UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de São José do Rio Preto

Leandro Bertini Lara Gonçalves

Abordagem para Geração Automática de Assinatura de Ataques baseada em Fluxos de Redes de Computadores

São José do Rio Preto
2019

Leandro Bertini Lara Gonçalves

Abordagem para Geração Automática de Assinatura de Ataques baseada em Fluxos de Redes de Computadores

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.

Financiadora: CAPES

Orientador:

Prof. Dr. Adriano Mauro Cansian

São José do Rio Preto
2019

G635a

Gonçalves, Leandro Bertini Lara
Abordagem para Geração Automática de Assinatura de Ataques baseada em Fluxos de Redes de Computadores / Leandro Bertini Lara Gonçalves. -- São José do Rio Preto, 2019
74 f. : il., tabs.

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto
Orientador: Adriano Mauro Cansian

1. Ciência da computação. 2. Computadores Medidas de segurança. 3. Crime por computador. 4. Reconhecimento de padrões. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Leandro Bertini Lara Gonçalves

Abordagem para Geração Automática de Assinatura de Ataques baseada em Fluxos de Redes de Computadores

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.

Financiadora: CAPES

Comissão Examinadora

Prof. Dr. Adriano Mauro Cansian
Unesp - Câmpus São José do Rio Preto
Orientador

Prof. Dr. Geraldo Francisco Donegá Zafalon
Unesp - Câmpus São José do Rio Preto

Prof. Dr. Robson de Oliveira Albuquerque
UnB - Universidade de Brasília

São José do Rio Preto
20 de Fevereiro de 2019

Agradecimentos

Agradeço, primeiramente, ao Laboratório ACME! pois sem ele não haveria a possibilidade de realizar o estudo nesta monografia descrito. Além disso, agradeço aos integrantes desse laboratório, em especial aos membros Raphael Campos Silva, Rafael Stefanini Carreira, Vinícius Oliveira Ferreira, Vinícius Vassoler Galhardi, Amanda Barbosa Sobrinho, Bruno Ferreira Leal e Pedro Ferracini de Barros por toda a ajuda e dúvidas tiradas durante as pesquisas e testes e pelos laços de amizades ali construídos. Assim como agradeço ao meu orientador, Prof. Dr. Adriano Mauro Cansian, pela paciência no processo e por dividir comigo seus conhecimentos e experiências.

Agradeço à minha namorada por todo o apoio incondicional dedicado, aos meus amigos e familiares pelo apoio constante dedicado à mim e às minhas ideias.

Em especial, agradeço meu amigo Matheus Gonçalves Ribeiro pela amizade, companhia e apoio, dividindo muitas experiências e conversas desde os primeiros dias da graduação. Agradeço aos meus amigos Guilherme Freire Roberto e Matheus Carreira Andrade por todo o apoio e suporte a mim sempre prontamente oferecidos, pelas ótimas conversas e pelo companheirismo.

Agradeço à Sra. Olga Maria Rissi Ferreira por todo o incentivo e pelos sempre excelentes conselhos que me motivaram durante o curso. Agradeço à minha amiga Adriana Felix Roberto Ártico pelo constante apoio, incentivo e conselhos desde o início da minha graduação.

Agradeço, também, à Universidade Estadual Paulista "Júlio de Mesquita Filho", campus de São José do Rio Preto, pelos cursos de graduação e de pós-graduação ali ministrados por professores de excelência.

O presente trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, à qual agradeço.

"Dietro ogni problema c’è un’opportunità."

-Galilei, Galileo

Resumo

Este trabalho apresenta um método para a geração automática de assinaturas de ataques em redes de computadores a partir de fluxos de redes. Nessa abordagem, utiliza-se um modelo de tráfego legítimo para a comparação dos dados e identificação dos elementos significativos. Esse modelo é composto de duas partes: uma formada por *clusters* e funções de distribuição acumuladas e outra formada por tráfegos pré-processados. A partir desse modelo composto, dispensa-se a necessidade do fornecimento de dados de múltiplas ocorrências do mesmo evento para a análise e extração de características. O sistema foi testado, quanto a sua capacidade de geração de assinaturas, para seis diferentes ataques e para tráfegos legítimos. O melhor resultado de assinatura gerado obteve F1-*Score* de 0.9801, o que é considerado bom pela literatura. Os tempos de geração de assinaturas para tráfegos maliciosos foram, em sua maioria, considerados rápidos, permanecendo abaixo de um minuto.

Palavras-chave: fluxos de rede, geração de assinaturas, segurança de redes.

Abstract

In this work we present a method for automatic signature generation of network malicious activity based in its network flow data. In this approach, we proposed a model of legitimate data for data comparison and significant features identification. Such model is composed of two parts: one formed by clusters and cumulative distribution functions, and another formed by preprocessed data. With this model, there is no need to provide data of multiple occurrences of the same event so the characteristics can be extracted. The system was tested for six different attacks and for legitimate traffic, so we could obtain its signature generation capacity. The best signature result has its F1-Score equal to 0.9801, which is good according to the literature. The signature generation time for malicious traffic was, in its majority, considered fast enough, remaining below one minute.

Keywords: *network flows, network security, signature generation.*

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

APT *Advanced Persistent Threats*

BIC *Bayesian Information Criterion* (Critério de Informação Bayesiano)

CERT.Br Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

DDoS *Distributed Denial of Service* (Negação de Serviço Distribuída)

DoS *Denial of Service* (Negação de Serviço)

E Especificidade

FDA Função de Distribuição de Probabilidade

FDP Função de Densidade de Probabilidade

FN Falso Negativo

FP Falso Postivo

HTTP *HyperText Transfer Protocol*

IDS *Intrusion Detection System* (Sistemas de Detecção de Intrusão)

IETF *Internet Engineering Task Force*

IP *Internet Protocol*

JSON *Javascript Object Notation*

NAT *Network Address Translator*

NIDS *Network Intrusion Detection System* (Sistema de Detecção de Intrução em Redes)

S Sensibilidade

SGBD Sistema Gerenciador de Banco de Dados

TFN Taxa de Falso Negativo

TFP Taxa de Falso Positivo

TVN Taxa de Verdadeiro Negativo

TVP Taxa de Verdadeiro Positivo

VN Verdadeiro Negativo

VP Verdadeiro Positivo

XSS *Cross Site Scripting*

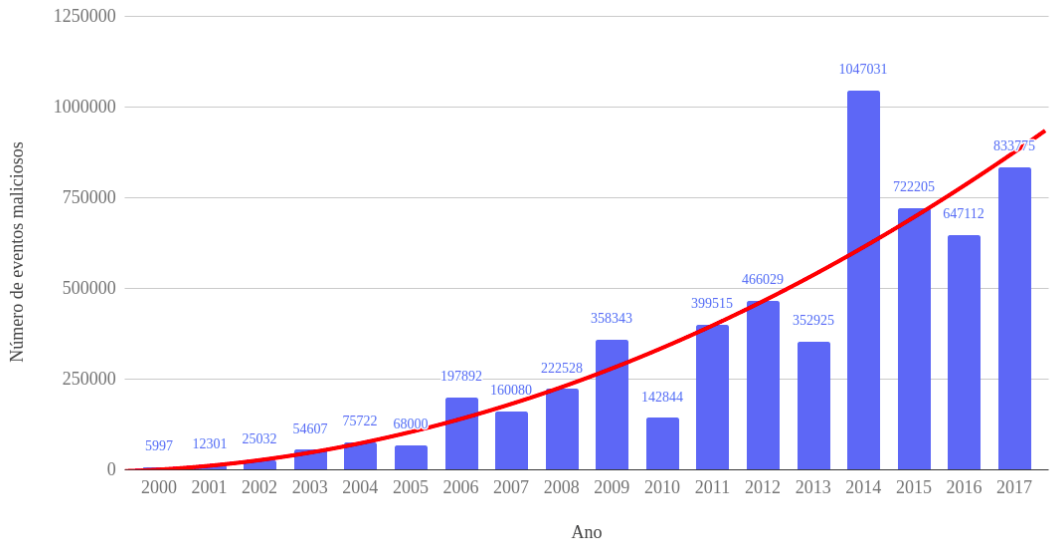
Sumário

Capítulo 1

Introdução

Eventos de segurança da informação têm se mantido como uma tendência de alta na última década, como pode ser observado no gráfico de incidentes reportados para o CERT.Br (2019), na Figura 1.1. O aumento desses incidentes de segurança requer maior quantidade de mão de obra para seu tratamento (GARCIA-TEODORO et al., 2015), o que implica em uma elevação de custos operacionais e exige uma significativa quantidade de tempo para se lidar com carga de trabalho (AFEK; BREMLER-BARR; FEIBISH, 2013).

Figura 1.1: Números de ataques reportados para o CERT.Br por ano.



Fonte: Adaptado de CERT.Br (2019).

Assim, a automatização de atividades de segurança se apresenta como uma solução atrativa para enfrentar a elevação do número de ocorrência de atividades maliciosas.

Dentre as funções passíveis de automatização pode-se incluir a geração de assinaturas para eventos maliciosos em redes de computadores.

Assinaturas para incidentes de segurança são padrões que podem ser utilizados para a identificação de eventos maliciosos. Esses padrões podem ser obtidos pela análise de dados de dois tipos: maliciosos, provenientes de ataques que possibilitam a geração de assinaturas de ataque, e dados provenientes de abusos de redes, que possibilitam a geração de assinaturas de abuso (CORRêA, 2009). Este trabalho concentra-se na geração de assinaturas desse último tipo.

A análise de dados para a identificação dos padrões para a formação de uma assinatura é um processo que demanda tempo e requer indivíduos altamente especializados. Por via de regra, pode-se dizer que a qualidade de uma assinatura pode variar de acordo com a proficiência do profissional que a gerou (SHIM et al., 2017). Também, a tendência crescente de ataques, como observado em CERT.Br (2019), e o aumento da diversidade de ataques dificultam a aplicação de políticas e controle de redes, uma vez que indivíduos maliciosos constantemente buscam passar pelas medidas de proteção implementadas em seus sistemas ou serviços alvos. Nesse cenário, a redução do tempo de resposta ou a automatização do processo de identificação de aspectos relevantes em atividades maliciosas podem trazer um diferencial para a aplicação de contramedidas ante os diversos eventos maliciosos.

1.1 Motivação e Justificativas

No tratamento de atividades maliciosas, variações de ataques a redes de computadores impõem um significativo problema de segurança e exigem a solução ativa de profissionais especializados, o que acarreta custos e pode restringir a capacidade de aplicação de práticas de segurança de qualidade.

Neste tocante, a geração automática de assinaturas torna uma parte do processo de aplicação de contramedidas mais simples para os profissionais da área. Aumentando a eficiência do combate a eventos maliciosos no geral. Para o tratamento de *malwares*, diversos estudos sobre geração automática de assinaturas foram desenvolvidos, resultando em métodos capazes de identificar e classificar comportamentos nocivos realizados por esses *softwares* maliciosos, como Kreibich e Crowcroft (2004), David e Netanyahu (2015) e Szynkiewicz e Kozakiewicz (2017). Em contrapartida, para ataques a redes de computadores, comparativamente, poucos estudos foram desenvolvidos sobre a geração e classificação automática de eventos maliciosos. Com isso, este trabalho tem como intuito preencher parte deste espaço, buscando contribuir para uma

melhor aplicação de segurança em redes de computadores.

Diversos trabalhos observados na literatura têm o propósito de identificar padrões em dados de ataques previamente conhecidos, dentre esses Afek, Bremler-Barr e Feibish (2013), Fallahi, Sami e Tajbakhsh (2016). Esses trabalhos encontram invariâncias entre os tráfegos de ataques e as utiliza como assinaturas. Contudo, ataques a redes de computadores podem ter trechos que não são representados por invariâncias, mas sim por intervalos. O uso desses intervalos aumenta a representação da assinatura, mas requer outros métodos de identificação.

Neste trabalho, estudou-se a possibilidade de se aplicar um modelo de tráfegos legítimos para a seleção de atributos característicos de tráfegos maliciosos. Este modelo é composto de duas partes: um modelo de *clusters* e Funções de Distribuição Acumuladas e uma base de dados de comparação.

Uma vez que métodos de clusterização foram bem sucedidamente utilizados para o agrupamento de fluxos de rede nos trabalhos Ferreira (2016), Galhardi (2017), Ravale, Marathe e Padiya (2015), espera-se que sejam capazes de formar um modelo confiável para a extração de padrões e posteriormente probabilidades de ocorrência para ponderar a relevância dos atributos analisados.

Assim, como tráfegos maliciosos tendem a apresentar características que os distinguem dos demais tráfegos idôneos em um dado momento, é possível a identificação e o isolamento dessas características para a identificação de demais tráfegos maliciosos semelhantes. Isso pode ser feito em duas etapas: a primeira consiste em identificar atributos invariantes no tráfego malicioso, a segunda é feita com a ordenação por discrepância dos atributos ante a tráfegos legítimo e sua posterior seleção por filtragem de dados.

1.2 Objetivos

Este trabalho teve como objetivo o desenvolvimento e a análise de uma abordagem para a geração de assinaturas de atividades maliciosas pela análise dos fluxos de tráfegos de rede considerados abusivos. Como requisito, essa geração deve ser feita em menos tempo do que manualmente e com F1-*Score* representando uma boa qualidade de assinaturas de acordo com a literatura, ou seja, acima de 0.9.

Como objetivos específicos desenvolvidos durante a execução deste trabalho, pode-se considerar: *a*) estudo sobre o processo de extração de assinaturas a partir de fluxos maliciosos; e *b*) o desenvolvimento de um método rápido para a análise e seleção de dados maliciosos em fluxos de redes.

1.3 Contribuições Obtidas

Com o processo de desenvolvimento deste estudo sobre métodos de geração automáticas de assinaturas para ataques de redes de computadores foram feitas as seguintes contribuições:

1. Identificação do impacto da quantidade de tráfegos legítimos para o treinamento do modelo de dados para a extração de assinaturas; e
2. Desenvolvimento de uma abordagem ágil para a extração de características e formação de assinaturas de ataques à redes de computadores baseado em fluxos.

1.4 Organização do Trabalho

Este trabalho está organizado da seguinte forma: O Capítulo 2 expõe a fundamentação teórica e o levantamento bibliográfico utilizado neste trabalho, o Capítulo 3 apresenta a metodologia utilizada na proposta de solução do problema, o Capítulo 4 apresenta os resultados da abordagem utilizada, o Capítulo 5 apresenta as conclusões do trabalho e é seguido das referências bibliográficas.

Capítulo 2

Fundamentação Teórica

Este capítulo tem como finalidade expor os conceitos utilizados neste trabalho, apresentando suas características e sua relevância neste estudo, assim como descrever os trabalhos mais relevantes presentes na literatura sobre geração automática de assinaturas de ataques em redes de computadores, apresentando suas abordagens e seus contextos de aplicação. Primeiro, é apresentado o conceito de fluxos de rede na seção 2.1, na seção 2.2 são apresentadas atividades maliciosas em redes de computadores, na seção 2.3 é introduzido o conceito de assinaturas de ataques, na seção 2.4 são apresentados métodos de clusterização, na seção 2.6 exibidos métodos de classificação, na seção 2.7 são exibidas as métricas de avaliação utilizadas nesse trabalho, na seção 2.8 são apresentados os trabalhos sobre geração automática de assinaturas e na seção 2.9 são apresentadas as conclusões parciais deste capítulo.

2.1 Fluxos de Rede

Fluxos de Redes de Computadores foram inicialmente pesquisados por volta de 1991 para contabilidade de uso da rede, pelo *Internet Accounting Working Group* - iniciativa do *Internet Engineering Task Force* (IETF), entidade internacional responsável pela evolução e desenvolvimento da internet. O primeiro protocolo de exportação de fluxos aceito pelo mercado foi o *Netflow*, desenvolvido pela *Cisco Systems* em meados de 1996 como um protocolo proprietário. Em 2013, o IETF, por meio do *IP Flow Information Export Working Group*, definiu o protocolo de exportação IPFIX, aberto e disponível para a comunidade de desenvolvimento e científica (HOFSTEDE et al., 2014).

Tais fluxos são conceitualmente caracterizados como “uma sequência uni ou bidirecional de pacotes entre dois usuários finais, ou hosts, com atributos comuns” (LI

et al., 2013). Exemplos desses atributos seriam: tempos inicial e final, endereços IP de origem e destino, portas de transporte de origem e destino, protocolos de camada de roteamento e de transporte, tipo de serviço, bytes transferidos no fluxo e interface lógica de entrada.

A obtenção dos fluxos é realizada a partir de um ponto de observação, que pode ser qualquer *switch* ou roteador instalado com capacidade de exportação de fluxos, fornecendo a perspectiva desse ponto do comportamento do tráfego, ou a partir da extração com base em arquivos de tráfego bruto (pcap).

Com base nesses dados de fluxo, é possível realizar contabilidade, análise e classificação do tráfego, planejamento de capacidade, cobrança, monitoramento, e análise de segurança em uma rede, como mencionado em Li et al. (2013).

Neste trabalho, fluxos de rede são utilizados como fonte de informações sobre os tráfegos de dados legítimos e maliciosos utilizados e também, servem como base para os atributos e parâmetros que compõem as assinaturas.

2.2 Atividades maliciosas em redes de computadores

As atividades maliciosas em redes de computadores podem ser identificadas pelos tráfegos gerados em decorrência de um ato de ataque ou comprometimento de sistema computacional visando o acesso não autorizado, coleta indevida de informações ou a disrupção de sistemas e serviços (WU; BANZHAF, 2010). O tráfego gerado por uma atividade maliciosa pode diferir de tráfegos gerados por atividades legítimas (aquelas geradas pelo uso normal de computadores e redes) (LIAO et al., 2013).

Os tipos de atividades maliciosas presentes em redes de computadores são amplamente heterogêneos, existindo ataques que causam amplo impacto e visibilidade pela interrupção do funcionamento de serviços e que são facilmente observáveis no tráfego de rede, assim como ataques que não deixam rastros aparentes no comportamento da rede, sendo semelhantes a uma navegação simples com um servidor HTTP, como um ataque de Injeção SQL, e que não apresentam necessariamente diferença de funcionamento do sistema - se o atacante estiver visando o roubo de informações, o ataque pode ser feito, em alguns casos, de modo transparente (HOQUE et al., 2014).

Não se limitando às diferenças de impacto e visibilidade, ataques diferenciam-se entre execuções de um mesmo tipo, podendo apresentar mais ou menos características que se diferenciam de um tráfego legítimo da rede. Em geral, um ataque de rede menos distinguível tem sua execução em um intervalo longo e é mascarado pela imprecisão de comportamento de tráfegos legítimos. Essa característica é utilizada pelas *Advanced*

Persistent Threats (MARCHETTI et al., 2016), conhecidas simplesmente como APTs, que não serão abordadas no escopo deste trabalho.

