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

ADRIANO CESAR RIBEIRO

Correlação e visualização de alertas de segurança em redes de computadores

São José do Rio Preto

2015

ADRIANO CESAR RIBEIRO

Correlação e visualização de alertas de segurança em redes de computadores

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

Orientador: Prof. Dr. Adriano Mauro Cansian

São José do Rio Preto

2015

ADRIANO CESAR RIBEIRO

Correlação e visualização de alertas de segurança em redes de computadores

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

Comissão examinadora:

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

Profa. Dra. Kalinka Regina Lucas Jaquie Castelo Branco
USP – São Carlos

Prof. Dr. Leandro Alves Neves
UNESP – São José do Rio Preto

São José do Rio Preto

2015

Ribeiro, Adriano Cesar.
Correlação e visualização de alertas de segurança em redes de computadores / Adriano Cesar Ribeiro. -- São José do Rio Preto, 2015
62 f. : il., tabs.

Orientador: Adriano Mauro Cansian
Dissertação (mestrado) – Universidade Estadual Paulista "Júlio de Mesquita Filho", Instituto de Biociências, Letras e Ciências Exatas

1. Computação. 2. Redes de computadores - Medidas de segurança. 3. Sistemas de detecção de intrusão (Medidas de segurança) 4. Visualização da informação. 5. Mineração de dados (Computação) I. Cansian, Adriano Mauro. II. Universidade Estadual Paulista "Júlio de Mesquita Filho". Instituto de Biociências, Letras e Ciências Exatas. III. Título.

CDU – 681.3.025

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

Agradecimentos

Aos meus pais, Marines e Vanderlei, pelo apoio e auxílio no decorrer de todo o percurso. Além deles, todos os integrantes da minha família que fizeram parte dessa curiosa etapa.

À minha avó, Linda (em memória), por toda ajuda e dedicação. Sua falta é sentida diariamente.

Ao meu irmão, que me proporcionou momentos de descontração e risos. Seu companheirismo jamais será esquecido.

À minha grandiosa esposa, por ter escolhido trilhar comigo essa e diversas outras aventuras que virão em nossas vidas. Sua sabedoria e compreensão me fascinam e iluminam.

Aos meus amigos, Hélio, Mariano Junior e Rodrigo, pela descontração e sabedoria obtidas em tantas conversas sólidas e bem fundamentadas.

Aos meus prezados amigos de laboratório, Zerati, Gaúcho, Casaca, Leandro e Rafael, bem como tantos outros que passaram pelo ACME! e que tive o prazer de conviver e compartilhar conhecimentos.

Aos meus grandes amigos, também do ACME!, Raphael, Vinicius Galhardi e Vinicius Oliveira pela grande companhia e orientação. Com amigos assim o caminho fica mais fácil.

Ao meu orientador, Adriano Mauro Cansian, pela orientação tanto acadêmica quanto para a vida.

Ao programa UOL Bolsa Pesquisa pela bolsa concedia para a realização inicial deste projeto.

À CAPES (Coordenação de Aperfeiçoamento de Pessoa de Nível Superior) pela bolsa de mestrado concedida para a conclusão deste projeto.

RESUMO

Os sistemas de detecção de intrusão fornecem informações valiosas em relação à segurança das redes de computadores. No entanto, devida à quantidade de ameaças inerentes aos sistemas computacionais, os registros dessas ameaças na forma de alertas podem constituir de grandes volumes de dados, muitas vezes bastante complexos para serem analisados em tempo hábil. Esta dissertação apresenta uma abordagem para correlacionar alertas de segurança. A metodologia tem como princípio a utilização de mineração de dados para a coleta de informações constituintes nos alertas providos pelos sistemas detectores de intrusão. Primeiramente, os alertas são classificados em tipos de ataques para que, na sequência, sejam clusterizados de forma a agrupar alertas com características semelhantes. Por fim, a correlação é realizada baseada na ocorrência dos alertas em cada cluster e, dessa forma, é obtida uma visão geral do cenário de ataque, utilizando de métodos de visualização de tais ocorrências maliciosas.

Palavras-chaves: Correlação de alertas. Redes de computadores. Visualização. Mineração de dados.

ABSTRACT

Intrusion detection systems provides valuable information regarding the security of computer networks. However, due to the amount of threats inherent in computer systems, records of these threats in the form of alerts can be large volumes of data, often quite complex to be analyzed in a timely manner. This paper presents an approach to correlate security alerts. The methodology is based on the use of data mining to the collection of constituent information in the alerts provided by intrusion detection systems. First, alerts are classified into types of attacks so that, later, are clustered in order to compose alerts with similar characteristics. Finally, the correlation is performed based on the occurrence of alerts in each cluster and thereby an overview of the attack scenario is obtained using visualization methods of such malicious events.

Keywords: Alert correlation. Computer network. Visualization. Data mining.

ÍNDICE

LISTA DE TABELAS

LISTA DE FIGURAS

Capítulo 1 – Introdução

1.1 Considerações iniciais

Cada vez mais a Internet evolui para uma estrutura maior e complexa. O volume de tráfego gerado, com a grande aderência de dispositivos móveis, cada vez mais comuns entre os usuários, culmina em uma estrutura extremamente crítica, para governos, empresas, instituições e os próprios usuários finais.

Evidentemente, existem ameaças inerentes aos sistemas computacionais que compõem as inúmeras redes ao redor do mundo. Normalmente, os sistemas que vigiam redes de computadores são conhecidos como IDS (*Intrusion Detection System*). Esses sistemas procuram por comportamentos maliciosos em redes ou *hosts*. Ao encontrar tais comportamentos, esses sistemas geram alertas de segurança, que são relatórios com diversas informações relevantes à tentativa de intrusão, como *hosts* participantes, portas utilizadas, nível de ameaça, processos, entre outras. Alguns trabalhos que contemplam mais detalhes sobre IDS podem ser analisados em (ROESCH, 1999), (PAXSON, 1998), (ISS, 2004) e (HERBELEIN, et al., 1990).

Devido à grande quantidade de serviços executando na Internet, os sistemas computacionais estão sujeitos a inúmeras ameaças. Para atender toda a gama maliciosa da Internet, existem diversos IDSs, cada um focado em uma estratégia, usando as mais variadas fontes de informação para detectar atividades maliciosas. Como por exemplo, um IDS pode fazer uso de detecção por anomalia ou por abuso. Com isso, para cada método utilizado, existe uma forma de exibir o resultado de sua análise.

Os quadros estatísticos das ameaças existentes na Internet nos últimos anos apresentados em (CERT.BR, 2015) evidencia o crescimento de problemas relacionados à segurança de computadores. Essa quantidade de ameaças comprova a importância da segurança de redes e sua necessidade de avanços e pesquisas. Dessa forma, a visibilidade do grande volume de informações que pode ser gerado por um IDS é uma tarefa bastante dispendiosa para um administrador de rede.

A visualização de eventos de redes pode ser uma ferramenta bastante útil. A quantidade de *logs* e alertas gerados por alguns dias podem ser sumarizados em uma imagem, provendo uma visão geral da rede. Essa visualização, evidentemente, varia de acordo com a necessidade de cada administrador. É possível que seja necessário apenas avaliar o consumo de banda em determinados horários, a média de acessos a servidores ou a frequência que os usuários acessam determinados serviços.

1.2 Identificação do problema e justificativa

A comunidade de pesquisa relacionada à segurança da informação tem se preocupado em conseguir mitigar da melhor forma possível as ameaças relativas a redes de computadores. Para cada diferente ponto de observação existe uma maneira de detectar alguma atividade maliciosa. Devido a isso, a quantidade de informação gerada pelos sistemas de intrusão é bastante volumosa. Assim, tornou-se difícil a análise de todo o conjunto de informações disponíveis ao administrador.

Cada sistema responsável por monitorar uma parte do que compõe uma rede possui problemas relacionados à falta de precisão, e em alguns casos gera alertas falsos. Dessa forma, além da quantidade de informação relacionada à rede, ainda existe a possibilidade de eventuais erros de detecção de intrusão.

Para aprimorar a detecção de intrusão, a utilização de métodos de correlação de eventos em redes é uma forma de melhorar esse quadro. Por exemplo, correlacionando alertas de forma a obter uma precisão maior e, consequentemente, eliminar alertas não relevantes ou falsos para o ambiente monitorado. Além disso, o modelo de IDS apenas se preocupa em gerar os alertas, reconhecendo e analisando as atividades maliciosas. Dessa forma, os IDSs não avaliam as consequências que atividades maliciosas isoladas podem gerar em uma rede, ou seja, não associam alertas entre si, de modo a montar um cenário de ataque. Obter uma visão global de cenários de ataques permite que um administrador poupe tempo de análise, avaliando melhor a estratégia e o alvo do atacante.

A correlação de ataques de múltiplos passos é um avanço em relação a análise da situação da rede. Entretanto, ainda é possível obter uma visão maior da situação.

Analisar eventos maliciosos em redes de computadores utilizando ferramentas gráficas pode aumentar a compreensão do ataque, além de promover novas deduções e inferências a respeito. Como exemplo, responder e propor novas questões. A visualização é capaz de elucidar novos pensamentos, permitindo que sejam observados novos padrões e estratégias. Outro exemplo reside na tomada de decisões, onde se fazem necessárias contra-medidas rápidas. Una-se a isso o fato de que visualizar inspira novas ideias e modelos, permitindo atingir novos níveis de compreensão e de segurança, em se tratando de redes de computadores.

Sendo assim, o objetivo do trabalho é correlacionar alertas de segurança de redes de computadores utilizando mineração de dados para obter cenários de ataques que correspondem a ameaças mais elaboradas e complexas. Além disso, ao encontrar um cenário de ataque, gerar uma visualização do cenário possibilitando maior compreensão do mesmo.

1.3 Organização da dissertação

Esta dissertação está assim dividida: no Capítulo 2 é descrita a fundamentação teórica relacionada ao tema, como os conceitos de ataques em redes, correlação de alertas e visualização voltada para segurança da informação. No Capítulo 3 são apresentados os trabalhos relacionados, tratando de correlação de alertas e visualização de informação. No Capítulo 4 é apresentada a metodologia e o ambiente utilizado para a correlação de alertas e visualização. O Capítulo 5 apresenta os resultados obtidos com a abordagem proposta. Por fim, no Capítulo 6 são feitas as considerações finais sobre o tema.

Capítulo 2 – Fundamentação teórica

2.1 Considerações iniciais

Este capítulo tem como objetivo descrever toda fundamentação teórica das tecnologias envolvidas para o desenvolvimento deste projeto. Na seção 2.2 são apresentados os conceitos e problemas de segurança inerentes às redes de computadores. Em seguida, na seção 2.3, é discutida a correlação de alertas de detectores de intrusão, incluindo a utilização de técnicas de mineração de dados para tal, bem como a apresentação de um padrão que normaliza os alertas gerados por diferentes sistemas detectores de intrusão. Na seção 2.4 são apresentadas as técnicas de visualização e sua importância no âmbito da segurança de computadores. A seção 2.5 finaliza com as considerações finais do capítulo.

2.2 Segurança de computadores e redes

Segurança da informação tem como objetivo proteger e preservar informações e sistemas computacionais. A segurança desses sistemas é de fundamental importância uma vez que são mantidas informações sigilosas e valiosas como, por exemplo, em sistemas bancários, sites de compras, banco de dados, entre outros. Dessa forma, é fundamental que sistemas computacionais possuam garantia de que sejam seguros contra as diversas ameaças a que estão suscetíveis.

Dentre as ameaças que podem ser direcionadas aos sistemas computacionais, é importante destacar as mais comuns que ocorrem em uma rede. As subseções a seguir abordam alguns eventos ilícitos de redes.

2.2.1 Mapeamento de rede

Normalmente, o mapeamento de rede é o primeiro passo para ataques que serão disparados futuramente. Essa prática, conhecida também como varredura, prospecção, ou *scan*, consiste em explorar um ambiente de rede a procura de *hosts*

ativos. Partindo do pressuposto de que um ataque desse tipo seja apenas o início do ataque total, um administrador de rede pode aumentar sua atenção para o tráfego originado do atacante.

De posse da informação de quais *hosts* estão ativos na rede, o atacante pode procurar por serviços que estão executando nos *hosts* para direcionar seus próximos ataques.

2.2.2 Mapeamento de serviços

Mapeamento de serviços é o levantamento de quais serviços executam em determinados *hosts* da rede. Normalmente, um *scan* de serviços é realizado para descobrir o que um *host* está executando para que em seguida um ataque possa ser feito especificamente para a aplicação alvo.

No entanto, algumas vezes essas técnicas não necessariamente são eventos maliciosos. Tanto o mapeamento de rede quanto o de serviços podem ser usados para auditoria e verificação do correto funcionamento da rede.

2.2.3 Negação de serviço

Ataques de negação de serviço (conhecido também como *Denial of Service* – DoS) tem como objetivo impedir que um *host* possa atender as requisições de usuários legítimos. Ataques desse tipo podem afetar tanto recursos providos por uma máquina na rede ou para esgotar a banda de uma rede inteira (ZARGAR; JOSHI; TIPPER, 2013).

Existem várias formas de realizar um ataque DoS, como por exemplo a utilização de falhas de protocolo, a partir de falhas de *software* ou a partir de pacotes mal formados. Um bom exemplo é conhecido como *spoofing* (falsificação) de um endereço de origem. O atacante ao construir um pacote mal formado, colocava o mesmo endereço de origem e destino, causando um laço no destinatário, fazendo-o responder para si mesmo. Com isso, os recursos como a memória são consumidos e, consequentemente, o serviço a usuários legítimos fica impedido.

Outro bom exemplo é o ataque de negação de serviço que faz uso das *flags* do TCP (*Transmission Control Protocol*). Alguns serviços são alvos de ataques de SYN *Flood*. A metodologia desse ataque é abrir várias conexões com a *flag* SYN do TCP sinalizadas fazendo com que o *host* reserve recursos para as falsas conexões, causando negação de serviço para requisições legítimas.

Em se tratando de ataques de negação de serviço que visam o esgotamento de banda, as técnicas empregadas necessitam de mais esforço do que se fosse para negar serviço de apenas um *host*. Ataques desse tipo normalmente são feitos utilizando *botnets*. As *botnets* são conjuntos de computadores que foram comprometidos ao redor do mundo e são controlados remotamente por um mestre. Esse mestre pode iniciar um ataque de negação de serviço distribuído (DDoS - *Distributed Denial of Service*), a qualquer momento, enviando comandos aos seus computadores infectados, conhecidos também como zumbis. Esse tipo de ataque gera um grande tráfego, ocasionando que seja impedido o serviço por esgotamento de banda.

2.2.4 Ataques por *malwares – worms*

Um *Malware* que possui a capacidade de se propagar sem o auxílio de outros programas, arquivos ou usuário, é denominado *worm*. Normalmente um *worm* é capaz de se propagar pela rede em busca de vulnerabilidades nos sistemas computacionais (BAZRAFSHAN et al., 2013). Os problemas gerados por *worms* são diversos, como por exemplo: negar serviço de forma similar a ataques DoS ou DDoS, consumir memória de um *host* infectado, demandar tempo de remoção desses artefatos maliciosos, dentre outros. Da mesma forma que os ataques de negação de serviço, é muito importante que esse tipo de ameaça seja detectado em tempo hábil e que as contra-medidas sejam executadas a tempo de minimizar os danos.

2.2.5 Classificação de sistemas detectores de intrusão

Os eventos ilícitos citados nas subseções anteriores apresentam motivações sérias em relação aos problemas de segurança em redes de computadores. Para combater essas ameaças, sistemas de detecção de intrusão (IDS – *Intrusion Detection*

System) ou sistemas de prevenção de intrusos (IPS – *Intrusion Prevention System*) são necessários para minimizar o êxito dessas ameaças espalhadas na rede.

Os IDSs possuem uma variedade de possibilidades interessantes para a detecção de intrusos. Existem diversas ferramentas que fazem uso desta metodologia. Um dos mais famosos e de domínio público, é o *Snort* (ROESCH, 1999). O *Snort* faz uso de uma base de assinaturas que pode ser atualizada frequentemente pela *Internet*. Essa base contém assinaturas de ataques e possíveis quebras de políticas de uso que são comuns. Entretanto, dependendo da motivação, é necessário que seja feita uma prevenção de um intruso. Para isso, os IPSs são eficazes em seu propósito.

Sistemas de prevenção de intrusão são relativamente recentes e possuem diversas aplicações (ALQAHTAMI; BALUSHI; JOHN, 2014), (EMRICK; HU, 2014) e (FANG et al., 2013). Um IPS é um sistema que combina características de *firewall* e de um IDS, sendo este, considerado proativo. Possui como característica de estender a funcionalidade de um firewall até a camada de aplicação, para que assim, utilize as técnicas de IDS para analisar o conteúdo dos pacotes (XINYOU; CHENGZHONG; WENBIN, 2004).

2.3 Correlação de ataques

A comunidade de segurança de computadores desenvolveu diversos IDSs, cada um com seu foco e suas funcionalidades. Com o número de sistemas desse tipo, os administradores de segurança tem em poder uma vasta quantidade de informação para avaliar, sendo que muitos alertas gerados podem ser falsos positivos. Dessa forma, o trabalho necessário para administrar uma rede utilizando mais de um IDS, por exemplo, é muito desgastante e até impreciso.

