Um Sistema Adaptativo de Detecção e Reação a Ameaças*

Antonio Gonzalez Pastana Lobato[1], Martin Andreoni Lopez[1,2], Gabriel Antonio F. Rebello[1], e Otto Carlos Muniz Bandeira Duarte[1]

[1]GTA / PEE-COPPE / UFRJ – Brasil
[2]LIP6/CNRS (UPMC Sorbonne Universités) – França

Abstract. *Attackers create new threats and constantly change their behavior to mislead current security systems. Moreover, threats are detected in days or weeks, while a countermeasure must be immediately triggered to avoid or reduce any damages. This paper proposes an adaptive threat detection system with a SDN based schema to perform countermeasures. The contributions of this work are the following: i) threat detection and prevention analyzing only five packets of each flow; ii) adaptive detection algorithms with real time training; iii) instant countermeasure trigger, without waiting the end of the flow; iv) effective threat block, even in scenarios with spoofed IP addresses. A SDN schema effectively monitors the five packets sequence and blocks the threat in its source, avoiding the waste of resources. The results show a high accuracy in threat detection, even with network behavior varying over time.*

Resumo. *Atacantes criam novas ameaças e constantemente mudam seu comportamento para enganar os sistemas de segurança atuais. Aliás, as ameaças são detectadas em dias ou semanas, enquanto uma contramedida deve ser imediatamente efetuada para evitar ou reduzir prejuízos. Este artigo propõe um sistema adaptativo de detecção de ameaças que possui um esquema baseado em Redes Definidas por Software (SDN) para realizar contramedidas. As contribuições do trabalho são: i) a detecção e prevenção de ameaças analisando uma sequência de apenas cinco pacotes de cada fluxo; ii) o desenvolvimento de algoritmos de detecção treinados em tempo real, com comportamento adaptativo; iii) o imediato acionamento de contramedidas sem esperar o fim do fluxo; e iv) o efetivo bloqueio de ameaças mesmo em cenários nos quais o endereço do pacote IP é mascarado. Um esquema baseado na tecnologia SDN efetua o monitoramento da sequência de cinco pacotes e o rápido bloqueio do ataque ainda na origem, evitando que recursos de rede sejam desperdiçados. Os resultados obtidos mostram uma alta acurácia na detecção de ameaças, mesmo com o comportamento da rede variando com o tempo.*

1. Introdução

A grande quantidade de dados transferidos pelos sistemas de comunicação atuais criam um cenário desafiador para a detecção de intrusão [Suthaharan 2014]. A prevenção consiste em uma ação que neutraliza ou mitiga o ataque e impede que maiores prejuízos aconteçam. Os atuais sistemas de segurança, como o *Security Information and Event Management* (SIEM), não possuem um desempenho satisfatório, pois cerca de 85% das intrusões de redes são detectadas semanas depois de terem ocorrido [Ponemon e IBM 2017]. É essencial que o tempo de detecção seja o mínimo possível para que a prevenção da intrusão possa ser efetiva [Wu et al. 2014].

*Este trabalho foi realizado com recursos do INCT INTERNET DO FUTURO, do CNPQ, da CAPES, e da FAPERJ

Além disso, os ataques de segurança vêm se aperfeiçoando e a simples análise e filtragem de pacotes pelo cabeçalho IP e porta TCP deixaram de ser efetivas. O tráfego atacante procura se esconder das ferramentas de segurança falsificando o IP de origem e alterando dinamicamente a porta TCP. Nesse contexto, uma alternativa promissora para classificar tráfego e detectar ameaças, que vem apresentando bons resultados, se baseia no uso de técnicas de aprendizado de máquina.

A prevenção de intrusão deve conseguir bloquear inteiramente as ameaças evitando assim que o tráfego malicioso afete o desempenho da rede. Uma abordagem para realizar esse bloqueio de tráfego é colocar o sistema de prevenção de intrusão diretamente na rota dos pacotes que ele analisa, de maneira que ele possa atuar sobre o tráfego malicioso. Contudo, essa forma apresenta problemas como: a introdução de latência, já que o sistema de detecção analisa o tráfego antes de encaminhar para o destino; e a baixa acurácia, visto que o sistema posicionado dessa maneira não tem acesso aos demais tráfegos da rede e pode classificar errado um ataque sem essas informações. Este trabalho usa uma abordagem alternativa, que não apresenta esses problemas, a detecção fora da rota, na qual os pacotes são espelhados para um elemento sensor. Uma forma de implementar esquemas de prevenção de intrusão fora da rota é através da tecnologia de redes definidas por *software* (*Software Defined Network* - SDN) [McKeown et al. 2008], que fornece um controle logicamente centralizado do tráfego de rede.

Este artigo propõe métodos adaptativos de detecção de ameaças e um esquema para a extração das características dos fluxos com o imediato bloqueio do tráfego malicioso. A abordagem de caracterização dos fluxos é baseada na análise periódica de um reduzido número de pacotes em sequência. O esquema provê a detecção automática de ameaças baseada na análise dessas características por algoritmos adaptativos de aprendizado de máquina. A proposta apresenta as seguintes vantagens: robusta para cenários de ataques por inundação, já que apenas os primeiros pacotes são analisados por período; precisa na identificação do tráfego malicioso, pois possui uma acurácia elevada na classificação de ataques e consegue se adaptar a novas ameaças; e eficaz no bloqueio dos ataques, pois com o uso das redes definidas por *software* é possível bloquear o tráfego atacante na sua origem, mesmo em cenários onde os endereços IP são mascarados.

O restante do artigo está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados. A Seção 3 apresenta a elaboração do conjunto de dados para avaliar o esquema de detecção apenas com os primeiros pacotes de cada fluxo. Os métodos de detecção adaptativos propostos são apresentados na Seção 4, a Seção 5 detalha o esquema para o desvio de tráfego e bloqueio de ameaças usando redes definidas por *software*. Por fim, a Seção 6 conclui o trabalho.