Para o desenvolvimento deste estudo, foram aplicadas algumas classes de ataques: Ataques de Força Bruta SSH, Ataques de Negação de Serviços, Varredura de Reconhecimento, Varredura de Vulnerabilidades, Injeção SQL, e *Cross Site Scripting* (XSS).

Ataques de Força Bruta SSH visam a identificação de credenciais para o acesso remoto em sistemas computacionais, isso é feito por sucessivas tentativas de autenticação alterando-se as credenciais de acesso, como identidade do usuário ou senha (SADASIVAM; HOTA; ANAND, 2018). Ataques de força bruta SSH podem, também, ser utilizados para identificar servidores SSH que aceitem uma autenticação específica (ABDOU; BARRERA; OORSCHOT, 2015), como as credenciais padrões de fábrica de dispositivos. Contudo, um ataque de força bruta SSH nesses moldes ganha aspectos de varredura de vulnerabilidade.

Ataques de Negação de Serviço (DoS, sigla em inglês) tem como finalidade a interrupção da funcionalidade de um ou mais serviços oferecidos pela vítima. A forma pela qual se atinge esse objetivo é diversa: pode ser feito pelo esgotamento de recursos (tipicamente memória ou processamento) no servidor que disponibiliza o serviço, pelo seu comprometimento via *software* com a exploração de eventuais vulnerabilidades presentes na versão em execução do programa, ou pela utilização de toda a banda de tráfego disponível para o servidor, impedindo que requisições legítimas o alcancem para que sejam atendidas (DESAI; HADULE; DUDHGAONKAR, 2017).

Varreduras de rede são caracterizadas pelo envio de pacotes arbitrários a alvos específicos com a finalidade de descoberta de informações que auxiliem o atacante na execução de um conjunto de atividades maliciosas. Varreduras podem ser divididas em dois grupos com finalidades distintas: varreduras de reconhecimento e varreduras de vulnerabilidades.

As varreduras de reconhecimento tem por finalidade encontrar a topologia de uma rede alvo. Por meio dessa varredura, pode-se descobrir quais são as máquinas disponíveis na rede alvo, seus endereços, os serviços que oferecem, e até em alguns casos, versões de sistemas operacionais e demais *softwares* em operação nos computadores vítimas (HEO; SHIN, 2018). Isso permite ao atacante o conhecimento de seus alvos diretos e indiretos para a execução do seu objetivo.

A varredura de vulnerabilidades, por sua vez, tem como finalidade a descoberta de brechas presentes nos sistemas em execução na rede alvo. Diferentemente da varredura de reconhecimento, os alvos da varredura de vulnerabilidades são muito bem definidos e o que varia são os tipos de pacotes enviados a esses alvos. Em geral, os

pacotes enviados ao alvo na varredura de vulnerabilidades têm conteúdos arbitrários que interagem com um serviço esperando uma resposta específica e conhecida caso ele tenha a falha procurada (RAHMAN et al., 2016). Esses dois tipos de varredura permitem, ao atacante, a descoberta da topologia e do estado de segurança da rede, auxiliando-o na decisão de qual abordagem utilizar para realizar seu ataque. Como exemplo, diversos *malwares* se utilizam de varreduras de rede no seu processo de propagação automática (AHMAD; WOODHEAD; GAN, 2016).

Ataques de Injeção SQL têm como finalidade a execução de código arbitrário em um Sistema Gerenciador de Banco de Dados (SGBD) ao qual o atacante não tem acesso ou autorização para a manipulação e visualização dos dados almejados. Esse ataque consiste em adicionar comandos SQL em um formulário cujas informações vão ser lidas para comparação ou armazenadas no banco de dados (ALWAN; YOUNIS, 2017). Em um ataque bem sucedido, o atacante é capaz de consultar, alterar, criar e destruir dados do banco sem que haja muita atividade suspeita, podendo o ataque nunca ser descoberto. Diversos aspectos das linguagens *web* possibilitam a execução de ataques de Injeção SQL, como a carência de distinção entre tipos de dados aceitos como entrada da aplicação *web*; a possibilidade que números sejam usados como string; e a falta de sanitização da entrada, não conferindo dados recebidos adequadamente (KINDY; PATHAN, 2011).

O ataque XSS tem como finalidade a execução de código arbitrário no navegador da vítima pela infecção de uma página *web* que ambos têm acesso. Isso pode ser feito pela inserção de código em textos da página, como comentários ou dados carregados de banco para compor formulários que, quando carregados pela vítima, são executados pelo seu navegador (JOHARI; SHARMA, 2012). Com isso, um atacante pode confundir e enganar a vítima para que esta lhe entregue informações, ou para que execute ações que lhe dê acesso ou mais autonomia no computador alvo. XSS é considerada uma ameaça importante por que aplicações *web* são parte intrínseca do cotidiano e da sociedade. Dessas aplicações *web*, inúmeras podem estar vulneráveis com a capacidade de afetar um número significativo de pessoas(NITHYA; PANDIAN; MALARVIZHI, 2015).

Esses ataques foram escolhidos pela sua diversidade de atuação e pelos seus diferentes impactos em redes de computadores. Os ataques de Força Bruta SSH, Negação de Serviço e Varreduras causam maior quantidade de tráfego e geram um maior número de conexões que o tráfego normal, o que acarreta em um maior ruído de rede. Em contrapartida, os ataques de Injeção SQL e XSS não causam muitas conexões nem geram muito tráfego, sendo, então, mais discretos por gerarem menos ruídos de rede. A

quantidade de ruído é relevante para este trabalho por que pode interferir diretamente na capacidade de distinção e seleção de atributos do sistema gerador de assinaturas, viabilizando ou não seu adequado funcionamento.

2.3 Assinaturas de Ataques

Assinaturas podem ser definidas como padrões para a identificação de um determinado evento (SHIM et al., 2017), não necessariamente malicioso ou legítimo. Em segurança de computadores e redes, utiliza-se frequentemente o conceito de assinatura de ataques. Esse tipo de assinatura tem como propósito discriminar características relevantes de atividades maliciosas de modo a identificar um evento nocivo no ambiente de rede.

Em geral, Sistemas de Detecção de Intrusão (IDS, sigla em inglês) baseados em assinatura utilizam o conceito assinaturas de ataques na proteção de redes de computadores (JAIGANESH; MANGAYARKARASI; SUMATHI, 2013). Para esses sistemas, é fornecido um banco de dados de padrões que é comparado com o tráfego da rede. Quando um padrão é satisfeito, esse tráfego é considerado malicioso e é emitido um alerta (INAYAT et al., 2016).

As assinaturas tratadas neste trabalho são discretamente diferentes. Elas são geradas pela análise de um comportamento anômalo no tráfego de redes e, portanto, recebem o nome de assinaturas de abuso. Assinaturas de abuso diferem de assinaturas de ataque por serem menos precisas e não serem geradas de modo analítico com parâmetros definidos pela causa do ataque (CORRêA, 2009). Para assinaturas de ataque, a parametrização é feita a partir da política de segurança a qual a rede protegida está sujeita, permitindo total controle sobre a generalidade e precisão da assinatura. Por sua vez, assinaturas de abuso são parametrizadas conforme o tráfego malicioso analisado, possibilitando, deste modo, menor controle sobre seus parâmetros.

Assinaturas de abuso podem ser aplicadas em IDSs baseados em assinatura, desde que o administrador de rede esteja ciente de que o comportamento de resposta do IDS passa a ser discretamente diferente do esperado para as mesmas classes de ataques em decorrência das diferentes características das assinaturas por ataque e por abuso.

2.3.1 Geração automática de assinaturas

A geração automática de assinaturas foi inicialmente proposta por Kreibich e Crowcroft (2004) com o objetivo de alimentar o IDS Snort com assinaturas de rede, permitindo que ele fosse atualizado sem “que um trabalho tedioso e que exige conheci-

mento detalhado sobre vulnerabilidades e suas explorações específicas de cada *software*" (KREIBICH; CROWCROFT, 2004, p. 51, tradução minha) fosse requerido para a criação de assinaturas para combater agentes maliciosos. Nesse trabalho, os autores fizeram uso do algoritmo de Ukkonem (uma variação de algoritmo baseada em árvores de sufixos para a solução do problema de *substrings* comuns mais longas).

Diversos outras abordagens foram propostas desde a publicação de Kreibich e Crowcroft (2004), tanto para o desenvolvimento de assinaturas para *malwares*, quanto para assinaturas de ataques em redes de computadores. Dentre os desenvolvimentos apresentados, pode-se citar a integração de métodos de geração com a estrutura de *honeypots* para o fornecimento automático de dados maliciosos (ALTWAIJRY; SHAHBAR, 2013), (KREIBICH; CROWCROFT, 2004); abordagens capazes de identificar atributos relevantes a partir da análise de dados maliciosos (OCAMPO; CASTILLO; GOMEZ, 2013), (USSATH; CHENG; MEINEL, 2016), outras abordagens que requerem bases de dados de comportamento legítimo (AFEK; BREMLER-BARR; FEIBISH, 2013), (GARCIA-TEODORO et al., 2015); e abordagens com diversos formatos de dados de entrada como *logs* de eventos maliciosos (OCAMPO; CASTILLO; GOMEZ, 2013), (USSATH; CHENG; MEINEL, 2016); dados binários de tráfego de rede (GÓMEZ et al., 2013), (SHIM et al., 2017); e fluxos de rede (BARABAS; DROZD; HANACEK, 2012), (FALLAHI; SAMI; TAJBAKHSH, 2016)[R2.23]. Esse último formato de dados de entrada é o utilizado no trabalho aqui apresentado.

A principal dificuldade envolvida na geração automática de assinaturas é o adequado isolamento e identificação dos atributos significativos em um conjunto de dados. Esses atributos significativos devem representar o evento malicioso de modo satisfatoriamente preciso, evitando desse modo que dados idôneos sejam identificados incorretamente. Ao mesmo tempo, os atributos significativos que compõem uma assinatura devem ser genéricos o bastante para identificar o ataque dentro de uma margem tolerável de variações (WERNER et al., 2009).

2.4 Clusterização

Clusterização pode ser definida pelo agrupamento de elementos semelhantes em uma massa de dados para a descoberta de padrões relevantes. Em segurança de computadores e redes são utilizados para a criação de modelos de comportamentos de dados legítimos para IDSs baseados em abuso (CHANDOLA; BANERJEE; KUMAR, 2009), para a redução de falsos-positivos (FERREIRA, 2016), para descoberta de *botnets* (GU et al., 2008), além de diversas aplicações na segurança de redes sem fio e de

sensores (MEHMOOD; UMAR; SONG, 2017), (SUN et al., 2006).

Neste trabalho, utilizou-se clusterização para o agrupamento de tráfegos semelhantes em um conjunto legítimo para a criação de funções de probabilidade e no processamento de fluxos maliciosos para a identificação de características relevantes.

Existem diversos algoritmos de clusterização, cada qual apresenta características distintas e têm suas devidas aplicações. Métodos apresentam diferentes abordagens para clusterização, dentre elas pode-se citar os: Métodos hierárquicos; Partições rápidas, Métodos mistura e Métodos de soma de quadrados. Por vezes, algoritmos de classificação utilizam mais de uma abordagem para a realização de sua tarefa (WEBB, 2003).

Métodos hierárquicos organizam os elementos clusterizados com base em diferenças absolutas entre cada atributo presente, resultando em dendograma. Essa estrutura aninha *clusters* com características semelhantes dentro de *clusters* maiores. Partições rápidas são utilizadas como abordagem de definição de *clusters* iniciais para outros métodos e operam pela identificação simples de vetores de relevância, provendo um particionamento inicial. Métodos mistura definem cada grupo de vetores no conjunto com distribuições de probabilidades diferentes ou com somatórios de densidades de componentes, resultando em uma descrição da população como uma distribuição de misturas finitas (WEBB, 2003). No contexto aqui apresentado, o método mais relevante é o de soma de quadrados: nesse método, um critério de clusterização é maximizado baseando-se em matrizes de dispersão intra e interclasses. Em geral, devido à restrições de custo computacional, os algoritmos que utilizam soma de quadrados são subótimos e se distinguem, especialmente, pelo critério de otimização ou pelo método utilizado para executar esta otimização.

Na literatura, diversos algoritmos de clusterização são aplicados em segurança de computadores e redes, como o K-Means, o X-Means, o DBSCAN, o Autoclass, entre outros (HE et al., 2018), (FERREIRA, 2016), (MUDA et al., 2016). Neste trabalho, em observação aos resultados encontrados por Ferreira (2016) e Galhardi (2017), observou-se mais detalhadamente os algoritmos K-Means e X-Means. Em decorrência dos resultados desses trabalhos, os algoritmos Autoclass e DBSCAN não foram explorados mais a fundo neste momento e, portanto, sua exploração e aplicação para a finalidade de extração de características é considerada um trabalho futuro.

O algoritmo K-Means opera sobre dados não categorizados e de modo não supervisionado. Esse algoritmo tem como finalidade a distribuição dos dados fornecidos como entrada em K *clusters*, sendo o valor de K definido como um parâmetro inicial. O algoritmo começa encontrando K centroides, cada qual pertencente a um *cluster*.

Iterativamente, a partir desse centroides são construídas células por meio do algoritmo de Llyoid, uma vez que as células estão prontas novos centroides são calculados, e o ciclo se repete até que não haja alteração de centroide e nova definição de *clusters*. Sendo μ_k a média dos elementos x_i pertencentes ao *cluster k* objetivo do K-Means é a minimzação da soma dos erros quadrados dos *clusters*, dada pela Equação 2.1 (JAIN, 2010).

$$J(C) = \sum_{k=1}^{k} \sum_{X_i \in c_k} ||x_i - \mu_k|| \tag{2.1}$$

Por sua vez, o algoritmo X-Means é baseado no algoritmo K-Means, opera de modo não supervisionado e não requer prévia categorização dos dados. A principal diferença dos dois métodos é a de que o X-Means é capaz de identificar por conta própria o número de *clusters* (PELLEG; MOORE et al., 2000). Essa definição é feita por meio da maximização do valor do Critério de Informação Bayesiano (BIC, sigla em inglês), dado pela Equação 2.2, no qual $\hat{l}_j(D)$ é o ponto máximo da função de probabilidade dos dados de acordo com o $j-ésimo$ modelo, e P_j é o número de parâmetros de M_j e R é o números de observações nesse modelo.

$$BIC(M_j) = \hat{l}_j(D) - \frac{P_j}{2} \log R \tag{2.2}$$

2.5 Funções de Probabilidade

Foram utilizadas neste trabalho funções de probabilidade para se encontrar a relevâncias dos atributos na assinatura. Assim, foram aplicadas Estimativa de Densidade de Kernel, Função de Densidade de Probabilidade, e, por fim, Função de Distribuição Acumulada neste processo.

A Estimativa de Densidade de Kernel é um método para o cálculo de Funções de Densidade de Probabilidade (FDPs) de modo não paramétrico (SCOTT, 2012). Kernels tendem a representar FDPs simétricas, a presença de mais de um grupo de concentração de dados pode resultar na excessiva suavização da FDP (SILVERMAN, 2018).

As FDPs são funções que descrevem a probabilidade de uma variável aleatória ser encontrada em um dado intervalo da distribuição observada. Essa probabilidade pode ser encontrada pela diferença das integrais de menos infinito aos valores do intervalo analisado. Também, a área sob a curva da FDP resulta na Função de Distribuição Acumulada (FDA) (MEYER, 1970).

Por fim, as FDAs são funções que descrevem uma dada distribuição. Para o cálculo da probabilidade de uma variável aleatória pertencer a um dado intervalo, calcula-se a diferença dos valores da FDA para os extremos do intervalo dado (MEYER, 1970).

2.6 Identificação de Similaridades

Assinaturas podem ser classificadas entre diversos grupos de ataques. As assinaturas pertencentes a um tipo de ataque geralmente apresentam atributos comuns entre si (AFEK; BREMLER-BARR; FEIBISH, 2013). Portanto, para a identificação da pertinência de uma assinatura a uma determinada classe de ataque, encontrou-se a similaridade entre os conjuntos de atributos da assinatura recém gerada e de um conjunto de atributos de referência para cada ataque.

Não é do escopo deste trabalho o estudo de métodos de classificação para assinaturas de tráfego de rede. Assim, cada atributo de cada diferente posição hierárquica da assinatura foi considerado um atributo distinto, a fim de representar esses níveis. Desse modo, aplicou-se o índice de Jaccard (LEYDESDORFF, 2008) para a geração de um valor de pertinência de uma assinatura para cada classe de ataque de utilizada para testes neste trabalho.

O índice de Jaccard é utilizado para o cálculo de similaridades entre conjuntos de elementos. Quanto maior a similaridade entre os conjuntos comparados, maior o valor do índice, assim como quanto menor a similaridade, menor o índice, que varia entre zero e um (LEYDESDORFF, 2008).

Uma vez que os atributos que compõem as assinaturas são ponderados de acordo com sua probabilidade de serem maliciosos, incorporou-se este grau de imprecisão pelo uso do Índice de Jaccard Ponderado (IOFFE, 2010). A Equação 2.3 foi utilizada para o cálculo dos valores de pertinência para cada classe.

$$J(S_a,T_s) = \frac{||S_a|| + ||T_s|| - ||S_a - T_s||}{||S_a|| + ||T_s|| + ||S_a - T_s||} \tag{2.3}$$

Na equação 2.3, S_a é conjunto de atributos de referência para um dado ataque a, e T_s é o conjunto de atributos que compõem uma dada assinatura s que se busca encontrar os valores de pertinência.

Esse índice tem aplicações em visão computacional para a identificações de imagens similares (GUPTA; ARBELAEZ; MALIK, 2013), em processamento de imagens para compensação de segmentação de objetos (SHI; NGAN; LI, 2014), em biometria aplicado em modelos de autenticação por comportamento de usuário (OLEJNIK;

CASTELLUCCIA, 2013), em coleta de dados para o desenvolvimento de método consistente ponderado de amostragem (IOFFE, 2010), entre outras.

2.7 Métricas de Avaliação

As métricas comumente utilizadas para a avaliação de assinaturas são apresentadas a seguir, assim como os diversos trabalhos que as adotam como forma de avaliação de desempenho quanto a qualidade de detecção, como Afek, Bremler-Barr e Feibish (2013), e Ocampo, Castillo e Gomez (2013). Falsos Positivos (FP) e Falsos Negativos (FN) são termos atribuídos a erros de classificação. Falsos positivos ocorrem quando uma atividade legítima é classificada como maliciosa, FNs ocorrem quando uma atividade maliciosa é classificada como legítima. Esses erros, no âmbito de geração de assinaturas, representam muito sobre a qualidade da assinatura gerada, uma vez que a assinatura é o critério utilizado para a classificação de itens de um conjunto (no caso deste trabalho, fluxos de rede). Se uma assinatura tiver muitos erros de classificação, sejam eles FP, FN, ou baixo F1-*Score*, ela não é adequada para a implementação (FALLAHI; SAMI; TAJBAKHSH, 2016).

