Detecção de Intrusão Através de Aprendizagem de Fluxo no Ambiente do Adversário

Eduardo K. Viegas[1], Altair O. Santin[1], Vilmar Abreu[1], Luiz E. S. Oliveira[2]

[1]Pontifícia Universidade Católica do Paraná (PUCPR) – Escola Politécnica – Programa de Pós-Graduação em Informática (PPGIa) – Curitiba, PR – Brasil

[2]Departamento de Informática - Universidade Federal do Paraná (UFPR) – Curitiba, PR Brasil

{eduardo.viegas,santin,vilmar.abreu}@ppgia.pucpr.br,
lesoliveira@inf.ufpr.br

Abstract. *Despite the existence of several works that uses anomaly-based intrusion detection mechanisms, such approaches are hardly used in production environments. In general, the literature does not consider the adversarial environment properties, in which an attack will attempt to evade the detection mechanism. This paper proposes and evaluates an approach to reliably perform intrusion detection through stream learning techniques in the adversarial environment. The proposal relies in class-specific anomaly detectors and a rejection mechanism in order to reliably update the detection mechanism. The proposal evaluation showed that the proposal provides resiliency to both causative and exploratory attacks.*

Resumo. *Apesar da existência de diversos trabalhos que utilizam técnicas de detecção de intrusão baseada em anomalia, dificilmente tais técnicas são utilizadas em produção. Percebe-se que, em geral, a literatura não considera o ambiente do adversário, em que um atacante tenta evadir o mecanismo de detecção. Neste artigo é proposto e avaliado uma abordagem para efetuar a detecção de intrusão em fluxo de dados de forma confiável no ambiente do adversário. A proposta utiliza detectores de anomalia específicos as classes consideradas e um mecanismo de rejeição para permitir a atualização do sistema de forma confiável. A avaliação da proposta mostrou que a abordagem provém resiliência a ataques causais e exploratórios.*

1. Introdução

Sistema de Detecção de Intrusão (*Intrusion Detection System - IDS*) é uma abordagem amplamente referenciada pela literatura para detecção de ataques em nível de rede.. Um IDS permite a detecção de ataques, uso malicioso ou inadequado de um sistema computacional ou uma rede de computadores [García-Teodoro et al. 2009]. As detecções de intrusões são efetuadas, geralmente, através da utilização de técnicas de aprendizagem de máquina [Jyothsna, V. et al. 2011]. Nessa abordagem, o IDS é treinado como um conjunto limitado de perfis/comportamentos. Posteriormente, durante seu uso em produção, qualquer evento (e.g. um pacote da rede) que desvia significativamente dos perfis previamente conhecidos é classificado como uma tentativa de intrusão [Denning, D. E. 2012]. Essa abordagem é chamada detecção de intrusão baseada em anomalia, onde é possível detectar novos ataques.

Apesar dos resultados promissores reportados na literatura, raramente a detecção de intrusão baseada em anomalia é utilizada em produção [Sommer, R. e Paxson, V. 2012]. Neste sentido, a abordagem baseada em assinatura que realiza a classificação através do confronto do evento com padrões conhecidos de ataques ainda é a mais utilizada [Axelsson, S. 2000]. Essa discrepância entre as abordagens da literatura e a utilização em produção ocorre principalmente devido ao alto número de falsos alertas emitidos [Sommer, R. e Paxson, V. 2012] e a necessidade de atualização do algoritmo de detecção de intrusão ao longo do tempo [Maggi, F. et al. 2009]. O algoritmo deve ser atualizado devido a mudanças de comportamentos no ambiente, tal como a ocorrência de novos tráfegos de rede e novos ataques [Axelsson, S. 2000].

Dessa maneira, nos últimos anos as técnicas de aprendizagem de fluxo (*stream learning*) [He, H. et al. 2011] estão ganhando destaque. Essas técnicas são geralmente utilizadas em cenários onde o conjunto de conceitos (classes) mudam ao longo do tempo [He, H. et al. 2011]. Por exemplo, no contexto de detecção de intrusão em rede, o comportamento legitimo (normal) muda ao longo do tempo com a implantação de novos serviços, enquanto o comportamento do atacante (intruso) também muda devido a geração de novos ataques [Sommer, R. e Paxson, V. 2012]. Nesse cenário em constante mudanças, a atualização do mecanismo de detecção é frequentemente efetuada em janelas de tempo deslizantes (*sliding windows*). Assim, os eventos mais recentes são identificados com maiores importâncias durante a etapa de classificação, enquanto eventos antigos são descartados [Bifet, A. et al. 2007].

Essa capacidade de lidar com ambientes que mudam ao longo do tempo, permite que diversos trabalhos argumentem que técnicas de *stream learning* possibilitam o uso de abordagens de detecção de intrusão baseada em anomalia em ambientes de produção [Jyothsna, V. et al. 2011]. Porém, na literatura, ainda existe carência na validação de tais abordagens [Sommer, R. e Paxson, V. 2012]. Em geral, os trabalhos na literatura não consideram o ambiente do adversário, onde um atacante pode tentar evadir o mecanismo de detecção, seja pela exploração das propriedades do mecanismo de detecção ou através da injeção de ataques durante a etapa de treinamento do sistema [Tygar, J. D. 2009].

Neste artigo é apresentado uma abordagem para a utilização de técnicas de detecção de anomalia para fluxos de dados que permite a detecção de intrusão de rede de maneira confiável. A proposta utiliza detectores de anomalias específicos a cada classe considerada para automaticamente e confiavelmente atualizar o mecanismo de detecção de intrusão ao longo do tempo. A proposta considera as características do ambiente do adversário através da rejeição de potenciais tentativas de evasão ou classificações não confiáveis.

2. Fundamentação

Esta seção apresenta duas abordagens consideradas para a detecção de intrusão baseada em anomalias: aprendizagem de máquina tradicional e *stream learning*.

