Instituto de Pesquisas Tecnológicas do Estado de São Paulo

Lígia Maria da Silva Danta

Método preventivo baseado em esquema de *ranking* e votação para detecção de intrusão em webservice

São Paulo
2018

Lígia Maria da Silva Danta

Método preventivo baseado em esquema de *ranking* e votação para detecção de intrusão em webservice

Dissertação de Mestrado apresentado ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo - IPT, como parte dos requisitos para a obtenção do título de Mestre em Engenharia da Computação: Infraestrutura Computacional

Data da aprovação ____/_____/_____

_____

Prof. Dr. Adilson Eduardo Guelfi (Orientador)
Mestrado Engenharia de Computação

Membros da Banca Examinadora:

Prof. Dr. Adilson Eduardo Guelfi (Orientador)
Mestrado Engenharia de Computação

Prof. Dr. Eduardo Takeo Ueda (Membro)
Mestrado Engenharia de Computação

Prof. Dr. Jorge Rodolfo Beingolea Garay (Membro)
USP – Universidade de São Paulo

Lígia Maria da Silva Danta

Método preventivo baseado em esquema de *ranking* e votação para detecção de intrusão em webservice

Dissertação de Mestrado apresentado ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo - IPT, como parte dos requisitos para a obtenção do título de Mestre em Engenharia da Computação: Infraestrutura Computacional

Orientador: Prof. Dr. Adilson Eduardo Guelfi

Coorientador: Prof. Dr. Anderson Aparecido Alves da Silva

São Paulo
Maio/2018

Ficha Catalográfica
Elaborada pelo Departamento de Acervo e Informação Tecnológica – DAIT
do Instituto de Pesquisas Tecnológicas do Estado de São Paulo - IPT

D192m **Danta, Lígia Maria da Silva**

Método preventivo baseado em esquema de ranking e votação para detecção de intrusão em webservice. / Lígia Maria da Silva Danta. São Paulo, 2018.
61p.

Dissertação (Mestrado em Engenharia de Computação) - Instituto de Pesquisas Tecnológicas do Estado de São Paulo. Área de concentração: Infraestrutura Computacional.

Orientador: Prof. Dr. Adilson Eduardo Guelfi
Coorientador: Prof. Dr. Anderson Aparecido Alves da Silva

1. Detecção de intrusão 2. Web service 3. Integridade de dados 4. Prevenção de ataque de indisponibilidade 5. Distribuição de Poisson 6. Tese I. Guelfi, Adilson, orient. II. Silva, Anderson Aparecido Alves da, coorient. III. IPT. Coordenadoria de Ensino Tecnológico III. Título

18-50 CDU 004.492.3 (043)

Bibliotecário responsável: Maria Darci Cornellas Narciso – CRB 8/3569

DEDICATÓRIA

Dedico esse trabalho ...

Aos meus pais Dionísio de Almeida e Eunice da Silva Dantas, meus verdadeiros exemplos de coragem, honestidade e luta.

Ao meu irmão Alceu Danta de Almeida, meu amigo e companheiro com quem eu posso contar a qualquer momento, sinônimo de trabalho e bom humor.

E ao meu filho Arthur Danta Gonzales Felix, meu Amor maior, para que um dia esse trabalho possa lhe servir de exemplo e incentivo.

AGRADECIMENTOS

Agradeço primeiramente a Deus e seus anjos guardiões, por ter me dado saúde, coragem, fé, perseverança e muitas outras bênçãos para eu concluir esse trabalho.

A minha mãe, Eunice da Silva Dantas, pela Vida, pelos ensinamentos, pelo exemplo de dignidade, força, perseverança e Amor, por cuidar com dedicação do meu filho para que eu estudasse.

Ao meu pai, Dionísio de Almeida, que sei, continua torcendo por mim onde quer que ele esteja.

Ao meu irmão Alceu Danta de Almeida, pela sua presença na minha vida.

Ao meu filho Arthur Danta Gonzales Felix, por me esperar pacientemente enquanto eu estudava e por ser esse Ser Maravilhoso que Deus colocou na minha para que eu aprendesse o verdadeiro significado do Amor.

Ao meu Amigo Adriano de Lima Barbosa, que esteve comigo desde o início da minha pesquisa, me auxiliando, embarcando nas minhas viagens e teorias e me auxiliando no desenvolvimento do meu experimento.

A minha amiga Sonia Henrique Araujo e aos meus amigos Diógenes Líbano Neves da Costa e Eliezer Freitas de Santana pelo apoio e incentivo nos momentos difíceis e por acreditarem que eu seria capaz de finalizar esse trabalho.

Ao orientador Adilson Eduardi Guelfi, por suas sugestões, correções e paciência no decorrer do desenvolvimento desse trabalho.

Aos professores membros da minha banca de dissertação, Prof. Dr. Eduardo Takeo Ueda e Prof. Dr. Jorge Rodolfo Beingolea Garay, pelas considerações.

Aos professores, alunos e funcionários do IPT que em todos os momentos foram solícitos às minhas necessidades e dificuldades.

E principalmente ao coorientador Prof. Dr. Anderson Aparecido Alves da Silva, meu exemplo de vida acadêmica e de paciência, que foi essencial na elaboração desse trabalho.

RESUMO

A tecnologia de *webservice* permite a integração e a troca de dados entre diferentes aplicações, independentemente da linguagem de programação, da plataforma ou dos protocolos utilizados. O Calculador de Preços e Prazos (CPP) é um *webservice* disponível gratuitamente na Internet por uma empresa de entregas, sendo que seus principais clientes são lojas virtuais de grandes empresas. Por tratar-se de um *webservice*, os servidores do CPP recebem constantes ataques na forma de consultas que causam indisponibilidade e lentidão no sistema. Esses eventos prejudicam o desempenho do serviço e causam distorções nos valores de fretes e prazos de entrega. Atualmente, o critério usado para a resolução do problema é o bloqueio dos dez endereços IP que apresentam o maior número de conexões estabelecidas com os servidores CPP no momento da lentidão no serviço. Esse critério é oneroso e eventualmente bloqueia clientes válidos. Desta forma, o objetivo desse trabalho é solucionar o problema de integridade dos dados e da indisponibilidade do serviço por meio de um método de detecção de ataques contra a disponibilidade baseado em um esquema de *rankings* e pesos que melhore a precisão e o tempo de resposta da detecção de conexões suspeitas em um *webservice*. A identificação de padrões é a metodologia que norteia esse trabalho. No experimento é criado um esquema de votação, gerado a partir do tráfego de rede, onde padrões baseados em estatística e regras pré-definidas são usadas para compor um *ranking* de suspeição para cada consulta. Como resultados esperados pretende-se melhorar a eficiência e o tempo de detecção de consultas maliciosas e diminuir a quantidade de bloqueios Falso Positivo (FP) e de liberações Verdadeiro Negativo (VN).

Palavras Chaves: *Webservice*; Distribuição de Poisson; Reconhecimento de padrões; Esquema de *ranking* e votação.

ABSTRACT

Preventive method based on ranking and voting scheme for webservice intrusion detection

The technology of webservice allows the integration and the exchange of data into different applications, regardless of programming language, platform or protocols utilized. The Price and Deadline Calculator (CPP) is a free webservice available on the Internet by a delivery company whose main customers are virtual stores and great companys. Since it is a webservice, the servers of CPP received constant attacks in the form of queries which cause unavailability and slowness in the system. This events harms the performance of service and cause distortion in the values of freights and delivery. Currently the critery used to solve the problem is blocking the first ten IP with greatest number of established connections with the CPP servers. Therefore, the goal of this work is propose a method to detect and solve the problem of data integrity and the unavailability of service with a schema of weighted ranks which improve the precision and response time of suspicion connections in a webservise. The identification of patterns is the methodology that guide this works. In the experiment is create a votting scheme, generate from network traffic, where patterns based in statistic and preset rules are used for compose a suspicion ranking to each query. The results are expected to improve the efficiency and detection time of malicious queries, in addition to the decrease of the number of False Positive (FP) blocks and True Negative (TN) access.

Keywords: Webservice*;* Poisson Distribution; Pattern Recognition; votting and ranking schema.

Lista de figuras

FIGURA 1 - PROPOSTA ILUSTRATIVA DO MP 32
FIGURA 2 - ESTRUTURA DO MP 33
FIGURA 3 - TOPOLOGIA DE REDE DO CPP 45
FIGURA 4 - EXEMPLO DO CICLO DE ANÁLISE 48
FIGURA 5 - PERCENTUAL DE SUSPEITOS E CRÍTICOS POR CENÁRIO 50
FIGURA 6 - PERCENTUAL COMPARATIVO DE DETECÇÕES EM TRÁFEGO NORMAL 52
FIGURA 7 - PERCENTUAL COMPARATIVO DE DETECÇÕES NA BLACK FRIDAY 53
FIGURA 8 - UTILIZAÇÃO TOTAL DE CPU – SERVER_CPP 54
FIGURA 9 - FLUXO PARA IMPLEMENTAÇÃO DO MP 55

Lista de tabelas

TABELA 1 - COMPARAÇÃO DOS TRABALHOS RELACIONADOS 30
TABELA 2 – DESCRIÇÃO DE CAMPOS DO *WEBSERVICE* UTILIZADOS NA ANÁLISE 34
TABELA 3 - CENÁRIO DE INTERVALOS DE TEMPO 35
TABELA 4 - CONTEÚDO DO BANCO DE REGRAS 36
TABELA 5 - EXEMPLO DO CÁLCULO DA ANÁLISE FIXA – DADOS FICTÍCIOS 38
TABELA 6 - EXEMPLO DO CÁLCULO DA ANÁLISE ESTATÍSTICA – DADOS FICTÍCIOS 40
TABELA 7 - IDENTIFICAÇÃO DE CLIENTES ESTATISTICAMENTE SUSPEITOS – DADOS FICTÍCIOS 41
TABELA 8 - CAPTURAS DE DADOS 47
TABELA 9 - PERÍODOS DE ANÁLISE 47
TABELA 10 - EXEMPLO DE ANÁLISE DE DADOS 49

Sumário

1 INTRODUÇÃO 11
1.1 Objetivo 13
1.1.1 Objetivo Específico 13
1.2 Contribuições 13
1.3 Método de trabalho 14
1.4 Organização da Dissertação 14
1.5 Resumo da Seção 15
2 REFERENCIAL TEÓRICO 16
2.1 Introdução 16
2.2 Aprendizado de máquina e *ranking* 16
2.3 Modelo de regressão de Poisson 18
2.3.1 Distribuição de Poisson 18
2.4 Técnicas de detecção de ataques 19
2.5 Resumo da Seção 21
3 TRABALHOS RELACIONADOS 22
3.1 Detecção e prevenção de Intrusão para proteção de ambiente em nuvem 22
3.2 Redução de alertas FP e VN por meio de esquema de votação ponderada baseada em credibilidade 22
3.3 Detecção de Anomalia por meio de aprendizado de máquina com voto por maioria 23
3.4 Classificação de alertas por meio de aprendizado de máquina 23
3.5 Classificação de Algoritmos de aprendizado de máquina baseado em esquema de *ranking* 24
3.6 Predição de anomalia no tráfego de rede usando filtro de Wiener adaptável e modelagem ARMA 25
3.7 Modelo de Previsão da Probabilidade Discreta de Distribuição do Ataque DoS 25
3.8 Gerenciamento de segurança de rede com agrupamento de padrões de tráfego 26
3.9 Detecção de *botnet* com base na análise de tráfego 26
3.10 Identificação de padrões de *botnets* por meio de monitoramento de rede 27
3.11 Comparação 30
3.12 Resumo da Seção 31
4 PROPOSTA DE TRABALHO 32
4.1 O Método 32
4.2 Camada de Tratamento de Dados 33
4.3 Camada de Aprendizado 35
4.3.1 Módulo 3 – Definição de Parâmetros 36

4.3.2 Módulo 4 Cálculo do *Ranking* 42
4.4 Camada de Resultados..... 42
4.4.1 Módulo 5 Liberar 42
4.4.2 Módulo 6 Bloquear..... 42
4.5 Resumo da Seção 42
5 EXPERIMENTO E ANÁLISE DE RESULTADOS..... 44
5.1 Introdução..... 44
5.2 Ambiente do CPP 45
5.3 Experimentos e resultados..... 46
5.3.1 Captura..... 46
5.3.2 Análise dos dados 47
5.3.3 Resultados 50
5.4 Resumo da Seção 55
6 CONCLUSÃO..... 56
REFERÊNCIAS..... 59

1 INTRODUÇÃO

Os processos críticos e dados sensíveis disponíveis na Internet dependem cada vez mais da segurança. Com isso a prevenção de intrusões e ataques de rede é uma questão de suma importância.

De acordo com Mudzingwa e Agrawal (2012), o *Intrusion Detection and Prevention Systems* (IDPS), é um dos principais mecanismos para manter os sistemas de informação seguros. IDPS são ferramentas de segurança usadas para monitorar, analisar e responder a possíveis violações de segurança contra sistemas de computador e rede. Essas violações podem ser causadas por intrusos externos não autorizados ou por usuários internos. Entre os sistemas e serviços de Internet mais comuns sujeitos a ataques estão os *webservices.*

A tecnologia de *webservice* permite a integração e troca de dados entre diferentes aplicações, independente da linguagem de programação, da plataforma ou dos protocolos utilizados. Segundo Lee e Jaffe (2016), os *webservices* são aplicações autocontidas que possuem interface baseada em XML e que descrevem uma coleção de operações acessíveis por meio da rede, independentemente da tecnologia usada na implementação do serviço, porém, a tendência na evolução da arquitetura do setor de *software* orientado a serviços é o microserviço.

De acordo com Lewis e Fowler (2014), microserviço é um nova abordagem de desenvolvimento de aplicações com vários pequenos serviços independentes, implementação automatizada e um gerenciamento mínimo centralizado. Esses serviços executam seus próprios processos e interagem entre si através de mecanismos como uma API HTTP, são escritos em diferentes linguagens e utilizam diferentes tecnologias para armazenamento de dados. A escalabilidade do microserviço agiliza o processo de correção de falhas, pois, por tratar-se de um conjunto de serviços independentes é possível alterar apenas o serviço que está com problemas, porém, essa descentralização de responsabilidades prejudica o gerenciamento de atualizações global do sistema.

O Calculador de Preços e Prazos (CPP) é um *webservice* disponível gratuitamente na Internet por uma empresa de entregas, não existe previsão para alteração da sua arquitetura. Os seus principais clientes são lojas virtuais de grandes empresas. O CPP integrado às aplicações das lojas virtuais fornece ao cliente o cálculo do valor do frete e o prazo de entrega da mercadoria. O cliente escolhe a mercadoria, aciona

a funcionalidade de cálculo de prazo e preço, digita o Código de Endereçamento Postal (CEP) da localidade de entrega e a aplicação, integrada ao *webservice* CPP, retorna as informações necessárias para a finalização da compra.

