Método preventivo baseado em esquema de *ranking* para detecção de ataques contra a disponibilidade em *webservice*

Ligia Maria da Silva Danta[1]; Adilson Eduardo Guelfi[2]; Anderson Aparecido Alves da Silva[3]; Marcelo Teixeira de Azevedo[4]; Sergio Takeo Kofuji[4]

[1]IPT, Departamento de Engenharia da Computação, São Paulo, SP, li.gonzales@gmail.com
[2]IPT/UNOESTE, Departamento de Engenharia da Computação, São Paulo, SP, guelfi@unoeste.br
[3]IPT/UNIP/SENAC/USP, Departamento de Engenharia da Computação, São Paulo, SP, anderson.silva@pad.lsi.usp.br
[4]USP, Departamento de Engenharia Elétrica, São Paulo, SP, marcelo.azevedo@pad.lsi.usp.br; kofuji@usp.br

Resumo

O Calculador de Preços e Prazos (CPP) é um *webservice* disponível gratuitamente na *Internet* para consulta de fretes que recebe entre 8 e 28 mil consultas diárias. Constantemente, o CPP sofre ataques na forma de consultas que causam indisponibilidade e lentidão no sistema. O atual critério usado para a resolução desse problema não tem por base o volume de tráfego, por ser oneroso e bloquear, eventualmente, clientes válidos. O objetivo deste trabalho é propor um método de detecção preventivo de ataques contra a disponibilidade, baseado em um esquema de *rankings*/pesos que melhore a precisão e o tempo de resposta da detecção de conexões suspeitas em um *webservice*. 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 usados para compor um *ranking* de suspeição para cada consulta.

Palavras-chave: *Webservice*; Distribuição de Poisson; Reconhecimento de Padrões; Esquema de *ranking* e votação.

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

Abstract: The Price and Deadline Calculator (CPP) is a gratuitous webservice available on the Internet for freight consultations that receive between 8 and 28 thousand of daily queries. CPP constantly receives attacks in the form of queries that cause unavailability and slowness in the system. The current criteria used to solve this problem are not based on traffic volume, because it is costly and may eventually block valid customers. This work aims to propose a method of preventive detection of attacks against availability, based on a rankings/weights scheme that improves the accuracy and response time of detecting suspicious connections in a webservice. In the experiment, a voting scheme is created, generated from network traffic, where standard-based statistics and predefined rules are used to compose a ranking of suspicion for each query.

Keywords: Webservice; Poisson Distribution; Pattern Recognition; Ranking and voting scheme.

1. Introdução

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. Porém, a tendência na evolução da arquitetura do setor de *software* orientado a serviços é o microsserviço. O microsserviço é uma abordagem de desenvolvimento de aplicações escalável, com vários pequenos serviços independentes, implementação automatizada e um gerenciamento mínimo centralizado [Lewis & Fowler 2014] [Assunção; Krüger & Mendonça 2020]. Entre as vantagens do uso de microsserviços estão a possibilidade de projetar, desenvolver, testar e executar serviços com agilidade, diminuindo bastante a interação humana [Di Francesco; Lago & Malavolta 2019].

O Calculador de Preços e Prazos (CPP) é um *webservice* disponível gratuitamente na *Internet* (http://www2.correios.com.br/sistemas/precosPrazos/) pelos Correios. Composto por 3 servidores virtuais, cada um com 4 processadores, 16 GB RAM e 4 instâncias, os principais clientes do CPP são lojas virtuais de grandes empresas varejistas. 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. São 220 mil consultas/hora em média, sendo que este número pode dobrar quando acontecem ataques.

Segundo [Queiroz *et al*. 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 se tratar de um *webservice*, o principal ataque analisado neste artigo é aquele direcionado à camada de aplicação contra a disponibilidade. Basicamente, o atacante conduz uma série gigantesca de consultas sequenciais, a fim de diminuir a capacidade de resposta do *webservice.* Além da dificuldade do discernimento entre consultas legítimas e maliciosas, este tipo de ação causa lentidão e indisponibilidade, principalmente em ocasiões de grandes eventos e promoções na *Internet*, tais como: Dia das Mães, *Black Friday* e Natal.

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.

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, utilizar um esquema de *ranking* com um conjunto de regras para classificação de acessos ao *webservice* CPP, possibilita que os eventos gerados possam ser rotulados como suspeitos ou normais.

Alguns trabalhos como os de [Priya et al. 2020][Dehkordi et al. 2020][Hock & Kappes 2014] aplicam algumas soluções preventivas, baseadas em estatística e ranqueamento, para esse tipo de ataque. Com base nisso, o objetivo geral desta pesquisa é propor um método preventivo para 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*. Os objetivos específicos incluem a redução do tempo de resposta à ataques contra a indisponibilidade do *webservice* (em geral, um ataque leva de 10 a 30 minutos para ser identificado) e a redução dos índices de detecção Falso Positivo (FP) e Verdadeiro Negativo (VN). A metodologia usada envolve a captura de tráfego e a análise da distribuição de Poisson, somada à análise do contexto das consultas suspeitas realizadas no *webservice* para a formação de um *ranking* que possa antecipar/diminuir o tempo de resposta aos ataques. A validação da proposta é feita por meio de um experimento com dados reais do CPP.

