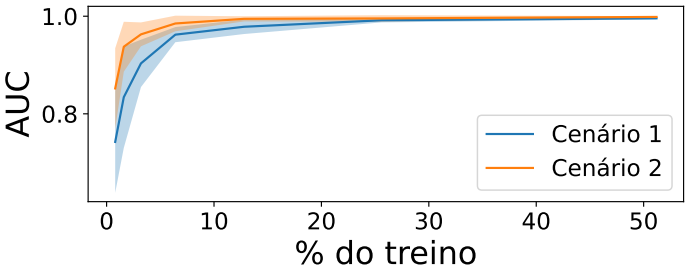


Figura 8. Eficiência de classificadores

Cenário	Falso positivo	Falso negativo
1	27	71
2	14	20

Tabela 3. Contagem de erros dos classificadores

Em todas as execuções, cerca de 40000 amostras foram utilizadas para testar os modelos de detecção de intrusão em cada cenário. A Tabela 3 mostra os erros de classificação de fluxos, onde falsos positivos correspondem a fluxos legítimos classificados como ataque e falsos negativos a ataques classificados como legítimos. O número de erros foi consideravelmente baixo em relação ao total de amostras utilizadas para teste, o que indica que os modelos de detecção de intrusão foram eficazes na detecção de ataques. Essa facilidade de detecção se deve ao alto volume de tráfego gerado pelos ataques utilizados, além da ausência de variáveis mais complexas, como múltiplas aplicações e protocolos de rede, que fogem ao escopo deste trabalho. Observa-se ainda que o Cenário 2 apresentou um número ligeiramente menor de erros em relação ao Cenário 1, o que indica que o ataque SYN Flood é mais facilmente detectável do que o ataque Slowloris.

A Figura 8 ilustra a eficiência dos classificadores utilizados nos experimentos quando treinados com diferentes proporções dos dados. Assim como discutido anteriormente, percebe-se que, em ambos os cenários, os classificadores atingem uma eficiência próxima de 1 quando treinados com cerca de 20% dos dados disponíveis para treino. No entanto, com menos de 20%, a eficiência cai de forma distinta para cada cenário. Nas curvas da Figura, podemos visualizar que para uma mesma quantidade de dados de treino, o Cenário 2 atinge uma eficiência acima de 0.8, enquanto o Cenário 1 atinge uma eficiência abaixo de 0.8, corroborando a hipótese de que o SYN Flood é mais facilmente detectável do que o ataque Slowloris, em concordância com a Tabela 3. Outras métricas pertinentes como *F1-score* e *recall* são detalhadas no repositório deste *dataset*.

Os ataques dos cenários 1 e 2 foram escolhidos por serem simples e facilmente detectáveis, servindo para validar a eficácia do MENTORED *Testbed* na geração de dados e desafios de detecção de intrusão realistas. No entanto, o MENTORED *Testbed* é capaz

de simular uma grande variedade de ataques e cenários de rede, permitindo a criação de desafios de detecção de intrusão personalizados. Essa flexibilidade possibilita experimentos mais complexos e realistas, que podem ser utilizados para treinar e avaliar modelos de detecção de intrusão mais sofisticados em trabalhos futuros.

5. Limitações e ameaças à validade

A implementação atual do suporte a multi-clusters no MENTORED *Testbed* possui algumas limitações. Em especial, a comunicação entre *clusters* distintos exige uma rede *Layer* 2 interconectando todos os nós do experimento. Atualmente, esse requisito é atendido por meio de uma VLAN que conecta ambas as ilhas. No futuro, tal requisito pode ser endereçado por meio de um túnel *Layer* 2 sobre uma rede IP, como por exemplo um túnel VXLAN entre localidades distintas do MENTORED *Testbed*, similarmente ao realizado pela EVPN BGP utilizada pelo Cluster Nacional RNP, simplificando a implantação de múltiplas ilhas do *testbed* em diferentes localidades.