Segundo Queiroz, Vieira e Fonseca (2014), servidores de *webservice* recebem constantes ataques, dentre eles a execução maliciosa de arquivos, falhas por injeção e principalmente ataques do tipo *Distributed Denial of Service* (DDoS) de variantes diversas, como ataques volumétricos, ataques de esgotamento do *Transmission Control Protocol* (TCP) e ataques na camada de aplicação. Por tratar-se de *webservice*, os servidores do CPP são frequentemente atingidos.

A Indisponibilidade ou lentidão do sistema afeta o desempenho do serviço e a integridade dos dados, pois, caso haja alguma alteração de tarifas ou alteração de itinerário no momento do problema, o cálculo dos valores de frete e prazos de entrega sofrem alterações que prejudicam os clientes, as lojas virtuais e a credibilidade da empresa. Os problemas de lentidão ocasionados pelo acúmulo de consultas ao CPP são potencializados em ocasiões de grandes eventos e promoções na Internet, tais como: promoção relâmpago nos finais de semana, dia das mães, Black Friday e natal. Apesar do *webservice* CPP estar protegido por *firewalls* e detectores de intrusão, os ataques seguem um padrão normal de comportamento, semelhante a consultas comuns. Em geral, a solução adotada quando um ataque é detectado é o bloqueio do endereço IP de origem com o maior número de conexões estabelecidas com os servidores CPP. Essa ação é reativa, ocorre após o início e maior impacto do ataque, sendo também trabalhosa porque no momento do ataque são consultadas várias ferramentas manuais de monitoramento, que são ineficientes e podem bloquear tráfego vindo de clientes válidos.

Diante deste contexto, formas mais efetivas de detecção de ataques e busca de padrões no tráfego de rede podem antecipar a detecção e proporcionar maior disponibilidade e integridade para o *webservice*. Neste sentido, considerando um esquema de *ranking* como um conjunto de regras utilizado para a classificação de acessos ao *webservice* CPP, os eventos gerados podem ser então classificados como suspeitos ou normais.

1.1 Objetivo

O objetivo desse trabalho é propor um método preventivo de ataques de indisponibilidade baseado em um esquema de *rankings* e pesos que melhore a precisão e o tempo de resposta da detecção de conexões suspeitas em um *webservice*.

1.1.1 Objetivo Específico

- Reduzir o tempo de resposta de ataques de indisponibilidade; no método atual o tempo de resposta de ataques de indisponibilidade é muito alto, o que dificulta a resposta a incidentes e prejudica o negócio da empresa.
- Reduzir os Falso Positivos (FP); hoje a utilização de método manual, sem nenhum processo de identificação confiável gera o bloqueio de vários clientes válidos.
- Aumentar a detecção de Verdadeiro Negativos (VN); atualmente a classificação de VN é mínima, pois raramente são identificados no método utilizado, dessa maneira, clientes suspeitos acessam livremente o *webservice* prejudicando o seu desempenho;
- Adequar limites e pesos usados na construção do *ranking*; a principal ideia é a construção de parâmetros que possam ser ajustados de acordo com o *webservice* analisado.

1.2 Contribuições

As principais contribuições desse trabalho são:

a) Melhorar a eficiência do método de detecção de consultas maliciosas por meio de metodologias baseadas na análise das conexões;
b) Criar um método de treinamento com regras e limites flexíveis que possam ser alterados;
c) Criar métodos de detecção e medições adaptáveis à outros *webservices;*
d) Diminuir o tempo de resposta da detecção de intrusão para ataques de indisponibilidade no webservice*.*
e) Comparar o tráfego da Black Friday com o tráfego normal

1.3 Método de trabalho

Esse trabalho é uma pesquisa quantitativa no qual, por meio de um experimento, são comparados os resultados práticos que servem como base para a criação de um método detecção baseado em um esquema de *ranking*.
Para a realização do experimento, são utilizados *logs* capturados em ambiente real.

1.4 Organização da Dissertação

Este trabalho está organizado da seguinte forma:

- Seção 2 REFERENCIAL TEÓRICO: nesta seção são explicados os principais conceitos e definições utilizadas no desenvolvimento desse trabalho, a posição atual das pesquisas sobre a utilização do esquema de *ranking*, aprendizado de máquina, reconhecimento de padrões e detecção de ataques de rede. Por fim são discutidos os principais conceitos da distribuição de Poisson;
- Seção 3 TRABALHOS RELACIONADOS: são descritos os principais trabalhos revistos durante a fase de levantamento bibliográfico. Ao final da seção, encontra-se um quadro comparativo, o qual objetiva posicionar este trabalho de pesquisa em relação aos trabalhos revisados.

Seção 4 PROPOSTA DE TRABALHO: neste capítulo é apresentada a proposta de trabalho. A forma de captura e classificação dos *logs,* a aplicação da metodologia de votação/*ranking* para antecipação e melhoria da detecção, a aplicação da distribuição de Poisson para os cálculos estatísticos**.**

- Seção 5 EXPERIMENTO E ANÁLISE DE RESULTADOS: neste capítulo um experimento com dados reais é realizado como prova de conceito da proposta, com todos os cálculos e constatações realizados a partir das informações dos *logs* coletados. A análise do tráfego é realizada por meio dos seguintes critérios: volumetria do tráfego total, tráfego considerado como ataque e percentual de ataques confirmados.
- Seção 6 CONCLUSÃO DO TRABALHO: exibe uma análise geral do trabalho, destacando as contribuições obtidas, as limitações, os trabalhos futuros e as considerações finais.

1.5 Resumo da Seção

O CPP é um *webservice* disponibilizado gratuitamente na internet. Esse sistema sofre constantemente ataques do tipo DDoS, esses ataques provocam a sua lentidão e indisponibilidade. Para o melhor entendimento desse problema e suas consequências, essa seção discorre sobre os objetivos, as contribuições, o método de trabalho utilizado e a organização da dissertação.

2 REFERENCIAL TEÓRICO

Nessa seção são detalhados os principais conceitos utilizados nesse trabalho. Para melhor organização, está dividido em quatro subseções: 2.1- Introdução, 2.2 – Aprendizado de máquina e ranking, 2.3 – Modelo de regressão de Poisson e 2.4 Técnicas de detecção de ataque.

2.1 Introdução

Essa subseção apresenta uma visão geral do conteúdo das subseções subsequentes. Na subseção 2.2 são descritos detalhes sobre pesquisas que apresentam as características dos principais métodos de aprendizado de máquina e *ranking*, com foco na organização e classificação de dados. Já a subseção 2.3 apresenta uma breve descrição do modelo de Poisson utilizado e a subseção 2.4, mostra algumas técnicas e métodos utilizados na detecção de ataques.

2.2 Aprendizado de máquina e *ranking*

O estudo de aprendizado de máquina não é um assunto novo, em 1959 Samuel (1959) realizou um experimento com Jogos de damas, para tanto, ele investigou o aprendizado de máquinas com redes neurais e com redes planejadas com características específicas, esse último caso se enquadrou melhor no experimento, pois utilizou várias características, parâmetros e decisões do jogo para construir a base utilizada no aprendizado de máquina. Samuel (1959) provou que o computador pode ser programado para aprender a jogar damas melhor e mais rápido do que o próprio desenvolvedor do sistema, no tempo estipulado entre 8 ou 10 horas de processamento, tempo esse extremamente curto para o padrão computacional da época, já Mitchell (1997) definiu aprendizado de máquina como um programa que melhora seu desempenho em alguma tarefa por meio da experiência, e as suas principais características são: a classe de tarefas, a medida de desempenho a ser melhorada e a fonte de experiência.

Segundo Mitchell (1997) o aprendizado de máquina é influenciado pelas seguintes áreas:

- Inteligência Artificial (IA);
- Métodos Bayesianos;
- Teoria da Complexidade Computacional;
- Teoria do Controle;
- Teoria da Informação;

- Filosofia;
- Psicologia e Neurologia;
- Estatística; dentre outros.

Os conceitos apresentados por Mitchell (1997) são amplamente utilizados nos dias atuais.

De acordo com Shai (2014), aprendizado de máquina refere-se à detecção automatizada de padrões de dados. Salienta que, desde os anos 2000 o aprendizado de máquina tornou-se uma ferramenta comum para tarefas de extração de informações de grandes conjuntos de dados. As tecnologias baseadas em aprendizado de máquina são os principais motores de busca que trazem os melhores resultados. Os softwares de *antispam* aprendem sobre as mensagens de e-mail, as transações de cartões de crédito são garantidas pelo *software* que aprende como detectar fraudes, as câmeras digitais aprendem a detectar faces e os telefones celulares reconhecem comandos de voz. Os carros são equipados com sistemas de prevenção de acidentes que são construídos usando algoritmos de aprendizado de máquina, assim como os mesmos algoritmos são utilizados na bioinformática, na medicina e na astronomia.

Shai (2014) descreve quatro parâmetros sobre os quais os paradigmas de aprendizagem podem ser classificados:

a) Supervisionado e não supervisionado: o aprendizado supervisionado ocorre quando o aprendiz utiliza rótulos, isto é, o ambiente fornece alguma informação que é utilizada para criar regras de classificação. No aprendizado não supervisionado o aprendiz não utiliza rótulo, aprende por meio da observação dos dados, analisando padrões e similaridades do ambiente;
b) Aprendizes ativos e passivos: o aprendiz ativo interage com o ambiente, enquanto o aprendiz passivo observa o ambiente;
c) Utilidade do professor: o professor pode ser um processo aleatório que fornece o subsídio de aprendizado. A figura de um professor adversário é aplicada na análise de pior cenário, onde é possível aprender a detectar fraudes;
d) Aprendizado em tempo real ou em lote: no aprendizado em tempo real, o aprendiz responde e faz os ajustes durante o processo de aprendizagem,

enquanto que no aprendizado em lote, o processo de aprendizagem ocorre após o processamento dos dados.

O esquema de *ranking* ou votação é uma classificação ordenada de acordo com determinados critérios.

O esquema de *ranking* ou votação para classificação de eventos suspeitos é abordado por diversos autores. O trabalho de Hock e Kappes (2014) utiliza um sistema de aprendizado de máquina baseado em um esquema de votação para futuramente desenvolver um sistema de detecção de anomalias. Bierma, Doak e Hudson (2016) utilizam aprendizado de máquina para criar um esquema de *ranking* e classificar alertas de segurança. Estes autores pretendem controlar melhor o nível de criticidade dos alertas, isto é, classificar e priorizar os alertas de segurança, para garantir que o tempo e a energia de um analista se concentre nos alertas mais importantes. A criação de um esquema de *ranking* é uma maneira simplificada de aprendizado de máquina.

2.3 Modelo de regressão de Poisson

De acordo com Tadano, Ugaya e Franco (2009), os Modelos Lineares Generalizados (MLG) refletem o encontro de modelos lineares e não lineares. O MLG pertence a uma distribuição da família exponencial e esta é formada por diversas distribuições, dentre elas encontra-se a distribuição de Poisson.

2.3.1 Distribuição de Poisson

Segundo NIST (2012) a distribuição de Poisson é usada para modelar o número de eventos ocorridos dentro de um determinado intervalo de tempo.

A fórmula para a função de massa (densidade) de probabilidade de Poisson é:

$$\mathrm{p}(\mathrm{x},\ \lambda\ ) = \frac{e^{\lambda}\lambda^{x}}{x!}\ \ \text{para}\ \mathrm{x} = 0,1,2,\ldots \qquad (1)$$

λ é o parâmetro que indica o número médio de eventos em determinado intervalo de tempo.

De acordo com Paula (2004), para λ grande, temos que X segue aproximadamente uma distribuição normal de média λ e desvio padrão $\sqrt{\lambda}$. Para aplicar um modelo linear para explicar λ, deve ser resolvido o problema do desvio padrão depender da média, o que inviabiliza o uso de um modelo normal linear homocedástico. Para contornar essa situação pode-se aplicar a transformação da resposta X, de modo a

alcançar a normalidade e a constância de variância, mesmo que aproximadamente. Dessa maneira, tem-se que X é Poisson, segue quando $\lambda \rightarrow \infty$ resulta em: $\{\sqrt{X} - E(\sqrt{X})\} \rightarrow_d N(0,1/4)$. Portanto, quando λ é grande, a variável aleatória $2\{\sqrt{X} - E(\sqrt{X})\}$ segue aproximadamente uma distribuição N(0,1).

2.4 Técnicas de detecção de ataques

A segurança da informação em uma organização é dependente de políticas, processos, controles e procedimentos internos, além de eventuais políticas regulatórias externas e do firme patrocínio do corpo diretivo. Este conjunto de práticas e de regulações é o que torna mais eficiente e eficaz a prevenção e o combate a eventuais incidentes que venham a afetar a confiabilidade, confidencialidade, integridade, autenticidade e disponibilidade dos serviços entregues ao cliente interno e externo da organização.

Mell, Kent e Nusbaum (2005), de maneira geral, defendem que uma combinação de medidas técnicas, dentre as quais se incluem o gerenciamento de atualizações em sistemas e redes, a aplicação de privilégios mínimos para usuários, a eliminação do compartilhamento de arquivos de forma não segura, a remoção e desativação de serviços desnecessários de servidores e estações de trabalho, a instalação de softwares de antivírus em âmbito corporativo, a instalação equipamentos e softwares tais como *firewalls* e IDPS, destinados à proteção de perímetro, são fundamentais para que seja possível manter o ambiente de uma organização efetivamente protegido.

Segundo Scarfone e Mell (2007), incidentes são possíveis violações à política de segurança da informação de uma organização, violações estas que atingem servidores, estações de trabalho e redes desta organização. Ainda segundo tais autores, a detecção de intrusão é um processo que, por meio da monitoração e da análise de eventos ocorridos em sistemas ou rede de computadores, é capaz de detectar sinais de violação. Um incidente pode ter inúmeras violações motivadoras, dentre as quais podemos citar: agentes invasores de posse de credenciais de acesso a sistemas, *malwares* (*softwares* que criam e exploram falhas de equipamentos e usuários com privilégios amplos e desnecessários para acesso a sistemas). Existem ainda os incidentes originados por erros na configuração de equipamentos e softwares, e aqueles motivados pela digitação errônea do endereço de um sistema, que leva à descoberta de uma funcionalidade até então não

conhecida. Scarfone e Mell (2007) declaram ainda que os IDPS são excelentes ferramentas para respostas a incidentes, pois são capazes de identificar, notificar, responder, registrar, bloquear e alterar o ambiente alvo. Porém, não são totalmente precisos e geram grande quantidade de alertas FP e VN, por esse motivo, são necessárias análises adicionais para diferenciar alertas falsos.

