Criação e análise de *datasets* de ataque de negação de serviço usando o MENTORED *Testbed*

Bruno Henrique Meyer[1], Davi Daniel Gemmer[2], Khalil G. Q. de Santana[3], João Vitor Ferreira[4], Emerson Ribeiro de Mello[5], Michele Nogueira[4,1], Michelle S. Wangham[2,3]

[1]Universidade Federal do Paraná - UFPR

[2]Rede Nacional de Ensino e Pesquisa - RNP

[3]Universidade do Vale do Itajaí - UNIVALI

[4]Universidade Federal de Minas Gerais - UFMG

[5]Instituto Federal de Santa Catarina - IFSC

Abstract. *The use of testbeds in cybersecurity research enhances the creation of representative datasets. Some works focus on creating a dataset using a dedicated testbed for the experimental scenario, limiting the exploration of variations and requiring the creation of new testbeds to generate new datasets. This work describes a workflow that allows the flexible creation of representative datasets using the MENTORED Testbed and presents and analyzes the MENTORED-SBRC2024 dataset with slowloris DDoS attacks. The proposed workflow's main highlight is the ability to recreate datasets through small changes in experiments. The created dataset was used to evaluate intrusion detection models using machine learning to analyze their applicability and representativeness. We executed DDoS scenario variations with up to 352 nodes.*

Resumo. *O uso de Testbeds em pesquisas de cibersegurança potencializa a criação de datasets representativos. Alguns trabalhos concentram-se na criação de um único dataset usando um testbed dedicado para o cenário de experimentação, o que limita a exploração de variações e exige a criação de novos testbeds para gerar novos datasets. Este trabalho descreve um fluxo que permite a criação flexível de datasets representativos usando o MENTORED Testbed e apresenta e analisa o MENTORED-SBRC2024 dataset com ataques DDoS slowloris. O fluxo proposto tem como principal destaque a possibilidade de recriar datasets, por meio de pequenas alterações nos experimentos. Para analisar a aplicabilidade e representatividade, o dataset criado foi utilizado para avaliar modelos de detecção de intrusão que usam aprendizado de máquina. Foram executadas variações de cenários de DDoS com até 352 nós.*

1. Introdução

A evolução constante de ameaças e ataques de segurança exige o desenvolvimento e o aprimoramento contínuo de técnicas de predição, prevenção e detecção. Neste contexto, os *datasets* representativos são fundamentais para soluções de segurança baseadas em aprendizado de máquina [Moustafa 2021, Ferrag et al. 2022], bem como para avaliar e comparar diferentes abordagens de cibersegurança.

A criação de *datasets* para experimentação em cibersegurança é uma tarefa complexa e desafiadora. Métodos que utilizam dados para desenvolver modelos de detecção de ataques, especialmente os baseados em fluxos de tráfego de rede, frequentemente, demandam uma variedade de conjuntos de dados para avaliação [Koroniotis et al. 2019]. Alguns *datasets* de cibersegurança populares têm sido criticados devido à obsolescência de ataques e à falta de realismo nas simulações [Alshaibi et al. 2022]. Diante destes problemas identificados após as publicações destes conjuntos de dados, destaca-se a necessidade de métodos que simplifiquem a atualização desses conjuntos.

Uma abordagem viável para gerar *datasets* consiste na execução de simulações e a reprodução de cenários com ataques cibernéticos atuais e diferentes em uma infraestrutura controlada para coletar dados representativos. Nesse sentido, o uso de *testbeds* tem se destacado como uma solução adequada e flexível para gerar *datasets* de cibersegurança [Koroniotis et al. 2019].

Com um *testbed* é possível reproduzir e replicar experimentos, além de realizar variações para explorar diferentes configurações de *softwares*, topologias e parâmetros de rede [Mirkovic and Benzel 2013, Gomez et al. 2023]. Contudo, na literatura, existem trabalhos [Alsaedi et al. 2020, Peterson et al. 2021] que focam na criação de um único *dataset* usando um *testbed* dedicado para o cenário de experimentação, o que limita a exploração de variações e exige a criação de novos *testbeds* para gerar novos *datasets*.

O MENTORED *Testbed* [Gemmer et al. 2023] foi construído sobre o Cluster Nacional[1] da RNP, que oferece um ambiente confiável e flexível para execução de experimentos em cima recursos físicos que estão distribuídos em diversas regiões do Brasil. O MENTORED *Testbed* foi projetado para conduzir experimentos em cibersegurança e em sua concepção teve como requisitos a reprodutibilidade, flexibilidade e escalabilidade, o que lhe permite criar cenários realistas e abrangentes, contribuindo para a geração de *datasets* mais representativos e úteis para pesquisas em cibersegurança.

Segundo [Veksler et al. 2018], simuladores não permitem representar adequadamente fenômenos complexos como a cadência (*timing*) de eventos. O MENTORED *Testbed* proporciona uma maior fidelidade se comparado com ferramentas baseadas em simulação, pois lida adequadamente com a cadência e permite o uso de softwares, na condução dos experimentos, sem demandar qualquer tipo de adequação dos mesmos.

O MENTORED *Testbed* é composto por três partes: **MENTORED portal** – módulos de interação com o usuário, que oferece possibilidades para definição de experimentos, além da coleta de resultados; **MENTORED *Master*** – que coordena e executa as operações requisitadas através do portal, a fim de alocar os experimentos nos recursos físicos de processamento; **Cluster Nacional RNP** – *cluster* Kubernetes usado para a execução de experimentos de larga escala em recursos físicos [Gemmer et al. 2023].

Este trabalho visa responder à pergunta de pesquisa: É possível definir um fluxo para criação e análise de *datasets* representativos, que usa o MENTORED *Testbed* e considera os critérios de reprodutibilidade, flexibilidade e escalabilidade? Para responder essa pergunta, um fluxo foi proposto e executado, considerando experimentos de ataques de negação de serviço distribuídos, especificamente o *slowloris*. Utilizando o fluxo proposto, gerou-se o MENTORED-SBRC2024 *dataset* a partir da execução de nove experimentos

[1] Anteriormente conhecido como Infraestrutura Definida por Software da RNP (IDS-RNP).

do ataque *slowloris*. O *dataset* resultante foi analisado considerando seu uso para avaliar a eficiência de diferentes algoritmos detecção de intrusão baseados em aprendizado de máquina, conforme explorado em outras pesquisas [Maseer et al. 2021].

