Maira Lambort Batista

Análise de Eventos de Segurança em Redes de Computadores Utilizando Detecção de Novidade

São José do Rio Preto
2012

Maira Lambort Batista

Análise de Eventos de Segurança em Redes de Computadores Utilizando Detecção de Novidade

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

Orientador: Prof. Dr. Adriano Mauro Cansian

São José do Rio Preto
2012

Batista, Maira Lambort.
Análise de eventos de segurança em redes de computadores utilizando detecção de novidade / Maira Lambort Batista. - São José do Rio Preto : [s.n.], 2012.
58 f. : il. ; 30 cm.

Orientador: Adriano Mauro Cansian
Dissertação (mestrado) - Universidade Estadual Paulista, Instituto de Biociências, Letras e Ciências Exatas

1. Computação. 2. Redes de computadores – Medidas de segurança. 3. Análise de fluxos. 4. Detecção de intrusão. I. Cansian, Adriano Mauro. II. Universidade Estadual Paulista, Instituto de Biociências, Letras e Ciências Exatas. III. Título.

CDU – 004.7

Ficha catalográfica elaborada pela Biblioteca do IBILCE
Campus de São José do Rio Preto - UNESP

Maira Lambort Batista

Análise de Eventos de Segurança em Redes de Computadores Utilizando Detecção de Novidade

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

Banca Examinadora

Prof. Dr. Adriano Mauro Cansian
UNESP – São José do Rio Preto
Orientador

Prof. Dr. Alex Sandro Roschildt Pinto
UNESP – São José do Rio Preto

Prof. Dr. Edson dos Santos Moreira
USP – São Carlos

São José do Rio Preto
12/Março/2012

Dedico este trabalho

Aos meus pais, Marta e Luís e aos meus irmãos Laiara e Gabriel, pelo incentivo, compreensão e amor durante todos os tempos.

AGRADECIMENTOS

Ao Deus maravilhoso pela vida, saúde e por tudo que tem feito por mim.

Aos meus pais, Marta e Luís, minha irmã Laiara e meu irmão Gabriel, por sempre me apoiarem e me incentivarem para chegar até onde cheguei.

Ao meu orientador, Prof. Dr. Adriano Mauro Cansian, pela orientação pessoal e acadêmica, pelo apoio durante os anos que permaneci no Laboratório ACME! e pelas horas de descontração.

Aos meus colegas de laboratório: Leandro, André Proto e Isabela em especial pela ajuda no projeto e pela enorme amizade que têm comigo.

Aos companheiros de laboratório, Jorge, Adriano, Heitor, Bruno, Vinícius Oliveira, Vinícius Galhardi e Raphael, obrigado pelo companheirismo, pelas horas de conversa e estudo.

Aos professores do Departamento de Ciências de Computação e Estatística, pelos conhecimentos passados ao longo desses anos.

Ao PPGCC – Programa de Pós Graduação em Ciência da Computação e a todos os docentes do Departamento de Ciências de Computação e Estatística (DCCE) pelas disciplinas e conhecimentos transmitidos durante minha formação.

Agradeço também à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pela bolsa de mestrado concedida para realização deste projeto.

A todos os amigos que estão sempre juntos.

“A mente que se abre a uma nova ideia
jamais voltará ao seu tamanho original”
Albert Einstein

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ART: *Adaptive Resonance Theory*
DNS: *Domain Name System*
FTP: *File Transfer Protocol*
GWR: *Grow When Required*
HIDS: *Host Intrusion Detection System*
IDS: *Intrusion Detection System*
IETF: *Internet Engineering Task Force*
IP: *Internet Protocol*
IPFIX: *IP Flow Information Export*
IPS: *Intrusion Prevention System*
MINDS: *Minnesota Intrusion Detection System*
NIDS: *Network Intrusion Detection System*
SSH: *Secure Shell*
SOM: *Self-Organizing Map*
SONDE: *Self-Organizing Novelty Detection*
SQL: *Structured Query Language*
TCP: *Transmission Control Protocol*
UDP: *User Datagram Protocol*

RESUMO

Este trabalho apresenta um sistema de detecção de eventos em redes de computadores baseado em métodos não-supervisionados. O sistema possui como base a utilização do padrão IPFIX (IP *Flow Information Export*) para exportação de informações sumarizadas de redes de computadores. O projeto tem como princípio a utilização de métodos não-supervisionados para a detecção de novidades. Detecção de novidade, neste projeto, é definida como detecção de eventos de rede que não são conhecidos previamente. O projeto mostra-se pioneiro na utilização de detecção de novidades baseada em métodos não-supervisionados utilizando fluxos bidirecionais, empreendendo características importantes aos sistemas de segurança computacional, como escalabilidade no monitoramento de redes de alta velocidade, detecção rápida a tentativas ilícitas de acesso, intrusão e ataques de negativa de serviço (DoS), consideradas grandes ameaças atualmente na Internet.

Palavras-chave: não-supervisionado, segurança, rede de computadores

ABSTRACT

In this work is presented an event detection system in a computer network based on unsupervised methods. The system is based on the use of the IPFIX standard (IP Flow Information Export) to export the summarized information of a computer network. The project is based in the use of unsupervised methods for novelty detection. Novelty detection, in this project, is defined as detection of network events that are not previously known. The project is pioneer in the use of novelty detection based on unsupervised methods using bidirectional flows, containing important characteristics to computer security systems, such as scalability in high-speed networks monitoring, fast detection of illicit activities, intrusion and denial of services attacks, considered major threats on the Internet today.

Keywords: unsupervised, security, network computer

Capítulo 1 – Introdução

1.1 Considerações Iniciais

Atualmente, a grande comodidade, eficiência e facilidade proporcionada pelos computadores e pela Internet propiciaram um grande aumento e crescentes avanços nessa área. Observa-se um crescimento na quantidade de dados sensíveis armazenados e trafegados pela rede, além disso, cada vez mais ferramentas e aplicações estão sendo desenvolvidas sem agregar segurança no desenvolvimento de seu projeto, o que leva ao crescimento do número de vulnerabilidades. Combinando o grande número de dados sendo transmitidos pela rede ou armazenados em computadores juntamente com programas vulneráveis, vê-se um cenário de grande facilidade de exploração por pessoas mal intencionadas que podem comprometer um sistema e/ou roubar informações.

Segundo o CERT, os principais ataques reportados no primeiro trimestre de 2011, são *scan*, tentativa de fraude e *worm* (CERT.BR, 2011). Assim, para auxiliar a proteção de computadores e redes de computadores surgiram novas metodologias e *software*. Nesse contexto, surgem os *firewalls*, antivírus, IDSs (*Intrusion Detection System*), IPSs (*Intrusion Prevention System*), entre outros. Os *firewalls* são dispositivos constituídos pela combinação de *software* e *hardware*, utilizados para dividir e controlar o acesso entre redes de computadores. Um tipo específico é o *firewall* pessoal, que é uma aplicação utilizada para proteger um computador contra acessos não autorizados vindos da Internet, ou até mesmo da rede local. Se alguém ou algum programa suspeito tentar se conectar ao seu computador, um *firewall* bem

configurado entra em ação para bloquear tentativas de invasão, podendo barrar também o acesso a *backdoors*, mesmo se já estiverem instalados em seu computador (CERT.BR, 2006).

Os IDSs são *software*, *hardware* ou combinação de ambos utilizados para detectar uma atividade intrusa. Um IDS pode ter diferentes capacidades dependendo de quão complexo e sofisticado são seus componentes. Um IDS pode utilizar tanto assinaturas, técnicas baseadas em anomalia ou ambos, podendo operar a nível de rede (NIDS – *Network* IDS) ou de *hosts* (HIDS – *Host* IDS) (REHMAN, 2003). Tanto os IDSs como *firewalls* são instalados em pontos estratégicos de uma rede de computadores, para assim poder analisar todo o tráfego da rede.

1.2 Identificação do problema e justificativa

Dentre os tipos de IDSs, o NIDS tornou-se uma tendência na comunidade de segurança da informação, pois possibilita a análise em âmbito de rede, e não apenas a análise individual de cada computador, como nos HIDS. Os IDSs podem ser baseados tanto em anomalia quanto em abuso. A grande dificuldade de IDSs baseados em abuso é a detecção de novos ataques, pois estes utilizam o comportamento de ataques previamente conhecidos para realizar a detecção. Já os IDSs baseados em anomalia possuem grande quantidade de falso-positivo (atividade considerada ataque, mas que na verdade é uma atividade lícita). Atualmente, o grande desafio na comunidade de segurança da informação é a detecção de ataques que não são conhecidos, também denominados novidades. Existem metodologias que tem como objetivo a detecção de novidades, como por exemplo, redes neurais e métodos estatísticos. Para esse trabalho, são utilizados métodos que fazem uso da abordagem não-supervisionada, ou seja, os padrões são apresentados para a rede e esta se encarrega de agrupar aqueles que possuem características similares. Para esses métodos não é necessário um prévio conhecimento dos grupos (ataques conhecidos), diferentemente da abordagem supervisionada.

Ainda no contexto de IDSs, o grande desafio de um NIDS é realizar a análise do tráfego de uma rede de computadores de grande porte. Ou seja, um NIDS deve consumir o mínimo possível dos recursos do dispositivo em que ele esteja instalado.

Sistemas como o SNORT (SOURCEFIRE, 2008), que são implementados em dispositivos como *firewalls*, se adaptam bem em ambientes de rede de pequeno e médio porte, mas devido a sua abordagem de análise de conteúdo para cada pacote trafegado, seu uso em redes de grande porte torna-se computacionalmente custoso.

No contexto de análise de tráfego de redes surge uma alternativa viável em termos computacionais: o padrão IPFIX (QUITTEK; *et al*, 2004). Criado por um grupo de trabalho do IETF (*Internet Engineering Task Force*), o padrão IPFIX propõe uma série de especificações para exportação de informações de rede através de dispositivos localizados estrategicamente nesses ambientes, como roteadores, *switches* e até mesmo computadores. Tais especificações foram implementadas por diversos protocolos, como por exemplo, o *NetFlow* desenvolvido pela Cisco que exporta fluxos unidirecionais. Diversos NIDSs utilizam o *Netflow* para a detecção de eventos em redes de computadores, como por exemplo, o MINDS e o ACHOW (vide seção 3.1) que utilizam a versão 5 do protocolo. Recentemente, em 2008, foi proposto um novo padrão para exportação de fluxos bidirecionais, definido no documento RFC 5103 (TRAMMELL; BOSCHI, 2008). As características desse novo protocolo implicam na diminuição do tamanho do banco de dados que armazena as informações do tráfego de rede o que reduz o tempo de consulta e permite correlacionar os eventos de maneira mais eficiente.

Em relação a detecção de ataques e monitoramento de redes de computadores, com a diminuição do tamanho do banco de dados devido as características do novo protocolo, o tempo de consulta no banco é menor e é possível correlacionar os eventos de maneira mais eficiente.

Esta dissertação tem como objetivo descrever o estado da arte de trabalhos de fluxos (bidirecionais e unidirecionais) com aplicação a detecção de intrusão e técnicas para detecção de novidades. Através das pesquisas relacionadas é proposto um novo sistema para identificação de eventos, que aliará a aplicação das técnicas de detecção de novidades utilizando métodos não-supervisionados em informações fornecidas pelo padrão IPFIX, buscando-se a identificação, associação e correlação de eventos em redes de computadores.

1.3 Organização da dissertação

Este documento é dividido como se segue: no capítulo 2 é descrita a fundamentação teórica relativa ao tema, como os conceitos de ataque em redes, fluxos bidirecionais de rede e métodos para detecção de novidades. No Capítulo 3 são descritas as pesquisas mais recentes da área, relacionadas à detecção de novidade e fluxos de dados de redes de computadores. No capítulo 4 é descrito o sistema implementado e o ambiente utilizado para a detecção de eventos em redes de computadores. No capítulo 5 são apresentados os resultados obtidos. Por fim, no capitulo 6 são feitas considerações finais sobre o tema.

Capítulo 2 – Fundamentação teórica

Este capítulo tem como objetivo descrever toda fundamentação teórica das tecnologias que foram utilizadas para o desenvolvimento deste projeto. Na seção 2.1 são descritos conceitos de segurança em computadores e redes, sendo detalhados alguns tipos de ataques relacionados a este trabalho. Em seguida na seção 2.2 são descritos os conceitos relativos ao padrão IPFIX e fluxos de rede. Na seção 2.3 são abordados métodos não-supervisionados para detecção de novidade. Por fim, na seção 2.4 são feitas considerações finais do capítulo.

2.1 Segurança de Computadores e Redes

Segurança da informação compreende um conjunto de medidas que visam proteger e preservar informações e sistemas computacionais. Segundo Gollmann (GOLLMANN, 1999), segurança de computadores e redes é baseada nos seguintes requisitos:

- **Autenticidade:** garante ao receptor que a mensagem é realmente procedente da origem informada em seu conteúdo;
- **Confidencialidade:** garante acesso a uma informação específica somente às pessoas devidamente autorizadas;
- **Integridade:** garante que dados não foram alterados, seja acidentalmente ou intencionalmente, prejudicando a veracidade das informações;

- **Disponibilidade:** garante que serviços/recursos de um sistema estão disponíveis sempre que forem necessários.

Esses elementos constituem os quatro pilares da segurança da informação e, portanto, são essenciais para assegurar a segurança em sistemas computacionais. Assim, a adequação de sistemas computacionais a esses requisitos é fundamental para a garantia da segurança em ambientes computacionais.

Dessa forma, um ataque pode ser definido como qualquer ação que vise subverter pelo menos um dos quatro requisitos da informação. Diversas técnicas para subversão de sistemas computacionais e redes de computadores são difundidas no meio digital, sendo acessíveis para qualquer pessoa conectada à Internet. Alguns dos principais ataques são descritos a seguir.

2.1.1 Prospecção de rede

A prospecção de rede também chamada de varredura de rede ou mapeamento de rede normalmente precede outros tipos de ataque mais específicos, sendo considerado como a fase de reconhecimento do ambiente alvo. A prospecção de rede de computadores tem como objetivo a identificação de quais computadores estão ativos e quais serviços estão sendo disponibilizados por eles. É amplamente utilizado por atacantes para identificar potenciais alvos, pois permite associar possíveis vulnerabilidades aos serviços habilitados em um computador (CERT.BR, 2006).

2.1.2 Ataque de Negação de Serviço

Um ataque de negação de serviço, ou DoS (*Denial of Service*), constitui em um cenário no qual o atacante utiliza apenas um computador para paralisar um serviço em execução ou tirar de operação um computador conectado à Internet (CERT.BR, 2006). O intuito do ataque é tornar o serviço indisponível como, por exemplo, servidores de e-mail, servidores de nomes (DNS), entre outros. É importante ressaltar que em um ataque de negação de serviço o alvo não é comprometido. Existem diferentes maneiras no qual um ataque DoS pode ser aplicado, entre elas estão a

geração de uma grande sobrecarga no processamento de dados de um computador, um grande volume de tráfego de dados para uma rede, de modo que qualquer computador fique indisponível e tire serviços importantes do ar, impossibilitando o uso do serviço.

Existe também uma variação de DoS, que é chamado de ataque de negação de serviço distribuído, ou DDoS (*Distributed Denial of Service*). O DDoS constitui na utilização de dois ou mais computadores para tirar de operação um ou mais serviços ou computadores conectados à Internet.

2.1.3 Códigos Maliciosos

Códigos maliciosos ou *malware*, como o próprio nome já diz, são programas especificamente desenvolvidos para executar ações danosas em um computador. Existem diferentes tipos de *malware*, dentre eles estão (CERT.BR, 2006):

- **Vírus:** são normalmente programas ou parte de programas maliciosos, no qual dependem da execução de outro programa ou arquivo, denominado hospedeiro, para que possa se tonar ativo e assim se propagar. O vírus se propaga inserindo cópias de si mesmo e se tornando parte de outros programas e arquivos;
- ***Worm*:** diferentemente de um vírus, o *worm* não embute cópias de si mesmo em outros programas ou arquivos e não necessita ser explicitamente executado para se propagar. O *worm* é capaz de se propagar automaticamente através da rede, enviando cópias de si mesmo para outros computadores, explorando vulnerabilidades existentes ou falhas na configuração de *software* instalados em computadores;
- **Cavalos de Tróia:** diferentemente do vírus e *worms* ele não infecta outros arquivos e nem propaga cópias de si mesmo automaticamente. Cavalos de tróia são programas, geralmente recebidos por meio de cartões virtuais, álbum de fotos, protetor de tela, que além de executar funções para as quais foi aparentemente projetado, também executa funções maliciosas. Dentre as diversas funções maliciosas que podem ser executas por um cavalo de tróia

estão o furto de senhas, números de cartões de crédito, alterações de arquivos, entre outros;

- ***Backdoors*:** programas que provê ao atacante um meio para retornar ao computador comprometido sem precisar recorrer aos métodos utilizados na realização da invasão. Na maioria dos casos, a intenção do atacante é poder retornar ao computador comprometido sem ser notado;
- ***Rootkits*:** programas que fornecem diversas funcionalidades como, por exemplo, instalação de *backdoors*, programas que realizam a remoção de evidências em arquivos de *logs*, mecanismos para esconder atividades e informações deixadas por um invasor, ferramentas para a realização de prospecção de rede, entre outras.