2. Aprendizado de máquina e *ranking*

De acordo com [Shai 2014], aprendizado de máquina refere-se à detecção automatizada de padrões de dados. [Kim 2020] salienta que o aprendizado de máquina é uma ferramenta comum para tarefas de extração simples e/ou complexas de informações de grandes conjuntos de dados. No trabalho de [Akune et al. 2021] é utilizado o aprendizado de máquina como forma de prevenção de vazamento de dados.

O esquema de *ranking* ou votação é uma classificação que segue determinados critérios. Esse esquema usado para classificação de eventos suspeitos é abordado por diversos autores. O trabalho de [Hock & Kappes 2014] utiliza um sistema de aprendizado de máquina baseado em um esquema de votação para futuramente desenvolver um sistema de detecção de anomalias. O trabalho de [Bierma *et al.* 2016] utiliza aprendizado de máquina para criar um esquema de *ranking* e classificar alertas de segurança. A criação de um esquema de *ranking*, como o utilizado no presente artigo, é uma forma simplificada de aprendizado de máquina. Vale ressaltar que o ranking é criado, justamente, a partir da verificação do enquadramento dos dados na distribuição de Poisson.

2.1. Modelo de regressão de Poisson

Os modelos de regressão de Poisson são muito usados para analisar dados de contagem. Para uma variável Y com valores inteiros $y=\{0,1,2,...\}$ e número médio de ocorrências $\mu>0$ a probabilidade da distribuição de Poisson é [Dobson 2008]:

$$P\{Y = y\} = \frac{e^{-\mu}\mu^{y}}{y!} \quad (1)$$

O efeito das variáveis independentes na variável de resposta Y é modelado através de regressão por meio do parâmetro μ. Seja $Y=(Y_1,...,Y_n)$ um conjunto de variáveis aleatórias independentes, onde Y_i é o i^{th} evento de n_i e $\theta=(\theta_1, ...,\theta_n)$ é o vetor de parâmetros da distribuição. Neste caso, o valor esperado de Y_i é $E(Y_i)=\mu_i=n_i\theta_i$ e a dependência do modelo em θ_i na variável independente é [Dobson 2008]:

$$\theta_i = e^{x_i^T \beta} \quad (2)$$

E o Modelo Linear Generalizado corresponde a:

$$E(Y_i) = \mu_i = n_i e^{x_i^T \beta} \quad (3)$$

Um detalhe vital para a proposta deste artigo é que na distribuição de Poisson a variância é igual à média: $E(Y)=var(Y)=\mu$. Essa propriedade é usada para diferenciar o tráfego das

consultas ao CPP entre legítimo e malicioso, dentro de um determinado intervalo de tempo.

3. Trabalhos relacionados

Nesta seção são abordadas as principais discussões que motivam o desenvolvimento da pesquisa: aprendizado de máquina, *ranking*, técnicas de detecção de intrusão e identificação de padrões.

Em relação ao aprendizado de máquina e esquemas de *ranking*, [Lin *et al.* 2013] propõem um esquema de votação ponderada baseada em credibilidade, para diminuição das taxas de alertas FP e VN entre múltiplos Sistemas de Detecção/Prevenção de Intrusões (SDPI). [Hock & Kappes 2014] implementam um modelo de voto por maioria, em conjunto com um esquema estatístico de aprendizado de máquina em um SDPI, que aprende o comportamento normal da rede por meio de um modelo estatístico, antes de realizar a detecção. Já o trabalho de [Robson & Thomas 2015] ajudou a definir as métricas da nossa proposta. Os autores avaliam o desempenho de dez algoritmos de aprendizado de máquina supervisionados, usando como métricas a taxa de alertas FP e VN, a precisão do algoritmo, a recuperação e a precisão da detecção.

Outras hipóteses sobre a detecção de anomalias são encontradas nos trabalhos de [Celenk *et al.* 2008] que prevê anomalias no tráfego de rede, usando a filtragem adaptativa de Wiener e o modelo *Auto Regressive Moving Average* (ARMA). Dentro deste mesmo contexto, [Zhao *et al.* 2008] cria um modelo de predição de ataques do tipo DoS, baseado no método, e agrupamento de algoritmos genéticos e métodos bayesianos. Já [Chiou 2014] aplica o reconhecimento de padrões em logs de DNS para identificar servidores *botnet,* que alteram dinamicamente nomes e domínios maliciosos para não serem detectados. Da mesma forma, [Priya et al. 2020] utilizam três algoritmos de aprendizado de máquina (*Naive Bayesian, k-Nearest Neighbors e Randon Forest)* para classificar ataques DDoS. Já [Dehkordi et al. 2020] expandem um pouco a detecção de ataques contra a disponibilidade. Os autores propõem um método que inclui modelos estatísticos e de aprendizado de máquina em uma *Software Defined Networks* (SDN).

De acordo com Kim et al. (2020), mesmo eventos complexos, como a observação de ondas gravitacionais, podem ser realizados com base em esquemas de *ranking*. Para observar outros eventos eletromagnéticos, os autores estudam a viabilidade de adoção de *machine learning* supervisionada, Método de aprendizagem (ML) para pontuação de classificação em eventos candidatos GW.

A Tabela 1 consolida as comparações das principais referências deste artigo.