2.1. Aprendizagem de Máquina Tradicional

A aprendizagem de máquina tradicional utiliza os algoritmos de reconhecimento de padrões tradicionais [Corona, I. et al. 2013], em que seu processo geralmente se baseia em um conjunto de dados fixo para treinamento. Nesse caso, o comportamento do usuário normal e do atacante são obtidos e armazenados em um conjunto fixo de dados. O

conjunto de dados, geralmente referido como conjunto de dados de treinamento, é usado durante a etapa de treinamento do classificador (algoritmo de reconhecimento de padrões). Durante esse processo, uma taxa de falso positivo (FP) e falso negativo (FN) é estimada através de outro conjunto de dados denominado conjunto de dados de teste. A taxa de FP refere-se à taxa em que os eventos normais são incorretamente classificados como uma tentativa de intrusão, enquanto a taxa de FN refere-se à taxa em que eventos de intrusão são incorretamente classificados como uma atividade normal.

No processo de aprendizagem da máquina tradicional, quando o comportamento do atacante muda, o algoritmo de reconhecimento de padrões deve ser treinado novamente. Esse processo de treinamento normalmente é uma tarefa dispendiosa, uma vez que o ambiente deve ser monitorado para que os novos comportamentos sejam identificados (geralmente de forma manual). Assim, o classificador deve ser treinado e as taxas de FP e FN devem ser novamente estimadas. Além disso, identificar mudanças de comportamentos no ambiente é uma tarefa desafiadora, uma vez que a identificação é frequentemente baseada no aumento das taxas de FP e FN, que também são identificadas manualmente através da assistência de especialistas [Sommer, R. e Paxson, V. 2012].

A facilidade de atualização torna-se relevante no campo de detecção de intrusão baseado em anomalias. Novos ataques são descobertos diariamente, juntamente com novos serviços e seus conteúdos. Assim, independentemente do mecanismo de detecção de intrusão usado, o sistema deve ser frequentemente atualizado [Sommer, R. e Paxson, V. 2012].

2.2. Aprendizagem de Máquina para Fluxos de dados (*Stream Learning*)

As técnicas tradicionais de aprendizagem de fluxo (*stream learning*) visam lidar automaticamente com as mudanças do ambiente ao longo do tempo [He, H. et al. 2011]. Para isso, o algoritmo de aprendizagem de fluxo é executado em um ciclo repetido, no qual o algoritmo é atualizado de acordo com os eventos recebidos do fluxo de dados da rede [Bifet, A. et al. 2007]. Esse processo ocorre de acordo com os limites de memória e processamento. Nesse caso, os limites de memória são definidos de acordo com o número de eventos considerados durante o processo de classificação. Para esse propósito, a maioria das estratégias depende de uma abordagem de janela deslizante (*sliding window*), na qual uma janela (intervalo pré-definido) de eventos recentes é mantida e os eventos mais antigos são descartados de acordo com um conjunto de regras [Bifet, A. et al. 2007].

Devido à sua natureza adaptativa, um algoritmo de classificação de fluxo de dados deve apresentar as seguintes propriedades [Bifet, A. et al. 2007]: (*i*) processar e inspecionar um evento por vez; (*ii*) usar quantidade limitada de memória; (*iii*) classificar os eventos em uma quantidade restrita de tempo e (i*v*) prever novos eventos a qualquer momento.

2.3. Aprendizagem de Máquina no Ambiente do Adversário

Ao longo dos últimos anos, as técnicas tradicionais de aprendizagem de máquina e de fluxo foram aplicadas com sucesso em vários campos [Sommer, R. e Paxson, V. 2012]. Por exemplo, foram aplicadas em reconhecimento de imagem, sistemas de recomendação de produtos, tradução de linguagem natural [Oliveira, L. S. et al. 2002] entre outros. No entanto, em detecção de intrusão ainda existe falta de aplicabilidade em ambientes de produção [Sommer, R. e Paxson, V. 2012].

Um campo de pesquisa emergente, conhecido como aprendizagem de máquina no ambiente do adversário [Tygar, J. D. 2009] considera o uso de técnicas de aprendizagem de máquina nas configurações de um adversário - chamado configurações adversárias. Nesses casos, o adversário (atacante) tentará evadir o mecanismo de detecção de intrusão usando técnicas sofisticadas de ataques, denominadas de ataques causais e exploratórios [Tygar, J. D. 2009]. Os ataques causais referem-se a ataques que ocorrem durante o processo de treinamento [Tygar, J. D. 2009]. Neste caso, o atacante tenta injetar tentativas de intrusões, estas incorretamente classificadas, no conjunto de dados de treinamento como eventos normais. Por outro lado, os ataques exploratórios visam explorar as propriedades do algoritmo de aprendizagem de máquina [Tygar, J. D. 2009], através da manipulação de uma tentativa de intrusão, de forma que o mecanismo de detecção classifique o evento como uma atividade normal.

Um exemplo de ataque causal em um *stream learning* tradicional é exibido na Figura 1. Nesse cenário, o classificador classifica anomalias (*outliers*) ao longo do tempo. No Tempo 1 (Figura 1, Tempo 1), a janela deslizante é totalmente preenchida com eventos normais (*inliers*), desta maneira os ataques são classificados como *outliers*. No entanto, independentemente da classe atribuída, o ataque, inicialmente classificado como *outlier*, é adicionado na janela deslizante, enquanto um *inlier* (evento mais antigo na janela deslizante) é removido. O mesmo procedimento ocorre no Tempo 2 (Figura 1, Tempo 2). No entanto, o novo ataque ainda é classificado como *outlier*, pois não há eventos suficientes em sua vizinhança. Finalmente, no Tempo 3 (Figura 1, Tempo 3), o atacante é capaz de injetar ataques suficientes na janela deslizante, pois o número de vizinhos está próximo o suficiente para formar um grupo e ser classificado como *inlier*.

