Um Ambiente de Experimentação em Cibersegurança para Internet das Coisas

Nelson G. Prates Jr. [1,4], Allex Magno Andrade[2], Emerson Ribeiro de Mello [2], Michelle Silva Wangham[3], Michele Nogueira[1,4]

[1]Universidade Federal de Minas Gerais - UFMG
nelsonprates@dcc.ufmg.br, michele@dcc.ufmg.br
[2]Instituto Federal de Santa Catarina - IFSC
mello@ifsc.edu.br, allex.m@aluno.ifsc.edu.br
[3]Universidade do Vale do Itajaí - UNIVALI
wangham@univali.br
[4]Universidade Federal do Paraná - UFPR

Abstract. *The large scale of the Internet of Things requires complex testbeds capable of supporting experimental scenarios with sufficient scale to evaluate cybersecurity solutions against botnet-based DDoS attacks efficiently. This work describes an architecture for the MENTORED Testbed, an experimentation environment created on the Software Defined Infrastructure from the National Research and Education Network (IDS-RNP). The testbed creates experiments with wide area networks, cloud technologies, and wireless devices inserted in the IDS-RNP servers. We analyzed the testbed's behavior through a case study that automatically reproduces the network captures in a realistic scenario to evaluate botnet detection proposals.*

Resumo. *A grande escala da Internet das Coisas exige estruturas complexas capazes de suportar cenários experimentais com escala suficiente para avaliar eficientemente soluções de cibersegurança contra ataques DDoS baseados em botnets. Este trabalho descreve uma arquitetura para o MENTORED Testbed, um ambiente de experimentação criado sobre a Infraestrutura Definida por Software da Rede Nacional de Ensino e Pesquisa (IDS-RNP). O testbed cria experimentos com redes de longa distância, tecnologias de nuvem e com dispositivos sem fio inseridos nos servidores do IDS-RNP. O comportamento do testbed foi analisado por meio de um caso de uso que reproduz automaticamente capturas de rede em um cenário realístico para avaliar propostas de detecção de botnets.*

1. Introdução

Os ataques distribuídos de negação de serviço (do inglês, *Distributed Denial of Service* - DDoS) representam o tipo de ataque mais nocivo e difícil de se defender na Internet [Vishwakarma and Jain 2020]. Estes ataques compreendem um vetor de outros que capturam dispositivos vulneráveis e geram um tráfego massivo com o intuito de tornar indisponível o serviço atacado. Os ataques DDoS comprometem a reputação das instituições, causam prejuízos, criam incerteza e pânico entre o público [Kolias et al. 2017]. O maior ataque DDoS registrado até então aconteceu em 2020 e gerou 2,3 Tbps de tráfego, sendo este aproximadamente 70% maior que seu antecessor, o qual em 2018 havia gerado

1,35 Tbps, e 200% maior que o terceiro ataque já registrado, que atingiu 1,1 Tbps em 2016 [BBC 2020, A10-Networks 2020, Wired 2018].

As técnicas de ataque DDoS mais sofisticadas são difíceis ou impossíveis de contra-atacar, e mesmo os ataques mais simples causam interrupções significativas [Benzel 2011]. Este problema é fundamentalmente difícil, pois os atacantes empregam uma série de outros ataques para aumentar o potencial e tornar a carga de um ataque virtualmente indistinguível do tráfego para uso legítimo de um serviço [Siboni et al. 2019]. Além disso, a Internet das Coisas (do inglês, *Internet of Things* - IoT) motivou um aumento exponencial na quantidade de dispositivos conectados, como câmeras IP, eletrodomésticos, etc. Os dispositivos da IoT se comunicam por meio de redes heterogêneas, com conexões fim-a-fim, empregam tecnologias de nuvem e, em sua maioria, possuem limitações relacionadas à capacidade de processamento, memória e energia. Devido ao baixo poder computacional, é comum que estes dispositivos não implementem mecanismos de segurança robustos resultando no surgimento constante de novas vulnerabilidades que dificultam o desenvolvimento de soluções eficientes. Estes ataques alternativos compreendem falsificar os endereços de origem, capturar novos dispositivos ou serviços comuns na Internet e se camuflar das próprias fontes de ataque, que são normalmente usuários legítimos inconscientes. Portanto, interromper um ataque em sua fonte apresenta uma série de desafios tecnológicos e administrativos.

A proposição de soluções de cibersegurança eficazes contra ataques DDoS sofisticados e de grande porte requer testes e experimentação em ambientes controlados e realistas. Na literatura existem estudos focados na criação de tecnologias e metodologias avançadas para implantar *testbeds* de pesquisa experimental de cibersegurança [Siboni et al. 2019, Muchtar et al. 2018, Gunduz and Das 2018]. Estes estudos apresentam uma série de requisitos (como fidelidade, validade, entre outros) que devem ser considerados na criação de novos *testbeds* de segurança em redes. Os *testbeds* de IoT consideram apenas aplicações específicas [Kumar and Lim 2019, Gunduz and Das 2018], não são isolados o suficiente para executarem os códigos maliciosos, capazes de infectar dispositivos automaticamente, ou não consideram o contexto de redes heterogêneas com conexões fim-a-fim como na Internet [Adjih et al. 2015, Arora et al. 2006]. Por outro lado, as soluções contra ataques DDoS não podem ser testadas de forma eficaz na Internet, pois causariam interrupções inaceitáveis para alguns serviços críticos dependentes desta infraestrutura mundial. Estes fatos evidenciam a necessidade de ambientes de testes (*testbed*) capazes de simular ou emular dispositivos IoT e a Internet em geral.