2.1.4 Ataques de dicionário e ataques de força bruta

Em um ataque de dicionário o atacante possui um banco de dados com possíveis usuários e senhas, que geralmente são utilizados pelos usuários. Esse banco de dados, também chamado de dicionário, é utilizado por um *script* ou ferramenta automatizada para tentar obter acesso a um serviço remoto executando na máquina alvo. O ataque, na maioria das vezes, é executado realizando conexões rápidas e sequenciais em um mesmo serviço (porta). Dentre os serviços mais visados, está o SSH, Telnet, FTP, entre outros.

Uma variante desse ataque é o ataque de força bruta, que tem o mesmo objetivo, porém a técnica ao invés de utilizar um dicionário de palavras, utiliza todas as possibilidades de formação de palavras. Dessa forma, o tempo de descoberta de usuário e senha é bem maior do que no ataque de dicionário, contudo cobre um número muito maior de possibilidades.

2.2 Fluxos de rede bidirecionais e unidirecionais

Atualmente o protocolo *Netflow*, desenvolvido pela Cisco, encontra-se na sua versão 9, contudo para fins didáticos, neste capítulo, é abordado a sua versão 5,

considerada mais simples que a versão 9. Um fluxo de rede é definido como uma sequência unidirecional de pacotes com algumas propriedades em comum que passam em um dispositivo de rede (CLAISE, 2004). De maneira geral, pode-se dizer que o *Netflow* provê a sumarização de tráfego de um dispositivo de rede. Um fluxo é o agrupamento de informações formado pelos seguintes atributos que possuem o mesmo valor: endereço IP de origem e de destino; porta de origem e destino (referente ao protocolo da camada de transporte); valor do campo *Protocol* do datagrama IP; byte *Type of Service* do datagrama IP; e interface lógica de entrada do datagrama no dispositivo de rede. Na figura 2.1 é possível observar os campos do cabeçalho e o formato do protocolo *Netflow*.

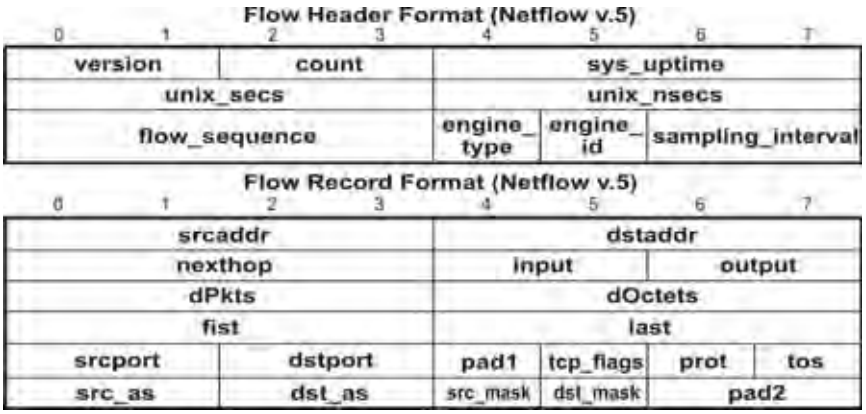


Figura 2.1 Formato de um datagrama *NetFlow*.

Os fluxos gerados pelos equipamentos são exportados para os dispositivos coletores e armazenados. As informações obtidas são uma fonte valiosa de informações, na qual é possível obter detalhes de cada conexão estabelecida no ambiente monitorado.

Uma rede de grande porte gera uma quantidade considerável de informações de rede, mesmo quando utiliza a fluxos de rede *Netflow*. Consequentemente, buscas nesse ambiente é algo computacionalmente custoso em termos de tempo e processamento. Além disso, para cada conexão são gerados dois fluxos distintos (um para cada sentido da conexão) e para correlação de eventos, na maioria das vezes, são necessários ambos os fluxos. Dessa forma, surgiu um novo protocolo para a exportação de fluxos bidirecionais, descrito no RFC 5103 (TRAMMELL; BOSCHI, 2008).

O documento descreve um novo modelo de protocolo baseado no padrão IPFIX intitulado *BiFlow*. Sua principal característica é unir em um único segmento de fluxo a informação bidirecional de uma conexão/sessão. A razão disto é simples: na Internet, a maioria dos protocolos e aplicações baseia-se na comunicação entre dois *hosts*, ou seja, são bidirecionais. Além disso, a criação deste novo modelo vem eliminar informações duplicadas que existiam nos fluxos unidirecionais como, por exemplo, os campos *tos* e *protocol* de um fluxo *NetFlow*. Na figura 2.3 é ilustrada a união de dois fluxos unidirecionais em um único fluxo bidirecional.

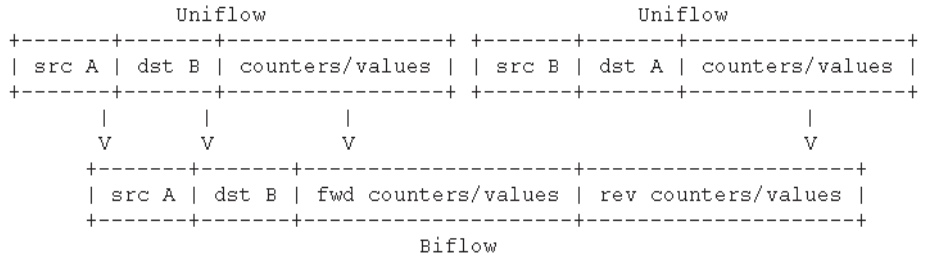


Figura 2.2 Fluxos unidirecionais unificados em um fluxo bidirecional (TRAMMELL; BOSCHI, 2008).

Com base no *BiFlow*, pode-se obter uma redução significativa no armazenamento das informações provindas pelo protocolo. De acordo com o documento, em alguns casos, mesmo utilizando-se o *Biflow* algumas conexões são representadas por fluxos unidirecionais, pois nem sempre é possível determinar os dois lados de uma conexão (origem e destino). Comparando o número de atributos de um fluxo unidirecional com bidirecionais, apesar do número de atributos de fluxos bidirecionais serem maior, uma consulta em um banco de dados tem menor custo computacional do que a mesma consulta realizada em um banco com um maior número de fluxos com menor quantidade de atributos.

2.3 Metodologias para detecção de novidades

Detecção de novidade é a identificação de dados ou sinais, novos ou desconhecidos, que um sistema não teve conhecimento durante a fase de treinamento. Detecção de novidade é um dos requisitos fundamentais de um bom sistema de classificação ou identificação, uma vez que algumas vezes os dados de

testes contêm informações sobre objetos que não eram conhecidos no momento do treinamento do modelo.

Desse modo, a detecção de novidade resolve um grande problema comum que ocorre em aplicações de aprendizado de máquina: nem sempre temos amostras de todas as classes disponíveis para o treinamento. Um exemplo que pode ser citado é o caso de detecção de intrusão, que é o tema deste trabalho: não são conhecidos todos os tipos de ataques existentes.

Detecção de novidade é uma tarefa complicada e desafiadora e por essa razão existem diversos modelos de detecção de novidade para diferentes tipos de dados. É claramente evidente que não há um modelo de detecção de novidade que atenda todas as necessidades (MARKOU; SINGH, 2003).

2.3.1 Técnicas Estatísticas

Para detectar novidade é preciso conhecer a distribuição dos dados, o que pode ser feito por meio da função densidade de probabilidade. Existem duas principais abordagens para a estimação da função de densidade de probabilidade: paramétrica e não-paramétrica (MARKOU; SINGH, 2003).

Técnicas estatísticas paramétricas consideram modelos previamente estabelecidos de distribuição e calculam os parâmetros necessários para adequar esses modelos aos dados. Métodos paramétricos têm seu uso limitado uma vez que necessitam de um conhecimento prévio dos dados. Essa dificuldade pode ser superada com o uso de técnicas não-paramétricas, em que a estimação da função de densidade de probabilidade é feita apenas com base nos exemplos do conjunto de treino. Como resultado, métodos não-paramétricos proveem uma grande flexibilidade em sistemas gerais. Entre os métodos não-paramétricos utilizados para detecção de novidade estão a janela de *Parzen* e os k vizinhos mais próximos (SPINOSA, 2008).

2.3.1.1 *K-médias*

O K-médias ou também conhecido como K-means é um método heurístico clássico da literatura que possui um algoritmo de aprendizagem que organiza N objetos da base de dados em K partições onde cada um representa um cluster. O método k-médias é um algoritmo simples, escalável e pode ser facilmente modificado para lidar com fluxo de dados e grandes bases de dados (NALDI, 2011).

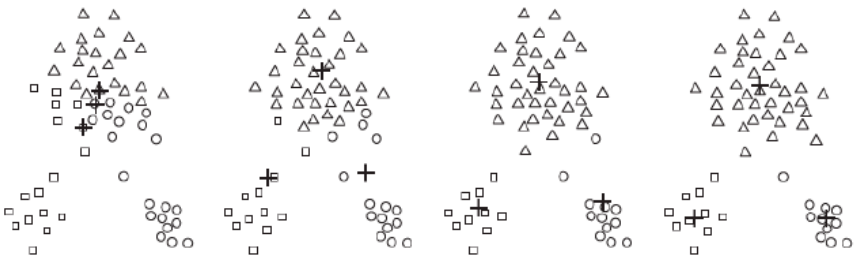
O funcionamento do k-médias consiste em particionar um conjunto de dados N em k grupos com base em uma medida de dissimilaridade fornecida (NALDI, 2011). O k-médias define um protótipo em termos de um centróide, no qual é geralmente a média de um grupo de pontos, e é tipicamente aplicado em objetos em um espaço continuo e n-dimensional (TAN; STEINBACH; KUMAR, 2006).

O algoritmo básico inicializa os grupos por meio de um conjunto de k protótipos, ou seja, pontos que representam estes grupos, que é um parâmetro especificado pelo usuário. Cada ponto é atribuído a um centróide mais próximo, e cada coleção de pontos atribuído a um centróide é um cluster. O centróide de cada cluster é atualizado baseado em cada ponto atribuído ao cluster. Os passos de atribuição e atualização são repetidos até que os centróides continuem os mesmos.

O k-médias é formalmente descrito pelo algoritmo que pode ser visualizado na figura 2.4 (TAN; STEINBACH; KUMAR, 2006). A operação do k-médias é ilustrado na figura 2.5.

```
1: Selecione K pontos como inicial centróides.
2: repita
3:    Forme K clusters pela atribuição de cada ponto a seu centróide
      mais próximo.
4:    Recalcule o centroide para cada cluster.
5: até que os centróides não mudem.
```

Figura 2.3 Algoritmo básico do k-médias.



Iteração 1. Iteração 2. Iteração 3. Iteração 4.

Figura 2.4 Funcionamento do k-médias.

2.3.1.2 *X-médias*

O x-médias ou ainda x-means é uma extensão do algoritmo k-médias. O algoritmo x-means foi proposto para gerar uma partição do conjunto de dados por meio do uso do k-means. O algoritmo recebe como parâmetro de entrada uma base de dados que será particionada e um intervalo de números que indicam o número de mínimo e máximo de clusters que a base deve ser particionada (NALDI, 2011).

Em essência, o algoritmo começa com k igual ao limite inferior do intervalo dado e continua adicionando centróides onde são necessários, até que o limite superior seja alcançado (PELLEG; MOORE, 2000). O funcionamento do x-means pode ser descrito no algoritmo descrito na figura 2.6.

```
1:  Inicialize o algoritmo por meio da aplicação de k-médias ao
    conjunto de dados de forma a gerar uma partição com o número
    mínimo de clusters.
2:  Avalie a partição resultante.
3:  repita
4:    Divida cada cluster em 2 por meio do k-médias.
5:    Avalie as divisões e mantenha as divisões que geraram melhores
      partições.
6:    Se nenhuma partição foi mantida faça
7:        Divida uma proporção dos grupos formados inicialmente, a
          partir das divisões que resultem nas melhores avaliações.
8:    Aplique o algoritmo k-médias para refinamento da partição
      resultante.
9:    Se a partição resultante for a melhor faça
10:       Armazena a partição.
11: até que a partição possua o número máximo de clusters.
12: Retorne a melhor partição encontrada.
```

Figura 2.5 Algoritmo básico do x-médias.

2.3.2 Técnicas baseadas em redes neurais artificiais

As redes neurais artificiais são técnicas de aprendizado de máquina inspiradas no funcionamento do cérebro. Assim como neurônios biológicos ligam-se uns aos outros para receber, processar e transportar sinais através de uma rede complexa, o neurônio artificial, unidade fundamental das redes neurais, também é responsável por receber um conjunto de sinais, processá-los e emitir um sinal de saída (SPINOSA, 2008).

Um neurônio básico consiste das seguintes características (BALESTRASSI, 2000):

- **Entradas:** representações numéricas das características de entrada do domínio do problema, podendo ser uma variável binária ou contínua;
- **Pesos:** representam coeficientes adaptativos para cada uma das entradas;
- **Bias:** meio alternativo de representar um limiar de um neurônio. O bias é tomado tipicamente de fora do corpo do neurônio e conectado a ele usando uma entrada adicional que permanece fixa;
- **Operação de somatório:** é o produto interno do vetor de entrada e vetor de pesos;
- **Função de ativação:** transforma o resultado do somatório na saída do neurônio;
- **Saída:** resultado final.

Um neurônio básico opera da seguinte maneira: ele soma as entradas ajustadas pelos pesos e passa essa soma através da função de ativação para produzir uma saída. Isso pode ser feito de uma forma binária ou contínua dependendo da função de ativação usada. Essa saída é então passada (em muitas arquiteturas) para neurônios subsequentes (BALESTRASSI, 2000).

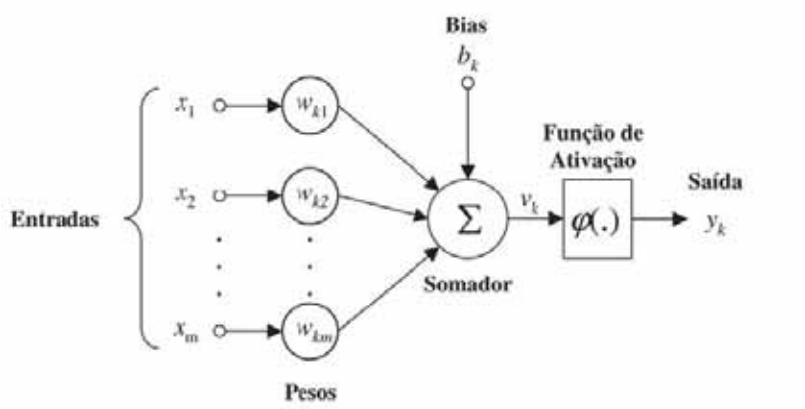


Figura 2.6 Representação de um neurônio artificial (MEDEIROS, 2009).

A maneira pela qual os neurônios de uma rede estão estruturados está intimamente ligada com o algoritmo de aprendizagem utilizado para treinar a rede. Os algoritmos de treinamento podem ser classificados de acordo com o paradigma de aprendizado utilizado. Dessa forma, o processo de treinamento pode ser dividido basicamente em supervisionado e não-supervisionado. No treinamento

supervisionado supõe-se a existência de um direcionador externo que orienta a rede neural para as saídas desejadas. Já no treinamento não-supervisionado não existe um direcionador, fazendo com que os resultados produzidos pela rede neural sejam considerados o melhor processamento possível obtido a partir dos dados disponíveis. (NAGANO; BENITE; SOBREIRO, 2007). A desvantagem do treinamento supervisionado é que valores de respostas para cada padrão de treinamento devem ser conhecidos (BALESTRASSI, 2000). Dessa forma, redes neurais não-supervisionadas possuem características que melhor se enquadram na utilização na detecção de novidade.

As redes neurais têm sido amplamente utilizadas para a detecção de novidade, com aplicações em problemas reais de variados domínios. Entre as diversas abordagens de redes neurais utilizadas para esse fim, são utilizadas as redes SOM (*Self-Organising Maps*), redes GWR (*Grow When Required*) (MARSLAND; SHAPIRO; NEHMZOW, 2002), redes ART (*Adaptive Resonance Theory*) (CARPENTER; GROSSBERG, 1998) e as redes SONDE (*Sefl-Organizing Maps*) (ALBERTINI; MELLO, 2007).

2.3.2.1 *Self-Organizing Map*

O Mapa Auto-Organizável, do inglês *Self-Organizing Map* (SOM), proposto por Kohonen, busca capturar as características essenciais dos mapas computacionais do cérebro e ainda se manter tratável do ponto de vista computacional (HAYKIN, 1998).