2. Trabalhos Relacionados

As técnicas de aprendizado de máquina podem ser supervisionadas ou não supervisionadas, dependendo se o conjunto de dados está rotulado ou não. Entre as técnicas de classificação mais conhecidas estão as redes neurais, as árvores de decisão e as máquinas de vetores de suporte (*Support Vector Machines* - SVM) [Buczak e Guven 2015]. Li *et al.* combinam uma técnica de correspondência de padrões, *Dynamic Time Warping* - DTW) e SVM para gerar regras de detecção de intrusão [Ji et al. 2016]. O método é avaliado através do conjunto de dados KDD tradicional para classificar os ataques de negação de serviço (*Denial of Service* - DoS), varredura de porta (*Probe*) e acesso remoto para Local (*Remote to Local* - R2L). No entanto, seu método só pode lidar com ataques conhecidos. Na análise não supervisionada, não há informações sobre a classe a que cada amostra pertence. A detecção de padrões aplica esse tipo de análise. Lakhina *et al.* propõem o uso de entropia de amostra para detecção de anomalia e os

autores mostram que esta métrica combinada para IPs e portas de origem e destino, juntamente com análise de volume pode detectar múltiplas fontes de anomalias [Lakhina et al. 2005].

Braga *et al.* propõem um método de detecção de ataques de negação de serviço distribuído (*Distributed Denial of Service* - DDoS) usando um controlador NOX-OpenFlow [Braga et al. 2010]. A proposta extrai as características dos fluxos das tabelas dos comutadores da rede tais como média de pacotes por fluxo, média de *bytes* por fluxo, entre outras. Logo, esses atributos são enviados a um classificador, baseado em um mapa auto organizável dentro do controlador, que determinam se o tráfego é normal ou se corresponde a um ataque. O processamento é feito pelo controlador da rede que é sobrecarregado pelo processo de detecção utilizado. A detecção e a mitigação de ataques de negação de serviço em ambientes SDN é proposto em [Andreoni Lopez et al. 2016]. Na mitigação proposta, o mecanismo de detecção envia uma mensagem segura ao controlador SDN com as informações do fluxo a ser bloqueado. O esquema não detalha como é feita a escolha dos comutadores que recebem as configurações de bloqueio, podendo ser muito drásticas e acabar bloqueando também fluxos lícitos.

Bernaille *et al.* [Bernaille et al. 2006] propõem o uso dos primeiros pacotes de cada fluxo para determinar a aplicação do tráfego analisado. As aplicações possuem um comportamento bem definido no início, por isso, com a análise dos primeiros pacotes é possível classificá-las. A proposta de Bernaille *et al.* para classificar aplicações é fortemente baseada na característica de tamanho de pacote, e os autores esclarecem que torna a proposta vulnerável a atacantes que podem ajustar esse parâmetro para enganar o sistema. Peng *et. al.* também utilizam os *n* primeiros pacotes aplicados a onze algoritmos supervisados de aprendizado de máquina para a identificação de tráfego na Internet [Peng et al. 2015]. Estes trabalhos comprovam a importância do parâmetro tamanho do pacote na acurácia da classificação das aplicações.

Uma plataforma de código aberto para a classificação de tráfego é proposta por Donato *et. al.*. A plataforma denominada *Traffic Identification Engine* - (TIE) [Donato et al. 2014], utiliza seis algoritmos de aprendizado de máquina que obtêm bom desempenho. Na proposta deste artigo a característica de tamanho do pacote é acrescentada a outras 25 características para detectar ameaças. Espera-se que o mesmo sucesso obtido na acurácia da caracterização de aplicações seja atingido na detecção de ameaças. As outras características ajudam a evitar que os atacantes dissimulem o ataque aumentando apenas o tamanho do pacote e praticamente impossibilita ao atacante ajustar de maneira efetiva todos esses parâmetros para se passar por tráfego legítimo.

3. Conjuntos de Dados de Avaliação

Poucos conjuntos de dados estão disponíveis para avaliar mecanismos de defesa e o principal motivo para não fornecer dados de segurança é a privacidade, uma vez que os dados de tráfego podem conter informações confidenciais. Os dois conjuntos de dados mais conhecidos são o DARPA [Lippmann et al. 2000] e o KDD 99 [Lee et al. 1999]. Tráfego TCP/IP, dados de sistema operacional e dados de ameaças coletados de uma rede de computadores simulada compõem o conjunto de dados DARPA. Já o KDD foi elaborado a partir da extração de características dos logs do DARPA. No entanto, esses conjuntos de dados apresentam diversas limitações [Sharafaldin et al. 2017]. Uma delas é que o tráfego não corresponde a um cenário de rede real, já que foi simulado. Outra questão é que esses conjuntos de dados têm mais de 17 anos e não representam ameaças atuais [Sommer e Paxson 2010].

Visando um conjunto de dados com tráfego real mais recente, neste trabalho são avaliados métodos de detecção adaptativo através de dois conjuntos de dados. Um dos conjuntos

de dados é composto por um tráfego normal de uma das principais operadoras de rede no Brasil combinado com ataques de *botnets*, e o outro conjunto de dados foi criado no laboratório GTA/UFRJ. Para ambos os conjuntos de dados, os pacotes são combinados em fluxos. Um fluxo é definido como uma sequência de pacotes do mesmo endereço de IP origem para o mesmo endereço de IP destino durante uma janela de tempo. A cada período de tempo, apenas os cinco primeiros pacotes de um fluxo são capturados para a extração de características. O valor de cinco pacotes foi escolhido através da medida de acurácia de três algoritmos de classificação: árvore de decisão, SVM e redes neurais. Esse valor foi a menor quantidade de pacotes que apresentou acurácia superior a 90%. Cada fluxo tem 26 características, gerados pelos dados de cabeçalho TCP/IP. As principais características são: taxa de pacotes TCP, UDP e ICMP; número de portas de origem e de destino; número de cada *flag* TCP; média e variância do tempo de chegada entre pacotes; média e variância do comprimento do pacote de fluxo; entre outros.

O conjunto de dados do operador de rede, doravante denominado NetOp, é composto por informações de acesso real de 373 usuários residenciais de banda larga da cidade do Rio de Janeiro por um período de uma semana. O tráfego de rede é anonimizado por questões de privacidade. Como a operadora filtrou os dados usando um sistema de detecção de intrusão (IDS) e o tráfego depois da filtragem é assumido como tráfego benigno. Foram analisados a quantidade de registros desse IDS e os ataques filtrados correspondem a 15% do tráfego total. Para avaliar os algoritmos de detecção de ameaças foi adicionado o tráfego malicioso real capturado no trabalho de García *et. al* [Garcia et al. 2014]. Os dados de ataque são de uma *botnet* e têm 13 cenários diferentes de infecção por *malware*. Foi então inserido no conjunto de dados o tráfego malicioso de *botnet* correspondendo a 15% de proporção de tráfego de ameaças.