Este trabalho descreve a modelagem e os esforços que estão sendo realizados no escopo do projeto MCTIC/FAPESP MENTORED em direção à criação de um ambiente controlado para experimentação em cibersegurança, a fim de oferecer uma infraestrutura para que pesquisadores possam demonstrar a viabilidade de suas soluções para redes seguras e com escala realista. A arquitetura do MENTORED *Testbed* e o seu primeiro caso de uso representam os resultados destes esforços. O WP4 desenvolveu a arquitetura do MENTORED *Testbed* com base nos requisitos levantados junto aos outros pacotes de trabalho do projeto e com base em outros ambientes de experimentação de sucesso, como o DETERLab e o FIT IoTLab [Benzel 2011, Adjih et al. 2015]. Além disso, o *testbed* proposto opera sobre a Infraestrutura Definida por Software da Rede Nacional de Ensino e Pesquisa (IDS-RNP), que tem por objetivo ofertar um ambiente externo para o desen-

volvimento de projetos de rede.

O caso de uso apresentado neste artigo parte do princípio que o *testbed* deve permitir criar experimentos de rede, de forma automática, para que possa reproduzir situações de cibersegurança e testar mecanismos de defesa que se baseiam no comportamento do tráfego de *botnets*. O estudo reproduziu 4 cenários de rede conforme as capturas de rede para apoiar a avaliação de duas propostas de detecção de *botnets*, baseadas nas características do tráfego. Os resultados apontam que a reprodução criou cenários realísticos de rede na IDS-RNP com fidelidade para suportar a avaliação das propostas de detecção de tráfego de *botnets*.

O restante do artigo está organizado como segue. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 apresenta a arquitetura do MENTORED *Testbed*. A Seção 4 descreve o caso de uso para avaliar ferramentas de detecção de *botnets* e discute os resultados obtidos. Por fim, a Seção 5 conclui o trabalho e apresenta as direções futuras.

2. Trabalhos Relacionados

Os *testbeds* são ambientes para experimentação que servem como uma plataforma de avaliação e aceleram o desenvolvimento de propostas para um amplo espectro de áreas de pesquisa, como, por exemplo, prevenção, predição, detecção e mitigação de ataques cibernéticos. Além disso, os *testbeds* também podem servir como um ambiente educacional e de treinamento para fornecer cursos práticos. Projetar *testbeds* é uma tarefa desafiadora, que envolve definir os requisitos ideais para os objetivos pretendidos, a rede overlay e o orquestrador dos experimentos. A seguir, apresentam-se os ambientes experimentais existentes na literatura relacionados a este trabalho.

O FIT-IoT Lab [Adjih et al. 2015] implementa cenários para prototipação de redes de sensores sem fio reais, ou seja, o usuário experimentador tem acesso desde a camada física até a camada de aplicação dos dispositivos para desenvolver soluções da IoT. Este *testbed* emprega diversas arquiteturas de processadores voltadas para a IoT e permite o desenvolvimento de experimentos com base em bibliotecas de código aberto ou sistemas operacionais existentes nos dispositivos disponibilizados. No entanto, o FIT-IoT Lab não provê isolamento de tráfego entre experimentos de diferentes usuários, pois todos os experimentos em execução compartilham os mesmos canais físicos de comunicação.

O DETERLab [Benzel 2011] é um *testbed* de cibersegurança pioneiro no provimento de ferramentas de experimentação no contexto da Internet. Este *testbed* usa tecnologias próprias de experimentação, baseadas em prototipação de rede e em orquestração de software para os usuários criarem facilmente experimentos de segurança em redes. A prototipação de rede se baseia em uma ferramenta de emulação de topologias de redes de larga escala, com dispositivos reais ou virtuais. A orquestração considera alocar os softwares necessários para gerar tráfego, simular ataques DDoS, clientes normais, aplicações do mercado, entre outros, conforme as especificações programadas pelos usuários [Benzel 2011]. Apesar de ser uma referência nos requisitos de segurança, pois isola de forma controlada os experimentos e está adequada para permitir experimentos de grande escala e com altas cargas de tráfego, o DETERLab não considera o contexto de redes sem fio e de longo alcance como na IoT. Ambos os *testbeds* citados possuem plataformas de acesso aos recursos físicos dos dispositivos e permitem a visualização dos experimentos e o controle da disponibilidade de recursos físicos, tanto pela interface

gráfica quanto por linha de comandos.

A maioria dos *testbeds* de cibersegurança na literatura é baseada em técnicas de virtualização [Kumar and Lim 2018]. Aqueles voltados à IoT são específicos para uma aplicação [Muchtar et al. 2018, Gunduz and Das 2018] e não suportam experimentos de cibersegurança ou não compreendem os dados do tráfego de rede [Gyrard and Serrano 2015]. As soluções de cibersegurança são cruciais para a IoT. Portanto, um *testbed* de cibersegurança na IoT deve considerar uma infraestrutura heterogênea com dispositivos reais com e sem fio que suporte a criação de experimentos de rede independentes das aplicações, sendo capaz de avaliar o comportamento dos ataques e as soluções de cibersegurança.