De acordo com Scarfone e Mell (2007), os IDPS mais conhecidos são:

- *Network-Based* (NB) - realiza o monitoramento e análise de tráfego de rede e seus protocolos;
- *Wireless* - monitora e analisa o tráfego de rede sem fio;
- *Network Behavior Analysis* (NBA) - identifica ameaças por meio de fluxo de tráfego de rede incomum, como varreduras, DDoS, etc.;
- *Host-Based* - monitora e analisa um único *host* e seus eventos em busca de atividades suspeitas.

Scarfone e Mell (2007) salientam que, as principais classes de metodologias de detecção são:

- Detecção baseada em assinatura: comparação de eventos com assinaturas de ameaças conhecidas. Classe ineficaz para detecção de ameaças desconhecidas e ataques de múltiplos eventos;
- Detecção baseada em anomalias: comparação de atividades consideradas normais com desvios do padrão de normalidade. Utiliza perfis que monitoram atividades típicas por um período determinado e posteriormente utiliza esses perfis para identificação de anormalidades. Esse método é eficaz na identificação de ameaças desconhecidas. Porém, um perfil incompleto pode levar a identificação de um grande número de alertas FP;
- Detecção por análise de protocolo *stateful*: comparação de protocolos comuns e protocolos com desvios.

O aumento dos ataques cibernéticos é um fato preocupante para Lazarevic, *et al.* (2003), principalmente por ter aumentado significativamente a manutenção de sistemas utilizados no setor militar e comercial.

Lazarevic*et al.* (2003) confirma que, o método de detecção de intrusão baseado em assinaturas é o mais utilizado contra o terrorismo cibernético, porém, a sua principal desvantagem é a detecção apenas de ataques conhecidos, pois, a sua base de

dados de assinatura deve ser atualizada manualmente sempre que surge um novo tipo de ataque.

Segundo Lazarevic*et al.* (2003), o método de detecção de intrusão baseada em anomalia e a detecção por uso indevido podem ser rotuladas, dessa maneira, um algorítimo de aprendizado de máquina pode aprender a informação desse rótulo e utilizá-la para detectar novos tipos de ataques

Outra ferramenta essencial para o administrador de redes é o chamado *log* de segurança. De acordo com Kent e Souppaya (2006) o número, volume e a variedade de *logs* de segurança entregues por computadores e sistemas de informação, motivou a criação de um processo para gerenciar esses registros, tendo por base as melhores formas de geração, transmissão, armazenamento, análise e descarte. A análise de *logs* auxilia na identificação de incidentes, violações de políticas, fraudes, problemas operacionais, auditorias e análises forenses.

2.5 Resumo da Seção

Os principais conceitos, teorias e modelos que sustentam as argumentações desse trabalho baseiam-se em técnicas de detecção de ataques, distribuição de Poisson, aprendizado de máquina e *ranking.* Essa seção contém os principais materiais bibliográficos utilizados no trabalho.

3 TRABALHOS RELACIONADOS

Nessa seção são abordadas as principais discussões que motivam o raciocínio lógico no desenvolvimento do trabalho sobre os temas aprendizado de máquina e *ranking*, técnicas de detecção de intrusão e identificação de padrões. Gupta; Kumar e Braham (2013), Lin *et al.* (2013), Hock e Kappes (2014), Bierma, Doak e Hudson (2016) e Robson e Thomas (2015), discorrem sobre as técnicas de aprendizado de máquina, enquanto nos trabalhos de Celenk *et al.* (2008), Zhao, Yin e Long (2008) e Chiou (2014) são encontradas abordagens importantes sobre técnicas de detecção de intrusão.

Nas subseções enumeradas de 3.1 a 3.10 os trabalhos são agrupados por técnicas detalhadas que influenciaram esse trabalho e ao final da seção são discutidos os principais aspectos que aproximam e/ou diferem este trabalho dos artigos referenciados.

3.1 Detecção e prevenção de Intrusão para proteção de ambiente em nuvem

Segundo Gupta; Kumar e Braham (2013), os sistemas de detecção de intrusão são baseados em assinaturas ou em anomalias, e somente esses dois métodos não são eficazes para a detecção de ataques direcionados para as nuvens computacionais. Por esse motivo, eles desenvolvem um protótipo de sistema baseado em perfil, mesclando assinaturas e anomalias conhecidas pelos detectores de intrusão. Esse sistema é capaz de detectar, impedir e responder a ataques de rede em nuvem com maior eficácia. Os autores fazem um estudo do comportamento de máquinas virtuais por meio do esquema de *ranking*.

Contudo, Gupta; Kumar e Braham (2013) tratam apenas um tipo de assinatura, já conhecida dos detectores de intrusão. Os autores também deixam de citar o tempo para a atualização dos perfis das máquinas virtuais (MV) e a possibilidade de identificação de ataques em redes não mapeadas.

3.2 Redução de alertas FP e VN por meio de esquema de votação ponderada baseada em credibilidade

Segundo Lin *et al.* (2013), as altas taxas de alertas FP e VN entre os Sistema de Detecção e Prevenção de Intrusão (SDPI) tornam o seu desempenho insatisfatório, e para a redução desse problema é proposto um esquema de votação ponderada baseada em credibilidade - *Creditability-based Weighted Voting* (CWV) - para

alavancar diferentes conhecimentos de domínios entre múltiplos SDPI. Existem quatro componentes no algoritmo criado: *Creditability Modeling* (CM), *Authority Selection* (AS), *Voter Exclusion* (VE) e *Weighted Voting* (WV). O CM identifica a capacidade de detecção dos SDPI para diferentes tipos de tráfego e determina a idoneidade por meio do aprendizado de experiências passadas. Por meio da análise do protocolo e do alerta a AS define quais SDPI serão autoridade. Caso nenhum SDPI possa ser autoridade a VE excluirá aqueles com baixo desempenho. O WV atribui pesos para os SDPI escolhidos por AS ou excluídos pela VE e usa esses pesos para determinar um traço como malicioso ou como benigno. O CWV, por considerar o conhecimento de domínios diferentes é mais preciso e eficiente do que o VM e dessa maneira reduz os alertas FP e VN.

Contudo, o trabalho de Lin *et al.* (2013) trata os alertas de maneira geral e não distingue se o alerta é baseado em rede ou baseado em *host*, um fato primordial para o ajuste da credibilidade dos SDPI.

3.3 Detecção de Anomalia por meio de aprendizado de máquina com voto por maioria

O trabalho de Hock e Kappes (2014) aborda a técnica de detecção por heurística do SDPI. As abordagens heurísticas normalmente têm uma fase de aprendizado e uma fase produtiva, sendo que os sistemas de detecção de anomalias aprendem o comportamento normal da rede na fase de aprendizado por meio de um modelo estatístico, e realizam a detecção propriamente dita na fase produtiva. Porém, o modelo é estático e depende do momento da coleta para refletir o tráfego real da rede. Os autores propõem resolver esse problema combinando uma abordagem de aprendizado de máquina com VM. Porém, ao invés de utilizar um único modelo para examinar o tráfego de rede, são usados vários modelos que são atualizados e substituídos constantemente para melhorar a análise dos dados e reduzir a quantidade de FP.

Todavia, Hock e Kappes (2014) não fizeram testes de frequência de substituição. Tal abordagem tende a melhorar significativamente a qualidade da detecção de anomalias.

3.4 Classificação de alertas por meio de aprendizado de máquina

O trabalho de Bierma, Doak e Hudson (2016), propõe a criação de um modelo mais eficiente para o tratamento de alertas de segurança e utiliza os dados do projeto de pesquisa e desenvolvimento voltado para aprendizado de máquina desenvolvido no Sandia National Laboratories (SNL) e denominado *Active Learning for Alert Triage* (ALAT). Os alertas classificados pelo ALAT são compostos por milhares de caractecterísticas extraídas de seus recursos. Alguns destes recursos são úteis para prever a gravidade de um alerta. A proposta dos autores utiliza os recursos extraídos dos alertas do ALAT para aumentar os seus vetores de recursos, cujo conteúdo é mesclado com as informações extraídas do *Security Identifier Database* (SIDD). O SIDD agrega o endereço IP e as listas de bloqueios de vários departamentos e agências. Bierma, Doak e Hudson (2016) também utilizam o recurso de empilhamento de modelos, no qual a saída de um modelo é tratada como um recurso a ser usado pelo modelo de classificação. A finalidade do aprendizado de máquina é classificar os itens e utilizar estimativas de probabilidade a partir de um classificador binário implementando um modelo de floresta randômica.

Bierma, Doak e Hudson (2016) fizeram testes do modelo em alertas com curtos intervalos de tempo. Entretanto, o modelo não foi avaliado para intervalos mais longos, por esse motivo, perde a precisão na medida que os alertas envelhecem.

3.5 Classificação de Algoritmos de aprendizado de máquina baseado em esquema de *ranking*

A proposta do trabalho de Robson e Thomas (2015) é a classificação de dez algoritmos de aprendizado de máquina supervisionados, baseados em esquema de *ranking* para avaliar qual dos algoritmos apresenta melhor desempenho. Os autores tomam como métricas a taxa de alertas FP e VN, a precisão, a recuperação e a precisão da detecção, pois alegam que existe uma compensação entre as métricas.

Um dos principais objetivos de Robson e Thomas (2015) é analisar algoritmos que minimizam os erros de Tipo I (VN) e Tipo II (FP) e maximizam a precisão e o *recall*, e essas características são submetidas ao *Multicriteria Decision Problems* (MCDA) do *software Visual Promethee* (http://www.promethee-gaia.net) que avalia as ações com base em critérios pré-definidos. O sistema funciona baseado em *ranking* e ao final de todo processamento é gerado uma tabela com a classificação dos algoritmos com melhor desempenho.

Apesar dos autores trabalharem com dados de identificação de ataques de inundação de rede, eles não classificam ataques DDoS. Todos os dados coletados são utilizados apenas para a classificação dos algoritmos de aprendizado de máquina.

3.6 Predição de anomalia no tráfego de rede usando filtro de Wiener adaptável e modelagem ARMA

O objetivo do trabalho de Celenk *et al.* (2008) é prever anomalias de dados de um fluxo de rede antes que sejam detectadas pelos métodos existentes, para tanto, utilizam a ferramenta de monitoramento Argus (http://www.qosient.com/argus/) na captura e estudo do comportamento da rede.

A proposta de trabalho dos autores é a criação de um método de detecção de anomalias através da análise estatística do fluxo de rede, para tanto é utilizada a filtragem adaptativa de Wiener para a redução do tráfego normal e o modelo do tipo *Auto Regressive Moving Average* (ARMA) para a identificação de sinais anômalos no tráfego.

De acordo com Ehlers (2007), o modelo ARMA é a combinação dos modelos *Autoregressive* (AR) e *Moving Average* (MA) e formam uma classe de modelos bastante úteis para descrever dados de séries temporais, já que as previsões são obtidas diretamente das equações. Celenk *et al.* (2008) afirmam que a aplicação desses métodos possibilita a prevenção de anomalias antes dos detectores de intrusão, porém, por não analisar características específicas como endereço IP e protocolo, a anomalia é detectada, mas são necessários estudos adicionais para a identificação do ataque, sua origem ou destino.

3.7 Modelo de Previsão da Probabilidade Discreta de Distribuição do Ataque DoS

A proposta dos autores é criação de um modelo de predição de ataques DoS baseado no método e agrupamento de algoritmos genéticos e métodos bayesianos.

Na visão de Zhao, Yin e Long (2008) a análise de agrupamento é um método de classificação não supervisionado, utilizado para obter a distribuição de dados ou o pré-processamento de dados sem classe atribuída. O objetivo do agrupamento é a busca de melhores linhas, para as quais as distâncias são mínimas a partir de pontos de amostragem O algoritmo genético é uma técnica de busca baseada em

dados trabalha com conjuntos de parâmetros, população, probabilidades, informações de custo e recompensa que são adotados para implementar a otimização dos métodos de agrupamento. O método Bayesiano de tomada de decisão é um método básico no reconhecimento de padrões estatísticos utilizado para converter a probabilidade prévia de cada amostra de subclasse, por meio da observação das distâncias entre o ponto de predição e cada ponto de amostragem.
Segundo os autores, o modelo de predição do artigo se dá por meio da distribuição de probabilidade discreta composta por várias séries de quantidade de ataque calculadas no decorrer do trabalho.

3.8 Gerenciamento de segurança de rede com agrupamento de padrões de tráfego

De acordo com Chiou (2014), existem muitas inundações de *malwares* na Internet, e a tarefa de identificar qual cliente foi infectado por qual tipo de ameaça é extremamente difícil
Segundo Chiou (2014), quando uma máquina torna-se vítima de uma *botnet*, ela tenta localizar um servidor de Comando e Controle (C&C) do qual a vítima irá baixar e executar códigos maliciosos. As máquinas se conectaram ao C&C por meio de nomes de domínios aleatórios. As vítimas geram dezenas de milhares de nomes com o algoritmo *Domain Generation Algorithm* (DGA). O *botmaster* utiliza-se da aleatoriedade de nomes para não ser identificado pelas listas negras. Sempre que descoberto, altera rapidamente o nome de domínio e se esquiva da detecção.
Todo o fluxo de trabalho de Chiou (2014) é baseado em *logs* do DNS.
O trabalho de Chiou (2014) permite apenas a análise de um dia de tráfego. Além disso, por ser baseado em *logs* de DNS exige poder computacional suficiente para a realização de análises de *logs* de longos períodos.

3.9 Detecção de *botnet* com base na análise de tráfego

O objetivo do trabalho de Kugisaki *et al.* (2007) é a detecção de computadores infectados por uma *botnet* por meio da análise do comportamento da rede. Semelhante ao trabalho de Chiou (2014), o autor reforça a ideia de que o ataque de uma *botnet* necessita da figura de um servidor que repasse as instruções recebidas de um atacante. No trabalho de Kugisaki *et al.* (2007), as operações dos clientes que se conectam ao servidor são monitoradas e observam-se as operações

consideradas comuns, as operações duvidosas, e os padrões de semelhança no tráfego de rede em determinados intervalos de tempo.

Segundo Kugisaki *et al.* (2007), o *Internet Relay Chat* (IRC) é o protocolo de comunicação utilizado pelos invasores para a troca de instruções entre os *botnets*. O IRC possui um mecanismo de *multicast* que possibilita o envio de instruções para muitos *botnets* simultaneamente

Kugisaki *et al.* (2007), utiliza dois métodos para detecção de *botnets.* O primeiro é baseado em assinaturas, e o segundo é baseado no estudo do comportamento de clientes IRC. De acordo com Kugisaki *et al.* (2007), clientes originários de uma *botnet* transmitem as informações de largura de banda e informações dos *hosts* para outros servidores, fato que não acontece em transmissões de mensagens de clientes de bate-papo. Os autores também observaram o comportamento da transmissão de mensagens nos intervalos de tempo de até 400 segundos e constataram que os gráficos de comunicações para o servidor IRC gerados por clientes de uma *botnet* diferem de gráficos gerados por clientes comuns.

