Universidade Estadual de Londrina

WELLINTON PIASSA

AGRUPAMENTO DE ALERTAS DE INTRUSÃO BASEADO NO MÉTODO *ATTRIBUTE-ORIENTED INDUCTION*

LONDRINA
2023

WELLINTON PIASSA

AGRUPAMENTO DE ALERTAS DE INTRUSÃO BASEADO NO MÉTODO *ATTRIBUTE-ORIENTED INDUCTION*

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Bruno Bogaz Zarpelão

LONDRINA
2023

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UEL

Piassa, Welinton.
AGRUPAMENTO DE ALERTAS DE INTRUSÃO BASEADO NO MÉTODO ATTRIBUTE-ORIENTED INDUCTION / Welinton Piassa. - Londrina, 2023.
46 f. : il.

Orientador: Bruno Bogaz Zarpelão.
Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Graduação em Ciência da Computação, 2023.
Inclui bibliografia.

1. Sistema de Detecção de Intrusão - TCC. 2. Alertas de Intrusão - TCC. 3. Agrupamento - TCC. 4. Attribute-Oriented Induction - TCC. I. Zarpelão, Bruno Bogaz. II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Graduação em Ciência da Computação. III. Título.

CDU 519

WELLINTON PIASSA

AGRUPAMENTO DE ALERTAS DE INTRUSÃO BASEADO NO MÉTODO *ATTRIBUTE-ORIENTED INDUCTION*

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA

Orientador: Prof. Dr. Bruno Bogaz Zarpelão
Universidade Estadual de Londrina

Prof. Dr. Fabio Sakuray
Universidade Estadual de Londrina

Rildo Antônio de Souza
Rede Nacional de Ensino e Pesquisa (RNP)

Londrina, 19 de Maio de 2023.

Este trabalho é dedicado aos meus pais que sempre acreditaram em mim e nunca mediram esforços em me apoiar nas minhas decisões.

AGRADECIMENTOS

Primeiramente agradeço ao Prof. Dr. Bruno Bogaz Zarpelão, por todos os conselhos, conhecimentos compartilhados e principalmente pela paciência ao longo da realização desse trabalho. Agradeço aos meus pais que sempre me apoiaram, confiaram e acreditaram na minha perseverança para alcançar meus objetivos. Aos meus colegas de curso por compartilharem comigo tantos momentos de descobertas e aprendizado. A todos os meus amigos que estiveram ao meu lado durante essa jornada, em especial meu amigo Vitor G.S. Ruffo que trilhou todos os anos de curso comigo compartilhando momentos e conhecimentos. Por fim, mas não menos importante, gostaria de agradecer todos os professores do departamento de computação, que me proporcionaram grandes conhecimentos e amadurecimento aos longo desses anos.

"Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar. Mas o mar seria menor se lhe faltasse uma gota."
(Madre Teresa de Calcutá)

PIASSA, W.. **Agrupamento de Alertas de Intrusão baseado no método *Attribute-Oriented Induction***. 2023. 46f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina, 2023.

RESUMO

Com o grande crescimento da Internet, vem-se criando uma necessidade de fortalecer a segurança das redes contra ataques e invasões de atores mal-intencionados. Os sistemas de detecção de intrusão (*intrusion detection system* - IDS) são uma das opções para ajudar a combater os ataques, informando ao administrador da rede sobre a ocorrência de eventos suspeitos que ameaçam a segurança de uma rede ou um *host*. Um problema que aparece ao utilizar esses sistemas é, primeiramente, a geração de grandes volumes de alertas. Soma-se a isso o fato de que, dentre os alertas gerados, existe uma parte deles que são falsos-positivos, atrapalhando o analista de segurança de redes a se concentrar nos alertas mais importantes. Visando abordar essa questão, este trabalho tem a proposta de estudar e aplicar uma técnica de clusterização baseada no algoritmo *Attribute-Oriented Induction* - AOI para reduzir o volume de alertas, e então apresentar cenários que possam fazer com que o analista de segurança consiga priorizar os alertas que merecem maior atenção em uma análise mais aprofundada. Essa proposta tem o objetivo de ajudar analistas de segurança de redes a lidar com grandes volumes de alertas, conseguindo detectar padrões e cenários que mereçam maior atenção. Além disso, nesse trabalho foi realizado um estudo de caso utilizando dados reais fornecidos por uma organização acadêmica e uma *blocklist* pública de endereços IP.

Palavras-chave: Sistema de Detecção de Intrusão. Alertas de Intrusão. Agrupamento. Attribute-Oriented Induction.