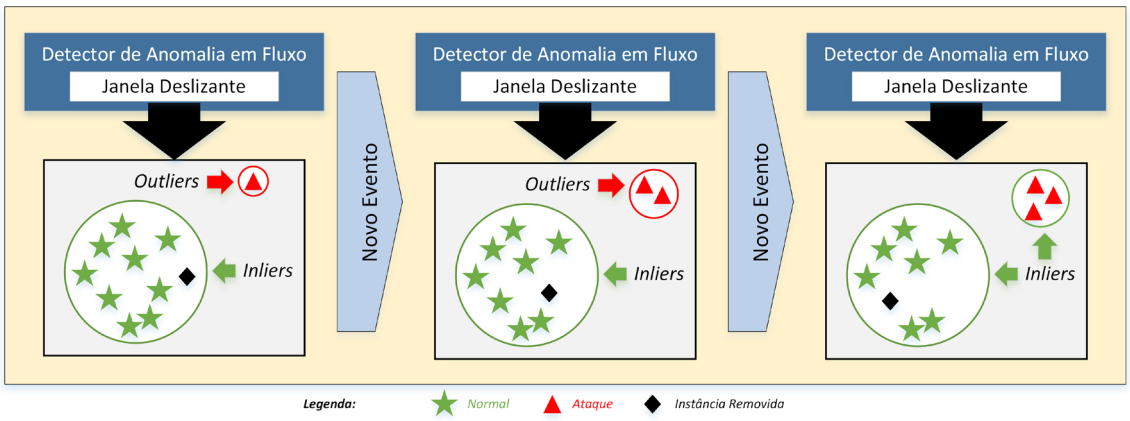

Figura 1. Comportamento da Janela Deslizante em detectores de anomalias em fluxo tradicionais.

Dessa maneira, o uso de técnicas de aprendizagem de máquina, especialmente técnicas de *stream learning*, na configuração do adversário visam a concepção de mecanismos de detecção que sejam capazes de resistir ou apresentar resiliência aos ataques (sofisticados) mencionados anteriormente.

3. Proposta

Esta proposta provê uma abordagem confiável de *stream learning* para detecção de intrusão baseada em anomalia, que permite a atualização automática do mecanismo de detecção ao longo do tempo. Para isso, a proposta utiliza detectores de anomalias

específicos a classe, provendo resiliência para ataques causais e exploratórios (seção 2.3). O método proposto é exibido na Figura 2 e descrito nas próximas subseções.

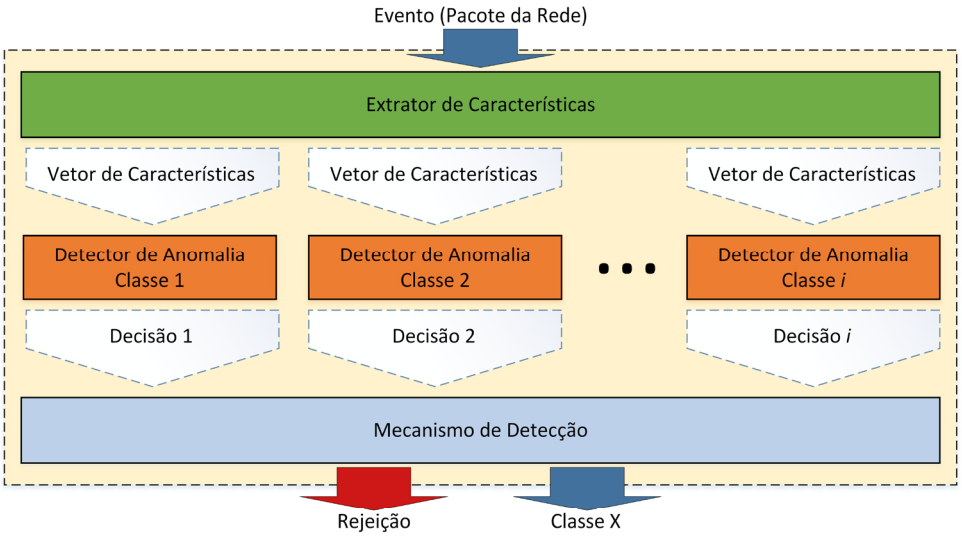

Figura 2. Método proposto para detecção de intrusão baseada em anomalia através de algoritmos de aprendizagem de fluxo no ambiente do adversário.

3.1. Esquema de Detecção

O método proposto considera que o esquema de detecção utiliza detectores de anomalias em fluxo específicos a classe. Por exemplo, um detector de anomalia para eventos normais e outro para ataques. A detecção é efetuada de acordo com a Figura 2. O processo é composto de 5 etapas: (*i*) o conjunto de características é extraído do evento a ser classificado, e.g. um pacote da rede; (*ii*) o vetor de características é fornecido a cada detector de anomalia; (*iii*) cada detector realiza a detecção atribuindo a classe ao evento como *inlier* (evento pertence ao grupo da classe do detector de anomalia) ou *outlier* (evento não pertence ao grupo do detector de anomalia); (*iv*) o mecanismo de detecção recebe a decisão de cada detector de anomalia e tenta encontrar um consenso entre as decisões; (*v*) se a decisão é unanime, a classe determinada é atribuída ao evento, caso contrário, a decisão sobre o evento é rejeitada.

Quando o mecanismo de detecção recebe uma decisão de evento, decide se a classificação do evento é confiável ou não. A confiabilidade da classificação de um evento (representado como "Classe X" na Figura 2) é provida da nulidade da interseção da decisão de todos os classificadores. O cálculo de confiabilidade é exibido na Fórmula 1, sendo que $Decisão_i$ determina cada saída dos detectores de anomalias.

$$\cap_{i=1}^{n}(Decisão_i) = \emptyset \qquad \text{(Fórmula 1)}$$

Como exemplo, considere dois algoritmos de detecção de anomalias, um para eventos normais e outro para ataques. Um evento que é classificado como *inlier* para normal e *outlier* para ataque é confiável, uma vez que a decisão é uma unanimidade. Ou seja, não há interseção entre classes de classificação nos diferentes mecanismos de detecção. No entanto, um evento que é classificado como *inlier* para mais de um detector de anomalia deve ser rejeitado, pois a decisão não é confiável. As classificações rejeitadas indicam que uma possível tentativa de evasão ou um falso alarme podem estar ocorrendo. Dessa maneira, outro mecanismo de detecção deve ser usado, por exemplo, um mecanismo de detecção de intrusão baseado em assinatura ou inspeção manual.