Este artigo possui a seguinte estrutura. A Seção 2 apresenta os principais trabalhos relacionados. A Seção 3 detalha o fluxo proposto para criação e análise de *datasets* usando o MENTORED *Testbed*. A Seção 4 contém a definição e especificação de experimentos e métodos utilizados para gerar o *dataset*, além de análises preliminares dos dados gerados. A Seção 5 apresenta um exemplo de uso do fluxo e do *dataset* em um sistema de detecção de intrusão. Por fim, a Seção 6 conclui o trabalho e apresenta direções futuras.

2. Trabalhos Relacionados

Um ataque de DDoS é comumente gerado a partir de uma rede composta por uma grande quantidade de dispositivos, espalhados geograficamente, oriundos de diferentes redes e com diferentes capacidades computacionais e de transmissão de dados. A reprodução fiel de ataques de DDoS em *testbeds* é uma limitação conhecida.

O Deterlab[2] é um relevante *testbed* de pesquisa em cibersegurança cujo fluxo de experimentação compreende: (*i*) criação da topologia usando *scripts* `NS` (*Network Simulator*), (*ii*) criação e execução do experimento via interface gráfica *web*, (*iii*) interação manual via SSH ou automática via *scripts*, (*iv*) modificação e (*v*) término do experimento. Ao finalizar, os dados dos experimentos são salvos em uma pasta compartilhada para análise. Este *testbed* possui limitações relacionadas à usabilidade para acessar e analisar dados dos experimentos e para a experimentação com topologia de redes que considerem a heterogeneidade de dispositivos IoT [Prates Jr et al. 2021].

Em [Koroniotis et al. 2019], os autores utilizaram um *testbed* construído especificamente para criar um conjunto de dados chamado Bot-IoT. O fluxo utilizado não permite adaptações para avaliar experimentos com topologias de diferentes tamanhos. Além disso, o *dataset* gerado considera um número baixo e fixo de dispositivos IoT, possui características inválidas e restrição no número de ataques [Peterson et al. 2021]. Devido à utilização exclusiva de um *testbed* não reutilizável, as limitações mencionadas só poderiam ser superadas mediante à recriação do *testbed*, uma tarefa potencialmente inviável ou que demanda um esforço considerável.

Em [Ferrag et al. 2022] é proposto um *testbed* focado em dispositivos IoT industriais (IIoT) para a criação do *dataset* Edge-IIoTset. Este *dataset* abrange capturas de tráfego de diversos cenários, incluindo múltiplos tipos de ataques DDoS, varredura de portas, ataques *Man-in-The-Middle*, além de tráfego IIoT legítimo. Os autores realizaram diversas transformações neste *dataset* para a extração de características, as quais foram utilizadas para treinar algoritmos de aprendizagem de máquina, visando categorizar o tipo de tráfego. Contudo, o estudo não fornece detalhes sobre como o *testbed* pode ser adaptado ou configurado para a execução de cenários distintos.

Para geração do *dataset* ToN_IoT [Alsaedi et al. 2020], que contempla tráfegos heterogêneos provenientes da IoT, um *testbed* específico também foi construído. O trabalho aborda a coleta de dados de quatro fontes distintas: Zeek[3], *logs* de sistema operacional,

[2]https://www.deterlab.net/

[3]https://zeek.org

registros de sensores e capturas de tráfego. O dataset ToN_IoT foi empregado para avaliar técnicas de detecção baseadas em aprendizado de máquina supervisionado, usando o método de avaliação cruzada (conhecida como *K-Fold*). Assim como os dois últimos trabalhos, este não está centrado no uso de um fluxo de criação e análise de *datasets* usando um *testbed* flexível.

Conforme discutido por [Moustafa 2021], a criação de *datasets* para cibersegurança deve considerar diversas características, como configuração realista de rede, tráfego de rede realista, dados rotulados, fontes heterogêneas, captura completa de interações e pacotes, além de múltiplos cenários com atividades maliciosas. A rotulagem dos dados é essencial para aplicação e avaliação de técnicas de aprendizado de máquina supervisionado. No entanto, os três últimos trabalhos citados não levam em conta a reprodutibilidade dos experimentos. Observa-se a inviabilidade de reexecutar experimentos ou variações de experimentos, pois os *testbeds* foram construídos exclusivamente para gerar um único *dataset*. É comum ainda que esses *datasets* se tornem obsoletos, demandando pequenas modificações como a alteração de topologia de rede e a inserção de novos ciberataques, por exemplo. Portanto, surge a necessidade de um fluxo que considere a reprodutibilidade, para que um *dataset* possa ser atualizado. Além disso, a análise desses *datasets* é crucial para determinar se são necessárias mudanças para ampliar a representatividade e utilidade dos dados em futuras pesquisas em cibersegurança.

No presente trabalho, uma nova abordagem é explorada, colocando a reprodutibilidade e flexibilidade como requisitos para criação de *datasets* representativos, o que permite a evolução simplificada de novos experimentos. Com o auxílio do fluxo proposto neste trabalho, a comunidade científica pode colaborar para criar variações de experimentos, executá-los e compará-los, conforme necessário, utilizando a infraestrutura disponível no Cluster Nacional da RNP e as facilidades de uso do MENTORED *Testbed*. O fluxo descrito neste trabalho não se restringe apenas ao uso de ferramentas ou metodologias que facilitem a criação de *datasets*, mas também contempla os métodos necessários para reprodução dos experimentos.