Toda monitoração é realizada nas portas de comunicação padrão do IRC, ou seja, portas TCP 6666 até 6669, sendo assim, a alteração de qualquer uma dessas portas por um atacante resulta em falha no sistema, impossibilitando a monitoração. O mesmo ocorre caso o tráfego seja criptografado, pois a monitoração não terá acesso aos dados do tráfego real da rede.

3.10 Identificação de padrões de *botnets* por meio de monitoramento de rede

A proposta de Tegeler *et al.* (2012) é a criação de um sistema chamado *Botfinder* capaz de detectar *bonets* por meio do monitoramento do tráfego de rede. Segundo os autores, conexões do servidor de comando de determinada *botnet* de uma mesma família possuem certos padrões regulares.

O *Botfinder* trabalha em duas fases: Na primeira fase aprende as propriedades estatísticas do tráfego de diferentes famílias de *botnet,* e na segunda fase cria modelos que podem identificar um tráfego semelhante.

Segundo Tegeler *et al.* (2012), o *Botfinder* não inspeciona cargas úteis, isto é, não inspeciona os pacotes de redes e sim o comportamento do tráfego de rede, por esse motivo, detecta ataques originários de tráfego criptografado. O protótipo do *Botfinder* é capaz de operar em redes de alto desempenho com centenas de milhares de hosts em tempo real.

3.11 Comparação

A Tabela 1 foi elaborada com o intuito de servir como comparativo. Nele são mostradas algumas características dos principais trabalhos relacionados.

Tabela 1 - Comparação dos trabalhos relacionados

Trabalhos relacionados	**Características de comparação**												**Legenda**
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**		
Gupta; Kumar e Braham (2013)	**x**		**x**		**x**	**x**							A – Aprendizado de máquina
Lin *et al.* (2013)	**x**		**x**	**x**	**x**				**x**				B – Previsão de ataques
Hock e Kappes (2014)	**x**		**x**	**x**	**x**	**x**							C – Esquema de *Ranking*
Bierma, Doak e Hudson (2016)	**x**		**x**										D – Identificação de FP
Robson e Thomas (2015)	**x**		**x**		**x**								E – Análise da camada TCP
Celenk *et al.* (2008)		**x**			**x**	**x**							F – Detecção de anomalias
Zhao, Yin e Long (2008)		**x**					**x**			**x**			G – Dados reais
Chiou (2014)		**x**			**x**		**x**		**x**				H – Comportamento Humano
Kugisaki *et al.* (2007),					**x**		**x**	**x**	**x**				I – Identificação de Padrões
Tegeler *et al.* (2012),	**x**	**x**					**x**	**x**	**x**		**x**		J – Possui método de predicção
Danta (2018) - Este trabalho	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**	**x**				K – Detecta tráfego criptografado

Fonte: Elaborado pelo autor

Os trabalhos relacionados tem grande variedade de material sobre a previsão, prevenção e detecção de ataques por meio de técnicas de aprendizado de máquina, reconhecimentos de padrões e classificação por meio de *ranking* e votação

Este material além de reforçar os princípios elaborados para o cumprimento dos objetivos deste trabalho, cria oportunidades para o desenvolvimento de novas pesquisas. Na Tabela 1 os trabalhos relacionados encontram-se na primeira coluna e nas demais colunas encontram-se as características de comparação.

3.12 Resumo da Seção

Esse trabalho foi motivado por discussões sobre temas de aprendizado de máquina e ranking, técnicas de detecção de intrusão e reconhecimento de padrões, para melhor entendimento os temas foram agrupados pela principais técnicas estudadas.

A Tabela 1 mostra os principais autores que influenciam esse trabalho e mostra um quadro comparativo entre os trabalhos relacionados e onze características de comparação, na tabela são assinaladas as características que os trabalhos tem em comum e pontos que não são cobertos por esse trabalho. No decorrer da seção são detalhados os principais trabalhos relacionados.

4 PROPOSTA DE TRABALHO

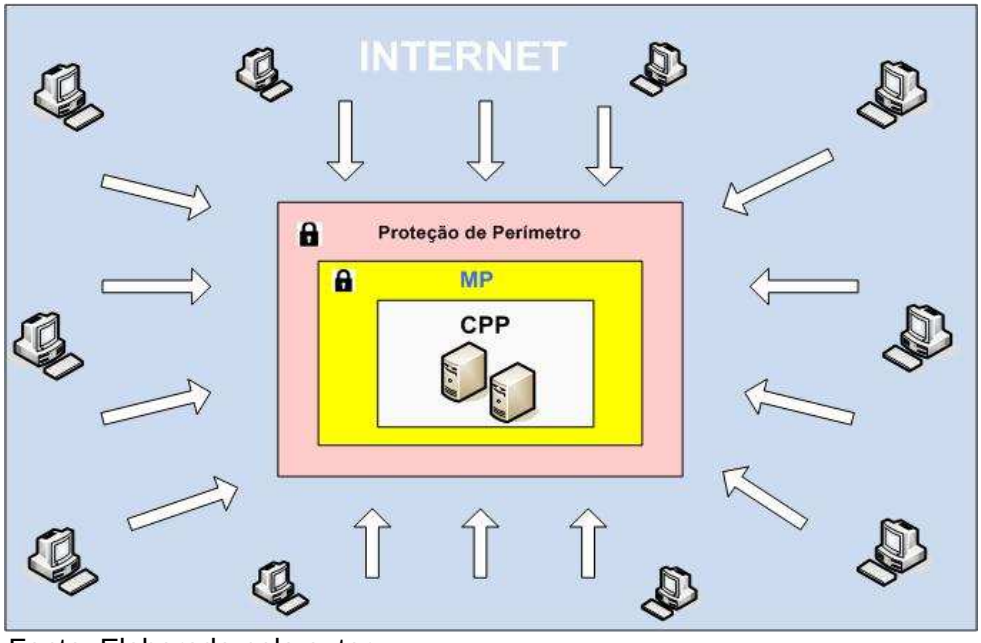
Nesta seção são apresentados os conceitos aplicados e os detalhes de operação do Método Preventivo (MP). Essa seção está dividida em cinco subseções onde, as quatro primeiras tem por objetivo explicar o que é e como funciona o MP e discorre explicitamente sobre cada uma das camadas que compõem o citado método: Tratamento de Dados, Aprendizado e Resultados. Já a última subseção apresenta um resumo geral dos assuntos apresentados nesta seção.

4.1 O Método

O MP é um método preventivo baseado nos conceitos de aprendizado de máquina em tempo real, capaz de melhorar a precisão e o tempo da identificação e do bloqueio de consultas maliciosas, tomando como base a análise de conexões e um conjunto de regras pré-definidas.

O CPP possui uma proteção de perímetro, porém, essa proteção não faz distinção entre consultas de clientes suspeitos ou não. A Figura 1 apresenta uma visão geral do cenário das consultas do CPP com a adição do MP.

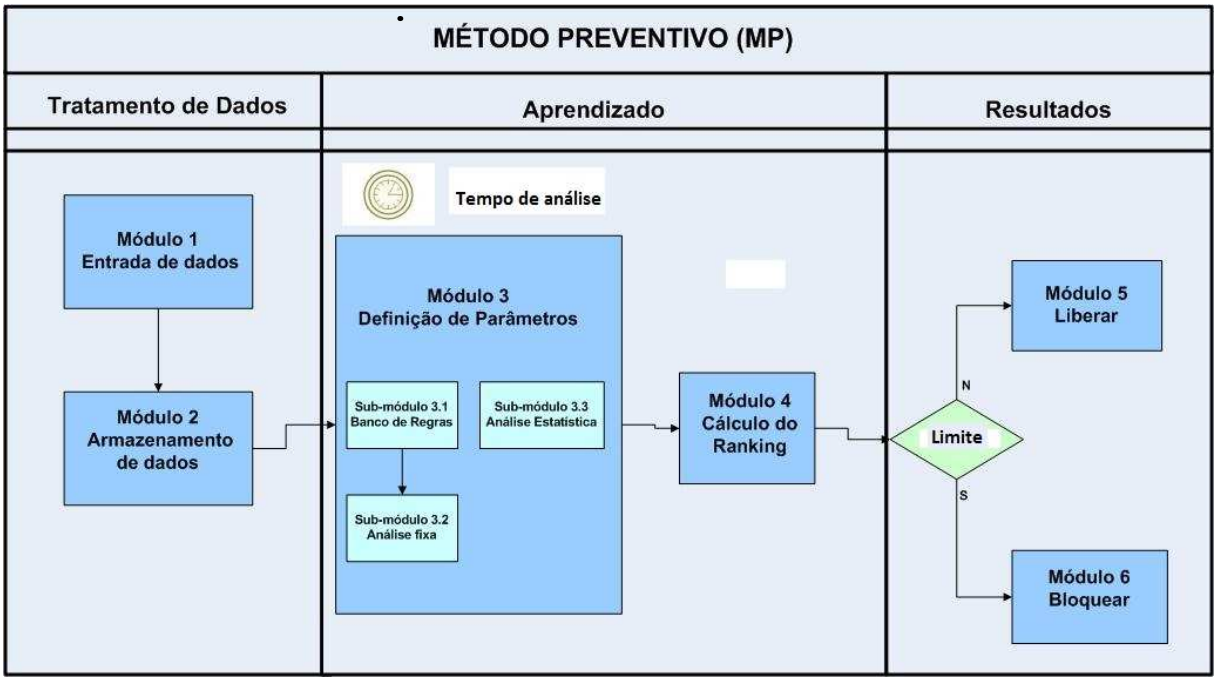
Figura 1 - Proposta ilustrativa do MP



Fonte: Elaborado pelo autor

A Figura 1 mostra o MP como uma proteção adicional ao CPP. Como pode ser visto na Figura 2, o MP é composto por diversas camadas e módulos que executam cada fase do processo.

Figura 2 - Estrutura do MP



Fonte: Elaborado pelo autor

A Figura 2 representa a estrutura do MP, e essa estrutura está dividida nas seguintes camadas: Tratamento de Dados, Aprendizado e Resultados.

Cada camada possui 2 módulos. Na camada de tratamento de dados, existem: o Módulo 1 (Entrada de Dados) e o Módulo 2 (Armazenamento de Dados). A camada de Aprendizado é composta pelo Módulo 3 (Definição de Parâmetros), sendo que esse módulo é subdividido nos Sub-módulos Banco de regras, Análise fixa e Análise estatística. O relógio no topo do módulo 3 representa os cenários de tempo de armazenamento utilizados no processo de validação do trabalho.

O Módulo 4 (Cálculo de *Ranking*) também faz parte da camada de Aprendizado. Na camada de Resultados, dependendo do valor do limite, o tráfego pode ser liberado por meio do Módulo 5 (Liberar) ou bloqueado por meio do Módulo 6 (Bloquear).

4.2 Camada de Tratamento de Dados

A camada de tratamento de dados discorre sobre como o dado é recebido, coletado, tratado e armazenado para que os demais módulos do MP possam então executar a análise e tomar uma ação com base em determinadas características e

comportamentos. Essa camada é composta por dois módulos que são responsáveis por preparar os dados para a camada de aprendizado.

No módulo 1 capturam-se as consultas realizadas pelos clientes que acessam o *webservice*. Os dados das consultas passam por uma análise prévia baseada na repetição do endereço IP de origem e são ordenados tendo essa premissa como base. Quando um determinado endereço IP de origem realiza mais de 30 consultas em um período mínimo de 5 segundos isto é considerado suspeito e os dados são enviados para análise – esta ação gera agilidade pois apenas as consultas suspeitas são analisadas. De acordo com Bussab e Morettin (2004) uma amostra de 30 elementos tem uma quantidade mínima para o cálculo de variância.

Tabela 2 – Descrição de campos do *webservice* utilizados na análise

Campo	**Descrição**	**Exemplo**
Data	AA:MM:DD	2016-05-31
Horário	HH:Min:Seg:Cent-Seg	15:04:00.097
IP de origem	xxx.yyy.zzz.www	12.345.678.90
CEP de origem	Numérico 8 dígitos	33114455
CEP de destino	Numérico 8 dígitos	1155678
Serviço	Código do serviço utilizado no CPP	40444
Peso	Peso (KG) da encomenda	8

Fonte: Elaborado pelo autor

Os seis campos mostrados na Tabela 2 são enviados pelo cliente a cada consulta ao CPP.

O módulo 2 é responsável pelo armazenamento dos dados em memória. Esse armazenamento ocorre em cinco cenários divididos em três blocos, mostrados na

Tabela 3 e exemplificados na sequência:

Tabela 3 - Cenário de intervalos de tempo

Cenários	**Bloco inicial**	**Bloco intermediário**	**Bloco final**
Cenário 1	5	10	20
Cenário 2	10	20	40
Cenário 3	15	30	60
Cenário 4	20	40	60
Cenário 5	40	50	60

Fonte: Elaborado pelo autor

Na Tabela 3 a primeira coluna equivale ao nome do cenário e as colunas subsequentes mostram o tempo em segundos para cada bloco de captura. Por exemplo, para o cenário 1, os dados são analisados dentro do intervalo de 0 - 5 s, gerando um bloco de dados inicial. Em seguida, os dados são analisados dentro do intervalo de 0 – 10 s, formando um segundo bloco de dados. Por último, os dados são analisados no intervalo de 0 – 20 s, formando o bloco de dados final. Como os blocos são baseados no tempo, a quantidade de registros em cada um deles é variável. Os dados pertencem a múltiplos clientes e são avaliados na Camada de Aprendizado.

4.3 Camada de Aprendizado

Na camada de aprendizado são executadas as funcionalidades reativas à inteligência do sistema. Nessa camada são realizados os principais cálculos para análise dos dados. O resultado é então encaminhado para a camada de Resultados. Esta camada é composta pelos módulos 3 (Definição de Parâmetros) e 4 (Cálculo do *Ranking*).

4.3.1 Módulo 3 – Definição de Parâmetros

O Módulo 3 descreve de maneira detalhada as regras, parâmetros e cálculos utilizados no MP apresentando três sub-módulos: 3.1 Banco de Regras, 3.2 Análise Fixa e 3.3 Análise Estatística.

4.3.1.1 Sub-módulo 3.1 - Banco de Regras

As consultas endereçadas ao *webservice* podem acontecer em alto volume (realizada por empresas) e tem por objetivo identificar o valor de frete e o prazo de entrega de mercadorias. Eventuais consultas com alto volume de tráfego podem ser consideradas maliciosas quando visam deixar os serviços lentos ou indisponíveis. Em alguns casos essas consultas maliciosas são realizadas por *scripts* automáticos.
O Banco de Regras é um conjunto de parâmetros, identificados por meio da análise do tráfego, capaz de identificar um ataque. A Tabela 4 mostra uma breve explicação do conteúdo do Banco de Regras.