Esse trabalho utiliza um segundo conjunto de dados criado no laboratório com tráfego de rede real para avaliar a proposta. O conjunto de dados foi elaborado através da captura de tráfego contendo tanto o tráfego normal quanto as ameaças de rede reais. O conjunto de dados possui diferentes tipos de ataques. Ao todo, o conjunto de dados contém sete tipos de negação de serviço (*Denial of Service* - DoS) e nove tipos de varredura de portas. Os ataques de DoS são *ICMP flood*, *land*, *nestea*, *smurf*, *SYN flood*, *teardrop*, e *UDP flood*. Os diferentes tipos de varredura no conjunto de dados são *TCP SYN scan*, *TCP connect scan*, *SCTP INIT scan*, *Null scan*, *FIN scan*, *Xmas scan*, *TCP ACK scan*, *TCP Window scan*, e *TCP Maimon scan*. As ferramentas da distribuição *Kali Linux*, que visa testar a segurança de redes de computadores, foram utilizadas para a criação dos ataques. Esses ataques foram rotulados com base no destino, já que os ataques foram enviados para *honeypots*. O conjunto de dados contém cerca de 95 GB de dados de captura de pacotes.

4. Métodos Adaptativos Propostos de Detecção de Ameaças

Os atacantes criam novos ataques e alteram os ataques convencionais de forma a ludibriarem e passarem despercebidos por sistemas de detecção de intrusão baseados em assinatura ou métodos de classificação *off-line*. Para aumentar a robustez da detecção de ameaças, este trabalho propõe um novo esquema adaptativo de coleta de dados capaz de aprender ataques novos e detectar ameaças que se modificam com o tempo. Este esquema de coleta de dados proposto garante adaptabilidade tanto para o comportamento legítimo como para o malicioso. Os métodos se adaptam a mudanças aceitáveis no uso da rede e aprendem novos ataques que são realizados nos *honeypots*. Assim, são propostos três métodos que adaptam seus parâmetros em tempo real à medida que os fluxos de dados chegam. Os dois primeiros algoritmos de detecção são algoritmos de classificação *on-line* que visam classificar os ataques com base na captura de comportamento de ataque conhecida pelo *honeypots*, enquanto o outro é um algoritmo de

detecção de anomalia. Os três algoritmos implementados possuem a capacidade de detectar novos ataques.

Os métodos de classificação *on-line* requerem dados rotulados. Assim, no método de detecção proposto, todos os dados provenientes de *honeypots* são rotulados como ameaças, uma vez que não há nenhum serviço real sendo oferecido nos *honeypots* e todo o acesso é destinado a explorar uma vulnerabilidade intencionalmente instalada. Portanto, todos os fluxos que chegam nos *honeypots* serão usados pelos algoritmos para atualizar os parâmetros. Esta realimentação de dados em tempo real garante comportamento adaptativo para a detecção de ameaças. Então, sempre que um atacante executa um novo ataque ou muda seu comportamento, os modelos de classificação são atualizados. Os métodos de classificação propostos assumem como tráfego benigno os fluxos recebidos nos sensores de tráfego na rede durante o tempo de configuração, que é um período em que o uso da rede é monitorado para garantir que todos os fluxos sejam legítimos. Após esse tempo, caso um fluxo proveniente dos sensores de rede for classificado como ameaça, os parâmetros no algoritmo não são atualizados. Se o fluxo for considerado normal os parâmetros do algoritmo são atualizadosadaptando-se às mudanças de comportamento normal da rede.

O Algoritmo Gradiente Estocástico Descendente com Momento é uma aproximação do algoritmo do gradiente estocástico descendente (*Stochastic Gradient Descent* - SGD), na qual o gradiente é aproximado por uma única amostra. Na aplicação de detecção de ameaças deste artigo são consideradas duas classes: normal e ameaça. Portanto, a função Sigmoid 1

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^\intercal x}} \tag{1}$$

é utilizada para executar a regressão logística. Na função Sigmoid, baixos valores de produto dos parâmetros $\theta^\intercal$ vezes o vetor da característica da amostra x retornam zero, enquanto valores altos retornam um. Quando uma nova amostra $x_{(i)}$ chega, o método avalia a função Sigmoid e retorna um para $h_\theta(x_{(i)})$ maiores que 0.5 e zero caso contrário. Esta decisão apresenta um custo associado, com base na classe real da amostra $y_{(i)}$. A função de custo é definida na Equação 2. Esta função é convexa e o objetivo do algoritmo SGD é encontrar o seu mínimo, expresso por

$$J_{(i)}(\theta) = y_{(i)} log(h_\theta(x_{(i)})) + (1 - y_{(i)}) log(1 - h_\theta(x_{(i)})). \tag{2}$$

Quando uma nova amostra chega, o algoritmo dá um passo em direção ao mínimo custo com base no gradiente desta função.

O Algoritmo 1 mostra a implementação do algoritmo do SGD. A cada amostra recebida do vetor $x_{(i)}$ o método determina a classe $y_{(i)}$ baseado no tipo da origem. Se a amostra vier de um *honeypot* a etiqueta é 1, enquanto se vier de uma sonda de trafego a etiqueta é 0. Se a amostra vem de um analisador de tráfego e o SGD prevê como uma ameaça, o algoritmo envia um alerta. Caso contrário, atualiza os parâmetros com base no gradiente da função custo. O termo $\Delta\theta$[1] é o momento e tem o valor da atualização anterior do parâmetro. No contexto do SGD, este termo considera o movimento passado ao atualizar os parâmetros θ. Os parâmetros α e β são periodicamente atualizados de forma *offline* com base no custo histórico de cada amostra.

A Figura 1 mostra o comportamento de acurácia ao longo do tempo para cada amostra

[1]Na física clássica, o momento estimado indica a dificuldade de mudar o movimento de um objeto em movimento circular.

Entrada: Características de fluxo de entrada x, Classe y
Saida : Classe Prevista $predict$, Parâmetros do treinamento θ

Inicializar θ, $\Delta\theta$, α, β;
para $i \leftarrow 1$ **até** m **faça**
  $h_\theta(x_{(i)}) = \frac{1}{1+e^{-\theta^\intercal x_{(i)}}}$;
  $predict = round(h_\theta(x_{(i)}))$;
  se $predict == 1$ *and* $y_{(i)} == 0$ **então**
    Envia Alerta;
  senão
    $\theta = \theta - \alpha\nabla J_{(i)}(\theta) + \beta\Delta\theta$;
    $\Delta\theta = \alpha\nabla J_{(i)}(\theta)$;
  fim
fim

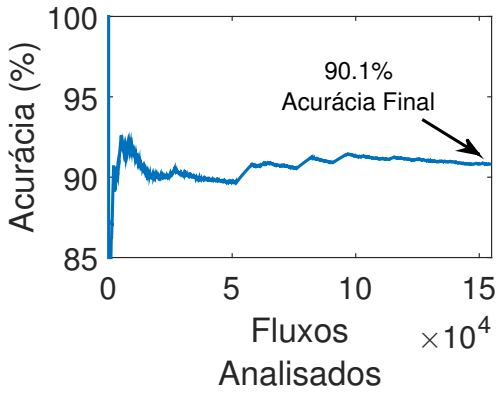
Algoritmo 1: Gradiente Estocástico Descendente com Momento.