3. Fluxo para criação e análise de *datasets* utilizando o MENTORED *Testbed*

Esta seção apresenta o fluxo para criação de *dataset*, assegurando que sua reprodução e atualização sejam facilmente realizadas, além de possibilitar análises posteriores. O fluxo contempla a definição, execução e análise de experimentos em cibersegurança, utilizando o MENTORED *Testbed*, conforme ilustrado na Figura 1 e detalhado a seguir:

1. **Definição do experimento**[4]**:** pelo MENTORED Portal, o usuário define os objetivos do experimento, métricas, tipos de ataques ou atividades maliciosas a serem executados, parâmetros do experimento e as topologias de rede;
2. **Alocação de recursos:** o MENTORED *Master* recebe as requisições do Portal e aloca os recursos físicos no Cluster Nacional. Essa alocação envolve a distribuição de máquinas virtuais, contêineres ou outros elementos, conforme necessário;
3. **Execução do experimento:** o MENTORED *Master* coordena a execução do experimento, monitorando as atividades e registrando os resultados obtidos. Além disso, o usuário monitora e acessa nós da topologia em tempo real para verificar se o experimento está sendo executado como o esperado;

[4]Mais informações em https://mentored.dcc.ufmg.br/tutorials

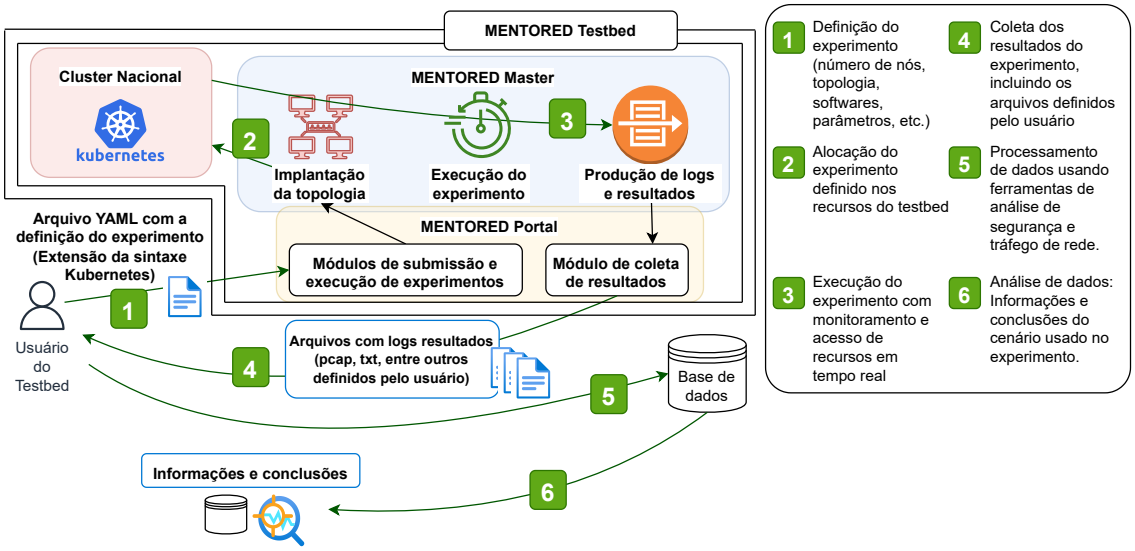


Figura 1. Fluxo para criação e análise de *datasets* usando o MENTORED *Testbed*

4. **Coleta e armazenamento de dados:** durante a execução do experimento os dados são armazenados de forma estruturada e organizada, para que possam ser usados em consultas futuras. Os dados contém as seguintes informações:
   - 4.1. Captura do tráfego de rede como arquivos `.pcapng`;
   - 4.2. Arquivos que evidenciam que o ataque foi realizado com sucesso como histórico de requisições negadas em cenários de ataques DDoS;
   - 4.3. *Logs* relacionados ao serviço ou protocolo utilizado como alvo do ataque;
   - 4.4. Lista e descrição de endereços IPs que identificam tipos de nós da topologia de rede simulada.
5. **Processamento de dados:** o usuário acessa os dados coletados para processá-los, por exemplo, para analisar os arquivos de tráfego de rede. Assim, os *datasets* são criados para analisar fenômenos e resolver desafios como a detecção de ataques;
6. **Análise de dados**: o usuário utiliza modelos de cibersegurança gerados com auxílio do conjunto de dados criado na etapa anterior. Diversos tipos de análises podem ser consideradas nesta etapa [Alshaibi et al. 2022], como a avaliação do balanceamento de tipos de ataques considerados, avaliação de ferramentas para detecção de ataques, entre outros. Por fim, o usuário poderá considerar se é necessário reexecutar o fluxo modificando alguma característica da definição do experimento. Dessa forma, o fluxo pode ser iterado quantas vezes necessário até que os dados gerados atendam às expectativas do usuário.

Apesar de o fluxo descrito delinear seis etapas para geração de *datasets*, a escolha das tecnologias e ferramentas para cada etapa pode representar um desafio significativo. A criação de um *dataset* deve ser guiada por sua finalidade específica, envolvendo a escolha de técnicas adequadas para a definição de experimentos, processamento e análise de dados. A próxima seção apresenta um estudo de caso, utilizando o fluxo proposto neste trabalho, que criou o *dataset* chamado MENTORED-SBRC2024, que pode ser usado para criar e avaliar modelos de detecção de intrusão. Detalhes técnicos serão abordados, desde a definição de experimentos usando arquivos YAML (formato usado pelo MENTORED *Testbed*) até ferramentas para coleta e processamento de tráfego de rede.

4. Estudo de caso e resultados experimentais

O fluxo apresentado na seção anterior foi utilizado para construir cenários nos quais ocorreram ataques DDoS com o tipo específico Slowloris (ver Seção 4.1). Nesses cenários, os atacantes, clientes e servidor *web* alvo foram alocados em diversos nós físicos ou *workers* (recursos de processamento) do Cluster Nacional. Os nós atacantes tinham como objetivo negar a resposta HTTP de um serviço *web* vulnerável, que também era acessado de forma legítima por nós clientes, simulados em *workers* distintos. Na definição dos novos cenários e experimentos (ver Seção 4.2), diversos *scripts* e *softwares* foram utilizados para coletar métricas. Essas métricas foram calculadas processando os dados relacionados ao tráfego de rede e utilizadas para compor o MENTORED-SBRC2024 *dataset* para que auxilie futuras pesquisas em cibersegurança. Os arquivos do *dataset* foram processados para extrair informações dos dados e gerar gráficos discutidos na Seção 4.3.