A rede neural SOM pertence à classe de redes neurais não-supervisionadas que baseiam-se no processo de aprendizagem competitiva, onde somente um neurônio de saída ou grupo local de neurônios fornece uma resposta ativa a um sinal de entrada corrente. O nível de ativação indica a similaridade entre o vetor de dados de entrada e o vetor de pesos do neurônio. Uma forma usual de expressar a similaridade é através da distância euclidiana entre esses vetores. Uma vez que a distância entre o vetor de pesos de um determinado neurônio e o vetor de dados de entrada é a mínima para todos os neurônios da rede, esse neurônio juntamente com um conjunto pré-definido de neurônios vizinhos terá seus pesos automaticamente reajustados pelo

algoritmo de aprendizagem da rede. A vizinhança de cada neurônio pode ser definida de acordo com a forma geométrica usada para representar os neurônios da rede, como por exemplo, na forma de um *array* retangular ou hexagonal (GONÇALVES; ANDRADE NETTO; ZULLO JÚNIOR, 1996).

A rede neural SOM é caracterizada pela formação de um mapa topográfico dos padrões de entrada no qual as localizações espaciais dos neurônios na grade (linha e coluna) são indicativas das características estatísticas intrínsecas contidas nos padrões de entrada especificados.

Uma vez que a grade tenha sido propriamente inicializada, há três passos básicos envolvidos na formação do mapa auto-organizável (HAYKIN, 1998):

- **Competição**: Para cada padrão de entrada, os neurônios da grade calculam seus respectivos valores de uma função discriminante, que fornece a base para a competição entre os neurônios. O neurônio particular com o maior valor da função discriminante é declarado vencedor da competição;
- **Cooperação***:* O neurônio vencedor determina a localização espacial de uma vizinhança topológica de neurônios estimulados, fornecendo assim a base para a cooperação entre os neurônios vizinhos;
- **Adaptação Sináptica***:* Esse mecanismo permite que os neurônios estimulados aumentem seus valores individuais da função discriminante em relação ao padrão de entrada através de ajustes aplicados aos seus pesos sinápticos. Os ajustes são tais que a resposta do neurônio vencedor à aplicação subsequente de um padrão de entrada similar é melhorada.

De modo geral, o algoritmo de aprendizagem do SOM pode ser descrito em três passos (GONÇALVES; ANDRADE NETTO; ZULLO JÚNIOR, 1996):

- Passo 1: Selecione uma padrão de treinamento e forneça como entrada à rede;
- Passo 2: Calcule as distâncias entre o vetor de entradas e o vetor de pesos de cada neurônio da rede;
- Passo 3: Selecione um neurônio com a distância mínima entre todos os outros neurônios e ajuste o seu vetor de pesos e de seus vizinhos.

No final do processo de aprendizagem cada neurônio ou grupo de neurônios vizinhos representará um padrão distinto dentro do conjunto de padrões fornecidos como entrada para a rede.

Uma das vantagens de se utilizar o SOM é que ele possui duas propriedades que não são encontradas em outros métodos não-supervisionados de *clustering*. Tais propriedades são:

- Preservar as relações topológicas (métricas) entre os vetores de dados de entrada;
- Produzir uma aproximação da função densidade de probabilidade dos vetores de dados de entrada.

2.4 Considerações finais

Este capítulo apresentou os principais conceitos e tecnologias envolvidos neste trabalho. Os conceitos de fluxo de dados *NetFlow*, *Biflow* e detecção de novidade são a base para o desenvolvimento do projeto. No próximo capítulo é descrito o estado da arte do assunto, relacionando diversos trabalhos que atuam na área de detecção de intrusão e detecção de novidade.

Capítulo 3 – Trabalhos relacionados

Neste capítulo são apresentados os trabalhos relacionados com o projeto proposto. Existem diversos trabalhos relacionados a segurança de sistema computacionais e detecção de novidade, contudo observa-se a ausência de trabalhos que utilizem em conjunto fluxos de rede IPFIX e detecção de novidade. Na seção 3.1 são descritos os trabalhos relacionados a detecção de intrusão utilizando fluxos de rede IPFIX. Em seguida, na seção 3.2, são descritos os trabalhos relacionados a detecção de novidade. Já na seção 3.3 são descritos os trabalhos que abordam detecção de intrusão utilizando métodos não-supervisionados. Por fim, na seção 3.4 são feitos os comentários finais.

3.1 Fluxos de redes e detecção de intrusão

Dentre os trabalhos que utilizam fluxos de redes IPFIX aplicados a detecção de eventos em redes de computadores está o trabalho de (CORRÊA *et al.*, 2009). O trabalho aborda uma nova metodologia para detecção de eventos em redes de computadores utilizando o protocolo *NetFlow* versão 5 e os dados são armazenados em um banco de dados relacional. O modelo utiliza duas vertentes na detecção dos eventos: a baseada em abuso e a baseada em anomalias. A arquitetura do sistema coleta, armazena e processa os fluxos confrontando-os com uma base de assinaturas de ataques para identificação de ataques. Além disso, utiliza a detecção de anomalia

para detecção que não possuem assinaturas e produz relatórios para o administrador. As assinaturas utilizadas pelo sistema são baseadas em passos e podem ser cadastradas por um administrador por meio de uma interface web. A arquitetura do sistema proposto está descrita na Figura 3.1.

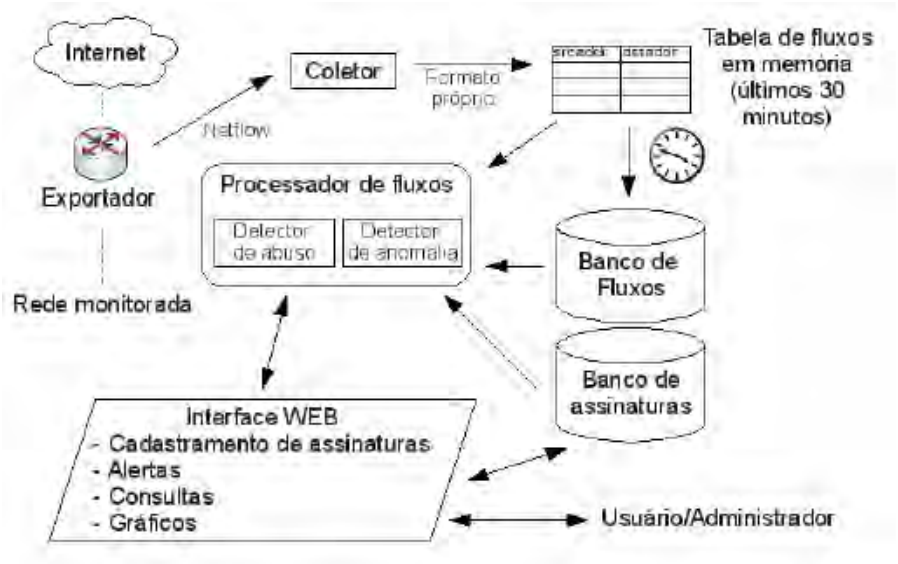


Figura 3.1 Arquitetura do sistema proposto por (CORRÊA *et al*, 2009).

Outro trabalho é o *Minnesota Intrusion Detection System* (MINDS) (GOGOI; BORAH; BHATTACHARYYA, 2010) é sistema baseado em *data mining* (mineração de dados) para detecção de intrusão em redes de computadores. O MINDS utiliza dados *Netflow* versão 5 coletados utilizando a ferramenta *flow-tools* (FLOW-TOOLS, 2010). O analista utiliza o MINDS para analisar em lotes os arquivos de dados coletados. A razão para executar o sistema no modo em lote não é devido ao tempo que é utilizado para analisar esses arquivos, mas porque é conveniente para o analista. Antes dos dados alimentarem o módulo detector de anomalia, uma etapa de filtragem dos dados é executada para remover tráfego que o analista não está interessado em analisar. O primeiro passo do MINDS é extrair as importantes características utilizadas na análise *data mining*. Após o passo de seleção de características, o módulo de detecção de ataque é utilizado para detectar conexões de rede que correspondem a ataques na qual a assinatura está disponível, e remove essas conexões de análises futuras. Depois, os dados alimentam o módulo de detecção de anomalia que utiliza um algoritmo de detecção de *outlier* (ponto discrepante). Por fim, um analista humano tem que analisar apenas as conexões anômalas e que não possuem assinaturas para determinar se elas são ataques reais ou

outro comportamento interessante. O módulo de análise de associações de padrões do MINDS sumariza conexões de rede que são comumente classificadas como anômalas pelo módulo de detecção de anomalia. O analista, depois de analisar os resumos criados, decide se estes resumos são úteis na criação de novas regras que podem ser utilizados na detecção de ataque conhecidos.

Já o trabalho de (ZHENQI; XINYU, 2008) propõe um sistema de detecção de intrusão que também utiliza o *Netflow* para detectar ataques como DDoS e disseminação de *worms*. Além de detectar ataques por meio de comparações entre os fluxos do ambiente e os fluxos considerados de ataques, o trabalho também propõe técnicas de contramedidas para esses ataques, como por exemplo, regras de bloqueio em roteadores ou *firewalls*. Entretanto, o trabalho possui taxas altas de falsos positivos na sua detecção.

3.2 Detecção de novidade

No trabalho de (SPINOSA, 2008), a detecção de novidade é tratada como o problema de identificação de conceitos emergentes em dados que podem ser apresentados em um fluxo contínuo. O trabalho propõe uma nova abordagem para detecção de novidade em fluxo de dados continuo. O OLINDDA (*OnLIne Novelty and Drift Detection Algorithm*) concentra-se no aprendizado contínuo não-supervisionado de novos conceitos. Tendo aprendido uma descrição inicial de um conceito normal, prossegue à analise de novos dados, tratando-os como um fluxo contínuo em que novos conceitos podem aparecer a qualquer momento. Com o uso de técnicas de agrupamento, OLINDDA pode empregar diversos critérios de validação para avaliar grupos em termos de sua coesão e representatividade. Grupos que são considerados válidos produzem conceitos que podem sofrer fusão, e cujo conhecimento é continuar incorporado. A técnica é avaliada experimentalmente com dados artificiais e reais. O módulo de classificação com uma classe é comparado a outras técnicas de detecção de novidade, e a abordagem como um todo é analisada sob vários aspectos por meio da evolução temporal de diversas métricas. Apesar de alguns falso-positivos e falso-negativos, os resultados obtidos pelo trabalho foram bons e novos conceitos foram detectados corretamente, objetivo do trabalho. No trabalho, alguns testes voltados para detecção de intrusão foram realizados com

dados da competição *KDD Cup 1999* (ELKAN, 2000), porém não foi o foco do trabalho.

Já no trabalho de (DASGUPTA; FORREST, 1995) é proposto um método para detecção de novidade, no qual é baseado na ideia de sistemas imunológicos. É um método probabilístico que percebe mudanças no comportamento normal sem exigir um conhecimento prévio das mudanças para as quais ele está procurando. Dessa forma, assemelha-se com a abordagem de detecção de novidade pela rede neural ART. Ambas, redes neurais e algoritmos do sistema imunológico são biologicamente inspirados em técnicas que têm a capacidade de identificar padrões de interesse. Os resultados obtidos mostraram que o algoritmo detectou a ruptura da ferramenta, e também pode detectar ruído em sinais.

3.3 Detecção de intrusão e métodos não-supervisionados

NSOM (*Network Self-Organizing Maps*) (GOGOI; BORAH; BHATTACHARYYA, 2010) é um ssistema de detecção de intrusão e pode ser classificado como um sistema de detecção baseado em anomalia. O sistema utiliza a rede neural SOM para classificar dados da camada de enlace em tempo real. Os dados da rede são constantemente coletados com a ferramenta *tcpdump* (TCPDUMP, 2010) de uma determinada porta. Esses dados são processados e características apropriadas para a classificação são selecionadas. O processo de classificação é então iniciado e o resultado da classificação é enviado para uma ferramenta gráfica que exibe as atividades que estão ocorrendo dinamicamente nas portas de rede. A hipótese é que esse tráfico de rotina que representa o comportamento normal seja agrupado em torno de um ou mais clusters centrais e todo tráfego irregular que representa o comportamento anormal ou possivelmente suspeito seja agrupado fora do agrupamento normal. O sistema é capaz de classificar tráfego regular e irregular, e possivelmente, tráfego de rede intrusivo para um determinado computador.

Em (ZANERO; SAVARESI, 2004) é proposta uma nova arquitetura para um sistema de detecção de intrusão baseado em rede utilizando métodos não-supervisionados e técnicas de *data mining*. A arquitetura é constituída de dois níveis: no primeiro é aplicado um algoritmo não-supervisionado de agrupamento no qual

reduz os *payloads* dos pacotes de rede para um tamanho tratável; o segundo é um algoritmo tradicional para detecção de anomalia, cuja eficiência é melhorada pela disponibilidade dos dados do conteúdo dos *payloads* dos pacotes. Para o agrupamento foram testados métodos não-supervisionados tais como a rede neural SOM e o algoritmo k-médias.

Já em (SMITH *et. al*, 2008) é proposto um sistema de correlação de alertas baseado em métodos não-supervisionados que é preciso e de baixa manutenção. O sistema é implementado em dois estágios de correlação. No primeiro, alertas são agrupados de modo que cada grupo forme uma etapa de um ataque. No segundo estágio, os grupos criados no primeiro estágio são combinados de modo que cada combinação de grupos contenha os alertas de precisamente um ataque completo. No trabalho foram testados vários algoritmos, porém o que obteve melhor resultado consiste na abordagem não-supervisionada, baseada na detecção de novidade, utilizada na primeira fase do sistema. Os resultados são experimentais, contudo mostraram que com modelo proposto, o número de alertas no qual um analista tem que lidar é significantemente reduzido.

Em (PENG *et. al.*, 2010) é proposto uma nova aplicação dinâmica para fluxos baseado no comportamento do tráfego que pode identificar e classificar, eficientemente, tráfego de aplicações desconhecidas. O tráfego da rede é capturado por um monitor que primeiramente analisa para pegar as características dos fluxos, no qual serve como um discriminador para identificar certas aplicações. Essas características são representadas como um vetor apropriado, e então é computada a distância do comportamento de tráfego entre eles. Os clusters coletados representam as várias aplicações encontras em redes de computadores no momento da coleta. De acordo com os resultados, os administradores podem obter claramente uma visão do comportamento da rede. Além disso, depois de um longo período de execução, é possível obter clusters relativamente estáveis. Uma vez que um novo cluster aparece, é possível afirmar que pode existir uma nova aplicação ou uma anomalia na rede. De acordo com (PENG *et. al.*, 2010), os resultados do protótipo implementado mostrou ser capaz de identificar a maioria das aplicações da Internet com excelente precisão na taxa de identificação, tanto na identificação de aplicações como anomalias, e bom desempenho.

3.4 Outros

Outro trabalho importante para o desenvolvimento do projeto é o proposto em (CORRÊA; PROTO; CANSIAN, 2008). O trabalho aborda uma nova metodologia para armazenamento dos dados do protocolo *Netflow* em um banco de dados relacional e a utilização de consultas SQL para a detecção de eventos em redes de computadores. O trabalho propõe uma arquitetura para o armazenamento de informações e realiza consultas para a detecção de alguns ataques. Métodos mais robustos para detecção de intrusão aliadas a arquitetura de armazenamento são sugeridos como trabalhos futuros.

Já no trabalho de (HSIAO; CHEN; WU, 2010) é proposta uma arquitetura para um sistema de detecção de sites maliciosos e a utiliza de um método espacial-temporal de agregação de variáveis para construir um módulo de detecção a partir de fluxos *Netflow*. No trabalho, é importante ressaltar os resultados obtidos com as variáveis criadas, derivadas das variáveis originais do *Netflow*. Os resultados mostraram que tais variáveis obtiveram um melhor resultado quando aplicadas as técnicas de detecção do que as variáveis originais do *Netflow*.

Em (PROTO; ALEXANDRE; CANSIAN, 2009) é proposta uma metodologia de detecção de eventos em redes de computadores de larga escala, cujo perímetro de defesa se estende a um ambiente de grande porte. A proposta aborda a detecção de eventos por anomalia utilizando o protocolo *NetFlow*, métodos estatísticos e o monitoramento do ambiente em tempo real. O trabalho utiliza a arquitetura de armazenamento proposto em (CORRÊA; PROTO; CANSIAN, 2008) como suporte ao processo de detecção de eventos em rede. Foram realizados testes com o monitoramento de quatro serviços bastante utilizados, o FTP, SSH, SMTP e o HTTP. Nos resultados foi possível observar que o serviço HTTP apresentou um número maior de falso-positivos explicados pelas características do serviço e pela sua escalabilidade no ambiente. Apesar dos testes terem sido realizados com tais serviços, o modelo pode ser aplicado a qualquer outro serviço. Em relação ao desempenho do sistema, o resultado foi satisfatório, principalmente no que se refere ao tempo de monitoramento do ambiente em tempo real. Tal medida reflete o baixo custo computacional para a análise do tráfego de uma rede de grande porte.