Tabela 1. Comparação dos trabalhos relacionados

Artigos Critérios	**A-2013**	**B-2014**	**C-2015**	**D-2008**	**E-2008**	**F-2014**	**G-2020**	**H-2020**	**I-2020**	**J-2021**
C1	sim	sim	sim				sim	sim	sim	**sim**
C2				sim	sim	sim		sim	sim	**sim**
C3	sim	sim	sim				sim		sim	**sim**
C4	sim	sim					sim		sim	**sim**
C5	sim	sim	sim	sim		sim		sim	sim	**sim**
C6		sim		sim						**sim**
C7					sim	sim	sim	sim	sim	**sim**
C8										**sim**
C9	sim					sim	sim			**sim**
C10					sim		sim		sim	**Sim**

Tópicos:	Artigos:
C1=Aprendizado de Máquina	A-2013 = Lin et al (2013)
C2=Previsão de ataques	B-2014 = Hock e Kappes (2014)
C3=Esquema de *ranking*	C-2015 = Robson e Thomas (2015)
C4=Identificação de FP	D-2008 = Celenk et al. (2008)
C5=Análise de camada TCP	E-2008 = Zhao, Yin e Long (2008)
C6=Detecção de anomalias	F-2014 = Chiou (2014)
C7=Dados Reais	G-2020 = Kim et al. (2020)
C8=Comportamento humano	H-2020 = Priya et al. (2020)
C9=Identificação de Padrões	I-2020 = Dehkordi et al. (2020)
C10=Possui método de predição	J-2021 = Nossa Proposta

Como usamos um esquema simplificado de aprendizado de máquina, é importante verificar trabalhos com essa abordagem. Os trabalhos que tratam a previsão de ataques, identificação de FP, análise de camada TCP, detecção de anomalias e identificação de padrões são essenciais para entender como identificar padrões de ataque. Os trabalhos que utilizam esquemas de *ranking*, auxiliam na categorização dos eventos e os trabalhos que exemplificam situações com dados reais e comportamento humano, colaboram para demonstrar que o método pode ser utilizado em situações reais e que os eventos são produzidos por seres humanos e não por aplicações automatizadas. Por fim, os trabalhos que abordam o método de predição, podem ser considerados como base para evolução da proposta.

4. Método Preventivo

O Método Preventivo (MP) é baseado nos conceitos de *ranking* em tempo real, sendo capaz de melhorar a precisão, o tempo da identificação e o bloqueio de consultas maliciosas, tomando como base a análise de conexões e um conjunto de regras pré-definidas - explicadas na descrição dos módulos e submódulos ilustrados na Figura 1. O CPP possui uma proteção de perímetro, porém, essa proteção não faz distinção entre consultas de clientes suspeitos ou não.

A Figura 1 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 submódulos 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).

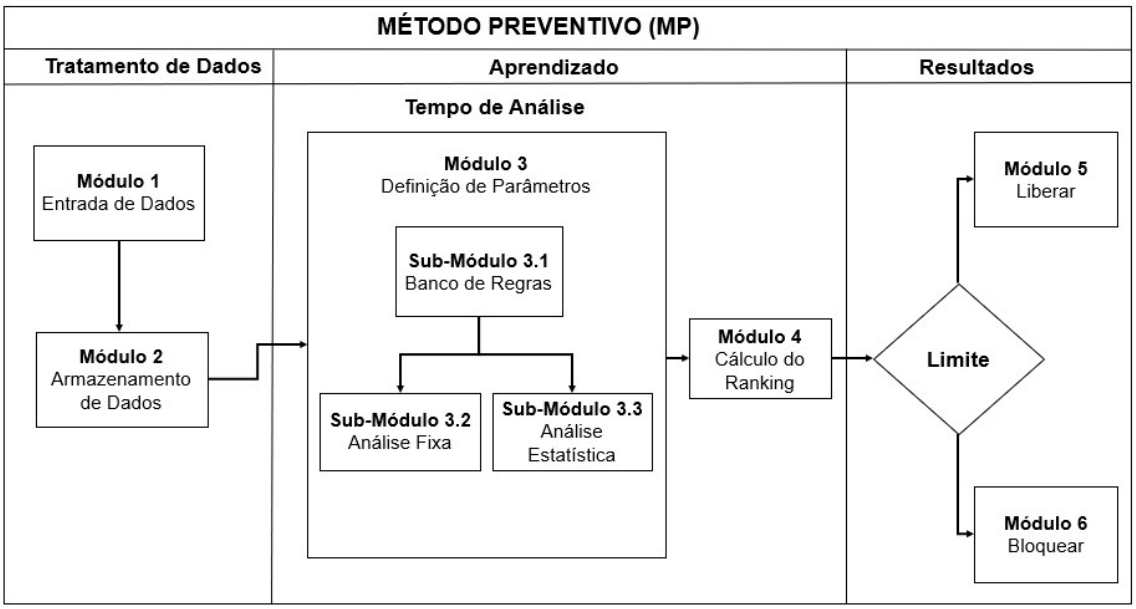

Figura 1. Estrutura do MP

4.1. 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 os dados são capturados por um *sniffer.* Os dados dizem respeito à tráfego de rede no formato *pcap,* [Sikos 2020] com diversos campos: data, horário, IP origem, CEP origem, CEP, destino, serviço, peso da mercadoria. Todo o tráfego é agrupado e ordenado por IP, mas a investigação no tráfego só ocorre quando um determinado IP realiza 30 ou mais consultas dentro de um período de 5 segundos – volume considerado suspeito de ser um ataque no referido tempo e mínimo para o cálculo de variância [Bussab & Morettin 2004]. Este procedimento gera agilidade, pois apenas o tráfego suspeito é analisado.