Este trabalho destaca-se pela facilidade de reprodução e adaptação dos experimentos, diferenciando-se de outros semelhantes. Essa facilidade é garantida pelo fato de que as únicas entradas necessárias do usuário são arquivos YAML que definem os experimentos, cujas sintaxes estendem as sintaxes da tecnologia Kubernetes, amplamente documentadas e conhecidas [Poniszewska-Marańda and Czechowska 2021].

4.1. Slowloris e os comportamentos das entidades participantes dos experimentos

Os ataques DDoS na camada de aplicação envolvem a realização de requisições computacionalmente custosas para interromper ou degradar o serviço para usuários legítimos por meio da saturação de recursos da vítima, como por exemplo, processamento ou memória [Alomari et al. 2012]. Tais ataques são populares devido à sua furtividade e baixo consumo de banda [Moustis and Kotzanikolaou 2013], enquanto possuem um impacto similar aos ataques DoS volumétricos tradicionais. O Slowloris, um tipo de ataque desse tipo, foi descoberto em 2007 e explora vulnerabilidades no protocolo HTTP. Ao enviar requisições HTTP GET incompletas, bloqueia o servidor, mantendo conexões abertas e exaurindo sua capacidade de atender clientes legítimos [Damon et al. 2012]. Um cliente Slowloris pode abrir várias conexões simultâneas, mantendo-as abertas , sendo que essas conexões podem ser mantidas indefinidamente através do envio contínuo de novos cabeçalhos parciais [Damon et al. 2012, Moustis and Kotzanikolaou 2013].

Para todos os cenários (descritos na subseção 4.2), têm-se: (1) uma única vítima – servidor *web* Apache HTTPd versão 2.4.56 vulnerável ao ataque Slowloris e que serve uma simples aplicação *web* desenvolvida com o *micro framework web flask*[5]; (2) um número variável de clientes – fazem requisições GET periódicas ao servidor *web*; (3) um número variável de atacantes – executam a ferramenta `slowloris.py` [Yaltirakli 2015]. A execução da ferramenta é parametrizada para, por exemplo, informar o número de conexões (*sockets*) paralelas abertas simultaneamente, o intervalo de tempo entre o envio de cada cabeçalho e a utilização ou não de identificadores de agente (*User-Agent*) aleatórios.

No experimento foram definidos atacantes que fazem uso de 150 *sockets* (valor padrão da ferramenta slowloris.py) e que fazem uso de 300 *sockets*, sendo que esses valores determinam o número de conexões HTTP em paralelo que o atacante estabelece com o servidor *web*. Cada cliente assume um dos seguintes comportamentos: (1)

[5]https://github.com/pallets/flask

efetua requisições periódicas a cada 500 milissegundos; (2) efetua requisições de forma aleatória, em intervalos que variam de 1 a 30 segundos. A captura do tráfego, por meio da ferramenta *TShark*, é efetuada no nó onde o servidor *web* é executado e, para avaliar a disponibilidade do servidor, um cliente denominado *monitor*, é mantido na mesma região geográfica do servidor.

4.2. Definição de cenários e experimentos

Conforme consta na Tabela 1, foram definidos três cenários (Cn^o) compostos por diferentes experimentos (En^o). A quantidade máxima de atacantes e clientes foi escolhida considerando a capacidade atual do Cluster Nacional. O cenário 1 (`C1`) tem por objetivo avaliar o comportamento de clientes (consulta periódica *vs* consulta com intervalo aleatório) e de atacantes (que faz uso de 150 ou 300 *sockets*). Em específico, o cenário `C1` conta com apenas um atacante e um cliente, e foi proposto para ser usado como referência e comparação. O cenário 2 (`C2`) é composto por quatro experimentos (diferencia o comportamento de atacantes e clientes), sendo que em todos os experimentos foi feito uso de 35 atacantes e 315 clientes. O cenário 3 (`C3`) inclui apenas um experimento com 18 atacantes usando 150 *sockets*, 18 atacantes usando 300 *sockets*, 157 clientes efetuando requisições em intervalos de $500ms$ e 157 clientes efetuando requisições em intervalos aleatórios que variam de 1 a $30s$.

O Cluster Nacional possui *nodes* Kubernetes geograficamente distribuídos pelo Brasil. Para o (`C1`), o *pod* Kubernetes[6] com o servidor *web* foi alocado no estado do Espírito Santo, o *pod* atacante em Goiás e o cliente em Pernambuco. Para o (`C2`) e (`C3`), o servidor foi alocado no Espírito Santo e os clientes e atacantes tiveram uma divisão na proporção de 1 atacante para 9 clientes sendo geograficamente distribuídos no Distrito Federal, Goiás, Pernambuco e Pará. Os limites de recursos impostos para o servidor e clientes foi de 2GB de memória RAM e 2 CPU. Os atacantes foram limitados a 128MB de memória RAM e 2 CPU. O *plugin* Kubernetes responsável pela comunicação entre os *pods* foi o MACVLAN, que funciona de forma similar a um *switch* conectado à interface do *host*, um dispositivo físico que é compartilhado com as interfaces virtuais.

Tabela 1. Configuração de Cenários

Cenário	Atacantes		Clientes	
	Total	Nº de *sockets*	Total	Intervalo de requisições (segundos)
`C1`-E1	1	150	1	0,5
`C1`-E2	1	300	1	0,5
`C1`-E3	1	150	1	[1, 30]
`C1`-E4	1	300	1	[1, 30]
`C2`-E1	35	150	315	0,5
`C2`-E2	35	300	315	0,5
`C2`-E3	35	150	315	[1, 30]
`C2`-E4	35	300	315	[1, 30]
`C3`-E1	18	150	157	0,5
	18	300	157	[1, 30]

Para processar os arquivos de captura de tráfego de cada cenário, foi utilizada a ferramenta OpenArgus[7]. Essa ferramenta permite a extração de valores para o cálculo

[6]O Kubernetes executa sua carga de trabalho colocando contêineres em *pods* para serem executados em *nodes*.