Tabela 4 - Conteúdo do Banco de Regras

Regra	**Campo**	**Descrição**
Regra 1	IP de origem	Consultas sequenciais com repetição do endereço IP
Regra 2	CEP de origem	Consultas sequenciais com repetição do CEP de origem
Regra 3	CEP de destino	Consultas sequenciais com incremento sequencial do CEP de destino
Regra 4	Serviço	Consultas sequenciais com repetição do código de serviço
Regra 5	Peso	Consultas sequenciais com repetição do valor do peso

Fonte: Elaborado pelo autor

Como visto na Tabela 4, determinadas consultas sequenciais podem indicar comportamento suspeito. Por exemplo, um evento de consulta que tem o mesmo endereço IP e o mesmo CEP de origem para diversos CEP de destino pode significar um evento suspeito. Se nesta mesma consulta também for constatada a utilização de um mesmo código de serviço ou de um mesmo peso, a suspeita se torna maior.
A regra 1 é a primeira sinalização de alerta de uma consulta suspeita (não indica necessariamente um ataque, apenas sinaliza), e nela é identificado que um mesmo endereço IP de origem está realizando várias consultas em um curto espaço de tempo. A regra 2 sinaliza que estão ocorrendo várias consultas com o mesmo CEP de origem. Porém, somente esta informação, quando não atrelada à repetição do endereço IP de origem, não caracteriza necessariamente uma consulta suspeita.

A regra 3 demonstra que se em consultas sequenciais o CEP de destino é consultado com valores incrementais (por exemplo: 00000001, 00000002,...), isto pode indicar um comportamento malicioso, já que essa característica não é padrão em consultas normais. A regra 4 representa a repetição do código do serviço em várias consultas. Da mesma forma que a regra 2, essa informação isolada não caracteriza uma consulta suspeita. O mesmo ocorre com a regra 5, que representa a repetição do valor do peso em várias consultas.
A partir desse ponto são realizados os cálculos da análise fixa.

4.3.1.2 Sub-módulo 3.2 Análise fixa

Basicamente a análise fixa pode ser enquadrada como um processo de reconhecimento de padrões. Ela é utilizada para determinar o quão suspeita pode ser uma consulta ao *webservice*, tendo como base o Banco de Regras.
É importante observar que o aumento no número de indicadores presentes no Banco de Regras aumenta também a probabilidade da consulta ser suspeita. Portanto, para classificar o potencial suspeito, é atribuído um valor ponderado para cada regra. A ponderação total, representada pela soma das ponderações individuais classifica o nível de suspeição da consulta. Por definição, o valor máximo da somatória das ponderações de todas as regras é 1.
A regra 1 é a mais representativa no processo de investigação e pode identificar quem está realizando o ataque, por isso recebe a maior ponderação: 0,3. As regras 2, 3 e 4 isoladamente não indicam consultas suspeitas, mas somadas à regra 1 indicam possível comportamento malicioso nas consultas ao *webservice* e, portanto, recebem uma ponderação de 0,2 cada uma. A regra 5 tem grande possibilidade de ocorrer em situações normais. Em uma sequência de consultas legítimas é normal que o peso do objeto a ser enviado seja sempre o mesmo. Por exemplo: um cliente quer postar 100 envelopes para vários destinatários. Há uma folha de papel em cada envelope. Este cliente deve fazer no mínimo 100 consultas todas com o mesmo peso, mesmo assim não se trata de um ataque. Justamente por essa regra ser subjetiva ela recebe o menor valor ponderado: 0,1.
Um detalhe importante a ser observado é que o cálculo com as regras da análise fixa só é feito quando a consulta tem no mínimo 30 ocorrências no tempo mínimo de 5 segundos para um mesmo cliente. Note que na análise fixa a amostra inicial de 5 s é suficiente para definir a suspeição da consulta e não é preciso coletar outro

conjunto de amostras do mesmo cliente. Perceba que na análise estatística existe necessidade de mais duas coletas (vide Sub-módulo 3.3 Análise Estatística).

Para melhor entendimento do cálculo da análise fixa, a somatória dos valores ponderados é chamada de (S). A exemplificação do cálculo pode ser visto na Tabela 5.

Tabela 5 - Exemplo do cálculo da análise fixa – dados fictícios

Cliente	**Regra 1(0,3)**	**Regra 2(0,2)**	**Regra 3 (0,2)**	**Regra 4 (0,2)**	**Regra 5 (0,1)**	**(S)**
Cliente 1	X	X		X		0,7
Cliente 2	X	X	X		X	0,8
Cliente 3	X	X	X	X	X	1
Cliente 4	X				X	0,4
Cliente 5	X	X		X		0,7

Fonte: Elaborado pelo autor

A Tabela 5 mostra as regras definidas no Banco de Regras que coincidem com as consultas de cada cliente e suas respectivas ponderações. A coluna (S) mostra a somatória dos valores ponderados de cada cliente.

Ao final é necessário definir o quanto uma determinada consulta é considerada suspeita pela análise fixa. Para isso um mínimo é estabelecido. O valor mínimo de (S) tem como base o cumprimento da regra com o maior peso ponderado (Regra 1 = 0,3) somado ao cumprimento de pelo menos uma das regras com peso ponderado igual a 0,2 (Regra 2, 3 ou 4). Dentro dessa lógica, para ser considerado suspeito na análise fixa o valor mínimo de (S) = 0,5.

É perfeitamente possível que o valor de (S) seja inferior a 0,5. Neste caso, a consulta não é considerada suspeita pela análise fixa. Por exemplo, uma consulta que tenha cumprido apenas a Regra 1 (peso ponderado = 0,3) e a Regra 5 (peso ponderado = 0,1) possui (S) = 0,4. Entretanto, apesar de não ser considerada suspeita, a consulta deve passar pela análise estatística e ter o *ranking* calculado.

4.3.1.3 Sub-módulo 3.3 Análise estatística

Como clientes legítimos e suspeitos concorrem pelo acesso ao *webservice* ao mesmo tempo, a análise estatística busca separá-los com base no comportamento de um conjunto de consultas de um mesmo cliente.

O primeiro passo é capturar amostras de dados (tráfego de rede) com três blocos de tempo (vide cenários da Tabela 3). Essas amostras contém consultas de diversos

clientes provenientes da análise fixa com mais de 30 consultas em um intervalo de 5 s. A ideia é realizar alguns cálculos estatísticos (média e variância) com a frequência de consultas realizadas sequencialmente no tempo de um segundo - dentro dessas amostras cada consulta de um cliente específico é tratada de forma binomial como válida (valor 1), enquanto uma consulta de outro cliente qualquer é tratada como nula (valor 0). As consultas dos clientes que ocorreram mais de uma vez no mesmo segundo são somadas. Ao final, para os **três blocos** da amostra há um conjunto booleano de valores válidos e inválidos para **cada cliente**.

No próximo passo calcula-se a média aritmética (μ) e a variância (σ^2) da quantidade de consultas válidas (c) realizadas para cada cliente em cada bloco:

$$\mu = \sum_{c=1}^{n} \frac{1}{n} c \qquad (2)$$

Onde: n corresponde ao tempo em segundos de cada bloco de dados (vide cenários da Tabela 3).

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \mu)^2 \qquad (3)$$

Na sequência calcula-se a razão ***(ratio)*** entre a média (μ) e a variância (σ^2). Isso é importante para se verificar se os dados estão dentro da distribuição de Poisson, que tem como característica fundamental os mesmos valores de média e variância (consequentemente, a razão entre os dois valores, na distribuição de Poisson, deve ser próximo de 1: $ratio=\mu/\sigma^2=1$).

Distribuições de contagem como a Poisson e a Binomial Negativa são usadas para descobrir um número de eventos independentes que ocorrem em um determinado período de tempo. Normalmente essas distribuições são usadas quando a amostra (n) é grande e a probabilidade (p) de ocorrência de um evento é baixa.

Levando-se em conta uma **ocorrência válida** como um valor positivo igual a 1 dentro de um determinado intervalo de tempo, a premissa seguida nesta proposta é que um conjunto de consultas considerado normal tende a ter uma baixa quantidade de **ocorrências válidas** e está muito próximo da distribuição de Poisson *(ratio=μ/σ²=1)*. O inverso ocorre quando a consulta é maliciosa, ou seja, a quantidade de **ocorrências válidas** dentro de um determinado intervalo tende a ser maior. Nesse caso, o valor do *ratio* está distante de 1. Um exemplo pode ser visto na Tabela 6.

Tabela 6 - Exemplo do cálculo da Análise estatística – dados fictícios

Consultas do Cliente	
Hora	**Soma das consultas do mesmo cliente**
17:04:21	1
17:04:22	1
17:04:23	5
17:04:24	1
17:04:25	1
17:04:26	2
17:04:27	7
17:04:28	1
17:04:29	2
17:04:30	3
17:04:31	1
17:04:32	3
17:04:33	5
17:04:34	3
17:04:35	3
Média (μ)	2,6
Variância (σ^2)	3,04
Ratio	0,76

Fonte: Elaborado pelo autor

A Tabela 6 exemplifica o cálculo da análise estatística de um bloco de um cenário de consultas de um determinado cliente. Nos exemplos da Tabela 6, as consultas tem início às 17:04:21h e término às 17:04:35h e são agrupadas por segundo. A Tabela 6 também mostra os valores calculados de média (μ), variância (σ^2) e *ratio*.

O próximo passo é verificar se o cliente é estatisticamente suspeito. Na análise estatística, consideram-se clientes suspeitos aqueles cujo valor de *ratio* seja menor que 0,5 e maior que 1,5, ou seja, distante pelo menos 0,5 ponto de 1.

Um peso estatístico, chamado de peso-e, é atribuído a cada bloco de dados (vide cenários da Tabela 3) de um mesmo cliente, de acordo com o seguinte critério: se *ratio* < 0,5 ou *ratio* > 1,5, o bloco recebe peso-e=1, caso contrário peso-e=0. Calcula-se então a média dos peso-e dos três blocos do cliente, cujos valores podem ser: 0; 0,33; 0,66; e 1. Um cliente com média de peso-e=0,33 (menor valor acima de 0) é considerado estatisticamente suspeito.

Na Tabela 7 é possível acompanhar os cálculos realizados para a identificação dos clientes estatisticamente suspeitos.

Tabela 7 - Identificação de clientes estatisticamente suspeitos – dados fictícios

	Critério para peso-e=1: (*ratio* < 0,5 ou *ratio* > 1,5)						
Clientes	Bloco 1		Bloco 2		Bloco 3		
	Ratio	Peso-e	*Ratio*	Peso-e	*Ratio*	Peso-e	Média
Cliente 1	1,00	0	0,62	0	0,84	0	0,00
Cliente 2	1,30	0	1,55	1	1,34	0	0,33
Cliente 3	8,27	1	10,88	1	11,93	1	1,00
Cliente 4	1,00	0	7,33	1	5,00	1	0,66
Cliente 5	7,54	1	5,96	1	5,21	1	1,00
Cliente 6	1,08	0	0,92	0	1,05	0	0,00

Fonte: Elaborado pelo autor

A Tabela 7 mostra os valores de *ratio* e peso-e para seis diferentes clientes. A média dos pesos-e dos blocos por cliente é mostrada na coluna Média.

4.3.2 Módulo 4 Cálculo do *Ranking*

Nesse módulo ocorre o cálculo do *ranking* que é utilizado para a classificação dos dados previamente analisados pela análise fixa e pela análise estatística. Para o cálculo do *ranking* são somados os resultados dos cálculos da análise fixa e da análise estatística.

Um valor de limite é definido para o *ranking,* de forma que a consulta de um determinado cliente seja considerada realmente maliciosa, adota-se a soma de (S) acrescentando também o peso-e. Considerando o valor mínimo (S) = 0,5 para que uma consulta seja considerada suspeita na análise fixa e o peso-e mínimo de 0,33 definido como suspeito na análise estatística, o limite mínimo do *ranking* ≥ 0,83 é usado para determinar se uma consulta realizada por um determinado cliente é maliciosa.

É importante notar que o limite do *ranking* ≥ 0,83 pode ser alcançado apenas pela análise fixa ou mesmo apenas pela análise estatística, já que cada uma pode alcançar um valor máximo de 1. Contudo, o valor limite do *ranking* pode variar de acordo com as necessidades de segurança que o ambiente exige.

4.4 Camada de Resultados

Nesta camada são realizados os bloqueios ou liberações de consultas a partir do limite do *ranking* definido.

4.4.1 Módulo 5 Liberar

Nesse módulo ocorre a liberação de consultas válidas, caso a consulta tenha o resultado da soma de (S) mais o peso-e menor que o valor limite do *ranking*.

4.4.2 Módulo 6 Bloquear

A execução do bloqueio de consultas suspeitas ocorre quando o resultado da soma de (S) com a adição do peso-e (p) é maior ou igual ao valor limite do *ranking* estabelecido. O bloqueio é uma ação temporária, com tempo (t), definido pelo administrador de acordo com a conveniência de segurança do ambiente.

4.5 Resumo da Seção

A proposta desse trabalho é a criação de um Método Preventivo (MP). Este método é composto por três camadas, a saber: Tratamento de dados, Aprendizado e

Resultados. As camadas são divididas em módulos e sub-módulos que apresentam detalhes para implementação do MP. A primeira camada é composta pelos módulos de entrada de dados e armazenamento de dados. A segunda camada contém os módulos de definição de parâmetros e cálculo do *ranking* e os módulos de liberação e bloqueio de tráfego suspeito estão contidos na terceira camada. Toda esta estrutura é utilizada na seção 5.

5 EXPERIMENTO E ANÁLISE DE RESULTADOS

Esta seção tem como objetivo descrever o ambiente e a estrutura montada para a análise de resultados do MP

5.1 Introdução

A validação da proposta é realizada por meio de dados coletados previamente nos segmentos de rede onde estão hospedados os servidores do CPP e a comprovação dos resultados é obtida por meio de cálculos e análises aplicadas sobre os dados coletados.

Na subseção **5.2 Ambiente do CPP**, é apresentado o ambiente onde foram realizadas as capturas e os equipamentos e periféricos que compõe a topologia do CPP.

