TeMIA-NT: Monitoramento e Análise Inteligente de Ameaças de Tráfego de Rede

Lucas C. B. Guimarães, Gabriel Antonio F. Rebello, Felipe S. Fernandes, Gustavo F. Camilo, Lucas Airam C. de Souza, Danyel C. dos Santos, Luiz Gustavo C. M. de Oliveira e Otto Carlos M. B. Duarte

Grupo de Teleinformática e Automação (GTA/PEE/COPPE)
Universidade Federal do Rio de Janeiro (UFRJ)

Resumo. *Ataques cibernéticos têm se tornado cada vez mais comuns e causam grandes danos a pessoas e organizações. A detecção tardia desses ataques aumenta a possibilidade de ocorrerem danos irreparáveis, com altas perdas financeiras sendo uma ocorrência comum. Este artigo propõe TeMIA-NT: Monitoramento e Análise Inteligente de Ameaças de Tráfego de Rede*[1]*, uma ferramenta para análise de tráfego em tempo real usando processamento paralelo de fluxos em um aglomerado. As principais contribuições da ferramenta TeMIA-NT são: i) a proposta de uma arquitetura modular para detecção em tempo real de intrusões de rede que suporta alta taxas de tráfego, ii) o uso da biblioteca structured streaming do Apache Spark e iii) dois modos de operação: em linha (online) e em tempo diferenciado (offline). O modo de operação em tempo diferenciado permite avaliar o desempenho de múltiplos algoritmos de aprendizado de máquina sobre um determinado conjunto de dados incluindo métricas como acurácia, F1-score e área sob a curva ROC. No modo em linha a ferramenta usa estruturas de dataframe e a biblioteca structured streaming no modo contínuo, o que permite a detecção de ameaças em tempo real e a rápida reação a ataques. De modo a minimizar os danos causados, TeMIA-NT atinge taxas de processamento de fluxo que chegam a 50 GB/s.*

1. Introdução

O cibercrime é um dos grandes desafios introduzidos pelo crescimento exponencial da Internet. Hoje, efeitos de ataques cibernéticos já geram perdas de até US$6 trilhões [Cybersecurity Ventures 2018], o que equivale a mais de três vezes o PIB brasileiro. Além disso, o crescimento e a popularização de áreas como Big Data e Internet das Coisas (*Internet of Things* - IoT) trazem desafios ainda maiores à segurança cibernética. A introdução de bilhões de dispositivos de baixa potência computacional conectados à rede aumenta o impacto de possíveis ataques, uma vez que estes dispositivos podem ser facilmente invadidos e comprometidos em larga escala [Bertino and Islam 2017, Azmoodeh et al. 2019, Symantec 2017]. O grande volume de dados a serem analisados também aumenta a complexidade da classificação do tráfego de rede e da detecção de ameaças [Rathore et al. 2016, Terzi et al. 2017]. Por fim, há ainda o tempo médio de detecção de um ataque como fator determinante para o impacto

Este trabalho foi realizado com recursos do CNPq, CAPES, FAPERJ e FAPESP (18/23292-0, 2015/24514-9, 2015/24485-9, 2014/50937-1).

[1]TeMIA-NT: *ThrEat Monitoring and Intelligent data Analytics of Network Traffic.*

das ameaças cibernéticas. O longo período de tempo necessário para detectar uma invasão, frequentemente levando de semanas até meses, aumenta exponencialmente o risco de perdas financeiras e o risco de danos irreparáveis [Verizon Enterprise 2018]. Isso se deve à necessidade de intervenção humana nessas situações, afetando significativamente a eficiência do tratamento de ameaças.

Nesse cenário no qual segurança é um aspecto fundamental, aumenta-se a necessidade de ferramentas capazes de garantir o uso seguro e confiável da rede. Soluções baseadas em ferramentas de gerenciamento de eventos e informações de segurança (*Security Information and Event Management* - SIEM) mitigam parcialmente o problema ao proverem um monitoramento em tempo real da rede. No entanto, este tipo de solução ainda é altamente dependente da intervenção do especialista e, por ser baseado em bancos de dados de assinaturas de ameaças, torna-se ineficiente para identificar novos ataques que surgirão. A utilização de algoritmos de aprendizado de máquina para a detecção de ameaças, por outro lado, automatiza o processo de detecção e atende à agilidade necessária no tratamento de ataques. Para isto, é essencial que sejam selecionados algoritmos que apresentem um bom desempenho no processo de classificação, sem que haja um impacto negativo na acurácia e em outras métricas de avaliação. Anteriormente, o GTA/UFRJ propôs CATRACA [Andreoni Lopez et al. 2019], uma ferramenta que utilizava aprendizado de máquina para detecção de ameaças em tempo real.