No Módulo 2 os dados são armazenados em cinco cenários (1 a 5) com três blocos (inicial, intermediário e final). Os tempos de armazenamento variam para cada bloco e cenário de acordo com a Tabela 2

Tabela 2. Cenários e blocos de tempo para captura de dados.

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

Na primeira coluna da Tabela 2, estão os cinco cenários e as colunas subsequentes mostram o tempo de captura dos dados em segundos. Por exemplo, no cenário 1: o bloco inicial captura os dados no intervalo de 0 a 5s; o bloco intermediário captura o tráfego no período de 0 a 10s; e o bloco final faz a captura no intervalo entre 0 e 20s.

4.2. Camada de aprendizado

Na camada de aprendizado são executadas as funcionalidades relativas 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*).

4.2.1. Submódulo Definição de Parâmetros

O Submódulo Banco de Regras é um conjunto cinco regras ponderadas que identifica um ataque a partir de uma consulta. São consideradas suspeitas consultas sequenciais: com o mesmo IP (Regra 1, peso 0,3); com o mesmo CEP de origem (Regra 2, peso 0,2); com o mesmo CEP de destino (Regra 3, peso 0,2); com o mesmo serviço (Regra 4, peso 0,2); e com o mesmo peso da mercadoria (Regra 5, 0,1). A ponderação crescente indica se o tráfego tem ou não potencial de ser um ataque.

O Submódulo Análise Fixa corresponde ao reconhecimento de padrões, a partir da ocorrência mínima de 30 consultas em 5s (bloco inicial do cenário 1). Uma determinada consulta tem seus pesos (ponderações) somados (valor S) quando batem com o Banco de Regras. Uma consulta que alcance o valor S de 0,5 é considerada suspeita pela análise fixa. A base para o valor mínimo de S é a ocorrência da Regra 1 (maior peso) somada à execução das Regras 2, 3 ou 4.

O Submódulo Análise Estatística consiste no cálculo da média (μ) e da variância (σ^2) no tempo de 1 segundo para cada consulta de um determinado IP. Em seguida, verifica-se se os dados estão dentro da distribuição de Poisson: a razão entre ($ratio$=μ/σ^2) deve ser próxima de 1.

Distribuições de contagem (Poisson e Binomial Negativa) servem para descobrir o número de eventos independentes que ocorrem em um período. 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/σ^2 é próxima de 1). O inverso ocorre quando a consulta é maliciosa: a quantidade de ocorrências válidas tende a ser maior. Neste caso, ($ratio$=μ/σ^2 é distante de 1). A variável peso-e=1 ($ratio$<0,5 ou $ratio$>1,5) indica quando uma sequência de consultas é suspeita. Quando a consulta não é suspeita, (0,5≥$ratio$≤1,5)

peso-e=0. Um exemplo pode ser visto na Tabela 3, onde peso-e=0 (legítima), já que o *ratio* das consultas sequenciais de um mesmo IP chega a 0,76.

Para ser considerada suspeita pela análise estatística, uma média dos três peso-e (Bloco 1, Bloco 2 e Bloco 3) é calculada, como pode ser visto no exemplo da Tabela 4 para seis diferentes usuários.

Tabela 3 - 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
Peso-e	0

Tabela 4 – 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 peso-e
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

4.2.2. Submódulo Cálculo do *Ranking*

O *ranking* é usado para definir se uma sequência de consultas de um mesmo IP é ou não maliciosa. Trata-se da soma das análises fixa (S) e estatística (média peso-e). O limite

mínimo do *ranking* para uma consulta ser considerada suspeita é 0,83 (valor mínimo de uma consulta suspeita pela análise fixa S=0,5 + média peso-e com pelo menos uma sequência de consultas considerada suspeita pela análise estatística).

4.3. Camada de resultados

Nesta camada são realizados os bloqueios (Submódulo Bloquear) ou liberações (Submódulo Liberar) de consultas a partir do limite do *ranking* definido.

5. Resultados

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.

5.1. Ambiente do CPP

A validação da proposta é realizada com dados coletados em um ambiente real. Na Figura 2 podem ser observados todos os equipamentos que compõe a solução do CPP.