3.5 Considerações finais

Neste capítulo foram descritos os trabalhos relacionados a fluxos de rede IPFIX (*NetFlow*), detecção de intrusão e detecção de novidade. Com base nestes trabalhos, foi desenvolvido um projeto de detecção de eventos utilizando detecção de novidade em fluxos de dados bidirecionais. O desenvolvimento do projeto e o ambiente são apresentados no próximo capítulo.

Capítulo 4 – Metodologia

Neste capítulo é descrita a metodologia utilizada no trabalho desenvolvido. O trabalho abrange a aplicação de métodos não-supervisionados em informações oriundas do padrão IPFIX para a detecção de eventos em redes de computadores. O capítulo é dividido em três seções: na primeira, é descrita a estrutura do ambiente que foi utilizada durante o desenvolvimento e testes deste trabalho; a segunda seção descreve a arquitetura do sistema de detecção de eventos desenvolvido, detalhando os módulos existentes e o funcionamento de cada módulo e por fim, na última seção são feitas as considerações finais sobre o projeto.

4.1 Objetivos

O projeto proposto tem como objetivo o desenvolvimento de um sistema de detecção de eventos em redes de computadores por meio de detecção de novidade baseada em métodos não-supervisionados. Os métodos não-supervisionados utilizados são algoritmos de agrupamento, ou seja, dado um conjunto de dados, tais algoritmos tem como objetivo agrupar os dados semelhantes. No projeto proposto são utilizados os métodos estatísticos k-médias e x-médias e também foi utilizada a rede neural SOM. Como fonte de informação de fluxos bidirecionais foi utilizado um coletor que recebe informações de fluxos unidirecionais *Netflow* versão 5 e gera fluxos bidirecionais (*Biflow*). O projeto envolve a aplicação dos métodos de agrupamento nos fluxos bidirecionais de dados obtidos do ambiente monitorado.

O ambiente utilizado pode ser observado na figura 4.1. Além dos usuários comuns, no ambiente também existe um computador que foi utilizado para aplicar ataques que tem como objetivo gerar tráfego anômalo, este computador é identificado pela cor vermelha na figura. Dentre os ataques aplicados estão: ataque de força bruta, varredura de porta, varredura de rede, ataques de negação de serviço (DoS) utilizando requisições ICMP e varredura de vulnerabilidades WEB. Os alvos dos ataques foram os servidores e as computadores da rede “Ambiente Usuários II“. Para coleta dos dados foi utilizada a ferramenta fprobe (FPROBE, 2011) instalada no *gateway*/*firewall* da rede. Os dados exportados foram enviados para um coletor, identificado na figura pelo computador na cor azul, na rede “Ambiente de Usuários II“. Na figura, ambas as redes foram monitoradas, “Ambiente de Usuários I” e “Ambiente de Usuários II”.

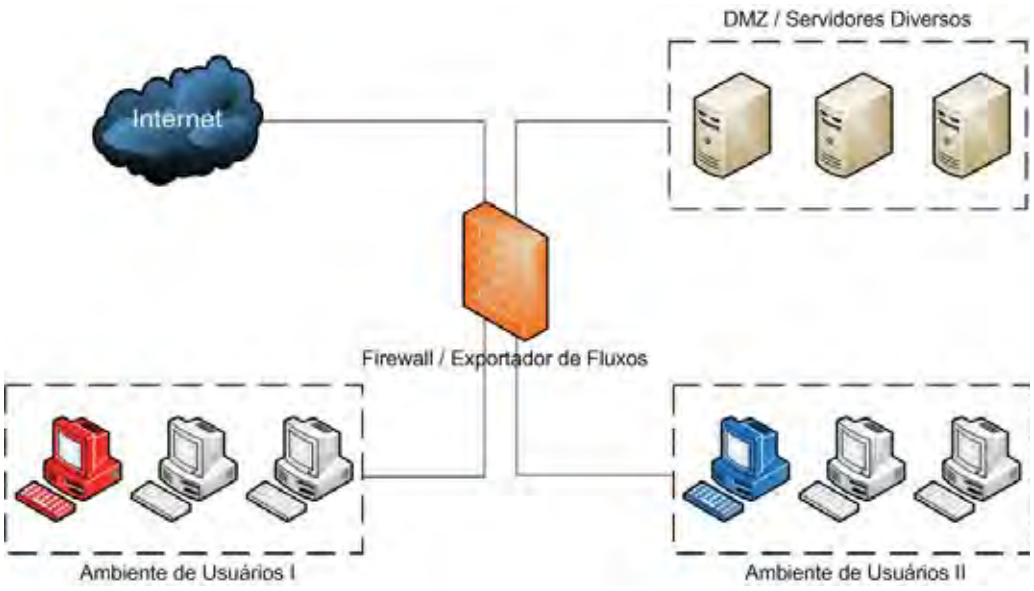


Figura 4.1 Estrutura do ambiente.

4.2 Arquitetura e funcionamento do sistema de detecção de eventos

A arquitetura do projeto pode ser visualizada na figura 4.2. O projeto pode ser divido nos seguintes módulos:

- **Módulo de exportação e coleta:** Este módulo é responsável pela exportação e armazenamento dos dados oriundos dos fluxos de dados unidirecionais e bidirecionais. Os dados são armazenados em um banco de dados relacional que

permite interação por meio da linguagem SQL. O modelo de armazenamento é baseado no trabalho de (CORRÊA; PROTO; CANSIAN, 2008);

- **Módulo de treinamento:** A principal tarefa deste módulo é buscar na base de dados associações de fluxos que possam identificar padrões de tráfego. Além disso, este módulo também é responsável por gerar os dados que serão utilizados pelos algoritmos de agrupamento. Neste módulo, tanto as redes neurais são treinadas como são gerados os modelos dos métodos estatísticos;
- **Módulo de detecção de eventos:** Este módulo executa a detecção de eventos utilizando os métodos implementados: redes neurais e métodos estatísticos. Nesse módulo também são utilizados os limiares para encontrar computadores com comportamento suspeito;
- **Módulo de comparações de métodos utilizados:** Este módulo é responsável por comparar resultados obtidos pelas redes neurais e redes estatísticas.

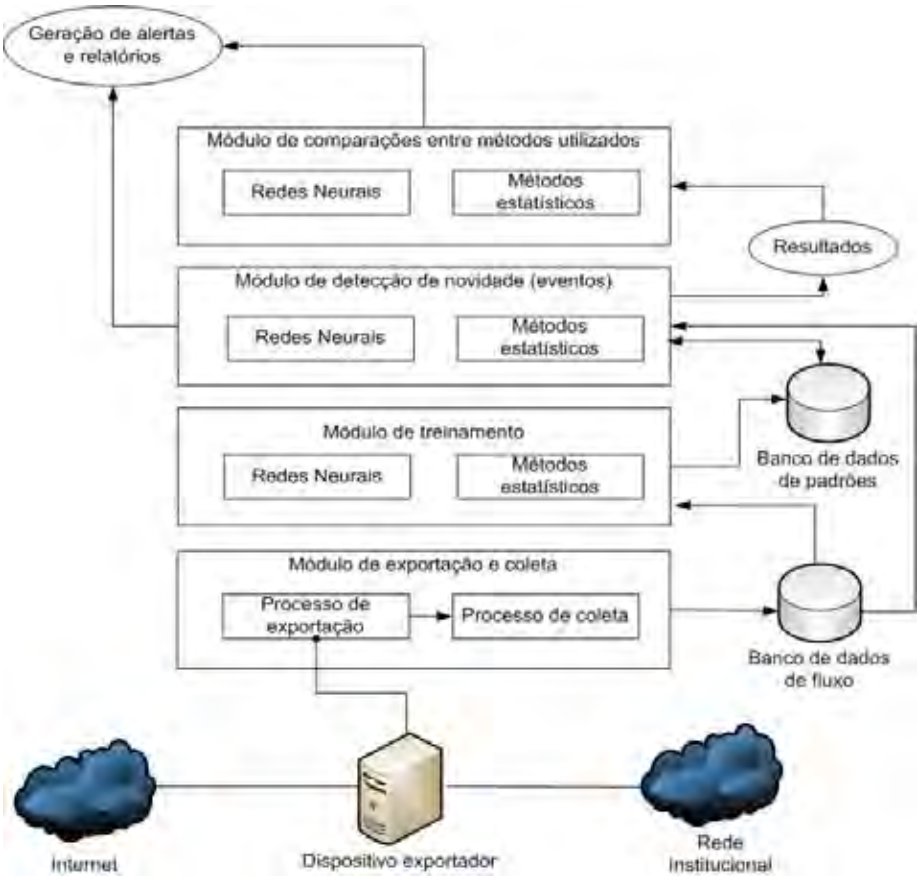


Figura 4.2 Arquitetura do sistema de detecção de eventos.

Dentre os módulos da arquitetura do sistema, o módulo de treinamento também faz o resumo das informações para que sejam utilizadas no treinamento. Os dados

são gerados a cada 1 minuto, obtendo assim uma quantidade significativa de dados a serem analisados. Uma outra característica dos dados coletados é que estes levam em consideração apenas os endereços IPs de origem. Na tabela 4.1 é apresentada uma descrição dos dados coletados.

Tabela 4.1 Dados Coletados.

Nome	**Descrição**
num_con	Número de fluxos/conexões que um determinado endereço IP de origem realizou.
num_syn	Número de conexões que tiveram a *flag* SYN do protocolo TCP ativada.
dist_port	Número de portas distintas de destino que um determinado endereço IP de origem acessou.
num_null	Número de conexões que um determinado endereço IP de origem realizou e que não obteve resposta.
num_rst	Número de conexões que tiveram a *flag* RST do protocolo TCP ativa.
num_pkts	Número de pacotes enviados por segundo.
num_port	Número de portas de destino bem conhecidas (*well-known*) utilizadas tanto no protocolo UDP quanto no protocolo TCP.
num_time	Número de conexões que não são requisições HTTP ou HTTPs que tiveram menos de 15 segundos de duração.
dist_addr	Número de endereços IPs distintos de destino acessados.
num_port_src	Número de portas *well-known* de origem acessadas.

Para utilizar os dados tanto nos algoritmos estatísticos, quanto na rede neural, foi necessária uma normalização desses dados. A normalização dos dados foi realizada de forma empírica, com base nos dados de teste. Dessa forma, para todos os dados relacionados com o número de conexão, num_null, num_pkts, num_port, num_time e num_syn, seguem a equação 1. Para os demais dados foi utilizada a equação 2.

$$N = \frac{N \times 10}{num\_con} \quad (1)$$

$$N_i = N_i \div \left(\left(\sum_0^j N_j \right) \div j \right) \quad (2)$$

O funcionamento dos módulos de treinamento e detecção podem ser visualizados na figura 4.3.

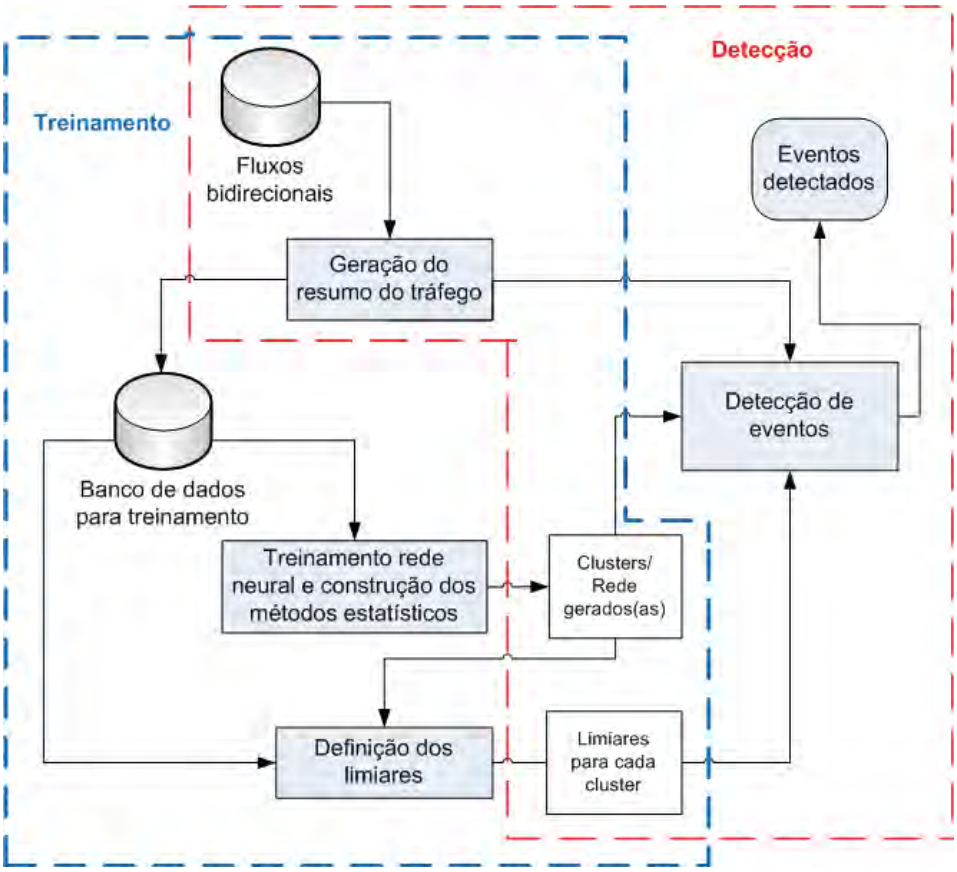


Figura 4.3 Funcionamento do sistema de detecção de eventos.

Os dados coletados, fluxos bidirecionais, passam por um processo de mineração e são selecionados os atributos listados na tabela 4.1. Esses dados são armazenados em um banco de dados para treinamento e são utilizados para o treinamento da rede neural e para a geração dos clusters dos algoritmos estatísticos k-médias e x-médias. Após a geração dos modelos estatísticos e da rede neural, é necessário encontrar os limiares para cada cluster, ou seja, conjunto de dados em que a distância diferiu muito do centro do cluster a que pertence, que pode ser considerado um conjunto suspeito e deve ser analisado. Os limiares foram calculados empiricamente de acordo com os dados obtidos com o conjunto de testes. O módulo

de detecção utiliza os modelos estatísticos e a rede neural para agrupar os dados semelhantes de acordo com os dados utilizados na fase de treinamento. Após a classificação, são aplicados os limiares para a detecção dos conjuntos de dados considerados com o comportamento anômalo.

Para a geração da rede neural SOM foi utilizado um algoritmo retirado de (LUDWIG JUNIOR; MONTGNOMERY, 2007) e para a geração dos clusters dos métodos estatísticos foi utilizado o programa WEKA (WITTEN; FRANK, 2000).

4.3 Considerações finais

Neste capítulo foi descrito o ambiente utilizado e também o sistema desenvolvido, bem como seu funcionamento. Além disso, foi feita uma descrição dos dados utilizados pelos métodos implementados. Para a utilização dos métodos k-médias, x-médias e a rede neural SOM, foi necessário uma normalização dos dados que também foi discutida neste capítulo. No próximo capítulo são apresentados os resultados obtidos com os três métodos utilizados.

Capítulo 5 – Resultados

Neste capítulo são apresentados os resultados obtidos. O capítulo é dividido em cinco seções: na primeira, são apresentados os resultados gerais obtidos com este trabalho, em seguida, as seções 5.2, 5.3 e 5.4 são compostas de resultados obtidos utilizando os métodos k-médias, x-médias e pela rede neural SOM. Por fim, na última seção são feitas as considerações finais do capítulo.

5.1 Resultados Gerais

Para os três métodos, foi utilizado um dia de fluxos de rede para o treinamento e dois dias para a validação. Em todos os dias foram aplicados ataques aleatórios com intervalo de 5 minutos, e cerca de 1 minuto de duração para cada ataque. No primeiro conjunto de dados de validação foi aplicado um número maior de ataques. O segundo conjunto de dados de validação, além dos ataques utilizados no treinamento, também foi feito um ataque de varredura de vulnerabilidades WEB, que não foi realizado na etapa de treinamento para verificação da detecção de novidade, ou seja, ataque que a rede neural não tinha conhecimento prévio.

Na tabela 5.1 é possível visualizar os resultados obtidos com os métodos implementados no dia utilizado para treinamento. No dia de treinamento foram aplicados 34 ataques, que geraram 46 tuplas no banco de dados. O total de tuplas,

tanto de ataques quanto de dados normais foi de 6493. Em uma computador equipado com processador Intel Core 2 Duo, com velocidade de processamento de 3GHz e memória de 4GB, o tempo despendido para construção dos modelos estatísticos k-médias e x-médias foi de aproximadamente 0.74 e 0.65 segundos, respectivamente. Já para o treinamento da rede neural SOM foi despendido aproximadamente 18.03 segundos. Observa-se que os modelos estatísticos foram gerados pelo programa WEKA, enquanto a rede neural foi gerada por um programa no qual faz as buscas dos dados em um banco de dados.

Tabela 5.1 Dados obtidos com os métodos implementados – dia treinamento.