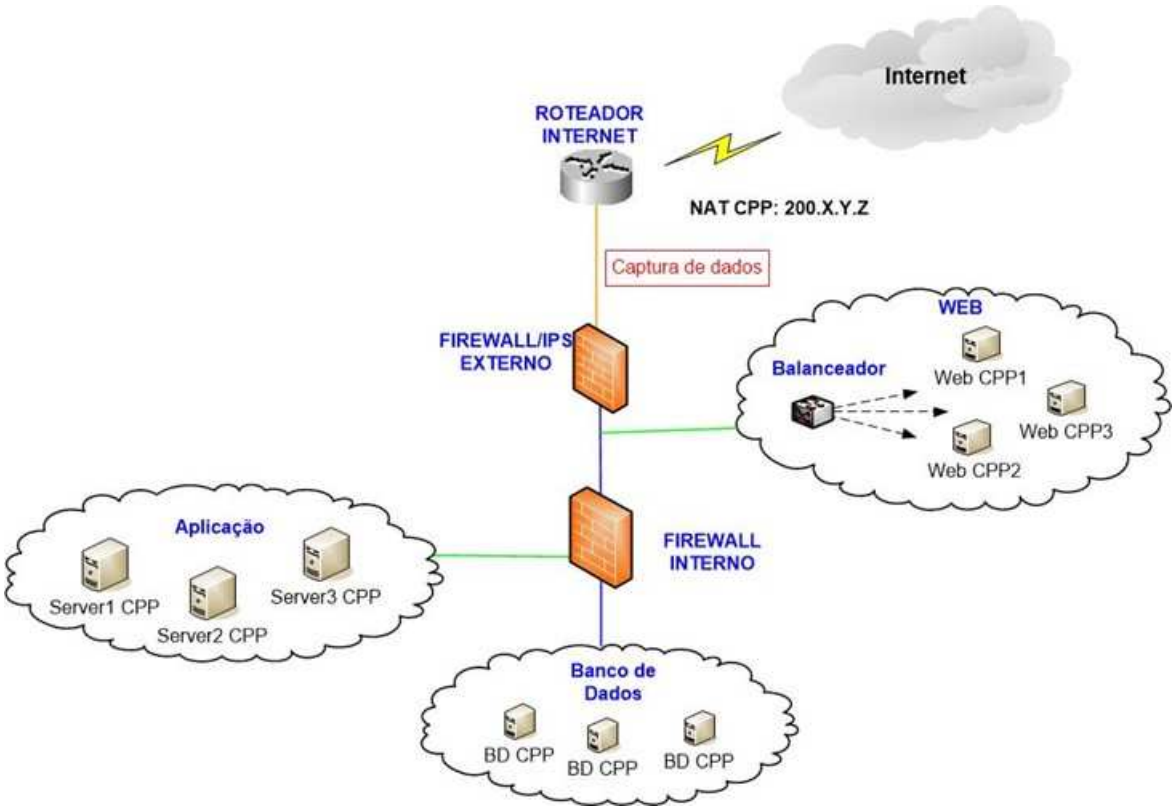
Na subseção **5.3 Experimentos realizados**, são descritos detalhes usados na implementação do experimento os quais são úteis para o entendimento dos resultados.

Na subseção **5.4 Coleta e análise dos resultados** são descritos os critérios e as fórmulas usadas para comprovar a efetividade do MP, os índices de bloqueio legítimos e os índices de FP.

5.2 Ambiente do CPP

Neste trabalho a validação da proposta é realizada com dados coletados em um ambiente real, na Figura 3 podem ser observados todos os equipamentos que compõe a solução do CPP.

Figura 3 - Topologia de rede do CPP



Fonte: Elaborado pelo autor

O ambiente do CPP é composto por 3 servidores de *webservice*, 3 servidores de aplicação, 3 servidores de banco de dados, 1 balanceador, 2 *firewall*s e 1 roteador. Todos os equipamentos estão em alta disponibilidade.

Os servidores que são acessados diretamente pelos clientes da Internet estão instalados em uma DMZ segmentada por um *firewall* externo e um *firewall* interno.

O *firewall externo* está conectado fisicamente ao roteador, esse equipamento é um *appliance* que possui as funcionalidades de filtro de pacotes e SDPI. Portanto, os servidores do CPP recebem apenas consultas direcionadas à porta correta e todos pacotes são analisados por assinaturas previamente habilitadas no SDPI.

Para aumentar o nível de segurança do CPP, os servidores de aplicação e os servidores de banco de dados estão instalados em duas DMZ distintas, enquanto no

firewall interno estão configuradas apenas as regras que permitem que os *webservices* se comuniquem com a aplicação e com o banco de dados.

A coleta de dados é realizado na interface externa do *firewall externo* por meio da ferramenta *tcpdump*, essa ferramenta apresenta uma expressão *booleana* que corresponde à descrição do conteúdo de pacotes em uma interface de rede. Por padrão, essa descrição é precedida por um carimbo de tempo, com horas, minutos, segundos e frações de segundo desde a meia-noite. O *tcpdump* pode ser executado com uma série de argumentos que permitem gravação e saída de dados customizada.

Por exemplo, o comando: **tcpdump -vvv -XX -w file.pcap -nNi eth3-01 host xxx.xxx.xxx.xxx**, possibilita a captura de dados detalhados em estado bruto, imprime o cabeçalho em hexadecimal ou ASCII e grava um arquivo de saída com extensão. pcap (compatíveis com o *sniffer* Wireshark). A captura é realizada para um *host* específico, na interface ligada diretamente ao roteador.

5.3 Experimentos e resultados

A proposta desse trabalho é a implementação de um método preventivo em tempo real. Os experimentos são realizados por meio da análise de *logs* de dados capturados entre o roteador e o *firewall externo*, conforme topologia mostrada na Figura 3. O experimento se divide em três fases: captura, análise dos dados e resultados.

5.3.1 Captura

A captura de dados é realizada no segmento de rede entre o *firewall externo* e o roteador. Para a comprovação do experimento são realizadas cinco capturas. Quatro delas nos meses de maio e junho de 2016 e a última realizada no dia 24 de novembro de 2017 durante o evento de compras promocionais Black Friday. Na

Tabela 8 pode ser observado o detalhamento das capturas realizadas.

Tabela 8 - Capturas de dados

Arquivo de captura	Data – Hora (Inicial)	Data – Hora (Final)	Tamanho do arquivo
Captura-1.pcap	19/05/2016 – 15:55:55	24/05/2016 – 09:38:31	113 GB
Captura-2.pcap	25/05/2016 – 15:24:15	27/05/2017 – 16:07:08	109 GB
Captura-3.pcap	31/05/2017 – 16:05:30	02/06/2017 – 16:11:54	99 GB
Captura-4.pcap	07/06/2016 – 17:05:30	10/06/2016 – 11:06:50	102 GB
Captura-5.pcap	24/11/2017 – 08:20:01	24/11/2017 – 16:12:00	14 GB

Fonte: Elaborado pelo autor

Na primeira coluna da Tabela 8 estão descritos os nomes dos arquivos de captura utilizados no experimento, na segunda e na terceira coluna estão descritos as datas e horários iniciais e finais da captura e na última coluna está mencionado o tamanho do arquivo de captura.

Os dados coletados são referentes ao protocolo TCP: *timestamp*, IP de origem, IP de destino, porta de origem, porta de destino e a área útil da camada de aplicação. Estes dados são organizados em ordem cronológica e separados em cinco cenários já descritos na Tabela 3.

5.3.2 Análise dos dados

Para efeito de análise de dados do experimento são selecionadas seis datas distintas na captura. Para efeito comparativo as datas foram agrupadas em dias da semana, duas quartas-feiras, duas quintas-feiras e duas sextas-feiras.

Tabela 9 - Períodos de análise

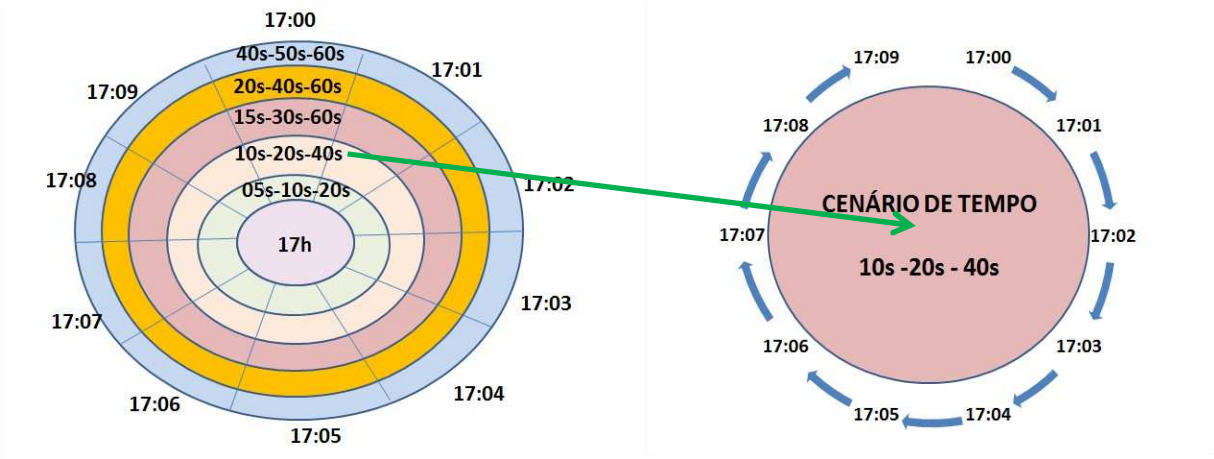
	Quarta-Feira	Quinta-Feira	Sexta-Feira
	01/06/2016 e 08/06/2016	26/05/2016 e 09/06/2016	20/05/2016 e 24/11/2017
10h	x	x	x
12h			x
15h	x	x	
17h	x	x	
23h	x	x	

Fonte: Elaborado pelo autor

Como pode ser observado na Tabela 9, as análises das quartas-feiras e das quintas-feiras são realizadas nos horários das 10h, 15h, 17h e 23h e as análises das sextas-

feiras são realizadas somente nos horários de 10h e 12h. Todas as análises são realizadas nos cenários definidos na Tabela 3.

Figura 4 - Exemplo do ciclo de análise



Fonte: Elaborado pelo autor

Na elipse da Figura 4, encontra-se a hora inicial da análise, as elipses seguintes representam os blocos de cada cenário de tempo de análise. Cada cenário está dividido em 10 partes, cada parte representa 1 minuto de análise (10 minutos no total). Para melhor entendimento é exemplifica a análise de um cenário específico com blocos de 10s, 20s e 40s, durante 10 minutos.

Após as análises, os dados são enviados para os cálculos. A Tabela 10 mostra exemplos de cinco cálculos realizados em um minuto de análise.

Tabela 10 - Exemplo de análise de dados

Análise realizada no período de: 01/06/2016 - 10h									
Análise 1 - 10:00 - 10:01									
	Cenário: 5-10-20				**Critério: peso-e=1 (raio < 0,5 ou ratio > 1,5)**				
Clientes	Fixa	Ratio1	Ratio2	Ratio3	peso-e	peso-e	peso-e	Média peso-e	ranking
Cliente 1	0,8	0,7028946	0,7538462	0,7729258	0	0	0	0,00	0,80
Cliente 2	0,9	4,1219512	4,586758	4,9541985	1	1	1	1,00	1,90
Análise 2 - 10:00 - 10:01									
	Cenário: 10-20-40				**Critério: peso-e=1 (raio < 0,5 ou ratio > 1,5)**				
Clientes	Fixa	Ratio1	Ratio2	Ratio3	peso-e	peso-e	peso-e	Média peso-e	ranking
Cliente 2	0,7	0,5460575	0,4733096	0,4350019	0	1	1	0,67	1,37
Análise 3 - 10:00 - 10:01									
	Cenários: 15-30-60				**Critério: peso-e=1 (raio < 0,5 ou ratio > 1,5)**				
Clientes	Fixa	Ratio1	Ratio2	Ratio3	peso-e	peso-e	peso-e	Média peso-e	ranking
Cliente 1	0,8	1,3473389	1,3095768	1,2491857	0	0	0	0,00	0,80
Análise 4 - 10:00 - 10:01									
	Cenário: 20-40-60				**Critério: peso-e=1 (raio < 0,5 ou ratio > 1,5)**				
Clientes	Fixa	Ratio1	Ratio2	Ratio3	peso-e	peso-e	peso-e	Média peso-e	ranking
Cliente 3	1	4,875	5,313253	5,7146597	1	1	1	1,00	2,00
Análise 5 - 10:00 - 10:01									
	Cenários: 40-50-60								
Clientes	Fixa	Ratio1	Ratio2	Ratio3	peso-e	peso-e	peso-e	Média peso-e	ranking
Cliente 3	1	4,7958237	5,0217391	5,4417476	1	1	1	1,00	2,00

Fonte: Elaborado pelo autor

No exemplo da Tabela 10, o experimento é referente a data de 01/06/2016 às 10h. A análise 1 refere-se aos eventos ocorridos no horário entre 10:00h e 10:01, no cenário de tempo de 5s, 10s e 20s. Nessa análise, quando ocorrem 30 eventos em 5 segundos para um mesmo cliente, o cálculo da análise fixa é realizado e gravado na coluna **fixa**. Em seguida, é realizada a análise estatística nos mesmos blocos de tempo (5s, 10s e 20s) e os resultados são gravadas nas colunas Ratio1, Ratio2 e Ratio3. Na sequência são calculados o peso-e de cada bloco e a média. Por fim, o *ranking* de cada cliente é calculado e comparado com o valor do Limite (0,83). Esse processo é repetido em todos os cenários a cada minuto subsequente. No exemplo da Tabela 10, quando o *ranking* ≥ 0,83 (em vermelho) o cliente é considerado suspeito e quando *ranking* < *0,83* (em verde) o cliente é considerado normal.