Tabela 1: Matriz de Confusão SGD para o conjunto de dados do Laboratório.

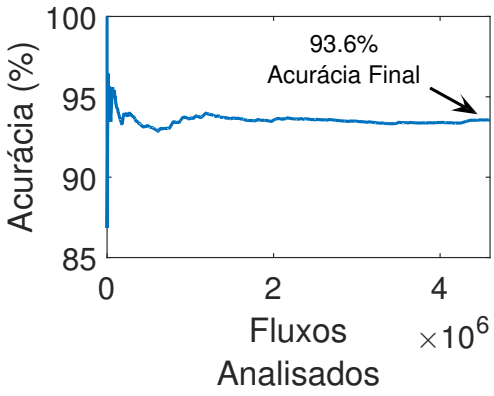
	Normal	**Ameaça**
Normal	104028	2927
Ameaça	11245	35987

Tabela 2: Matriz de Confusão SGD para o conjunto de dados NetOp.

	Normal	**Ameaça**
Normal	4172989	21431
Ameaça	287441	341722



(a) Conjunto de dados do Laboratório.



(b) Conjunto de dados do Operador.

Figura 1: Acurácia do Gradiente Estocástico Descendente. Em ambos os conjuntos de dados, a acurácia permanece estável mesmo com novos ataques e mudanças de comportamento de uso legítimo.

recebida, e as Tabelas 1 e 2 mostram a matriz de confusão final para ambos conjuntos de dados. No início da análise, o *o*verfit ocorre porque o algoritmo tem poucas amostras e se adapta especificamente a elas, resultando em uma precisão muito alta. No entanto, à medida que as amostras chegam, o SGD estabiliza e adquire uma ótima capacidade para generalizar e adaptar a novas amostras de tráfego, terminando com precisão de 90,1% para o conjunto de dados do laboratório e 93,6% para o conjunto de dados NetOp. Apesar das taxas de detecção de ameaças de 76,2% e 54,3%, respectivamente, esse algoritmo possui taxas falso positivas muito baixas de 2,7% e 0,5%. A baixa taxa de falsos positivos é uma característica desejada e um resultado importante da proposta, pois assegura confiança nos alertas de segurança da plataforma.

Entrada: Características de fluxo de entrada x, Classe y
Saída : Classe prevista $predict$, parâmetros do treinamento θ

Inicializar θ, α, λ;
para $i \leftarrow 1$ **até** m **faça**
  $predict = sign(\theta^{\intercal} x_{(i)})$;
  se $predict == 1$ *and* $y_{(i)} == -1$ **então**
    Envia Alerta;
  senão
    se $y_{(i)} \theta^{\intercal} x_{(i)} > 1$ **então**
      $\nabla_{(i)} = \theta$;
    senão
      $\nabla_{(i)} = -\lambda y_{(i)} x_{(i)}$;
    fim
    $\theta = \theta - \alpha \nabla_{(i)}$
  fim
fim

Algoritmo 2: Máquina de Vetores de Suporte *Online*.

Tabela 3: Matriz de Confusão SVM para o conjunto de dados do Laboratório.

	Normal	**Ameaça**
Normal	100658	6297
Ameaça	5900	41332

Tabela 4: Matriz de Confusão SVM para o conjunto de dados NetOp.

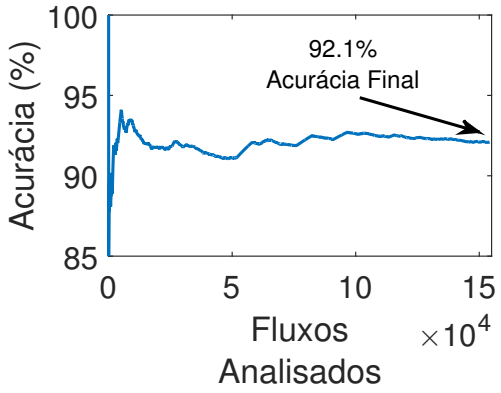
	Normal	**Ameaça**
Normal	4099766	94654
Ameaça	117051	512112

A máquina de vetores de suporte (*Support Vector Machine* - SVM) *online* é um classificador binário com base no conceito de plano de decisão que define os limites das classes. Um hiperplano construído em um espaço multidimensional divide os dados. O algoritmo SVM *online* usa uma aproximação de margem suave com a função convexa de perda de articulação (*hinge loss*), dada por

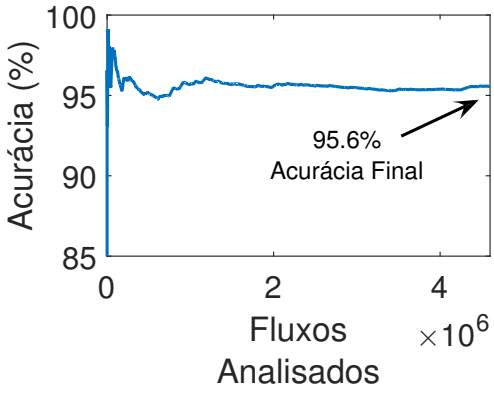
$$\max\{0, 1 - y\theta^{\intercal} x\}. \tag{3}$$

O objetivo deste algoritmo é minimizar a função de perda. Assim como o algoritmo anterior, o método determina a classe $y_{(i)}$ com base na origem. Se vier de um *honeypot*, o rótulo é 1, enquanto que se ele vem de um sensor de rede, o rótulo é -1. Novamente, quando um tráfego de uma sonda é classificado como ameaça, o método envia um alerta e não atualiza o modelo. O Algoritmo 2 mostra a implementação do SVM *online*. Os parâmetros α, λ são periodicamente atualizados com base na avaliação da função de perda sobre as amostras históricas.