[7]https://openargus.org/

das métricas por fluxo de conexão identificada no tráfego. Cada fluxo foi rotulado como malicioso ou legítimo, de forma que um problema de classificação pudesse ser definido, no qual as métricas calculadas de cada fluxo pudessem ser utilizadas por modelos de detecção para determinar o rótulo. Assim como apresentado pelos autores do Bot-IoT [Koroniotis et al. 2019], que também utilizaram a ferramenta OpenArgus, os experimentos deste trabalho utilizaram 10 principais atributos obtidos a partir das métricas coletadas do OpenArgus para definir cada fluxo: "Proto", "Sport", "Dport", "Seq", "StdDev", "Min", "Mean", "DstRate", "SrcRate" e "Max", utilizadas para detectar categorias de tráfego [Moustafa 2021]. A técnica t-SNE [Van der Maaten and Hinton 2008] de visualização de dados multidimensionais, implementada na biblioteca Scikit-Learn, foi utilizada para criar gráficos de dispersão que ilustram a representação de cada cenário, o que será explorado na Seção 5.

4.3. Análise dos resultados

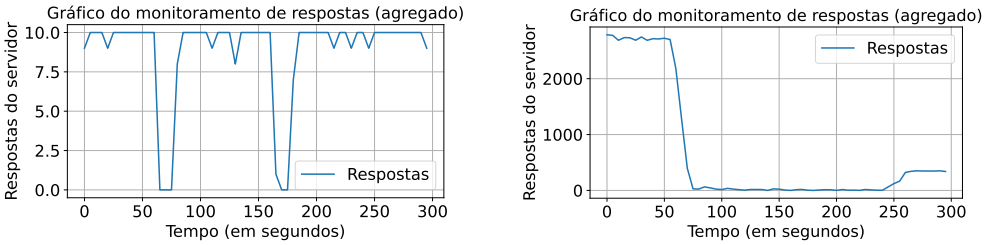
Os dados coletados após a execução dos cenários da Tabela 1 foram usados para formar o *dataset* MENTORED-SBRC2024[8]. Diferentes aspectos foram coletados, tais como:

1. Tráfego de rede: arquivo *.PCAPNG* contendo a captura de tráfego realizada no servidor *web* por meio da ferramenta *TShark*;
2. Registro de acesso do servidor: arquivo texto contendo o *log* de acesso do servidor Apache;
3. Registro de requisições respondidas (por cliente): arquivo no formato CSV contendo o tempo e latência de cada requisição respondida;
4. Mapeamento de endereços: arquivo no formato JSON com os endereços de cada instância no cenário, como clientes, atacantes e o servidor;
5. Dados rotulados com OpenArgus: arquivo no formato CSV contendo métricas de fluxos de rede (agrupamento de pacotes) criados com o OpenArgus e um rótulo indicando se o fluxo é relacionado a um ataque (valor 1) ou tráfego legítimo (0). A rotulagem foi realizada por um *script* que usa o arquivo de mapeamento de endereço. Os dados estão estruturados de forma simples para que se possa avaliar técnicas bem conhecidas de aprendizado de máquina [Moustafa 2021].

O *dataset* gerado foi analisado por meio de um *script* em Python, que emprega as bibliotecas Pandas e Numpy para realizar a leitura e processamento dos dados, além da biblioteca Scapy para a extração de dados adicionais, para permitir o cálculo de métricas como a vazão a partir do arquivo de captura de tráfego (*pcapng*) gerado. Por fim, os gráficos das métricas obtidas foram criados por meio da biblioteca Matplotlib.

Tendo em vista o número de cenários criados, uma amostragem destes será examinada a seguir. Contudo, a análise completa dos cenários está disponível juntamente ao *dataset*. Na Figura 2a é possível observar o número de requisições HTTP atendidas pelo servidor durante o cenário `C1`-E1. Como é possível observar, o serviço é brevemente negado em dois momentos. Enquanto isto, o cenário `C2`-E2 conta com um intervalo de serviço negado mais significativo (veja Figura 2b), que demostra que a negação do serviço escala conforme o número de atacantes e o número de conexões (*sockets*) por atacante.

[8]O *dataset* MENTORED-SBRC2024 está disponível em: https://drive.google.com/drive/folders/1q4DY-ATD-H4GlbT8HXv7mSgJ1xmRck1I?usp=sharing

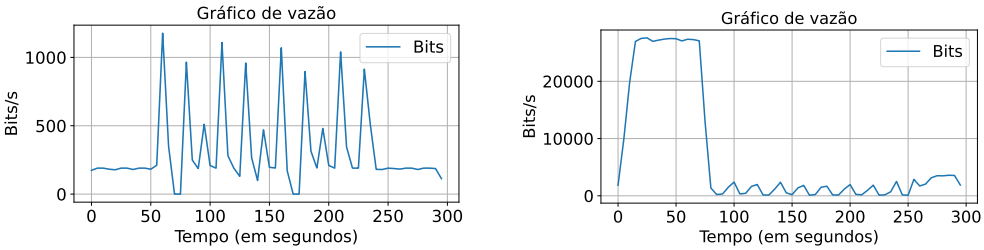


(a) Cenário C1-E1 **(b) Cenário C2-E2**

Figura 2. Monitoramento de respostas do servidor

As Figuras 3a e 3b ilustram a vazão dos experimentos `C1`-E1 e `C2`-E2, respectivamente. Nota-se que no primeiro cenário há vários picos de vazão devido a proporção $1:1$ de clientes e atacantes. Enquanto isto, o segundo cenário apresenta um patamar de vazão mais elevado devido ao maior número de clientes, contudo, o mesmo cai consideravelmente durante o intervalo de execução dos ataques.