3. Visão Geral do MENTORED Testbed

O projeto MENTORED tem como objetivos identificar, modelar e avaliar comportamentos maliciosos associados à IoT de forma a auxiliar na construção de soluções avançadas para possibilitar: prevenção, predição, detecção e mitigação de ataques de DDoS. O projeto se divide em quatro pacotes de trabalho (*Work Packages* - WPs), sendo os dois primeiros pacotes de trabalho (WP1 e WP2) dedicados à prevenção e à predição de botnets e ataques DDoS, respectivamente; o terceiro pacote (WP3) é dedicado à detecção e à mitigação de ataques DDoS; e o quarto pacote (WP4) foca em criar um ambiente, denominado MENTORED *Testbed*, para permitir a experimentação de sistemas de cibersegurança.

O MENTORED *Testbed* é um ambiente para experimentação em cibersegurança para avaliar soluções contra ataques de alto impacto, como os ataques DDoS, gerados por dispositivos da IoT. Este ambiente vem sendo definido tomando como base a Infraestrutura Definida por Software da RNP (IDS-RNP) e as demandas dos outros pacotes de trabalhos do projeto MENTORED. A infraestrutura IDS-RNP se baseia na plataforma de orquestração de contêineres Kubernetes[1] e funciona como um ambiente externo para o desenvolvimento de projetos de rede e nuvem. A Figura 1 apresenta a visão geral dos servidores dispostos fisicamente em 13 Pontos de Presença (*Point of Presence* - PoP) da RNP, sendo 5 servidores localizados na região nordeste e 8 servidores localizados nas regiões sul, sudeste e centro-oeste. Os circuitos de rede do nordeste e uma conexão com o sudeste possuem 100Gbps, os demais circuitos possuem entre 10Gbps e 40Gbps de capacidade. Cada nó do IDS possui um *switch* gerenciável e um servidor de virtualização equipados minimamente com 2 processadores Intel® Xeon® E5-2630, 64GB de memória RAM e disco com 4TB de armazenamento.

A plataforma *Kubernetes* gerencia o ciclo de vida de aplicativos e serviços em contêineres, usando métodos que fornecem monitoramento, escalabilidade e alta disponibilidade. Cada contêiner é uma área isolada dentro do sistema operacional e possui seu próprio sistema de arquivos, compartilhamento de CPU, memória, espaço de processo para executar serviços ou aplicações individualmente. Estes contêineres são logicamente alocados em *pods*, conjuntos de contêineres que compartilham a mesma área do sistema operacional e de rede. Um conjunto de *pods* se comunica através da rede e forma um *cluster*. Os servidores físicos (Workers) executam o sistema operacional, os pods e os módulos de controle. O *Kubernetes* emprega um módulo *Master* com uma API que administra a comunicação de rede entre os workers para criar *clusters* em múltiplos *workers*

[1] `https://kubernetes.io/`, Acessado em: 22/02/2021

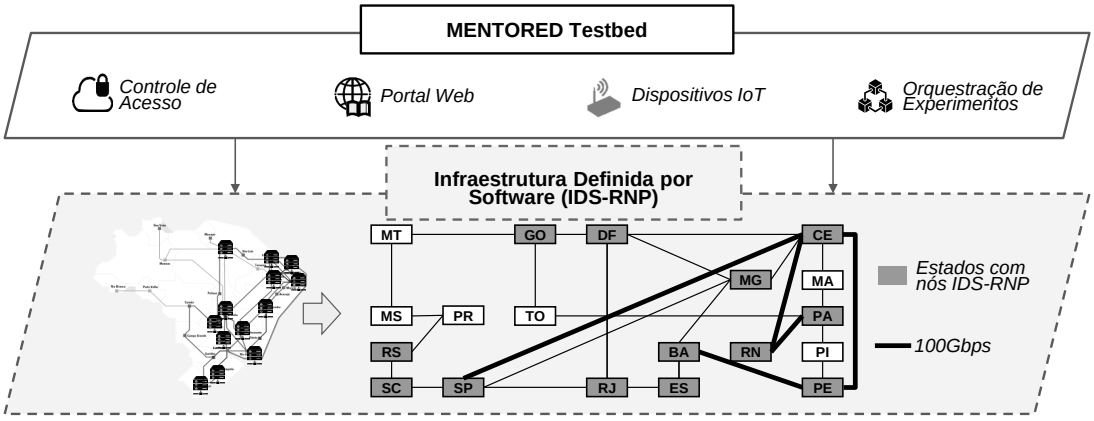


Figura 1. Visão Geral

remotos. O *Kubernetes Master* recebe *scripts* de configuração e controla localmente cada *Worker* para executar serviços e aplicações de forma portátil em *clusteres*.

Os *Workers* executam três módulos locais básicos: o *Kubelet, Proxy e de Networking*. O *Kubelet* é um agente local do *Kubernetes* para administrar os pods hospedados localmente em cada *Worker*. O módulo de *Proxy* administra as regras locais de rede, intermediando as conexões e controlando os acessos. O módulo de *Networking* distribui as regras para cada *Proxy* e estabelece a comunicação entre os contêineres. Então, um usuário do *Kubernetes* pode definir como seus aplicativos devem ser executados e como eles devem interagir com outros aplicativos ou com a Internet. Estas características permitem o desenvolvimento modular do *testbed*. Além disso, o *Kubernetes* compreende primitivas de software livre, para o desenvolvimento de novas versões capazes de cumprir os requisitos estudados sobre redes heterogêneas de longa distância.