A Figura 2 mostra as mudanças da acurácia ao longo do tempo em cada amostra recebida, e as Tabelas 3 e 4 mostram a matriz de confusão final para ambos conjuntos de dados. A acurácia final é melhor do que o algoritmo SGD, porque o SVM é um classificador robusto que maximiza a margem para os limites de decisão do hiperplano. A Figura 2 também mostra que o algoritmo se adapta muito bem às mudanças de uso e novos ataques, mantendo a alta precisão. As acurácias finais são 92,1% para o conjunto de dados do laboratório e 95.6% para o NetOp. As matrizes de confusão mostram que a taxa de detecção de ataque é maior do que a SGD, com valores de 87,5% e 81,4%, respectivamente. Além disso, o SVM também obtém uma boa taxa de falso positivo, com valores de 5,9% e 2,3%.



(a) Conjunto de dados do Laboratório.



(b) Conjunto de dados do Operador.

Figura 2: Máquinas de vetores de suporte *online*. Novamente, mesmo com o comportamento mudando, a acurácia permanece estável.

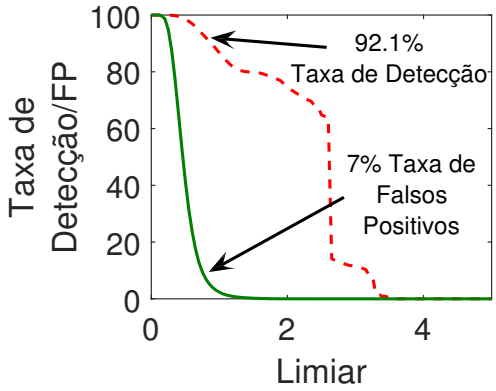
A detecção de anomalia é o terceiro esquema de detecção proposto, que tem a capacidade de descobrir ataques de dia zero que são difíceis de se detectar, já que ainda não existem dados sobre ele. Os parâmetros são atualizados durante o tempo de configuração com base apenas nos dados capturados pelos sensores de tráfego. Após esse período, se uma amostra for considerada normal, os parâmetros serão atualizados. Por outro lado, se for considerada uma ameaça, um alerta é emitido e os parâmetros de algoritmos não são atualizados. Para avaliar os algoritmos, utilizamos 70% dos dados normais para ambos os conjuntos de dados no tempo de configuração e os outros 30% para avaliar os falsos positivos. A taxa de detecção de ataque é obtida com base em todas as ameaças no conjunto de dados.

A identificação de uma anomalia é obtida pela análise do valor de entropia de uma janela deslizante de fluxos. A entropia de amostra, expressa por

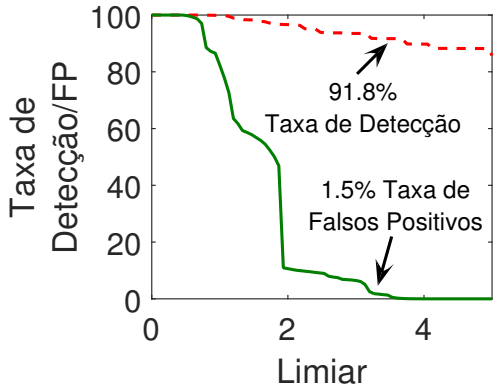
$$H(X) = -\sum_{i=1}^{N}(\frac{n_i}{S})\log_2(\frac{n_i}{S}), \tag{4}$$

indica o grau de concentração ou dispersão de uma característica, onde S é o número total de observações, n_i é o número de observações dentro do intervalo i de valores e N é o número de intervalos. Quando todos os valores estão concentrados em um intervalo, $H(X)$ é igual a zero e quando cada valor está em uma faixa diferente i, o valor de $H(X)$ é $\log_2(N)$. Então, dada uma série de observações X, a entropia de amostra resume o nível de concentração em um único valor. Foi definida uma janela deslizante de 40 fluxos e calculado o valor de $H(X)$ para cada uma dessas janelas, gerando as séries temporais. Outro parâmetro determinado na fase de configuração é o intervalo contendo a maioria das amostras. Estes parâmetros são atualizados conforme novas amostras chegam. Os valores normais de entropia do tráfego tendem a ser concentrados e o valor mais frequente tende a estar no centro. Assim, o esquema de detecção de ameaça por anomalia proposto define um limiar que determina a distância aceita da entropia $H(X)$ para o intervalo mais frequente.

A Figura 3 mostra os resultados para diferentes valores de limiar. Para valores de limiar pequenos, a taxa de detecção é muito alta, resultando também em uma alta taxa de falsos positivos. Para valores baixos, ocorre o oposto, uma baixa taxa de falsos positivos ao custo de uma baixa taxa de detecção. No entanto, existem valores de limiar que apresentam um *trade-off* excelente entre essas taxas. Para o conjunto de dados do laboratório, o limite 1,3 resulta em



(a) Conjunto de dados do Laboratório.



(b) Conjunto de dados do Operador de Rede.

Figura 3: Série Temporal de Entropia.

92,1% de taxa de detecção de ataque com 7% de taxa de Falso Positivo, enquanto que para o conjunto de dados NetOp, o limite de 3,4 obtém uma taxa muito baixa de Falsos positivos, de 1,5%, com uma taxa de detecção de ataque alta, de 91,8%.

5. Desenvolvimento e Avaliação de um Protótipo do Sistema Proposto

No protótipo desenvolvido a detecção de intrusão é realizada fora da rota através do espelhamento do tráfego real para não gerar latência no tráfego de produção e para poder reunir informações de outras rotas e correlacionar dados importantes para a detecção correta dos ataques. Com isso, a prevenção do tráfego malicioso é feita de maneira eficiente utilizando a programabilidade das redes definidas por *software* (*Software Defined Networks* - SDN) para realizar o espelhamento de apenas alguns pacotes de um fluxo por período.

A detecção dos ataques é feita com base na análise periódica dos primeiros pacotes de cada fluxo. Para fazer isso, sempre que um fluxo indefinido chega a um comutador SDN, ele o encaminha ao controlador, que então instala uma regra com duas ações, a primeira encaminhando o fluxo para o seu destino e a segunda replicando o tráfego e o encaminhando para a máquina de análise. O início de cada fluxo sempre é encaminhado à máquina de análise, já que fluxos novos não estão definidos. Isso é importante, pois aplicações possuem um comportamento bem definido no início. Por sua vez, a máquina de análise mantém um controle do número de pacotes de cada fluxo que ela está analisando e quando esse número chega a cinco pacotes ela envia as informações do fluxo para o sistema de detecção de intrusão e também uma mensagem avisando ao controlador que a análise dos pacotes daquele fluxo específico foi concluída. Um fluxo é definido como todos os pacotes de um mesmo IP origem e mesmo IP destino e a máquina analisadora extrai as 26 características do fluxo. Após o término da análise de fluxo, o controlador recebe esta informação, lista todos os fluxos que possuem os IPs de origem e destino do fluxo e retira a ação de desvio. O controlador mantém ainda a ação de encaminhamento para o destino, o que preserva o funcionamento da rede. É importante ressaltar que o sistema proposto é robusto contra ataques de negação de serviço e possui uma grande capacidade de analisar fluxos devido ao fato do fluxo não ser mais encaminhado o sistema depois de realizada a análise. É igualmente importante ressaltar que quando o controlador retira a ação de desvio, a ação voltará a ser ativa após o término do temporizador do fluxo definido no controlador. Este procedimento faz com que os fluxos sejam checados periodicamente, aumentando a segurança da rede e evitando que atacantes burlem o sistema proposto ao começar com um fluxo normal e depois mudar para o ataque. Como o fluxo será analisado novamente após o

término do temporizador, o ataque será detectado, mesmo com o início sem ameaças.