5.3.3 Resultados

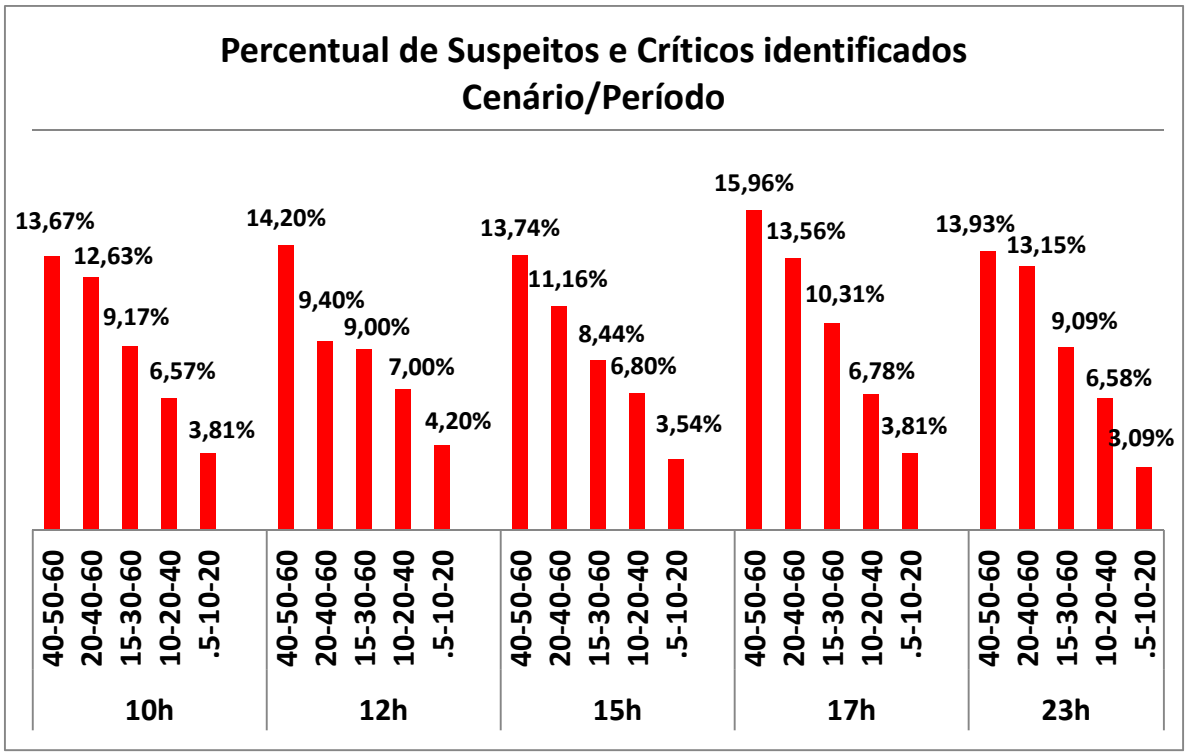
No método antigo o critério usado para detecção de consultas suspeitas ao CPP não é eficaz, o analista responsável pelo sistema identifica alto consumo de recursos (processamento e memória) nos servidores do CPP, solicita uma análise da equipe de segurança que, por meio de uma coleta no *firewall* por dez minutos identifica os dez primeiros IPs com maior número de conexões e efetua o bloqueio por tempo indeterminado. Esse critério possui dois grandes problemas, o primeiro é a demora de aproximadamente 30 minutos entre o início do ataque e o bloqueio dos IPs suspeitos e o segundo problema é o eventual bloqueio de clientes válidos.

O experimento analisa 30.584 clientes, 1.105.542 consultas no tempo total de 3 horas e 20 minutos de análise.

Para efeito de comparação, são contabilizados os eventos de uma sexta feira comum (20/05/2016 com 88379 consultas) e de uma Black Friday (24/11/2017 com 280562 consultas), que possui quase o triplo de consultas.

A Figura 5 mostra o percentual de eventos considerados suspeitos em relação ao total de eventos analisados agrupados por períodos e horários.

Figura 5 - Percentual de suspeitos e críticos por cenário



Fonte: Elaborado pelo autor

Pode-se perceber na Figura 5 que os maiores percentuais de clientes suspeitos ocorrem justamente nos cenários nos quais os blocos iniciais são maiores. Isso ocorre porque um tempo maior de detecção inicial significa uma análise estatística mais apurada para as consultas de cada cliente. Por exemplo, justamente por possuir o maior bloco inicial (40), o cenário 40-50-60 identifica o maior número de consultas suspeitas. As análises são realizadas às 10h, 12h, 15h, 17h e 23h, mas os resultados não mostram diferenças significativas nos percentuais devido aos horários.

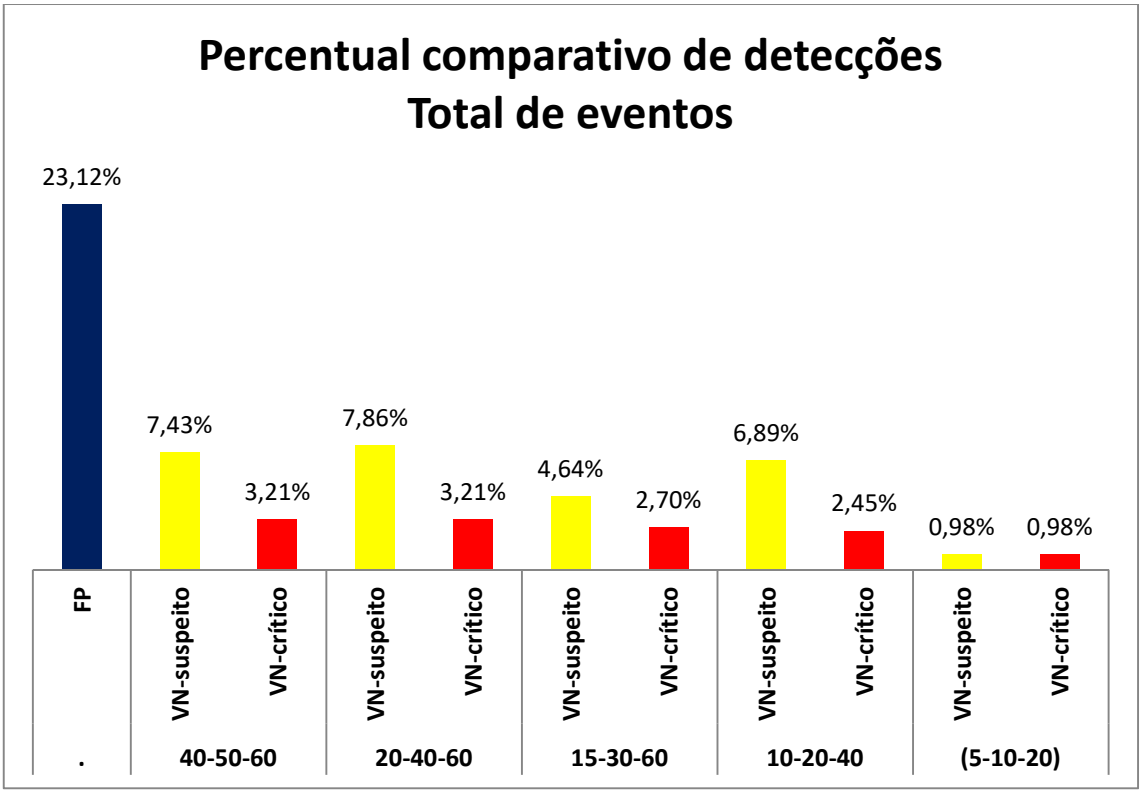
O cálculo dos percentuais de detecções FP e VN são baseados na comparação do método antigo de identificação de consultas suspeitas e com o MP. No método antigo são coletadas consultas por dez minutos. As consultas são agrupadas por cliente e classificadas em ordem decrescente de quantidade de consultas. Os primeiros dez clientes (com o maior números de consultas) são considerados suspeitos no método antigo.

O percentual de detecções FP é calculado com base nos dez clientes considerados suspeitos pelo método antigo. Dentre esses clientes, aqueles considerados normais (não suspeitos) pelo MP compõem o percentual de detecções FP. Já o percentual de detecções VN é representado pelos clientes maliciosos que não foram detectados como suspeitos pelo método antigo. Portanto, o percentual de detecções VN corresponde aos clientes que aparecem a partir da décima primeira posição do método antigo até o último cliente classificado como suspeito ou crítico pelo MP.

Os VN são divididos em duas categorias: VN-suspeitos são os clientes cujo *ranking*, apesar de considerado suspeito, não alcançou o valor máximo, já os VN-críticos atingiram a pontuação máxima e devem ser bloqueados de imediato

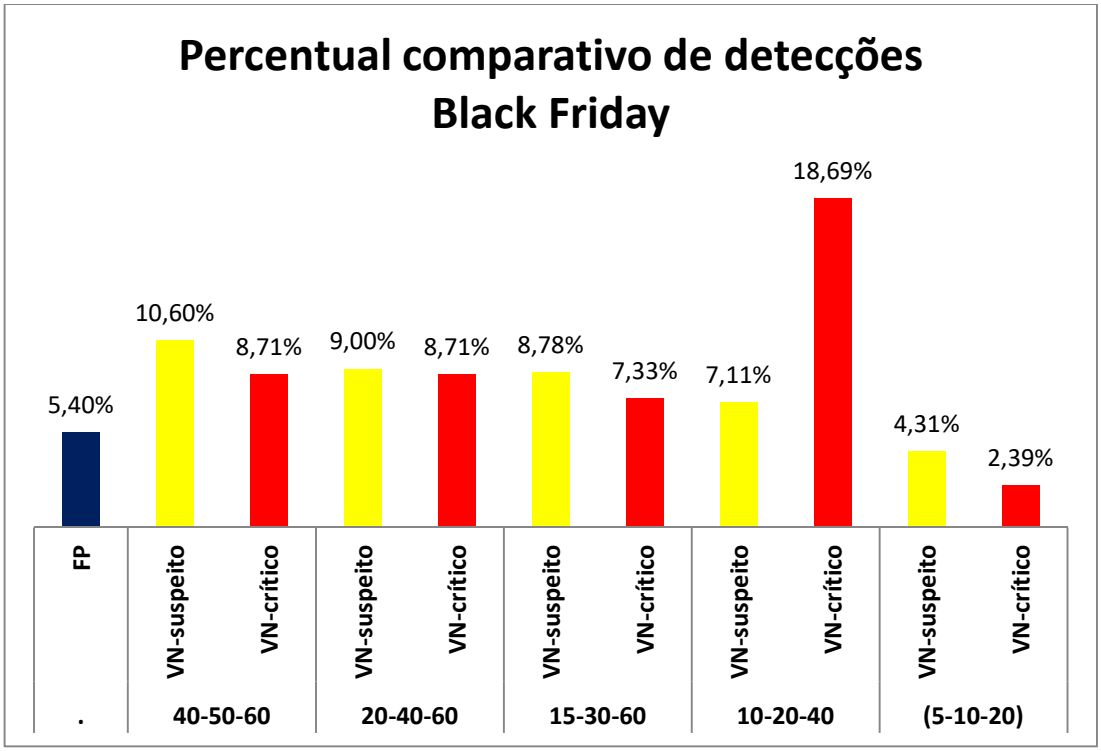
As Figuras 6 e 7 mostram quais cenários tem a maior propensão à ocorrência de VN-suspeito ou VN-crítico e FP, com relação ao total de clientes analisados.

Figura 6 - Percentual comparativo de detecções em tráfego normal



Fonte: Elaborado pelo autor

Figura 7 - Percentual comparativo de detecções na Black Friday



Fonte: Elaborado pelo autor

A Figura 6 mostra os resultados acumulados de todos os períodos analisados, exceto a Black Friday, percebe-se que o percentual de FP é bastante expressivo ultrapassando os 20% e os percentuais de VN-suspeito e VN-crítico, mantém percentuais equilibrados, variando em torno de 1% a maior ou a menor.

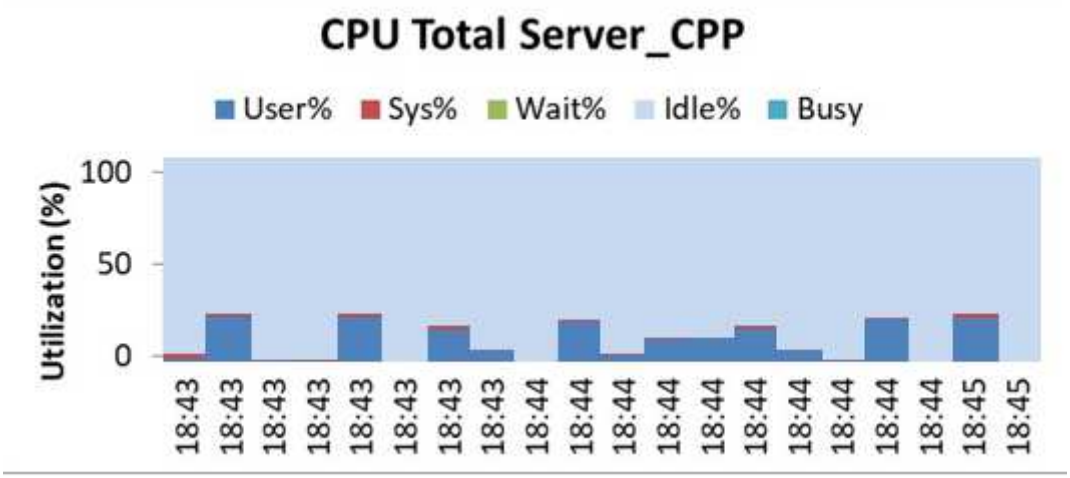
A Figura 7 mostra um baixo percentual de FP proveniente da Black Friday, pouco mais de 5%, essa redução de FP é acompanhado de dois fatores relevantes, o aumento significativo da quantidade de consultas, nesse caso, a triplicação das mesmas se comparadas aos dias normais, seguido pela utilização do mesmo critério de bloqueio de IPs utilizado pelo método antigo. A junção desses fatores causa a diluição dos resultados, possivelmente a alteração do critério de bloqueio dos dez IPs com maior quantidade de consultas para os trinta IPs com maior quantidade de consultas traga um percentual de FP mais coerente com a situação. O mesmo equilíbrio entre os VN-suspeitos e VN-críticos se repete no tráfego da Black Friday, exceto o VN-crítico do cenário de 10-20-40, que atinge o percentual de 18,69%. Para garantir a veracidade desse dado, o tráfego desse cenário foi reavaliado e percebeu-se que 50% dos clientes apresentavam falhas ou erros de configuração nas conexões e os outros 50% foram clientes que realmente atingiram a pontuação

máxima de *ranking.* Portanto, pelo MP, são clientes altamente suspeitos que devem ser bloqueados.

Este trabalho propõe a análise de tráfego em tempo real, porém, o experimento é realizado por meio da leitura e processamento de *logs* capturados e analisados via script. Para verificar o quanto o algoritmo proposto consome de processamento do CPP, em relação ao método antigo, é realizado um monitoramento de desempenho por *software* em um servidor Debian GNU/Linux 7.10 (wheezy) kernel 3.2.0-4-amd64, com 4 GB RAM, 2,6 GHz e 26,8 GB de disco.

De acordo com Griffiths (2003), o **nmon** é um software livre disponibilizado pela IBM para monitoramento de desempenho em modo interativo ou em modo de captura. Para simular o custo do MP o nmon é utilizado em modo captura por dois minutos. Simultaneamente o *script* do MP é executado em momentos aleatórios para verificação de seu desempenho. O nmon a*nalyser* é uma planilha customizada pela IBM que gera gráficos que auxiliam na interpretação dos dados gerados no nmon. O desempenho pode ser observado na Figura 8.