3.2. Provendo Resiliência a Adversários – Exploratório

Ao contrário dos algoritmos tradicionais de aprendizagem de fluxo, a abordagem proposta fornece resiliência a ataques exploratórios através do comportamento imutável de cada algoritmo de detecção de anomalias. O comportamento imutável é definido por uma restrição que não permite que um *outlier* se torne um *inlier* no algoritmo de detecção de anomalias ao longo do tempo. A abordagem proposta considera que para detecção de intrusão baseada em anomalias, um evento inicialmente classificado como *outlier* não se tornará *inlier* em nenhum momento do tempo. Por exemplo, um ataque que foi classificado como *outlier* (ataque) pelo algoritmo de detecção de anomalia da classe normal não deve ser classificado como um evento normal, mesmo que a sua ocorrência aumente na janela deslizante ao longo do tempo.

Ao usar o comportamento imutável, o atacante não poderá explorar o intervalo da janela deslizante para perverter (poluir) a classificação de eventos que estão sendo analisados por um detector de anomalia. É importante notar que os eventos classificados como *inlier* continuam a serem adicionados à janela deslizante de aprendizagem de fluxos. Portanto, o algoritmo ainda é capaz de se adaptar às mudanças no fluxo. Porém, a proposta mitiga um possível ataque de evasão, em que a quantidade de eventos *outliers* se tornam predominantes em uma janela deslizante, desta maneira, tornando um evento *outlier* em *inlier*.

3.3. Provendo Resiliência a Adversários – Causais

Para proporcionar resiliência a ataques causais, a proposta baseia-se tanto no comportamento imutável (seção 3.2) quanto nos detectores de anomalias específicos da classe. A proposta considera que a resiliência a ataques causais deve ser fornecida em duas etapas: treinamento inicial e readaptação contínua (atualização).

O treinamento inicial está relacionado a população inicial da janela deslizante do detector de anomalias. Nesta fase, as janelas deslizantes dos detectores de anomalias ainda estão sendo preenchidas, ou seja, estão suscetíveis a ataques causais. Desta maneira, a abordagem proposta pressupõe que há, pelo menos, uma população inicial que permita a classificação correta em pelo menos um detector de anomalias, uma vez que a janela deslizante será atualizada de acordo com os eventos iniciais. Preencher as janelas deslizantes iniciais com um número predominante de cópias do mesmo evento *inlier* é uma maneira de garantir a confiabilidade. Dessa maneira, o detector de anomalias é confiável, sendo que as saídas das classificações podem ser confiáveis se a unanimidade de classificação for alcançada, caso contrário a classificação é rejeitada.

A fim de fornecer readaptação contínua, a proposta depende do mecanismo de detecção de anomalia específico a classe e do comportamento imutável (Seção 3.2). O mecanismo de detecção de anomalia específico a classe fornece resiliência à manipulação do comportamento do evento. Por exemplo, o atacante deve manipular o comportamento do evento de modo em que o evento se comporte como um evento normal, enquanto também é um evento anômalo para o mecanismo de detecção de anomalias de ataque. Desta maneira, o comportamento imutável dificulta o ataque sobre a janela deslizante, uma vez que o atacante deve ter habilidades para manipular os eventos de maneira que perverta todos os detectores de anomalias.

4. Avaliação

4.1. Cenário de testes

Devido às limitações conhecidas das abordagens atuais em relação as bases de dados para avaliação de IDSs [Sommer, R. e Paxson, V. 2012], uma abordagem utilizada em trabalho anterior foi adotada [Viegas et al. 2017]. Desta forma, duas classes de tráfego de rede foram utilizadas para compor o conjunto de dados: normal e ataque. O ambiente de testes é composto por 100 máquinas clientes, 3 hosts atacantes e um único servidor.

Para gerar o tráfego normal, os serviços fornecidos no cenário de teste foram: HTTP, SSH, SMTP, SNMP e DNS. Foram gerados três conjuntos de ataques: SYNFlood, UDPFlood e ICMPFlood. O cenário de testes foi executado durante 30 minutos. A quantidade total de tráfego de rede gerada para cada serviço e ataque é relatada na Tabela 1. Para cada pacote de rede, são extraídos 23 atributos [Viegas et al. 2017].

Para fins de teste, o algoritmo de detecção de *outliers* em fluxo Micro-cluster (MCOD) [Kontaki, M. et al. 2011] foi considerado no método proposto (Figura 2, Seção 3). Para fins de comparação, foram consideradas duas outras abordagens: a Aprendizagem de Máquina Tradicional (Seção 2.1) e Aprendizagem de Fluxo Tradicional (Seção 2.2).

Tabela 1. Distribuição do tráfego no ambiente de testes

Classe	Tráfego Gerado	Quantidade de Pacotes
Normal	HTTP	20.238.802
	SMTP	2.298.222
	SSH	1.048.482
	SNMP	3.017.731
	DNS	135.188
Ataque	SYNFlood	471.288
	UDPFlood	121.645
	ICMPFlood	130.698

4.2. Obtenção do Modelo

Para o método proposto (Seção 3), duas classes foram consideradas: normal e ataque. Para cada teste, foram utilizados dois detectores de anomalias: um para o normal e outro para o ataque. Uma janela deslizante de 10.000 eventos foi considerada. Um total de 50 eventos foram estabelecidos como a quantidade de vizinhos próximos (k). Cada detector de anomalias específico a classe tem seu próprio parâmetro de raio. Uma série de testes foram efetuados para estabelecer os parâmetros do raio, sendo que o critério escolhido era minimizar o valor de objetivo da Fórmula 2.

$$objetivo = erro_{taxa} + rejeição_{taxa} \quad \text{(Fórmula 2)}$$

O $erro_{taxa}$ e $rejeição_{taxa}$ foram definidos através da detecção dos 10.000 eventos normais iniciais seguidos pela detecção de 10.000 eventos de ataque do conjunto de dados de treinamento (Tabela 1). Os valores de raio para cada detector de anomalia de classe, normal e ataque, foram variados em um intervalo de 0,01 entre 0,00 a 2,00.