Outro ponto notável no cenário `C2`-E2 é a vazão observada nos períodos pré, durante e posteriores ao ataque. Observa-se que a vazão (Figura 3b) inicia em um patamar superior a `25.000 bps`, cai para níveis inferiores a `5.000 bps` e não retorna após os ataques ao patamar inicial. Estipula-se que isto se deve aos últimos pacotes dos atacantes serem enviados aos `240s`, os quais podem manter os *sockets* ocupados até o *timeout* do servidor, o qual pode ocorrer após os 60 segundos restantes até o final do experimento.



(a) Cenário C1-E1 **(b) Cenário C2-E2**

Figura 3. Vazão durante os cenários

Por fim, a Tabela 2 apresenta um sumário da quantidade de pacotes e respostas HTTP por cenário, assim como o número de requisições com sucesso e requisições com erros[9]. É possível observar que diferentes padrões de negação de serviço estão presentes em cada cenário. Por exemplo, o cenário `C2`-E1 contém um total de 40.384 respostas HTTP, das quais 1.553 tiveram status de erro (cerca de 3.8%). Por outro lado, o cenário `C2`-E4 apresenta uma percentagem próxima a 34%. Outra característica consiste no fato de que, em alguns cenários, como o `C1`-E3 e `C1`-E4, houve mais respostas HTTP com status de erro do que status "OK", o que pode ser explicado pelo fato que nesses cenários os clientes não aproveitam o período de pré-ataque para obter respostas com status OK do servidor. Essa diversidade de dados permite o estudo e avaliação de diversas características relacionadas a pesquisas em cibersegurança, como por exemplo, o desbalanceamento de tipos de tráfego [Alshaibi et al. 2022].

[9]Status da requisição: estas métricas foram coletadas a partir do cabeçalho HTTP `Status Code`, sendo as requisições com sucesso demarcadas pelo código 200 e os erros com respostas `Bad Request` e

Tabela 2. Estatísticas dos experimentos executados

Cenário	Pacotes	Respostas HTTP	Status OK	Status Erro	% Erros
C1-E1	18.849	1.566	966	600	38.3%
C1-E2	24.179	1.900	852	1.048	55.1%
C1-E3	14.050	1.082	482	600	55.4%
C1-E4	20.469	1.519	472	1.047	68.9%
C2-E1	423.993	40.384	38.831	1.553	3.8%
C2-E2	429.432	80.629	39.533	1.530	1.8%
C2-E3	70.990	4.291	2.641	1.650	38.4%
C2-E4	71.030	4.251	2.779	1.472	34.3%
C3-E1	233.848	21.151	19.519	1.632	7.7%

5. Análise de bases de dados e aplicação de Sistemas de Detecção de Intrusão

Nesta seção, será apresentado um método para análise do *dataset* criado na seção anterior usando ferramentas para detecção de intrusão. Esta análise buscou verificar se o fluxo proposto neste trabalho pode gerar dados adequados para avaliar métodos de cibersegurança baseados em dados.

Foram utilizadas técnicas baseadas em aprendizado de máquina em Sistemas de Detecção de Intrusão (SID) para comparar diferentes cenários do *dataset*. Os cenários escolhidos foram: C1-E1, C2-E1 e C2-E4. Esses cenários foram selecionados devido às diferentes complexidades de configuração do experimento em cada um deles, levando em consideração o número de pacotes gerados, o método de acesso dos clientes ao servidor e os parâmetros do ataque *slowloris*. O uso dessas técnicas baseadas em aprendizado tem sido explorado em diversas pesquisas que buscam melhorar a eficiência da detecção [Tsai et al. 2009, Maseer et al. 2021]. Assim como nos trabalhos citados, a métrica F1-Score, descrita na Equação 1, foi utilizada para avaliar a eficiência da detecção de diferentes algoritmos baseados em aprendizado de máquina.

$$F_1 - score = 2 * \frac{\text{Precisão} + \text{Revogação}}{\text{Precisão} * \text{Revogação}} \tag{1}$$

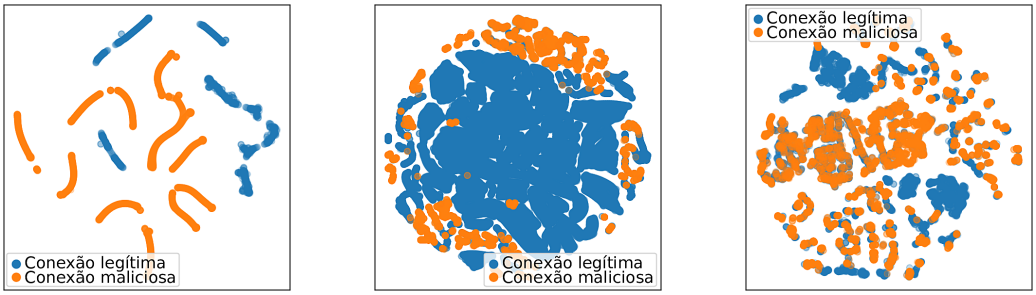
Os algoritmos de aprendizado utilizados foram implementados através da biblioteca Scikit-Learn da linguagem de programação Python. Os modelos de classificadores utilizados foram: *KNeighborsClassifier* (KNN), *SVC* (*Support Vector Machine*), *DecisionTreeClassifier* (DT), *RandomForestClassifier* (RF), *MLPClassifier* (MLP), *AdaBoostClassifier* (ADB) e *GaussianNB* (GNB). Devido ao foco do estudo ser o fluxo para criação e análise de *datasets*, as análises e possíveis melhorias das técnicas de SDI não foram aprofundadas.

Em cada cenário, a lista de fluxos foi dividida utilizando a metodologia K-*Fold* estratificado, na qual 10 conjuntos foram criados de forma aleatória, mantendo a mesma proporção de tráfego malicioso e legítimo. Cada classificador considerado neste trabalho foi treinado e avaliado 10 vezes para cada cenário, de modo que um dos 10 conjuntos fosse usado para treinamento e os demais para teste, gerando F1-Scores. Em seguida, foi calculada a média do F1-Score para cada classificador em cada cenário.