Diante desse problema, a correlação de alertas gerados pelos diferentes sistemas de segurança é uma abordagem pertinente nesse cenário de atuação. Existem diversas metodologias para correlacionar alertas, podendo-se destacar a utilização de métodos probabilísticos, redes neurais, árvores de decisão, mineração de dados, entre outros. Além disso, é também possível variar na abordagem da correlação. Por exemplo, correlacionar alertas de modo a obter uma minimização de

falsos positivos, ou ainda, correlacionar alertas de modo a obter uma sequência de passos que um atacante realizou até obter sucesso na intrusão.

2.3.1 Padronização de alertas

Para alcançar o principal objetivo da correlação de alertas, é recomendado que exista uma padronização desses alertas. Uma vez que cada IDS ou IPS opera de uma forma, suas saídas, ou seja, seus alertas são diferentes. Para atingir essa padronização o IDMEF (*Intrusion Detection Message Exchange Format*), descrito no RFC 4765 (DEBAR; CURRY; FEINSTEIN, 2007) provê o suporte necessário. O IDMEF, exemplificado na Figura 2.1, é um padrão voltado para alertas de segurança, podendo exibir informações bastante simples ou até mesmo um alerta mais detalhado. Essa flexibilidade deriva da necessidade de atender os mais diversos tipos de alertas que podem ser gerados pelos inúmeros sistemas de segurança, sendo assim, necessário atender desde o sistema mais simples até o mais elaborado.

```
<IDMEF-Message version="0.3">
  <Alert ident="48562" impact="unknown">
    <Analyzer analyzerid="Snort:1.8.6:26.100.101.3">
      <Node><name>brp-snort</name></Node>
    </Analyzer>
    <CreateTime ntpstamp="0xc12b141a.0xa5baa000"/>
    <Source><Node>
        <Address category="ipv4-addr">
          <address>4.22.161.203</address></Address>
    </Node></Source>
    <Target><Node>
        <Address category="ipv4-addr">
          <address>26.100.101.1</address></Address>
    </Node></Target>
    <Classification origin="vendor-specific">
    <name>ICMP PING NMAP</name></Classification>
  </Alert>
</IDMEF-Message>
```

Figura 2.1 Exemplo de um alerta no padrão IDMEF (VALEUR et al., 2004).

Pode-se obervar na figura 2.1 um alerta gerado pelo *Snort*, representando um *ping* ICMP (*Internet Control Message Protocol*) originado em 4.22.161.203 com destino a 26.100.101.1.

A obtenção de uma saída padronizada é importante para que exista uma independência do uso dos variados sistemas de defesa de intrusão. Por exemplo, um sistema que realiza algum tratamento nos alertas não precisa ficar restrito a apenas um ou poucos IDSs ou IPSs, podendo agregar novos sistemas em suas análises.

2.3.2 Mineração de dados

Dentre os diversos tipos de alertas que podem ser gerados é comum que seja necessária a utilização de algoritmos que auxiliem em sua classificação. Classificar um alerta significa agrupa-los num mesmo tipo. Com essa informação é possível traçar um perfil ou cenário de ataque, baseando-se em métodos de correlação de eventos.

Os algoritmos de classificação são os mais variados, sendo os mais comuns os que utilizam redes neurais artificiais ou métodos probabilísticos. Existem ferramentas que implementam diversos algoritmos estatísticos. Como exemplo pode-se citar o Weka (WEKA, 2015). O Weka é uma coletânea de algoritmos *open source* para mineração de dados. Esses algoritmos possuem aplicações para pré-processamento de dados, regras de associação, classificação, correlação e visualização.

A relevância na utilização de algoritmos de mineração de dados é fundamental. Em se tratando de uma rede de grande porte, um IDS possivelmente emitirá um número consideravelmente alto de alertas. Considerando que existam alguns falsos positivos dentro desse conjunto de alertas, um pré-processamento é necessário para a eliminação de alertas irrelevantes. Esse pré-processamento ajuda também na eliminação de alertas que não se correlacionariam com outros alertas.

Além da fase de pré-processamento, as fases de classificação e associação também fazem uso desses algoritmos estatísticos. Para classificação, um algoritmo

bastante promissor é o *AutoClass* (CHEESEMAN; STUTZ, 1996). No trabalho como o de (ERMAN; MAHANTI; ARLITT, 2006), é apresentado que o *AutoClass*, em comparação com o método supervisionado de *Naive Bayes*, é 9% superior em relação a este último citado, envolvendo classificação de tráfego de rede. Em outro trabalho dos mesmos autores, o *AutoClass* foi comparado com outros algoritmos, como o *K-means* e o *DBSCAN* (ERMAN; ARLITT; MAHANTI, 2006). Novamente o *AutoClass* mostrou melhor precisão na classificação.

O *AutoClass* é um algoritmo de clusterização particional e classificação *fuzzy*. O mecanismo de funcionamento tem como base o modelo probabilístico de *Bayes* (HANSON; STUTZ; CHEESEMAN, 1991). A classificação é realizada de forma automática gerando os *clusters*, também conhecidos como classes. Esse agrupamento ocorre baseado nas características semelhantes de um dado grupo de dados para treinamento. O resultado obtido com o agrupamento é um conjunto de classes, que compreendem características similares entre si, que pode ser usado posteriormente para classificação de novos dados.

A separação de um conjunto de dados em classes e clusters pode ser realizada utilizando o algoritmo *K-means*. Esse algoritmo utiliza métodos heurísticos que permite organizar X objetos pertencentes a uma base de dados em K partições, sendo cada uma dessas partições um cluster. O *K-means* é um algoritmo simples, escalável e pode ser usado em grandes bases de dados (NALDI, 2011).

O método utilizado pelo *K-means* consiste em particionar um conjunto de dados X em uma quantidade K de clusters baseado em uma medida de dissimilaridade fornecida (NALDI, 2011). O *K-means* cria um protótipo de clusters baseado em um centróide, sendo que, geralmente, é a média de um grupo de pontos constituintes de um grupo de objetos de um espaço contínuo e n-dimensional (TAN; STEINBACK; KUMAR, 2006).

O algoritmo inicializa os grupos de objetos por meio de N protótipos, ou seja, pontos que representam estes grupos. Cada ponto é atribuído a um centróide que esteja mais próximo. Sendo assim, um cluster é a coleção de pontos que foram atribuídos a um centróide. O centróide é atualizado de acordo com cada ponto que

for atribuído ao cluster. O algoritmo realiza iterações de atribuições de objetos e atualização dos centróides até que os centróides continuem os mesmos.

O *K-means* é formalmente descrito em (TAN; STEINBACK; KUMAR, 2006). Uma ilustração do algoritmo *K-means* pode ser vista na Figura 2.2.

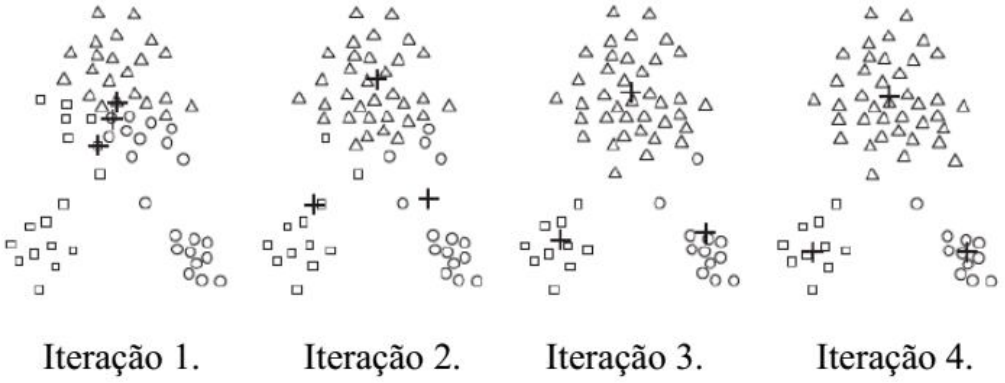

Figura 2.2. Exemplo de execução do *K-means*.

Com relação à mineração de associações em uma base de dados o algoritmo *Apriori* ainda possui aplicação científica (QUIN; TANG; CHENG, 2013), (YI et al., 2014) e (SINGH; KUMAR; MAURYA, 2014) e é um dos mais consolidados e aprimorados na literatura. Esse algoritmo consiste em buscar por padrões freqüentes em uma base de dados, ou seja, ocorrências freqüentes de diversos tipos. Por exemplo, compra e venda de produtos, tomada de ações, eventos, entre outros.

O algoritmo *Apriori* faz uso de duas métricas que são o *support* e o *confidence*. O parâmetro *support* é a probabilidade de que um evento contenha os itens relacionados na associação, ou seja, X U Y. O parâmetro *confidence* é a probabilidade condicional que um evento que possua o item X também possua o item Y. Por exemplo, suponha que exista a seguinte relação;

[tomate] [cenoura] => [suco de laranja]

Sendo o *support* igual a 50 e o *confidence* 80. Esses valores indicam que, para o caso do *support*, o conjunto tomate, cenoura e suco estão presentes em 50% das ocorrências consideradas. Para o caso do *confidence* indica que nas ocorrências onde existe os itens tomate e cenoura ocorreu o suco de laranja em 80% dos casos.

Além desses valores, existem ainda outras métricas relacionadas com a correlação estatística dos dados que constituem a base. Apesar dos parâmetros *support* e *confidence* conseguirem eliminar regras de associação que não sejam interessantes, ainda existe a possibilidade de que algumas regras encontradas não sejam válidas. Para auxiliar a encontrar regras conhecidas como forte é utilizada, por exemplo, a medida *lift*. Essa medida é a probabilidade condicional dos itens envolvidos na relação, conforme a equação 2.1.

$$lift(A, B) = P(A \cup B) / P(A)P(B) \qquad \text{equação 2.1}$$

A métrica *lift* é simples. A ocorrência de um item A é independente de um item B se $P(A \cup B) = P(A)P(B)$. Se não, existe uma correlação entre os itens. Se o valor da equação 2.1 for menor do que 1, então a ocorrência de A é negativamente correlacionada com a ocorrência de B. Se o resultado for maior do que 1, então A e B estão positivamente correlacionados, significando que a ocorrência de um implica na ocorrência do outro. Caso o resultado seja 1, então A e B são independentes e não existe nenhuma correlação entre eles. Além do *lift* existem outras métricas como o *cosine*, *all_confidence*, χ^2, entre outras.

Todos os métodos discutidos anteriormente nesta seção são importantes para correlacionar alertas de segurança. Em se tratando de correlacionar alertas de diferentes IDSs, é possível obter uma maior precisão na detecção de intrusão, minimizando os casos de falso positivo. Relacionado à correlação de ataques visando a montagem de um perfil ou cenário de ataque, é possível obter uma visão mais global da ameaça, requerendo menos esforço na análise de diversos alertas.

2.4 Visualização para segurança de redes

A visualização voltada para a segurança de redes é um termo relativamente novo, segundo (MARTY, 2008) e (CONTI, 2007). Sua aplicação visa melhorar e

facilitar a compreensão de grandes quantidades de informação, ajudando a identificar padrões, ameaças, estruturas, relações, entre outras. Visualização de redes não é só eficiente, mas também é bastante eficaz em mostrar a comunicação das informações (WARE, 2004).

A área de segurança de redes trata de diversos sistemas de segurança e, obviamente, enfrenta inúmeras ameaças como vulnerabilidades, *exploits*, *malwares*, entre outras. A complexidade e dinamicidade de uma rede demandam profundo conhecimento e compreensão de suas operações internas. Um sistema que possibilita visualizar tudo relacionado a essa rede, permite que seja possível minimizar os esforços e maximizar a compreensão dessa rede como um todo.

A visualização promove diversas inferências que muitas vezes ficam obscuras e perdidas no grande volume de informação a ser analisado. Os benefícios adquiridos podem ser:

- **Responder e fazer perguntas**: muitas vezes uma imagem ajuda a abstração de um conjunto de informações, facilitando as respostas. Além disso, a imagem pode gerar novas deduções, ocasionando perguntas que antes eram inalcançáveis em outros formatos.

- **Tomar decisões**: é possível tomar decisões de maneira mais rápida e precisa. Em se tratando de segurança da informação, esse tipo de resposta é fundamental para uma contra medida eficaz e o bom funcionamento da rede.

- **Aumentar a eficiência**: uma visão geral da rede reduz o tempo gasto com análise de *logs* ou alertas, aumentando a eficiência do trabalho e permitindo que os esforços sejam direcionados a outras tarefas.

A visualização, obviamente, necessita de uma fonte de informação. Existe uma grande gama de fontes de informações. De acordo com a Tabela 2.1 é possível observar a quantidade de fontes de informações que podem ser usadas para visualização de segurança de redes.

Tabela 2.3 Fontes de informação para visualização (SHIRAVI; SHIRAVI; GHORBANI, 2012).

Event Type	Data Source	Devices
Network Traces	–Raw Packets	Tcpdump, Tshark
	–Netflow Records	Cisco NefFlow NDE, Cisco NSEL Netflow
Security Events	–Intrusion Detection Systems	Cisco CSA, Cisco IDS, Enterasys Dragon, Fortinet Fortigate, Juniper ISG, SNORT, Niksun NetVCR, SourceFire Intrusion Sensor
	–Intrusion Prevention Systems	ForeScout ConterACT, Juniper NetScreen IDP, McAfee Intrushield, Radware Defense Pro, FireEye, Tipping Point X, IPAngel
	–Firewalls	Check Point, Linux Iptables, PaloAlto PA, Cisco ACE
	–Virtual Private Networks	ArraySP, Nortel VPN Gateway, Checkpoint VPN-1, Cisco ASA
	–Anti-virus	Mcafee, Sophos, Symantec, Trend Micro
Network Activity Context	–Layer 7 application context	Q1 Labs QFlow, Foundry SFlow, Juniper JFlow, Packeteer FDR
User/Asset Context	–Vulnerability Scanners	NMap, eEye REM, Nessus, Rapid7 NeXpose, SecureScout nCircle IP360, Patchlink Scan, Qualys, Saint
	–Identity and Access Management	Microsoft ForeFront Identity Manager, Identity Forge Quest Identity Manager One, EmpowerID
Network Events	–Switches	Cisco CatOS, Cisco Catalyst, 3Com 8800 Series
	–Routers	Cisco Routers, Enterasys Router, Juniper Router, Nortel Router
	–Servers	Apache, BlueCoat SG, Cisco Ironport, IIS, Sun Sendmail
	–Hosts	Windows, Linux, Solaris, IBM AIX RACF, HP Tandem
Application Logs	–Application Databases	IBM DB2, SQL Server, Oracle, Imperva SecureSphere, Sybase
	–Workflow	
	–Enterprise Resource Planning	
	–Management Platforms	

De posse das fontes de informações escolhidas um tipo específico de visualização pode ser obtido. A figura 2.3 ilustra como a visualização de redes pode ser útil para problemas de segurança em redes.

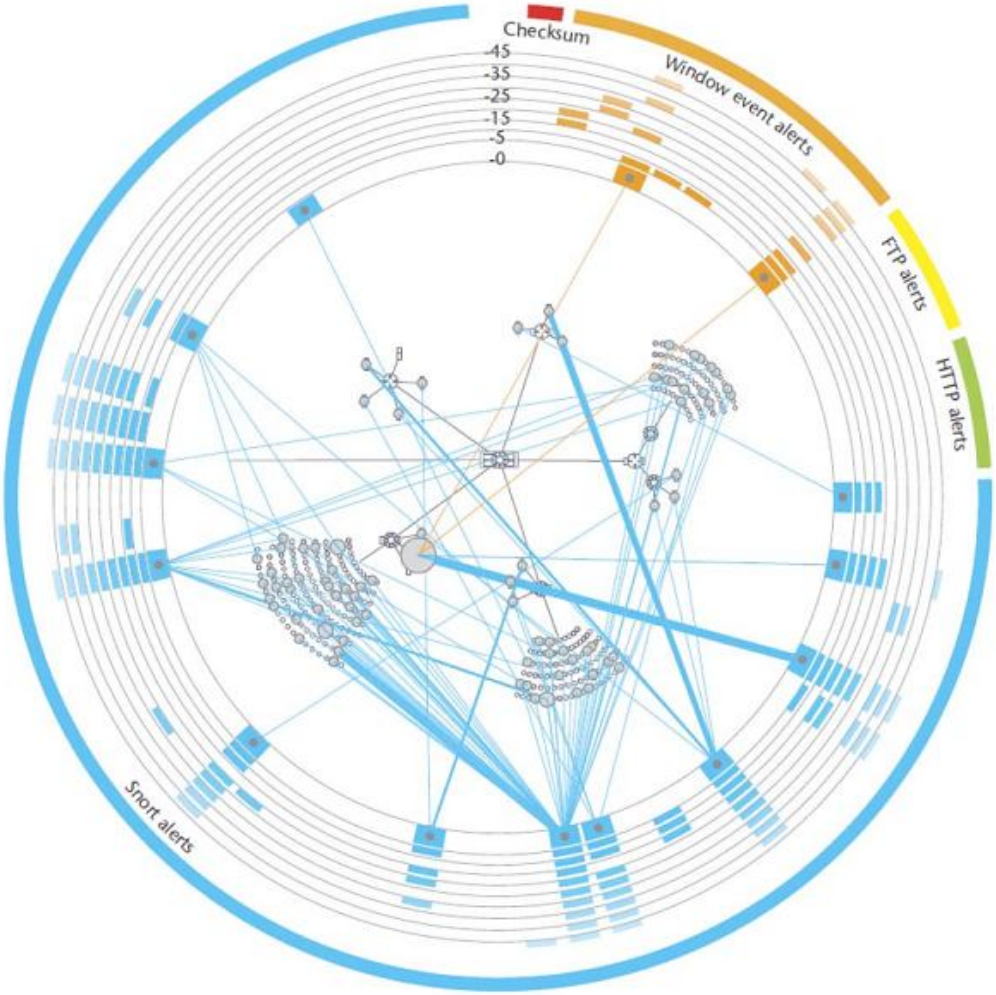

Figura 2.3 Visualização de uma topologia de rede através de alertas (LIVNAT et al., 2005).