O IDS-RNP auxilia a concepção do MENTORED *Testbed* ao fornecer esta estrutura de nível nacional com o *Kubernetes* e a equipe técnica que auxilia a implantação dos novos requisitos. Esta infraestrutura possibilita a criação de cenários controlados com as redes de longa distância, recursos computacionais e softwares bem conceituados tanto na academia quanto no mercado. O MENTORED *Testbed* utiliza estas características do IDS-RNP para configurar automaticamente experimentos e para testar mecanismos de cibersegurança, com cenários de rede realísticos, escalável e com segurança suficiente para não prejudicar a disponibilidade da infraestrutura.

O WP4 do projeto MENTORED tem como objetivo a implantação de um ambiente usando tecnologias e metodologias avançadas para pesquisa experimental sob demanda em cibersegurança com foco em cenários realísticos da IoT. Portanto, a equipe do projeto realizou um levantamento de requisitos considerando os requisitos do *testbed* de cibersegurança DETERLab [Benzel 2011], do *testbed* de IoT FIT-IoT Lab [Adjih et al. 2015] e nas definições apresentadas pelos pesquisadores dos outros pacotes de trabalho do projeto. O grupo escolheu analisar estes *testbeds*, pois estes são pioneiros em experimentação em redes com acesso aos dispositivos físicos e porque possibilitam que o próprio experimentador defina o comportamento dos agentes em cada cenário experimental, diferentemente dos demais *testbeds* que emulam os dispositivos ou são baseados em uma aplicação específica. Assim, chegou-se aos seguintes requisitos para o MENTORED *testbed*:

- **Fidelidade**: se refere à capacidade de obter precisão suficiente na reprodução dos fenômenos específicos, sendo estudados em um experimento particular, dentro de um ambiente que pode ou não corresponder a qualquer rede real existente. Assim, além da representação física de uma rede real, o *testbed* pode também considerar e empregar modelos acurados para realizar avaliações fiéis;
- **Validade**: as limitações do próprio *testbed* não devem distorcer acidentalmente os resultados de um experimento. O *testbed* deve identificar e denunciar as violações dessas condições experimentais exigidas, alertando o usuário para possíveis falhas de validade do experimento;
- **Escalabilidade**: O *testbed* deve ser capaz de prover suporte a experimentos em tamanho suficiente para ser representativo para capturar efeitos complexos dos ataques relacionados ao tráfego massivo de dados na escala da Internet;
- **Segurança**: exige que o *testbed* garanta que nenhum código ou usuários maliciosos obtenham o acesso indevido ou prejudiquem outras infraestruturas de rede, informações ou códigos do próprio *testbed* ou da Internet em geral;
- **Reprodutibilidade**: garante que um experimento, uma vez executado, pode ser exportado e, em seguida, executado em um ambiente idêntico em um momento posterior, para produzir resultados idênticos;
- **Transparência**: o *testbed* deve possibilitar o monitoramento em tempo real e não intrusivo do tráfego de rede e dos recursos computacionais, além de empregar ferramentas de visualização destes recursos tanto de forma gráfica quanto pela linha de comandos;
- **Perspectiva centrada no usuário**: o *testbed* deve oferecer liberdade para os usuários desenvolverem novas classes de ferramentas que facilitam as pesquisas experimentais, além das funções tradicionais da pesquisa experimental, como a configuração de experimentos e o monitoramento do tráfego;
- **Acesso em Tempo Real**: o software de orquestração deve fornecer acesso em tempo real aos dispositivos, para que um usuário possa redefinir, reprogramar e monitorar o estado de cada dispositivo durante a execução dos experimentos.

A arquitetura proposta neste trabalho se apoia nas redes de longa distância, nos recursos com alta disponibilidade e na modularidade da plataforma *Kubernetes* para criar cenários experimentais com redes heterogêneas, com conexões fim-a-fim e tecnologias de nuvem. Além disso, este trabalho também avalia e avança a IDS-RNP ao projetar e adicionar as funcionalidades específicas de um *testbed* para cibersegurança na IoT com base nos requisitos levantados. A Figura 2 ilustra a arquitetura do MENTORED *testbed*, que é composta por três componentes principais: as *Ilhas MENTORED*, as *Redes Virtuais de Controle e Teste* e o *MENTORED Master*.

As **Ilhas MENTORED** constituem-se de ilhas do IDS-RNP que foram acrescidas de um Ponto de Acesso sem fio (*Access Point* - AP) e um conjunto de placas Raspberry PI 4 para formarem uma rede local sem fio. Esses novos dispositivos poderão ser usados para aumentar a heterogeneidade de equipamentos (ARM e X86) e meios de transmissão e, com isso, prover a fidelidade do *testbed* ao criar redes IoT. Para suportar tal heterogeneidade, utiliza-se a plataforma Kubespray [2], um modulo *Kubernetes* que administra *clusters* multi-arquiteturais. As placas Raspberry PI 4 também possuem conexão de rede cabeada,