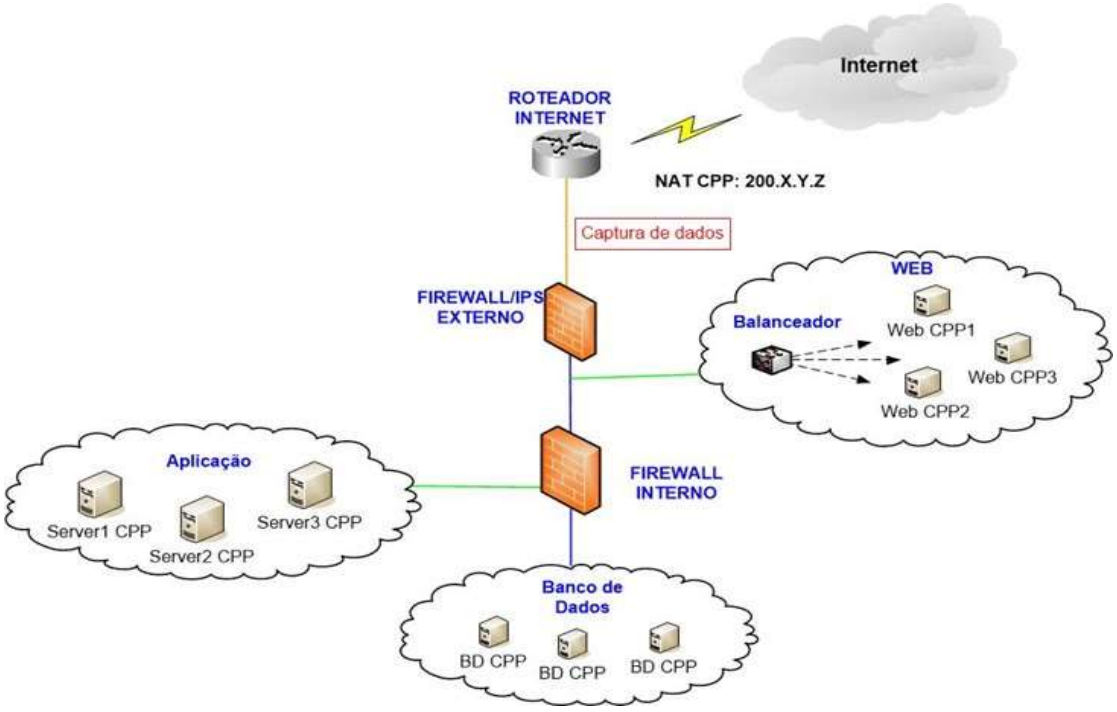

Figura 2. Topologia de rede do CPP

O ambiente do CPP é composto por 3 servidores de *webservice*, 3 servidores de aplicação, 3 servidores de banco de dados, 1 balanceador, 2 *firewalls* 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 os 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 realizada na interface externa do *firewall* externo por meio da ferramenta *tcpdump*, que coleta o *timestamp* e o conteúdo dos pacotes que passam por uma interface.

5.2. Experimentos realizados

A proposta deste 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 2. O experimento se divide em três fases: captura, análise dos dados e resultados.

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 5, pode ser observado o detalhamento das capturas realizadas, com nomes dos arquivos, tamanho e horários iniciais e finais de captura.

Tabela 5. Captura de Dados

Arquivo	Tamanho	Início	Fim
Captura-1.pcap	113 GB	19/05/2016 15:55:55	24/05/2016 09:38:31
Captura-2.pcap	109 GB	25/05/2016 15:24:15	27/05/2017 16:07:08
Captura-3.pcap	99 GB	31/05/2016 16:05:30	02/06/2017 16:11:54
Captura-4.pcap	102 GB	07/06/2016 17:05:30	10/06/2016 11:06:50
Captura-5.pcap	14 GB	24/11/2017 08:20:01	24/11/2017 16:12:00

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, conforme descrito na Tabela 6.

Tabela 6. Cenário de Intervalo de Tempos

Cenários	Bloco Inicial	Bloco Intermediário	Bloco Final
1	5	10	20
2	10	20	40
3	15	30	60
4	20	40	60
5	40	50	60

Na Tabela 6, 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 - 5s, gerando um bloco de dados inicial. Em seguida, os dados são analisados dentro do intervalo de 0 – 10s, formando um segundo bloco de dados. Por último, os dados são analisados no intervalo de 0 – 20s, 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.

5.3. Coleta e análise de 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 7. Períodos de Análise

	Quarta	Quinta	Sexta
	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	

Como pode ser observado na Tabela 7, 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.

Na elipse da Figura 3, 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, o exemplo mostra a análise de um cenário específico com blocos de 10s, 20s e 40s, durante 10 minutos.