Este artigo propõe a ferramenta TeMIA-NT: Monitoramento e Análise Inteligente de Ameaças de Tráfego de Rede, um sistema inteligente de monitoramento e detecção de ameaças baseado em aprendizado de máquina e processamento distribuído em aglomerados. A ferramenta TeMIA-NT segue os mesmos objetivos da ferramenta CATRACA, porém com a parte de processamento distribuído completamente nova e melhorias significativas na parte de aprendizado de máquina. O foco desta nova ferramenta são a inteligência, a escalabilidade e o desempenho necessários para processar grandes volumes de dados, e a diversificação e otimização de algoritmos de aprendizado de máquina para atender à diversidade dos novos ataques. Visando o aumento de desempenho, o TeMIA-NT implementa o processamento distribuído integralmente em linguagem Scala, em vez de Python, e utiliza estruturas de dados de *dataframe*, em vez da estrutura padrão de *Resilient Distributed Datasets* (RDD) na plataforma de código aberto Apache Spark. As opções de algoritmos de aprendizado de máquina foram expandidas e há possibilidade de otimização de hiperparâmetros, permitindo testar, selecionar e ajustar os parâmetros do melhor algoritmo para cada tipo de cenário. A implementação da detecção de ameaças em tempo diferenciado utiliza a nova biblioteca *structuted streaming*, lançada em 2017, que permite o processamento de fluxos em lotes, com intervalos reduzidos e tolerância a falhas. A detecção de ameaças em linha se serve do modo de processamento contínuo (*continuous processing*) da biblioteca, que permite à ferramenta se comportar próxima a uma ferramenta de processamento nativo de fluxo.

O restante do artigo está organizado como segue. A Seção 2 apresenta trabalhos com temas relacionados ao artigo. A Seção 3 apresenta a arquitetura e as funcionalidades da ferramenta desenvolvida, assim como os resultados de desempenho obtidos. A Seção 4 apresentas as considerações finais, descreve a demonstração da ferramenta a ser feita no salão de ferramentas do SBRC, indica os manuais e o vídeo.

2. Trabalhos Relacionados

Novos desafios na área de sistemas de detecção de intrusão surgem devido ao grande volume de tráfego, grande número de dispositivos de IoT, ataques de negação de serviço distribuídos e ataques novos (*zero-day attack*) [Pelloso et al. 2018, Viegas et al. 2019, Campiolo et al. 2018]. Para atender alguns destes desafios, se popularizou o uso de técnicas de aprendizado de máquina para classificar fluxos em tempo real [Andreoni Lopez et al. 2019, Lobato et al. 2018]. As ferramentas para classificação de grandes volumes de dados a altas velocidades disponíveis baseiam-se em três plataformas de processamento distribuído principais: Apache Spark, Apache Storm e Apache Flink. A principal diferença entre as plataformas é que o Spark realiza processamento em lotes enquanto as plataformas Storm e Flink efetuam processamento nativo de fluxo.

O Open Security Operations Center (OpenSOC) [Cisco Systems 2014] é uma estrutura de segurança analítica para monitorar grandes massas de dados que foi descontinuado dando origem a um novo projeto, o Apache Metron. O Metron é uma ferramenta que compreende aquisição de diversos tipos de dados, processamento distribuído, enriquecimento, armazenamento e visualização dos resultados. O Metron permite a correlação de eventos de segurança de diferentes fontes, como *logs* de aplicativos e pacotes de rede. Para esse fim, a estrutura emprega fontes de dados distribuídas, como sensores na rede, *logs* de eventos de elementos de segurança e dados enriquecidos chamados fontes de telemetria. A estrutura também conta com uma base histórica de ameaças de rede da Cisco.