Figura 8 - Utilização total de CPU – Server_CPP



Fonte: Nmon

O nmon e o script são executados simultaneamente entre o horário das 18:43 e 18:45. No gráfico da Figura 5 percebe-se que a utilização de CPU não chega a 20%. Os picos de gravação em disco mostrados pela ferramenta chegam no máximo a 700 KB/sec. O gráfico da Figura 8 reflete o consumo de CPU do *webservice* durante a execução do script de extração de dados utilizado no experimento.

O resultado demonstrado pelo nmon *analyser* não reflete a realidade do MP em tempo real, mas serve como base para o dimensionamento de recursos. No entanto,

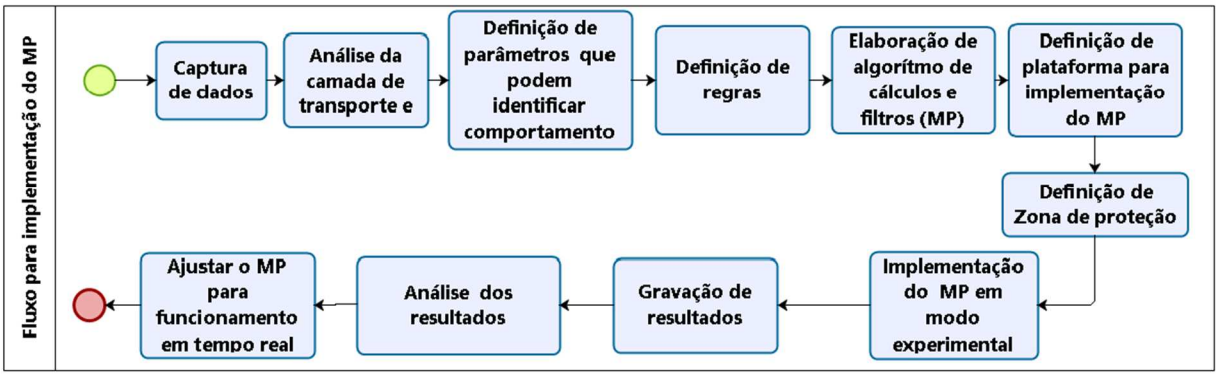
a adição de uma nova camada de análise no segmento de rede pode acarretar uma perda mínima de desempenho.

5.4 Resumo da Seção

O MP é dividido em 3 camadas. Os módulos 1 e 2 compõe o processo inicial para o tratamento dos dados que são utilizados nos demais módulos. O módulo 3 é o responsável pelas principais análises e definições para o funcionamento do MP, como definição de regras, reconhecimento de padrões e análise de comportamento dos clientes. No módulo 4 são realizados cálculos para a definição de um *ranking* que é utilizado para liberação ou bloqueio de consultas suspeitas.

Toda esta proposta de funcionamento do MP é validada no experimento, detalhado na seção 5 desse trabalho e a Figura 9 apresenta um fluxo a ser utilizado para a replicação do MP em outro ambiente.

Figura 9 - Fluxo para implementação do MP



Fonte: O autor

O fluxo representado na Figura 9 deve ser adaptado de acordo com o *webservice* que será analisado. A etapa inicial assemelha-se ao MP aplicado no CPP, pois trata-se da captura de dados, porém, a execução as demais etapas depende de um estudo criterioso das características do novo *webservice* e do comportamento do tráfego do mesmo.

6 CONCLUSÃO

Nessa seção são discutidas as conclusões finais e apresentadas algumas limitações e sugestões para trabalhos futuros.

Este trabalho se dedica a encontrar formas de diminuir o problema de lentidão e indisponibilidade ocasionado por ataques do tipo DDoS no webs*ervice* CPP. Para tanto, propõe um método preventivo de ataques de indisponibilidade, baseado em esquema de *ranking*, pesos e conceitos de aprendizado de máquina em tempo real, cujo principal objetivo é melhorar a precisão e o tempo de resposta da detecção de conexões suspeitas direcionadas ao *webservice*.

O atual critério adotado para o bloqueio do endereço IP tem dois grandes problemas, o primeiro é o tempo decorrido do início do ataque até a identificação do endereço IP suspeito (cerca de 30 minutos). O segundo é o eventual bloqueio de clientes válidos.

Indo de encontro ao objetivo descrito na Seção 1, o MP contribui para a melhoria e eficiência de detecção de consultas maliciosas, diminuindo o número de consultas válidas bloqueadas indevidamente e identificando consultas realmente suspeitas, para tanto, realiza uma comparação entre a detecção baseada no método antigo e seus critérios atuais. Dessa maneira são contabilizados os percentuais de bloqueio de ocorrências de FP e liberações de ocorrências de VN.

O MP identificou no tráfego normal 23,12% de detecções de FP, e 5,40% de detecções de FP na Black Friday. Em contrapartida o MP identificou uma média de 5,56% de detecções de VN-suspeito e 2,51% de detecções de VN-crítico no tráfego normal, enquanto na Black Friday, a média de detecções de VN-suspeito é 7,96% e a média de detecções de VN-crítico é 9,16%. Comparando-se o tráfego normal com o tráfego da Black Friday percebe-se que o percentual de detecções de FP diminuíram e as detecções de VN-suspeito e VN-críticos tiveram um crescimento máximo de cinco pontos percentuais, esse crescimento foi mínimo comparado com o crescimento do tráfego que foi triplicado. Portanto, na Black Friday a maior parte do tráfego é benigno e o aumento de ataques, apesar de existir (cinco pontos percentuais acima de um dia normal) é bastante pequeno.

É importante lembrar que os VNs não são tratados no método antigo e a inclusão dessa camada de proteção por si só já traz um ganho significativo de proteção para o *webservice* CPP

Os tempos de análises variaram nos cenários de 5s a 60s e, de acordo com os resultados, são obtidos resultados similares nos cenários de 10-20-40, 15-30-60, 20-40-60 e 40-50-60. Portanto, o MP pode ser implantado no cenário de 10-20-40, melhorando ainda mais a sua eficiência na identificação de ataques de 40s para 10s. No experimento são criados cenários com três tempos para a validação dos cálculo, porém essa validação pode ser feita apenas em dois tempos, dessa maneira pode-se ganhar desempenho e rapidez na análise do MP.

A distribuição de Poisson foi utilizada para modelar o número de eventos ocorridos de acordo com cada cenário, para efeito de validação, foi feita uma pesquisa por amostragem em 10% dos clientes analisados pelo MP e o modelo de distribuição de Poisson identificou 21,1% de consultas válidas no tráfego normal e 24,4% na Black Friday.

Após a análise dos resultados alcançados, podemos afirmar que os VN-críticos são realmente suspeitos, uma vez que atingem a pontuação máxima proposta pelo MP. Em contrapartida, os VN-suspeitos apresentam uma pontuação extremamente baixa. Para que seja realizada uma alteração de parâmetros, percentuais estatísticos e limite, faz-se necessário reavaliar os VN-suspeitos, de tal forma que sejam detectados padrões de tráfego que possam ser classificados como “bom” ou “ruim”. Até que essa reavaliação seja finalizada, na Camada de resultados do MP os bloqueios serão realizados para os VN-críticos e as liberações serão realizadas para os FPs.

Baseado nos resultados do experimento o Método Preventivo (MP) comprova a sua eficiência na identificação de ataques de indisponibilidade, diminuindo o tempo de resposta, reduzindo ocorrências de FP e detectando ocorrências de VN.

Apesar dos bons resultados apresentados, o MP possui algumas limitações descritas abaixo:

- As pontuações definidas nos cálculos das análises fixa e estatística, apesar de bem explicados, são mutáveis e influenciam diretamente no limite definido;
- O MP não foi testado em tempo real, portanto, não foi possível o dimensionamento real do custo de implantação;
- Não foi comprovado a eficiência do MP em ambiente genérico;

- O *webservice* é uma tecnologia relativamente antiga e a evolução da sua arquitetura é o microserviço. O MP é um método baseado na camada de aplicação e análise de tráfego e não foi testado na arquitetura de microserviço.

Para complementação e desenvolvimento dos aspectos abordados neste trabalho seguem algumas sugestões de trabalhos futuros:

a) Estudos sobre a viabilidade de criação de cenário com tempo máximo de 60 segundos, porém com validação em dois tempos;
b) Estudos sobre a identificação novos padrões de tráfego para definição de novos parâmetros e percentuais estatísticos, afim de definir novo valor de limite;
c) Estudos sobre a aplicabilidade da lógica de identificação de padrões utilizada nesse trabalho em outro *webservice*;
d) Estudos sobre a submissão dos resultados alcançado a algoritmos de aprendizado de máquina supervisionado, afim de agregar novo parâmetro de proteção ao MP.

7 REFERÊNCIAS

BIERMA,M.; DOAK,J.E.; HUDSON,C. **Learning to Rank for Alert Triage**. IEEE Symposium on Technologies for Homeland Security, p. 1-5, 2016.

BUSSAB,W.O.; MORETTIN, P. **A. Estatística básica**. Saraiva, São Paulo, 2004.

CELENK,M.; CONLEY,T.; GRAHAM,J.; WILLIS,J. **Anomaly Prediction in Network Traffic Using Adaptive Wiener Filtering and ARMA Modeling**: IEEE International Conference on Systems, Man and Cybernetics. p. 3548-3553, 2008.

CHIOU,T; TSAI,S; LIN,Y. **Network security management with traffic pattern clustering.** Soft Computing, p. 1757-1770, 2014.

EHLERS, R.S. **Análise de series temporais**. Universidade Federal do Paraná, Paraná, 2007.

GRIFFITHS, N. **nmon performance: A free tool to analyze AIX and Linux performance.** Disponível em: <https://www.ibm.com/developerworks/aix/library/au-nmon_analyser>. Acessado em: 11/03/2018.

GUPTA, S.; KUMAR, P.; ABRAHAM, A. **A Profile Based Network Intrusion Detection and Prevention System for Securing Cloud Environment**. International Journal of Distributed Sensor Networks, v.2, p. 12, 2013.

HOCK,C.; KAPPES,M. **A Self-Learning Network Anomaly Detection System using Majority Voting**. Tenth International Network Conference, p. 59-69, 2014.

JARRAH,O.Y.Al; CHIU,S.A; YOO,P.D.; MUHAIDAT,S.; KIM,K. **Machine-Learning-Base Feture Selection Techniques for Large-Scale Network Intrusion Detection**. IEEE International Conference on Distributed Computing Systens Workshops, p. 177-181, 2014.

KUGISAKI, Y., et al. **Bot detection based on traffic analysis**. In: Intelligent Pervasive Computing. IPC. The 2007 International Conference on. IEEE, p. 303-306, 2007.

WIRESHARK DISPLAY FILTER REFERENCE. **Consulta à documentação oficial de referência do software**. Disponível em: <http://www.wireshark.org/docs/dfref/>. Acesso em: 15/04/2015.

LAZAREVICet al. **Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection.** Proceedings of the Third SIAM International Conference on Data Mining, San Francisco, EUA, p. 25–36, 2003.

LEE, T,B; JAFFE,J. W3C. **Consulta geral a homepage oficial**. Acesso em: < https://www.w3.org/>. Acesso em 10/04/2016.

LEWIS, J.; FOWLER; M. **Microservices: a definition of this new architectural term**. Acessado em :< http://martinfowler.com/articles/microservices.html>. Acesso em 10/01/2018.

LIN, Y. D.; LAI, Y. C.; HO, C. Y.; TAI, W. H. **Creditability-based weighted voting for reducing false positives and negatives in intrusion detection**. Computers and Security, v. 39, n. PART B, p. 460–474, 2013.

MELL, P.; KENT, K.; NUSBAUM, J. **Guide to malware incident prevention and handling**. Computer Security Division, Information Technology Laboratory, NIST, MD, USA, p. 110, 2005.

MITCHELL, R; CHEN,I.**On survivability of mobile cyber physical systems with Intrusion detection.** Wireless personal communications. p. 1377-1391, 2013.

MITCHELL, T.M; et al. **Machine learning**. McGraw Hill series in computer science. I-XVII,1-414, 1997.

MUDZINGWA,D; AGRAWAL,R. **A study of Methodologies used in Intrusion Detection and Prevention Systems (IDPS)**: Proceedings of IEEE Southeastcon, 2012.

NIST/SEMATECH e-Handbook of Statistical Methods. **Poisson Distribution**. Disponível em: < http://www.itl.nist.gov/div898/handbook/eda/section3/eda366j.htm>. Acesso em 10/01/2017.

PAULA, G.A. **Modelos de regressão: com apoio computacional**. São Paulo: IME-USP, 2004.

PYTHON SOFTWARE FOUNDATION. **Consulta à documentação oficial de referência do software**. Disponível em: < https://www.python.org/>. Acesso em: 12/07/2016.

QOSIENT . **Monitoring networks in a whole different way**. Disponível em: < http://www.qosient.com/argus/> Acesso em 10/07/2016.

QUEIROZ,D.V.; VIEIRA,J.C. M.; FONSECA,I.E. **Detecção de Ataques de Negação de Serviço Utilizando Ferramentas de Monitoramento e Análise de Tráfego**.Revista de Tecnologia da Informação e Comunicação, Vol.4, Número 1,2014.

ROBSON, R; THOMAS,C. **Ranking of Machine learning Algorithms Based on the Performance in Classifying DDoS Attacks**. IEEE Recent Advances in Intelligent Computacional Systems (RAICS),p. 185-190, 2015.

SAMUEL, A.L. **Some Studies in Machine Learning Using the Game of Checkers**. IBM Journal of Research and Development, Vol.3 , p. 210-229, 1959.

SCARFONE, K.; MELL, **P. Guide to Intrusion Detection and Prevention Systems (IDPS)**. Computer Security Division, Information Technology Laboratory, NIST, MD, USA, p. 127, 2007.

SCHMIDT, C. M. C. **Modelo de regressão de Poisson aplicado à área da saúde**. Ijúi, 2003. 98 f. Dissertação (Mestrado em Modelagem Matemática) - Universidade Regional do Noroeste do Estado do Rio Grande do Sul, Ijúi, 2003.

SHAI, S. S.; SHAI, B. D. **Understanding machine learning: from theory to algorithms**. [S.I.], 2014.

TCPDUMP SOFTWARE MANPAGE. **Consulta à documentação oficial de referência do software** .Disponível em: <http://www.tcpdump.org/manpages/tcpdump.1.html>. Acesso em: 12 abr 2015.

TEGELER, et al. Botfinder: **Finding bots in network traffic without deep packet inspection**. Proceedings of the 8th international conference on Emerging networking experiments and technologies. ACM. p. 349-360, 2012.

ZEILEIS, A.; KLEIBER, C.; JACKMAN, S. **Regression models for count data.** Journal of statistical software. 1-25, 2008.

ZHAO,W.; YIN,J.; LONG,J. **A Prediction Model of DoS Attack's Distribution Discrete Probability**. The Ninth International Conference on Web-Age Information Management. p. 625-628, 2008.