Verdadeiros Positivos (VP) são as atividades maliciosas corretamente classificadas, e Verdadeiros Negativos (VN) são as atividades legítimas adequadamente identificadas. Essas métricas são complementares às FP e FN. Elas são utilizadas para relações mais elaboradas, como taxa de verdadeiro positivo, taxa de verdadeiro negativo, taxa de falso positivos, taxa de falso negativos, precisão, e F1-*Score*.

A Taxa de Verdadeiro Negativo (TVN), dada pela Equação 2.4, conhecida também como especificidade (E), representa a proporção de negativos corretamente classificados dentre todos os negativos presentes no espaço amostral.

$$E = \frac{TN}{TN + FP} \tag{2.4}$$

Por sua vez, a Taxa de Verdadeiro Positivo (TVP), conhecida por sensibilidade (S), ou *recall*, é dada pela Equação 2.5 e representa a proporção de positivos corretamente identificados dentre todos os positivos do espaço amostral.

$$recall = \frac{TP}{TP + FN} \tag{2.5}$$

A Taxa de Falso Positivo (TFP) é dada pela Equação 2.6 ou por 1-TVN, representando a proporção de elementos negativos erroneamente classificados como positivos dentre todos os negativos do conjunto.

$$TFP = \frac{FP}{TN + FP} \tag{2.6}$$

Por fim, a Taxa de Falso Negativos (TFN) é dada pela Equação 2.7, ou 1-TFP, e significa a proporção dos elementos positivos erroneamente classificados como negativos dentre o conjunto de positivos do espaço amostral.

$$TFN = \frac{FN}{TP + FN} \tag{2.7}$$

A precisão é dada pela Equação 2.8, e o F1-*Score* é dado pela Equação 2.9. A precisão é utilizada para indicar a proporção de acertos da classificação pelo montante total de identificações positivas (corretas ou não) de um dado classificador. Uma alta precisão indica que poucos falsos positivos foram identificados, mas não diz respeito ao número de elementos que foram inadequadamente classificados como negativos. Essa relação é dada pelo *recall*.

$$precisão = \frac{TP}{TP + FP} \tag{2.8}$$

Por sua vez, o F1-*Score* é utilizado para indicar a qualidade da classificação para conjuntos de dados cujas classes têm grande disparidade de número de elementos (DAVIS; GOADRICH, 2006). Essa métrica relaciona precisão e *recall*, apresentando um índice qualitativo mais amplo do classificador estudado. Essa métrica foi utilizada por outros trabalhos, como em Fallahi, Sami e Tajbakhsh (2016), e Naidu, Whalley e Narayanan (2018).

$$F1 - Score = 2 * \frac{precisão * recall}{precisão + recall} \tag{2.9}$$

A partir dessas métricas, é possível quantificar o desempenho das assinaturas geradas neste trabalho, permitindo a análise de sua qualidade e da capacidade da abordagem apresentada.

2.8 Levantamento Bibliográfico

Diversos estudos foram desenvolvidos sobre a geração automática de assinaturas. Distinguem-se por diversos fatores, como, por exemplo, a forma de entrada de dados maliciosos, a necessidade de utilização de comparação com modelo de tráfego legítimo, os métodos e algoritmos utilizados para o isolamento e identificação das características relevantes para a formação da assinatura, entre outros que influenciam a aplicação

desses trabalhos em contextos distintos.

Vários estudos sobre geração automática de assinaturas buscam encontrar características relevantes para a formação das assinaturas por meio da comparação de vários dados maliciosos, identificando quais são suas características comuns (ALTWAIJRY; SHAHBAR, 2013), (USSATH; CHENG; MEINEL, 2016). Alguns outros utilizam abordagem distinta, encontrando características importantes para a composição da assinatura pela comparação do tráfego malicioso com informações extraídas e sumarizadas de tráfegos legítimos (AFEK; BREMLER-BARR; FEIBISH, 2013), (GARCIA-TEODORO et al., 2015). Esta diferença de perspectiva tem resultados importantes porque concentra o foco do objeto de estudo em diferentes frentes de análise. Para o primeiro caso, a análise dos atributos maliciosos em entradas de dados visa identificar semelhanças geralmente invariantes presentes nos dados maliciosos e, assim, produzir uma assinatura. Por sua vez a abordagem de comparação com tráfegos legítimos tem por finalidade identificar anormalidades no comportamento do tráfego suspeito, o que expande a capacidade de descoberta de atributos significativos, mas eleva a incerteza do resultado.

Os dados maliciosos utilizados pelas diversas abordagens estudadas são, em geral, de duas origens distintas: Fornecidos como entrada de dados para os sistemas por meio de um repositório (WERNER et al., 2009), (FALLAHI; SAMI; TAJBAKHSH, 2016), e providos por meio de integração com *honeypots*. Essa última foi utilizada em diversos trabalhos (ALTWAIJRY; SHAHBAR, 2013), (SAGALA, 2015) com a finalidade de automatizar a coleta de dados de atividades suspeitas ou maliciosas gerados recentemente. Isso possibilita a captura de comportamentos inéditos e tende a prover um grau mais elevado de segurança.

A extração das características relevantes foi realizada por meio da utilização de abordagens heterogêneas. Essas abordagens aplicaram tanto com a utilização de algoritmos tradicionais de otimização e seleção de características e métodos estatísticos (GÓMEZ et al., 2013), (OCAMPO; CASTILLO; GOMEZ, 2013), quanto novos procedimentos propostos especificamente para a solução de problemas encontrados no processo análise dos dados, como *Double Heavy Hitters* (AFEK; BREMLER-BARR; FEIBISH, 2013), algoritmo de sequência de padrões modificados (SHIM et al., 2017), entre outros.

Um trabalho que analisa semelhanças entre suas entradas para identificação de padrões é Shim et al. (2017). Neste trabalho, é fornecido um conjunto binário de dados de tráfego de rede. A análise executada é estratificada em três níveis de complexidade com o objetivo de refinar os elementos significativos para a composição da assinatura.

Inicia-se com a identificação de sequências semelhantes em dados binários de pacotes. A partir dessas sequências, os pacotes que as contêm são comparados em busca de similaridades. E, por fim, os fluxos de dados que contêm os pacotes selecionados são comparados para a identificação dos atributos que formam a assinatura. O algoritmo utilizado para a identificação das sequências relevantes é uma variação de algoritmo de padrão de sequências inicialmente proposto para uso no mercado financeiro. Os resultados alcançados se mostraram promissores para a identificação de padrões de comportamentos de rede para atividades no geral, inclusive aplicações legítimas de rede.

No trabalho Ussath, Cheng e Meinel (2016) apresenta-se um método para a geração de assinaturas com múltiplos passos capazes de representar ataques complexos. Para a geração das assinaturas, são utilizados *logs* de eventos de IDSs. Esses *logs* são processados e geram um grafo de atividades de ataques. A partir desse grafo são derivados os atributos para a formação da assinatura de ataques complexos. Essa abordagem permite a geração de assinaturas precisas, contudo não tem muita capacidade de representar características intra-eventos.

Diversas abordagens observadas produziram saídas de dados projetadas para a aplicação em IDSs baseados em assinatura atuantes em ambientes de rede (NIDSs). Essa arquitetura tem a vantagem de aplicar rapidamente as regras identificadas na análise de dados.

A abordagem apresentada no trabalho Sagala (2015) identifica atributos a partir da análise de tráfego de *honeypots* e tem, em sua arquitetura, meios para a aplicação das assinaturas em IDSs. Contudo, essas assinaturas não são diretamente utilizadas, uma vez que dependem da avaliação humana para a verificação e eventual correção de características selecionadas.

O trabalho Altwaijry e Shahbar (2013) faz uso de semelhanças de *substrings* para sumarizar e identificar elementos importantes em um bloco de dados de *logs* de *honeypots* por um método que os autores denominaram *"signature container array"*. Esses dados são coletados a partir de uma rede protegida com SNORT e esse mesmo IDS recebe as assinaturas geradas visando a melhoria da sua capacidade de detecção de atividades maliciosas. Esta abordagem foi capaz de reduzir significativamente o número de FN gerados pelo SNORT no ambiente de testes, ainda que tenha elevado sensivelmente o número de FP.

O método proposto em Gómez et al. (2013) utiliza tráfego binário de redes como entrada de dados. Esses dados são aplicados em um algoritmo evolucionário multi objetivos para a seleção de quais regras são mais relevantes para compor a assinatura.

Utiliza-se dois objetivos para a otimização: uma função objetivo de agregação único e otimização de Pareto. O uso de múltiplos objetivos no método de seleção permitiu uma maior variabilidade de indivíduos no algoritmo evolutivo, o que resultou em melhores combinações do que o uso de funções objetivo simples.

No trabalho Ocampo, Castillo e Gomez (2013) é apresentado um método para geração de assinaturas a partir de dados de *logs* de eventos de *honeypots* e tráfego binário de rede. Tráfegos de *honeypots* são processados a fim de se extrair métricas de tráfegos que serviram como características em potencial para a assinatura. Após isso, foram testados os algoritmos Naïve Bayes, C4.5, Árvores Aleatórias, Florestas Aleatórias, e Regressão Logística para a seleção de características. Sendo que o algoritmo Árvores Aleatórias apresentou o melhor resultado dentre os algoritmos testados.

Um método para geração de assinaturas para tráfego HTTP é apresentado no trabalho Garcia-Teodoro et al. (2015). Tráfegos binários de redes são utilizados como fonte maliciosa de dados e são comparados com uma base de tráfegos legítimos para a identificação de eventos maliciosos e atribuição de valor de probabilidade. Esses eventos, por sua vez, eram desmembrados e encontrados os componentes que mais contribuíram para a caracterização do tráfego como malicioso. Após isso, as assinaturas candidatas passam por comparações a fim de minimizar a redundância de assinaturas geradas.

O trabalho Werner et al. (2009) apresenta uma abordagem com a finalidade da geração de assinaturas de rede a partir de tráfegos gerados por *malwares*. Os dados binários de rede são interpretados e clusterizados visando identificar comportamentos de rede similares. Os dados pertencentes a um mesmo *cluster* são comparados para a identificação de sequências comuns. Essas sequências são convertidas em regras para o IDS Snort. A abordagem se mostrou capaz de lidar com ataques inéditos e não requer dados legítimos para a geração das assinatura.

Ataques com grandes volumes de dados podem ser analisados para a geração de assinaturas, como apresentado em Afek, Bremler-Barr e Feibish (2013). Neste trabalho, fluxos binários de rede de ataques DoS distribuidos (DDoSs) são comparados com o objetivo de identificar trechos binários constantes em pacotes de dados gerados por ferramentas de ataques massivos. Esses trechos, então, são comparados com uma base de dados de binários legítimos a fim de excluir possíveis geradores de FP conhecidos. Para a identificação dos trechos binários, os autores propõem a abordagem *Double Heavy Hitters* de comparação de dados para aumentar a eficiência do processo. O método foi capaz de lidar com ameaças inéditas, porém pode gerar um grande número de assinaturas por não realizar agrupamento o que pode ocasionar redundâncias.

O tipo de dado utilizado para a alimentação do sistema nos trabalhos estudados se concentraram em três grupos: *logs* de eventos provenientes de IDSs ou *honeypots*, tráfego binário de rede, e fluxos de rede. Fluxos de rede foram utilizados como fonte de dados no trabalho Fallahi, Sami e Tajbakhsh (2016). Nesse trabalho, os fluxos de rede foram processados para a criação de um conjunto de métricas. Após isso, foram aplicados nos algoritmos C5.0 e *Ripper* para a identificação dos padrões relevantes e geração das assinaturas. Obteve-se assinaturas com F1-*Score* superiores a 0.9 para os ataques de força bruta e DoS Distribuído, contudo os atributos de tempo não foram levados em consideração no trabalho.

2.9 Considerações Parciais

Observa-se, com base no conjunto de trabalhos presentes no estado da arte que a utilização de fluxos de redes para a geração automática de assinaturas é uma ocorrência recente e que não foram produzidos muitos estudos a seu respeito. Inspeção profunda de pacotes é geralmente utilizada para a identificação de atributos específicos de ataques em nível de aplicação, atingindo níveis de precisão que fluxos não são capazes de atingir e mesmo viabilizando algumas detecções cujo traço em nível de rede é, em geral, transparente.

Também, observa-se que diversos trabalhos requerem somente tráfegos maliciosos para a identificação de características relevantes, isso pode causar problemas como a demanda de muitas ocorrências do mesmo ataque ou a falta de capacidade de extração de características cujo normal varia conforme o ambiente, como, por exemplo, o número de fluxos ou a quantidade de portas de destino da camada de transporte. Essas fatores em aberto apresentam uma oportunidade de abordagem para este trabalho.

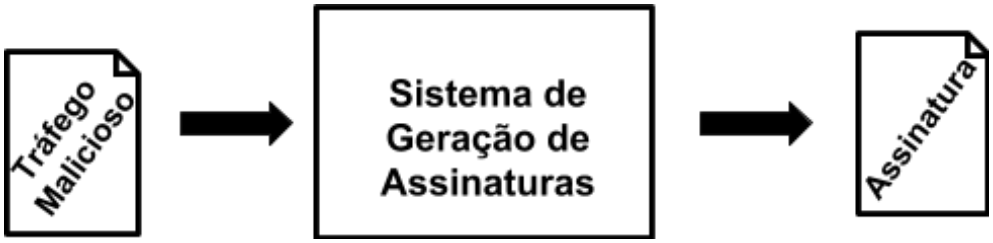
Capítulo 3

Metodologia

Neste capítulo, descreve-se a abordagem de solução para o problema de geração automática de assinaturas. Inicia-se com a exposição de coletas de dados na seção 3.1, seguindo do método de modelagem de tráfego legítimo na seção 3.2, a apresentação do modelo de assinaturas na seção 3.3, a extração de características na seção 3.4, a forma de classificação das assinaturas geradas pelo sistema na seção 3.5, o método de avaliação da abordagem na seção 3.6 e das conclusões parciais do capítulo na seção 3.7.

O sistema desenvolvido visa a extração de características significativas de uma seção de fluxos de tráfego de rede para a formação de uma assinatura de abuso, como ilustrado na Figura 3.1. Por princípio, assume-se que todo o tráfego de entrada do sistema é malicioso. Tráfegos legítimos que por ventura acompanhem tráfegos maliciosos podem prejudicar o processo de geração de assinatura, seja pela ocultação de padrões ou pela possível geração de assinaturas para atividades legítimas.

Figura 3.1: Diagrama geral da aplicação do sistema proposto.

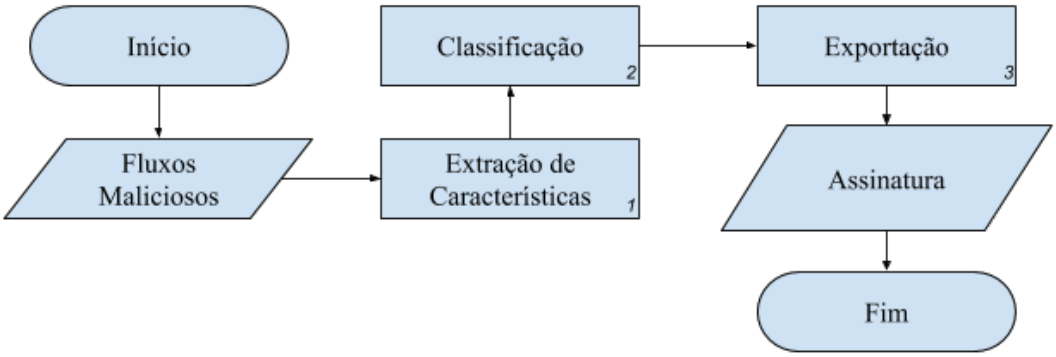


Fonte: Elaborado pelo autor.

O sistema, assumindo que o tráfego de entrada apresente características significativas, exporta um conjunto de atributos acompanhados de valores e operadores adequadamente concatenado de modo que permita a filtragem desses aspectos em um ponto de observação da rede.

O encadeamento das etapas do processo de geração de assinaturas é ilustrado na Figura 3.2. Fluxos de rede foram utilizados como forma de entrada de dados por consistirem em uma forma sumarizada e de baixo custo de manipulação quando comparada com inspeção profunda de pacotes. Também, fluxos permitem a compatibilidade com diversos sistemas de segurança e controle de redes de computadores (LI et al., 2013). Além de fornecerem um mais alto grau de privacidade do que o provido por inspeção profunda de pacotes.

Figura 3.2: Fluxograma da geração de assinaturas



A etapa *1* é apresentada na seção 3.4; a etapa *2* é apresentada na seção 3.5; e a etapa *3* consiste na conversão em JSON e conclusão da geração da assinatura.
Fonte: Elaborado pelo autor.

Para o cruzamento de dados realizado, observou-se que o vínculo dos valores dos fluxos das duas vias de comunicação do tráfego de rede analisado é importante, portanto, os fluxos simples são convertidos em *biflow* (TRAMMELL; BOSCHI, 2008) antes do seu processamento. Nessa conversão, os seguintes atributos apresentados na Tabela 3.1 devem estar disponíveis para o adequado funcionamento do sistema.

Cada um dos atributos apresentados na Tabela 3.1 tem sua relevância para o processo de extração de características. Por meio dos endereços IP é possível extrair a cardinalidade do tráfego, permitindo o adequado encaminhamento de processamento. As portas utilizadas na camada de transporte permitem a descoberta de serviços alvo de ataques. O número de pacotes e o de octetos são utilizados para observar características quantitativas de tráfego. Os tempos de início e fim permitem o cálculo da duração do *biflow*, assim como a existência de intervalos que configuram um passo na assinatura de abuso. A presença de *flags TCP* é utilizada para indicar sua utilização como constante em um ataque. O protocolo de transporte serve para distinguir os fluxos e encaminhar adequadamente o processamento de comparação de comportamentos de tráfego. Por fim, a duração permite a comparação do intervalo de atividade de tráfego.

Tabela 3.1: Atributos presentes nos *biflows* utilizados.

Atributo	Descrição
srcaddr	Endereço IP de origem do *biflow*
destaddr	Endereço IP de destino do *biflow*
srcport	Porta da camada de transporte utilizada pela origem
destport	Porta da camada de transporte utilizada pelo destino
dPkts_source	Quantidade de pacotes emitidos pela origem
dPkts_response	Quantidade de pacotes emitidos pelo destino
dOctets_source	Quantidade de octetos enviados pela origem
dOctets_response	Quantidade de octetos enviados pelo destino
first	Tempo quando o primeiro pacote passou pelo elemento de comutação que extraiu o fluxo, início do fluxo.
last	Tempo quando o último pacote passou pelo elemento de comutação que extraiu o fluxo, fim de fluxo.
tcp_flags_src_S	Presença da *flag TCP Sync* no tráfego proveniente da origem
tcp_flags_src_A	Presença da *flag TCP Ack* no tráfego proveniente da origem
tcp_flags_src_F	Presença da *flag TCP Fin* no tráfego proveniente da origem
tcp_flags_src_R	Presença da *flag TCP Reset* no tráfego proveniente da origem
tcp_flags_dest_s	Presença da *flag TCP Sync* no tráfego proveniente do destino
tcp_flags_dest_a	Presença da *flag TCP Ack* no tráfego proveniente do destino
tcp_flags_dest_f	Presença da *flag TCP Fin* no tráfego proveniente do destino
tcp_flags_dest_r	Presença da *flag TCP Reset* no tráfego proveniente do destino
proto	Protocolo de transporte utilizado na comunicação
duration	Duração do *biflow*

3.1 Coleta e preparação dos dados

A abordagem desenvolvida faz uso de três diferentes grupos de dados: *1*) Dados maliciosos de ataques; *2*) Dados legítimos da rede em observação; e *3*) Assinaturas de ataques maliciosos previamente geradas pelo sistema e manualmente classificadas.