Baseadas na Plataforma Apache Spark têm-se as ferramentas Apache Spot, Stream4Flow [Jirsik et al. 2017] e Hogzilla. O Apache Spot é um projeto ainda em estágio de incubação que usa técnicas de telemetria e aprendizado de máquina para análise de pacotes para detectar ameaças. O protótipo Stream4Flow usa a pilha Elastic para a visualização dos parâmetros da rede, no entanto, carece de inteligência para realizar a detecção de anomalias. A ferramenta Hogzilla[2] possui suporte para Snort, SFlows, GrayLog, Apache Spark, HBase e libnDPI, fornecendo detecção de anomalias de rede. O Hogzilla também permite realizar a visualização do tráfego da rede, usando o Snort para captura de pacotes e obtendo características pela inspeção profunda de pacotes. Stream4Flow captura pacotes usando IPFIXcol e só considera informação dos cabeçalhos. Nesse trabalho utilizou-se o flowtbag que captura várias estatísticas do fluxo. Além disso, fez-se o processamento em modo diferenciado, afim de fazer uma adaptação contínua do modelo.

A ferramenta TeMIA-NT proposta também se baseia na plataforma Apache Spark, porque é a plataforma mais adotada, a que oferece mais possibilidades de algoritmos de aprendizado de máquina e a com a maior comunidade ativa. No entanto, a nosso conhecimento, TeMIA-NT é a única ferramenta disponível a utilizar a recente tecnologia de *structured streaming* em modos de lotes e contínuo no Apache Spark, a permitir a seleção dentre diversos algoritmos de aprendizado de máquina, e a operar em modos diferenciado e em linha.

3. Arquitetura, Funcionalidades e Avaliação de Desempenho da Ferramenta

A ferramenta proposta possui dois modos de operação: em linha (*online*) e em tempo diferenciado (*offline*). O modo em linha realiza a classificação em tempo real de

[2]http://ids-hogzilla.org/, acessado em abril de 2020.