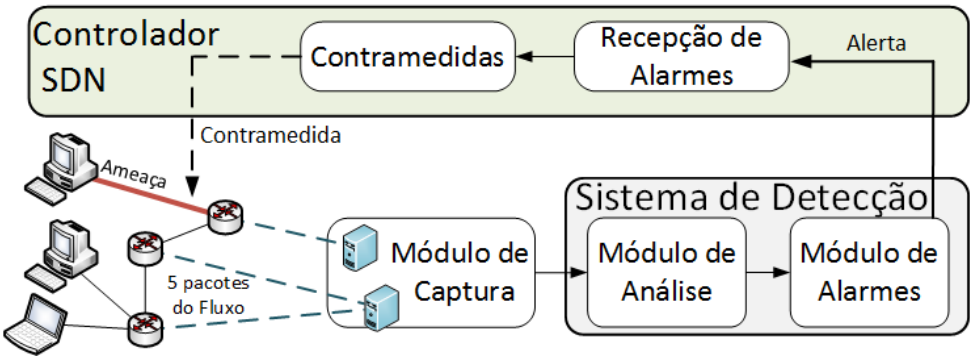


Figura 4: Esquema de prevenção de ameaças proposto: i) o controlador instala regras de desvio para o módulo de captura, ii) o tráfego desviado é analisado pelo sistema de detecção que gera alertas e iii) o controlador bloqueia as ameaças.

A Figura 4 mostra uma rede exemplo sendo analisada e protegida pelo sistema de detecção e prevenção de intrusão proposto. Todos os comutadores SDN possuem uma interface para qual o tráfego é duplicado e enviado até a máquina de análise. O controlador é o responsável por instalar essa regra e também por retirá-la quando a máquina já recebeu os cinco pacotes necessários. O módulo de captura, composto pelas máquinas analisadoras, encaminha então as informações dos fluxos para o sistema de detecção, que utiliza algoritmos adaptativos de aprendizado de máquina em tempo real para fazer sua análise. Caso alguma ameaça seja detectada, o sistema de detecção envia um alerta para o controlador, que então bloqueia o IP de origem do atacante em todos os comutadores.

5.1. Estratégia de Defesa contra Ataques com IP Mascarado

A regra de bloqueio do IP origem nos comutadores não é efetiva contra ataques que alternam periodicamente o IP de origem para enganar os sistemas de defesa. A situação é ainda mais crítica quando o encaminhamento do tráfego é feito usando as redes definidas por *software*, já que além dos danos do ataque, outras duas consequências são a sobrecarga no controlador para definir novos fluxos a cada mudança de IP e a ocupação excessiva das tabelas de encaminhamento dos comutadores pelos fluxos criados com os IPs falsos. Para resolver esse problema, uma estratégia de defesa contra ataques com IP mascarado foi proposta e desenvolvida, baseando-se em uma sequência de alertas e na marcação de caminho dos fluxo. A intuição por trás desse conceito é a seguinte: se o controlador instalou uma regra de bloqueio contra um ataque e, mesmo assim, um alarme chegou logo depois, pode significar que a regra de bloqueio não foi eficaz. Portanto, o controlador mantém um controle de tempo entre o recebimento de alertas e quando dois alertas chegam em um intervalo de tempo curto, o controlador ”suspeita”de que o IP do ataque está sendo mascarado. A partir dessa suspeita, o controlador passa a mapear o caminho de todos os fluxos da rede por um determinado período de tempo. Isto é possível graças à visão global do controlador, que conhece toda a topologia da rede. Agora, se um terceiro alerta chega ao controlador, além de fazer a regra tradicional de bloqueio do IP de origem, o controlador também consegue descobrir por qual porta do comutador o atacante entra na rede e pode instanciar uma regra de bloqueio diretamente na porta em questão. Portanto, se um atacante estiver mascarando o IP do ataque, após três alertas, seu tráfego de ataque será bloqueado. Um aspecto importante a ser destacado é que o processamento adicional para o controlador monitorar o caminho de todos os fluxos só ocorre quando dois alertas ocorrem em um curto período de tempo e é desfeito após um tempo sem receber nenhum novo alerta.

Além disso, a regra de bloqueio do IP que está atacando é mantida, pois pode não se tratar de um ataque com IP mascarado, mas sim um ataque distribuído. Contra um ataque distribuído, é essencial que se tente bloquear todas as origens e, ao impedir o tráfego tanto da interface de entrada na rede quanto do IP de origem, a regra de bloqueio será mais efetiva.

5.2. Monitoramento e Bloqueio de Ameaças

Foi desenvolvido em um ambiente real uma rede de teste sobre uma plataforma de experimentação para avaliar o protótipo do sistema proposto, o esquema de monitoramento de apenas cinco pacotes e o bloqueio do tráfego usando redes definidas por *software*. A virtualização é realizada através do hipervisor Xen e o encaminhamento do tráfego é desempenhado pelo OpenFlow. A Figura 5 mostra a topologia montada para os experimentos, que é formada por três máquinas clientes que se comunicam com uma máquina servidora. A comunicação é feita através de comutadores Open vSwitch que são controlados por uma aplicação programada no controlador POX. Além disso, também há a presença de uma máquina de análise que caracteriza os fluxos usando a ferramenta Bro. Na ligação entre os comutadores e a máquina de análise é necessário um túnel GRE (*Generic Routing Encapsulation*), que encapsula os pacotes e coloca o endereço da máquina de análise como destino. Na máquina de análise, os pacotes são então desencapsulados e analisados. Os experimentos foram realizados em um servidor Intel Xeon X5690 com 24 núcleos de processamento, cada um deles com frequência de 3.47GHz de relógio e com 48 GB de memória RAM.