Os dados do primeiro tipo são aqueles cujo propósito é o da geração de assinaturas; os dados do segundo tipo são necessários para o treinamento de um modelo de dados legítimos a fim de parametrizar o cálculo de probabilidades para a identificação

da pertinência de determinados atributos na seleção de características. Também, são utilizados para a filtragem de atributos significativos em contraste com uma massa de dados legítimos. Os dados do terceiro tipo são gerados controladamente pelo sistema com base em uma entrada conhecida de determinados tipos de ataques para a produção de material utilizado para o treinamento do classificador de assinaturas empregado na abordagem.

Os dados legítimos utilizados neste trabalho foram coletados da rede interna do Laboratório ACME!. Foram utilizados para o treinamento do modelo legítimo de 50 mil a 300 mil fluxos por dia de 10 dias úteis, totalizando de 500 mil a 3 milhões de fluxos. Este intervalo foi utilizado para minimizar possíveis interferências ocasionais no padrão comportamental da rede que possam ter ocorrido em algum dos dias coletados, melhorando a qualidade do perfil de tráfego da amostra utilizada.

Utilizou-se dados maliciosos coletados em ambiente controlado para o desenvolvimento deste trabalho. Estes fluxos foram gerados pela execução manual de ataques e extraídos com o uso da ferramenta *Nprobe* (DERI; SPA, 2003). Foi utilizado o protocolo Netflow versão 5 por ser o protocolo que apresenta a menor quantidade dentre as necessárias para a aplicação na ferramenta e é um dos primeiros padrões comercialmente adotados para a exportação de fluxos.

O ambiente de simulação utilizado foi constituído por duas redes locais conectadas por um roteador, que também provia comunicação entre o ambiente virtual e a internet. Uma das redes locais era composta por seis máquinas virtuais, a outra era composta por três. Foram utilizados internamente IPs não válidos globalmente, conectados à internet por meio de NAT.

As classes de ataques utilizadas no trabalho foram: Força bruta SSH, DoS, Varredura de reconhecimento, Varredura de vulnerabilidades, Injeção SQL, e XSS. Cada classe de ataques foi gerada com a aplicação de ferramentas apresentadas na Tabela 3.2. Para cada classe de ataques foram realizadas trinta execuções distintas, esse grupo de execuções foi dividido em três partes iguais e utilizadas para treinamento do classificador, para a geração de assinatura e, por fim, para a validação das assinaturas.

Essas classes de ataques foram escolhidas pela intensidade de seus ruídos em redes de computadores, quatro delas apresentam ruídos intensos (força bruta, DoS, e varreduras) e duas suaves (Injeção SQL e XSS).

Uma vez coletados os dados, faz-se o seu pré-processamento a fim de gerar *dataframes* que contenham agrupamentos de atributos a partir de atributos índices. Os atributos índices utilizados neste trabalho são aqueles que compõem a identidade de um fluxo: *srcaddr*, *destaddr*, *srcport*, *destport*, e *proto*. Os atributos de *dataframe* são

Tabela 3.2: Ataques e ferramentas utilizadas para as execuções.

Ataque	**Efeito**	**Ferramenta**
Força Bruta SSH	Tentativa de identificação de credenciais de acesso.	hydra
DoS	Provoca negação de serviço.	hulk
Varredura de Reconhecimento	Identifica serviços e máquinas disponíveis.	nmap
Varredura de Vulnerabilidade	Identifica vulnerabilidades em serviços.	aracna
Injeção SQL	Executa código arbitrário, nesse caso comprometendo a função de autenticação˙	firefox,nginx, mysql
XSS	Infecta site intermediário (vítima direta) a fim de executar código arbitrário no computador de usuários (vítima indireta e alvo principal)	firefox,nginx

apresentados na Tabela 3.3 e serão utilizados no trabalho para a geração das assinaturas.

Nos *dataframes*, os índices podem ser compostos de combinações de atributos. Quando um determinado atributo é utilizado no índice, o atributo que seria gerado pelo agrupamentos ou contagem desse atributo utilizado no índice é removido do *dataframe*, como apresentado na Tabela 3.4.

Tabela 3.3: Atributos presentes no *dataframe* utilizados para a extração de características.

Atributo	**Descrição**
nDestaddr	Número de endereços destino para cada índice.
nSrcaddr	Número de endereços origem para cada índice.
avgBytesSrc	Média do número de octetos provenientes da origem para cada índice
varBytesSrc	Variância do número de octetos provenientes da origem para cada índice
avgBytesDest	Média do número de octetos provenientes do destino para cada índice
varBytesDest	Variância do número de octetos provenientes do destino para cada índice
avgPacketSrc	Média do número de pacotes provenientes da origem para cada índice
varPacketSrc	Variância do número de pacotes provenientes da origem para cada índice

avgPacketDest	Média do número de octetos provenientes do destino para cada índice
varPacketDest	Variância do número de octetos provenientes do destino para cada índice
nPortSrc	Número de portas de transporte provenientes da origem para cada índice
nPortDest	Número de portas de transporte provenientes do destino para cada índice
flow_count	Número de fluxos agrupados para cada índice
duration	Média da duração dos fluxos agrupados para cada índice.
varDuration	Variância da duração dos fluxos agrupados para cada índice.
nProto	Número de protocolos da camada de transporte presentes nos fluxos agrupados para cada índice
tcp_src_a	Indica se todos os fluxos da origem agrupados tinham a *flag TCP Ack* para cada índice
tcp_src_f	Indica se todos os fluxos da origem agrupados tinham a *flag TCP Fin* para cada índice
tcp_src_r	Indica se todos os fluxos da origem agrupados tinham a *flag TCP Reset* para cada índice
tcp_src_s	Indica se todos os fluxos da origem agrupados tinham a *flag TCP Syn* para cada índice
tcp_dest_a	Indica se todos os fluxos do destino agrupados tinham a *flag TCP Ack* para cada índice
tcp_dest_f	Indica se todos os fluxos do destino agrupados tinham a *flag TCP Fin* para cada índice
tcp_dest_r	Indica se todos os fluxos do destino agrupados tinham a *flag TCP Reset* para cada índice
tcp_dest_s	Indica se todos os fluxos do destino agrupados tinham a *flag TCP Syn* para cada índice
nBytesSrc/ nPacketSrc	Número médio de octetos por pacote provenientes da origem para cada índice
nBytesDest/ nPacketDest	Número médio de octetos por pacote provenientes do destino para cada índice

Tabela 3.4: Índices do *dataframe* e atributos removidos.

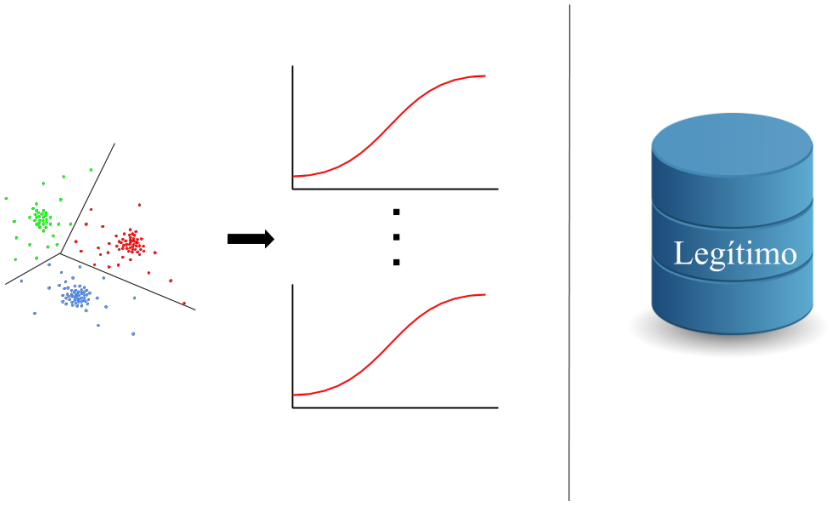
***Dataframe* de referência**	**Atributo(s) removido(s)**
destaddr	nDestAddr
srcaddr	nSrcAddr
destaddr-destport	nDestAddr, nPortDest
destaddr-destport-proto-srcport	nDestAddr, nPortDest, nProto, nPortSrc
destaddr-destport-srcport	nDestAddr, nPortDest, nPortSrc
destaddr-proto	nDestAddr,nProto
destaddr-proto-destport	nDestAddr,nProto,nDestPort
destaddr-proto-srcport	nDestAddr,nProto,nSrcPort
destaddr-srcport	nDestAddr,nSrcPort
srcaddr-destaddr	nSrcAddr,nDestAddr
srcaddr-destaddr-destport	nSrcAddr,nDestAddr,nDestPort
srcaddr-destaddr-destport-proto-srcport	nSrcAddr,nDestAddr, nDestPort,nProto,nSrcPort
srcaddr-destaddr-destport-srcport	nSrcAddr,nDestAddr,nDestPort,nSrcPort
srcaddr-destaddr-proto	nSrcAddr,nDestAddr,nProto
srcaddr-destaddr-proto-destport	nSrcAddr,nDestAddr,nProto,nDestPort
srcaddr-destaddr-proto-srcport	nSrcAddr,nDestAddr,nProto,nSrcPort
srcaddr-destaddr-srcport	nSrcAddr,nDestAddr,nSrcPort
srcaddr-destport	nSrcAddr,nDestPort
srcaddr-destport-proto-srcport	nSrcAddr,nDestPort,nProto,nSrcPort
srcaddr-destport-srcport	nSrcAddr,nDestPort,nSrcPort
srcaddr-proto	nSrcAddr,nProto
srcaddr-proto-destport	nSrcAddr,nProto,nDestPort
srcaddr-proto-srcport	nSrcAddr,nProto,nSrcPort
srcaddr-srcport	nSrcAddr,nSrcPort

3.2 Modelo de tráfego legítimo

O modelo de tráfego legítimo utilizado neste trabalho é formado por duas partes, independentes entre si e alimentadas com dados distintos, ilustradas na Figura 3.3, compostas por *clusters* e FDAs e por uma base de dados de tráfegos legítimos pré-processados. Utiliza-se para a atribuição de probabilidades aos atributos e identificação dos relevantes a partir dos dados maliciosos de entrada no processo de geração de assinatura.

O tráfego legítimo é relevante para este trabalho por fazer parte de duas etapas do processo de geração de assinaturas: a geração de modelo para a atribuição de probabilidades e a seleção de atributos significativos. Na Figura 3.4 à esquerda apresenta-se o fluxograma para a criação do modelo de *clusters* utilizado para a atribuição de probabilidades e à direita o fluxograma para a formação da base de comparação, usada para a seleção de atributos.

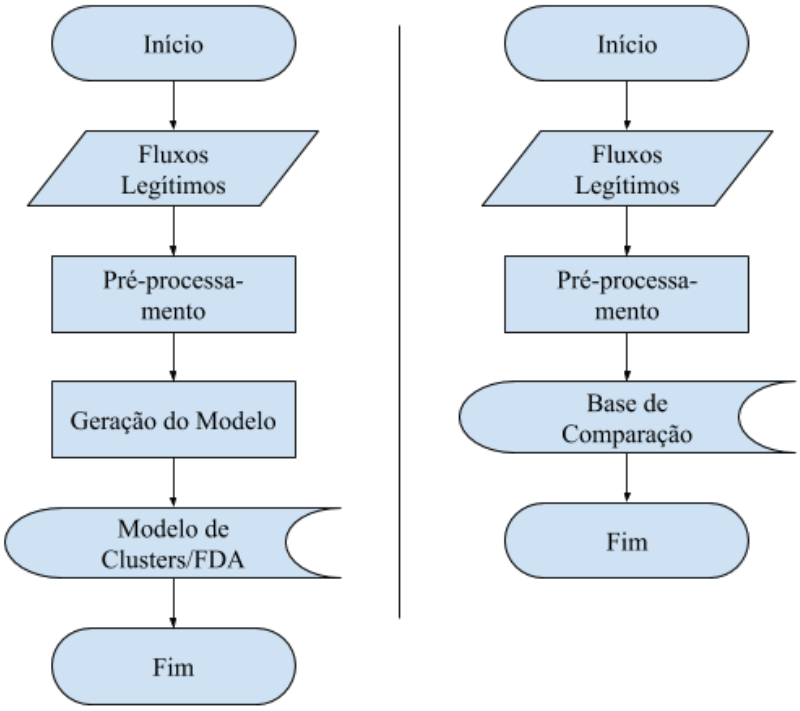
Figura 3.3: Diagramas exemplificando o modelo legítimo.



À esquerda o modelo de probabilidade baseado em *clusters* e FDAs. À direita, a base de comparações.
Fonte: Elaborado pelo autor.

Na primeira etapa de uso de tráfegos legítimos, os atributos *avgBytesDest*, *avgBytesSrc*, *avgPacketDest*, *avgPacketSrc*, *flow_count*, *duration*, e *nSrcAddr* são extraídos e utilizados para a produção de um conjunto de *clusters* baseado no algoritmo K-Means. Inicialmente, o algoritmo X-Means havia sido utilizado pela sua capacidade de identificação do melhor número de *clusters* para os dados de entrada, contudo, observou-se que o número de *clusters* escolhidos pelo algoritmo era constantemente o valor máximo parametrizado. Tendo isso em vista, o algoritmo foi alterado para o K-Means, o que resultou em um significativo ganho de desempenho sem diferença de qualidade

Figura 3.4: Fluxograma de criação do modelo legítimo.



À esquerda o modelo de probabilidade. À direita, a base de comparações.
Fonte: Elaborado pelo autor.

de *clusters*. Isso ocorre porque o método de geração de *clusters* do K-Means e do X-Means é semelhante. Por observação de desempenho da geração das FDAs, o número de *clusters* foi fixado em 75, uma vez que mais *clusters* que esse valor resultava em *clusters* com poucos dados e menos *clusters* não eram capazes de representar as diferenças entre os tráfegos legítimos utilizados.

O conjunto de *clusters* de tráfego legítimo é utilizado para a segmentação dos fluxos de rede por similaridade, assim como para a definição de um centroide que possibilita uma segunda divisão dos dados: agrupá-los entre maiores e menores do que o centroide para cada um dos seus eixos (atributos). Isso possibilita que seja calculada a FDA para cada um desses subgrupos, totalizando o número de FDAs igual a duas vezes o número de atributos do vetor de dados para cada *cluster*. Em conjunto com os *clusters*, as FDAs compõem o modelo de atribuição de probabilidades.

Cada FDA é calculada com base na FDP extraída de cada subgrupo de dados. Essa FDP é criada utilizando Estimativa de Densidade de Kernel Gaussiana. A divisão dos dados por meio de clusterização e pelo centroide de cada *cluster* reduz a incidência de múltiplos conjuntos de concentração nos dados aplicados à estimativa de densidade de kernel. Com isso, a FDP tende a apresentar estimativas mais verossimilhantes.

As probabilidades são associadas a cada um dos atributos de um vetor de dados malicioso, para qual inicialmente se define qual o *cluster* e qual o vetor de dados mais se aproxima. Após isso, define-se qual é a FDA adequada para cada atributo pela comparação do valor do atributo com o centroide do *cluster*. Então, calcula-se o valor da probabilidade aplicando o valor do atributo do vetor avaliado à FDA selecionada.

As variáveis que indicam a presença das *flags TCP* são isoladas e para cada uma delas é calculada a sua probabilidade (P_{f_c}) utilizando a Equação 3.1 com base na ocorrência do conjunto de valores de *flags* (N_c) e na ocorrência total de cada *flag* (N_{t_f}).

$$P_{f_c} = \frac{N_c}{N_{t_f}} \tag{3.1}$$

O outro grupo de dados de tráfego legítimo tem a função de compor a base de comparação. Essa base é utilizada para a definição final de pertinência de um atributo quantitativo na assinatura. Isso é feito incorporando o atributo potencialmente malicioso na assinatura candidata e em seguida filtrando esta base legítima com os critérios definidos na assinatura. Caso haja um retorno, então o critério não é satisfatório e uma outra combinação de atributos é testada. Isso se repete até que o retorno seja nulo ou não haja combinações disponíveis.

A formação da base de comparação é feita pela conversão de fluxos simples em *dataframes* e armazenado para busca e comparação no módulo responsável. Os dados exigidos nos fluxos legítimos para a utilização na base de dados de comparação são os mesmos exigidos para os fluxos maliciosos.

3.3 Padrão de assinaturas gerado

O padrão de assinaturas utilizado neste trabalho é baseado nas assinaturas aplicadas em Corrêa (2009). Essas assinaturas são capazes de descrever tanto comportamentos de atividades maliciosas, quanto comportamentos de atividades abusivas.

Como definido no trabalho Corrêa (2009), a identificação é baseada em intervalos de trinta minutos de fluxos. Esses intervalos correspondem ao tempo de manutenção de um fluxo na tabela de fluxos de um elemento de comutação.

Assume-se em Corrêa (2009) que um período acima de cinco minutos de silêncio de comunicação define um outro "passo" na assinatura, portanto as assinaturas geradas preveem esse tipo de ocorrência e possibilitam o encadeamento lógico necessário. Para que uma assinatura identifique positivamente um tráfego como malicioso, esse tráfego deve satisfazer a todos os passos presentes em uma assinatura.

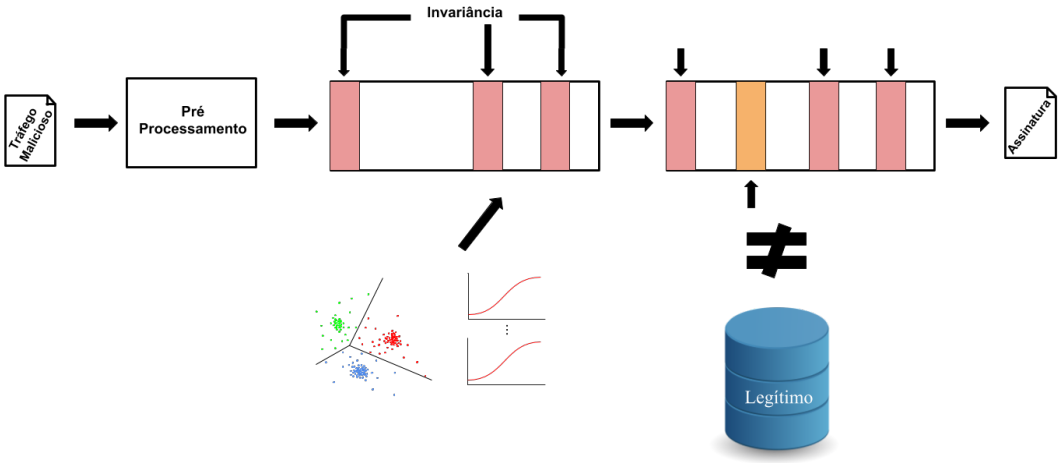
A cardinalidade do ataque é considerada na assinatura. Esse aspecto é utilizado

para a definição de atributos relevantes para cada caso, uma vez que os componentes analisados para cada cardinalidade podem diferir uns dos outros.