A Figura 2.3 ilustra os alertas gerados em uma rede. O anel mais externo representa os tipos de alerta e qual é a fonte dessa informação. Os anéis mais internos representam a severidade dos alertas e um histórico do tempo em que foram gerados.

Existem diversos outros sistemas voltados para visualização, bem como ferramentas que auxiliam o desenvolvimento de uma técnica para visualizar os dados obtidos. Como exemplo, o *PrimeFaces* (PRIMEFACES, 2015) que é uma biblioteca em JSF (*Java Server Faces*) que possui grande flexibilidade e aplicação. Outro sistema que possui o foco voltado para gerar grafos voltados para segurança de redes

de computadores é o *AfterGlow* (AFTEGLOW, 2015). O *AfterGlow* é uma ferramenta voltada para a segurança da informação, possuindo atributos relevantes para evidenciar informações que facilitam a observação do que acontece na rede monitorada.

As diversas formas de visualizar os dados possibilitam um conjunto bastante vasto de técnicas e maneiras de dispor as informações. Cada técnica possui seu foco e sua interpretação, sendo necessária uma avaliação do tipo de informação que se deseja ilustrar e quais são suas variáveis.

2.5 Considerações finais

Este capítulo apresentou os principais conceitos e tecnologias envolvidos neste trabalho. A utilização de alertas é fundamental para o trabalho, sendo necessária sua padronização para que posteriormente seja possível aplicar os métodos estatísticos de classificação e correlação. Por fim, a visualização é necessária para facilitar a compreensão de eventos de segurança em redes de computadores.

Capítulo 3 – Trabalhos relacionados

3.1 Considerações iniciais

Neste capítulo são apresentados os trabalhos relacionados com o projeto proposto. Na seção 3.2 serão apresentados trabalhos relacionados com correlação de ataques em redes de computadores. A seção 3.3 discutirá os trabalhos relativos à visualização de informação voltada para segurança de redes de computadores. Por fim, a seção 3.4 apresenta as considerações finais.

3.2 Correlação de ataques

Em (YUAN; ZOU, 2011), é feita uma revisão sobre os métodos de classificação e correlação. Além disso, propõe uma arquitetura para classificar e correlacionar eventos de segurança. Os tipos de classificação são abordados, diferenciando-os em classificação baseada em informação e baseada em associação. A classificação baseada em informação é aquela que faz uso de *logs* de sistema, alertas de IDSs e IPSs, *logs* de *firewall*, entre outros. Já a classificação baseada em associação é o método que cria regras de relações ou padrões entre as informações relativas à rede. Por fim, o trabalho apresenta uma arquitetura sobre como pode ocorrer as analises. A Figura 3.1 ilustra a arquitetura proposta pelos autores.

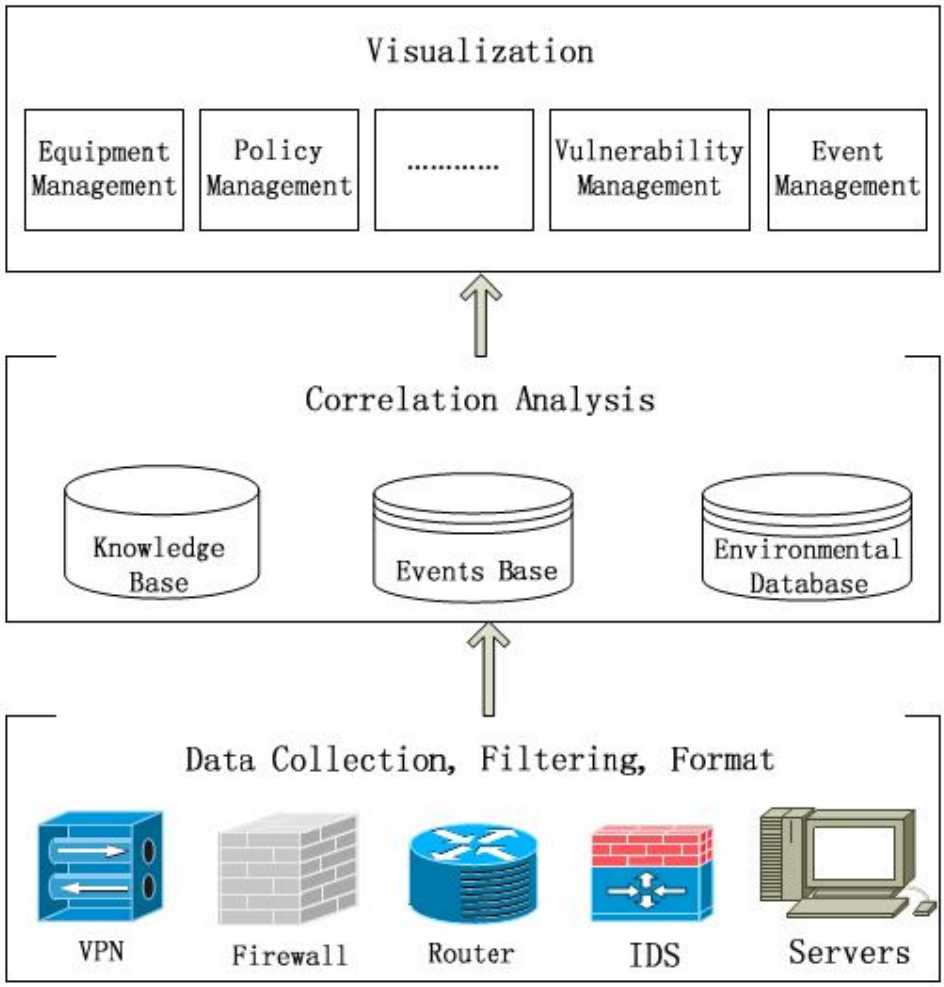

Figura 3.1 Arquitetura proposta por (YUAN; ZOU, 2011).

No trabalho de (XIAO et al., 2008), é apresentada uma fusão de alertas baseada em clusters (agrupamentos) e correlação. A fusão de alertas é uma técnica utilizada para reduzir possíveis alertas redundantes, caso esteja sendo utilizada mais de uma fonte de informação. Esse cenário é mais comum quando se trabalha com mais de um IDS. A Figura 3.2 ilustra o modelo utilizado no trabalho.

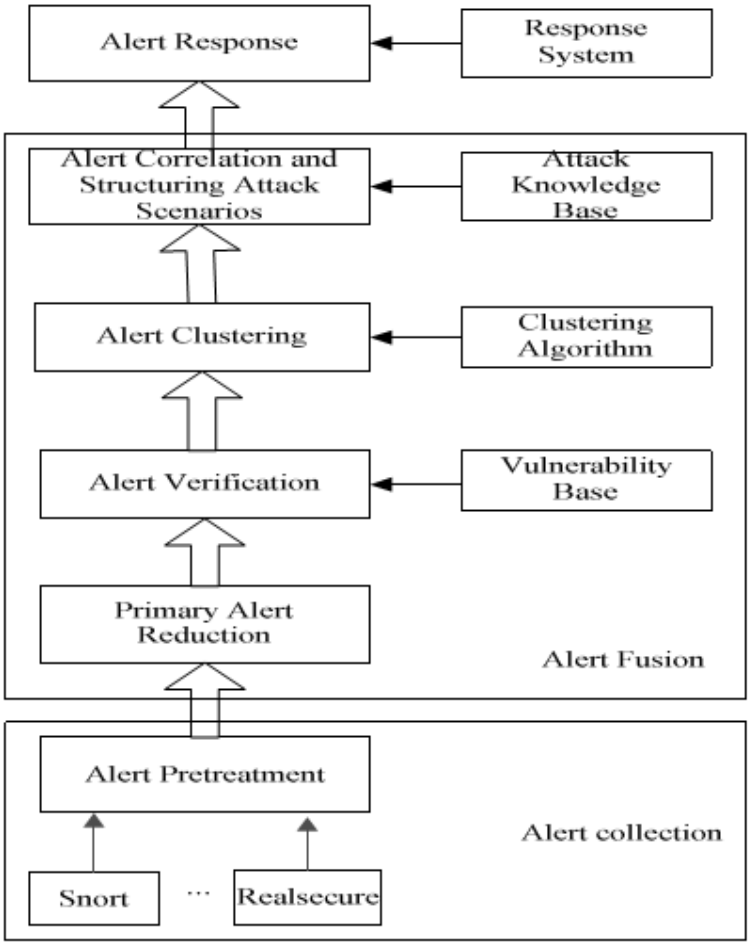

Figura 3.2 Modelo proposto pelos autores em (XIAO et al., 2008).

Como método para agrupamento de alertas foi utilizado um algoritmo de lógica *fuzzy*. A verificação dos alertas é feita visando a redução de falsos alertas. Os autores apresentam uma desvantagem na verificação, uma vez que é feita uma busca para analisar se o determinado alvo existe e possui alguma vulnerabilidade. Verificar os alertas pode, na maioria das vezes, gerar novos alertas. Como resultados, o modelo proposto foi testado em no conjunto de dados de ataques DARPA 2000 (DARPA, 2000), apresentando bons resultados.

No trabalho apresentado por (XUEWEI et al., 2011), é proposto um *framework* para análise da situação da segurança da rede. Esse framework é dividido

em três partes, que são: informação sobre a situação, correlação de eventos e avaliação da situação. A Figura 3.3 ilustra o modelo proposto pelos autores.

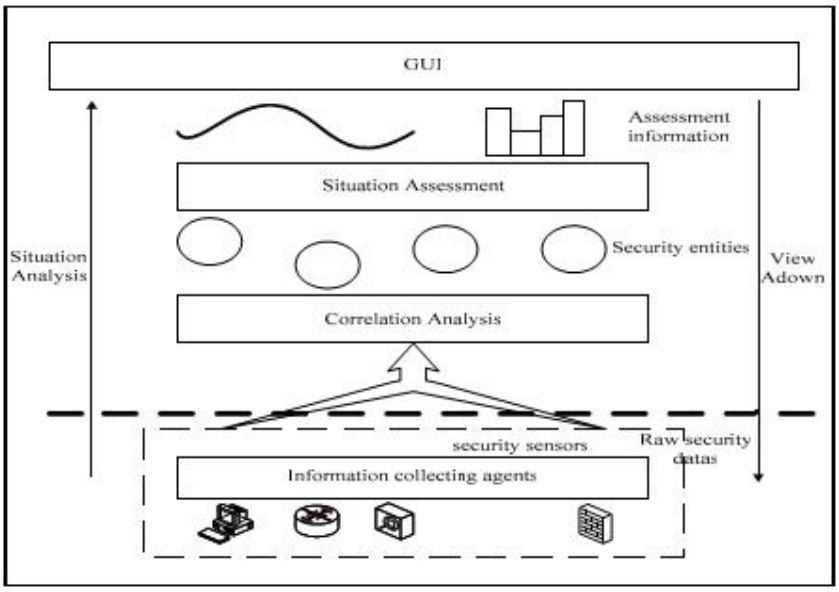

Figura 3.3 Framework para analisar situação da segurança da rede proposto em (XUEWEI et al., 2011).

A coleta de informações é proveniente de diversas fontes. O sistema de correlação é responsável por fazer uma ligação lógica entre as vulnerabilidades, o ataque, os serviços que executam na rede e os mecanismos de defesa disponíveis. Essa correlação é feita por uma rede neural devida sua complexidade. Por fim, após essa análise, é realizada uma avaliação da segurança da rede monitorada.

Em (AHMADINEJAD; JALILI, 2009), é apresentado um modelo de classificação e correlação de alertas visando diminuir a quantidade dos mesmos, transformando-os em um alerta global. A metodologia utilizada se baseia na utilização de janelas de tempo com a intenção de minimizar a quantidade de comparações entre os alertas para gerar um novo alerta global, além de melhorar a precisão.

Os alertas são correlacionados utilizando a ferramenta Weka (GARNER, 1995) com o algoritmo *Boosting* (FREUND,1995). Como atributos para

correlacionar os alertas foram utilizados os seguintes: IP (*Internet Protocol*) de origem e destino e porta de origem e destino. Dessa forma, foram criadas similaridades entre esses atributos de acordo com o apresentado em (ZHU; GHORBANI,2006). Os testes foram realizados utilizando o conjunto de informações de ataques DARPA 2000. Os resultados mostraram que o método de correlação de alertas reduziu bastante necessidade de comparações para gerar um novo alerta global.

O trabalho proposto por (KAVOUSI; AKBARI, 2012), analisa o problema de detectar novas estratégias de ataques por meio de correlação de alertas. Para isso, os autores fazem uso de métodos estatísticos como o mecanismo de Bayes. Dessa forma, são gerados novos padrões de ataques que podem ser usados posteriormente em IDSs.

A Figura 3.4 ilustra o sistema de correlação utilizado pelos autores.

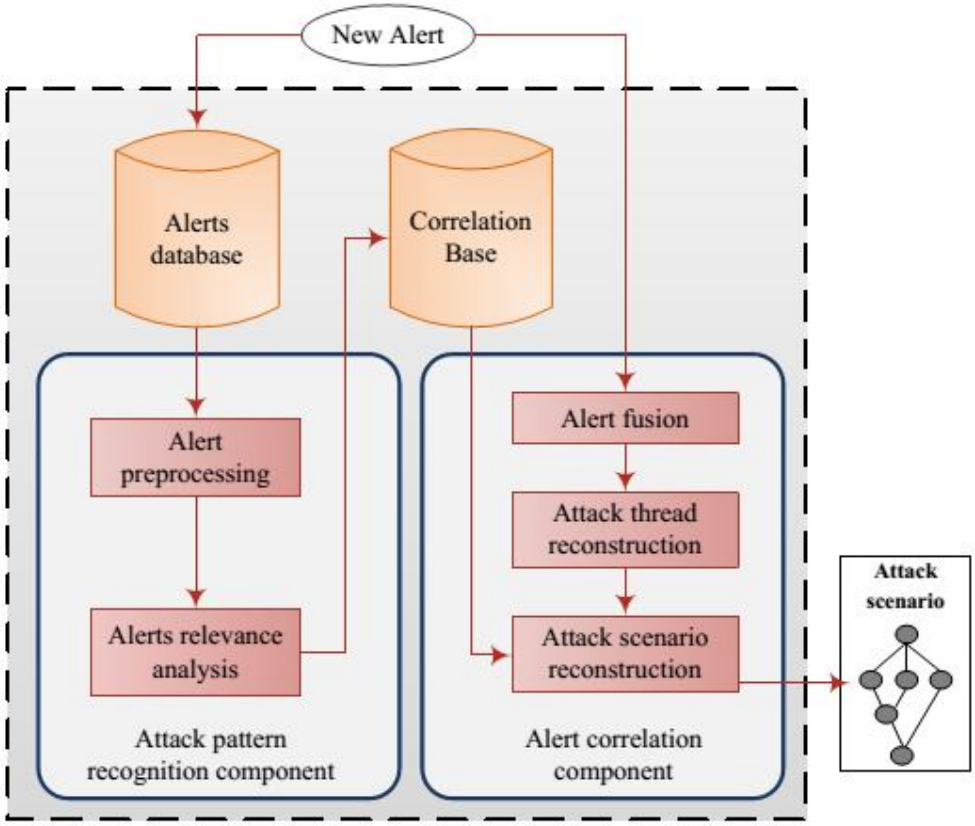

Figura 3.4 Sistema de correlação de alertas proposto em (KAVOUSI; AKBARI, 2012).

O módulo de pré-processamento tem a função de reduzir a quantidade de alertas redundantes. Normalmente esses alertas são de nível mais baixo, não representando grande ameaça isoladamente. Para isso, foi especificada uma janela de tempo, na qual pode auxiliar na redução desses alertas. O módulo de análise de relevância realiza a tarefa de inferir relações nos alertas baseado em restringir os atributos mais adequados. Para correlacionar os alertas é usada a teoria estatística de *Bayes*, na qual é construída uma árvore de com os integrantes que foram correlacionados entre si, gerando uma visão do ataque identificado. Os testes foram realizados também com o conjunto de alertas DARPA 2000.