PIASSA, W.. **Intrusion Alert Prioritization based on the Attribute-Oriented Induction method**. 2023. 46p. Final Project (Bachelor of Science in Computer Science) – State University of Londrina, Londrina, 2023.

ABSTRACT

Along with the Internet growth comes the need for improving its security against attacks and intrusions from malicious agents. Intrusion Detection Systems are mechanisms for countering network attacks. They alert network administrators when security events take place, and even execute pre-defined actions to reduce network damage. Those systems usually generate a ton of alerts. Commonly, many of them are false alerts, or false positives, which prevent the administrators from analyzing and dealing with the real ones. This final project aims to study and apply an Attribute Oriented Induction-based clustering algorithm for reducing alarm volume and providing a context that helps security analysts to prioritize the most relevant alarms. That algorithm helps administrators to deal with high volumes of alerts by identifying patterns and scenarios that need more attention. A study case was developed by using a real-world dataset provided by an academic organization and a public IP blocklist.

Keywords: Intrusion Detection System. Intrusion Alert. Grouping. Attribute-Oriented Induction method.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

IDS	Intrusion Detection System
CSV	Comma-separated values
AS	Autonomous System
ASN	Autonomous System Number
API	Application Programming Interface
NIDS	Network Intrusion Detection Systems
HIDS	Host Intrusion Detection Systems
AOI	Attribute-Oriented Induction
SSH	Secure Socket Shell
SVM	Support Vector Machine
FTP	File Transfer Protocol

SUMÁRIO

1 INTRODUÇÃO

Com a grande evolução da tecnologia, surgiram ferramentas que se tornaram essenciais no mundo atual e dentre elas está a Internet. A Internet é uma enorme quantidade de computadores interligados em todo o mundo, conectados e trocando informações entre si [1]. A partir disso, vários serviços de grande importância puderam ser ofertados através da Internet, tais como lojas virtuais, banco de dados, serviços financeiros, serviços governamentais, dentre outros.

Sendo assim, dados importantes e confidenciais são transmitidos e armazenados nessas máquinas conectadas à Internet, despertando atenção de usuários mal intencionados que aproveitam vulnerabilidades das redes para realizar ataques. Portanto, medidas de segurança são fundamentais para combater todo tipo de ameaça e proteger dados sensíveis que podem ter valores inestimáveis para uma instituição [2].

Considerando isso, surgiram vários desafios para a área de segurança de redes, pois o rápido desenvolvimento da tecnologia da informação dificulta a construção de redes totalmente confiáveis. Portanto, detectar e combater todo tipo de ataque é uma tarefa bastante árdua, já que existem vários tipos diferentes de ataques, e frequentemente surgem novos tipos e novas vulnerabilidades. Dentre esses ataques, pode ser mencionado o ataque de negação de serviço *(Denial of Service* - DoS), considerado um dos ataques mais comuns [3].

Para auxiliar no combate e na prevenção de ataques, algumas ferramentas estão sendo desenvolvidas e aprimoradas. Uma dessas ferramentas são os IDSs (*Intrusion Detection System* - Sistema de Detecção de Intrusão), que há pelo menos duas décadas se tornou um tema relevante em pesquisas, já que é uma ferramenta importante para a segurança de computadores [4]. Os IDSs mais comuns, como o *Snort*, monitoram o tráfego de rede, verificando violações de políticas de segurança e atividades maliciosas, e caso algum tipo de atividade prejudicial seja descoberta, é relatado ao administrador da rede [5].

Entretanto, um IDS pode gerar um enorme volume de alertas. Dentre esse grande volume de alertas produzidos, geralmente grande parte deles são alertas classificados como falsos-positivos. Alertas falsos-positivos são alertas que o IDS gerou por julgar um comportamento normal da rede como uma ameaça [6]. Sendo assim, esse cenário pode trazer dificuldades para o analista de segurança, por ter que lidar com um grande volume de alertas. Além disso, dificulta a detecção de situações mais críticas que precisam de mais atenção, fazendo com que seja muito complicado enxergar quais alertas devem ser priorizados e analisados com mais ênfase.

Pensando nesse problema, a proposta deste trabalho é estudar e aplicar uma técnica de clusterização baseada no método *Attribute Oriented Induction*. O intuito dessa proposta é reduzir o volume de alertas e então apresentar cenários e padrões que possam ajudar o analista de segurança a priorizar os alertas que necessitam de maior atenção em uma análise mais aprofundada. Nesse trabalho foi realizado um estudo de caso aplicando a proposta em dados reais de uma organização acadêmica. Para auxiliar na identificação de padrões de alertas foi utilizada uma *blocklist* pública.