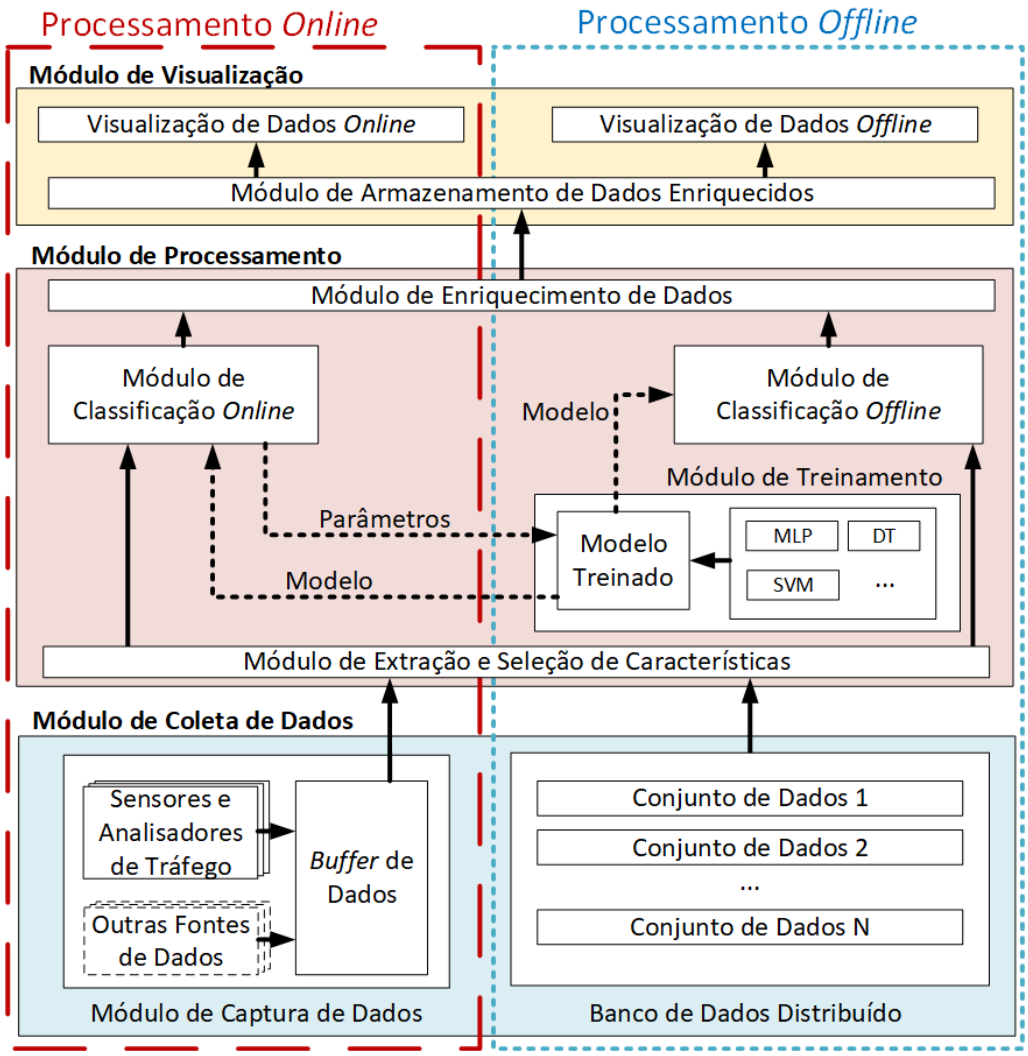

Figura 1. Arquitetura modular da TeMIA-NT nos modos em linha e em tempo diferenciado.

fluxos, enquanto o modo diferenciado permite observar o desempenho de múltiplos classificadores para determinado conjunto de dados, disponibilizando as métricas resultantes no módulo de visualização. A arquitetura proposta, apresentada na Figura 1, é modular e consiste em três módulos principais: coleta de dados, processamento e visualização.

O módulo de coleta de dados captura e abstrai em fluxos do tráfego de rede. Também armazena os conjuntos de dados usados no processamento em tempo diferenciado. O processo de captura espelha o tráfego de rede através da biblioteca libpcap. A seguir, a ferramenta flowtbag[3] abstrai a sequência de pacotes em fluxos e suas 45 características, incluindo a duração dos fluxos e a quantidade total de pacotes de cada fluxo. A pertinência de um pacote a um fluxo é definida pelos cinco campos do cabeçalho do pacote TCP/IP endereço IP de origem, endereço IP de destino, porta de origem, porta de destino e protocolo. Os fluxos são então enviados à um canal da plataforma Apache Kafka, que atua como um *buffer* de dados. Um banco de dados distribuído implemen-

[3]https://github.com/DanielArndt/flowtbag

tado no *Hadoop Distributed File System* (HDFS) armazena os conjuntos de dados que são utilizados na obtenção de modelos de classificação.

O módulo de processamento trata do processo de classificação desses fluxos. O módulo de processamento é implementado em um aglomerado Apache Spark. Esta plataforma apresenta vantagens ao desenvolvimento da ferramenta, pois possui bibliotecas voltadas à implementação de algoritmos de aprendizado de máquina e ao rápido processamento de dados em tempo real, utilizando o método de micro-lotes com o mecanismo de *structured streaming*. O módulo de treinamento extrai o modelo de classificação utilizando um conjunto de dados rotulado do HDFS. No modo de operação em linha os pacotes são coletados e adicionados a um canal do Apache Kafka e os fluxos são então classificados como legítimos ou maliciosos pelo modelo de classificação obtido anteriormente. Para permitir que os fluxos sejam analisados e classificados em tempo real no modo de processamento em linha, foi desenvolvido um módulo de classificação responsável por coletar estes dados conforme eles são adicionados a um determinado canal do Apache Kafka; os fluxos são então classificados como legítimos ou maliciosos pelo modelo de classificação obtido anteriormente. Para a execução no modo diferenciado, o módulo de classificação *offline* executa testes sobre variados algoritmos e conjuntos de dados, obtendo métricas de desempenho para cada combinação. Os resultados da classificação, tanto em linha quanto em tempo diferenciado, são então enviados a um servidor Elasticsearch, utilizando a biblioteca para integração com o Apache Spark.