3.4 Metodologia para extração de características de fluxos maliciosos

A metodologia utilizada para a extração de características de fluxos maliciosos é ilustrada na Figura 3.5. Nessa, o processo de extração de características dos fluxos maliciosos de entrada, como apresentado no fluxograma presente na Figura 3.6, é composto de duas etapas: A primeira é a seleção de características de dados qualitativos pela identificação de atributos invariantes e a segunda é a seleção de dados quantitativos pela seleção probabilística. Juntos, esses dados compõem um agrupamento de cardinalidade da assinatura.

Figura 3.5: Diagrama da metodologia para extração de características para fluxos maliciosos.

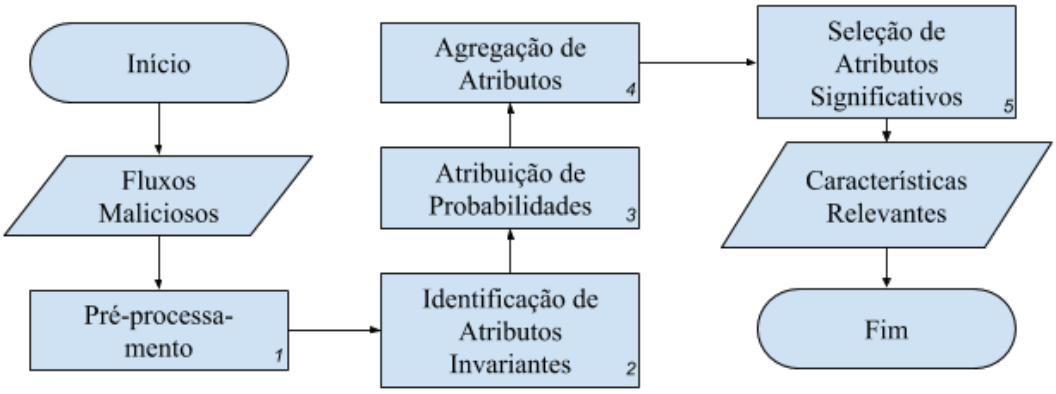


Fonte: Elaborado pelo autor.

Um agrupamento de cardinalidade é definido a partir das relações de quantos endereços IP estão presentes na rede de destino e quantos estão na(s) rede(s) de origem para um dado conjunto de fluxos maliciosos. A partir desse agrupamento, são selecionados os *dataframes* adequados para a continuidade do processo de seleção de atributos.

Esses *dataframes* são previamente calculados para todas as 24 possíveis combinações de atributos qualitativos, como apresentados na Tabela 3.3. Essas combinações servem como referência de agrupamento dos valores dos demais atributos presentes

Figura 3.6: Fluxograma para a extração de características na abordagem proposta.



O processo *1* é apresentado na seção 3.1; os processos de *2* a *5* são descritos nesta seção.
Fonte: Elaborado pelo autor.

no vetor, possibilitando uma posterior análise de padrões desses valores sumarizados.

A análise de cardinalidade tem quatro possíveis tipos de interações: 1-1 (interação um para um), 1-N (interação um para muitos), N-1 (interação muitos para um), e N-N (interação muitos para muitos). As cardinalidades 1-N e N-1 simples geram também um agrupamentos 1-1, isso permite a análise individual da interação entre atacante e vítima, visando a geração de um padrão complementar que atenda a ataques singulares preventivamente. A cardinalidade N-N não pode ser diretamente utilizada por não permitir a identificação de padrões em fluxos individuais, portanto ela é fragmentada em duas outras 1-N e N-1.

Cada um dos agrupamentos de cardinalidade passa pelo processo de extração de padrões e, se dentro de um mesmo passo um deles identificar um determinado tráfego, esse tráfego é considerado malicioso.

Identificação de atributos invariantes

Para a análise de cardinalidade, selecionam-se os dois *dataframes* iniciais: *srcaddr* e *destaddr*. Nesses dois *dataframes* observa-se o valor e a variância dos atributos *nDestAddr*, e *nSrcAddr*, respectivamente. Se algum desses dois atributos tiver valor igual a um e variância igual a zero, ele é selecionado. Caso ambos os atributos sejam selecionados, é definido que o tráfego é um 1-1. Caso somente *nSrcAddr* seja selecionado, então é definido que o tráfego é 1-N. Caso somente *nDestAddr* seja selecionado o tráfego é definido como N-1, e se nenhum deles for selecionado, o tráfego é definido como N-N e fica sujeito a tratamento adicional.

Para o caso de um tráfego N-N, o tratamento realizado consiste na separação do

tráfego em duas cardinalidades mais simples: uma 1-N e outra N-1. Essa divisão é feita criando duas cópias do *dataframe* original e para cada um deles retirando os IPs comuns entre origem e destino, no qual, para a formação do 1-N, retira-se da coluna de origem e, para a formação do N-1, retira-se da coluna de destino.

Simultaneamente à realização da análise de cardinalidade, observa-se a variância dos atributos *nPortSrc*, *nPortDest*, e *nProto* para os *dataframes srcaddr* e *destaddr*. Os atributos que apresentam variância nula e valor um são selecionados, juntamente com os atributos de cardinalidade, para compor o conjunto de atributos índices com a finalidade de definir o *dataframe* a ser analisado no processo posterior de extração de atributos relevantes quantitativos.

Para esses mesmos *dataframes*, observa-se as variâncias dos atributos *flags TCP* e nBytes/pacotes. Para as *flags TCP*, seleciona-se todas que apresentarem valor verdadeiro e variância nula para a composição do agrupamento de cardinalidade da assinatura. Para o atributo nBytes/pacotes, tanto de origem quanto de destino, seleciona-se todos os atributos que apresentarem variância de parte inteira nula para a composição do agrupamento de cardinalidade da assinatura.

Uma vez que definida a cardinalidade e o conjunto de atributos índices, seleciona-se o *dataframe* que apresenta a configuração de índices compatível para o processo de extração de atributos relevantes quantitativos. Esse *dataframe* é, a partir de então, utilizado na segunda etapa de extração de características.

A segunda etapa de extração de características é dividida em três subetapas: *1*) a atribuição de probabilidades para os valores de atributos do *dataframe* selecionado na etapa anterior; *2*) a unificação dos vetores de tráfego malicioso; e, por fim, *3*) a definição dos atributos significativos para o agrupamento de cardinalidade.

Associação de probabilidades

A primeira subetapa, que consiste na associação de probabilidade aos valores de atributos, é realizada da seguinte forma: inicialmente isola-se cada vetor malicioso presente no *dataframe*, então encontra-se qual o *cluster* a que cada vetor mais se aproxima. A partir desse *cluster*, obtém-se o seu centroide. Então observa-se as diferenças entre cada atributo do vetor e cada atributo correspondente no centroide a fim de se selecionar a FDA apropriada para cada atributo. Uma vez obtida, essa FDA é utilizada para se calcular o valor de probabilidade a partir do valor do atributo, e é, assim, associada a esse valor de atributo.

Agregação de atributos

Na segunda subetapa, uma vez que os atributos de todos os vetores do *dataframe* selecionado estejam associados com suas devidas probabilidades, seleciona-se dentre esses vetores, para cada atributo, o valor que apresenta maior probabilidade de ser malicioso, resultando em um vetor de máximos. Esse vetor de máximos será utilizado como entrada para a terceira subetapa do processo de extração de características quantitativas.

Seleção de atributos significativos

A terceira subetapa de extração de características quantitativas utiliza a base de dados legítimos de comparação e o vetor de máximos obtido na subetapa dois como entrada os atributos selecionados na etapa de seleção de atributos qualitativos. Assim, um filtro contendo os atributos qualitativos selecionados é montado e permanece fixo durante o processo. Os atributos quantitativos, que compõem o vetor de máximos, são organizados em ordem decrescente de probabilidade maliciosa, e a partir disso são feitas sucessivas buscas na base de dados legítima de comparação. Cada busca visa encontrar tráfegos que atendam a todos os requisitos do filtro. Portanto, espera-se um retorno vazio na busca, uma vez que os valores de atributos têm origem maliciosa. O retorno não vazio da busca indica que os atributos que compõem o agrupamento de cardinalidade da assinatura é insatisfatório e então uma nova combinação de atributos quantitativos é utilizada para o processo de busca. Isso se repete enquanto houverem combinações de atributos quantitativos não testados. Caso não haja retorno vazio mesmo após todos os testes, o agrupamento de cardinalidade é descartado.

Quando um agrupamento de cardinalidade é concluído, ele é incorporado ao passo da assinatura. Os agrupamentos se relacionam pela operação “OU”, portanto basta que um deles seja satisfeito para que o passo da assinatura seja atendido.

3.5 Classificação da assinatura

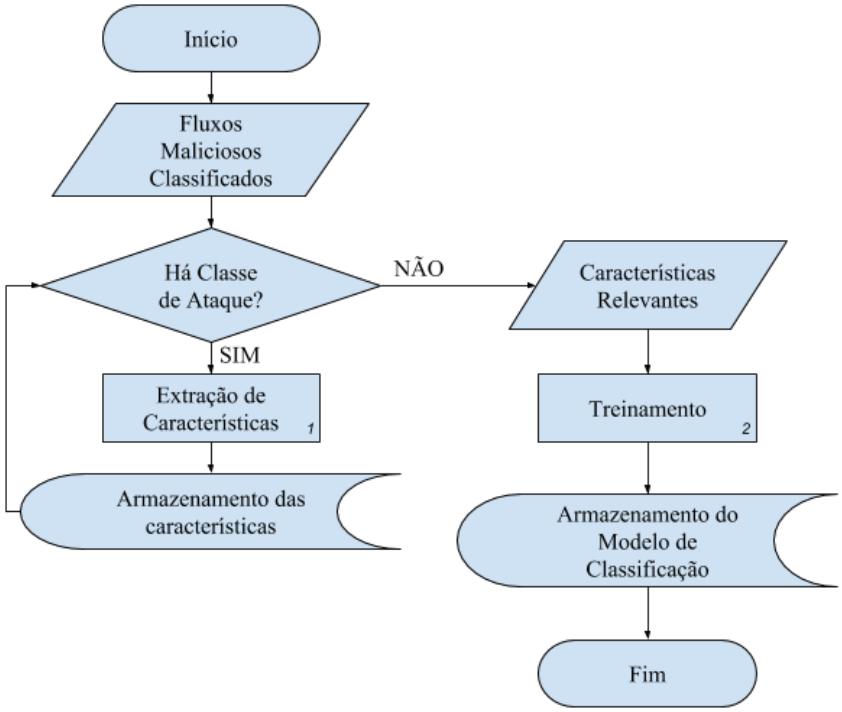
A classificação de uma assinatura é importante para o entendimento da sua causa geradora e a melhoria da atuação da resposta ao incidente. As assinaturas geradas nesta abordagem passam por um processo de classificação que lhes atribui um índice de proximidade com cada classe conhecida pelo classificador.

O classificador utilizado nesta abordagem aplica o conceito de similaridade de conjuntos. Isso foi definido em decorrência da estrutura da assinatura gerada (composta

por subsegmentos, como passos e cardinalidades): cada atributo de classificação carrega as informações de passo, cardinalidade e atributo de assinatura, formando um elemento único que tem como peso a probabilidade associada ao atributo de assinatura. No processo de classificação de uma nova assinatura, a presença dos atributos de classificação é comparada com o modelo. Utiliza-se o Índice de Jaccard Ponderado, apresentado na seção 2.6, para calcular o valor de pertinência a cada classe.

O processo de treinamento do modelo é feito com base em assinaturas geradas pelo sistema a partir de tráfegos maliciosos de classes previamente conhecidas, uma vez que o classificador utiliza uma abordagem supervisionada para a criação de seu modelo de aprendizado.

Figura 3.7: Fluxograma para a geração de dados de classificação e treinamento do classificador utilizado.



A etapa *1* é apresentada na seção 3.4, e a etapa *2* é descrita nesta seção.
Fonte: Elaborado pelo autor.

No treinamento, cada atributo presente na assinatura cria um atributo de classificação que contém informações de passo e a cardinalidade a qual o atributo de assinatura pertence, assim como o valor de probabilidade a ele associado. Os atributos de classificação compõem conjuntos de classe. Cada um dos atributos semelhantes e pertencentes aos conjuntos de mesma classe são unificados pela média dos seus pesos, resultando em um conjunto de atributos para cada classe de entrada no treinamento.

Nota-se que uma vez que as probabilidades associadas aos atributos de assinatura são relevantes para a qualidade da classificação, é coerente que o classificador seja treinado com assinaturas geradas utilizando o novo modelo legítimo a cada atualização do sistema de geração de assinaturas.

A classificação de assinaturas é realizada automaticamente após a sua geração e à assinatura são associados os valores de pertinência de cada classe conhecida pelo classificador. O processo de classificação ocorre da seguinte forma: *1*) Converte-se os atributos da assinatura em atributos de classificação; *2*) Para cada classe, aplica-se a Equação 2.3, obtendo-se como retorno o valor de pertinência da assinatura para a classe comparada; e *3*) Após o cálculo dos valores de pertinência, eles são organizados em ordem decrescente e adicionados à assinatura como metadados dedicados.

Nota-se que o modelo de classificação define com índice máximo uma assinatura que apresenta atributos e pesos iguais ao do modelo utilizado, se um peso de um atributo de uma assinatura a ser classificada possui peso tanto superior quanto inferior ao presente no atributo do modelo, o valor de pertinência da assinatura nessa classe é inferior ao máximo.

Após a classificação, a assinatura é exportada para o diretório de destino no formato JSON (BRAY, 2017), concluindo o processo de geração automática de assinaturas para o tráfego malicioso fornecido.

3.6 Avaliação da abordagem

A avaliação da abordagem apresentada neste trabalho foi feita com base em tráfegos legítimos coletados em ambiente real e tráfegos maliciosos gerados por meio de simulação. Esta abordagem foi avaliada em quatro aspectos distintos: *1*) Quanto a sua capacidade de geração de assinaturas a partir de fluxos de tráfego, subdivididos em maliciosos e legítimos; *2*) Quanto a qualidade das assinaturas de tráfego malicioso geradas pelo sistema; *3*) Quanto a quantidade de dados utilizados para treinamento do modelo legítimo em relação ao impacto causado na qualidade das assinaturas que esses modelos são capazes de gerar; e *4*) Quanto ao tempo de geração da assinatura e sua comparação com a qualidade obtida por essa.

Os testes de capacidade de geração de assinaturas tiveram como métrica o número de assinaturas produzidas para os diferentes tipos de ataques e tráfegos legítimos aplicados. Para esses testes esperou-se a não produção de assinaturas para tráfegos legítimos e a geração de assinaturas para fluxos maliciosos.

Uma vez que tráfegos legítimos não devem ter assinaturas produzidas pelo sistema,

sua presença no teste é um indício de incapacidade de filtragem adequada de tipos de tráfegos. Por sua vez, caso haja ausência, então o indício é de que o mecanismo de seleção de atributos por comparação com base legítima é funcional para a contenção da presença de tráfego legítimo na saída. Tráfegos maliciosos, por sua vez, foram avaliados quanto à sua geração.

Para o segundo aspecto observado, foram extraídas a precisão, *recall*, TFP, TFN, e o F1-*Score* como métricas de avaliação de assinaturas por serem amplamente utilizadas em trabalhos que abordam a classificação de tráfegos de rede, como apresentado na seção 2.7.

Para o terceiro aspecto observado, foram avaliados os modelos gerados modelos com quantidades diferentes de dados de tráfegos legítimos, 500 mil, 1 milhão, 2 milhões e 3 milhões de fluxos de rede e para cada um deles foram geradas assinaturas a partir de tráfegos maliciosos e legítimos. Essas assinaturas foram testadas quanto sua qualidade seguindo os critérios dos aspectos anteriores a fim de se encontrar uma relação entre quantidades de dados legítimos utilizados no treinamento e qualidade das assinaturas geradas pelos seus modelos resultantes.

Por fim, para o quarto aspecto, foram observadas as métricas de tempo de geração em segundos e F1-*Score*/segundo, visando quantificar a qualidade pelo tempo de produção da assinatura. Para essa métrica, quanto maior ela for, melhor o seu F1-*Score* e menor seu tempo de produção.

3.7 Considerações Parciais

A abordagem apresentada neste trabalho compara dados maliciosos com legítimos em busca de disparidades. Os atributos presentes nos *dataframes* são extraídos de dois modos distintos: por invariância, a fim de extrair características comuns aos fluxos maliciosos e por probabilidade de maliciosidade.

São atribuídas probabilidades de maliciosidade às disparidades identificadas. Para a seleção de características quantitativas, combinações desses atributos são testados ante um conjunto de dados legítimos. Esse método tem a desvantagem de requerer um significativo volume de dados e poder causar atrasos exponenciais caso não seja encontrada uma combinação de atributos satisfatória, uma vez que compara e verifica combinações de atributos.

Assume-se que assinaturas de uma mesma classe tenham atributos semelhantes e portanto, utiliza-se identificação de similaridades para a classificação das assinaturas.

A exportação das assinaturas e de seus metadados no formato JSON permite a

utilização dessas assinaturas em outros sistemas de modo simples e integrável.

Capítulo 4

Resultados

Os resultados apresentados neste capítulo têm como finalidade embasar a comprovação ou a refutação da hipótese de que a utilização da análise por invariância e a seleção dos atributos mais discrepantes em um fluxo malicioso são capazes de gerar assinaturas de ataques de redes de computadores baseadas em fluxos.

Esses resultados, adicionalmente, buscam indicar a qualidade e a coerência das assinaturas geradas para os tipos de ataques gerados, assim como identificar se existe uma relação entre a qualidade das assinaturas e a quantidade de informação utilizada no modelo. Para isso foram testados quatro modelos, cada um gerado com quantidades diferentes de fluxos legítimos: *1*) 500 mil; *2*) 1 milhão; *3*) 2 milhões; e *4*) 3 milhões de fluxos de rede, que foram distribuídos em duas partes iguais destinados ao treinamento de *clusters* e à formação da base de comparação.