O classificador k-Vizinhos Próximos (*k-Nearest* Neighbors - kNN) foi utilizado para a aprendizagem de máquina tradicional (Seção 2.1). Para permitir a comparação, um total de 5000 eventos para cada classe (normal e ataque) foram usados. Os 5000 eventos de cada classe são definidos pelo algoritmo de agrupamento *k-means* [Alsabti, K. et al. 1997] usando o conjunto de dados de treinamento (25% dos eventos escolhidos

aleatoriamente da Tabela 1). Os vizinhos do kNN configurados não são atualizados durante o processo de classificação. Finalmente, para a aprendizagem de fluxo tradicional (Seção 2.2), o MCOD é usado. No entanto, apenas a classe normal é considerada, conforme comumente é realizado em trabalhos relacionados [Kontaki, M. et al. 2011]. Finalmente, o processo de obtenção do raio foi estabelecido apenas pela minimização do erro.

4.3. Avaliação tradicional

Inicialmente, o processo de avaliação tradicional foi considerado para as abordagens avaliadas. Para a avaliação tradicional, as configurações do adversário (Seção 2.3) não são consideradas.

Para o classificador kNN, o conjunto de dados foi dividido em: treinamento, validação e teste, sendo que o conjunto de dados (Tabela 1) foi distribuído como 25% para treinamento, 25% para validação e 50% para teste. Por efetuar a sua aprendizagem em fluxo, o MCOD foi utilizado todo o conjunto de dados durante a avaliação tradicional. Neste caso, os eventos são reproduzidos exatamente na mesma ordem em que aparecem no conjunto de dados original (Tabela 1). A Tabela 2 exibe as taxas de acurácia em relação a cada uma das abordagens avaliadas. Sendo que a coluna do método se refere à abordagem utilizada durante a fase de detecção. Cada abordagem é testada com um conjunto diferente de ataques utilizados durante a etapa de treinamento, exibidos entre parênteses na coluna do método.

Tabela 2. Avaliação da abordagem proposta e das abordagens tradicionais.

Método	Acurácia (Rejeição)			
	Acurácia Normal (Rejeição)	Acurácia SYNFlood (Rejeição)	Acurácia UDPFlood (Rejeição)	Acurácia ICMPFlood (Rejeição)
Abordagem Proposta MCOD (SYNFlood)	100,00 (0.04)	100,00 (0,00)	- (100,00)	- (100,00)
Aprendizagem de Máquina Tradicional kNN (SYNFlood)	99,83 (N.A.)	100,00 (N.A.)	100,00 (N.A.)	0,01 (N.A.)
Abordagem Proposta MCOD (UDPFlood)	100,00 (0.98)	- (100,00)	100,00 (0,10)	- (100,00)
Aprendizagem de Máquina Tradicional kNN (UDPFlood)	99,93 (N.A.)	49,97 (N.A.)	100,00 (N.A.)	0,01 (N.A.)
Abordagem Proposta MCOD (ICMPFlood)	100,00 (0.97)	- (100,00)	- (100,00)	100,00 (0,12)
Aprendizagem de Máquina Tradicional kNN (ICMPFlood)	100,00 (N.A.)	3,23 (N.A.)	100,00 (N.A.)	100,00 (N.A.)
Aprendizagem de Fluxo Tradicional MCOD	99,19 (N.A.)	0,81 (N.A.)	0,69 (N.A.)	0,22 (N.A.)

É possível notar que tanto a abordagem proposta, como a aprendizagem de máquina tradicional (kNN) conseguem detectar o mesmo conjunto de ataques com uma taxa de precisão significativamente alta. Em relação à detecção de ataques, tanto a abordagem proposta como o kNN apresentaram uma taxa de FN de zero por cento, ao detectarem o mesmo conjunto de ataques ao qual o sistema foi treinado. Considerando a taxa de FP, o classificador kNN alcançou 0,17, 0,07 e zero por cento quando treinados com ataques SYNFlood, UDPFlood e ICMPFlood, respectivamente. A abordagem proposta alcançou uma taxa de FP de zero por cento em todos os casos testados. No entanto, a abordagem proposta rejeitou classificações potencialmente erradas. Nesse caso,

0,04, 0,98 e 0,97 por cento dos eventos normais foram rejeitados pelos ataques SYNFlood, UDPFlood e ICMPFlood, respectivamente. Pode-se observar que a abordagem proposta apresenta uma precisão de detecção semelhante quando comparada à abordagem tradicional de aprendizagem por máquina. No entanto, a abordagem proposta rejeita classificações potencialmente erradas, que podem ser observadas comparando a taxa de FP do kNN e a taxa de rejeição da abordagem proposta.

Considerando a abordagem tradicional de aprendizagem de fluxo, foi possível notar que quando os eventos são reproduzidos exatamente na mesma ordem que são apresentados no conjunto de dados original (Tabela 1), o método é capaz de detectar apenas os ataques iniciais - quando a janela deslizante é quase totalmente preenchida com eventos normais. No entanto, à medida que a ocorrência de ataques aumenta, os eventos de ataque adicionais são classificados como *inliers* (normal). Tal propriedade ocorre devido à natureza adaptativa dos algoritmos de aprendizagem de fluxo, que permitem que um evento que foi inicialmente classificado como *outlier* (ataque), seja adicionado na janela deslizante. Permitindo assim que um ataque se torne um *inlier* ao longo do tempo, pervertendo o comportamento dos detectores de anomalias. Desta forma, os algoritmos tradicionais de aprendizagem de fluxo devem considerar essa propriedade no campo de detecção de intrusão, que é tratado neste trabalho, através do comportamento imutável (Seção 3.2).

4.4. Ambiente do adversário – Ataques exploratórios

Dois tipos de ataques exploratórios foram avaliados: a evasão tradicional e a exploração da janela deslizante.

4.4.1. Evasão tradicional

A evasão tradicional refere-se à detecção de ataques que possuem tipo diferente de comportamento para o ataque no qual o sistema foi treinado, no entanto produzem o mesmo resultado ao alvo. Por exemplo, ataques que objetivam gerar uma quantidade significativa de tráfego de rede na vítima, independentemente do protocolo considerado, e.g. inundação UDP, TCP ou ICMP. Desta forma, cada uma das abordagens consideradas também foi avaliada com ataques de inundação diferentes do qual o sistema foi treinado. A precisão obtida é exibida na Tabela 2.