O módulo de visualização permite que o administrador de rede visualize o histórico de classificações e o estado atual da rede, assim como os resultados de algoritmos testados. O módulo de visualização é feito utilizando os softwares Elasticsearch[4] e Kibana[5], ambos desenvolvidos pela Elastic. O Elasticsearch é responsável por implementar um servidor de buscas distribuído e eficiente, baseado em documentos JSON. Ele recebe e armazena os dados conforme ele são enviados pelo módulo de processamento, após finalizado o processo de classificação. O Kibana é responsável por disponibilizar uma interface de usuário por meio de *dashboards*, exibindo ao administrador de rede os dados recebidos pelo Elasticsearch em tempo real para ambos os modos de execução. Ele também permite a consulta por dados históricos, empregando as funcionalidades de servidor de buscas do Elasticsearch.

A classificação para determinar o que constitui tráfego legítimo ou malicioso requer uma fase de treino para se obter o modelo de classificação; por sua vez, o treino requer um conjunto de dados rotulados. Nesse artigo o conjunto de dados utilizado foi obtido a partir do tráfego real de uma operadora de telecomunicações no Rio de Janeiro em fevereiro de 2017. Os pacotes foram abstraídos em fluxos através da ferramenta *flowtbag*, obtendo-se assim 40 novas características responsáveis por auxiliar os algoritmos no processo de obtenção de modelos de classificação. A rotulação dos fluxos como legítimos ou maliciosos foi feita através da ferramenta de detecção de intrusão Suricata.

No processo de obtenção dos modelos no modo de processamento diferenciado, os testes foram executados repartindo o conjunto de dados de modo que 70% fossem parte de um conjunto de treino e os outros 30% parte do conjunto de teste. Foi também utilizada validação cruzada *k-fold*, com $k = 10$, para garantir a capacidade de generalização

[4]https://github.com/elastic/elasticsearch

[5]https://github.com/elastic/kibana

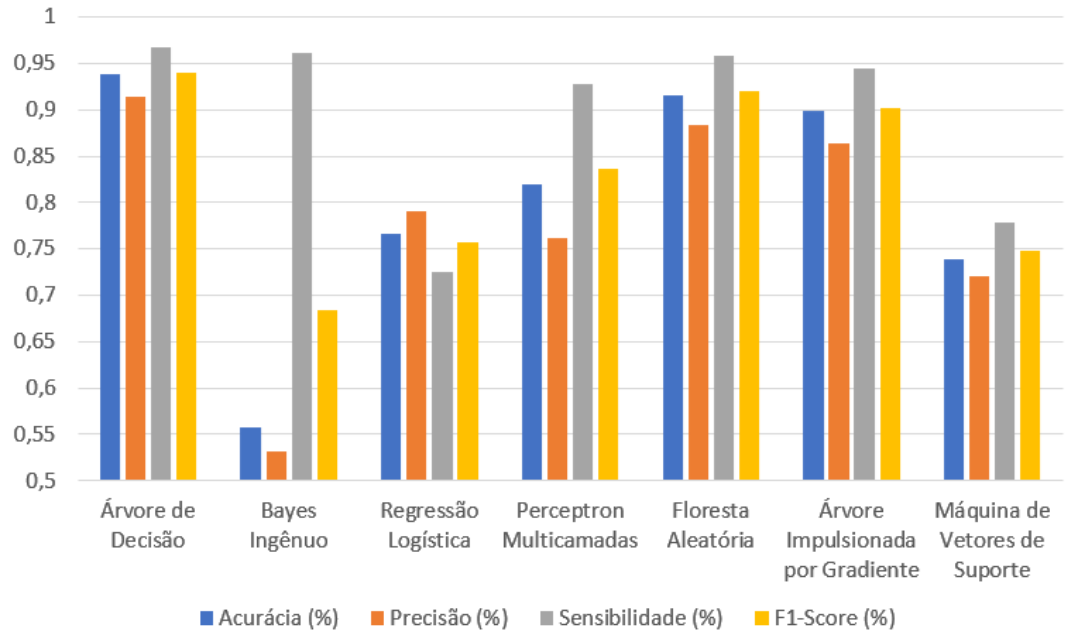

Figura 2. Comparação das métricas de avaliação para os sete classificadores.

do modelo. Por fim, foi utilizado o método de grade para realizar a sintonização dos hiperparâmetros em cada algoritmo. A Figura 2 mostra que os algoritmos: floresta aleatória, árvore de decisão e árvore impulsionada por gradiente apresentam melhor desempenho.