Além disso, é importante notar que o suporte a múltiplos *clusters* Kubernetes operando simultaneamente num mesmo experimento depende da camada de abstração provida pelo Karmada. Tal ferramenta se mostrou adequada para tal finalidade e possui um bom ritmo de desenvolvimento. No entanto, essa tecnologia pode se tornar obsoleta caso não haja interesse contínuo da comunidade científica e da indústria, ou se mudanças em futuras versões do Kubernetes inviabilizarem sua manutenção. Caso tais riscos se materializem, outras ferramentas de abstração podem ser consideradas, ou mesmo um *fork* de uma solução existente, com apenas as funcionalidades estritamente necessárias para o MENTORED *Testbed*.

Durante a implantação do Karmada, identificou-se que a replicação de *Custom Resource Definitions* (CRDs) não estava funcionando. Essas CRDs são essenciais para o MENTORED *Testbed*, pois definem quais interfaces de rede do dispositivo hospedeiro devem ser disponibilizadas a cada nó do experimento. Essa limitação foi contornada com a replicação manual das CRDs entre as ilhas.

Atualmente, a ilha MENTORED IoT UFMG possui um único ponto de acesso WiFi, o que pode ser um fator limitante em certos experimentos. Testes sintéticos indicam que cada dispositivo IoT pode consumir entre 80 e 120 Mbps, enquanto o roteador conta com apenas uma porta WAN de 1 GbE. É importante ressaltar que esta limitação é do atual protótipo e não da arquitetura desenvolvida, e novos equipamentos estão sendo adquiridos para endereçar este gargalo no *testbed*.

Os dispositivos IoT utilizados podem apresentar comportamentos indesejados sob estresse computacional. No Cenário 2, por exemplo, os clientes iniciaram suas requisições com atraso em relação ao programado. Contudo, este comportamento é explicável ao considerar a limitação de recursos impostas aos dispositivos modelados no experimento. Como potenciais soluções a este problema estão o uso de dispositivos IoT de maior poder computacional, assim como um maior número dos mesmos, distribuindo o estresse computacional por dispositivo em cenários de larga escala.

6. Conclusão e trabalhos futuros

Este trabalho apresenta a evolução do MENTORED *Testbed*, incluindo suporte a dispositivos IoT, múltiplos *clusters* Kubernetes, imagens multi-arquitetura, topologia de rede e

enlaces sem fio (Wi-Fi). Dois cenários de ataques DDoS foram executados para validar o protótipo da evolução e gerar um conjunto de dados para pesquisa em cibersegurança. Por fim, esse conjunto foi utilizado na criação de desafios de detecção de tráfego malicioso, empregados na avaliação de modelos de detecção baseados em aprendizado de máquina.

Como trabalhos futuros, propõe-se a expansão do *testbed* com uma segunda ilha de dispositivos IoT em outra localidade. Isso permitirá distribuir geograficamente os recursos IoT, refletindo latências realísticas entre regiões. Novos protocolos e aplicações IoT serão incorporados, e já estão sendo testados tráfegos MQTT, RSTP e DNS, além da modelagem de ataques como HTTP/2 RAPID Reset. Por fim, novos protocolos de enlace podem ser explorados, como Zigbee, LoRaWAN e Bluetooth, assim como estudos sobre redes definidas por software e ataques mais sofisticados e variados.

Agradecimentos

Este trabalho foi financiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP (#2018/23098-0 e #2022/07976-2), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES (PROSUC), Rede Nacional de Ensino e Pesquisa (RNP) e Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (#141179/2021-0).

Referências

Airola, A., Pahikkala, T., Waegeman, W., De Baets, B., and Salakoski, T. (2009). A comparison of auc estimators in small-sample studies. In *Machine learning in systems biology*, pages 3–13. PMLR.

Al-Hawawreh, M. and Sitnikova, E. (2020). Developing a security testbed for industrial internet of things. *IEEE Internet of Things Journal*, 8(7):5558–5573.

Aqeel, M., Ali, F., Iqbal, M. W., Rana, T. A., Arif, M., and Auwul, M. R. (2022). A review of security and privacy concerns in the internet of things (iot). *Journal of Sensors*, 2022(1):5724168.