Método	**Total de eventos detectados**	**Falso-positivo**	**Falso-negativo**	**Ataques detectados /Total ataques**	**Ataques detectados/Total detectados**
k-médias	52	7	1	97,8%	86.5%
x-médias	50	5	1	97,8%	90%
SOM	51	5	0	100%	90.2%

A maior quantidade de eventos falso-positivo obtido pelo método k-médias é explicada por uma grande quantidade de consultas de protocolo DNS, tráfego HTTP e também o grande número de IPs de destinos acessados. A quantidade de consultas e endereços IPs acessados foram considerados anômalos devido a discrepância dos seus valores quando comparados com os valores do conjunto de dados considerados normais. Dentre os 7 falso-positivo, apenas um foi detectado e constatado como sendo utilização do Skype, serviço que quando inicializado tem como característica o acesso a diversos endereços IPs em portas altas. O x-médias teve um comportamento parecido, sendo que foram detectados dois eventos comuns em categoria falso-positivo obtidos pelos métodos k-médias e x-médias. Contudo, dos 5 eventos falso-positivo, 4 foram de um mesmo endereço IP. Os falso-positivos obtidos pelo k-médias tiveram como característica um grande número de consultas DNS, tráfego HTTP e também a utilização do Skype foram observados.

Dentre os resultados obtidos pela rede neural SOM, dois eventos do tipo falso-positivo foram detectados por ambos métodos, k-médias e x-médias. Dos três restantes, um foi detectado pelo método k-médias, outro pelo método x-médias.

Dessa forma, apenas um falso-positivo foi detectado somente pela rede neural SOM, e que também possui as características já citadas, alto número de consultas DNS, tráfego HTTP e vários endereços distintos de destino.

Na tabela 5.2 é possível observar os resultados no primeiro dia de validação em que nenhum ataque novo foi realizado. No segundo conjunto de dados, foram aplicados 280 ataques, gerando 379 tuplas de ataques do total de 7410. O número de eventos falso-positivo obtido pelo método k-médias pode ser divido em basicamente em três grupos: alguns foram detectados como anômalo devido a utilização do Skype, outros devido a grande quantidade de consultas DNS e tráfego HTTP e outros ainda devido a grande quantidade de pacotes enviados. Todos os eventos falso-positivo obtidos pelo método x-médias foram detectados também pelo k-médias.

Tabela 5.2 Dados obtidos com os métodos implementados – dia validação 1.

Método	**Total de eventos detectados**	**Falso-positivo**	**Falso-negativo**	**Ataques detectados /Total ataques**	**Ataques detectados/Total detectados**
k-médias	397	21	3	99.2%	94.7%
x-médias	386	13	6	98.4%	96.6%
SOM	386	9	2	99.4%	97.7%

Diferentemente do x-médias e k-médias, a rede neural SOM detectou alguns IPs que tiveram comportamento suspeito. Dentre 9 eventos falso-positivos, 4 foram detectados tanto pelo método k-médias quanto pelo método x-médias e 5 foram detectados somente pela rede neural SOM, todos com comportamento anômalo. Os 5 eventos falso-positivos, detectados somente pela rede neural SOM, com comportamento suspeito tiveram como característica a utilização de várias portas de destino em vários endereços IPs inválidos da rede 192.168.1.0/24.

Na tabela 5.3 é possível observar os resultados no segundo dia de validação em que foram realizados ataques novos. No terceiro conjunto de dados, foram aplicados 30 ataques, gerando 53 tuplas de ataques do total de 7972.

Tabela 5.3 Dados obtidos com os métodos implementados – dia validação 2.

Método	**Total de eventos detectados**	**Falso-positivo**	**Falso-negativo**	**Ataques detectados /Total ataques**	**Ataques detectados/Total detectados**
k-médias	63	17	8	85.2%	73%
x-médias	53	7	8	85.2%	86.8%
SOM	56	3	0	100%	94.6%

O número de eventos falso-positivo obtido pelo método k-médias foi, principalmente, devido à grande quantidade de consultas DNS e tráfego HTTP e também a grande quantidade de pacotes enviados. Todos os eventos falso-positivo obtidos pelo método x-médias foram detectados também pelo k-médias.

Dentre os resultados obtidos pela rede neural SOM, os dois eventos do tipo falso-positivo foram detectados também pelo método k-médias.

A quantidade de tuplas que cada tipo de ataque gerou em cada dia pode ser visualizada na tabela 5.4. Nas tabelas 5.5, 5.6 e 5.7 podem ser visualizadas as porcentagens de acerto e quantidade de tuplas que foram corretamente detectadas nos dias de treinamento, validação 1 e validação 2, respectivamente.

Tabela 5.4 Quantidade e tipos de ataques aplicados por dia.

Dia	**Força Bruta**	**Varredura**	**DoS**	**Vulnerabilidades WEB**	**Total**
Treinamento	14	22	10	-	46
Validação 1	161	148	70	-	379
Validação 2	16	12	5	20	53

Tabela 5.5 Quantidade de acerto por tipo de ataque – dia de treinamento.

Método	**Força Bruta**	**Varredura**	**DoS**	**Total**
k-médias	100% (14)	95.4% (21)	100% (10)	97.8% (45)
x-médias	100% (14)	95.4% (21)	100% (10)	97.8% (45)
SOM	100% (14)	100% (22)	100% (10)	100% (46)

Tabela 5.6 Quantidade de acerto por tipo de ataque – dia de validação 1.

Método	Força Bruta	Varredura	DoS	Total
k-médias	97.4% (159)	99.3% (147)	100% (70)	99.2% (376)
x-médias	97.4% (159)	97.3% (144)	100% (70)	98.4% (373)
SOM	100% (161)	98.6% (146)	100% (70)	99.5% (377)

Tabela 5.7 Quantidade de acerto por tipo de ataque – dia de validação 2.

Método	Força Bruta	Varredura	DoS	Vulnerabilidade WEB	Total
k-médias	100% (16)	100% (12)	100% (5)	65% (13)	84.9% (46)
x-médias	100% (16)	100% (12)	100% (5)	65% (13)	84.9% (46)
SOM	100% (16)	100% (12)	100% (5)	100% (20)	100%(53)

5.2 Resultados k-médias

No método k-médias, diferente do método x-médias, o número de clusters é fixo, definido pelo usuário. Dessa forma, para os testes foram utilizados 5 clusters, definidos empiricamente. A distribuição dos dados entre os clusters do método k-médias do dia de treinamento pode ser visualizada na figura 5.1. Dentre os 5 clusters do k-médias, grande parte dos dados do dia de treinamento concentraram-se em dois principais clusters, 2 e 4. O restante foi distribuído entre os clusters 1, 3 e 5 com 113, 4 e 28 elementos, respectivamente.

Na figura 5.2 podem ser visualizados como ficaram dispostos os dados considerados anômalos entre os clusters, bem como os dados que eram ataques e as taxas de eventos falso-positivo e falso-negativo para o dia de treinamento. Analisando os dados que foram classificados como anômalos é possível observar que concentraram-se em apenas 3 clusters, 1, 2 e 5. Na figura 5.3 é possível observar a distribuição dos tipos de ataques detectados em cada cluster.

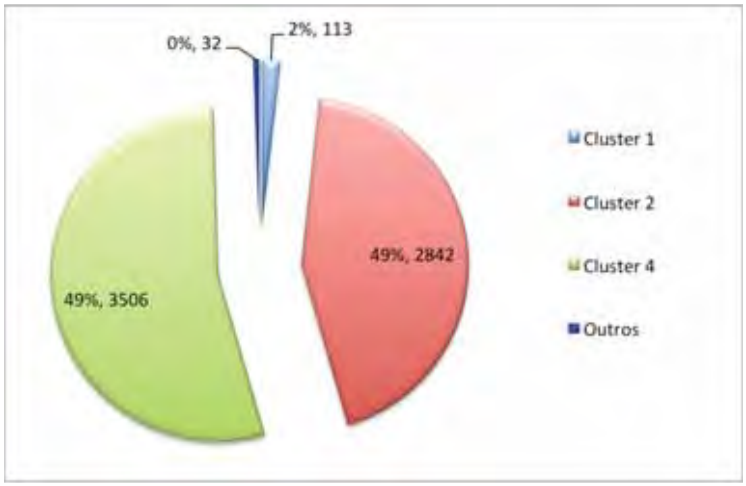


Figura 5.1 Distribuição dos dados entre os clusters do método k-médias – dia treinamento.

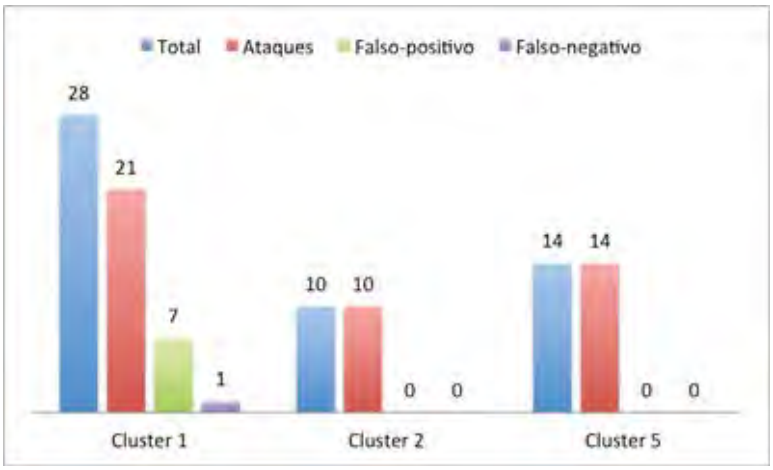


Figura 5.2 Distribuição dos dados considerados anômalos entre os clusters do método k-médias – dia treinamento.

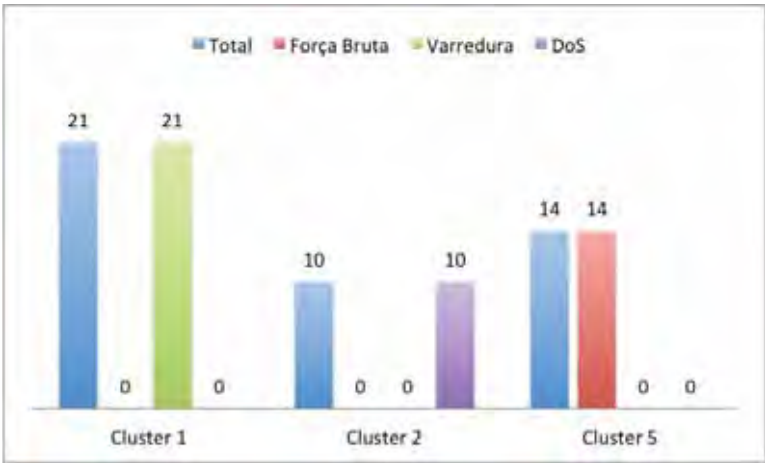


Figura 5.3 Distribuição dos ataques detectados entre os clusters do método k-médias – dia treinamento.

Na figura 5.4 é ilustrada a distribuição dos dados entre os clusters para o dia de validação. A grande parte dos dados ficaram concentradas em 2 clusters, assim como no dia de treinamento. O restante foi distribuído entre os cluster 1, 3 e 5 com 300, 5 e 181 elementos, respectivamente. As taxas de eventos falso-positivo, falso-negativo, quantidade de ataques detectados por cluster podem ser visualizadas na figura 5.5. Já os tipos de ataques concentrados em cada cluster podem ser visualizados na figura 5.6.

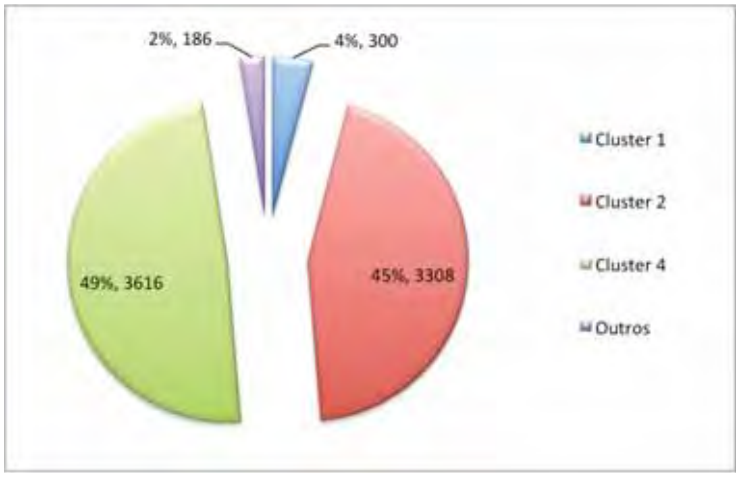


Figura 5.4 Distribuição dos dados entre os clusters do método k-médias – dia validação 1.

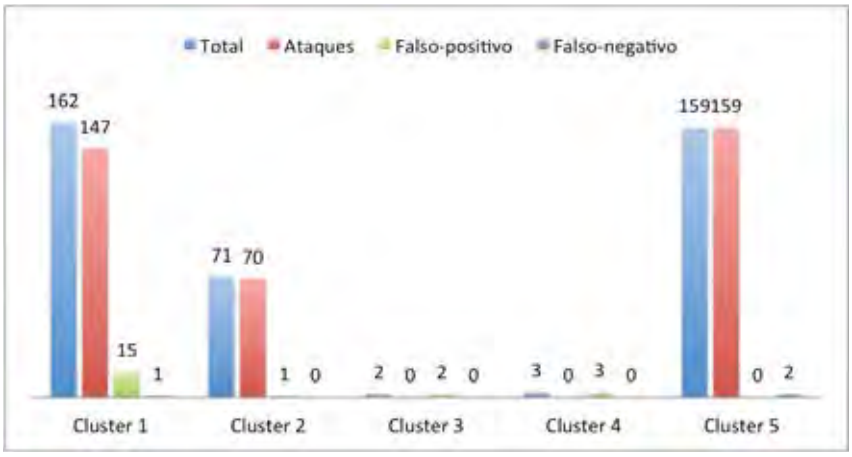


Figura 5.5 Distribuição dos dados considerados anômalos entre os clusters do método k-médias – dia validação 1.

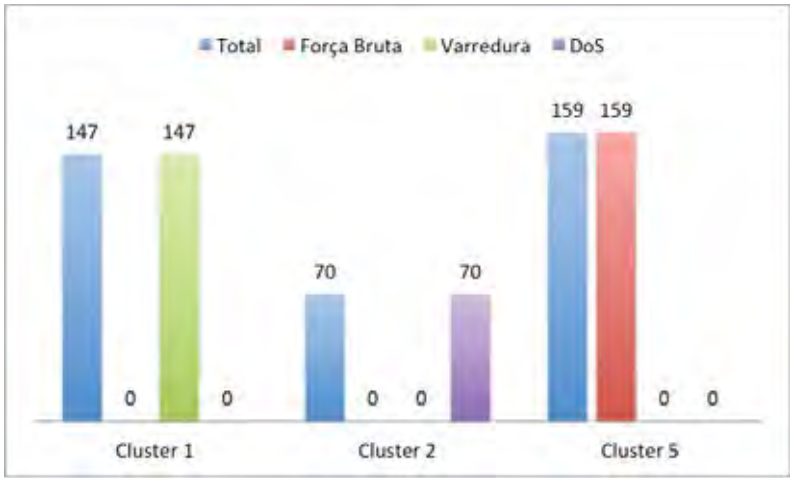


Figura 5.6 Distribuição dos ataques detectados entre os clusters do método k-médias – dia validação 1.

Para ambos os dias de teste, treinamento e o primeiro dia de validação, é possível observar um comportamento semelhante na distribuição dos dados. Além disso, nota-se um comportamento peculiar na distribuição dos tipos de ataques, no qual o mesmo tipo de ataque ficou concentrado em um mesmo cluster, em ambos os dias.

Na figura 5.7 é ilustrada a distribuição dos dados entre os clusters para o segundo dia de validação. Grande parte dos dados ficaram concentradas em 2 clusters, assim como no dia de treinamento e o primeiro dia de validação. O restante foi distribuído entre os cluster 1, 3 e 5 com 218, 4 e 55 elementos, respectivamente.

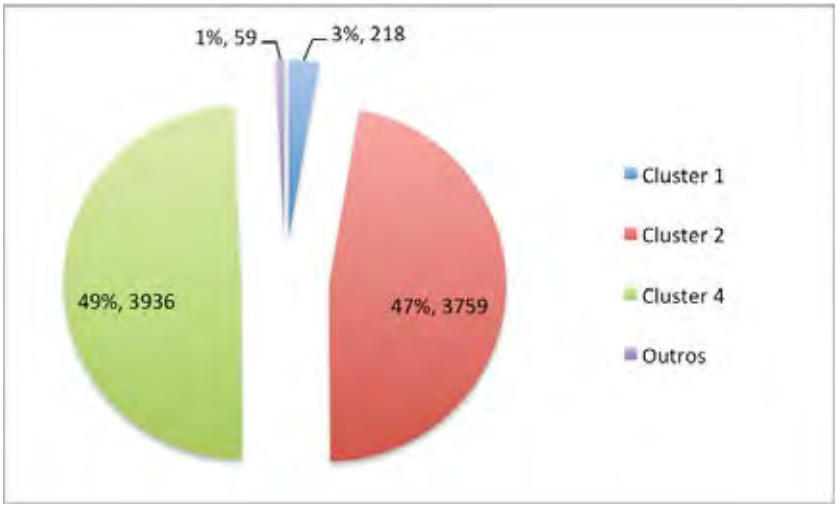


Figura 5.7 Distribuição dos dados entre os clusters do método k-médias – dia validação 2.