Dez ocorrências de tráfegos foram dadas como entrada no sistema para cada um dos seis ataques analisados: ataque de força bruta SSH, DoS, varreduras de vulnerabilidade e reconhecimento, injeção SQL e XSS. Também, foram dados como entrada tráfegos legítimos de dez dias em três horários diferentes por dia, 10:00, 12:00, e 14:00 para que o comportamento do sistema fosse observado quanto a geração de assinaturas diante de entradas de dados legítimos.

As assinaturas geradas com base nessas entradas de dados foram utilizadas para a avaliação do sistema conforme as métricas apresentadas ante as seguintes quatro perspectivas: Capacidade de geração de assinaturas na seção 4.1, qualidade dessas assinaturas na seção 4.2, relevância da quantidade fluxos legítimos no treinamento de modelos para a geração de assinaturas na seção 4.3 e quanto ao tempo de geração de tais assinaturas na seção 4.4.

4.1 Capacidade de geração de assinatura

O sistema gerou assinaturas de todas as entradas de dados a ele alimentadas, tanto para tráfegos maliciosos, quanto para tráfegos legítimos. Com isso, é possível concluir que encontra-se padrões para os tráfegos maliciosos assim como para os legítimos, ainda que a definição de padrões para tráfegos legítimos tenda a ser minimizado no sistema pela probabilidade a partir de legítimos e da comparação com tráfegos idôneos.

Isso não era esperado em decorrência do uso da base de comparação para a seleção de atributos e reforça o problema da inserção de tráfegos maliciosos e legítimos em um mesmo conjunto no sistema, pelo seu considerável potencial impacto negativo. Ainda que tenha sido gerada saída de dados, espera-se que essas saídas não tenham significado prático e, portanto, não identifiquem tráfegos. A análise da capacidade de identificação de tráfegos das assinaturas geradas pelo sistema é apresentada na seção 4.2.

Para as saídas geradas a partir de dados maliciosos, foram produzidas assinaturas em número conforme o esperado. Ainda assim, essas assinaturas devem ser avaliadas quanto a sua qualidade, uma vez que o resultado do processamento do sistema deve ter significância. Tendo em vista que o sistema gerou saída para todos os tráfegos legítimos inseridos, é impossível afirmar que a simples produção de saída tenha algum significado qualitativo.

A Figura 4.1 ilustra uma assinatura produzida pelo sistema no formato JSON. Foram criados três grupos de dados principais: metadados, assinatura e classificação. No campo de metadados foram incluídas informações gerais, como data de criação e *hash* da assinatura. O campo "assinatura"comporta os dados extraídos pela análise do fluxos e o campo de classificação armazena os dados de pertinência a cada classe previamente conhecida.

Cada atributo é composto de quatro campos: valor, valor de referência, operador e relevância. Esses campos de atributo armazenam respectivamente os dados do valor encontrado no tráfego malicioso, o valor comparado no modelo legítimo, o operador indicado para a aplicação no uso da assinatura e a probabilidade do atributo ser malicioso.

4.2 Qualidade das assinaturas geradas

As assinaturas geradas pelo sistema foram submetidas a testes. Nesses testes, cada assinatura foi utilizada como critério para a classificação de fluxos de rede em maliciosos ou legítimos. Isso foi feito para avaliar as assinaturas quanto a sua capacidade

Figura 4.1: Exemplo de assinatura gerada pelo sistema.


```
{
    "metadata": {
        "uuid": "c06e895beef869112b2a822f873d421c",
        "creation_date": "2019-01-17 13:53:39.349228"
    },
    "signature": {
        "step0": {
            "1-N": {
                "all_have_tcp_S": {
                    "value_ref": true,
                    "value": true,
                    "operator": "==",
                    "relevance": 0.786049278213722
                },
                "all_have_tcp_A": {
                    "value_ref": true,
                    "value": true,
                    "operator": "==",
                    "relevance": 0.6746074087819813
                },
                "flow_count": {
                    "value_ref": 363.60869565217394,
                    "value": 469.0,
                    "operator": ">",
                    "relevance": 0.9769566608648234
                },
                "destport": {
                    "value_ref": 22,
                    "value": 22,
                    "operator": "==",
                    "relevance": 1.0
                },
                "proto": {
                    "value_ref": 6,
                    "value": 6,
                    "operator": "==",
                    "relevance": 1.0
                }
            }
        }
    },
    "classification": {
        "xss": 0.11320493986729575,
        "sql_injection": 0.0,
        "dos": 0.0,
        "bruteforce": 0.2264632345030636,
        "vulnerability_scan": 0.11639152915679488,
        "reconnaissance_scan": 0.0974337181233036
    }
}
```


Fonte: Elaborado pelo autor.

de identificação dos seus ataques alvo em ambiente externo ao gerador de assinatura. Inicialmente, analisou-se as assinaturas maliciosas. O conjunto de fluxo legítimos para teste das assinaturas maliciosas consistiu em 215.463 fluxos TCP e UDP originados de fluxos de atividades legítimas de dez horários distintos por dia em dez dias. A isso, foram adicionados os fluxos coletados de dez atividades de ataque para cada uma de suas classes, que totalizaram números de fluxos de acordo com o apresentado na Tabela 4.1 para cada classe de ataque analisada. A partir desses testes foram extraídas as métricas de avaliação da qualidade da assinatura, como *recall*, TVN, TFP, TFN, Precisão e F1-*Score*. A discussão dos resultados obtidos é feita por classe de ataque.

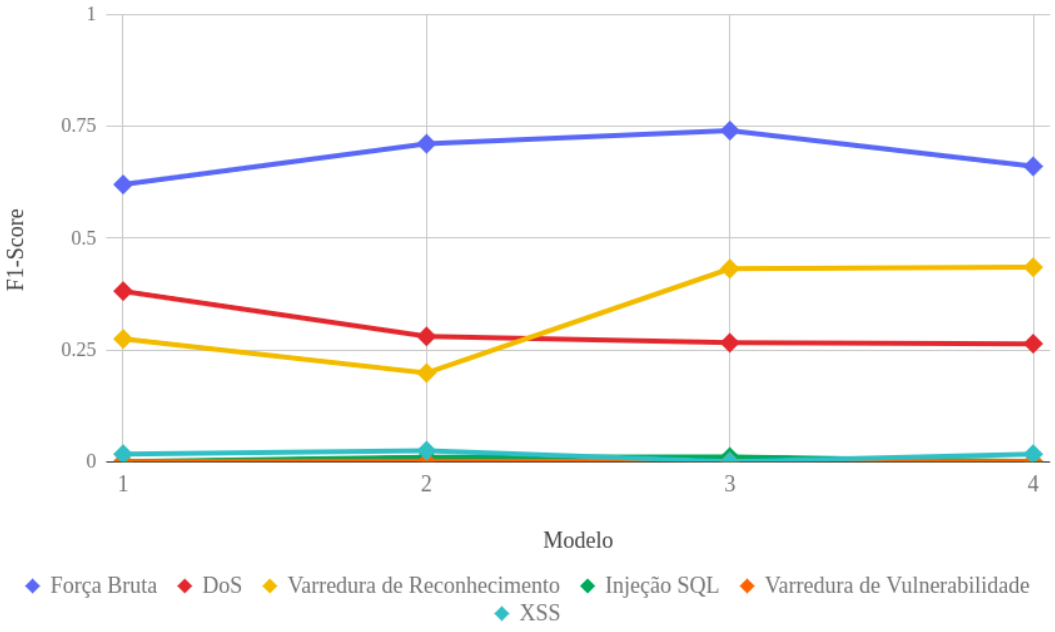
Tabela 4.1: Quantidades de fluxos utilizados para os testes do sistema.

Ataque	**Quantidade de fluxos**	**Total**
Ataque de Força Bruta	25000	240463
DoS	222117	437580
Varredura de Reconhecimento	25000	240463
Varredura de Vulnerabilidades	25580	241043
Injeção SQL	124	215587
XSS	102	215565

Ataques de injeção SQL e XSS

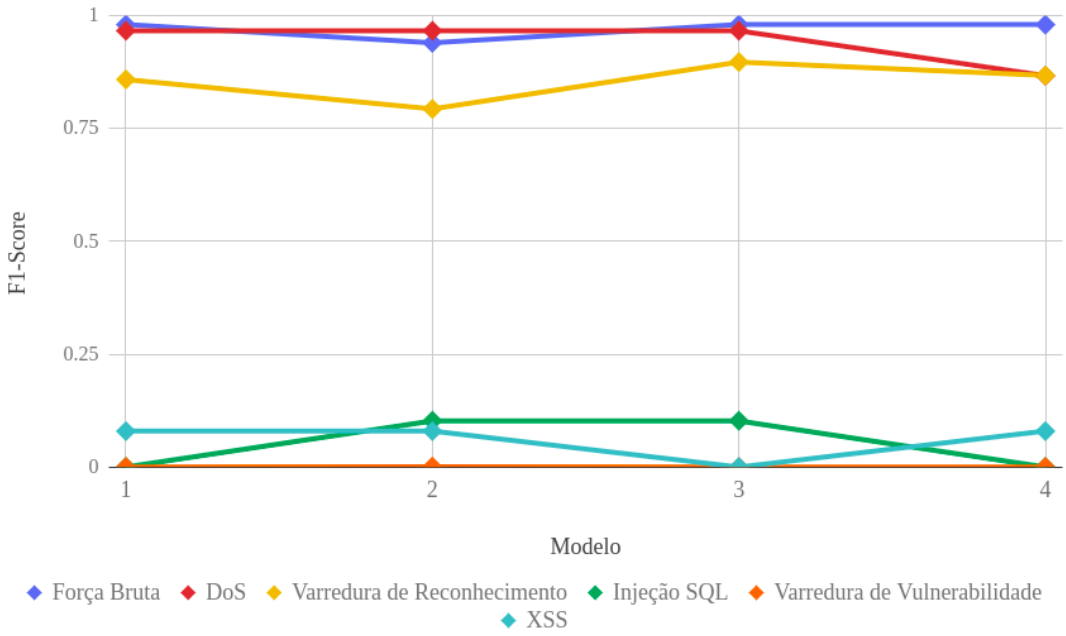
Para os ataques de injeção SQL e XSS, o desempenho médio das assinaturas geradas foi considerado muito baixo independentemente do modelo que as gerou, como pode ser observado na Figura 4.2, assim como nas Tabelas A.1 e A.2 do Apêndice A. As assinaturas dessas classes de ataques tiveram F1-*Score* de aproximadamente zero, sendo incapazes de identificar seus ataques. Mesmo observando o F1-*Score* máximo dessas classes de ataque na Figura 4.3, nota-se que não alcançaram 0.11. Uma assinatura considerada boa na literatura deve ter F1-*Score* maior ou igual a 0.90 (FALLAHI; SAMI; TAJBAKHSH, 2016), o que leva à conclusão de que o sistema como apresentado neste

Figura 4.2: Gráfico do F1-*Score* médio dos ataques para assinaturas geradas a partir de cada um dos quatro modelos testados.



Fonte: Elaborado pelo autor.

Figura 4.3: Gráfico do F1-*Score* máximo dos ataques para assinaturas geradas a partir de cada um dos quatro modelos testados.



Fonte: Elaborado pelo autor.

trabalho é inapto para a geração de assinaturas para as classes de injeção SQL e XSS.

As assinaturas para Injeção SQL e XSS apresentaram falta de restrições ou restrições incoerentes. Isso é resultado do baixo ruído de rede o que oculta os traços do ataque, fazendo com que se pareçam com tráfegos legítimos. Para a identificação desses tipos de ataques e geração de suas assinaturas é necessário o estudo de uma abordagem de reconhecimento de ruídos sensíveis, visto que a abordagem utilizada neste trabalho se comportou melhor com a identificação de padrões em ruídos menos discretos.

Ataque de varredura de vulnerabilidades

Para o ataque de varredura de vulnerabilidades, observa-se na Figura 4.2, assim como na Tabela A.3 do Apêndice A, que o F1-*Score* médio foi praticamente zero. Ainda que o ataque de varredura de vulnerabilidades tenha um ruído de rede mais significativo o sistema não foi capaz de isolar adequadamente seus atributos característicos. Mesmo observando a Figura 4.3 que expõe a curva de desempenho máximo obtido pelas assinaturas dessa classe, nota-se que esse desempenho continua nulo. Com base nisso, conclui-se que o sistema como apresentado neste trabalho é inapto para a geração de assinaturas para ataques de varredura de vulnerabilidades.

Mesmo assim, analisando manualmente diversas assinaturas de varredura de vulnerabilidade, observou-se que 9 de 10 assinaturas geradas a partir de cada um dos modelos apresentavam restrições excessivas, o que por princípio elimina em demasiado fluxos no processo de filtragem, resultando nos baixos *recall*, precisão e F1-*Score* observados. Isso pode ser melhorado pela aplicação métodos de otimização no processo de seleção de características para ajustar o critério de ordenação da fila de inclusão de atributos na assinatura com o objetivo de aumentar a relevância dos atributos escolhidos, ou pela alteração do critério de unificação dos diversos vetores dos *dataframes* da segunda subetapa da extração de características.

Ataque de varredura de reconhecimento

Para o ataque de varredura de reconhecimento, observa-se na Figura 4.2, assim como na Tabela A.4 do Apêndice A, que o seu F1-*Score* médio oscilou entre 0.1982 e 0.4347. Ainda que seja considerado um valor baixo, indicando assinaturas com pouca qualidade para as assinaturas geradas por todos os modelos mesmo com uma pequena elevação de qualidade média para as assinaturas geradas pelos modelos 3 e 4, seu desempenho máximo observado na Figura 4.3 é de um F1-*Score* de 0.8971, o que é considerado praticamente bom.

Ataque DoS

Para o ataque DoS, como pode ser observado na Tabela A.5 do Apêndice A, não houve ocorrência de FP para os fluxos testados, e a TVP em média foi de 0.2465, o que indica uma parcela baixa de fluxos de ataques identificados para a média das assinaturas. O *recall* varia entre 0 e 0.9349, sendo este último um valor bom de qualidade de assinatura. Conforme pode ser observado na Figura 4.2, o valor médio do F1-*Score* fica entre o.2636 e 0.3812, significando uma qualidade média abaixo do esperado para as assinaturas independentemente do modelo que a gerou. Contudo, foram geradas assinaturas de boa qualidade entre elas, sendo que o F1-*Score* máximo alcançado foi de 0.9664, como pode ser observado na Figura 4.3. Para esse ataque, em Fallahi, Sami e Tajbakhsh (2016) foram geradas assinaturas com F1-*Score* de 0.578, o que indica que o método proposto neste trabalho tem melhor eficiência na geração de assinaturas de DoS do que o presente na literatura.

Ataque de Força Bruta SSH

Os resultados de detecção das assinaturas para o ataque de força bruta são apresentados na Figura 4.2, assim como na Tabela A.6 do Apendice A. Observa-se que para esse ataque não foram encontrados falsos positivos, contudo nem todos os fluxos maliciosos foram identificados, com o *recall* variando de 0.03512 a 0.96096. Isso indica que nem todos os ataques geraram assinaturas significativas, resultando em um F1-*Score* médio entre 0.6199 e 0.7406. Contudo, observa-se na Figura 4.3 que o desempenho máximo das assinaturas geradas foi acima de 0.9, entre 0.9397 e 0.9801. Isso representa assinatura com alta qualidade, independentemente do modelo que as gerou, e se equipara com os resultados obtidos por Fallahi, Sami e Tajbakhsh (2016), que obtiveram F1-*Score* de 0.987, no seu melhor caso, para ataque de força bruta SSH.

Tráfegos Legítimos

Ainda que o sistema tenha gerado saídas de dados para as entradas de tráfego legítimo - que foram utilizados propositalmente com a finalidade de analisar a capacidade de filtragem de interferências em tráfego do sistema - essas saídas não têm significância, não sendo capazes de identificar qualquer tráfego no testes realizados.

4.3 Quantidade de dados para treinamento do modelo

Os diferentes modelos utilizados na validação deste trabalho foram formados utilizando diferentes quantidades de dados de entrada, como apresentado anteriormente, no início deste capítulo. Para cada modelo foram fornecidas as mesmas entradas de dados para geração de assinaturas tanto maliciosas quanto legítimas.

Esses diferentes modelos foram gerados e utilizados com o intuito de se observar o impacto da melhoria esperada de qualidade do modelo, pelo aumento da quantidade de dados utilizadas na sua produção, na qualidade das assinaturas geradas.

Observando-se as Figuras 4.2 e 4.3 nota-se que diferença qualitativa dessas assinaturas a fim de identificar a influência da quantidade de dados de treinamento do modelo de tráfego legítimo. O desempenho de assinaturas geradas para o ataque de força bruta atingiu seu máximo para os modelos 1, 3 e 4, com F1-*Score* máximo de 0.9801. Com relação ao F1-*Score* médio, o modelo 3 apresentou melhor desempenho dentre os 4, gerando o menor número de assinaturas com pouca qualidade.

Para o ataque DoS, o desempenho das assinaturas geradas foi decaindo gradualmente à medida que se aumentou a quantidade de dados de treinamento do mo-

delo. Nesse caso, o modelo 1 apresentou o melhor desempenho médio, com F1-*Score* 0.3811. O desempenho máximo foi semelhante para os modelos 1, 2, e 3, com F1-*Score* de 0.9664.

O ataque de varredura de reconhecimento teve melhor desempenho no modelo 3, com F1-*Score* máximo de 0.8971. Seu desempenho médio foi equivalente nos modelos 3 e 4 com valor de 0.4314 e 0.4347, respectivamente.

Os ataques de varredura de vulnerabilidades, injeção SQL e XSS não demonstraram desempenho muito significativo, e portanto, são indiferentes para esta análise. Como todos os tráfegos legítimos fornecidos para o sistema para a geração de assinaturas produziram saídas, esses tráfegos indicam que não há diferença significativa entre os modelos no que tange o comportamento quanto ao tráfego legítimo.

De modo simplista, pode-se concluir que o modelo 3, treinado com 2 milhões de fluxos de rede, tem o melhor desempenho entre os quatro modelos testados. Contudo, como pode-se notar observando as Figuras 4.2 e 4.3 sua vantagem média ou máxima frente aos demais é muito sutil e, portanto, como fica dentro da margem de erro, não configura um padrão. Desse modo, o volume de dados utilizado para a produção do modelo legítimo no intervalo testado é pouco significativo.

4.4 Tempo para geração de assinaturas

As assinaturas de ataques foram avaliadas quanto ao seus tempos de produção. Esses tempos são exibidos na Figura 4.4 e são principalmente influenciados pelas sucessivas tentativas de combinação de atributos quantitativos para a formação da assinatura. Isso indica que a complexidade para análise dos tipos de ataques não tem relação direta com a qualidade das assinaturas geradas, como pode ser observado na Figura 4.5, uma vez que tanto assinaturas geradas para o ataque com melhor desempenho (Força Bruta SSH) e para o ataque com pior desempenho (varredura de vulnerabilidades) apresentaram tempos médios semelhantes para os diversos modelos legítimos utilizados.