Request Timeout, código 400 e 408, respectivamente.

5.1. Análise dos resultados de detecção

A Figura 4 apresenta os resultados do algoritmo t-SNE para três cenários considerados na base de dados gerada nesta pesquisa. Observa-se que o Cenário C1-E1 possui o padrão mais simples de dados, com uma clara separação entre fluxos maliciosos e legítimos. No entanto, essa separação não é tão evidente nos cenários C2-E1 e C2-E4, onde há uma sobreposição visível entre os dois tipos de tráfego. Além disso, pode-se notar que o cenário C2-E1 contém uma proporção significativamente maior de fluxos legítimos, a maioria deles distintos dos maliciosos. Por outro lado, o cenário C2-E4 apresenta uma menor quantidade de conexões legítimas, mas com uma alta interseção com o tráfego malicioso. No cenário C2-E4, é esperado que os tráfegos legítimos e maliciosos sejam não monótonos, uma vez que tanto as requisições dos clientes quanto o ataque possuem parâmetros que definem aleatoriedade, gerando características únicas para cada conexão.



(a) Cenário C1-E1 (b) Cenário C2-E1 (c) Cenário C2-E4

Figura 4. Visualização dos fluxos de tráfego processados utilizando a técnica t-SNE. Pontos próximos representam conexões com características similares extraídas com a ferramenta Argus.

Na Figura 5, é apresentado o F1-Score para indicar a capacidade de diferentes classificadores em detectar ataques DDoS em cada cenário. Observa-se que, como esperado, o cenário C1-E1 foi aquele em que os classificadores tiveram maior facilidade em detectar ataques DDoS, alcançando a melhor eficiência possível com qualquer classificador. Apesar do cenário C1-E1 não ser representativo quando comparado com cenários realistas, o conjunto de dados desse cenário pode ser utilizado como referência ao ser comparado com outros cenários.

Em cenários mais complexos (C2-E1 e C2-E4), não foi possível obter um resultado de detecção tão bom quanto no cenário C1-E1. Especificamente, quando o cenário C1-E1 é desconsiderado, o cenário C2-E1 obteve o maior F1-Score de 0.86 com o algoritmo *GaussianNB*, enquanto o cenário C2-E4 obteve 0.78 também com o *GaussianNB*. Essa pequena diferença pode ser explicada pela complexidade envolvida no ataque do cenário C2-E4, que utilizou um parâmetro do slowloris que cria agentes aleatórios simulando o acesso de diferentes dispositivos.

O fluxo proposto foi empregado em várias instâncias, com ajustes nas definições de cenários, a fim de investigar o comportamento de Sistemas de Detecção de Intrusões (SDI). Dada a diversidade e os distintos níveis de complexidade dos experimentos, a análise dos resultados nesta seção sustenta a hipótese de que o fluxo atende os requisitos de escalabilidade e reprodutibilidade e gera *datasets* representativos e que podem ser utilizados para analisar SDIs.

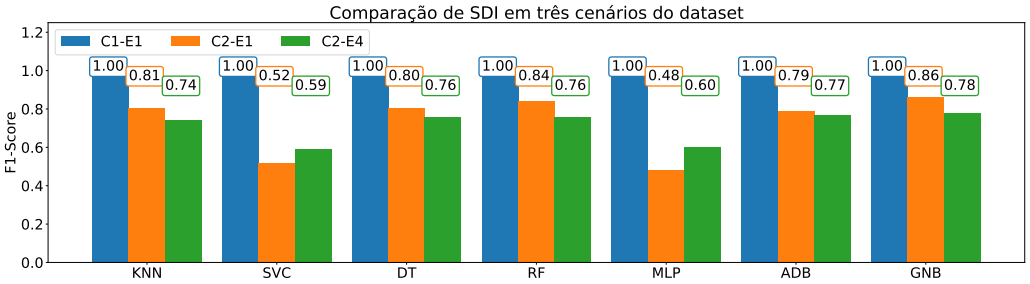


Figura 5. Eficiência de diferentes algoritmos de aprendizado de máquina para classificar fluxos como legítimos ou maliciosos.

Contudo, a obtenção de *datasets* representativos e realistas ainda é uma limitação observada em diversos *testbeds* e *datasets* populares [Alsaedi et al. 2020]. Para superar esses problemas, o fluxo proposto neste trabalho pode ser utilizado para elaborar cenários ainda mais complexos com maior diversidade de serviços, ataques e tráfego de rede. Por fim, embora apenas técnicas associadas a SDI tenham sido aplicadas nos experimentos deste estudo, ressalta-se que o fluxo proposto também é aplicável a outras técnicas relacionadas à cibersegurança.

5.2. Limitações e ameaças à validade

É importante destacar algumas premissas feitas para garantir a implementação efetiva do estudo experimental. O MENTORED *Testbed* foi utilizado como base para a geração de dados e simulação de ciberataques. No entanto, é necessário que o *testbed* ofereça diferentes garantias, como isolamento para não interferir em outras tecnologias e usuários do Cluster Nacional. Este isolamento é provido por meio de tecnologias nativas ao Kubernetes, como *namespaces*, contudo tal isolamento ainda está sendo desenvolvido e testado no *testbed*. Além disso, pressupõe-se a consistência das execuções, garantindo que a reexecução dos experimentos neste trabalho produza resultados equivalentes.

Atualmente, o Cluster Nacional conta apenas com dispositivos x86-64, que foram utilizados para simular topologias de rede nos *workers* com alto poder de processamento. A ausência de dispositivos IoT neste *cluster* impede a execução de experimentos relacionados a esses dispositivos. Logo, a análise do comportamento de dispositivos IoT em ataques DDoS em grande escala é inviável. Outra limitação conhecida deste estudo é a simplicidade dos cenários analisados, em que apenas dois tipos de comportamentos de clientes, ataques (*slowloris*) e um serviço alvo foram considerados. Além disso, o ataque *slowloris* não se trata de um ataque volumétrico, e, portanto, não deve ser utilizado para avaliar tais cenários. Embora essas características sejam suficientes para observar aspectos do ataque DDoS, elas não contemplam a diversidade de atividades maliciosas encontradas na internet.