Em relação a aprendizagem de máquina tradicional (kNN), o atacante é capaz de evadir o sistema ao mesmo tempo em que gera um ataque que produz o mesmo resultado no alvo. Quando o kNN foi utilizado, a possibilidade de evasão foi evidenciada para todos os ataques avaliados: SYNFlood, UDPFlood e ICMPFlood. Por exemplo, quando o sistema foi treinado com ataques SYNFlood, o invasor ainda é capaz de evadir o mecanismo de detecção através da geração de ataques ICMPFlood. Por outro lado, a abordagem proposta aceitou apenas os resultados das classificações em relação aos ataques que sistema foi treinado. A elevada taxa de rejeição e o aumento da confiabilidade ocorreram devido ao possível aumento da taxa de erro, no qual ocorreu devido à falta de decisão unanime entre os detectores de anomalia que realiza a rejeição das classes atribuídas aos eventos.

4.4.1. Exploração da Janela Deslizante

O segundo ataque exploratório avaliado é chamado de exploração da janela deslizante. Esse ataque visa avaliar a precisão do algoritmo de aprendizagem de fluxo tradicional de acordo com a ocorrência de ataques em sua janela deslizante. Conforme observado na Seção 4.2, o aumento na frequência de ataques na janela deslizante torna o algoritmo de aprendizagem de fluxo não confiável. A Figura 3 (gráfico inferior) exibe a taxa de erro em relação a cada um dos ataques avaliados, durante os 8 a 9 milhões de pacotes no conjunto de dados criado. A taxa de erro é avaliada em um intervalo de 1.000 pacotes.

É possível observar que durante a detecção de eventos normais, a taxa de erro do algoritmo de aprendizagem de fluxo tradicional continua a ser semelhante à taxa obtida durante a avaliação tradicional (seção 4.2, 0,81 por cento). No entanto, à medida que os ataques começam a ocorrer (em torno do pacote 8,2 milhões), a taxa de erro para a detecção dos ataques aumenta devido ao aumento da ocorrência dos ataques. Desta forma, o atacante é capaz de explorar a janela deslizante do algoritmo de aprendizagem de fluxo tradicional. Isso é possível aumentando a ocorrência do ataque (Figura 3, gráfico superior), fazendo com que os ataques sejam classificados como *inliers* (normais) devido ao aumento de frequência na janela deslizante.

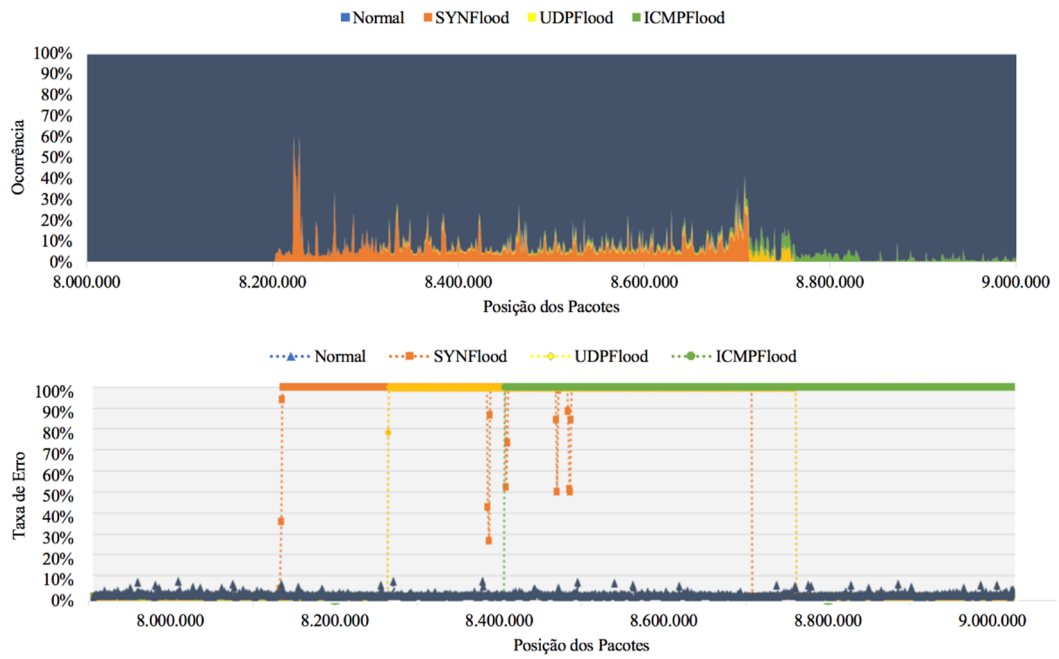

Figura 3. Comportamento da aprendizagem de fluxo tradicional durante a ocorrência de ataques de inundação; o gráfico superior exibe a ocorrência dos ataques; o gráfico inferior exibe a taxa de erro durante a ocorrência dos ataques.

A exploração da janela deslizante não ocorre na abordagem proposta devido ao comportamento imutável (Seção 3.2) e ao mecanismo de detecção de anomalia específico a classe (Seção 3.1). Os resultados são mostrados na Tabela 2. O invasor não é capaz de adicionar ataques na janela deslizante do detector de anomalia da classe normal devido ao comportamento imutável. Embora, se o mecanismo de detecção classificar erroneamente um evento e, assim, adicioná-lo em sua janela deslizante, o evento será rejeitado. Isso ocorre porque não será possível estabelecer uma unanimidade entre os outros detectores de anomalias.

4.4. Ambiente do adversário – Ataques Causais

Finalmente, para avaliar a resistência a ataques causais, foi adotada uma abordagem de manipulação da base de dados de treinamento. A aprendizagem de máquina tradicional e a abordagem proposta foram avaliados quanto à influência que os ataques que foram inicialmente injetados no conjunto de dados de treinamento como eventos normais, têm sobre a precisão final dos algoritmos. Assim, o objetivo foi avaliar cada um dos métodos considerados em relação à sua resiliência aos ataques de manipulação da base de dados de treinamento. A Figura 4 exibe a relação entre a taxa de detecção de ataques e o percentual de controle do atacante sobre o conjunto de dados de treinamento, enquanto injeta com sucesso ataques erroneamente classificados como eventos normais.