A análise de padrões gerados por entradas de tráfego legítimo pelo sistema apresentou significativo consumo de tempo - como pode ser visto na Tabela 4.2 - e resultou em saídas sem relevância, conforme era esperado. Isso indica que a presença de tráfegos legítimos nos fluxos inseridos no sistema podem afetar a confiabilidade das assinaturas geradas, visto que esses tráfegos legítimos podem causar a inclusão de atributos ou valores de atributos que excluam os fluxos alvo no processo de seleção, ou mesmo que acarretem a finalização prematura da formação da assinatura.

No tocante à formação dos modelos uma mesma quantidade de dados foi utili-

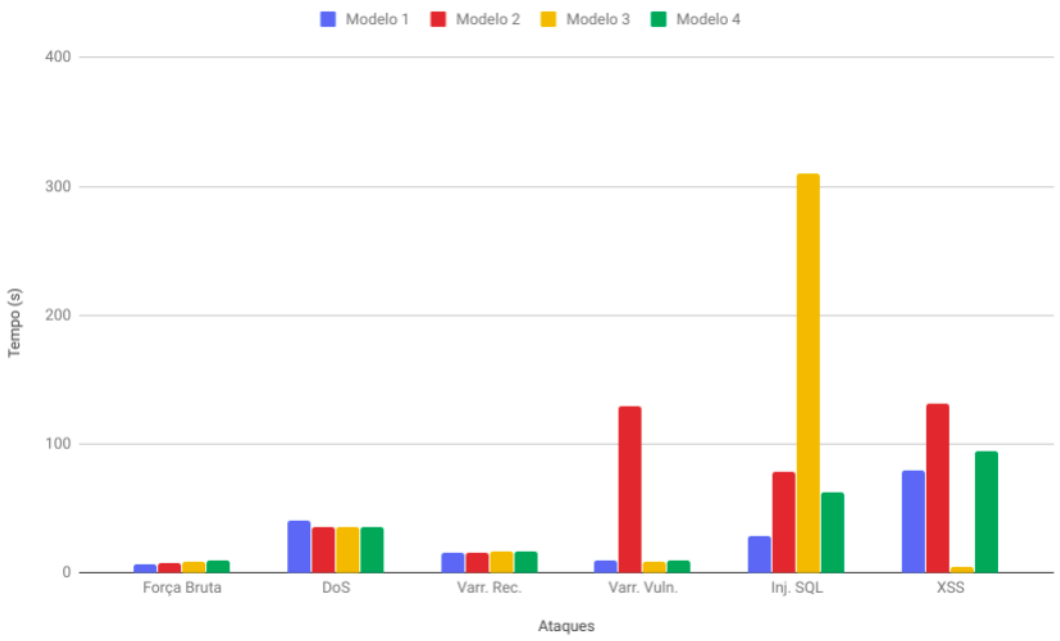
Tabela 4.2: Tempos médios de geração de assinaturas (em segundos).

Malicioso						
Modelo	**XSS**	**Injeção SQL**	**DoS**	**Força Bruta**	**Varredura de Vulnerabilidade**	**Varredura de Reconhecimento**
1	79	29	41	7	10	16
2	131	78	36	8	129	16
3	5	310	36	9	9	17
4	94	62	36	10	10	17

Legítimo			
Modelo	**10:00**	**12:00**	**14:00**
1	61	61	60
2	79	61	60
3	60	60	60
4	175	174	176

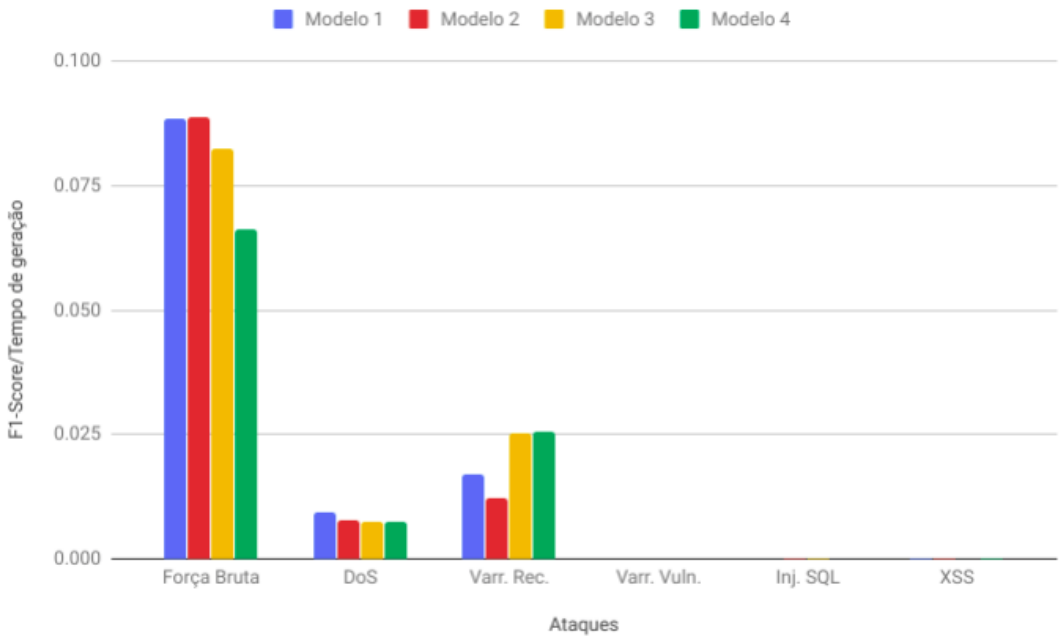
zada, para ambas as etapas. Mas a geração dos *clusters* exigiu significativamente mais tempo de treinamento do que a montagem da base de comparação, como apresentado na Figura 4.6. Em contrapartida, no momento da geração de assinaturas o tempo exigido para a utilização dos *clusters* é irrelevante quando comparado com o tempo de seleção de atributos por comparação (essa tarefa consome quase a totalidade do tempo

Figura 4.4: Gráfico dos tempos de geração para as assinaturas de cada classe de ataques por cada um dos quatro modelos testados.



Fonte: Elaborado pelo autor.

Figura 4.5: Gráfico dos F1-*Score* médios pelos tempos de geração médios.



Fonte: Elaborado pelo autor.

utilizado na produção de assinaturas).

Desse modo, observando a relação de tempo consumido para a geração completa do modelo legítimo, conforme pode ser visto na Figura 4.6, constata-se que o aumento de tempo tem comportamento exponencial, o que traz limitações à quantidade de dados utilizada para o treinamento do modelo legítimo. Comparando com a pequena diferença entre a qualidade das assinaturas geradas pelos diversos modelos testados, o custo de geração de modelos com grande quantidade de dados em *clusters* não se justifica, no que diz respeito aos parâmetros utilizados neste trabalho.

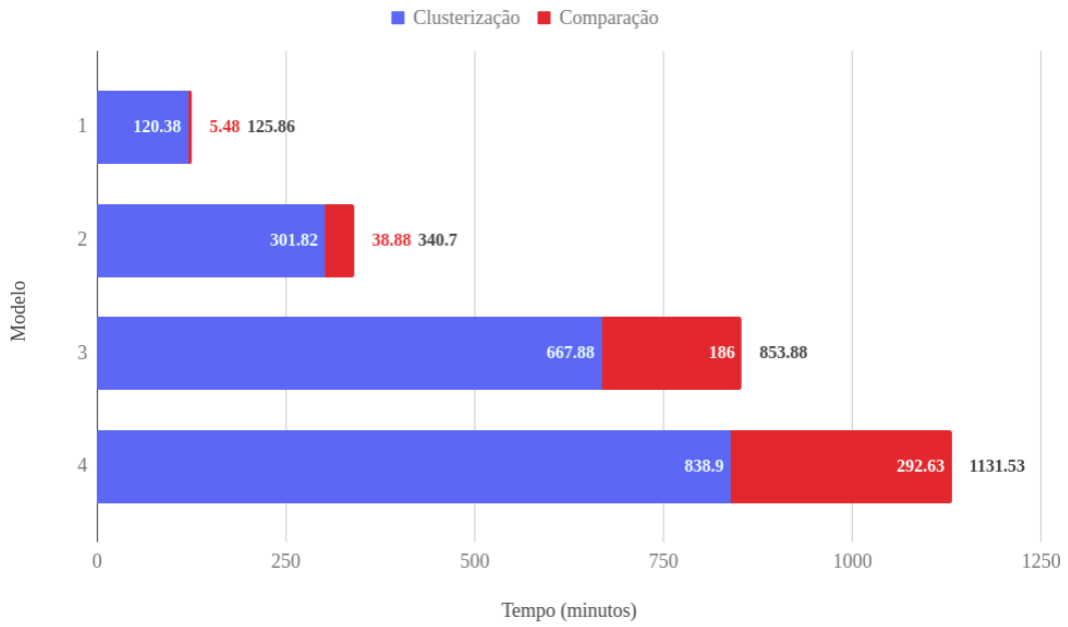
4.5 Considerações Parciais

Observou-se nos resultados do trabalho que há a identificação de padrões no conjunto de tráfegos maliciosos, contudo esse conjunto nem sempre é capaz de representar direta e adequadamente os elementos característicos da atividade maliciosa.

Também, tráfegos legítimos não são totalmente filtrados pelo sistema e suas entradas geram saídas com nenhuma significância e que podem reduzir a qualidade das assinaturas se inseridas em conjunto com tráfegos maliciosos.

Ainda que a geração de assinaturas não atinja sempre níveis de excelência, algumas delas obtiveram alto grau de desempenho, para os ataques avaliados, com F1-*Score*

Figura 4.6: Tempo de treinamento dos modelos legítimos (em minutos).



Fonte: Elaborado pelo autor.

próximos e acima de 0.9, como por exemplo, para os ataques de Força Bruta SSH, DoS e varredura de reconhecimento. Isso indica que, para alguns ataques, o desempenho da abordagem apresentada é funcional e que melhorias devem ser feitas como trabalhos futuros no método de seleção de características para priorizar atributos quantitativos mais significativos.

Para o ataque DoS, o método apresentado neste trabalho demonstrou melhor desempenho que o obtido por Fallahi, Sami e Tajbakhsh (2016) e desempenho equivalente ao alcançado por esses autores para o ataque de Força Bruta SSH.

O modelo utilizado teve pouca influência empírica da quantidade de dados fornecida, uma vez que gerou assinaturas para todas as classes de ataques com métricas de qualidade médias e máximas, cada qual dentro de um mesmo estrato, isso indica que a quantidade de dados necessária para a produção de assinaturas significativas é muito superior aos volumes analisados, ou que os montantes utilizados se encontram em um platô, com pouca diferença entre si.

O tempo de geração de assinaturas de até aproximadamente um minuto é rápido para o volume de dados analisado, sendo, no quesito velocidade, mais eficiente que um indivíduo para o processamento dos dados de entrada.

Por fim, observa-se que o sistema atende ao requisito de tempo proposto, mas requer melhorias na identificação de atributos significativos para a representação do

ataque e não somente sua distinção dos tráfegos legítimos. Ainda assim, alguns fluxos de ataques com alto ruído de rede geraram assinaturas satisfatórias.

Capítulo 5

Conclusões

Tráfegos presentes em redes de computadores podem ser sumarizados em fluxos de rede, esses fluxos contêm diversas métricas que podem ser utilizadas para, entre outras, a aplicação de segurança em redes de computadores. Tráfegos maliciosos podem definir valores característicos nos fluxos de redes, possibilitando, portanto, sua identificação.

Neste trabalho foi desenvolvida uma abordagem para a identificação automática desses atributos maliciosos e, com isso, a formação de uma assinatura possibilitando a identificação de outras ocorrências dessa atividade na rede. Nesse escopo, utilizou-se a análise de invariâncias e comparação com dados legítimos para a identificação de atributos maliciosos relevantes.

O alto desempenho no processamento de dados maliciosos era um requisito do objetivo do projeto e este foi alcançado com o tempo máximo de geração de saída de dados na ordem de minutos e com 70% dos ataques com tempo de produção de até 41 segundos.

Observou-se, contudo, que a qualidade das assinaturas geradas pelo sistema em sua maioria é baixa, com F1-*Score* menor do que 0.8. Os principais fatores que levam a este problema são a seleção de atributos com alto grau de distinção, mas com baixa significância - como, por exemplo, a escolha do número de portas de origem da camada de transporte ao invés do número de fluxos em uma assinatura de varredura - o que reduz a caracterização do ataque e a seleção de conjuntos de atributos que pode atender o critério de não identificação de fluxos legítimos, mas também não é capaz de descrever o ataque.

Ainda que em sua maioria as assinaturas não tenham atingido altos níveis de qualidade, o sistema foi capaz de gerar algumas assinaturas com F1-*Score* acima de 0.9. Comparando com o trabalho Fallahi, Sami e Tajbakhsh (2016), o ataque DoS conse-

guiu desempenho superior na geração de assinaturas e o ataque de força bruta SSH desempenho praticamente equivalente. Essas assinaturas têm como diferencial a seleção dos atributos que permitiram a significância para o ataque analisado. Esses resultados embasam a hipótese de que os problemas de seleção de características maliciosas significativas comentados anteriormente possam ser tratados com a modificação do critério de seleção de atributos quantitativos, seja, por exemplo, pela alteração da ordenação dos atributos submetidos ao teste de seleção, ou pela forma de unificação de atributos quantitativos. Isso pode ser feito com a aplicação de técnicas de aprendizado de máquina ou métodos estatísticos e é considerado como um dos trabalhos futuros a este.

Uma característica intrínseca à geração de assinaturas baseadas em uma amostra de eventos maliciosos é a excessiva especialização da assinatura gerada. São esperados resultados com alto grau de especificidade para a variação do ataque analisada. Isso requer métodos de generalização tanto para atributos quanto para valores de atributos. Tais métodos estão sendo desenvolvidos em paralelo a este trabalho no Laboratório ACME!.

A presença de tráfegos legítimos não foi totalmente ignorada pelo sistema, mesmo com a comparação com esse mesmo tipo de tráfego. Esse resultado tem consequências negativas para a aplicação do sistema em produção, uma vez que é esperado esse tipo de tráfego em meio ao fluxo malicioso. Uma proposta para se contornar esse problema é a segmentação do tráfego por IP de origem e destino, fragmentando os dados antes de seu processamento pelo sistema e com isso, possibilitando o processamento do tráfego malicioso com menos ou nenhuma interferência.

A análise e a aplicação de diferentes parâmetros para o K-Means, assim como outros algoritmos de clusterização, podem ser exploradas em trabalhos futuros para a melhoria da verosimilhança dos valores de probabilidade gerados, o que pode ter impacto positivo na seleção de atributos e melhorar a qualidade média das assinaturas produzidas pelo sistema. A composição dos métodos de associação de probabilidade e seleção de atributos significativos foram os elementos de maior complexidade de desenvolvimento na elaboração deste trabalho. Apresentam, além disso, um vasto conjunto de possibilidades de desenvolvimentos alternativos com suas próprias vantagens e desvantagens cujo comportamento para a aplicação deste trabalho ainda não é empiricamente conhecida.

Assim, este trabalho atinge os seus objetivos com sucesso parcial, sendo capaz de gerar automaticamente e em décimos de segundos assinaturas satisfatórias para os ataques de varredura de reconhecimento, DoS e força bruta SSH. Além de abrir

margens para desenvolvimentos futuros quanto à melhoria da qualidade das assinaturas geradas e quanto ao aumento da capacidade de tratamento de tipos de ataques.

Apêndice A

Tabelas de resultados para cada ataque

Tabelas de resultados apresentando métricas de avaliação para assinaturas geradas a partir de cada um dos quatro modelos testados para as classes de ataques analisadas.

Tabela A.1: Resultados das assinaturas para o ataque de Injeção SQL.

Modelo Gerador		**TVN**	***recall***	**TFP**	**TFN**	**Precisão**	**F1-*Score***
1	**Média**	0.9985	0.0000	0.0015	1.0000	0.0000	0.0000
	Coeficiente de Variação	0.0022	0.0000	1.4549	0.0000	0.0000	0.0000
	Mínimo	0.9952	0.0000	0.0000	1.0000	0.0000	0.0000
	Máximo	1.0000	0.0000	0.0048	1.0000	0.0000	0.0000
2	**Média**	0.9976	0.0601	0.0024	0.9399	0.0050	0.0093
	Coeficiente de Variação	0.0027	3.3166	1.1493	0.2121	3.3166	3.3166
	Mínimo	0.9935	0.0000	0.0000	0.3387	0.0000	0.0000
	Máximo	1.0000	0.6613	0.0065	1.0000	0.0554	0.1023
3	**Média**	0.9977	0.0661	0.0023	0.9339	0.0055	0.0102
	Coeficiente de Variação	0.0026	3.1623	1.1351	0.2239	3.1623	3.1623
	Mínimo	0.9935	0.0000	0.0000	0.3387	0.0000	0.0000
	Máximo	1.0000	0.6613	0.0065	1.0000	0.0554	0.1023
4	**Média**	0.9974	0.0000	0.0026	1.0000	0.0000	0.0000
	Coeficiente de Variação	0.0025	0.0000	0.9487	0.0000	0.0000	0.0000
	Mínimo	0.9952	0.0000	0.0000	1.0000	0.0000	0.0000
	Máximo	1.0000	0.0000	0.0048	1.0000	0.0000	0.0000

Tabela A.2: Resultados das assinaturas para o ataque de XSS.

Modelo Gerador		**TVN**	***recall***	**TFP**	**TFN**	**Precisão**	**F1-*Score***
1	**Média**	0.9293	0.2804	0.0707	0.7196	0.0087	0.0163
	Coeficiente de Variação	0.6530	0.0000	0.0000	0.2059	0.0000	0.0000
	Mínimo	0.6530	0.0000	0.0000	0.2059	0.0000	0.0000
	Máximo	1.0000	0.7941	0.3470	1.0000	0.0425	0.0794
2	**Média**	0.9277	0.3598	0.0723	0.6402	0.0148	0.0243
	Coeficiente de Variação	0.6530	0.0000	0.0000	0.2059	0.0000	0.0000
	Mínimo	0.6530	0.0000	0.0000	0.2059	0.0000	0.0000
	Máximo	1.0000	0.7941	0.3470	1.0000	0.0425	0.0794
3	**Média**	0.9997	0.0000	0.0003	1.0000	0.0000	0.0000
	Coeficiente de Variação	0.9976	0.0000	0.0000	1.0000	0.0000	0.0000
	Mínimo	0.9976	0.0000	0.0000	1.0000	0.0000	0.0000
	Máximo	1.0000	0.0000	0.0024	1.0000	0.0000	0.0000
4	**Média**	0.9279	0.2804	0.0721	0.7196	0.0087	0.0163
	Coeficiente de Variação	0.6530	0.0000	0.0000	0.2059	0.0000	0.0000
	Mínimo	0.6530	0.0000	0.0000	0.2059	0.0000	0.0000
	Máximo	1.0000	0.7941	0.3470	1.0000	0.0425	0.0794

Tabela A.3: Resultados das assinaturas para o ataque de Varredura de Vulnerabilidades.