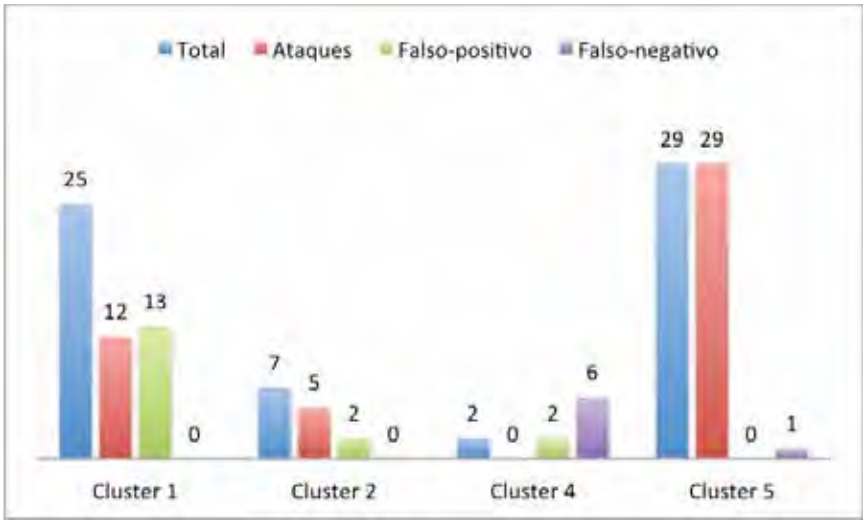


Figura 5.8 Distribuição dos dados considerados anômalos entre os clusters do método k-médias – dia validação 2.

As taxas de eventos falso-positivo, falso-negativo, quantidade de ataques detectados por cluster podem ser visualizadas na figura 5.8. Já os tipos de ataques concentrados em cada cluster podem ser visualizados na figura 5.9.

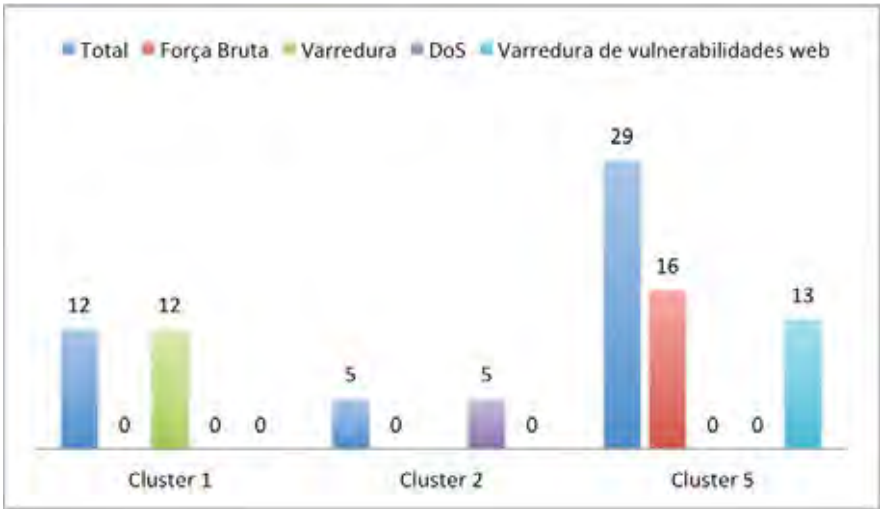


Figura 5.9 Distribuição dos dados considerados anômalos entre os clusters do método k-médias – dia validação 2.

Assim como o dia de treinamento e o primeiro dia de validação, o segundo dia de validação teve um comportamento semelhante. Contudo, devido ao novo ataque, a composição do cluster 5 foi alterada, deixando de possuir apenas os ataques de força bruta para agora também possuir ataques de varredura de vulnerabilidades web.

5.3 Resultados x-médias

Em relação aos clusters gerados pelo método x-médias, na figura 5.10 é possível verificar a distribuição dos dados entre os clusters no dia de treinamento. No método x-médias, durante o treinamento foram gerados 4 clusters. Nota-se que os dados, assim como no método k-médias, ficaram concentrados em dois clusters.

Na figura 5.11 é possível observar as taxas de eventos falso-positivo, falso-negativo, total de ataques e total de tuplas consideradas anômalas para o dia de treinamento. Já na figura 5.12 é possível visualizar a distribuição dos tipos de ataques entre os clusters. Diferente do k-médias, em um dos clusters do x-médias foi possível encontrar tanto ataques de força bruta quanto varredura.

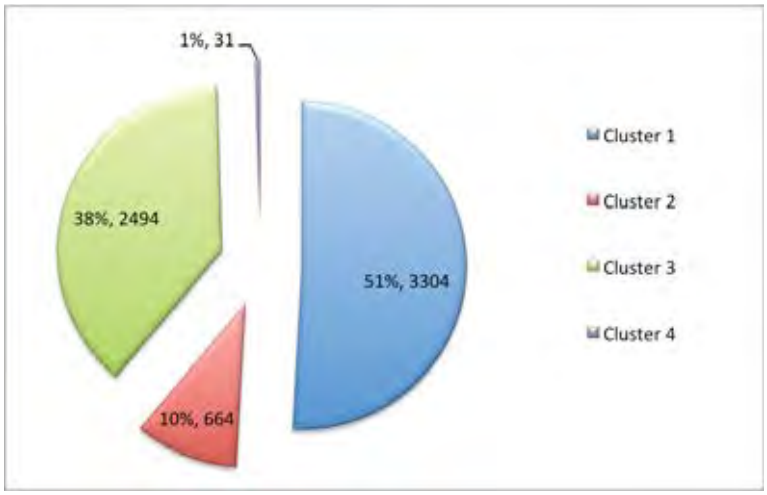


Figura 5.10 Distribuição dos dados entre os clusters do método x-médias - dia treinamento.

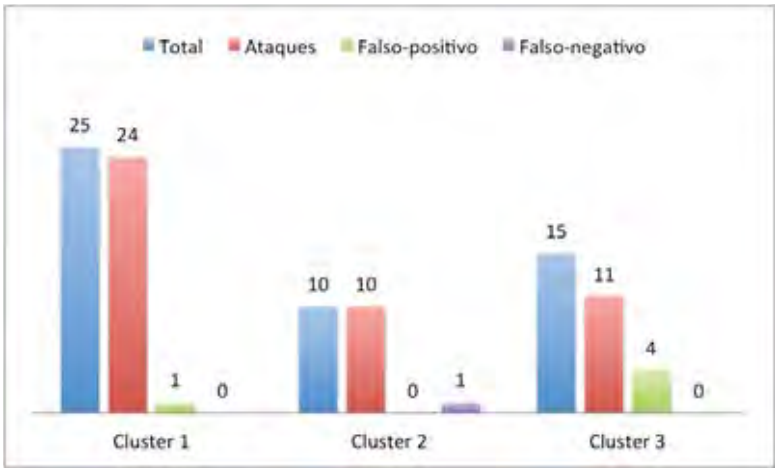


Figura 5.11 Distribuição dos dados considerados anômalos entre os clusters do método x-médias – dia treinamento.

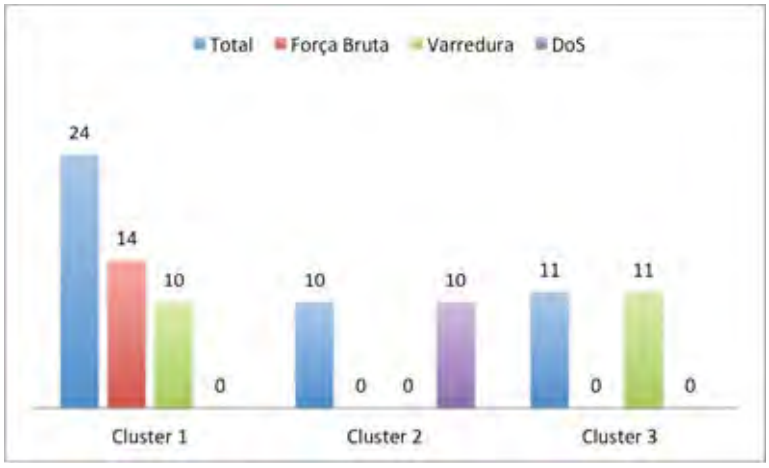


Figura 5.12 Distribuição dos ataques detectados entre os clusters do método x-médias – dia treinamento.

No primeiro dia de validação, a distribuição dos dados entre os clusters pode ser visualizada na figura 5.13. É possível observar que os dados seguiram o comportamento do dia de treinamento, concentraram-se em dois clusters. Já na figura 5.14 é possível visualizar as taxas de eventos falso-positivo, falso-negativo, quantidade de ataques e quantidade de tuplas consideradas anômalas por cluster. Na figura 5.15, nota-se que a distribuição dos ataques por cluster teve o mesmo comportamento do dia de treinamento, no qual em um cluster é possível encontrar dois tipos de ataques.

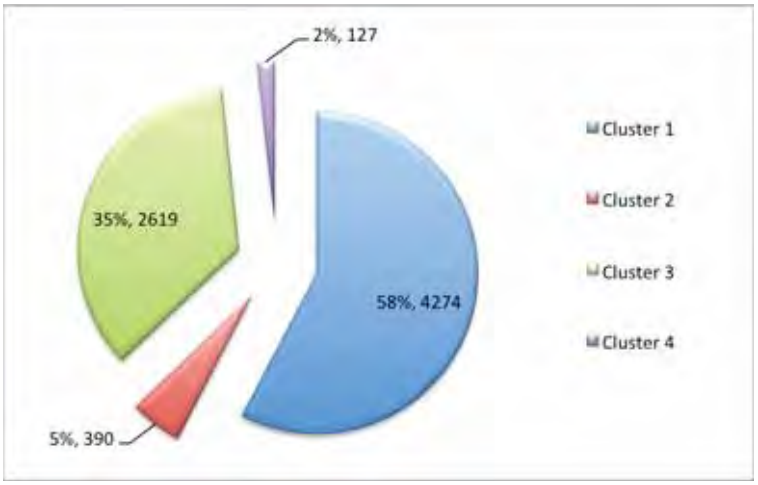


Figura 5.13 Distribuição dos dados entre os clusters do método x-médias - dia validação 1.

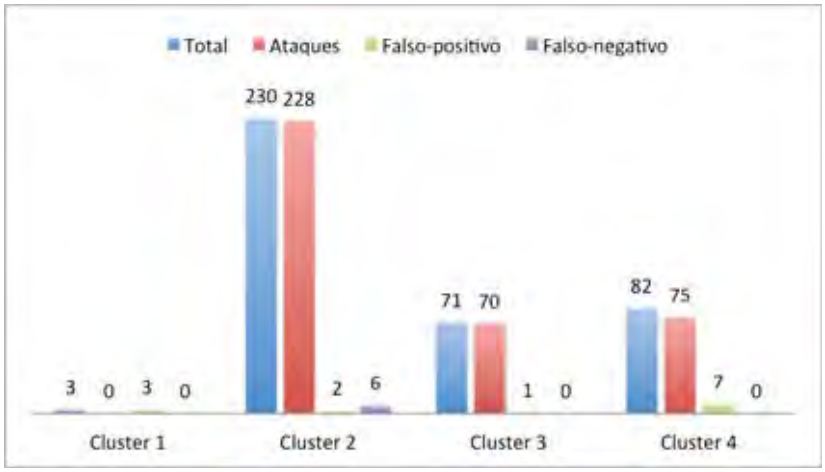


Figura 5.14 Distribuição dos dados considerados anômalos entre os clusters do método x-médias – dia validação 1.

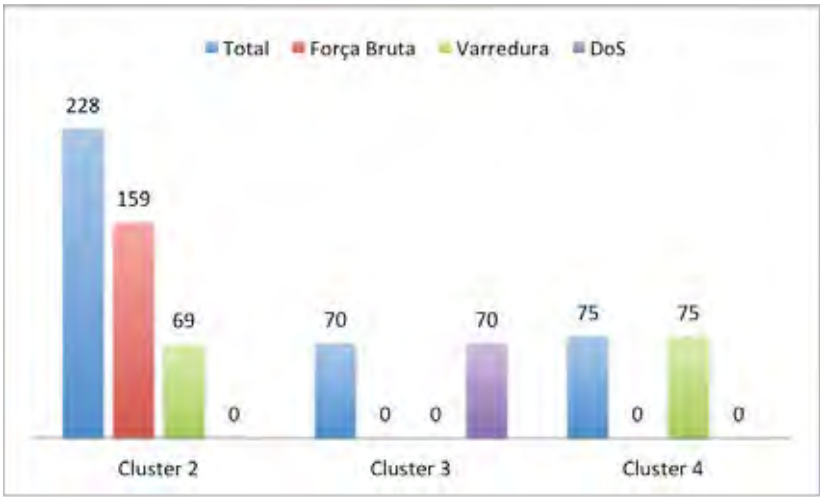


Figura 5.15 Distribuição dos ataques detectados entre os clusters do método x-médias – dia validação 1.

No segundo dia de validação, a distribuição dos dados entre os clusters pode ser visualizada na figura 5.16. É possível observar que os dados seguiram o comportamento do dia de treinamento, concentraram-se em dois clusters. Já na figura 5.17 é possível visualizar as taxas de eventos falso-positivo, falso-negativo, quantidade de ataques e quantidade de tuplas consideradas anômalas por cluster. Na figura 5.18, nota-se que a distribuição dos ataques por cluster teve um alteração em relação ao dia de treinamento e ao primeiro dia de validação, devido ao novo ataque (varredura de vulnerabilidades em páginas WEB), no qual em um cluster, que antes era composto por dois tipos de ataques, agora é possível encontrar três tipos de ataques.

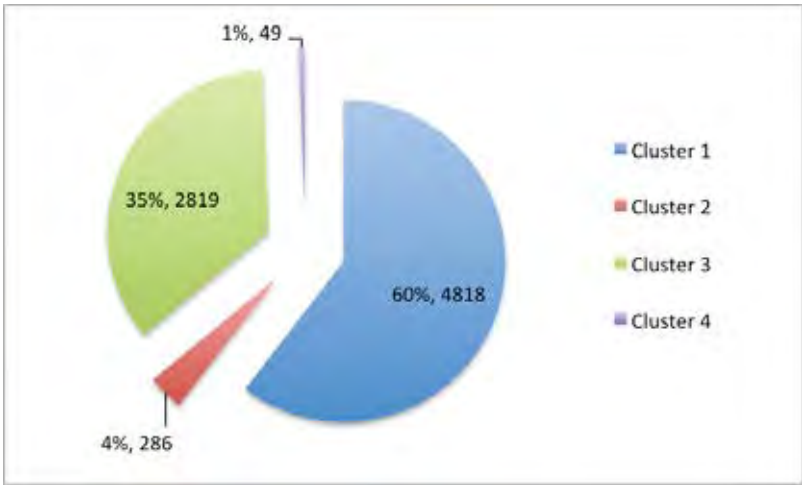


Figura 5.16 Distribuição dos dados entre os clusters do método x-médias - dia validação 2.

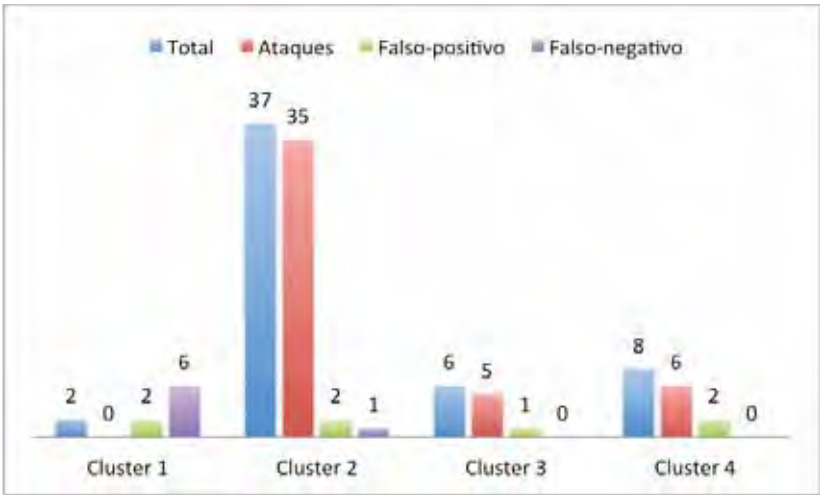


Figura 5.17 Distribuição dos dados considerados anômalos entre os clusters do método x-médias – dia validação 2.