Este trabalho está organizado da seguinte maneira. O capítulo 2 apresenta a fundamentação teórica, trazendo definições e detalhes de um Sistema de Detecção de Intrusão e também métodos de processamento de alertas, como agregação, correlação e redução de alertas de intrusão. No capítulo 3 é descrita a solução proposta do trabalho. O capítulo 4 descreve o estudo de caso realizado, apresentando todos os resultados obtidos e discussões acerca deles juntamente com as contribuições que esse trabalho traz. Por fim, no capítulo 5 é apresentada a conclusão do trabalho com possíveis propostas de para se desenvolver em trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICO-METODOLÓGICA

2.1 Sistema de Detecção de Intrusão

Um Sistema de Detecção de Intrusão (IDS, do inglês *Intrusion Detection System*), é uma ferramenta de segurança cujo objetivo é monitorar o comportamento de eventos buscando encontrar atividades maliciosas e violações de políticas de segurança [5, 7]. Quando algum comportamento ou evento suspeito é detectado pelo IDS, um alerta é enviado ao administrador de rede para que sejam tomadas as medidas necessárias e para que ele possa analisar detalhadamente o evento ocorrido [8, 5, 9, 10, 11].

O IDS pode ser dividido em duas categorias dependendo onde ele é implantado e o que o administrador de rede visa proteger. As categorias são denominadas como: sistema baseado em redes (NIDS - *Network Intrusion Detection Systems*) e sistema baseado em *host* (HIDS - *Host Intrusion Detection Systems*) [4, 12, 9]. Um NIDS geralmente se encontra no ponto de interconexão da rede com outras redes analisando os pacotes trafegados entre a rede interna e externa, na busca de detectar alguma ameaça [4]. Já o HIDS é instalado em um único sistema, realizando o monitoramento do tráfego de rede daquele dispositivo, e, além disso, analisando *logs*/eventos do sistema, rastreando processos e tendo acessos a mudanças de arquivos do sistema [13, 12, 9].

Existem dois métodos de detecção em IDS, denominados detecção por assinatura e por anomalia [4, 10, 14, 1], cada um tendo suas vantagens e desvantagens. No método por anomalia, a detecção é baseada no comportamento da rede, ou seja, é estabelecido um comportamento padrão usando aprendizado de máquina, métodos baseados em estatística ou baseados em conhecimento [7] e, uma vez que ocorre o desvio do comportamento padrão ele será considerado uma anomalia ou um ataque [4]. A vantagem de utilizar a detecção por anomalia é conseguir detectar ataques tanto conhecidos como desconhecidos, possibilitando que o IDS consiga lidar com a constante mudança da natureza de ataques [14]. Sua desvantagem seria o potencial de gerar um grande número de alertas falsos-positivos, podendo caracterizar comportamentos legítimos como anomalia [13, 3, 8].

No método de detecção por assinatura, o sistema possui armazenados padrões de vários ataques já conhecidos e caso a situação que está sendo analisada se encaixe em um desses padrões, ele será identificado como uma atividade maliciosa [1, 5]. A vantagem dessa abordagem é ter um volume bem menor de alertas falsos-positivos em comparação com a abordagem de detecção por anomalia, e sendo mais eficientes com ataques conhecidos [8]. Já sua desvantagem seria a impossibilidade de detectar ataques desconhecidos, pois esses não se enquadram nas regras definidas na base de conhecimento do IDS [4, 7].

A Figura 1 apresenta uma visão geral das categorias nas quais podemos classificar um IDS de acordo com a sua implantação e os métodos utilizado para detectar ataques.