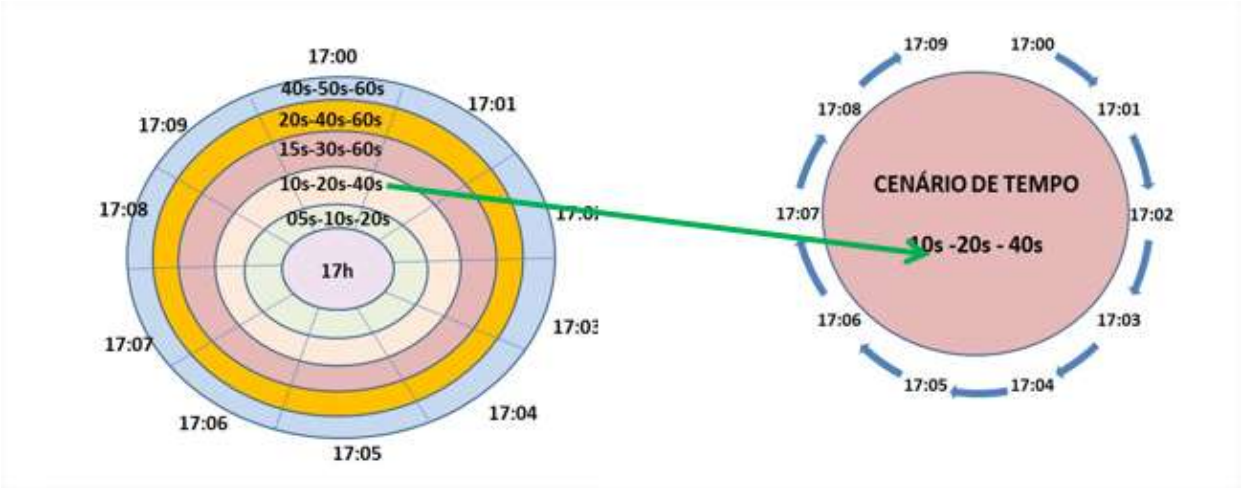

Figura 3. Exemplo do ciclo de análise

Estatisticamente, a amostragem de dados é suficiente e representativa, tendo em vista a baixa variabilidade dos dados [Bussab & Morettin 2004]. Além disso, o volume de 660 mil consultas por hora e a demora de cerca de 10 minutos para identificação de uma consulta maliciosa também corrobora a amostragem.

Os períodos escolhidos para coleta abrangem os períodos de maior uso do CPP. Após as análises, os dados são enviados para os cálculos. A Tabela 5 mostra exemplos de cinco cálculos realizados em um minuto de análise.

No exemplo da Tabela 8, o experimento é referente a data de 01/06/2016 às 10h. A análise 1 refere-se aos eventos ocorridos no horário entre 10:00 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 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 5, quando o *ranking* $\geq$ 0,83 (em vermelho) o cliente é considerado suspeito e quando *ranking* $<$ 0,83 (em verde) o cliente é considerado normal.

Tabela 8. 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 (ratio < 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 (ratio < 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 (ratio < 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 (ratio < 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

5.4. Resultados

No método antigo, o critério usado para detecção de consultas suspeitas ao CPP não é eficaz, o analista responsável pelo sistema identifica um alto consumo de recursos (processamento e memória) nos servidores do CPP, solicita a 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 10 a 30 minutos entre o início do ataque e o bloqueio dos IPs suspeitos e o segundo 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. Para efeito de comparação, são contabilizados os eventos de uma sexta-feira comum (20/05/2016 com 88.379 consultas) e de uma *Black Friday* (24/11/2017 com 280.562 consultas), que possui quase o triplo de consultas.

A Figura 4 mostra o percentual de eventos considerados suspeitos em relação ao total de eventos analisados agrupados por períodos e horários.

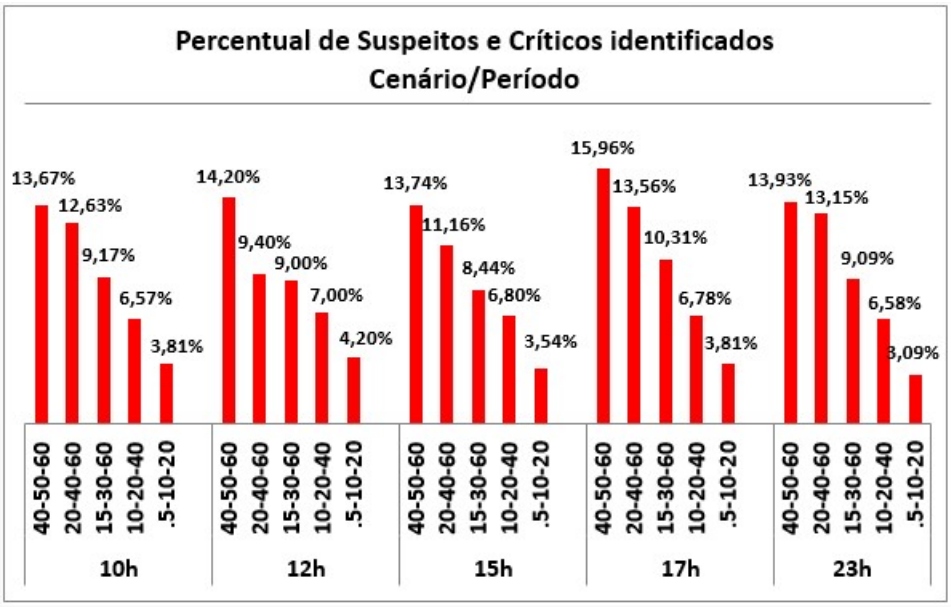

Figura 4 - Percentual de suspeitos e críticos por cenário

Pode-se perceber na Figura 4 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 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 número de consultas) são considerados suspeitos por esse método.

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

A Figura 5 mostra os resultados acumulados de todos os períodos analisados, exceto a *Black Friday*. Percebe-se que o percentual de FP é bastante expressivo ultrapassando 20%

e os percentuais de VN-suspeito e VN-crítico, mantém percentuais equilibrados, variando em torno de 1% a maior ou a menor.