O modo de operação em linha se serve do modelo que apresenta a maior capacidade de processamento e uma boa acurácia. A Tabela 1 mostra que o algoritmo de árvore de decisão apresentou a taxa máxima de volume de fluxos de 50 GB/s. O modelo de classificação de floresta aleatória apresenta desempenho inferior por ter que processar múltiplas árvores, sendo necessário primeiramente obter o resultado para todas as árvores de modo a se obter o resultado final da classificação.

Como o modelo de arvore de decisão apresenta os melhores resultados tanto em acurácia quanto em capacidade de classificação, este é o modelo utilizado por padrão na execução da ferramenta. Entretanto, outros modelos também podem ser utilizados, de acordo com as necessidades do usuário. O tempo de convergência e treinamento do modelo além da a velocidade de processamento devem ser considerados no contexto de análise em tempo real. Os dados mostrados até então supõem uma modelagem prévia dos classificadores. A ferramenta TeMIA-NT usa a estrutura de dados *dataframe* que

Tabela 1. Eficiência de processamento dos modelos com os melhores resultados quanto a número e volume de fluxos classificados por segundo.

	Fluxos/s	GB/s
Floresta Aleatória	586.563,32	21,95
Árvore de Decisão	1.330.732,59	49,80
Árvore Impulsionada por Gradiente	1.206.962,94	45,17

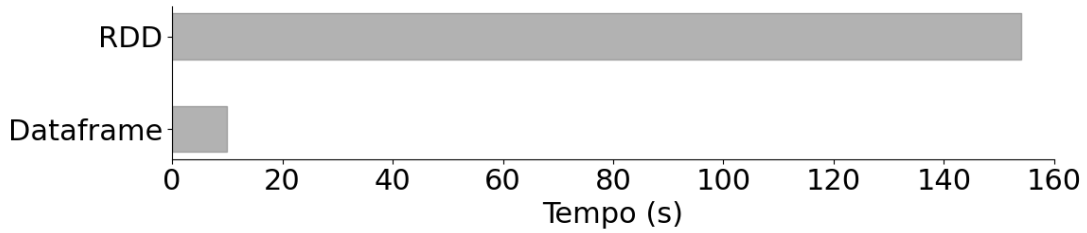

Figura 3. Impacto da estrutura de dados no tempo de treino da árvore de decisão.

permite uma série de otimizações quanto ao tempo de processamento e uso de memória. Essa escolha da estrutura de dados para armazenar os fluxos para ajuste do modelo é fundamental no desempenho do treino como mostra a Figura 3.

4. Considerações Finais, Demonstração e Documentação da Ferramenta

Este artigo apresenta a ferramenta TeMIA-NT, desenvolvida de modo a permitir a análise de tráfego utilizando processamento paralelo de fluxos. A ferramenta possui dois modos de operação: em linha e tempo diferenciado. O modo de operação em linha permite monitorar e detectar ameaças de segurança de rede em tempo real. O modo de operação em tempo diferenciado permite a avaliação do desempenho de múltiplos modelos de classificação obtidos a partir de diferentes algoritmos e conjuntos de dados. TeMIA-NT também permite a seleção dentre sete algoritmos de aprendizado de máquina na obtenção de modelos de classificação. A detecção de ameaças em tempo real com baixas latências é alcançada graças a estrutura de dados *dataframe* usada e ao modo *continuous processing* da biblioteca *structured streaming*.

Os resultados obtidos de um conjunto de dados baseado em tráfego legítimo demonstram a alta capacidade de processamento da ferramenta em fluxos por segundo. Também observa-se o desempenho de cada algoritmo de aprendizado de máquina implementado, com os modelos de árvore de decisão e floresta aleatória apresentando altos valores em métricas como acurácia e *f1-score*.