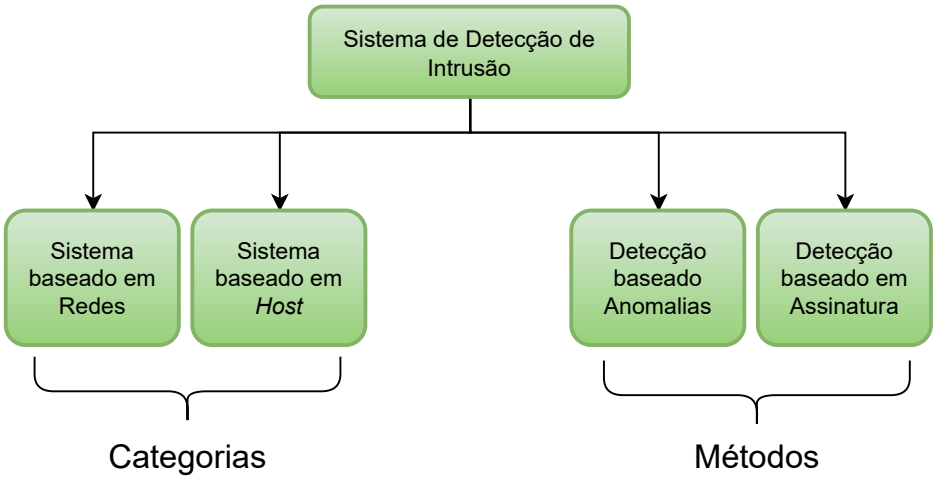

Figura 1 – Diagrama de classificação de um IDS

Como já mencionado anteriormente, no momento em que os IDSs detectam um ataque, é realizado o disparo de um alarme ao administrador de redes [1]. O alarme mostra informações do possível ataque ou um comportamento incomum. Os alarmes gerados podem ser classificados em dois tipos [1], sendo eles:

- Verdadeiro Positivo: sistema foi atacado e o alerta foi gerado corretamente.
- Falso Positivo: foi gerado um alerta, porém não houve ataque.

O Verdadeiro Positivo é considerado o ideal absoluto por conta da sua consistência e confiabilidade, pelo fato de gerar o alerta somente quando realmente estiver ocorrendo um ataque. Já o Falso Positivo pode apenas atrapalhar o administrador de redes em sua tarefa de analisar os alertas que realmente são ataques. Outra situação prejudicial além do falso positivo são os eventos maliciosos que ocorrem e não são detectados, o que pode ser considerado mais crítico pelo fato de ter ocorrido um ataque e o IDS sequer ter disparado um alerta.

2.2 Processamento de Alertas

A utilização do IDS pode ajudar os administradores de redes a minimizar os danos causados por diferentes ataques [15]. No entanto, eles podem gerar um grande volume de alertas, que frequentemente são falsos ou irrelevantes, dificultando o trabalho do administrador de redes [16]. Para lidar com esse problema, duas soluções podem ser exploradas, sendo a primeira trabalhar dando enfoque diretamente na configuração do IDS e a segunda seria trabalhar sobre as saídas geradas pelo dispositivo de monitoramento, ou seja, tratar os alertas [17].

Sendo assim, um dos objetivos de processar os alertas é reduzir ou descartar alertas falsos e priorizar os que são mais críticos [17]. Essa seção apresenta algumas formas de reduzir o grande volume de alertas. Além disso, o processamento de alertas é uma etapa importante para buscar semântica entre os alertas para que o administrador de redes tome as decisões e ações necessárias. Por mais que as mensagens de alertas tragam informações sobre os eventos detectados, é necessário aplicar técnicas de processamento que vão permitir ter uma visão mais precisa e abrangente dos diferentes eventos que estão ocorrendo no sistema monitorado e possivelmente enxergar problemas que estão implícitos nesses conjuntos de dados.

Os métodos abordados aqui são a correlação, agregação e redução de alertas, onde cada um deles possui suas técnicas e taxonomias.

2.2.1 Agregação de Alertas

Segundo [18] a agregação é realizada no intuito de unir alertas similares, reduzindo a quantidade redundante dos mesmos. Outra definição seria sua utilização para remover ou reduzir o número de alertas duplicados [19]. Na literatura, são apresentados alguns métodos para realizar a agregação de alertas. Alguns deles são: clusterização e a comparação de atributos [18].

2.2.1.1 Métodos de Agregação de Alertas

No método de clusterização, é feita a separação dos alertas similares dos dissimilares [18], sendo assim, um só alerta representará todo o *cluster*, ou seja, todos os outros alertas similares. Dentre as técnicas de clusterização apresentadas por [18] estão Mapas Auto-Organizáveis (*Self-Organizing Map* - SOM) com k-médias, *Density-based Spatial Clustering of Application with Noise* (DBSCAN) e Indução Orientada a Atributo (*Attribute-Oriented Induction* - AOI).

O método de comparação de atributos pode ser considerado mais simples em relação aos métodos de clusterização. Dentre as técnicas encontrados para esse método na literatura estão presentes comparações simples de atributos e funções de similaridade [18]. O método de comparação simples de atributos realiza o agrupamento de alertas com mesmos valores em determinados atributos, como, por exemplo, endereços IP de origem e/ou destino. Já o método de função de similaridade tem como base funções probabilísticas para cada atributo, e quando a probabilidade dos atributos atendem um valor mínimo definido, os alertas são agrupados [20].