No trabalho de (AI, 2013), é utilizado o algoritmo de associação *Apriori* para reduzir o volume de informação constituinte na base de dados de ataques, além de

gerar regras de associações a partir dos ataques na base de dados. De acordo com o tamanho da base, o algoritmo pode demorar a extrair as regras de associação. O trabalho propõe uma melhoria no algoritmo de modo que seja mais rápida a extração de regras. Os resultados obtidos mostraram que pode ser obtida uma melhora considerável quando os ataques redundantes são excluídos da base a medida que as regras são criadas. Além disso, a redução na base reduz o espaço necessário para armazenamento e tempo de análise.

Em (MARKAM; DUBEY, 2012), é proposto um modelo de IDS utilizando mineração de dados. As etapas para a detecção envolvem gerar regras de associação, classificação e, por fim, lógica *fuzzy*. O algoritmo utilizado para classificação é o *K-means* e o algoritmo para gerar as regras de associação foi o *Apriori*. A estratégia para detecção é definida na lógica *fuzzy*, na qual se baseia em detecção por *flags* do protocolo TCP. Os resultados são apresentados sob a perspectiva de desempenho de detecção. O IDS proposto apresentou resultados relativamente satisfatórios em comparação com outra técnica de detecção.

No trabalho de (MIRSHAHJAFARI; GHAVAMNIA, 2014), é proposto um modelo de rotular alertas que são similares entre si, utilizando extração de palavras para identificar palavras chaves que servem para rotular os alertas. Os testes foram realizados utilizando o Snort para gerar os alertas. Foram utilizados mil alertas e estes foram classificados, ou seja, rotulados manualmente. Os resultados mostraram que do total de alertas utilizados para o teste, 93% foram classificados de alguma forma e que 73% deles foram classificados corretamente.

Em (RAFTOPOULOS; DIMITROPOULOS, 2014), é introduzido um IDS que realiza a correlação de alertas para detectar hosts que estejam infectados por malwares. Uma comparação entre o IDS apresentado e o IDS Snort mostrou que o primeiro detectou 60% mais ameaças do que o segundo, além de ter gerado 15% menos falso-positivo. Os testes foram realizados em um ambiente acadêmico, o que culminou na detecção de 4358 hosts infectados nessa rede. O trabalho mostrou que máquinas que são infectadas tem mais probabilidade de serem alvos novamente de outros tipos de ataques, incluindo variações de famílias de malwares, bem como se tornam praticantes de ataques alvejando os hosts da própria rede.

No artigo apresentado por (XUEWEI et al., 2014), é proposta uma abordagem para minerar dados em busca de conhecimento casual automaticamente utilizando a propriedade de Markov. Os alertas são divididos em clusters baseados nos IP de origem e, dessa forma, é aplicado mineração de dados para pode obter os valores necessários para criar a matriz de Markov. Cada matriz representa um pedaço do conhecimento casual. Em seguida, é feita a fusão das sobreposições encontradas nas matrizes e, por fim, é obtido um padrão de ataque. Os testes foram realizados utilizando o conjunto de ataques DARPA 2000.

Em (GHASEMIGOL; GHAEMI-BAFGHI, 2013), é proposto um framework para correlação de alertas sem que seja necessário um conhecimento predefinido. É utilizado o conceito de entropia para cada alerta em busca de similaridades entre eles visando criar clusters de alertas. A representação utilizada para cada cluster encontrado é chamada de hyper-alert. Utilizando do conjunto de alertas DARPA 2000, os resultados obtidos mostraram uma eficiência de redução de alertas redundantes de 99,83% além de conseguir encontrar cenários de ataques com a obtenção do hyper-alert.

3.3 Visualização

Em (SHIRAVI; SHIRAVI; GHORBANI, 2012), é realizada uma revisão bastante detalhada sobre visualização em segurança de redes. O trabalho apresenta diversas formas de obtenção de informações e evidencia a importância da escolha correta da fonte desejada. Além disso, os autores ressaltam que em conjunto com a fonte de informação, a forma como esses dados serão dispostos, varia muito, podendo comprometer a correta interpretação dos dados.

Essa extensa revisão realizada pelos autores esclarece as variadas formas de visualização, além de apresentar ferramentas já existentes para esse propósito. Para determinado fim um tipo de visualização é mais adequado. É, portanto, abordada uma classificação das formas como pode ser apresentada uma visualização voltada para segurança de redes. Como exemplos de classificação, tem-se: monitorar servidores/*hosts*, chamadas internas de sistemas operacionais, atividades de portas, padrões de ataques e comportamento de roteamento. Para cada um dessas

classificações são apresentadas ferramentas de visualização que foram desenvolvidas para auxiliar na segurança dos sistemas computacionais.

No trabalho de (YANG et al., 2010), é apresentado uma metodologia de visualização de alertas gerados pelo sistema detector de intrusão *Snort*. Além disso, é realizada uma classificação dos alertas gerados, bem como uma associação entre esses alertas. A arquitetura proposta é bem simples, sendo esta, a coleta dos alertas, em seguida, a classificação e a geração de regras de associação, e consequentemente, analisando o comportamento do ataque e, concluindo com a visualização do cenário obtido.

Para geração das regras de associação foi utilizado o algoritmo *Apriori*. As figuras 3.5 e 3.6 apresentam os resultados obtidos no trabalho. A Figura 3.5 representa uma regra de associação obtida pelo algoritmo utilizado, enquanto a Figura 3.6 mostra um exemplo de como os resultados do trabalho são representados visualmente.

1) PORTSCAN -> SNMPattack
2) PORTSCAN -> SNMPattack -> ETSCANNMAP
3) PORTSCAN -> ETSCANNMAP

Figura 3.5 Regras de associação obtidas em (YANG et al., 2010).

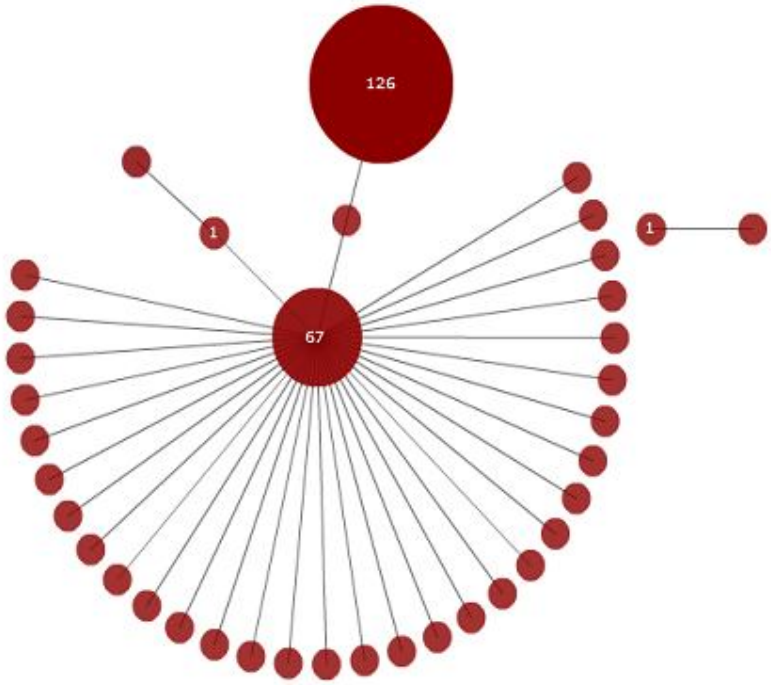

Figura 3.6 Visualização de uma topologia monitorada (YANG et al., 2010).

O trabalho de (ZHUO; NADJIN, 2012), apresenta uma ferramenta para visualização de *malwares*. Essa ferramenta auxilia no estudo do comportamento de variações de *malwares*, facilitando a compreensão dos mesmos. Como fonte de informação foi utilizado um capturador de pacotes (*sniffer*). Em seguida, foi identificado o rastro do *malware*, muito comum por possuir um padrão de comportamento. A Figura 3.7 ilustra como é a visualização da abordagem proposta no trabalho.

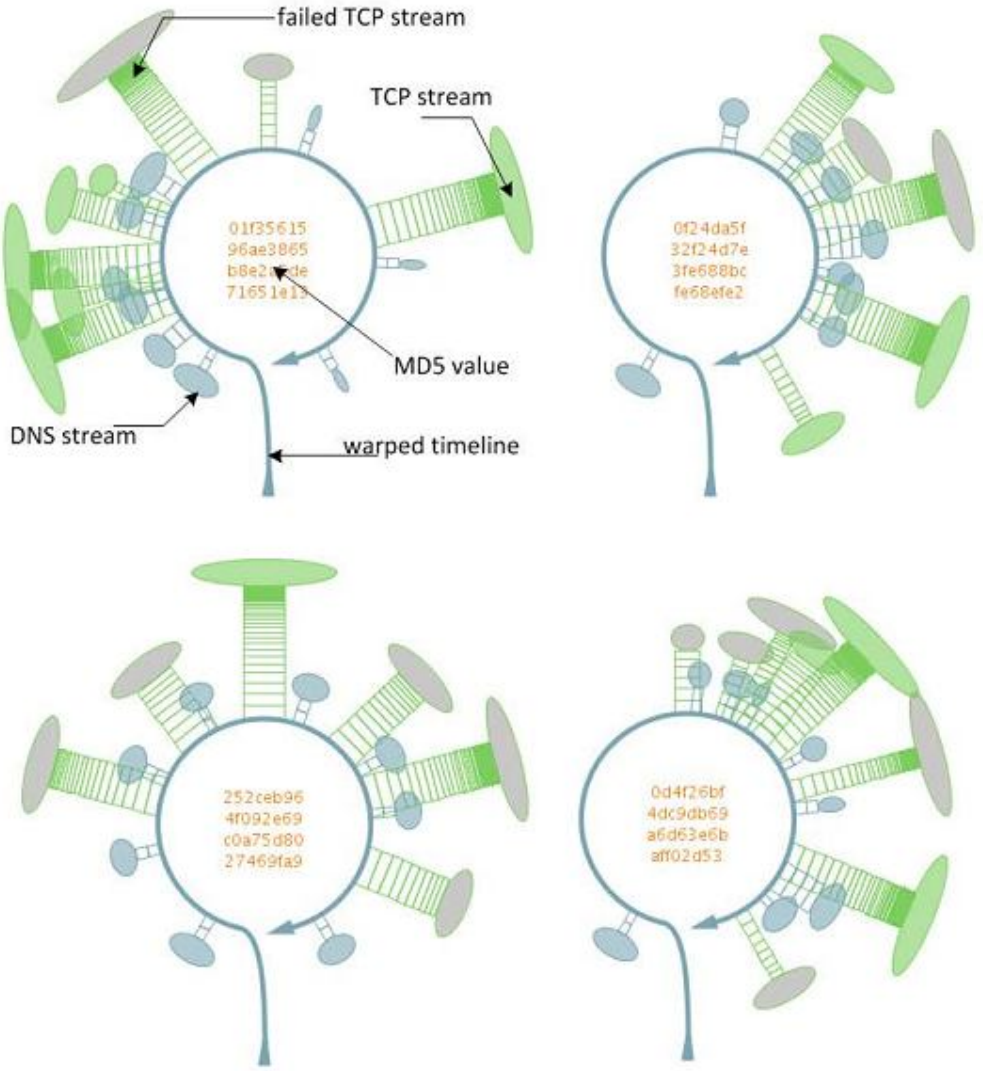

Figura 3.7 Visualização de variações de um mesmo malware (ZHUO; NADJIN, 2012).

Cada *malware* utilizado no trabalho possui um MD5 que o diferencia dos demais. A linha do tempo em que o *malware* começa a realizar suas atividades segue o sentido horário. A figura 3.7 auxilia na compreensão de como um mesmo *malware* com quatro variações possuem comportamentos diferentes em suas execuções.

Em (SHAHRESTANI et al., 2012), é discutida a importância de ferramentas visuais para observar o tráfego de eventos maliciosos, focando em *botnets*. Devido a

maioria dos IDS possuírem uma saída de relatório apenas em modo texto, a dificuldade em se extrair informações relevantes e em tempo hábil de um ataque de *botnets* é algo preocupante. Para auxiliar na melhoria do cenário de visualização de ataques envolvendo *botnets*, o trabalho apresenta uma abordagem de visualizações através de gráficos de dispersão, grafos direcionados e histogramas. A visualização através de grafos é utilizada para mostrar o tráfego entre os sistemas finais envolvidos, diferenciando através de cores o volume de tráfego. A visualização através de gráficos de dispersão é utilizada para avaliar as diferenças entre as *botnets*, sendo assim, utilizada para comparar umas com as outras em busca de variações ou padrões. A visualização por histogramas é utilizada para descobrir a existência de infecções de *botnets* na rede interna, monitorando a quantidade de saída dos sistemas que compõem a rede. Os resultados foram medidos de acordo com os usuários, sendo eles especialistas em segurança e os que não são especialistas. Os usuários avaliaram diversas características do sistema proposto e sua aceitabilidade foi grande de uma forma geral.

3.4 Considerações finais

Este capítulo apresentou os trabalhos relacionados à correlação de alertas gerados por eventos em redes de computadores e à visualização de informações. Esses artigos foram utilizados como base para o desenvolvimento da proposta deste trabalho, que será apresentado no capítulo seguinte.

Capítulo 4 – Metodologia

4.1 Considerações iniciais

Este capítulo descreve a metodologia realizada no trabalho desenvolvido. A metodologia consiste em utilizar alertas gerados por um sistema detector de intrusão e utilizar técnicas de mineração de dados para classificar, clusterizar e correlacionar os alertas visando obter cenários de ataques. Em seguida, são utilizados métodos de visualização para ilustrar os ataques e facilitar a compreensão desses cenários. A seção 4.2 apresenta os objetivos do trabalho e a estrutura do ambiente utilizada no desenvolvimento e teste deste trabalho. Na seção 4.3 é apresentada a arquitetura de desenvolvimento e funcionamento do trabalho. Por fim, na seção 4.4 são apresentadas as considerações finais.

4.2 Objetivos

O projeto proposto utiliza uma abordagem envolvendo mineração de dados para a correlação e, consequentemente, a criação de um cenário de ataque. Este cenário de ataque consiste em elencar ataques que foram realizados passo a passo em uma rede ou determinado *host*. Os métodos de mineração de dados utilizados na metodologia são algoritmos de classificação, clusterização e busca de itens frequentes. O algoritmo de classificação utilizado é o *AutoClass*. Esse é um algoritmo não-supervisionado que utiliza a teoria de Bayes para encontrar semelhança entre os parâmetros usados como entrada. O algoritmo de clusterização utilizado é o *K-means*. Esse também é um algoritmo não-supervisionado que utiliza elementos similares entre os alertas. Por fim, é realizada a correlação dos alertas baseado nos resultados obtidos pelos métodos anteriormente mencionados.

O ambiente ilustrado na figura 4.1 constitui de usuários comuns e uma máquina atacante, identificado pela cor vermelha na figura. Os ataques aplicados foram: varredura de rede, varredura de portas, força bruta e negação de serviço utilizando requisições ICMP. Os alvos dos ataques fazem parte do “Ambiente de

Usuários". A coleta de dados foi realizada com a ferramenta *fprobe* (FPROBE, 2015) que foi instalada no *gateway* da rede. Esses dados de tráfego foram exportados para um coletor que também faz parte do "Ambiente de Usuários". Neste mesmo coletor são executados os IDSs e os relatórios de cada um deles são armazenados de acordo com suas especificações. Em seguida, é possível realizar o tratamento dos alertas, realizando a classificação, clusterização e correlação dos resultados obtidos, finalizando com a visualização do cenário de ataque.

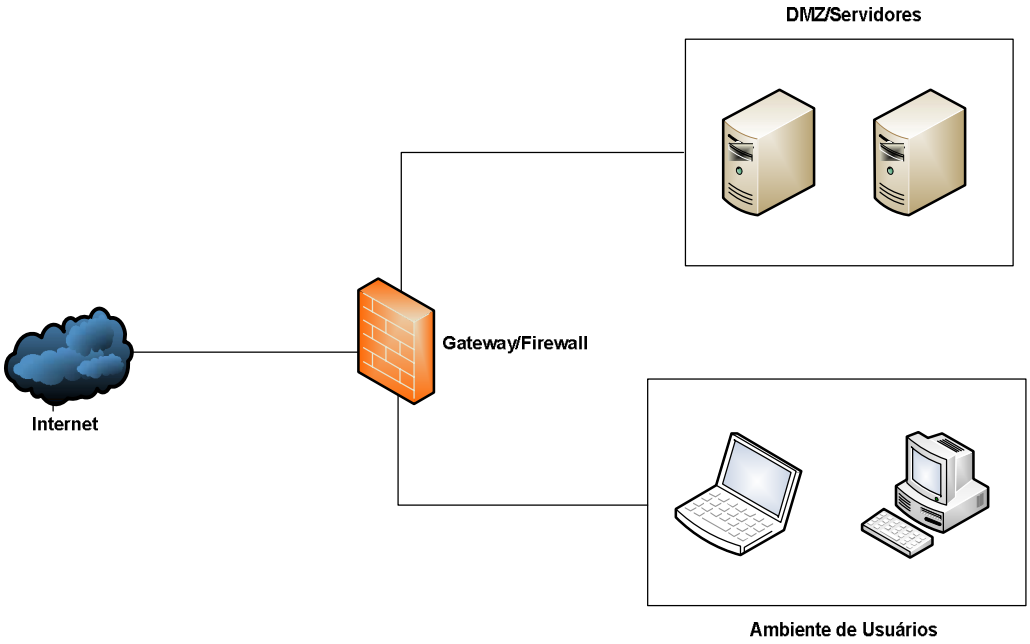

Figura 4.1 Estrutura do ambiente.