Grossmann, J. and Duponchelle, J. (2008). Graphical network simulator-3. https://gns3.com/. [Accessed 20-02-2024].

Imperva (2023). 2023 Imperva Bad Bot Report — Resource Library — imperva.com. https://www.imperva.com/resources/resource-library/reports/2023-imperva-bad-bot-report/. [Accessed 15-02-2024].

Karmada Authors (2025). Open, Multi-Cloud, Multi-Cluster Kubernetes Orchestration — karmada — karmada.io. https://karmada.io/. [Accessed 20-01-2025].

Koroniotis, N., Moustafa, N., Schiliro, F., Gauravaram, P., and Janicke, H. (2021). The sair-iiot cyber testbed as a service: A novel cybertwins architecture in iiot-based smart airports. *IEEE Transactions on Intelligent Transportation Systems*.

Koroniotis, N., Moustafa, N., Sitnikova, E., and Turnbull, B. (2019). Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems*, 100:779–796.

Meyer, B. H., Gemmer, D. D., Andrade, A. M., de Mello, E. R., Nogueira, M., and Wangham, M. S. (2022). Criaçao de redes virtuais no mentored testbed: Uma análise experimental. In *Anais do I Workshop de Testbeds*, pages 24–35. SBC.

Meyer, B. H., Gemmer, D. D., de Santana, K. G., Ferreira, J. V., de Mello, E. R., Nogueira, M., and Wangham, M. S. (2024). Criação e análise de datasets de ataque de negação de serviço usando o mentored testbed. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pages 812–825. SBC.

Moustafa, N. (2021). A new distributed architecture for evaluating ai-based security systems at the edge: Network ton_iot datasets. *Sustainable Cities and Society*, 72:102994.

OWASP (2018). IoT Top 10. Technical report, OWSAP.

Sáez-de Cámara, X., Flores, J. L., Arellano, C., Urbieta, A., and Zurutuza, U. (2023). Gotham testbed: a reproducible iot testbed for security experiments and dataset generation. *IEEE Transactions on Dependable and Secure Computing*.

Santana, K. G. Q. d., Schwarz, M., and Wangham, M. S. (2024). Cybersecurity testbeds for IoT: A systematic literature review and taxonomy. *Journal of Internet Services and Applications*, 15(1):450–473.

SentinelOne (2024). Key Cyber Security Statistics for 2025 — sentinelone.com. https://www.sentinelone.com/cybersecurity-101/cybersecurity/cyber-security-statistics/. [Accessed 24-01-2025].

Siaterlis, C., Garcia, A. P., and Genge, B. (2012). On the use of emulab testbeds for scientifically rigorous experiments. *IEEE Communications Surveys & Tutorials*, 15(2):929–942.

Siboni, S., Sachidananda, V., Meidan, Y., Bohadana, M., Mathov, Y., Bhairav, S., Shabtai, A., and Elovici, Y. (2018). Security testbed for internet-of-things devices. *IEEE transactions on reliability*, 68(1):23–44.

Sritharan, K., Elagumeeharan, R., Nakkeeran, S., Mohamed, A., Ganegoda, B., and Yapa, K. (2022). Machine learning based distributed denial-of-services attacks detection and mitigation testbed for sdn-enabled iot devices. In *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–6. IEEE.

SUSE Rancher (2025). RKE2. https://docs.rke2.io/. [Accessed 20-01-2025].

The Admiralty Authors (2024). Multi-Cluster Kubernetes. Simplified. https://admiralty.io/. [Accessed 20-01-2025].

Thom, J., Das, T., Shrestha, B., Sengupta, S., and Arslan, E. (2021). Casting a wide net: An internet of things testbed for cybersecurity education and research. In *2021 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, pages 1–8. IEEE.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Şenel, B. C., Mouchet, M., Cappos, J., Fourmaux, O., Friedman, T., and McGeer, R. (2021). Edgenet: the global kubernetes cluster testbed. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–2.