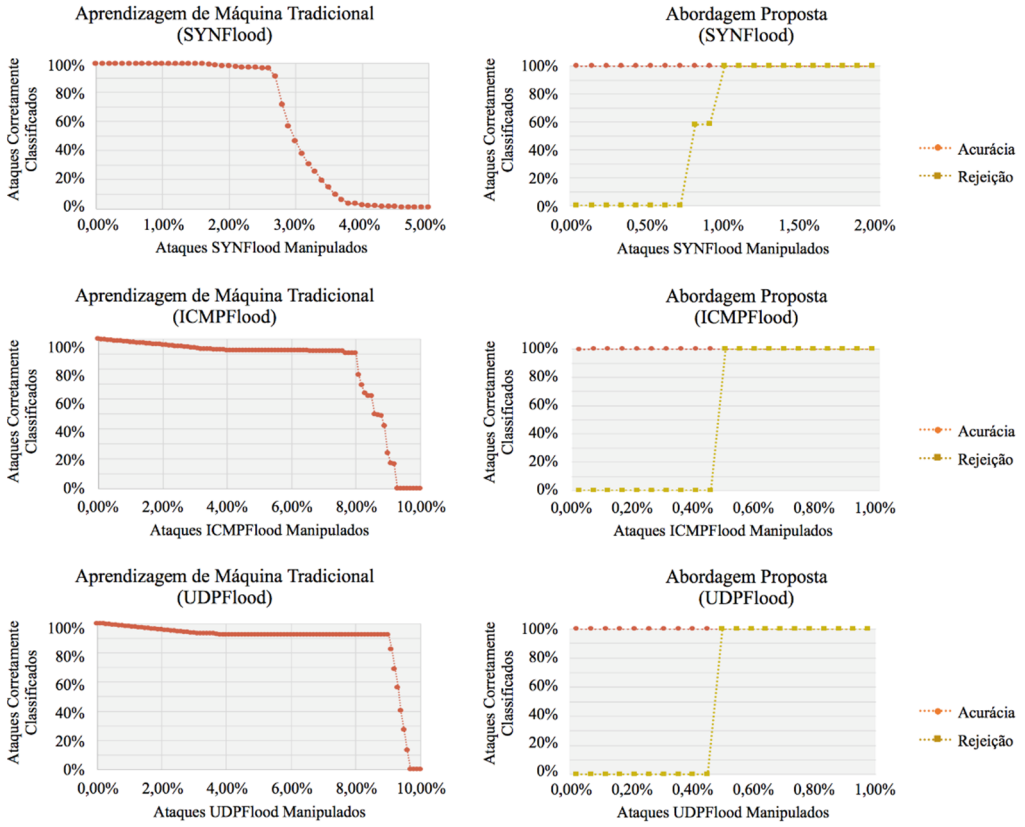

Figura 4. Avaliação da resiliência a ataques causais da abordagem de aprendizagem de máquina tradicional e da abordagem proposta.

Em relação a aprendizagem de máquina tradicional, é possível observar que para os três ataques avaliados, a evasão do mecanismo de detecção é possível. A taxa de precisão dos ataques de SYNFlood caiu para 50% quando apenas 3% dos eventos de ataques foram injetados como eventos normais na base de treinamento. Enquanto, para ICMPFlood e UDPFlood, o atacante poderia evadir o mecanismo de detecção quando apenas 9% dos ataques fossem manipulados. Por outro lado, a abordagem proposta poderia detectar quando os ataques foram manipulados no conjunto de dados de treinamento e rejeitar as classificações adicionais. Tal característica ocorre devido a utilização de detectores de anomalias específicos a classe. Assim, os ataques injetados no conjunto de dados de treinamento como eventos normais resultam na falta de unanimidade durante o processo de classificação, implicando que os eventos sejam rejeitados.

5. Trabalhos Relacionados

A baixa utilização de métodos de detecção de intrusão baseados em anomalias em ambientes de produção foi observada nos últimos anos por vários trabalhos [Sommer, R. e Paxson, V. 2012]. Essa lacuna pode ser causada por vários aspectos. No entanto, existe um consenso que o método de detecção deve ser pelo menos confiável e de fácil atualização [Sommer, R. e Paxson, V. 2012].

A confiabilidade da detecção é muitas vezes considerada em outras áreas [Oliveira, L. S. et al. 2002]. Para esse fim, em geral, os autores [Cavalcanti, G. D. C. et al. 2016] utilizaram a probabilidade da classe de saída para rejeitar ou não as decisões. Outras abordagens utilizam um conjunto de classificadores para estabelecer a confiabilidade de classificação através de uma abordagem de votação por maioria [Nelson, B. et al. 2008]. Apesar de ser muitas vezes considerado em outras áreas, até o melhor de nosso conhecimento, a confiabilidade da classificação ainda não foi considerada no campo de aprendizagem de fluxo. Alguns autores, no entanto, consideraram o ambiente do adversário na detecção de intrusão baseada em anomalia.

Tygar e seus colegas [Tygar, J. D. 2009] definiram uma taxonomia utilizada em seu trabalho para classificar possíveis ataques adversários contra o sistema de aprendizagem de máquina. Os autores também avaliaram o impacto que um conjunto de dados de treinamento manipulado incorrem na precisão final do classificador. Em todos os casos avaliados, o classificador tornou-se pouco confiável quando o conjunto de dados de treinamento teve ataques incorretos injetados.

No cenário de detecção de spam, Blaine Nelson e seus colegas [Nelson, B. et al. 2008] avaliaram o impacto da manipulação do conjunto de dados de treinamento na acurácia do sistema. Os autores relataram um aumento na taxa de erro da classificação de até 36% quando o atacante tem controle de apenas 1% do conjunto de dados de treinamento. Os autores também avaliaram uma abordagem de resistência a ataques causais, através da identificação de melhorias na acurácia, quando uma nova instancia era adicionada a base de treinamento. Apesar dessa abordagem ser efetiva, os autores utilizaram um conjunto de dados supervisionado (quando todas as instâncias são previamente classificadas). Essa abordagem não pode ser utilizada em produção, pois as instâncias não são rotuladas previamente e a precisão não pode ser estimada em tempo real. No cenário de detecção de PDF malicioso, Srndic e Laskov [N. Srndic e Laskov, P. 2014] avaliaram um conjunto de ataques contra uma conhecida ferramenta de classificação de PDF maliciosos. Os autores conseguiram reduzir a precisão da classificação de quase 100% para 28%. Os autores também sugeriram que um sistema de classificação de múltiplos classificadores deveria fornecer mais resiliência a tais ataques de adversários, devido à necessidade de evadir vários classificadores complementares.