[2] `https://kubespray.io/`, Acessado em: 22/02/2021

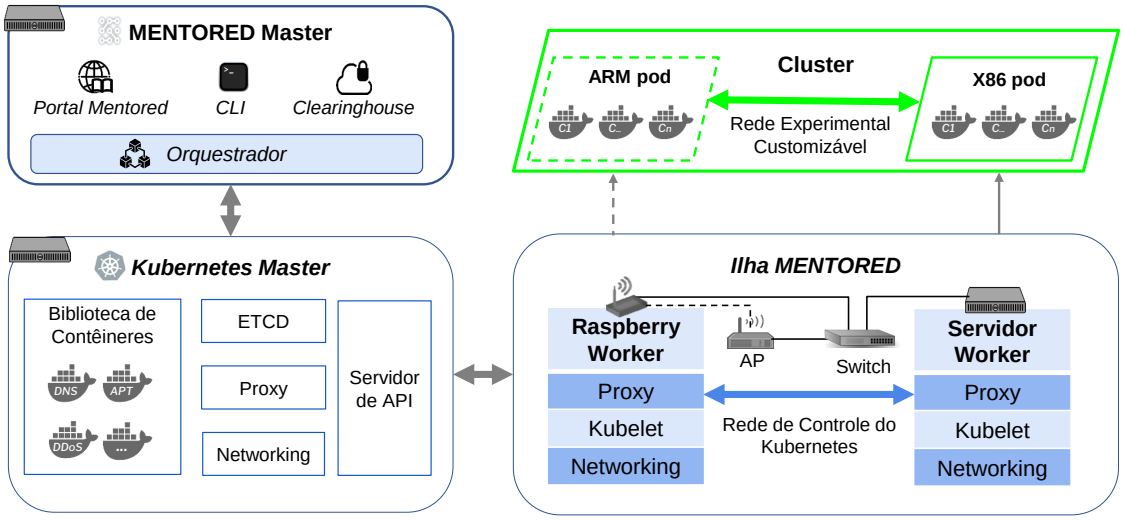


Figura 2. Arquitetura do MENTORED

porém, a mesma só é usada para o tráfego de controle e gerência dos dispositivos. Ou seja, o tráfego dos experimentos só trafega pela interface de rede sem fio.

As **Redes Virtuais de Controle e Experimentação** servem para trafegar a comunicação entre os componentes administrativos, responsáveis pela criação dos experimentos e a configuração dos dispositivos e a carga gerada pelos experimentos, respectivamente. Estas redes podem ser configuradas por diferentes módulos baseados nos *Conteineres Networks Interfaces* (CNI) do *Kubernetes*. Exemplo disso é o módulo *Weave Net*, que foi implantado na primeira versão do IDS-RNP e que possibilita a descoberta automática entre os contêineres, divide as interfaces virtualmente e cria redes L2 customizadas. Esta CNI tem como característica permitir que qualquer participante consiga atingir qualquer outro participantes, o que fere o requisito de isolamento entre os experimentos. A segunda versão do IDS disponibiliza ainda a CNI Calico que implementa nativamente políticas de segurança dentro da rede, forçando criar regras de comunicação entre os elementos. Diversas CNIs podem ser implantadas em conjunto. Além disso, essa modularidade permite que o *testbed* defina os melhores módulos de rede ou desenvolver um próprio conforme as especificidades do projeto.

O **MENTORED Master** é uma camada de software sobre a API (do inglês, *application programming interface* - API) do Kubernetes que administra a iteração entre os quatro módulos de software. Ele controla os demais componentes do *testbed*: o *Portal MENTORED*, a *ClearingHouse (CH)*, a *Interface de Linha de Comandos (do inglês, Command-Line Interface - CLI*) e o *Orquestrador*. O *Portal MENTORED* e a *CH* lidam com a interação com o usuário, gestão de identidade e controle de acesso mediante uma interface gráfica para o usuário (do inglês, *Graphical User Interface* - GUI). A CLI fornece comandos específicos para utilizar o orquestrador *MENTORED* e controlar os componentes do *testbed*. O *Portal MENTORED* será um site para os usuários se cadastrem e, por meio da *CH*, adquiram as credenciais necessárias para estabelecer uma conexão remota e segura com a CLI. O Orquestrador faz a interface entre os demais módulos e a API do *Kubernetes* para fornecer as funções específicas de criação de experimentos, controle e monitoramento dos recursos. A API do Orquestrador simplifica a configuração dos

cenários de rede e emprega as ferramentas necessárias automaticamente para cumprir os requisitos definidos. Assim, o usuário autenticado reserva os recursos de hardware, configura os contêineres a serem executados e, por fim, cria e submete um *script* de descrição do experimento pretendido conforme os comandos específicos da API. O Orquestrador interpreta o *script* e configura automaticamente a infraestrutura com base no *Kubernetes* para disponibilizar os recursos necessários e alocar os módulos para cada *namespace*.

Este conjunto de componentes permite que o MENTORED *Testbed* atinja os requisitos estudados e aloque serviços pré-estabelecidos em uma biblioteca de contêineres específicos. Então, caso um usuário precise de um servidor DNS para executar experimentos, o *testbed* possui um contêiner com um conjunto de configurações prontas para executar avaliações sobre este tipo de serviço na biblioteca. Esta funcionalidade colabora o requisito *Perspectiva Centrada no Usuário*, pois metodologias de pesquisa experimental são, elas mesmas, uma área de pesquisa ativa e os avanços nesta área permitem a criação de novas metodologias de pesquisa [Benzel 2011]. Neste sentido, tanto o WP4 quanto os usuários podem criar conjuntos de contêineres com experimentos configurados para colaborar com novas metodologias de pesquisa e com a biblioteca de contêineres.