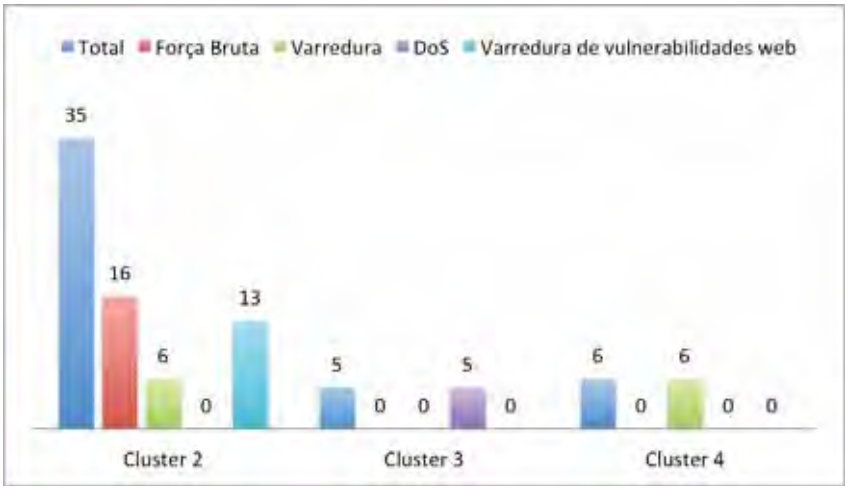


Figura 5.18 Distribuição dos ataques detectados entre os clusters do método x-médias – dia validação 2.

5.4 Resultados SOM

Em relação a rede neural SOM, foi utilizado um mapa com dimensão 5x5, ou seja, 25 neurônios na composição da rede, porém somente 6 neurônios foram ativados após o processamento dos dados utilizados para treinamento. Na figura 5.19 é possível verificar a distribuição dos dados entre os neurônios ativados, também chamados de clusters. Nota-se que, diferente dos métodos k-médias e x-médias, a grande parte dos dados ficou concentrado em apenas 1 cluster. O restante foi distribuído entre os cluster 3, 4, 5, 9 e 10 com 7, 11, 7, 16 e 96 elementos, respectivamente.

Na figura 5.20 é possível observar as taxas de eventos do tipo falso-positivo, falso-negativo, total de ataques e total de tuplas consideradas anômalas para o dia de treinamento. Já na figura 5.21 é possível visualizar a distribuição dos tipos de ataques entre os clusters. Diferente do k-médias e do x-médias, na rede neural SOM alguns ataques ficaram em clusters isolados, ou seja, os elementos de determinados clusters foram só ataques.

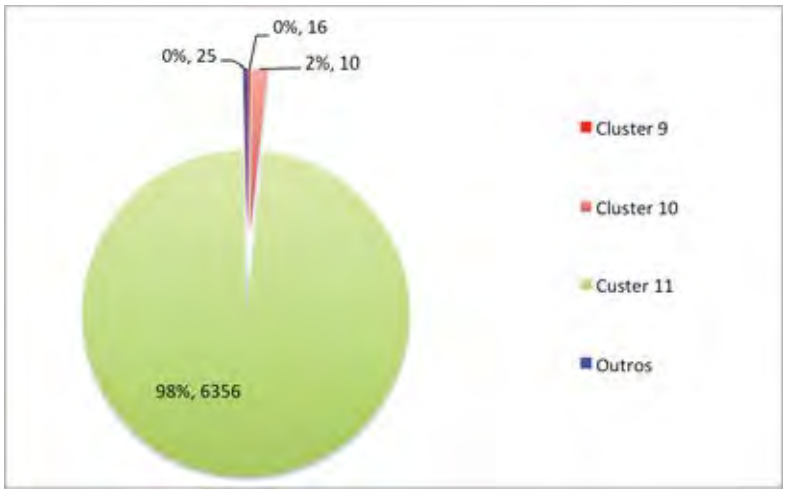


Figura 5.19 Distribuição dos dados entre neurônios da rede neural SOM – dia treinamento.

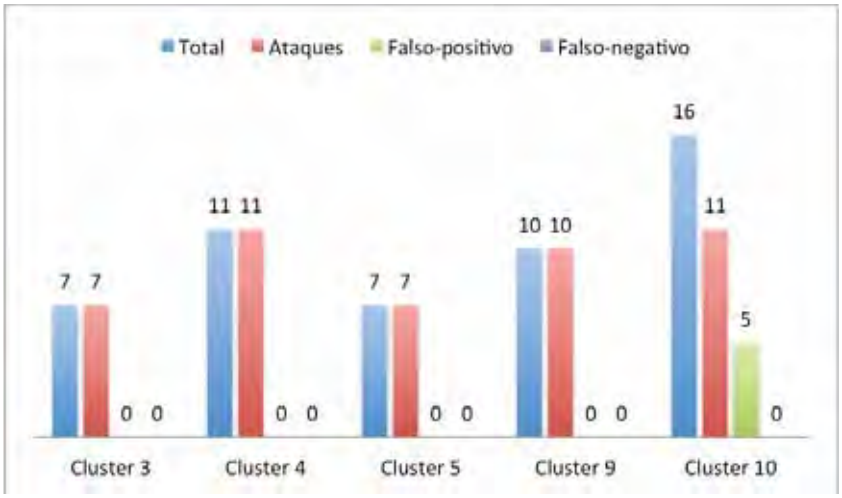


Figura 5.20 Distribuição dos dados considerados anômalos entre neurônios da rede neural SOM – dia treinamento.

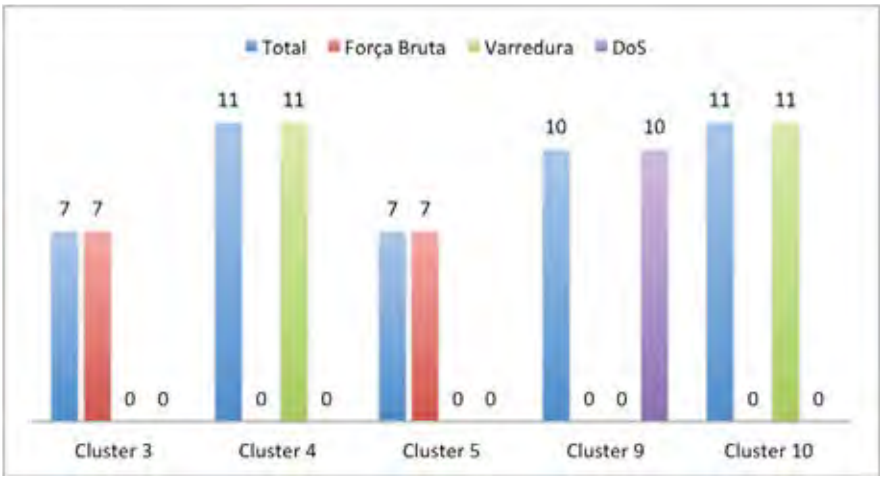


Figura 5.21 Distribuição dos ataques detectados entre os clusters da rede neural SOM – dia treinamento.

No primeiro dia de validação, a distribuição dos dados entre os clusters pode ser visualizada na figura 5.22. É possível observar que os dados seguiram o comportamento do dia de treinamento: a maioria dos dados concentram-se em um único cluster. Os clusters 3, 4, 5, 9 e 10 ativados também no conjunto de treinamento tiveram 65, 75, 91, 76 e 13 elementos, respectivamente. Entretanto, alguns clusters que não haviam sido ativados no conjunto de treinamento, foram ativados no conjunto de validação. Dessa forma, todos os elementos dos novos clusters ativados são considerados anômalos. Os novos clusters ativados foram o 1, 2, e 7 com 5, 73 e 3 elementos respectivamente.

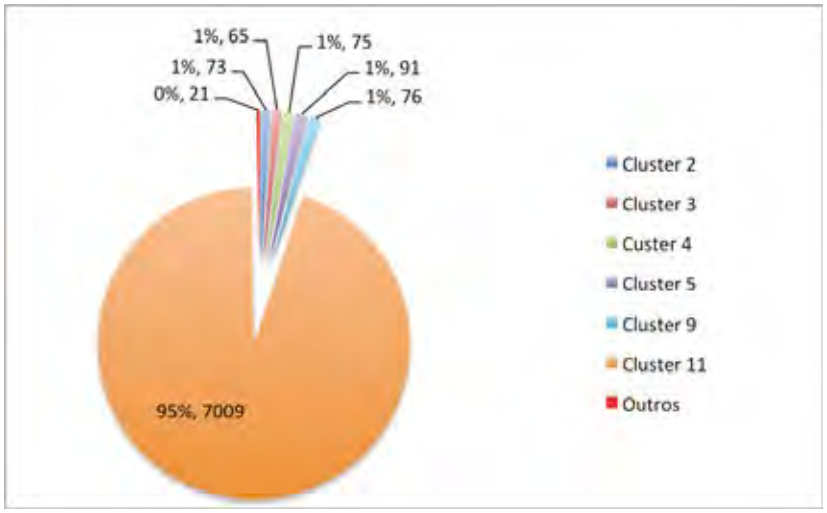


Figura 5.22 Distribuição dos dados entre neurônios da rede neural SOM – dia validação 1.

Na figura 5.23 é possível visualizar as taxas de eventos falso-positivo, falso-negativo, quantidade de ataques e quantidade de tuplas consideradas anômalas por cluster. Na figura 5.24 é possível observar que a distribuição dos ataques por cluster teve o mesmo comportamento do dia de treinamento, no qual em determinados clusters todos os elementos são praticamente composto só de ataques. Diferente dos métodos k-médias e x-médias, a rede neural SOM detectou 5 tuplas que tiveram comportamento anômalo, acessando IPs de uma rede de endereçamento privado em diversas portas. Três dessas cinco tuplas ativaram o cluster 7 e duas foram detectados no cluster 2, no gráfico exibidos como falso-positivo pois não faziam parte dos ataques aplicados pelo computador de testes.

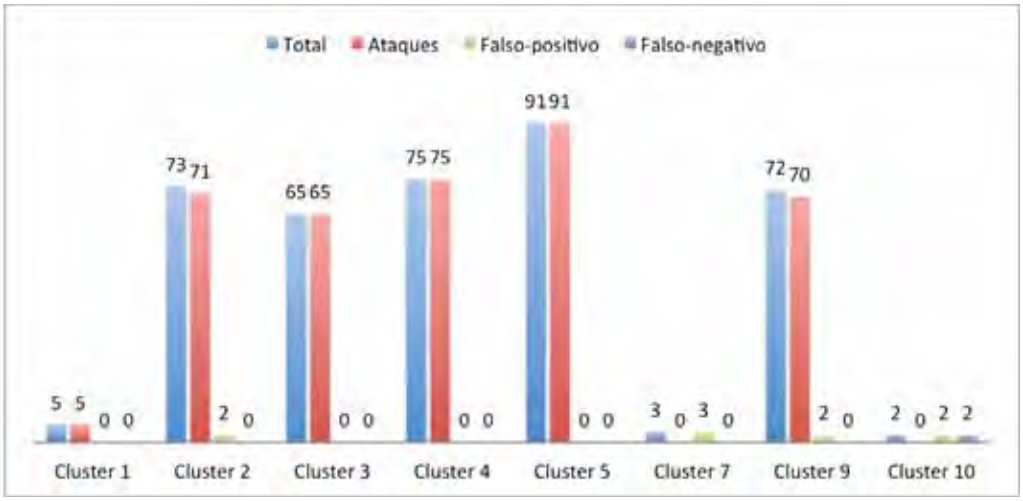


Figura 5.23 Distribuição dos dados considerados anômalos entre neurônios da rede neural SOM – dia validação 1.

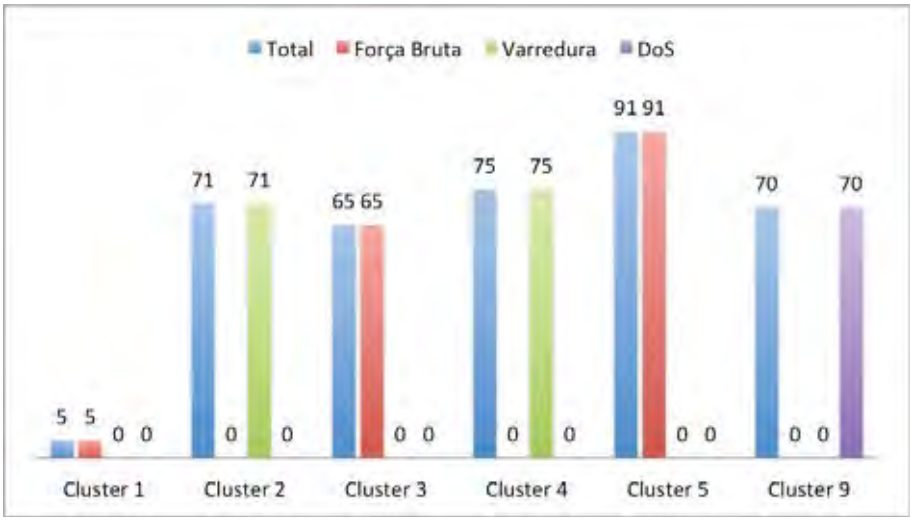


Figura 5.24 Distribuição dos ataques detectados entre os clusters da rede neural SOM – dia validação 1.

No segundo dia de validação, a distribuição dos dados entre os clusters pode ser visualizada na figura 5.25. É possível observar que os dados seguiram o comportamento do dia de treinamento e do primeiro dia de validação, no qual a maioria dos dados concentram-se em um único cluster. Os outros clusters ativados foram 1, 2, 3, 4, 5, 9 e 10 e tiveram 8, 2, 21, 6, 8, 15 e 13 elementos, respectivamente. Como no primeiro dia de validação, alguns clusters que não haviam sido ativados no conjunto de treinamento, foram ativados no segundo conjunto de validação.

Na figura 5.26 é possível visualizar as taxas de eventos falso-positivo, falso-negativo, quantidade de ataques e quantidade de tuplas consideradas anômalas por cluster.

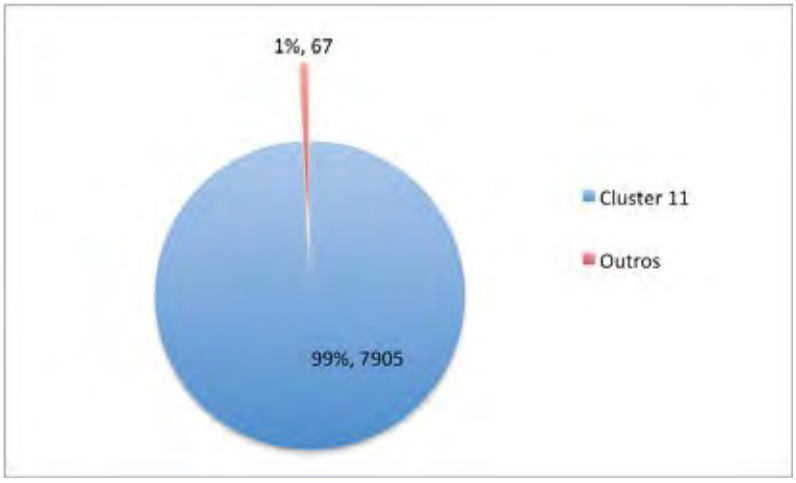


Figura 5.25 Distribuição dos dados entre neurônios da rede neural SOM – dia validação 2.

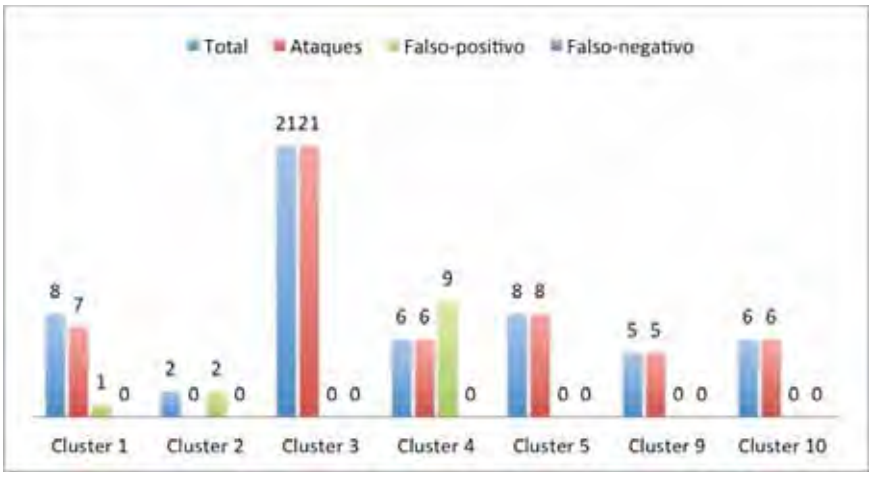


Figura 5.26 Distribuição dos dados considerados anômalos entre neurônios da rede neural SOM – dia validação 2.

Na figura 5.27 é possível observar que a distribuição dos ataques por cluster teve o mesmo comportamento do dia de treinamento e do primeiro dia de validação, no qual todos os elementos de determinados clusters são classificados como ataques.