4.3 Arquitetura e funcionamento

A arquitetura proposta para o projeto pode ser visualizada na Figura 4.2:

- **Gateway/Firewall**: esta máquina é responsável por receber todo o tráfego da rede e exportá-lo para a máquina coletora.

- **Coletor e IDS**: esta máquina recebe o fluxo exportado pelo firewall e armazena esse fluxo no formato *Netflow* e *Biflow*. Além disso, ela é responsável por executar os IDSs e padronizá-los utilizando o IDMEF, armazenando os alertas gerados em uma base de dados.

- **Módulo de mineração de dados**: este módulo realiza a mineração de dados nos alertas. Primeiramente é realizada a classificação dos alertas e em seguida é utilizado o algoritmo de clusterização e, por fim, a correlação. O resultado é armazenado em uma nova base de dados que constitui em uma fonte de informação mais limpa e clara.

- **Módulo de visualização**: este módulo será responsável por gerar as imagens a partir dos resultados obtidos nos módulos anteriores.

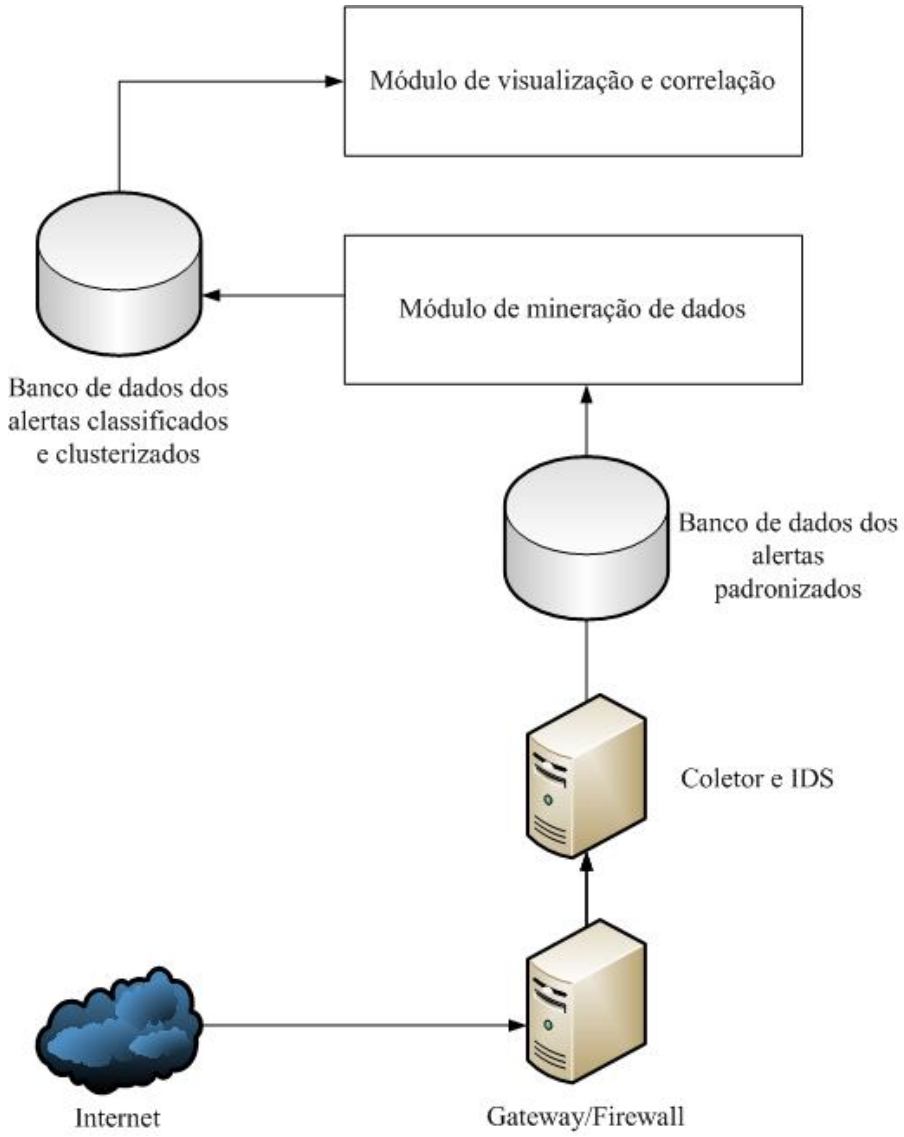

Figura 4.2 Arquitetura do sistema proposto.

O IDS utilizado para gerar alertas foi desenvolvido por uma ex-integrante do laboratório ACME! como parte de um projeto de mestrado e pode ser analisado em (BATISTA; CANSIAN, 2012) e (BATISTA; CANSIAN, 2011). Esse sistema, assim como outros IDSs, está em execução e já existe uma base de dados com grande quantidade de alertas. No entanto, para o presente projeto foi realizado novos ataques visando testar a metodologia em um ambiente controlado.

4.4 Método proposto

A metodologia proposta consiste em utilizar alertas gerados por um sistema detector de intrusão e, utilizar técnicas de mineração de dados para correlacionar os alertas visando obter cenários de ataques. A utilização de métodos de mineração de dados permite filtrar e analisar de forma mais precisa a alocação de cada alerta, seja sua classe de ataque, ao grupo a que pertence ou se existe um cenário de ataque mais elaborado, ou seja, com mais passos de execução. A abordagem não necessita de um treinamento prévio ou de um conhecimento específico para correlacionar os alertas. A utilização das técnicas de mineração de dados emite uma correlação entre os alertas que deverá ser avaliada.

Para cada tipo de IDS uma saída é utilizada visando o propósito de cada sensor. Devido às inúmeras formas e focos de detecção de intrusão, cada IDS realiza uma análise e emite um alerta em um formato particular, focado em sua metodologia de detecção. Sendo assim, na fase de pré-processamento uma normalização dos alertas é necessária, ou seja, padronizar os alertas obtidos pelos IDS. Essa normalização é feita utilizando o protocolo IDMEF. O protocolo transforma o arquivo de alerta gerado por um IDS qualquer em um arquivo no formato XML. Dessa forma, não importa qual IDS esteja sendo utilizado, pois toda a saída será normalizada e pode ser utilizada pela abordagem proposta sem necessidade de extração de informações dedicadas a cada tipo de IDS.

A abordagem proposta possui 3 etapas: classificação, clusterização e, por fim, correlação. A primeira etapa é realizada utilizando o *AutoClass* para fazer a

classificação dos alertas emitidos pelo IDS. A visualização é realizada a parte após todo o tratamento dos dados.

Primeiramente, o módulo de padronização faz parte de um projeto de iniciação científica de outro membro do grupo, onde todos os alertas gerados pelos IDSs são convertidos para o formato IDMEF. Este módulo já está em operação durante o desenvolvimento deste projeto e existe uma base de dados com diversos alertas padronizados.

A utilização de um algoritmo não supervisionado permite que a abordagem não necessite de uma base de conhecimento prévio para poder classificar os alertas, facilitando a utilização de todo e qualquer tipo de alerta que uma IDS possa gerar. Os parâmetros utilizados para classificar os alertas são descritos na tabela 4.1. Esses parâmetros escolhidos fornecem informações valiosas para que seja possível encontrar similaridades nos alertas do ponto de vista de ataques observados na rede monitorada.

Tabela 4.1 Descritores utilizados no *AutoClass*.

Nome	**Descrição**
qnt_dst	Quantidade de destinos distintos.
qnt_port	Quantidade de portas distintas.
src_octets	Quantidade de bytes enviados.
src_pkts	Quantidade de pacotes enviados.
dst_octets	Quantidade de bytes recebidos.
dst_pkts	Quantidade de pacotes recebidos.
num_con	Quantidade de fluxos/conexões realizadas.
nullrate	Taxa dos fluxos que não houve resposta do destino.
timerate	Taxa de conexões que não são requisições HTTP ou HTTPs que

	tiveram menos de 15 segundos de duração.
synrate	Taxa dos fluxos que tiveram a *flag* SYN do protocolo TCP ativada.
rstrate	Taxa dos fluxos que tiveram a *flag* RST do protocolo TCP ativada.
distinct_dstaddrs	Taxa de IPs distintos de destino acessados.
distinct_dstports	Taxa de portas distintas de destino acessadas.
distinct_dstwkps	Taxa de portas distintas do tipo *Well Known Ports* (WKP) de destino acessadas.
distinct_srcwkps	Taxa de portas distintas do tipo *Well Known Ports* (WKP) de origem acessadas.

O conteúdo de cada alerta gerado possui informações que são bastante específicas de cada ataque. Sendo assim, as informações contidas nos alertas são relacionadas no *AutoClass* e, como resultado, o algoritmo emite a probabilidade de cada um dos alertas estar contido em determinada classe ou em mais de uma, dependendo de suas características.

A segunda etapa é realizada utilizando o algoritmo *K-means* para gerar o agrupamento entre os alertas. Um algoritmo de clusterização permite reunir alertas que possuam características similares entre si de forma que seja mais provável que os alertas que compõem um determinado cluster estejam mais suscetíveis a fazerem parte de um cenário de ataque maior (PENG et al., 2008). A busca por similaridades nos algoritmos de classificação e de clusterização possuem diferenças sutis. Na classificação, a busca por similaridades usa como parâmetros informações quantitativas que compõem o alerta, como por exemplo, quantidade de destinos, quantidade de bytes enviados, quantidade de portas alvo, entre outros. Na clusterização, os parâmetros utilizados são as informações dos agentes participantes daquele alerta, por exemplo: endereço IP de origem e destino, porta de destino, protocolo envolvidos, entre outros. Dessa forma, a busca por padrões e mineração

dos dados se torna mais aprofundada e refinada, possibilitando que sejam encontrados cenários de ataques baseados nos resultados obtidos dessas duas abordagens.

Apenas com o resultado obtido na clusterização, é possível dizer que os alertas que foram agrupados em um mesmo cluster, ou seja, que possuem similaridades entre si são mais prováveis de fazerem parte de um cenário de ataque, devido aos parâmetros utilizados para realizar o agrupamento.

A terceira etapa é a correlação dos alertas tendo como base os resultados obtidos das duas abordagens anteriores. Uma vez que os alertas estão classificados e clusterizados é estabelecida uma relação entre os dois resultados. Ambos os métodos são baseados em probabilidade e, portanto, alocam os alertas de acordo com as similaridades encontradas nos parâmetros usados para sua execução. Dessa forma, existem alertas com origens e destinos diferentes em um mesmo cluster e em uma mesma classe. Diante disso, é necessário então extrair a correlação entre os alertas que estão nos clusters e nas classes.

Partindo do princípio de que os alertas que estão no mesmo cluster são similares entre si de acordo com o endereço de origem e destino e a porta de destino, os alertas que constituem cada cluster serão usados como referência para o início da correlação. De acordo com os parâmetros utilizados na clusterização, pode-se observar que existirá alguma diversidade dentro de cada cluster. Isso ocorre devido ao fato de que apesar de algumas variações de algum parâmetro, determinado alerta ainda possui alguma semelhança com os demais ali alocados. Por exemplo, alertas que possuam endereços IP de origem e porta de origem iguais, mas que variam os endereços de IP de destino, podem indicar uma varredura de hosts em uma rede.

Dessa forma, usando como base os clusters obtidos pelo *K-means*, são destacados os alertas que possuem o endereço de origem e o de destino iguais. Entretanto, para estabelecer um ponto de partida inicial e mais direcionado, pode-se usar um algoritmo de busca por itens frequentes que realiza a associação entre os endereços de origem e destino. Para isso, foi utilizado o algoritmo *Apriori* que apresenta os participantes mais ativos dentre os alertas avaliados e, dessa forma, fornece informação relevante para o início da busca. A partir disso, os alertas

existentes com essas características são buscados nas classes de modo a avaliarem se eles aparecem em todas as classes de ataques encontradas pelo *AutoClass*, pelo menos uma vez. É realizada a soma para cada alerta encontrado nas classes de ataque e que possui as características de um ataque específico. Dessa forma, é possível obter uma medida que, baseada no total de alertas existentes dentro de um cluster X, saber se o nível de ataque com as características selecionadas tem uma alta relevância ou não.

Após essa etapa os alertas são organizados de acordo com o tempo em que foram detectados e o cenário de ataque é encontrado. Dependendo da quantidade de alertas existentes nas classes de ataques o nível do ataque é mais complexo e possui mais passos para obter seu objetivo. Dessa forma, pode-se afirmar que uma quantidade de alertas que possua um nível baixo não necessariamente tenha obtido êxito no ataque, e muito provavelmente se tratam de alertas irrelevantes para o analista. A tabela 4.2 ilustra um exemplo de relação entre os clusters e as classes de ataques.

Tabela 4.2 Exemplo de relação entre os clusters e a classe de ataque.

	Cluster A	**Cluster B**	**Cluster C**	**Total**
Scan	1	2	2	5
SSH	4	2	1	7
DoS	1	2	0	3
WEB	0	0	1	1
Total	6	6	4	16

A tabela 4.2 apresenta um exemplo de resultado da mineração de dados de um conjunto de alertas que foram classificados e agrupados de modo a formarem uma relação entre si. A quantidade de alertas existentes em um cluster está dividido de acordo com sua classificação de ataque. A partir dessa relação é então executado o algoritmo de correlação que buscará por origens e destinos iguais dentro de um determinado cluster, de acordo com a diferença entre as classes.

O nível de relevância de uma cenário de ataque será medido de acordo com a ocorrência de um ataque com a mesma origem e destino em pelo menos uma das classes de ataques. A equação que expressa esse nível é descrita em (1).

$$nível = n/x \quad (1)$$

Onde n é a quantidade de alertas com a mesma origem e destino que existem em cada uma das classes de ataques e x é o total de alertas existentes no cluster utilizado. Quanto maior for esse valor maior será a quantidade de ataques distintos que o atacante realizou contra um alvo, evidenciando um cenário mais completo e detalhado do objetivo do atacante.

Por exemplo, baseando-se no cluster A da tabela 4.2 pode-se observar a existência de 4 alertas com o mesmo endereço de origem e o mesmo endereço de destino. O nível de relevância obtido é de 0,67, indicando um alto nível em comparação com a ocorrência dos demais alertas.

4.5 Considerações finais

Este capítulo fez uma descrição do ambiente utilizado e do seu funcionamento, bem como a metodologia abordada para a realização desta dissertação. Além disso, foi apresentada uma descrição dos dados utilizados nos métodos de mineração de dados do trabalho. O próximo capítulo apresenta os resultados obtidos.

Capítulo 5 – Resultados

Neste capítulo são apresentados os resultados obtidos. Na seção 5.1 são apresentados os tipos de ataques realizados no ambiente de testes. A seção 5.2 apresenta os resultados gerais obtidos com esta metodologia. Além disso, são apresentadas quatro subseções com resultados específicos obtidos pelo uso do *AutoClass*, do *K-means*, da correlação e da visualização, de acordo com o previsto para os objetivos desse projeto. Na seção 5.3. são apresentados os resultados obtidos utilizando o conjunto de dados de ataque DARPA 2000 como fonte de informação para a metodologia proposta. Por fim, na seção 5.4 são feitas algumas considerações finais para o capítulo.

5.1 Descrição dos ataques e testes

Os ataques realizados possuem uma variedade e ordem pré-estabelecida. Cada ataque foi executado em intervalos de tempo curtos, de 180 segundos entre um ataque e outro. Os ataques são ordenados de acordo com uma lógica de um cenário de ataque visando obter algum nível de acesso ou apenas causar a ruptura do serviço alvo. Ao todo foram 4 ataques distintos: varredura de rede, varredura de portas, ataque de força bruta em servidor SSH (*Secure Shell*) e ataque de negação de serviço utilizando o protocolo ICMP. Essa sequência de ataques foi repetida 6 vezes e distribuída em 6 dias. Ou seja, cada sequência de ataque foi realizada em um dia distinto. O intervalo de tempo entre uma sequência e outra foi um valor aleatório, tendo como propósito parecer o mais próximo possível de um ambiente real.

Ao longo dos 6 dias de testes foram obtidos um total de 38 alertas referentes aos ataques realizados no ambiente de testes. Os ataques realizados ao longo dos 6 dias de testes são idênticos e compreendem testes de validação da metodologia. A tabela 5.1 apresenta a quantidade de ataques realizados nos 6 dias de teste e a quantidade de alertas gerados por esses ataques.

Tabela 5.1. Quantidade de ataques realizados e alertas obtidos nos 6 dias de testes.