O desenvolvimento do *testbed* segue ciclos iterativos de desenvolvimento e teste tendo como prioridade atender as demandas de próprio projeto. As métricas de avaliação do MENTORED *Testbed* estão baseadas nos objetivos de cada experimento e em melhorar cada requisito supracitado. Este modelo permite que o *testbed* esteja em constante desenvolvimento e alinhado com as demandas recorrentes entre os diferentes experimentos. O foco principal deste trabalho foi conduzir experimentos a fim de verificar a aderência do *testbed* com os requisitos dos demais pacotes de trabalho do MENTORED, como a execução de um único experimento por vez, fidelidade, validade e reprodutibilidade.

4. Caso de Uso

Esta seção descreve o primeiro caso de uso do ambiente do MENTORED *Testbed*, que corresponde a uma demanda dos pesquisadores para avaliar duas propostas de detecção *online* de *botnets*, baseadas nas mudanças de conceito ocorridas nas distribuições estatísticas relacionadas ao tráfego de rede. Neste experimento, os cenários de rede foram criados para avaliar como o MENTORED *testbed* apoia essas duas propostas. O experimento reproduziu capturas de tráfego de *botnets* em um cenário de rede com enlaces reais criados automaticamente. A reprodução do tráfego substitui a implementação dos ataques e evita incidentes de segurança como o vazamento de um código malicioso para os outros dispositivos que não fazem parte do experimento ou até para a Internet. Esta abordagem permite avaliar outras propostas de cibersegurança baseadas em análise de tráfego e nas características físicas dos canais de rede.

A partir de uma captura de tráfego, deve-se interpretar as conexões existentes nela, criar uma réplica da rede física com os contêineres e inicializar a reprodução do tráfego em sincronia. Assim, o usuário desenvolve e posiciona a sua proposta de detecção para executar junto da réplica da rede. O cenário do experimento executava apenas contêineres *Docker* e ainda não suportava a plataforma *Kubernetes*. Neste caso de uso, o IDS-RNP ofereceu uma rede virtual de camada 2 (L2) entre as ilhas posicionadas na Universidade Federal do Rio Grande do Sul e na Universidade Federal de Minas Gerais. Cada ilha recebeu automaticamente um dos módulos de reprodução denominados cliente ou servidor. O

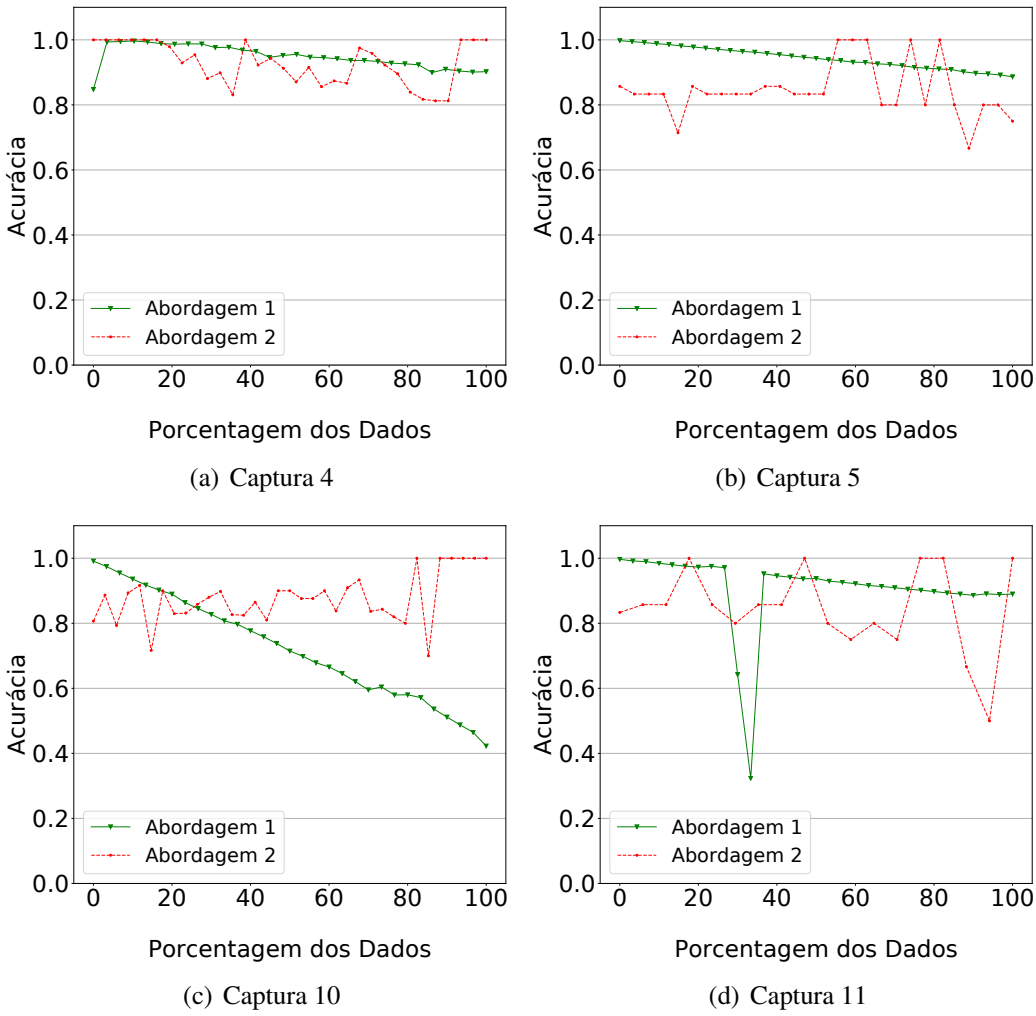
módulo cliente criava uma série de contêineres distribuídos para reproduzir a rede local. O módulo servidor reproduzia o tráfego dos dispositivos externos, portanto, implementava um contêiner que enviava e recebia as mensagens, conforme os endereços fora da faixa de IP da rede local da captura de tráfego. O terceiro módulo *Master* dividia a captura por IP, distribuía os contêineres entre as Ilhas e iniciava os contêineres, respeitando a ordem dos eventos no tráfego.