6. Conclusão e trabalhos futuros

Este trabalho apresenta uma proposta de fluxo para a criação e análise de conjuntos de dados para cibersegurança utilizando o MENTORED *Testbed*. Para comprovar a efetividade do processo de criação de conjuntos de dados, o fluxo foi utilizado para gerar o MENTORED-SBRC2024 *dataset*, que consiste em nove cenários com variações de experimentos com ataques DDoS do tipo slowloris.

Os resultados demonstraram que pequenas variações no padrão de comportamento do ataque e dos clientes refletiram nos dados e nos resultados dos experimentos, proporcionando uma variedade de situações que podem ser utilizadas por pesquisas futuras que empreguem esse *dataset*. Além disso, um estudo de uso do *dataset*, considerando um sistema de detecção de intrusão baseado em aprendizado de máquina, revelou que os cenários mais complexos apresentaram maior dificuldade para a detecção, destacando a utilidade do conjunto de dados como um *benchmark* para avaliação de técnicas de detecção. Em comparação com trabalhos similares, a principal característica do fluxo proposto é a sua simplicidade e flexibilidade na reprodução de cenários, permitindo a criação de vários *datasets* sem a necessidade de reconfigurar a infraestrutura, graças ao MENTORED *Testbed*. Os experimentos realizados possibilitaram simulações de ataques em topologias com até 352 nós distribuídos em diversas regiões do Brasil. A capacidade de escalabilidade dos experimentos está diretamente relacionada à infraestrutura em constante desenvolvimento e expansão do Cluster Nacional.

Para ampliar a contribuição deste trabalho, todos os dados e códigos pertinentes foram disponibilizados junto com o conjunto de dados, inclusive os dados processados para aplicação de técnicas de aprendizado de máquina.

Para trabalhos futuros, é recomendada uma análise mais aprofundada dos dados de cada cenário do MENTORED-SBRC2024 *dataset*, incluindo os arquivos de *logs* dos servidores e as respostas dos clientes. Além disso, para o avanço desta pesquisa pretende-se incluir dispositivos IoT na topologia dos experimentos de simulação, explorando protocolos relacionados à IoT para adicionar maior complexidade e realismo à captura de tráfego. Por fim, sugere-se explorar mais ataques e acessos legítimos aos serviços, utilizando o fluxo de criação e análise proposto, a fim de gerar novos *datasets* que considerem uma maior diversidade de aplicações em cibersegurança.

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

Alomari, E., Manickam, S., Gupta, B. B., Karuppayah, S., and Alfaris, R. (2012). Botnet-based distributed denial of service (ddos) attacks on web servers: Classification and art. *International Journal of Computer Applications*, 49.

Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., and Anwar, A. (2020). Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems. *Ieee Access*, 8:165130–165150.

Alshaibi, A., Al-Ani, M., Al-Azzawi, A., Konev, A., and Shelupanov, A. (2022). The comparison of cybersecurity datasets. *Data*, 7(2):22.

Damon, E., Dale, J., Laron, E., Mache, J., Land, N., and Weiss, R. (2012). Hands-on denial of service lab exercises using slowloris and rudy. In *Proceedings of the*

2012 Information Security Curriculum Development Conference, InfoSecCD '12, page 21–29, New York, NY, USA. Association for Computing Machinery.

Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., and Janicke, H. (2022). Edge-iiotset: A new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning. *IEEE Access*, 10:40281–40306.

Gemmer, D. D., Meyer, B. H., de Mello, E. R., Schwarz, M., Wangham, M. S., and Nogueira, M. (2023). A scalable cyber security framework for the experimentation of ddos attacks of things. In *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pages 1–7. IEEE.

Gomez, J., Kfoury, E. F., Crichigno, J., and Srivastava, G. (2023). A survey on network simulators, emulators, and testbeds used for research and education. *Computer Networks*, 237:110054.

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.

Maseer, Z. K., Yusof, R., Bahaman, N., Mostafa, S. A., and Foozy, C. F. M. (2021). Benchmarking of machine learning for anomaly based intrusion detection systems in the cicids2017 dataset. *IEEE access*, 9:22351–22370.

Mirkovic, J. and Benzel, T. (2013). Deterlab testbed for cybersecurity research and education. *Journal of Computing Sciences in Colleges*, 28(4):163–163.

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.

Moustis, D. and Kotzanikolaou, P. (2013). Evaluating security controls against http-based ddos attacks. In *IISA 2013*, pages 1–6.

Peterson, J. M., Leevy, J. L., and Khoshgoftaar, T. M. (2021). A review and analysis of the bot-iot dataset. In *2021 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, pages 20–27.

Poniszewska-Marańda, A. and Czechowska, E. (2021). Kubernetes cluster for automating software production environment. *Sensors*, 21(5):1910.

Prates Jr, N. G., Andrade, A. M., de Mello, E. R., Wangham, M. S., and Nogueira, M. (2021). Um ambiente de experimentação em cibersegurança para internet das coisas. In *Anais do VI Workshop do testbed FIBRE*, pages 68–79. SBC.

Tsai, C.-F., Hsu, Y.-F., Lin, C.-Y., and Lin, W.-Y. (2009). Intrusion detection by machine learning: A review. *expert systems with applications*, 36(10):11994–12000.

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

Veksler, V. D., Buchler, N., Hoffman, B. E., Cassenti, D. N., Sample, C., and Sugrim, S. (2018). Simulations in cyber-security: a review of cognitive modeling of network attackers, defenders, and users. *Frontiers in psychology*, 9:691.

Yaltirakli, G. (2015). Low bandwidth dos tool. slowloris rewrite in python. https://github.com/gkbrk/slowloris.