Tipos de ataques	Quantidade de ataques realizados	Quantidade de alertas gerados
Varredura de rede	6	12
Varredura de portas	12	10
Força bruta SSH	6	6
DoS (ICMP)	6	10

A maioria dos IDS necessita de configurações de limiares para realizar as detecções. Como se sabe, esse procedimento é comumente necessário quando se trata de detecção por anomalia. Essa prática pode ocasionar a ocorrência de alertas equivocados, conhecidos como falso-positivo. Evidentemente, não é desejável que existam muitos alertas falso-positivo. No entanto, a metodologia aqui proposta auxilia na redução ou eliminação dos alertas relevantes, permitindo que seja possível evidenciar alertas que não se correlacionam entre si, caracterizando um alerta falso ou um ataque sem risco significativo ao ambiente.

Para ilustrar o cenário utilizado nos testes, a figura 5.1 apresenta como o ambiente foi estruturado. A máquina com a cor vermelha é o atacante e seu alvo será a rede protegida pelo *firewall* apresentado na figura.

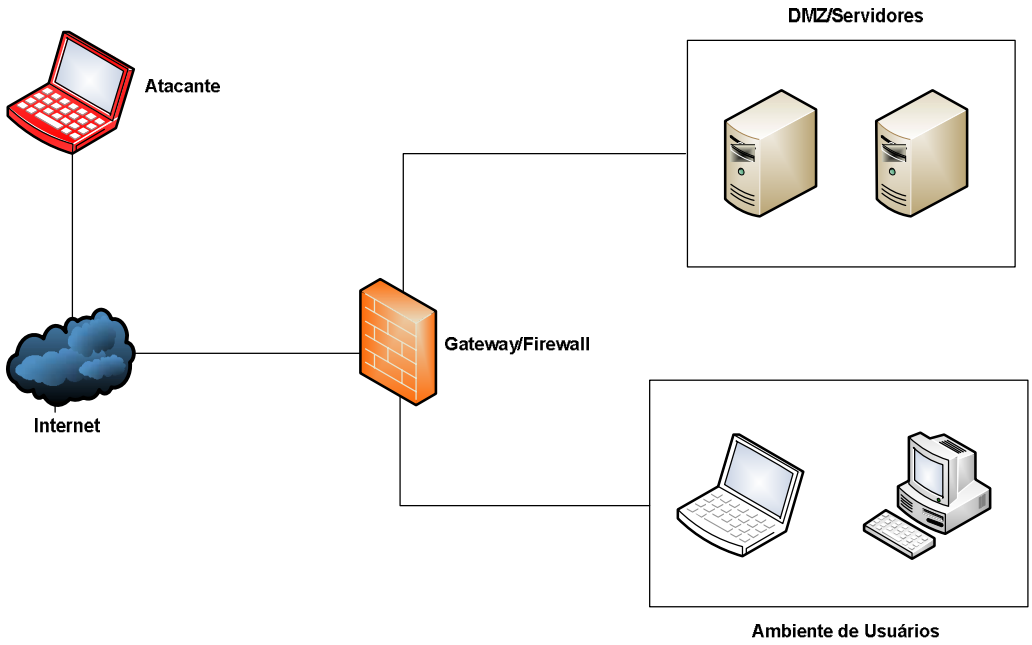

Figura 5.1 Estrutura do ambiente de teste.

5.2 Resultados gerais de validação

Dentre os resultados gerais obtidos pela análise realizada é possível separar cada etapa da metodologia em resultados específicos obtidos por cada método empregado. As duas subseções 5.2.1, 5.2.2 e 5.2.3 apresentam os resultados obtidos especificamente com a utilização do algoritmo de classificação *AutoClass*, e em seguida com o algoritmo de clusterização *K-means*, finalizando com a correlação dos alertas. Por fim, a subseção 5.2.4 englobará a fusão desses dois resultados juntamente com a visualização dos alertas correlacionados.

Os ataques realizados são ataques idênticos, e foram assim adotados para fins de validação. Dessa forma, os resultados obtidos ao longo dos 6 dias de testes possuíram os mesmos resultados entre si. Sendo assim, cuidando-se para não apresentar resultados redundantes, foi utilizado apenas um dia de testes para expressar os resultados obtidos pela metodologia. Além disso, considere para todo este capítulo que os endereços IP foram sanitizados.

5.2.1 Resultados do algoritmo de classificação *AutoClass*

Os atributos utilizados para a execução do algoritmo *AutoClass* foram aqueles apresentados anteriormente na tabela 4.1. Conforme discutido anteriormente, o *AutoClass* utiliza os parâmetros para buscar semelhanças entre os alertas e os agrupa em classes, sendo possível distinguir cada alerta utilizando-se o parâmetro do tipo de ataque relacionado. Apesar do algoritmo não ser supervisionado, o primeiro dia de testes foi utilizado como parâmetro para classificação dos demais dias. Usando como referência as ocorrências do primeiro dia de testes o *AutoClass* pode aprender com o comportamento da rede, e assim classificar os alertas dos demais dias. Dessa forma, o *AutoClass* separou os alertas de cada um dos dias em 10 classes distintas.

A Tabela 5.2 apresenta a classificação obtida pelo *AutoClass* para todos os 6 dias de teste. A tabela concentra-se nos alertas que são relativos aos testes realizados e, portanto, apenas são apresentadas as classes que contêm os alertas dos ataques, prezando por uma melhor visualização do resultado.

Tabela 5.2 Classificação dos alertas utilizando o *AutoClass* para os 6 dias de testes.

	Scan	**Força bruta**	**DoS**
Classe 3 (Força Bruta)	0	**6**	0
Classe 5 (Scan)	**21**	0	0
Classe 6 (DoS)	0	0	**10**
Classe 9	**1**	0	0

De acordo com a tabela, é possível observar que o *AutoClass* pode classificar esses ataques em grupos distintos. Assim é possível diferenciar os alertas em seus respectivos tipos de ataque. Entretanto, apenas um alerta foi classificado em um grupo diferente dos demais. O alerta em questão era relativo a um ataque de

varredura de portas e foi atribuído à classe 9. Isso pode ser explicado devido ao fato do algoritmo ser baseado em lógica *fuzzy*, ou seja, existe uma probabilidade de que um mesmo alerta pertença a classes distintas, de modo que, neste caso, o alerta possuía uma maior probabilidade de pertencer à classe 9 do que à classe 5.

5.2.2 Resultados do algoritmo *K-means*

Após essa fase de classificação é realizada a etapa de clusterização. Os parâmetros utilizados para gerar os clusters são os endereços de origem, de destino e portas de destino. A clusterização é feita utilizando o algoritmo *K-means* para encontrar as similaridades entre os alertas.

Para a criação dos clusters, o algoritmo necessita que seja estabelecida uma quantidade de clusters arbitrariamente. Testes realizados com diferentes quantidades de clusters, variando entre 5 e 10 clusters mostraram que os resultados eram os mesmos em relação ao agrupamento dos alertas de testes. Conforme discutido anteriormente no capítulo 4, o *K-means* tende a agrupar os alertas que fazem parte de um mesmo cenário de ataque. Essa afirmação feita por (PENG et al., 2008) se mostrou verdadeira diante dos resultados obtidos. Devido a variedade e a quantidade de alertas e, ainda, pela metodologia proposta, optou-se por utilizar 5 clusters para a análise realizada neste trabalho. A variação da quantidade de cluster implica na qualidade dos cenários que serão encontrados. Quanto menos clusters forem utilizados, maiores serão as chances de existir cenários mais completos. Ao se testar outra quantidade de clusters, como 10, 15 ou 20, observou-se que os alertas relativos aos ataques tendiam a continuar a ser agrupados nos mesmos clusters, porém, alguns alertas eram alocados em cluster diferentes, podendo levar a cenários isolados e incompletos. Portanto, a quantidade de 5 clusters foi utilizada para proporcionar cenários maiores e menos discrepantes.

Destacando apenas um dia de teste de validação, é possível observar por meio da Figura 5.2 a distribuição de alertas relativos aos ataques realizados dentro do cluster ao qual eles foram alocados.

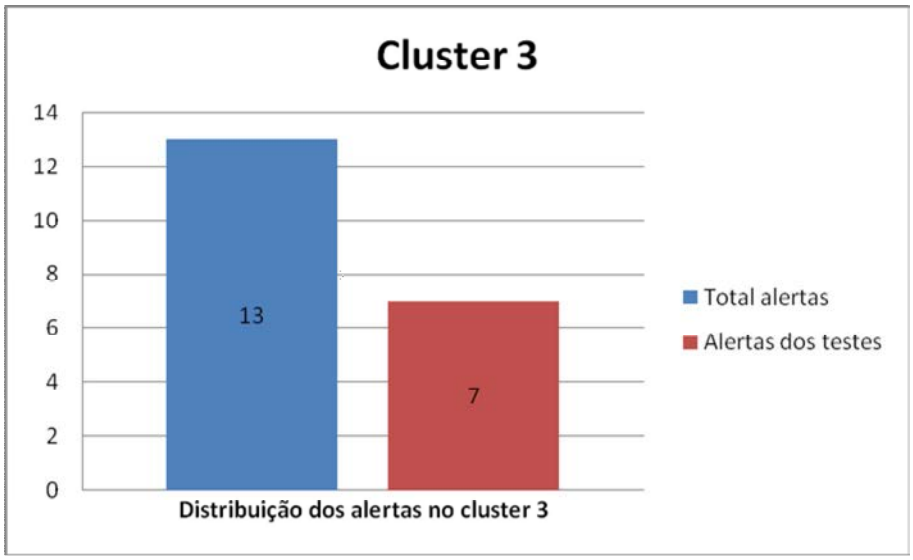

Figura 5.2. Distribuição dos alertas contidos no cluster 3.

De acordo com a Figura 5.2 pode-se observar que dentre os 13 alertas que foram alocados no cluster 3, 7 deles pertencem ao cenário de testes realizados para o dia em questão. Além disso, para este dia de teste, foram obtidos 7 alertas ao todo e todos eles foram agrupados no mesmo cluster. Portanto, todos eles foram corretamente agrupados em um mesmo cluster, corroborando com a hipótese de que alertas de um mesmo cenário tendem a se agrupar em um mesmo cluster.

Diante do fato de que em todos os 6 dias de testes foram utilizados os mesmos tipos de ataques, o algoritmo *K-means* agrupou todos esses ataques em um mesmo cluster. Para a análise da metodologia será avaliado apenas um dia por vez, separadamente, para que não ocorra a correlação redundante em uma única análise. A Tabela 5.3 apresentada na subseção 5.2.3 ilustra a relação encontrada entre a clusterização e a classificação, ambos importantes para todo o processo de correlação dos alertas.

5.2.3 Correlação

A última etapa tem a função de correlacionar os alertas utilizando os resultados obtidos dos algoritmos de mineração de dados. Para isso é necessário encontrar uma relação inicial entre as classes de ataques e os grupos de ataques a que os alertas obtidos pertencem. A relação de alertas contidos em cada cluster com a classe de ataque pode ser avaliada na figura 5.3. De posse da relação inicial entre classe e cluster, a correlação é aplicada de modo a encontrar alertas com a mesma origem e destino que pertencem a um mesmo cluster, mas que estejam distribuídos entre as diferentes classes de ataque. Dessa forma, é possível afirmar a existência de ataques distintos entre atacantes e alvos de modo a criar um cenário passo a passo do ataque.

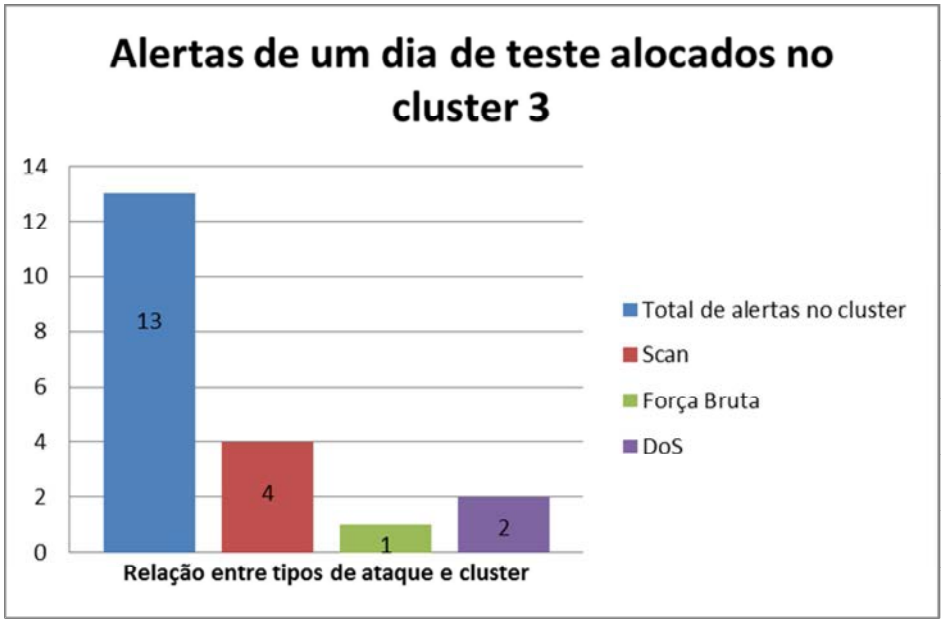

Figura 5.3. Relação dos alertas de um dia de teste de acordo com seu tipo de ataque e o cluster ao qual foram alocados.

A relação obtida na figura 5.3 mostra que a metodologia é eficaz no que se diz respeito à separação dos tipos de alertas e agrupamento por similaridades. Por exemplo, dentre os 13 alertas que o cluster 3 possui, 7 deles representam o cenário de ataque utilizado como teste. Ainda, observando os tipos de ataques dos 7 alertas de

teste, tendo o resultado a classificação do *AutoClass*, pode-se observar que, de fato, o cluster contém um cenário de ataque passo a passo. Sendo assim, a correlação é realizada buscando em um cluster por endereços de origem e destino que estejam em classes diferentes. Ao identificar a ocorrência de um alerta que possua esses endereços, o algoritmo procura por outra ocorrência em uma classe diferente. Se encontrar, passa para outra classe, e assim por diante. Ao terminar a busca em todas as classes, é possível observar a existência de uma cadeia de alertas que possuam características similares e que compreendem um cenario de ataque.

Para que seja possível estabelecer um ponto de partida para realizar a busca por endereços de origem e destino, pode-se utilizar algoritmos que encontram itens frequentes em um conjunto de dados. Para isso, utilou-se o algoritmo *Apriori*, que encontra associações entre os itens, relacionando-os com sua frequência no conjunto de dados. Dessa forma, é possível estabelecer um ponto de partida para as buscas, devido ao resultado obtido pelo *Apriori*, uma vez que será evidenciado quais são os participantes mais frequentes dentre os alertas analisados. A tabela 5.3 apresenta o resultado do algoritmo *Apriori* em relação ao conjunto de dados utilizados para o dia de testes avaliado.

Tabela 5.3. Resultado do algoritmo *Apriori* para o dia de teste analisado.

Origem	**Quantidade de alertas de origem**	**Destino**	**Quantidade de alertas de destino**
444.79.180.124	7	111.222.333.A	4
444.79.180.124	7	111.222.333.B	1
444.79.180.124	7	111.222.333.C	1
444.79.180.124	7	111.222.333.D	1

O resultado do algoritmo *Apriori* apresenta um início para a busca por correlação, uma vez que seu resultado é baseado na frequencia em que esses elementos, no caso endereços de origem e destino, aparecem no conjunto de alertas

de um dia de teste. O resultado do algoritmo mostra ainda a quantidade de alertas existentes para cada destino de acordo com a origem, podendo ser utilizada essa informação para mostrar que em determinado destino houve maior dedicação no ataque.

Para atribuir um nível de correlação entre os diversos alertas existentes, uma métrica é utilizada para obter um valor que pode ser utilizado como direção de cenários de ataques mais completos. Ao se avaliar a ocorrência de alertas em cada classe de ataque com as determinadas características, é feita uma soma desse valor e dividido pela quantidade de alertas existentes no respectivo cluster. O nível do cenário será avaliado de acordo com o valor obtido. Os ataques de testes foram realizados a partir de uma única origem para vários destinos, tornando-se mais específicos à medida em que o atacante encontrava um alvo vulnerável. Dessa forma, foram obtidos alertas que compreendiam diversos clusters. Entretanto, em apenas uma categoria foi dado continuidade, como é o caso do cluster 3. O nível de correlação obtido ao se avaliar o cluster 3 é maior do que nos demais, devido a existência de alertas que pertencem aos três tipos de ataques que foram detectados naquele dia.

Outros níveis de correlação também são úteis para avaliar o estado da rede monitorada, mas quanto maior for o nível de correlação encontrado, maior será a chance de um ataque ser mais específico e mais preocupante para a rede.

5.2.4 Visualização

Os resultados obtidos com a utilização dos métodos anteriores possibilitam que seja realizada uma correlação dos alertas. A variação dos tipos de ataques é muito grande e, normalmente, existe uma lógica entre o uso de cada tipo. Sendo assim, separar os tipos de ataques é um primeiro passo para identificar um cenário de ataque. Além disso, ainda existe o problema da quantidade de alertas existentes que precisam ser analisados. Diminuir a quantidade, ressaltando os alertas mais expressivos, permite que a análise seja menos dispendiosa. Ainda assim, é possível extrair uma relação entre os alertas, permitindo que a análise seja ainda mais focada e

pontual. A junção dessas abordagens consiste em uma valiosa informação. No entanto, ainda é preciso juntar os benefícios de cada método para gerar uma fonte ainda mais rica em informação.