O desenvolvimento dos módulos utilizou as ferramentas *Tshark e Docker* e as linguagens de programação *Python 3* e *Shell script*. A ferramenta *Tshark* divide as capturas de tráfego por endereço IP da rede local e da rede externa. Os módulos cliente e servidor foram desenvolvidos em *Python 3* com a biblioteca Scipy para ler a captura de tráfego, trocar os endereços de destino da rede local e enviar os pacotes. Os *scripts* em *Shell* compreendiam o módulo *Master*, no qual extraia as conexões com base nos endereços de IP das capturas e criava a rede local de clientes e o servidor com contêineres *Docker*. Então, neste cenário cada pacote foi transmitido conforme a temporização dos seus respectivos fluxo de rede entre cada cliente e o servidor.

O experimento avaliou duas propostas de detecção de *bots*, que utilizam técnicas de aprendizado de máquina, tendo como base as características dos fluxos de rede, como o número de bytes e o intervalo entre os pacotes. No entanto, estas técnicas de aprendizado de máquina geram modelos específicos de detecção que ficam defasados quando ocorrem as mudanças de conceito, por exemplo, as geradas pelo surgimento de novas *botnets*, e perdem a eficiência. As propostas avaliadas compreendem uma abordagem que evita a defasagem do modelo, empregando duas abordagens de detecção de mudanças de conceito. A **abordagem 1** observa as distribuições dos dados e a **abordagem 2** se baseia na observação dos resultados dos classificadores. A detecção de mudanças de conceito serve como um gatilho para adaptar os modelos de forma online.

Neste experimento, utilizaram-se as capturas de tráfego 4, 5, 10 e 11 da base de dados CTU-13 [Garcia et al. 2014]. As capturas 4 e 5 compreendem a comunicação dos *bots* com o servidor de controle da *botnet* e na prospecção de vulnerabilidades nos demais dispositivos da rede local. As capturas 10 e 11 possuem a comunicação com o servidor de controle e algumas rajadas de tráfego. Cada captura abrange as *botnets Rbot e Virut* e exigiu diferentes cenários de rede. Os experimentos conduzidos avaliaram a captura do tráfego e a detecção junto do contêiner que representava o roteador de borda da captura.

A Figura 3 mostra os quatro gráficos referentes à acurácia obtida pelas técnicas de aprendizagem de máquina no decorrer da reprodução do tráfego. A Abordagem 1 apresentou uma acurácia média de 95% e 94% nas duas primeiras avaliações, cujo os resultados foram apresentados nas Figuras 3(a) e 3(b), e a Abordagem 2 de 92% e 83% respectivamente. Nos dois últimos cenários apresentados nas Figuras 3(c) e 3(d), a Abordagem 1 atingiu uma acurácia média de 72% e 90% e a Abordagem 2 atingiu 88% e 85%, respectivamente. Ambas as abordagens empregam características do tráfego para realizar as análises, portanto, são sensíveis à fidelidade do tráfego. Os resultados foram calculados utilizando a informação verdadeira contida na documentação da captura de rede [Garcia et al. 2014]. Então, as altas taxas de acurácia atingidas pelas abordagens confirmam que o caso de uso apresentado neste trabalho atingiu os níveis de fidelidade necessários na reprodução dos tráfegos de rede.



(a) Captura 4 (b) Captura 5

(c) Captura 10 (d) Captura 11

Figura 3. Resultados do Estudo de Caso

5. Conclusão

Este trabalho descreveu a arquitetura do MENTORED *Testbed* e um caso de uso. A arquitetura se baseia na plataforma *Kubernetes* do IDS-RNP e adiciona uma camada de software para simplificar de forma segura as rotinas de configuração que originalmente seriam desenvolvidas pelos usuários. Desta forma, o *testbed* se aproveita da modularidade oferecida pelo *Kubespray* e adiciona dispositivos físicos sem fio para colaborar com a fidelidade ao reproduzir redes IoT heterogêneas, com conexões fim-a-fim, com recursos computacionais e energéticos limitados. O caso de uso compreendeu criar com fidelidade cenários experimentais com base em uma captura de tráfego de rede e apoiou a avaliação de duas propostas de detecção de tráfego de *botnets*. Os experimentos conduzidos apontaram que o caso de uso criou e reproduziu fielmente as capturas de rede, pois as propostas foram capazes de detectar o tráfego de *botnets*.

A IDS-RNP supre parcialmente as demandas requisitadas por experimentos de cibersegurança na IoT, uma vez que este não foi projetado para experimentos cujo principal objetivo é violar as camadas de segurança para coletar evidências e analisar seus comportamentos. Ainda não há estudos que comprovam ou não sua capacidade para esse tipo

de utilização, respeitando os requisitos desejáveis para um *testbed*, como validade, fidelidade, reprodutibilidade, escalabilidade, segurança e centrado na perspectiva do usuário.