A versão da ferramenta que será usada para demonstração roda em uma máquina virtual para facilitar que seja baixada e instalada [6]. Será necessário um computador pessoal com processador core i7 de oitava geração ou maior, 16 GB de RAM e HD com mais de 100 GB disponível. Um monitor maior que 32 polegadas é necessário. A demonstração do modo em linha analisa e classifica em tempo real o tráfego de rede. Os resultados da classificação, incluindo as ameaças detectadas, o momento da detecção e sua localização geográfica, serão exibidos pela ferramenta. O modo em tempo diferenciado, no qual alguns classificadores são executados sobre um conjunto de dados de teste, também terá seus resultados exibidos pelo módulo de visualização. Ambas as etapas serão executadas no computador do salão de ferramentas, no qual o conjunto de dados da operadora de telecomunicações é utilizado para gerar tráfego para a ferramenta. Posteriormente, a ferramenta será utilizada para detectar ataques de rede em tempo real, que serão enviados por Wi-Fi por uma máquina com sistema operacional Kali Linux.

[6]Caso acesso de boa qualidade a Internet esteja disponível, uma demonstração da ferramenta será feita utilizando o aglomerado do Grupo de Teleinformática e Automação no Rio de Janeiro.

A ferramenta, assim como sua documentação, licença e código, estão disponíveis em: https://www.gta.ufrj.br/TeMIA-NT/. Dentre as informações disponíveis constam manuais detalhando os processos de instalação e execução da ferramenta, assim como um vídeo introduzindo a ferramenta e suas funcionalidades.

Referências

Andreoni Lopez, M., Mattos, D. M. F., Duarte, O. C. M. B., and Pujolle, G. (2019). Toward a monitoring and threat detection system based on stream processing as a virtual network function for big data. *Concurrency and Computation: Practice and Experience*, 31(20):e5344. e5344 cpe.5344.

Azmoodeh, A., Dehghantanha, A., and Choo, K.-K. R. (2019). Big data and internet of things security and forensics: Challenges and opportunities. In *Handbook of Big Data and IoT Security*, pages 1–4. Springer.

Bertino, E. and Islam, N. (2017). Botnets and internet of things security. *Computer*, 50(2):76–79.

Campiolo, R., dos Santos, L. A. F., Monteverde, W. A., Suca, E. G., and Batista, D. M. (2018). Uma arquitetura para detecção de ameaças cibernéticas baseada na análise de grandes volumes de dados. In *WSCDC 2018*, Porto Alegre, RS, Brasil. SBC.

Cisco Systems (2014). OpenSOC: The Open Security Operations Center. *https://opensoc.github.io/*. Acessado em 11/03/2020.

Cybersecurity Ventures (2018). Cybersecurity Market Report. *https://cybersecurityventures.com/*. Acessado em 11/03/2020.

Jirsik, T., Cermak, M., Tovarnak, D., and Celeda, P. (2017). Toward Stream-Based IP Flow Analysis. *IEEE Communications Magazine*, 55(7):70–76.

Lobato, A. G. P., Lopez, M. A., Sanz, I. J., Cardenas, A. A., Duarte, O. C. M. B., and Pujolle, G. (2018). An adaptive real-time architecture for zero-day threat detection. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6.

Pelloso, M., Vergutz, A., Santos, A., and Nogueira, M. (2018). A self-adaptable system for DDoS attack prediction based on the metastability theory. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6.

Rathore, M. M., Ahmad, A., and Paul, A. (2016). Real time intrusion detection system for ultra-high-speed big data environments. *Journal of Supercomputing*, 72(9):3489–3510.

Symantec (2017). Internet Security Threat Report. *https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf*. Acessado em 11/03/2020.

Terzi, D. S., Terzi, R., and Sagiroglu, S. (2017). Big data analytics for network anomaly detection from netflow data. In *UBMK 2017*, pages 592–597. IEEE.

Verizon Enterprise (2018). Data Breach Investigations Report. *https://enterprise.verizon.com/resources/reports/dbir/*. Acessado em 11/03/2020.

Viegas, E., Santin, A., Bessani, A., and Neves, N. (2019). Bigflow: Real-time and reliable anomaly-based intrusion detection for high-speed networks. *Future Generation Computer Systems*, 93:473–485.