A figura 5.4 ilustra a visualização gerada a partir da correlação dos alertas e da associação, obtidos para o mesmo dia de teste utilizado anteriormente, através dos métodos abordados. Na figura 5.4 é possível observar o cenário de ataque, baseado na correlação obtida, tendo como referencial a origem, destacada em vermelho, e quais foram os alvos, destacados em azul, simbolizados por uma aresta direcionada.

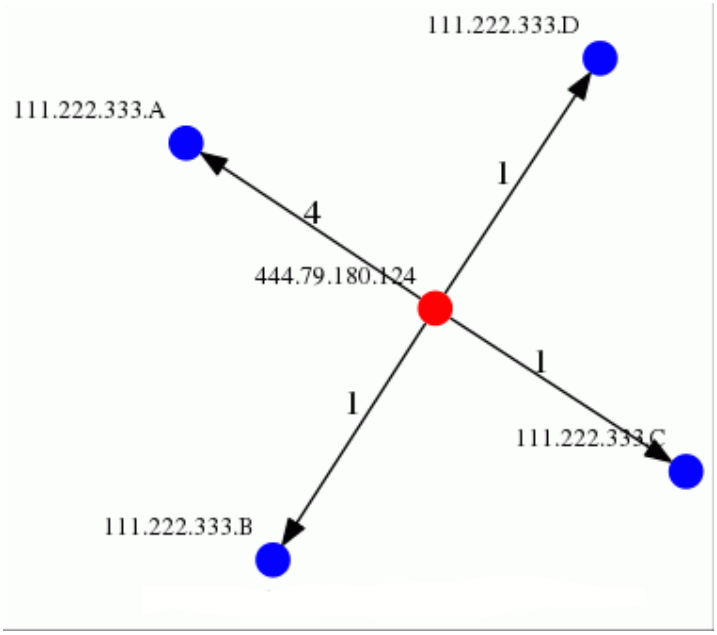

Figura 5.4. Cenário de ataque para um dia de teste de validação.

Os valores de cada aresta indicam a quantidade de alertas existentes entre a origem e o destino. Diante deste grafo, pode-se observar que o atacante primeiramente realizou uma varredura na rede alvo em busca de máquinas ativas. Em seguida, ao encontrar um alvo com algum serviço aberto realizou outras tentativas de acesso, podendo ser evidenciada pela aresta que tem sua origem no atacante até a vítima 111.222.333.A.

A obtenção desse grafo é baseada na correlação dos alertas de segurança. A ocorrência de alertas em um mesmo cluster permite que seja traçado um cenário de

ataque como o apresentado na figura 5.4. O foco dessa abordagem de visualização tem como intuito mostrar a existência de um atacante e seus alvos através da quantidade de alertas que existem entre os participantes dos ataques. A linha temporal pode ser traçada após avaliar a figura do cenário de ataque, onde o analista poderá recuperar todos os detalhes dos alertas do cenário com maior facilidade e precisão, como por exemplo, quais os serviços que foram os alvos, podendo posteriormente, analisar os *logs* da vítima em busca de tentativas de invasão de forma mais direcionada.

Em se tratando especificamente deste cenário de validação, ao observar que o cenário de ataque com maior nível de correlação teve como origem o endereço 444.79.180.124 e teve como destino o endereço 111.222.333.A. Dessa forma, é possível analisar os 4 alertas gerados pela atividade que o atacante realizou em seu alvo. Dentre os ataques estão varredura de portas, tentativa de força bruta na porta 22 e negação de serviço utilizando o protocolo ICMP.

5.3 Resultados finais

Os resultados apresentados aqui nesta seção tem como base de alertas de ataques o LLS_DDOS_1.0 que é um cenário de ataque que faz parte do conjunto de ataques contidos no DARPA 2000. Esse cenário de ataque é descrito em (DARPA, 2000) e possui detalhes de como o atacante procedeu no ataque e quais foram os alvos. A intenção de apresentar esses resultados é de discutir a metodologia de correlação e visualização de forma mais aprofundada e utilizando um novo conjunto de dados para análise.

Os dados providos são na forma de um arquivo de saída do *tcpdump*. Para transformar esse arquivo em alertas, os quais são a verdadeira fonte de informação para esta metodologia, utilizou-se o Snort para gerar os alertas necessários. Ao término da execução do Snort, foram obtidos 5391 alertas referentes ao cenário de ataque.

Utilizando o Snort, eliminou-se uma etapa da metodologia, que corresponde a classificação realizada pelo *AutoClass*. O Snort consegue identificar o tipo de ataque

que foi realizado por intermédio de sua base de assinaturas e de sua metodologia de detecção. Sendo assim, é realizada a etapa seguinte que corresponde à clusterização dos alertas.

Devido ao grande número de alertas obtidos, a figura 5.5 ilustra apenas os clusters que contem os alertas relativos ao ataque em foco.

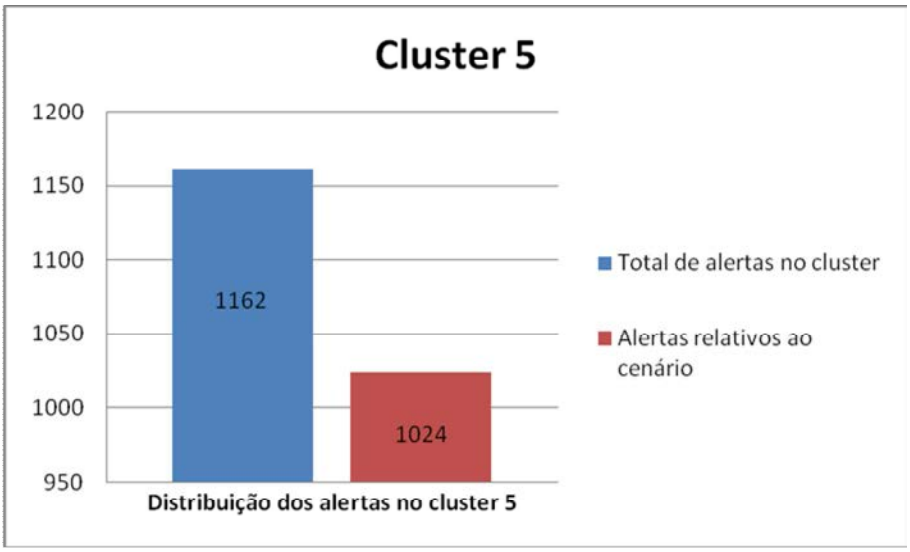

Figura 5.5. Cluster 5 contendo os alertas referentes ao cenário de ataque.

A figura 5.5 apresenta a quantidade de alertas que foram classificados como pertencentes ao cluster 5. Ao avaliar esse cluster, pode-se observar que os ataques que nele estão contidos são referentes a ataques de varredura de hosts, evidenciados devido a grande variação de endereços IPs de destino partindo de uma mesma origem, no caso o endereço 555.77.162.213. A variação de portas mostra que, primeiramente, o atacante realizou uma varredura utilizando requisições ICMP em busca de hosts ativos nas redes e, em seguida, dentre os hosts ativos, foi buscado quais destes hosts possuíam a porta 111 aberta.

Com o resultado obtido com o cluster 5, pode-se realizar a correlação em busca de alertas mais específicos, como por exemplo, quais alvos estavam ativos e que tiveram maior foco. Para isso, o algoritmo *Apriori* foi utilizado para encontrar os

endereços mais freqüentes dentro deste cluster. A tabela 5.4 apresenta as associações encontradas pelo *Apriori*.

Tabela 5.4. Resultado do algoritmo *Apriori* para o cluster 5.

Origem	Quantidade de alertas de origem	Destino	Quantidade de alertas de destino
555.77.162.213	1024	333.16.113.148	26
555.77.162.213	1024	333.16.115.87	26
555.77.162.213	1024	333.16.112.105	26
555.77.162.213	1024	333.16.115.20	16
555.77.162.213	1024	333.16.112.10	12
555.77.162.213	1024	333.16.113.105	26
555.77.162.213	1024	333.16.112.194	26
555.77.162.213	1024	333.16.112.50	16

A tabela 5.4 apresenta os itens mais freqüentes no cluster 5 e, como pode ser observado, o endereço de origem 555.77.16.213 tem como destino diversos endereços distintos, caracterizando uma varredura por hosts ativos.

De posse dessas informações, pode-se gerar um grafo que exemplifica o que aconteceu nesta rede e avaliar melhor sem a necessidade de buscar pelas informações no grande volume de alertas. A figura 5.6 ilustra como fica esse cenário de ataque.

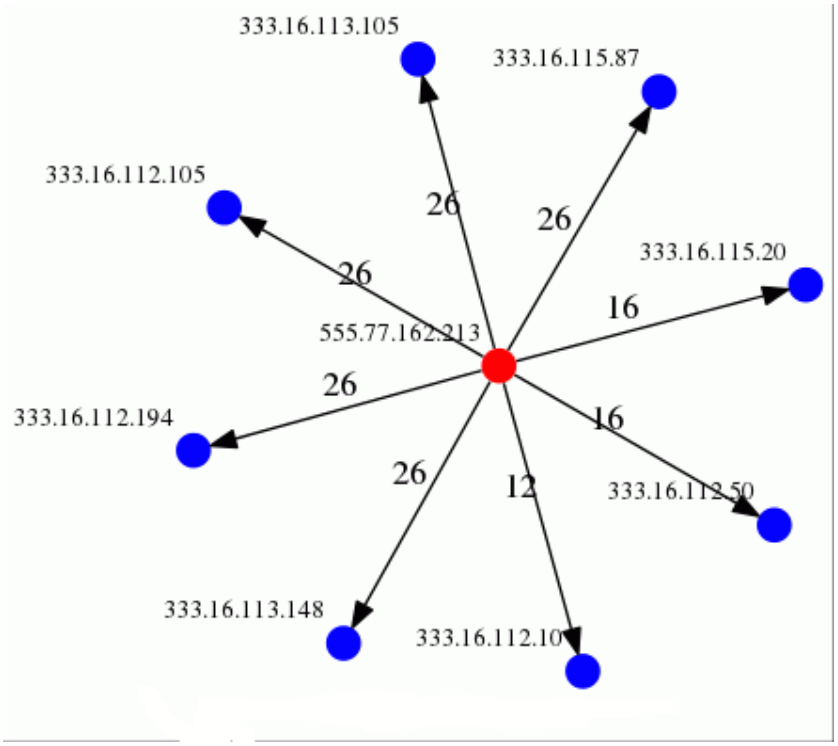

Figura 5.6. Cenário de ataque baseado nos alertas do cluster 5.

O grafo apresentado na figura 5.6 ilustra o resultado da correlação dos alertas baseado nas ocorrências dos endereços mais freqüentes dentre todos os alertas existentes no cluster 5. É possível observar que dentre os alertas que foram agrupados no cluster 5, existem vários outros que possuem outros alvos, além daqueles apresentados na tabela 5.6. No entanto, devido ao fato de que a metodologia faz uso da correlação baseada na ocorrência de alertas, a quantidade de vezes que um endereço de destino aparece não possui o nível de correlação suficiente para que seja caracterizado como um cenário de risco. Além disso, devida a pouca quantidade de alertas com essas características, o algoritmo *Apriori* também não encontra uma associação forte e, portanto, não é relevante ao cenário total. Dessa forma, a metodologia proposta possibilita que seja encontrado um cenário mais robusto e que, a partir disso, seja possível ater-se com maior eficiência aos ataques que possuem maior chance de sucesso.

Ao avaliar outros clusters, pode-se observar que existe um que corresponde ao cenário de ataque proposto pelo conjunto de dados analisado. O cluster 3 apresenta outras informações bastante valiosas para a criação de um cenário ainda mais robusto para avaliação. Primeiramente, a figura 5.7 ilustra os alertas relativos ao cenário que pertencem ao cluster 3.

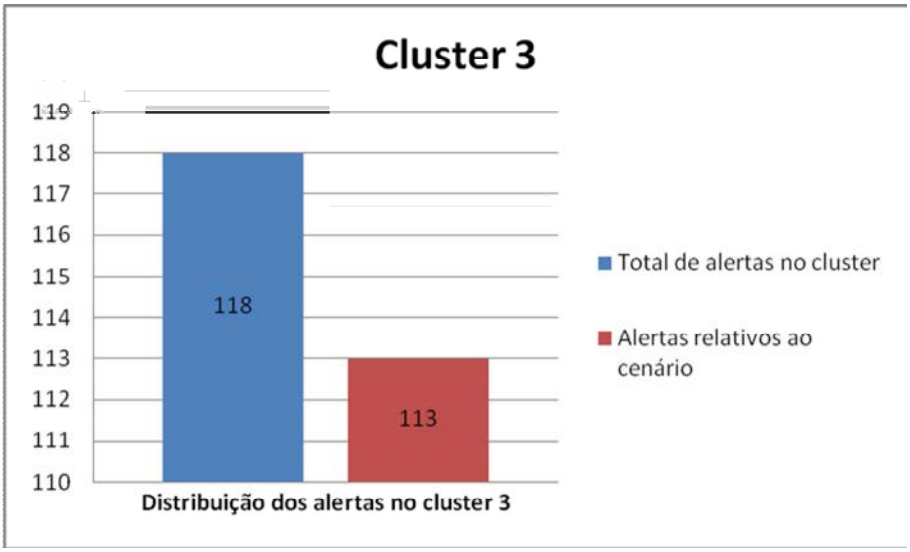

Figura 5.7. Cluster 3 contendo os alertas referentes ao cenário de ataque.

Ao executar o algoritmo *Apriori* observa-se os endereços mais frequentes neste cluster. A tabela 5.5 apresenta o resultado obtido.

Tabela 5.5. Resultado do algoritmo *Apriori* para o cluster 3.

Origem	**Quantidade de alertas de origem**	**Destino**	**Quantidade de alertas de destino**
555.77.162.213	113	333.16.114.10	13
555.77.162.213	113	333.16.114.20	13
555.77.162.213	113	333.16.114.30	13
555.77.162.213	113	333.16.115.20	26

555.77.162.213	113	333.16.112.10	18
555.77.162.213	113	333.16.112.50	30

Baseado no resultado do algoritmo *Apriori*, pode-se realizar a correlação dos alertas partindo dos endereços mais freqüentes encontrados no cluster 3. Dessa forma, pode-se observar que todos os 113 alertas que fazem parte do cenário de ataque provido pelo conjunto de dados possuem níveis elevados de correlação. Sendo assim, o grafo que pode ser gerado a partir dessa correlação pode ser analisado na figura 5.8.

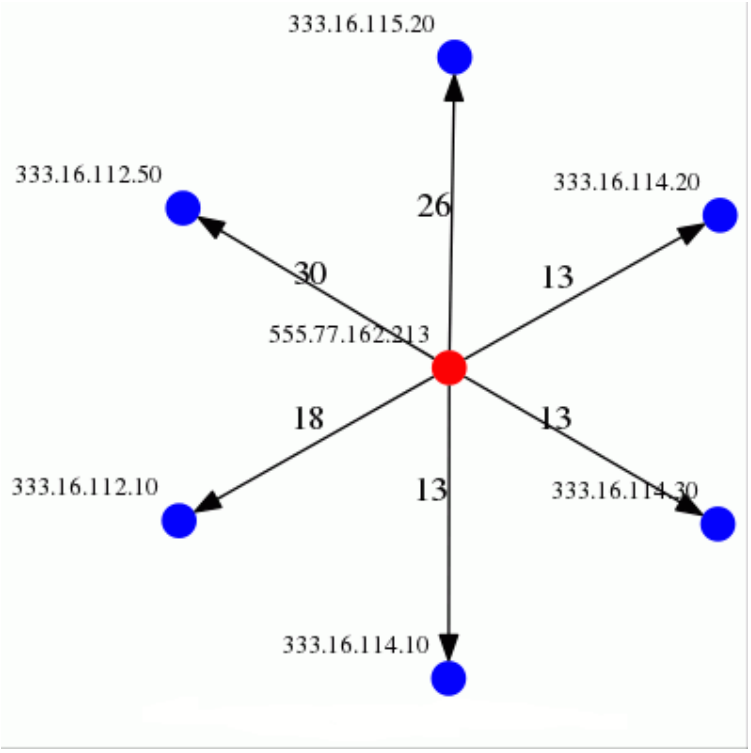

Figura 5.8. Cenário de ataque baseado nos alertas do cluster 3.

Juntando os cenários obtidos nos dois clusters, pode-se afirmar algumas hipóteses. De acordo com o cenário obtido no cluster 5 é possível dizer que essa

etapa compreende na busca por hosts ativos nas redes alvo e em busca por serviços específicos nos hosts encontrados. Baseado no cenário do cluster 3, pode-se dizer que houve sucesso na exploração de vulnerabilidades, sendo que as portas alvo eram altas, ou seja, mudou o padrão de ataque que até no cenário anterior apresentado no cluster 5 buscava por portas 111. Dessa forma, os alertas contidos no cluster 3 podem caracterizar uma comunicação entre máquinas mestre e escravo.