Poucos autores abordam ataques causais no campo de detecção de intrusão de rede [Wang, G., et al. 2014]. Joseph e seus colegas [Joseph, A. D. e Taft, N. 2009] desenvolveram o ANTIDOTE, que se baseia em um PCA robusto e um limiar de Laplace que é menos susceptível aos ataques de manipulação. No entanto, sua abordagem permanece exposta a ataques exploratórios.

Para o melhor de nosso conhecimento, este é o primeiro trabalho a abordar ataques causais e exploratórios em algoritmos de aprendizagem de fluxo para o campo de detecção de intrusão. Nossa abordagem permanece confiável durante ambos os ataques,

causal e exploratório através de um mecanismo de rejeição e detectores de anomalias específicos a classe.

6. Conclusão

A detecção de intrusão baseada em anomalia tem sido amplamente estudada nos últimos anos. No entanto, apesar de resultados promissores, essa abordagem é pouco utilizada em ambientes de produção. Isso ocorre principalmente devido à ausência de métodos de detecção confiáveis e atualizáveis. O principal problema encontrado é o uso da aprendizagem de máquina em configurações adversárias, em que um atacante tenta evadir o mecanismo de detecção.

Este artigo apresentou um novo método de detecção de intrusão baseado em anomalia, que aborda o uso de aprendizagem de fluxo em configurações adversárias. A abordagem proposta utiliza detectores de anomalias específico a classe. Permitindo assim, a detecção de possíveis tentativas de evasão, e ao mesmo tempo que fornece um mecanismo de detecção confiável e atualizável. A confiabilidade é alcançada rejeitando classificações potencialmente erradas ou tentativas de evasão. Através do conjunto de avaliações efetuadas, o método proposto apresentou sua resistência para ataques causais e exploratórios.

Para trabalhos futuros, busca-se a redução da taxa de rejeição, enquanto ainda resistimos aos ataques adversários. Para este fim, planeja-se empregar uma abordagem híbrida, que se baseie em algoritmos de aprendizagem de fluxo e algoritmos de aprendizagem de máquinas tradicionais.

Referências

Alsabti, K., Ranka, S. e Singh, V. (1997). An efficient k-means clustering algorithm. Electical engineering and Computer Science, v. 43 p. 2-7.

Axelsson, S. (2000). The Base-Rate Fallacy and the Difficulty of Intrusion Detection. ACM Transactions on Information and System Security, v. 3, n. 3, p. 186–205.

Bifet, A., Gavalda, R. e Gavaldà, R. (2007). Learning from Time-Changing Data with Adaptive Windowing. Sdm, v. 7, p. 2007.

Cavalcanti, G. D. C., Oliveira, L. S., Moura, T. J. M. e Carvalho, G. V. (2016). Combining diversity measures for ensemble pruning. Pattern Recognition Letters, v. 74, p. 38–45.

Corona, I., Giacinto, G. e Roli, F. (2013). Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues. Information Sciences, v. 239, p. 201–225.

Denning, D. E. (1987). An intrusion-detection model. Proceedings - IEEE Symposium on Security and Privacy, n. 2, p. 118–131.

García-Teodoro, P., Díaz-Verdejo, J., Maciá-Ferná Ndez, G. e Vá Zquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, v. 28, p. 18–28.

He, H., Chen, S., Li, K. e Xu, X. (2011). Incremental learning from stream data. IEEE Transactions on Neural Networks, v. 22, n. 12, p. 1901–14.

Joseph, A. D. e Taft, N. (2009). ANTIDOTE: Understanding and Defending against. Traffic, SIGCOMM, p. 1–14.

Jyothsna, V., V Rama Prasad, V. e Munivara Prasad, K. (2011). A Review of Anomaly based Intrusion Detection Systems. International Journal of Computer Applications, v. 28, n. 7, p. 26–35.

Kontaki, M., Gounaris, A., Papadopoulos, A. N., Tsichlas, K. e Manolopoulos, Y. (2011). Continuous Monitoring of Distance-Based Outliers over Data Streams.

Maggi, F., Robertson, W., Kruegel, C. e Vigna, G. (2009). Protecting a moving target: Addressing web application concept drift. Lecture Notes in Computer Science, v. 5758 LNCS, p. 21–40.

Nelson, B., Barreno, M., Chi, F. J., et al. (2008). Exploiting machine learning to subvert your spam filter. In Proceedings of the First Workshop on Large-scale Exploits and Emerging Threats (LEET), n. April, p. Article 7.

N. Srndic e Laskov, P. (2014). Practical Evasion of a Learning-Based Classifier: A Case Study, IEEE Symposium Security and Privacy, p. 197-211.

Oliveira, L. S., Sabourin, R., Bortolozzi, F. e Suen, C. Y. (2002). Automatic recognition of handwritten numerical strings: A Recognition and Verification strategy. IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 24, n. 11, p. 1438–1454.

Sommer, R. e Paxson, V. (2010). Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. 2010 IEEE Symposium on Security and Privacy, v. 0, n. May, p. 305–316.

Tygar, J. D. (2011). Adversarial machine learning. IEEE Internet Computing, v. 15, n. 5, p. 4–6.

Viegas, E. K., Santin, A. O. e Oliveira, L. S. (2017) Toward a reliable anomaly-based intrusion detection in real-world environments. Computer Networks, v. 127, p. 200-216.

Wang, G., Barbara, S., Wang, T., Zheng, H. e Zhao, B. Y. (2014). Man vs. Machine: Practical Adversarial Detection of Malicious Crowdsourcing Workers. the 23rd USENIX Security Symposium, p. 239–254.