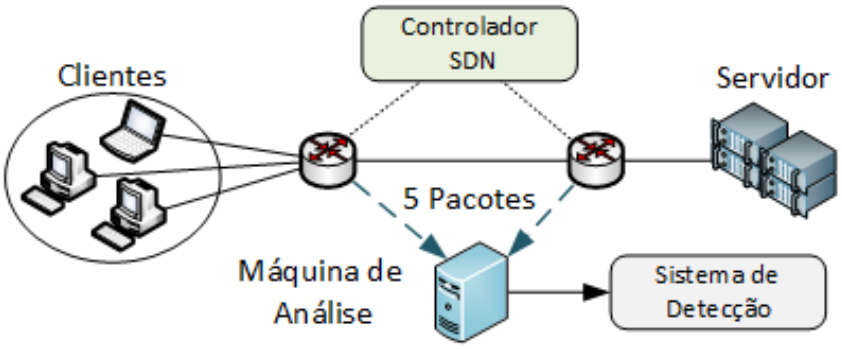
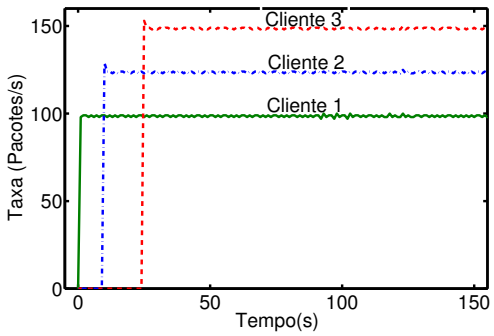


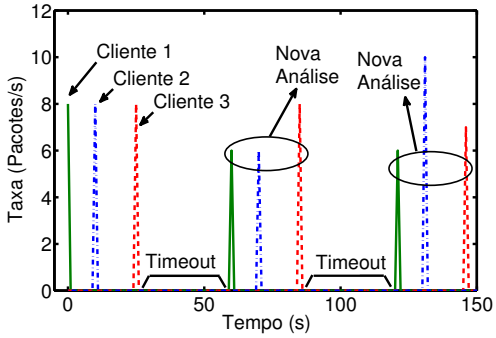
Figura 5: Topologia de rede implementada para os testes do esquema de análise de apenas cinco pacotes e do bloqueio do tráfego malicioso.

A Figura 6 mostra o funcionamento da regra de desvio de tráfego e seu posterior fim após a análise de cinco pacotes. Nesse experimento, as três máquinas clientes enviam tráfego a uma taxa constante para a máquina servidora. A Figura 6a mostra que o esquema proposto funciona muito bem, sem influir no tráfego de produção recebido pela máquina servidora. Já a Figura 6b mostra a taxa de pacotes recebida pela máquina de análise, que envia uma mensagem ao controlador depois de conseguir capturar os cinco pacotes necessários para caracterizar o fluxo. Por isso, apesar da taxa de envio ser bem mais elevada, a máquina de análise recebe poucos pacotes. A razão pela qual ela recebe um pouco mais que cinco pacotes é o tempo de comunicação entre a máquina de análise e o controlador. Outro comportamento que pode ser observado nessa figura é que os fluxos são analisados periodicamente e, sempre que o tempo de duração do fluxo definido no controlador se esgota, o fluxo é analisado novamente. Nesse experimento, foi definido o tempo de 60 segundos e é possível perceber que a máquina de análise recebe os pacotes a serem analisados de cada cliente periodicamente. Dois aspectos importantes de serem ressaltados nessa proposta são o tempo necessário para caracterizar o fluxo e a capacidade de se aumentar a quantidade de fluxos a serem analisados pela máquina

de análise. Como a máquina só requer cinco pacotes para caracterizar o fluxo, ela não precisa esperar até o fim do fluxo ou da conexão para enviar as informações para o sistema de detecção de intrusão, o que tem como consequência um tempo menor na detecção e bloqueio de ameaças. Além disso, a máquina de análise fica imune à ataques de inundação, pois não recebe todos os pacotes de um fluxo. Pela mesma razão, também se pode analisar uma quantidade muito maior de fluxos, garantindo robustez e um potencial para escalabilidade.



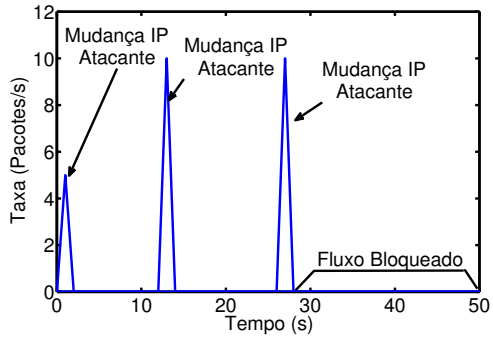
(a) Taxa de pacotes recebidos pela máquina servidora.



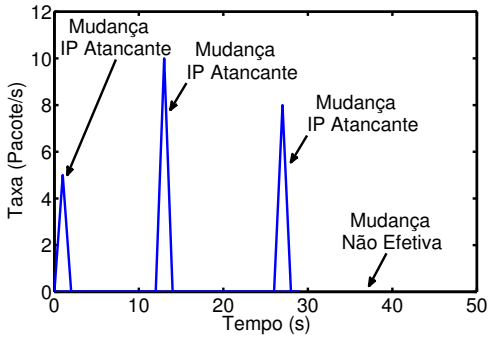
(b) Taxa de pacotes recebidos pela máquina de análise.

Figura 6: Funcionamento da rede quando há a comunicação de três máquinas com o servidor. O servidor recebe o tráfego sem ser afetado pelo esquema, enquanto a máquina de análise recebe poucos pacotes de cada fluxo.

O segundo experimento mostra uma situação de ameaça na qual uma das máquinas cliente ataca o servidor. A máquina atacante, além de realizar a ameaça, mascara o IP de origem para dificultar ainda mais a detecção. De maneira semelhante ao primeiro experimento, a Figura 7 mostra o tráfego recebido pela máquina servidora e pela máquina de análise. Após o ataque começar e a máquina análise enviar as informações para o sistema de detecção de intrusão, uma regra bloqueando o IP de origem imediatamente é instalada em todos os comutadores, o que faz com que o tráfego recebido pela servidora fique com taxa zero após o tempo de detecção, como ilustrado na Figura 7a.



(a) Taxa de pacotes do atacante recebidos pela máquina servidora.



(b) Taxa de pacotes do atacante recebido pela máquina de análise.