5.4 Considerações finais

Neste capítulo foram apresentados os resultados obtidos com os métodos implementados de classificação, clusterização, associação e visualização. Cada método possui uma função primordial em relação ao resultado final, que é a visualização de alertas correlacionados. A classificação através do *AutoClass* permitiu realizar a separação dos tipos de ataques que foram executados na rede. Em seguida, foi utilizado o algoritmo de clusterização *K-means* para agrupar alertas com características semelhantes. Com essas informações é possível separar os alertas de modo a traçar a estratégia utilizada pelo atacante. Gerar associações possibilitou obter os conjuntos de relações mais freqüentes e, dessa forma, fornecem informação para proceder com a correlação. Dessa forma, foi possível analisar com maior precisão as ameaças mais significativas. Por fim, a visualização permite montar todo o cenário de forma simples e clara, permitindo que seja possível uma tomada de decisões mais eficaz, além de ressaltar outros possíveis problemas que até o momento não eram tão evidentes.

Capítulo 6 – Conclusões

O trabalho apresentado auxilia o administrador de rede, provendo conteúdo previamente analisado e ilustrando de forma adequada as ameaças de seguranças ao qual está sujeito. Para obter esses resultados foram utilizados métodos de mineração de dados e ferramentas de visualização. O projeto visa obter uma visão maior e mais precisa da rede, permitindo que seja possível uma administração mais focada em soluções e melhorias, ao invés de gasto com análise e incertezas.

Em relação à abordagem utilizada o intuito é possibilitar uma análise de todo o fluxo de dados que trafega em uma rede de computadores. A arquitetura proposta permite que os dados que trafegam na rede sejam exportados a um coletor e que sejam tratados de forma a padronizá-los para as etapas seguintes. Ao criar uma base de dados com alertas provenientes de quaisquer IDS pode-se analisar o risco a que esta rede está sendo submetida. A avaliação dos alertas de modo a proporcionar um cenário de ataque, seja ele o mais simples possível, possibilita que sejam tomadas atitudes para mitigar problemas de exploração de falhas.

Os resultados mostraram que a metodologia é eficiente e proporcionou facilidade e legibilidade no que se refere à análise de alertas. Os testes de validação realizados ao serem submetidos aos métodos de mineração de dados apresentaram boa resposta mesmo com uma quantidade pequena de alertas, por terem sido obtidos em um ambiente controlado. Ao ser utilizada uma quantidade relativamente maior de alertas, a metodologia ainda conseguiu proporcionar resultados relevantes e promissores em relação ao cenário ocorrido na rede monitorada. Dessa forma, a avaliação é satisfatória no que diz respeito às necessidades de cada rede e, consequentemente, de cada IDS utilizado.

Resumindo os resultados obtidos, pode-se observar alguns pontos interessantes. A possibilidade de integração de vários IDS e a independência desses sensores possibilita a obtenção de outros tipos de alertas, enriquecendo a fonte de informações. A integração de métodos de mineração de dados trabalhando em conjunto, de forma que os resultados isolados de cada um se relacionem e possam culminar na correlação de alertas de segurança. A avaliação da correlação de modo a

nivelar os cenários encontrados, permitindo que alguns cenários menos significantes sejam classificados como irrelevantes, isolados ou falso-positivo. Dessa forma, descartando os alertas que compõem o cenário em questão. Por fim, criar a visualização do cenário, permitindo ao analista responder perguntas sobre sua rede, defender os computadores, estabelecer novas políticas de serviços, entre outras questões pertinentes ao cenário apresentado.

O projeto contribui de forma a não haver necessidade de um conhecimento prévio dos possíveis ataques, nem que seja necessário haver algum treinamento para que os alertas possam ser correlacionados. A metodologia possibilita classificar alertas, mesmo que eles não possuam rótulos sobre os tipos de ataques, agrupar os alertas de forma a juntar cenários em um mesmo grupo, facilitando a correlação e encontrar os participantes mais freqüentes nos alertas, auxiliando na correlação final dos alertas. Essas métricas permitem que seja reduzido o volume de dados a serem analisados e eliminam alertas isolados ou falsos-positivos que não representam um risco considerável ao ambiente monitorado.

Os métodos utilizados formam uma linha de análises que permitem atingir níveis de relações entre os alertas. Cada etapa depende da outra para que juntas façam sentido e possibilitem correlacionar os alertas. Além disso, assim como a independência de IDS, os métodos utilizados não necessitam que seja necessária uma base de dados com ataques pré-correlacionados para que a partir disso sejam buscados cenários semelhantes. Independente dos cenários existentes no conjunto de alertas é possível extrair valiosa informação sobre a situação do ambiente, podendo ser analisados cenários novos e, dessa forma, observar novas estratégias de ataque.

6.1 Trabalhos futuros e dificuldades encontradas

Como trabalhos futuros seria necessário incluir mais IDSs para melhorar a detecção e aumentar os tipos de ataques que podem ser detectados. Além de utilizar dos IDSs provenientes do laboratório ACME!, seria interessante utilizar o Snort. O Snort realiza a detecção de ataques fazendo uso de métodos que analisam o *payload*,

ou seja, o conteúdo dos pacotes. Dessa forma, será possível aprofundar ainda mais no cenário de ataque, possibilitando uma melhor contra medida.

Além disso, alguns cenários de ataques não são detectados pela metodologia apresentada. Por exemplo, ataques de negação de serviço distribuída. Uma vez que esse tipo de ataque tem como origem diversos endereços distintos à correlação baseada em ocorrência não é o suficiente para encontrar esse tipo de ameaça e apresentar ao analista. Sendo assim, será necessário abordar o problema de outra forma, tendo em vista essa dificuldade.

Outra abordagem visando a melhoria do trabalho seria utilizar os *logs* do sistema. Com essa informação seria possível criar um cenário mais rico, no caso de um atacante obter êxito na invasão, seja possível detectar o que foi feito dentro do sistema invadido. Utilizar os *logs* do sistema permitiria uma abordagem ainda mais profunda do cenário, além de focar especificamente em problemas de vulnerabilidades específicas dos serviços explorados.

A metodologia de correlação de alertas de segurança foi o ponto de maior dificuldade encontrada durante todo o projeto. Ao abordar uma metodologia que permitirá correlacionar alertas de segurança de redes, encontram-se barreiras que não permitem a utilização de métodos triviais ou únicos. Elaborar um método que realize essa tarefa consiste em um conjunto de ações e diretrizes que possibilitarão encontrar relações entre os alertas.

Além disso, foi necessário estudar detalhadamente quais deveriam ser os parâmetros adequados para utilizar tanto na classificação, utilizando o *AutoClass*, quanto na clusterização, utilizando o *K-means*. Para esse estudo foi necessário analisar quais parâmetros o IDS gerava ao construir seu alerta.

Referências Bibliográficas

AFTERGLOW. **Security visualization tool**. Disponível em: <http://afterglow.sourceforge.net/>. Acesso em 04 de Mar. 2015.

AHMADINEJAD, S.H.; JALILI, S., **Alert Correlation Using Correlation Probability Estimation and Time Windows**. Computer Technology and Development, 2009. ICCTD '09. International Conference on , vol.2, pp.170,175, 13-15 Nov. 2009.

AI, H. **Large-scale network security situational awareness based on association rule research**, Instrumentation and Measurement, Sensor Network and Automation (IMSNA), 2013 2nd International Symposium on , vol., no., pp.767,770, 23-24 Dec. 2013.

ALQAHTANI, S.M.; BALUSHI, A. M.; JOHN, R. **An Intelligent Intrusion Prevention System for Cloud Computing (SIPSCC)**. Computational Science and Computational Intelligence (CSCI), 2014 International Conference on , vol.2, pp.152,158, 2014.

BATISTA, M. L. **Análise de Eventos de Segurança em Redes de Computadores Utilizando Detecção de Novidades**. 2012. 72 f. Dissertação (Mestrado em Ciência da Computação) – Instituto de Biociências Letras e Ciências Exatas, Universidade Estadual Paulista, São José do Rio Preto. 2012.

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

BAZRAFSHAN, Z.; HASHEMI, H.; FARD, S.M.H.; HAMZEH, A. **A survey on heuristic malware detection techniques**. Information and Knowledge Technology (IKT), pp.113,120, 28-30 Mai. 2013.

CERT.BR. **Estatísticas dos Incidentes Reportados ao CERT.br**. 2015. Disponível em: <http://http://www.cert.br/stats/incidentes/>. Acesso em 04 de Mar. 2015.

CONTI, G. **Security Data Visualization**. 1ª Edição, No Starch Press, 2007.

CHEESEMAN, P.; STUTZ, J. **Bayesian classification (auto-class): theory and results**. In: FAYYAD, U. et al. (Ed.). Advances in knowledge discovery and data mining. Menlo Park, p.153–180, 1996.

DARPA. **Intrusion Detection Data Sets**. Disponível em: <http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/index.html>. Acesso em 04 de Mar. 2015.

DEBAR H.; CURRY D.; FEINSTEIN B. **RFC 4765: The Intrusion Detection Message Exchange Format (IDMEF)**. 2007. Disponível em: <http://www.ietf.org/rfc/rfc4765.txt>. Acesso em: 04 Mar. 2015.

EMRICK, E.S.; HU, Y. **An adaptive anomaly-based intrusion prevention system for databases.** Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on, pp.3382,3389, 2014.

ERMAN, J.; ARLITT, M.; MAHANTI, A. **Traffic classification using clustering algorithms**. In: SIGCOMM WORKSHOP ON MINING NETWORK DATA (MineNet '06), 2006, New York. Proceedings... New York, 2006. p. 281-286.

ERMAN, J.; MAHANTI, A.; ARLITT, M. **QRP05-4: Internet Traffic Identification using Machine Learning**. Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE, p.1-6, Nov. 27 2006.

FANG, W.; QIAN, H.; YONG, W.; LINLIN, Y. **A P2P and rule-based Web application intrusion prevention system**. Communications and Networking in China (CHINACOM), 2013 8th International ICST Conference on, pp.410,414, 2013.

FPROBE. **Fprobe**. Disponível em: <http://sourceforge.net/projects/fprobe/>. Acesso em: 04 Mar. 2015.

FREUND, Y. **Boosting a weak learning algorithm by majority**. In: COLT, pp. 202–216, 1990.

GARNER, S. **Weka: The Waikato Environment for Knowledge Analysis**. In Proc. of the New Zealand Computer Science Research Students Conference. Citeseer, pp. 57–64, 1995.

GHASEMIGOL, M.; GHAEMI-BAFGHI, A. **A new alert correlation framework based on entropy**. Computer and Knowledge Engineering (ICCKE), 2013 3th International eConference on , 184-189, 2013.

HANSON, R., STUTZ, J., CHEESEMAN, P. **Bayesian classification theory**. Technical report, NASA Ames Research Center, 1991.

HEBERLEIN, L. T.; DIAS, G. V.; LEVITT, K. N.; MUKHERJEE, B.; WOOD, J.; WOLBER, D. **A Network Security Monitor.** Proc. IEEE Symp. Research in Security and Privacy, pp. 296-304, May 1990.

ISS. **Realsecure**. Disponível em: <https://www.enisa.europa.eu/activities/cert/support/chiht/tools/iss-realsecure/>. Acesso em: 04 Mar. 2015.

KAVOUSI, F.; AKBARI, B. **Automatic learning of attack behavior patterns using Bayesian networks**. Telecommunications (IST), 2012 Sixth International Symposium on , pp.999,1004, 6-8 Nov. 2012.

LIVNAT, Y.; AGUTTER, J.; MOON, S.; ERBACHER, R.; FORESTI, S. **A Visualization Paradigm for Network Intrusion Detection**. IEEE SMC Information Assurance Workshop, p. 92-99, 2005.

MARKAM, V.; DUBEY, L. S. M. **A General Study of Associations rule mining in Intrusion Detection System**. International Journal of Emerging Technology and Advanced Engineering, v. 2, n. 1, p. 347-356, 2012.

MARTY, R. **Applied Security Visualization**. 1ª edição, Addison-Wesley, 2008.

MIRSHAHJAFARI, M., GHAVAMNIA, H. **Classifying IDS Alerts Automatically for use in Correlation System**s. Information Security and Cryptology (ISCISC), 2014 11th International ISC Conference on , 126-130, 2014.

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.

PAXSON, V. **Bro: A System for Detecting Network Intruders in Real-Time**. Proc. Seventh USENIX Security Symp., Jan. 1998.

PENG, X.; ZHANG, Y.; XIAO, S.; WU, Z.; CUI, J.; CHEN, L.; XIAO, D.. **An Alert Correlation Method Based on Improved Cluster Algorithm**. Computational Intelligence and Industrial Application, PACIIA '08. Pacific-Asia Workshop on , vol.1, 342-347, 2008.

PRIMEFACES. **PrimeFaces**. Disponível em: <http://primefaces.org/>. Acesso em 04 de Mar. 2015.

QIN, F.; TANG, X.; CHENG, Z., **Application of Apriori Algorithm in Multi Label Classification.** Computational and Information Sciences (ICCIS), 2013 Fifth International Conference on, pp.717,720, 2013

RAFTOPOULOS, E.; DIMITROPOULOS, X., **IDS Alert Correlation in the Wild With EDGe**. Selected Areas in Communications, IEEE Journal on , vol.32, no.10, 1933-1946, 2014.

ROESCH, M. **Lightweight intrusion detection technology**. Disponível em: < http://www.snort.org/ >. Acesso em: 04 de Mar. 2015.

ROESCH, M. **Snort – Lightweight Intrusion Detection for Networks**. Proc. USENIX LISA, 1999.

SHAHRESTANI, A; FEILY, M.; MASOOD, M.; MUNIANDY, B., **Visualization of invariant bot behavior for effective botnet traffic detection**, Telecommunication Technologies (ISTT), 2012 International Symposium, pp.325,330, 26-28 Nov. 2012.

SHIRAVI, H.; SHIRAVI, A.; GHORBANI, A.A. **A Survey of Visualization Systems for Network Security.** Visualization and Computer Graphics, IEEE Transactions on , vol.18, no.8, pp.1313,1329, Aug. 2012.

SINGH, A.K.; KUMAR, A.; MAURYA, A.K., **An empirical analysis and comparison of apriori and FP- growth algorithm for frequent pattern mining**. Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on, pp.1599,1602, 2014.

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

VALEUR, F.; VIGNA, G.; KRUEGEL, C.; KEMMERER, R.A. **Comprehensive approach to intrusion detection alert correlation**. Dependable and Secure Computing, IEEE Transactions on, v. 1, n. 3, p. 146-169, 2004.

WARE, C. **Information Visualization: Perception for Design**. Morgan Kaufmann, 2004.

WEKA. **Data mining software in java**. Disponível em: < http://www.cs.waikato.ac.nz/ml/weka/ >. Acesso em: 04 de Mar. 2015.

XIAO, S.; ZHANG, Y.; LIU, X; GAO, J. **Alert Fusion Based on Cluster and Correlation Analysis**. Convergence and Hybrid Information Technology, 2008. ICHIT '08. International Conference on , vol., no., pp.163,168, 28-30 Aug. 2008.

XINYOU, Z.; CHENGZHONG, L.; WENBIN, Z. **Intrusion prevention system design. Computer and Information Technology**, 2004. CIT '04. The Fourth International Conference on, p. 386-390, 2004.

XUEWEI, F.; DONGXIA, W.; SHANWEN, K.; GUOQING, M; JIN, J. **A Framework of Network Security Situation Analysis Based on the Technologies of Event Correlation and Situation Assessment**. Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on , pp.376,380, June 30 2011-July 2 2011.

XUEWEI, F.; DONGXIA, W.; MINHUAN, H.; XIAOXIA, S., **An Approach of Discovering Causal Knowledge for Alert Correlating Based on Data Mining**. Dependable, Autonomic and Secure Computing (DASC), 2014 IEEE 12th International Conference on , 57-62, 2014.

YANG, L.; GASIOR, W.; KATIPALLY, R.; CUI, X.. **Alerts Analysis and Visualization in Network-based Intrusion Detection Systems**. Social Computing (SocialCom), 2010 IEEE Second International Conference on , vol., no., pp.785,790, 20-22 Aug. 2010.

YI, J.; LI, S.; WU, M.; YEUNG, H.H.A.; FOK, W.W.T.; WANG, Y.; LIU, F. **Cloud-Based Educational Big Data Application of Apriori Algorithm and K-Means Clustering Algorithm Based on Students Information**. Big Data and Cloud Computing (BdCloud), 2014 IEEE Fourth International Conference on , vol., no., pp.151,158, 2014.

YUAN, S; ZOU, C. **The security operations center based on correlation analysis**. Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on , vol., no., pp.334,337, 27-29 May 2011.

ZARGAR, S. T.; JOSHI, J.; TIPPER, D. **A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks**. Communications Surveys & Tutorials, IEEE , vol.15, no.4, pp.2046,2069, 2013.

ZHU, B.; GHORBANI, A. **Alert correlation for extracting attack strategies.** International Journal of Network Security, vol. 3, no. 3, pp. 244–258, 2006.

ZHUO, W.; NADJIN, Y. **MalwareVis: entity-based visualization of malware network traces**. In Proceedings of the Ninth International Symposium on Visualization for Cyber Security (VizSec '12). ACM, New York, NY, USA, 41-47, 2012.