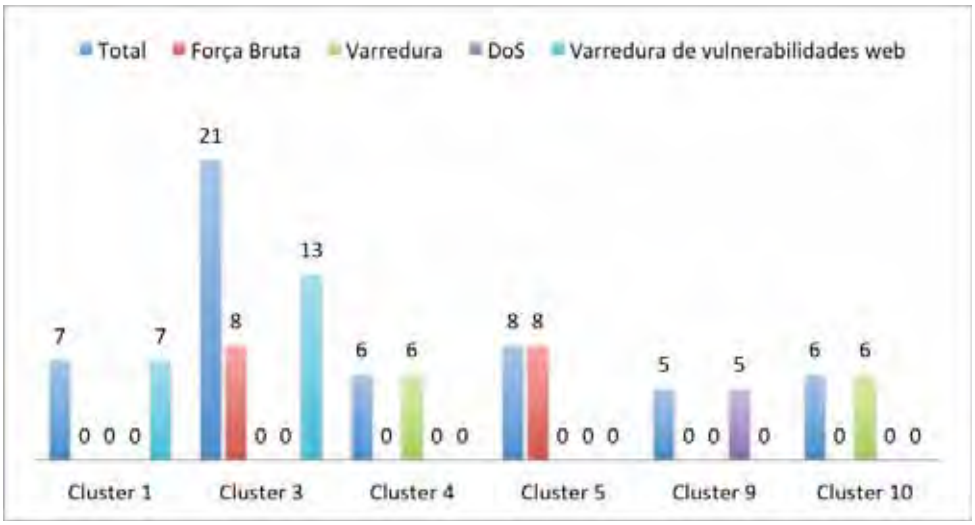


Figura 5.27 Distribuição dos ataques detectados entre os clusters da rede neural SOM – dia validação 2.

5.5 Considerações finais

Neste capítulo foram apresentados os resultados obtidos com os métodos implementados: k-médias, x-médias e a rede neural SOM. Cada método apresentou um comportamento diferente, contudo, foi possível observar uma maior diferença no

comportamento da rede neural SOM. Em relação a taxas de acertos, apesar do método k-médias obter uma taxa maior de acerto do que o x-médias, ela também obteve uma maior taxa de eventos falso-positivo. Foi possível observar que entre os três métodos, a rede neural SOM obteve uma maior taxa de acertos e menor taxa de falso-positivo e falso-negativo. Todos os métodos conseguiram detectar a presença de um novo ataque, contudo, a rede neural SOM conseguiu detectar também tuplas nas quais o comportamento foi suspeito no primeiro dia de validação. Além disso, os falso-positivos obtidos em todos e métodos e dias foram explicados pelo discrepância de uma ou mais variáveis quando comparados com os valores do conjunto de dados considerados normais. No próximo capítulo são feitas as conclusões sobre o trabalho.

Capítulo 6 – Conclusões e trabalhos futuros

O monitoramento de redes é extremamente importante, pois possibilita verificar a qualidade dos serviços oferecidos, garantir a segurança da informação e controlar o tráfego de acordo com o estabelecido pela política de uso da rede. Atualmente, as redes estão cada vez maiores devido a sua grande utilização, e o monitoramento desses ambientes é uma tarefa cada vez mais complexa devido à grande quantidade de tráfego gerado, o que implica em uma grande quantidade de informação a ser analisada. Na busca de escalabilidade e da redução da carga computacional exigida para análise, os fluxos foram escolhidos como fonte de informações. Quando coletados de um dispositivo de rede permitem uma visão completa de todas as sessões e conexões do ambiente, sem a análise de conteúdo dos pacotes, não inserindo, portanto, nenhum tipo de latência devido ao desencapsulamento (CORREA, 2009). Dessa forma, fluxos de rede é uma opção atrativa já que permite que o tráfego de um ambiente de grande porte seja analisado de forma eficiente e escalável.

O trabalho desenvolvido tem como objetivo a detecção de eventos de redes de computadores, incluindo ataques que não são conhecidos previamente, por meio de análises de fluxos bidirecionais. Para tanto, são utilizados métodos não-supervisionados, como redes neurais e modelos estatísticos. O projeto visa obter o máximo de desempenho na detecção de eventos sem afetar o desempenho de uma rede de computadores, visto que análise dos dados ocorre de modo isolado ao funcionamento da rede, sob uma base de dados. Além disso, a utilização de um

protocolo de exportação baseado no *Biflow* auxiliou ainda mais no que diz respeito ao armazenamento dos dados da rede e na sua análise.

Em relação a escolha do local de análise de dados, é importante ressaltar que a exportação é feita por um dispositivo ativo na rede, geralmente um concentrador de tráfego, e a análise é feita em um dispositivo separado, dedicado apenas para a análise. Esta arquitetura tem como objetivo a manutenção do desempenho e escalabilidade da rede, não sobrecarregando o dispositivo exportador e não comprometendo a segurança nas análises de fluxos de redes, uma vez que caso o dispositivo analisador seja o próprio dispositivo exportador e membro ativo na rede, as informações, em caso de um ataque ao dispositivo exportador, podem ser subvertidas.

Os resultados mostraram que o sistema é eficiente pois detectou aproximadamente todos os ataques aplicados no ambiente. Mesmo com o aumento do número de ataques ao ambiente, as taxas de eventos falso-positivo e falso-negativo continuaram relativamente baixas. As tuplas classificadas como eventos falso-positivo, em todos os métodos, tiveram de fato uma ou mais variáveis com valores elevados, comparados com os valores de dados considerados normais, o que demonstra uma anormalidade detectada pelos métodos.

Em relação aos três métodos implementados, foi possível observar que o comportamento da rede neural SOM diferiu dos métodos k-médias e x-médias, e também obteve um resultado melhor, já que a taxa de eventos falso-positivo e falso-negativo foram menores. Além disso, os três métodos foram capazes de detectar ataques sem conhecimento prévio durante a fase de treinamento.

Como resultados iniciais deste projeto, o artigo intitulado “Detecção de eventos em redes de computadores utilizando detecção de novidade” (BATISTA; CANSIAN, 2011) foi publicado nos anais do evento Conferência IADIS Ibero-Americana WWW/Internet, realizado nos dias 5, 6 e 7 de novembro de 2011 na cidade do Rio de Janeiro, Brasil.

6.1 Dificuldades encontradas

Durante o desenvolvimento deste trabalho algumas dificuldades foram enfrentadas durante o período. Uma das dificuldades encontradas foi a definição de quais dados são relevantes para a detecção de anomalias. Para tanto, foi necessário um estudo das informações disponíveis pelo protocolo *biflow* que são relevantes para análise.

Outra grande dificuldade encontrada foi no desenvolvimento do programa no que coleta e agrupa as informações providas pelo protocolo *biflow*. Foi necessária uma otimização nas consultas no banco de dados para que o programa tivesse um bm desempenho e os resultados fossem obtidos em tempo hábil para análise. Além disso, outras dificuldades foram encontradas em como seria feita a normalização dos dados que são utilizados pelos métodos implementados, uma vez que sem a normalização as taxas de eventos falso-positivo e falso-negativo aumentam significativamente. Por fim, outra dificuldade encontrada foi na definição dos parâmetros utilizados na rede neural SOM, que foram definidos empiricamente, por meio de testes.

6.2 Trabalhos futuros

Os testes mostraram que os métodos são eficientes, no entanto, ainda é possível melhorar os resultados para maximizar o número de acertos e diminuir a taxa de eventos falso-positivo. Para isso, como trabalho futuro são propostos:

- Normalização dos dados: consiste na adequação dos dados para a diminuição da taxa de eventos falso-positivo e falso-negativo;
- Teste em ambientes de grande porte: aplicar os métodos em ambientes maiores para verificar o desempenho da metodologia em ambientes com grande quantidade de dispositivos de rede;
- Servidores: utilizar a mesma metodologia para monitoramento de servidores, adequando os parâmetros de treinamento para as características de tráfego;
- Ajuste da distância na detecção de anomalias: para melhorar a detecção de eventos, é necessário ajustes da distância para cada método implementado;

- Implementar módulo de contramedidas: desenvolver um módulo capaz de responder de forma ativa e automática aos eventos detectados pelos três métodos apresentados.

Referências Bibliográficas

ALBERTINI, M. K.; MELLO, R. F. **A self-organizing neural network for detecting novelties.** SAC ’07: Proceedings of the 2007 ACM symposium on Applied computing. Nova York, NY, EUA 2007. DOI=10.1145/1244002.1244110

BALESTRASSI, P. P. **Identificação de Padrões em Gráficos em Gráficos de Controle Estatísticos de Processos, em Tempo Real, Utilizando Séries Temporais e Redes Neurais Artificiais.** Tese (Doutorado em Engenharia de Produção): Universidade Federal de Santa Catarina: 217 p. 2000.

BATISTA, M. L. ; CANSIAN, A. M. . **Detecção de eventos em redes de computadores utilizando detecção de novidade**. Conferência IADIS Ibero-Americana WWW/Internet 2011. Rio de Janeiro – RJ 2011.

CARPENTER, G. A.; GROSSBERG, S. 1998. **Adaptive Resonance Theory**. Michael A. Arbib (Ed.), The Handbook of Brain Theory and Neural Networks, Segunda Edição, 2003, pp. 87-90.

CARVALHO, S.; CAMPOS, W. **Estatística Básica Simplificada.** Elsevier, 2008. 608p.

CERT.BR. **Cartilha de Segurança para Internet – Parte II: Riscos envolvidos no uso da Internet e métodos de prevenção.** 2006. Disponível em: <http://cartilha.cert.br/prevencao/>. Acesso em: 9 de Nov. 2010.

CERT.BR. **Centro de Estudos, Reposta e Tratamento de Incidentes de Segurança no Brasil.** 2011. Disponível em: <http://www.cert.br>. Acesso em: 17 de Jun. 2011.

CLAISE, B. **RFC 3954: Cisco Systems NetFlow Services Export Version 9.** 2004. Disponível em: < http://www.ietf.org/rfc/rfc3954.txt >. Acesso em: 10 Nov. 2010.

CORRÊA, J. L. **Um modelo de detecção de eventos em redes baseado no rastreamento de fluxos.** Dissertação (Mestrado em Ciências de Computação): Instituto de Biociências, Letras e Ciências Exatas – IBILCE-UNESP: 111 p. 2009.

CORRÊA, J. L.; PROTO, A.; CANSIAN, A. M. **Modelo de armazenamento de fluxos de rede para análises de tráfego e de segurança.** VIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg). Gramado - RS 2008.

CORRÊA, J. L.; PROTO, A; ALEXANDRE, L. A.; CANSIAN, A. M. **Detectando eventos em redes utilizando um modelo de rastreamento de fluxos baseado em assinaturas.** IX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais. Campinas - SP 2009.

DASGUPTA, D.; FORREST, S. **Novelty Detection in Time Series Data using Ideas from Immunology.** In Proceedings of The International Conference on Intelligent Systems. 1995.

ELKAN, C. **Results of the KDD'99 classifier learning.** ACM SIGKDD Explorations. 2000.

FLOW-TOOLS. **Flow-tools information.** Disponível em: < http://www.splintered.net/sw/flow-tools/>. Acesso em: 13 de Novembro de 2010.

FPROBE. **Fprobe**. Disponível em: <http://sourceforge.net/projects/fprobe/>. Acesso em: 04 de Agosto de 2011.

GOGOI, P; BORAH, B.; BHATTACHARYYA, D. K. **Anomaly Detection Analysis of Intrusion Data using Supervised & Unsupervised Approach.** Journal of Convergence Information Technology. 2010.

GOLLMANN, D. **Computer Security.** John Wiley & sons ltda., 1999. ISBN 0471978442.

GONÇALVES, M. L.; ANDRADE NETTO, M. L.; ZULLO JÚNIOR, J. **Um sistema neural modular para classificação de imagens utilizando Mapas de Kohonen.** VIII Simpósio Brasileiro de Sensoriamento Remoto. Salvador - BA 1996.

HAYKIN, S. **Neural Networks: A Comprehensive Foundation.** 2ª Edição. Prentice Hall 1998. ISBN: 9780132733502.

HSIAO, H.; CHEN, D.; WU, T. J. **Detecting Hiding Malicious Website Using Network Traffic Mining Approach.** 2nd International Conference on Education Technology and Computer (ICETC). Shanghai, China 2010.

LUDWIG JUNIOR, O.; MONTGNOMERY, E. **Redes Neurais: Fundamentos e Aplicações com Programas em C**. Ciência Moderna, 2007, 136p.

MEDEIROS, J. P. **Estudos e Implementação de Algoritmos Inteligentes para a Detecção e Classificação de Falhas na Medição de Gás Natural.** Dissertação (Mestrado em Ciência e Engenharia de Petróleo): Universidade Federal do Rio Grande do Norte: 77 p. 2009.

MORETTIN, P.A.; BUSSAB, W. O. **Estatística Básica.** 5° Edição. Saraiva, 2003. 526p.

MARKOU, M.; SINGH, S. **Novelty Detection: a review – part 1: neural network based approaches.** 2003. Signal Processing.

MARSLAND, S.; SHAPIRO, J.; NEHMZOW, U. **A self-organizing network that grows when required. Neural Networks.** Neural Network, Oxford, UK, 2002. DOI: 10.1016/S0893-6080(02)00078-3

NAGANO, M. S.; BENITE, M.; SOBREIRO, V. A. **Aplicação de Redes Neurais Artificiais para Classificação de Países e Exploração de Dados Macroeconômicos.** E & G. Economia e Gestão, v. 7, p. 101-120, 2007.

NALDI, M. C. **Técnicas de Combinação para o Agrupamento Centralizado e Distribuído de Dados.** Tese (Doutorado em Ciências de Computação e Matemática Computacional): Instituto de Ciências Matemáticas e de Computação – ICMC-USP: 276 p. 2011.

NETKIT. **NetKit**. Disponível em:< http://wiki.netkit.org/index.php/Main_Page>. Acesso em: 04 de Agosto de 2011.

PELLEG, D; MOORE, A. W. **X-means: Extending K-means with Efficient Estimation of the Number of Clusters**. Proceedings of the Seventeenth International Conference on Machine Learning (ICML '00), Pat Langley (Ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 727-734.

PENG, B.; GUO, W.; LIU, D.; FU, J. **Dynamic application flow cluster based on traffic behavior distance.** Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference. Chengdu, China, 2010. DOI: 10.1109/ICACTE.2010.5579013

PROTO, A.; ALEXANDRE, L. A.; CANSIAN, A. M. **Um modelo estatístico aplicado a fluxo de dados para detecção de eventos em redes.** 8th International Information and Telecommunication Technologies Symposium. Florianópolis – SC 2009.

QUITTEK, J.; ZSEBY, T; CLAISE, B.; ZANDER, S. **RFC 3917: Requirements for IP Flow Information Export: IPFIX.** 2004. Disponível em: < http://www.ietf.org/rfc/rfc3917.txt >. Acesso em: 09 Novembro de 2010.

REHMAN, R. U. **Intrusion detection systems with Snort: advanced IDS techniques using Snort, Apache, MySQL, PHP, and ACID.** 1ª Edição. Prentice Hall Ptr 2003. ISBN 9780131407336.

SILVESTRE, L. A. **Análise de Dados e Estatística Descritiva.** Escolar Editora, 2007. 352p.

SMITH, R.; JAPKOWICZ, N.; DONDO, M.; MASON, P. **Using unsupervised learning for network alert correlation.** Proceedings of the Canadian Society for computational studies of intelligence, 21st conference on Advances in artificial intelligence. Windsor, Canadá 2008.

SOURCEFIRE. **Snort.org.** 2008. Disponível em: < http://www.snort.org >. Acesso em: 01 Agosto de 2011.

SPINOSA, E. J. **Detecção de novidade com aplicação a fluxos contínuos de dados.** Tese (Doutorado em Ciências de Computação e Matemática Computacional): Instituto de Ciências Matemáticas e de Computação – ICMC-USP: 133 p. 2008.

TCPDUMP. **TCPDUMP & Lipcap.** Disponível em : <http://www.tcpdump.org/>. Acesso em: 13 de Jullho de 2010.

TAN, P.; STEINBACH, M.; KUMAR, V. **Instroduction to Data Mining.** Pearson Addison Wesley, 2006. 769p.

TRAMMELL, B.; BOSCHI, E. **RFC 5103: Bidirectional Flow Export Using IP Flow Information Export (IPFIX).** 2008. Disponível em: < http://tools.ietf.org/html/rfc5103 >. Acesso em: 09 de Novembro de 2010.

ZANERO, S.; SAVARESI, S. M. **Unsupervised learning techniques for an intrusion detection system.** Proceedings of the 2004 ACM symposium on Applied computing. Nova York, Estados Unidos 2004.

ZHENQI, W.; XINYU, W. **NetFlow Based Intrusion Detection System. International Conference on Multimedia and Information Technology.** Phuket, Thailand 2008.

WITTEN, I. H.; FRANK, E. 2000. **Data mining: practical machine learning tools and techniques with java implementations**. New York, NY, USA: Morgan Kaufmann Publishers. 629.