Figura 7: Funcionamento do sistema proposto sob um ataque que mascara o endereço IP de origem. As regras de bloqueio deixam de ser efetivas quando a máquina modifica o IP de origem e o bloqueio volta a ser efetivo após a identificação e localização da interface pela qual o ataque entra na rede.

Assim, quando a máquina altera o IP de origem pela primeira vez, o fluxo volta a chegar na máquina servidora e é novamente desviado e analisado, como mostrado na Figura 7b.

Porém, desta vez quando o controlador recebe o alerta, ele passa a marcar o caminho dos fluxos para poder mapear a entrada na rede de cada fluxo. Mais uma vez o fluxo do IP de origem é bloqueado e a máquina servidora para de receber tráfego. Novamente, em torno de 28 segundos a máquina atacante muda seu IP de origem e o tráfego é novamente analisado e um novo alerta é gerado. Contudo, dessa vez, quando o controlador recebe o alerta, ele procura por qual interface do comutador o tráfego desse IP está vindo e cria uma regra para bloquear o tráfego dessa interface. Agora, quando o atacante altera novamente o IP, o tráfego dele não chega mais na máquina servidora, nem é analisado novamente. Isso ocorre pois o tráfego malicioso é bloqueado o mais perto possível da origem. Uma observação importante é que apesar das Figuras 7a e 7b serem muito parecidas, a máquina de análise só recebe essa quantidade pequena de pacotes por causa da regra de fim de desvio que o controlador instala, enquanto no resto da rede o bloqueio acontece em um tempo curto após a geração do alerta.

6. Conclusão

Este artigo apresentou um sistema adaptativo e eficaz de detecção e bloqueio de tráfegos maliciosos. A detecção do sistema proposto é feita com base na análise das características de cinco pacotes de cada fluxo. Essa abordagem se mostra eficiente em diversos aspectos. Primeiro, os resultados mostraram um excelente desempenho, classificando as amostras de fluxo com acurácias maiores que 90%. Além disso, ao se analisar apenas cinco pacotes, a detecção de ameaças é efetuada praticamente de maneira instantânea, sem ter de esperar o fim das conexões para conseguir classificar os fluxos como legítimos ou maliciosos. Por fim, a máquina que analisa o tráfego da rede é colocada fora da rota do pacote e, como só recebe poucos pacotes de cada fluxo, ela fica imune a ataques de inundação que poderiam vir a impedir que ela extraísse as características de todos os fluxos.

Três algoritmos de detecção de ameaças foram propostos: dois por classificação e um por anomalia. Os três utilizam os dados provenientes de *honeypots* como assinaturas de ataques para adaptar seus modelos em tempo real e aprender a detectar novas ameaças. O algoritmo de detecção de anomalias apresenta uma boa relação entre detecção e falsos positivos e os de classificação se mostraram capazes de se adaptar a novas ameaças e mudança de comportamento legítimo. Por fim, todo o esquema para a prevenção de intrusão foi realizado com o auxílio das redes definidas por *software*. Como prova de conceito, foi implementado um protótipo em uma plataforma híbrida de testes que combina o hypervisor Xen com o OpenFlow. Além disso, há uma estratégia de defesa contra os tipos de ataques que mudam dinamicamente de IP. Isso é essencial, pois, além de ser uma ameaça à máquina que estiver sendo atacada, o mascaramento do IP também tem como consequência uma negação de serviço do controlador. Os resultados mostraram o efetivo bloqueio de uma ameaça ao se mapear o caminho do fluxo de acordo com o intervalo de tempo dos alertas, que podem indicar a presença de um ataque com IP mascarado.

Referências

Andreoni Lopez, M., Mattos, D. M. F. e Duarte, O. C. M. B. (2016). An elastic intrusion detection system for software networks. *Annales des Telecommunications/Annals of Telecommunications*, 71(11-12):595–605.

Bernaille, L., Teixeira, R., Akodkenou, I., Soule, A. e Salamatian, K. (2006). Traffic classification on the fly. *ACM SIGCOMM Computer Communication Review*, 36(2):23–26.

Braga, R., Mota, E. e Passito, A. (2010). Lightweight DDoS flooding attack detection using NOX/OpenFlow. Em *IEEE 35th Conference on Local Computer Networks*, páginas 408–415.

Buczak, A. e Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys Tutorials*, (99):1–26.

Donato, W. D., Pescape, A. e Dainotti, A. (2014). Traffic identification engine: an open platform for traffic classification. *IEEE Network*, 28(2):56–64.

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

Ji, S.-Y., Jeong, B.-K., Choi, S. e Jeong, D. H. (2016). A multi-level intrusion detection method for abnormal network behaviors. *Journal of Network and Computer Applications*, 62:9–17.

Lakhina, A., Crovella, M. e Diot, C. (2005). Mining anomalies using traffic feature distributions. Em *ACM SIGCOMM Computer Communication Review*, volume 35, páginas 217–228. ACM.

Lee, W., Stolfo, S. J. e Mok, K. W. (1999). Mining in a data-flow environment: Experience in network intrusion detection. Em *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, páginas 114–124. ACM.

Lippmann, R. P., Fried, D. J., Graf, I., Haines, J. W., Kendall, K. R., McClung, D., Weber, D., Webster, S. E., Wyschogrod, D., Cunningham, R. K. et al. (2000). Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. Em *Proceedings of DARPA Information Survivability Conference and Exposition. DISCEX'00.*, volume 2, páginas 12–26. IEEE.

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S. e Turner, J. (2008). OpenFlow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev., 2008*.

Peng, L., Yang, B. e Chen, Y. (2015). Effective packet number for early stage internet traffic identification. *Neurocomputing*, 156:252 – 267.

Ponemon, I. e IBM (2017). 2017 cost of data breach study: Global analysis. `www.ibm.com/security/data-breach/`. Acessado: 15/07/2017.

Sharafaldin, I., Gharib, A., Lashkari, A. H. e Ghorbani, A. A. (2017). Towards a reliable intrusion detection benchmark dataset. *Software Networking*, 2017(1):177–200.

Sommer, R. e Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. Em *IEEE Symposium on Security and Privacy (SP)*, páginas 305–316. IEEE.

Suthaharan, S. (2014). Big data classification: Problems and challenges in network intrusion prediction with machine learning. *ACM SIGMETRICS Performance Evaluation Review*, 41(4):70–73.

Wu, K., Zhang, K., Fan, W., Edwards, A. e Yu, P. S. (2014). RS-Forest: A rapid density estimator for streaming anomaly detection. Em *IEEE International Conference on Data Mining (ICDM)*, páginas 600–609.