Um dos desafios que envolve a implementação do MENTORED *Testbed* está relacionado com o compartilhamento da infraestrutura com outros usuários do IDS-RNP e entre múltiplos experimentos simultâneos. O isolamento do tráfego de rede e a garantia de qualidade de serviço interferem nas avaliações dos experimentos conduzidos pelo *testbed*. O Kubernetes fornece ferramentas que permitem monitorar a segurança e a saúde dos elementos que são executados dentro da sua infraestrutura, bem como suas interfaces de rede. No entanto, no IDS-RNP esse serviço de monitoramento dos ativos é restrito à equipe da RNP, o que traz dificuldade para coleta dos resultados das experimentações por parte dos pesquisadores.

Como trabalhos futuros, pretende-se conduzir novas avaliações experimentais do MENTORED *testbed* para analisar o cumprimento dos requisitos especificados neste artigo e para planejar e implementar as melhorias para atingi-los. Nesta direção, o projeto prevê a inclusão de dispositivos de IoT nas ilhas do *testbed* para melhor atender aos requisitos de fidelidade e escalabilidade. Com o intuito de disponibilizar o *testbed* para a comunidade acadêmica[3], planeja-se desenvolver o Portal do MENTORED *testbed* e a *Clearinghouse* para que estejam alinhados com os experimentos específicos de segurança cibernética e a IDS-RNP, e ainda elaborar recursos pré-configurados de apoio e outras facilidades para configuração e execução de experimentos.

6. Agradecimentos

Os autores agradecem à FAPESP – Fundação de Amparo à Pesquisa do Estado de São Paulo (Processos nº 2018/23098-0 e 20/05884-8) pelo apoio financeiro essencial para realização dessa pesquisa.

Referências

A10-Networks (2020). Attack shows it is imperative for a ddos zero-trust approach and continued diligence. `https://www.a10networks.com/blog/aws-hit-by-largest-reported-ddos-attack-of-2-3-tbps/`. Último Acesso: Março de 2021.

Adjih, C., Baccelli, E., Fleury, E., Harter, G., Mitton, N., Noel, T., Pissard-Gibollet, R., Saint-Marcel, F., Schreiner, G., Vandaele, J., and Watteyne, T. (2015). FIT IoT-LAB: A Large Scale Open Experimental IoT Testbed. In *IEEE World Forum on Internet of Things*, Milan, Italy.

Arora, A., Ertin, E., Ramnath, R., Nesterenko, M., and Leal, W. (2006). Kansei: a high-fidelity sensing testbed. *IEEE Internet Computing*, 10(2):35–47.

BBC (2020). Amazon 'thwarts largest ever ddos cyber-attack'. `https://www.bbc.com/news/technology-53093611`. Último Acesso: Fevereiro de 2021.

Benzel, T. (2011). The science of cyber security experimentation: The deter project. In *Proceedings of the 27th Annual Computer Security Applications Conference*, ACSAC '11, page 137–148, New York, NY, USA. Association for Computing Machinery.

[3]Maiores informações serão divulgadas em breve sobre a liberação da plataforma para a comunidade científica.

Garcia, S., Grill, M., Stiborek, J., and Zunino, A. (2014). An empirical comparison of botnet detection methods. *computers & security*, 45:100–123.

Gunduz, M. Z. and Das, R. (2018). A comparison of cyber-security oriented testbeds for iot-based smart grids. In *International Symposium on Digital Forensic and Security*, pages 1–6.

Gyrard, A. and Serrano, M. (2015). Fiesta-iot: Federated interoperable semantic internet of things (iot) testbeds and applications. In *ICT*.

Kolias, C., Kambourakis, G., Stavrou, A., and Voas, J. (2017). Ddos in the iot: Mirai and other botnets. *Computer*, 50(7):80–84.

Kumar, A. and Lim, T. J. (2018). A secure contained testbed for analyzing iot botnets. In *International Conference on Testbeds and Research Infrastructures*, pages 124–137. Springer.

Kumar, A. and Lim, T. J. (2019). A secure contained testbed for analyzing iot botnets. In Gao, H., Yin, Y., Yang, X., and Miao, H., editors, *Testbeds and Research Infrastructures for the Development of Networks and Communities*, pages 124–137, Cham. Springer International Publishing.

Muchtar, F., Abdullah, A. H., Abd Latiff, M. S., Hassan, S., Abd Wahab, M. H., and Abdul-Salaam, G. (2018). A technical review of manet testbed using mobile robot technology. In *Journal of Physics: Conference Series*, volume 1049, page 012001. IOP Publishing.

Siboni, S., Sachidananda, V., Meidan, Y., Bohadana, M., Mathov, Y., Bhairav, S., Shabtai, A., and Elovici, Y. (2019). Security testbed for internet-of-things devices. *IEEE Transactions on Reliability*, 68(1):23–44.

Vishwakarma, R. and Jain, A. K. (2020). A survey of ddos attacking techniques and defence mechanisms in the iot network. *Telecommunication systems*, 73(1):3–25.

Wired (2018). Github survived the biggest ddos attack ever recorded. `https://www.wired.com/story/github-ddos-memcached/`. Último Acesso: Março de 2021.