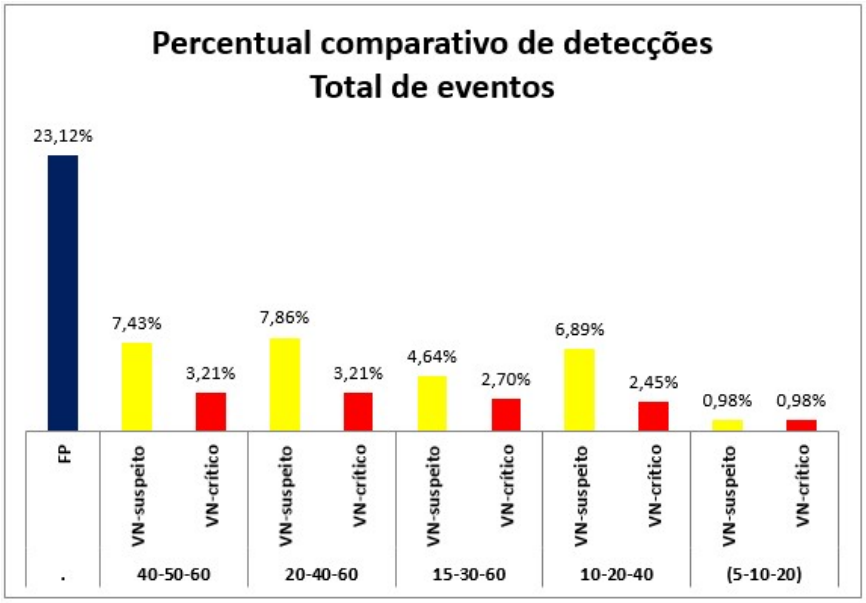

Figura 5 - Percentual comparativo de detecções em tráfego normal

A Figura 6 mostra um baixo percentual de FP proveniente da *Black Friday*, pouco mais de 5%, essa redução de FP é acompanhada de dois fatores relevantes, o aumento significativo da quantidade de consultas, nesse caso, a triplicação delas quando 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.

É possível que 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.

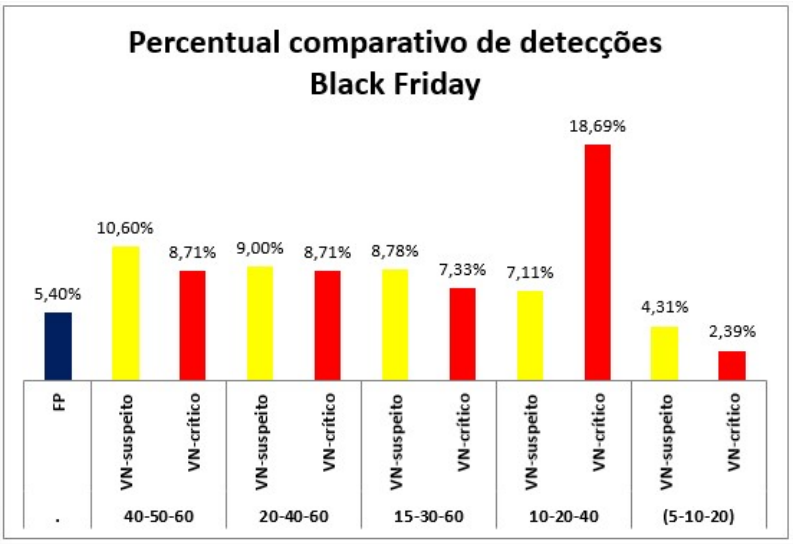

Figura 6 - Percentual comparativo de detecções na *Black Friday*

Esta pesquisa 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*. Como comparação, o tempo de detecção de consultas maliciosas, sem o MP, era de 10 minutos, no melhor caso. Com a implementação do MP esses tempos se reduziram, no pior dos casos, para 1 minuto – este é o valor máximo dos cenários utilizados (10-20-40, 15-30-60, 20-40-60 e 40-50-60).

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.

O Nmon e o *Nmon Analyser* são softwares livres, disponibilizados pela IBM para monitoramento de desempenho [Griffiths 2003]. 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 do desempenho. O desempenho pode ser observado na Figura 7.

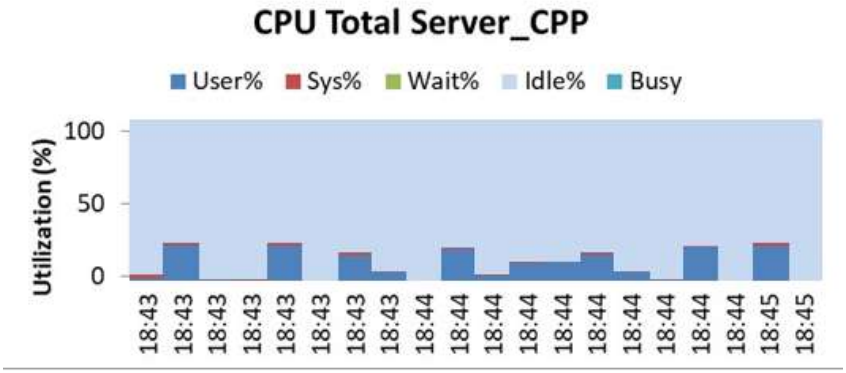

Figura 7 – Utilização total de CPU – Server_CPP

O Nmon e o *script* são executados simultaneamente entre o horário das 18:43 e 18:45. Os picos de gravação em disco mostrados pela ferramenta chegam no máximo a 700 KB/s. O gráfico da Figura 7 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.

6. Conclusões

Esta pesquisa se dedicou a buscar formas de diminuir o problema de lentidão e indisponibilidade ocasionado por ataques contra a disponibilidade no *webservice* CPP. Para tanto, propõe um método preventivo de ataques de indisponibilidade, baseado em esquema de *ranking*, 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 10 a 30 minutos). O segundo é o eventual bloqueio de clientes válidos.