Modelo Gerador		**TVN**	***recall***	**TFP**	**TFN**	**Precisão**	**F1-*Score***
1	**Média**	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000
	Coeficiente de Variação	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Mínimo	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000
	Máximo	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000
2	**Média**	0.9986	0.0001	0.0014	0.9999	0.0262	0.0002
	Coeficiente de Variação	0.0023	1.4626	1.5904	0.0001	3.0065	1.4798
	Mínimo	0.9952	0.0000	0.0000	0.9996	0.0000	0.0000
	Máximo	1.0000	0.0004	0.0048	1.0000	0.2500	0.0007
3	**Média**	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000
	Coeficiente de Variação	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Mínimo	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000
	Máximo	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000
4	**Média**	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000
	Coeficiente de Variação	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Mínimo	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000
	Máximo	1.0000	0.0000	0.0000	1.0000	0.0000	0.0000

Tabela A.4: Resultados das assinaturas para o ataque de Varredura de Reconhecimento.

Modelo Gerador		**TVN**	***recall***	**TFP**	**TFN**	**Precisão**	**F1-*Score***
1	**Média**	0.9514	0.2754	0.0486	0.7246	0.4386	0.2742
	Coeficiente de Variação	0.1079	1.2646	2.1113	0.4807	1.1235	1.1644
	Mínimo	0.7450	0.0000	0.0000	0.0424	0.0000	0.0000
	Máximo	1.0000	0.9576	0.2550	1.0000	1.0000	0.8586
2	**Média**	0.9882	0.1888	0.0118	0.8112	0.2427	0.1982
	Coeficiente de Variação	0.0377	1.6500	3.1623	0.3841	1.7334	1.6371
	Mínimo	0.8822	0.0000	0.0000	0.2423	0.0000	0.0000
	Máximo	1.0000	0.7577	0.1178	1.0000	1.0000	0.7933
3	**Média**	1.0000	0.3461	0.0000	0.6539	0.7000	0.4315
	Coeficiente de Variação	0.0000	0.9566	0.0000	0.5063	0.6901	0.8752
	Mínimo	1.0000	0.0000	0.0000	0.1865	0.0000	0.0000
	Máximo	1.0000	0.8135	0.0000	1.0000	1.0000	0.8971
4	**Média**	0.9984	0.3536	0.0016	0.6464	0.6849	0.4347
	Coeficiente de Variação	0.0049	0.9348	3.1623	0.5113	0.6934	0.8459
	Mínimo	0.9845	0.0000	0.0000	0.2342	0.0000	0.0000
	Máximo	1.0000	0.7658	0.0155	1.0000	1.0000	0.8674

Tabela A.5: Resultados das assinaturas para o ataque de DoS.

Modelo Gerador		**TVN**	***recall***	**TFP**	**TFN**	**Precisão**	**F1-*Score***
1	**Média**	1.0000	0.3269	0.0000	0.6731	0.4706	0.3812
	Coeficiente de Variação	0.0000	1.1880	0.0000	0.5769	1.0973	1.1333
	Mínimo	1.0000	0.0000	0.0000	0.0651	0.0000	0.0000
	Máximo	1.0000	0.9349	0.0000	1.0000	1.0000	0.9664
2	**Média**	1.0000	0.2377	0.0000	0.7623	0.3529	0.2801
	Coeficiente de Variação	0.0000	1.5634	0.0000	0.4874	1.4631	1.4969
	Mínimo	1.0000	0.0000	0.0000	0.0651	0.0000	0.0000
	Máximo	1.0000	0.9349	0.0000	1.0000	1.0000	0.9664
3	**Média**	1.0000	0.2249	0.0000	0.7751	0.3529	0.2658
	Coeficiente de Variação	0.0000	1.6659	0.0000	0.4835	1.3686	1.5620
	Mínimo	1.0000	0.0000	0.0000	0.0651	0.0000	0.0000
	Máximo	1.0000	0.9359	0.0000	1.0000	1.0000	0.9664
4	**Média**	1.0000	0.1966	0.0000	0.8034	0.5000	0.2636
	Coeficiente de Variação	0.0000	1.3709	0.0000	0.3354	1.0541	1.2305
	Mínimo	1.0000	0.0000	0.0000	0.2353	0.0000	0.0000
	Máximo	1.0000	0.7657	0.0000	1.0000	1.0000	0.8666

Tabela A.6: Resultados das assinaturas para o ataque de Força Bruta SSH.

Modelo Gerador		**TVN**	***recall***	**TFP**	**TFN**	**Precisão**	**F1-*Score***
1	**Média**	1.0000	0.4937	0.0000	0.5063	1.0000	0.6199
	Coeficiente de Variação	0.0000	0.5470	0.0000	0.5333	0.0000	0.4127
	Mínimo	1.0000	0.1000	0.0000	0.0390	1.0000	0.1818
	Máximo	1.0000	0.9610	0.0000	0.9000	1.0000	0.9801
2	**Média**	1.0000	0.6305	0.0000	0.3695	1.0000	0.7111
	Coeficiente de Variação	0.0000	0.5443	0.0000	0.9287	0.0000	0.4625
	Mínimo	1.0000	0.1000	0.0000	0.1137	1.0000	0.1818
	Máximo	1.0000	0.8863	0.0000	0.9000	1.0000	0.9397
3	**Média**	1.0000	0.6518	0.0000	0.3482	1.0000	0.7406
	Coeficiente de Variação	0.0000	0.4837	0.0000	0.9054	0.0000	0.3825
	Mínimo	1.0000	0.1677	0.0000	0.0390	1.0000	0.2873
	Máximo	1.0000	0.9610	0.0000	0.8323	1.0000	0.9801
4	**Média**	1.0000	0.5571	0.0000	0.4429	1.0000	0.6606
	Coeficiente de Variação	0.0000	0.5584	0.0000	0.7023	0.0000	0.4675
	Mínimo	1.0000	0.0351	0.0000	0.0390	1.0000	0.0679
	Máximo	1.0000	0.9610	0.0000	0.9649	1.0000	0.9801

Referências Bibliográficas

ABDOU, A.; BARRERA, D.; OORSCHOT, P. C. V. What lies beneath? analyzing automated ssh bruteforce attacks. In: SPRINGER. *International Conference on Passwords*. [S.l.], 2015. p. 72–91.

AFEK, Y.; BREMLER-BARR, A.; FEIBISH, S. L. Automated signature extraction for high volume attacks. In: IEEE PRESS. *Proceedings of the ninth ACM/IEEE symposium on Architectures for networking and communications systems*. [S.l.], 2013. p. 147–156.

AHMAD, M. A.; WOODHEAD, S.; GAN, D. Early containment of fast network worm malware. In: IEEE. *Information and Computer Science (NICS), 2016 3rd National Foundation for Science and Technology Development Conference on*. [S.l.], 2016. p. 195–201.

ALTWAIJRY, H.; SHAHBAR, K. (whasg) automatic snort signatures generation by using honeypot. *JCP*, v. 8, p. 3280–3286, 2013.

ALWAN, Z. S.; YOUNIS, M. F. Detection and prevention of sql injection attack: A survey. *International Journal of Computer Science and Mobile Computing*, v. 6, n. 8, p. 5–17, 2017.

BARABAS, M.; DROZD, M.; HANACEK, P. Behavioral signature generation using shadow honeypot. *World Academy of Science, Engineering and Technology, ser*, n. 65, p. 829–833, 2012.

BRAY, T. *The javascript object notation (json) data interchange format*. [S.l.], 2017.

CERT.BR. *Estatísticas dos Incidentes Reportados ao CERT.br*. 2019. <https://www.cert.br/stats/incidentes/>. Accessed on January 15, 2019.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, ACM, v. 41, n. 3, p. 15, 2009.

CORRêA, J. L. *Um modelo de detecção de eventos em redes baseado no rastreamento de fluxos*. Universidade Estadual Paulista, 2009. Disponível em: <http://www.dominiopublico.gov.br/pesquisa/DetalheObraForm.do?select_action=&co_obra=153259>.

DAVID, O. E.; NETANYAHU, N. S. Deepsign: Deep learning for automatic malware signature generation and classification. In: *2015 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2015. p. 1–8. ISSN 2161-4407.

DAVIS, J.; GOADRICH, M. The relationship between precision-recall and roc curves. In: ACM. *Proceedings of the 23rd international conference on Machine learning*. [S.l.], 2006. p. 233–240.

DERI, L.; SPA, N. nprobe: an open source netflow probe for gigabit networks. In: *TERENA Networking Conference*. [S.l.: s.n.], 2003.

DESAI, S. P.; HADULE, P. R.; DUDHGAONKAR, A. A. Denial of service attack defense techniques. 2017.

FALLAHI, N.; SAMI, A.; TAJBAKHSH, M. Automated flow-based rule generation for network intrusion detection systems. In: *2016 24th Iranian Conference on Electrical Engineering (ICEE)*. [S.l.: s.n.], 2016. p. 1948–1953.

FERREIRA, V. O. Classificaçao de anomalias e reduçao de falsos positivos em sistemas de detecçao de intrusao baseados em rede utilizando métodos de agrupamento. Universidade Estadual Paulista (UNESP), 2016.

GALHARDI, V. V. Detecção adaptativa de anomalias em redes de computadores utilizando técnicas não supervisionadas. Universidade Estadual Paulista (UNESP), 2017.

GARCIA-TEODORO, P.; DIAZ-VERDEJO, J. E.; TAPIADOR, J. E.; SALAZAR-HERNÁNDEZ, R. Automatic generation of http intrusion signatures by selective identification of anomalies. *Computers & Security*, Elsevier, v. 55, p. 159–174, 2015.

GÓMEZ, J.; GIL, C.; BAÑOS, R.; MÁRQUEZ, A. L.; MONTOYA, F. G.; MONTOYA, M. A pareto-based multi-objective evolutionary algorithm for automatic rule generation in network intrusion detection systems. *Soft Computing*, Springer, v. 17, n. 2, p. 255–263, 2013.

GU, G.; PERDISCI, R.; ZHANG, J.; LEE, W. Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection. 2008.

GUPTA, S.; ARBELAEZ, P.; MALIK, J. Perceptual organization and recognition of indoor scenes from rgb-d images. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2013. p. 564–571.

HE, D.; CHEN, X.; ZOU, D.; PEI, L.; JIANG, L. An improved kernel clustering algorithm used in computer network intrusion detection. In: IEEE. *Circuits and Systems (ISCAS), 2018 IEEE International Symposium on*. [S.l.], 2018. p. 1–5.

HEO, H.; SHIN, S. Who is knocking on the telnet port: A large-scale empirical study of network scanning. In: ACM. *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. [S.l.], 2018. p. 625–636.

HOFSTEDE, R.; ČELEDA, P.; TRAMMELL, B.; DRAGO, I.; SADRE, R.; SPEROTTO, A.; PRAS, A. Flow monitoring explained: From packet capture to data analysis with netflow and ipfix. *IEEE Communications Surveys & Tutorials*, IEEE, v. 16, n. 4, p. 2037–2064, 2014.

HOQUE, N.; BHUYAN, M. H.; BAISHYA, R. C.; BHATTACHARYYA, D. K.; KALITA, J. K. Network attacks: Taxonomy, tools and systems. *Journal of Network and Computer Applications*, Elsevier, v. 40, p. 307–324, 2014.

INAYAT, Z.; GANI, A.; ANUAR, N. B.; KHAN, M. K.; ANWAR, S. Intrusion response systems: Foundations, design, and challenges. *Journal of Network and Computer Applications*, Elsevier, v. 62, p. 53–74, 2016.

IOFFE, S. Improved consistent sampling, weighted minhash and l1 sketching. In: IEEE. *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. [S.l.], 2010. p. 246–255.

JAIGANESH, V.; MANGAYARKARASI, S.; SUMATHI, D. P. Intrusion detection systems: A survey and analysis of classification techniques. *International Journal of Advanced Research in Computer and Communication Engineering*, v. 2, n. 3, p. 1629–1635, 2013.

JAIN, A. K. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, Elsevier, v. 31, n. 8, p. 651–666, 2010.

JOHARI, R.; SHARMA, P. A survey on web application vulnerabilities (sqlia, xss) exploitation and security engine for sql injection. In: IEEE. *2012 International Conference on Communication Systems and Network Technologies*. [S.l.], 2012. p. 453–458.

KINDY, D. A.; PATHAN, A. K. A survey on sql injection: Vulnerabilities, attacks, and prevention techniques. In: *2011 IEEE 15th International Symposium on Consumer Electronics (ISCE)*. [S.l.: s.n.], 2011. p. 468–471. ISSN 2159-1423.

KREIBICH, C.; CROWCROFT, J. Honeycomb - creating intrusion detection signatures using honeypots. *Computer Communication Review*, v. 34, p. 51–56, 01 2004.

LEYDESDORFF, L. On the normalization and visualization of author co-citation data: Salton's cosine versus the jaccard index. *Journal of the American Society for Information Science and Technology*, Wiley Online Library, v. 59, n. 1, p. 77–85, 2008.

LI, B.; SPRINGER, J.; BEBIS, G.; GUNES, M. H. A survey of network flow applications. *Journal of Network and Computer Applications*, Elsevier, v. 36, n. 2, p. 567–581, 2013.

LIAO, H.-J.; LIN, C.-H. R.; LIN, Y.-C.; TUNG, K.-Y. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, Elsevier, v. 36, n. 1, p. 16–24, 2013.

MARCHETTI, M.; PIERAZZI, F.; COLAJANNI, M.; GUIDO, A. Analysis of high volumes of network traffic for advanced persistent threat detection. *Computer Networks*, Elsevier, v. 109, p. 127–141, 2016.

MEHMOOD, A.; UMAR, M. M.; SONG, H. Icmds: Secure inter-cluster multiple-key distribution scheme for wireless sensor networks. *Ad Hoc Networks*, Elsevier, v. 55, p. 97–106, 2017.

MEYER, P. L. Probabilidade: aplicações à estatística. In: *Probabilidade: aplicações à estatística*. [S.l.]: Livro Técnico, 1970.

MUDA, Z.; YASSIN, W.; SULAIMAN, M.; UDZIR, N. K-means clustering and naive bayes classification for intrusion detection. *Journal of IT in Asia*, v. 4, n. 1, p. 13–25, 2016.

NAIDU, V.; WHALLEY, J.; NARAYANAN, A. Generating rule-based signatures for detecting polymorphic variants using data mining and sequence alignment approaches. *Journal of Information Security*, n. 9, p. 265–298, 2018.

NITHYA, V.; PANDIAN, S. L.; MALARVIZHI, C. A survey on detection and prevention of cross-site scripting attack. *International Journal of Security and Its Applications*, v. 9, n. 3, p. 139–52, 2015.

OCAMPO, F. B. C. D.; CASTILLO, T. M. L. D.; GOMEZ, M. A. N. Automated signature creator for a signature based intrusion detection system with network attack detection capabilities (pancakes). *International Journal of Cyber-Security and Digital Forensics*, The Society of Digital Information and Wireless Communications, v. 2, n. 1, p. 25–36, 2013.

OLEJNIK, L.; CASTELLUCCIA, C. Towards web-based biometric systems using personal browsing interests. In: IEEE. *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*. [S.l.], 2013. p. 274–280.

PELLEG, D.; MOORE, A. W. et al. X-means: Extending k-means with efficient estimation of the number of clusters. In: *Icml*. [S.l.: s.n.], 2000. v. 1, p. 727–734.

RAHMAN, A.; KAWSHIK, K. R.; SOURAV, A. A.; GAJI, A. Advanced network scanning. *American Journal of Engineering Research (AJER)*, v. 5, n. 6, p. 38–42, 2016.

RAVALE, U.; MARATHE, N.; PADIYA, P. Feature selection based hybrid anomaly intrusion detection system using k means and rbf kernel function. *Procedia Computer Science*, Elsevier, v. 45, p. 428–435, 2015.

SADASIVAM, G. K.; HOTA, C.; ANAND, B. Honeynet data analysis and distributed ssh brute-force attacks. In: *Towards Extensible and Adaptable Methods in Computing*. [S.l.]: Springer, 2018. p. 107–118.

SAGALA, A. Automatic snort ids rule generation based on honeypot log. In: IEEE. *Information Technology and Electrical Engineering (ICITEE), 2015 7th International Conference on*. [S.l.], 2015. p. 576–580.

SCOTT, D. W. Multivariate density estimation and visualization. In: *Handbook of computational statistics*. [S.l.]: Springer, 2012. p. 549–569.

SHI, R.; NGAN, K. N.; LI, S. Jaccard index compensation for object segmentation evaluation. In: IEEE. *Image Processing (ICIP), 2014 IEEE International Conference on*. [S.l.], 2014. p. 4457–4461.

SHIM, K.-S.; YOON, S.-H.; LEE, S.-K.; KIM, M.-S. Sigbox: Automatic signature generation method for fine-grained traffic identification. *J. Inf. Sci. Eng.*, v. 33, n. 2, p. 537–569, 2017.

SILVERMAN, B. W. *Density estimation for statistics and data analysis*. [S.l.]: Routledge, 2018.

SUN, K.; PENG, P.; NING, P.; WANG, C. Secure distributed cluster formation in wireless sensor networks. In: IEEE. *Computer Security Applications Conference, 2006. ACSAC'06. 22nd Annual*. [S.l.], 2006. p. 131–140.

SZYNKIEWICZ, P.; KOZAKIEWICZ, A. Design and evaluation of a system for network threat signatures generation. *Journal of computational science*, Elsevier, v. 22, p. 187–197, 2017.

TRAMMELL, B.; BOSCHI, E. Bidirectional flow export using ip flow information export (ipfix)", rfc 5103. Internet Engineer Task Force, 2008.

USSATH, M.; CHENG, F.; MEINEL, C. Automatic multi-step signature derivation from taint graphs. In: IEEE. *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on*. [S.l.], 2016. p. 1–8.

WEBB, A. R. *Statistical pattern recognition*. [S.l.]: John Wiley & Sons, 2003.

WERNER, T.; FUCHS, C.; GERHARDS-PADILLA, E.; MARTINI, P. Nebula-generating syntactical network intrusion signatures. In: IEEE. *Malicious and Unwanted Software (MALWARE), 2009 4th International Conference on*. [S.l.], 2009. p. 31–38.

WU, S. X.; BANZHAF, W. The use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing*, v. 10, n. 1, p. 1 – 35, 2010. ISSN 1568-4946. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1568494609000908>.

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de São José do Rio Preto

TERMO DE REPRODUÇÃO XEROGRÁFICA

Autorizo a reprodução xerográfica do presente Trabalho de Conclusão, na íntegra ou em partes, para fins de pesquisa.

São José do Rio Preto, 20/02/2019

_____

Assinatura do autor