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, além de diminuir o tempo de detecção de consultas maliciosas de 10 para 1 minuto. 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, se 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 cálculos, 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:
a) As pontuações definidas nos cálculos das análises fixa e estatística são mutáveis e influenciam diretamente o limite definido;
b) A eficiência do MP não foi generalizada para outros ambientes. O *webservice* é uma tecnologia relativamente antiga e a evolução da sua arquitetura é o microsserviço;
c) A implementação do MP não foi realizada de forma definitiva no ambiente produtivo. Contudo, essa implementação é possível porque o método é baseado em *scripts* que facilitam sua automação;
d) Como identifica consultas maliciosas de forma individual, o MP é vulnerável à ataques massivos DDoS.

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) Identificação de novos padrões de tráfego e parâmetros para a definição de novo valor de limite;
c) Aplicação do MP em outro *webservice*;
d) Adaptação do *script* usado para outros algoritmos de *machine learning* mais sofisticados.

Agradecimento
Este trabalho teve suporte técnico da parceria Huawei-USP.

REFERÊNCIAS

1. AKUNE, L.; SILVA, A.; GUELFI, A.; AZEVEDO, M.; ALCÁZAR, J.; KOFUJI, S. Um sistema de prevenção de vazamento de dados de imagens baseado em aprendizado de máquina. *Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação*, [S.l.], v. 1, n. 13, jan. 2021.
2. BIERMAM, M.; DOAK, E.; HUDSON, C. Learning to Rank for Alert Triage. IEEE Symposium on Technologies for Homeland Security, p. 1-5, 2016.
3. 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.
4. CHIOU, T.; TSAI, S.; LIN, Y. Network security management with traffic pattern clustering. Soft Computing, p. 1757-1770, 2014.
5. DEHKORDI, A.B.; SOLTANAGHAEI, M.R.; BOROUJENI, F.Z. The DDoS attacks detection through machine learning and statistical methods in SDN. Springer Science + Business Media, LLC, part of Springer Nature 2020.
6. DI FRANCESCO, P.; LAGO, P.; MALAVOLTA, I. Architecting with microservices: A systematic mapping study. Journal of Systems and Software, v. 150, p. 77-97, 2019.
7. Dobson, A.J., Barnett, A.: An introduction to generalized linear models. CRC press (2008)
8. 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.
9. HOCK, C.; KAPPES, M. A Self-Learning Network Anomaly Detection System using Majority Voting. Tenth International Network Conference, p. 59-69, 2014.
10. KIM, K.; Li, T.G.F.; LO, R.K.L.;SACHDEV,S.;YUEN, R.S.H. Ranking candidate signals with machine learning in low-latency searches for gravitational waves from compact binary mergers. Physical Review D 101, 83006 (2020). p 083006-1-083006-11, 2020.
11. 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.
12. 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.
13. 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.
14. 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.
15. PAULA, G.A. Modelos de regressão: com apoio computacional. São Paulo: IME-USP, 2004.
16. PRIYA, S. S.; YUVARAJ, D.; SIVARAM, M.; JAYANTHILADEVI, A. Machine Learning based DDOS Detection. 2020 International Conference on Emerging Smart Computing and Informatics (ESCI). p. 234-237, 2020.

17. QUEIROZ, V.; VIEIRA, J.; FONSECA, I. 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.
18. ROBSON, R.; THOMAS, C. Ranking of Machine learning Algorithms Based on the Performance in Classifying DDoS Attacks. IEEE Recent Advances in Intelligent Computational Systems (RAICS), pp.185-190, 2015.
19. SCARFONE, K.; MELL, P. Guide to Intrusion Detection and Prevention Systems (IDPS). Computer Security Division, Information Technology Laboratory, NIST, MD, USA, p. 127, 2007.
20. SHAI, S.; SHAI, D. Understanding machine learning: from theory to algorithms. [S.I.], 2014.
21. SIKOS, L. F. Packet analysis for network forensics: A comprehensive survey. Forensic Science International: Digital Investigation, v. 32, p. 200892, 2020.
22. ASSUNÇÃO, W. K. G; KRÜGER, J.; MENDONÇA, W. D. F. Variability management meets microservices: six challenges of re-engineering microservice-based webshops. In: Proceedings of the 24th ACM Conference on Systems and Software Product Line: Volume A-Volume A. 2020. p. 1-6.
23. ZHAO, W.; YIN, J.; LONG, A. Prediction Model of DoS Attack's Distribution Discrete Probability. The Ninth International Conference on Web-Age Information Management. p. 625-628, 2008.

Marcelo Teixeira de Azevedo possui pós-doutorado pela POLI-USP e é professor universitário nos cursos de engenharia e computação.

Ligia Danta é mestre em Engenharia da Computação pelo IPT. Trabalha com segurança da informação desde 2006.

Sergio Kofuji atualmente é Professor Doutor da Escola Politécnica da USP. Atualmente ele tem se concentrado em IoT, 5G e Cidades Inteligentes.

Anderson Silva é professor universitário da área de segurança da informação com pós-doutorado pela Poli-USP.

Adilson Guelfi é Doutor em Engenharia pela Poli-USP. Atualmente, é Reitor de Pesquisa e Pós-Graduação da UNOESTE.

(esta página par está propositadamente em branco)