2.2.1.2 Situações de Agregação

Em muitos casos, eventos isolados não são considerados significativos. Portanto, os alertas agregados podem apresentar situações presentes no conjunto de dados, ou seja, um cenário que pode representar o formato de um possível ataque. Para buscar essas situações, podem ser feitos agrupamentos com base em seus atributos. Em [21] são apresentadas algumas situações que podem ser visualizadas mediante agregações de alertas. Os atributos de exemplo são: a classe do alerta e as identificações da origem e do destino do alerta.

- **Situação 1:** agregar alertas com a mesma origem, o mesmo destino e a mesma classe. Isso permite detectar, por exemplo, um invasor que está lançando uma série de ataques ao servidor Web contra um único servidor Web.
- **Situação 2:** agregar alertas com a mesma origem e destino. Esta situação destina-se a detectar, por exemplo, um invasor que executa uma série de ataques contra os vários serviços disponíveis na máquina de destino.
- **Situação 3:** agregar alertas com o mesmo destino e mesma classe de alerta. Esta situação pode ser usada para detectar um ataque distribuído contra um único alvo.
- **Situação 4:** agregar alertas com a mesma origem e mesma classe. Essa situação permite, por exemplo, encontrar um invasor que está executando uma série de ataques de servidor de nomes contra um conjunto de nomes servidores.

2.2.2 Correlação de Alertas

A correlação de alertas é a interpretação conceitual de vários alarmes de modo que novos significados sejam atribuídos a ele [22]. A agregação e a correlação de alertas podem ser conceitos muito semelhantes, porém a agregação de alerta é usada para remover ou reduzir o número de alertas duplicados, enquanto a correlação de alerta é usada para diminuir as taxas de falsos negativos e descobrir a estratégia de ataque [19].

O objetivo das correlações é formar um grupo de alertas para apresentar a visão de um cenário de ataque que está acontecendo, ou prever um que pode acontecer [15]. A correlação de alertas pode ser classificada em três métodos, sendo eles o método baseado em similaridade, método baseado em causa e sequência e o método baseado em casos [22, 23] conforme ilustrado pela Figura 2.

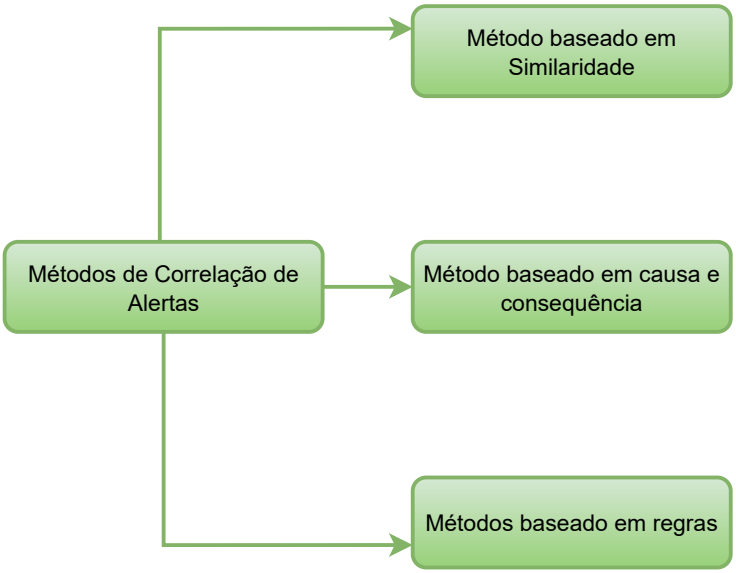

Figura 2 – Diagrama dos métodos da correlação de alertas

O método baseado em similaridade tem em vista reduzir o número de alertas, realizando um agrupamento dos alertas que possuem semelhanças em seus atributos [24]. A correlação por similaridade pode ser efetuada por análise estatística, conjunto de regras, distância Euclidiana e outras métricas de similaridade [23]. Alguns exemplos de atributos principais são número de porta de origem e destino, endereço IP de origem e destino, *timestamp*, protocolo e a descrição do alerta. Além disso, a correlação pode ser realizada dando foco na base temporal pelo fato de que os relacionamentos temporais podem trazer informações valiosas [22].

No método baseado em causa e consequência, a correlação se concentra inicialmente em um conjunto de alertas preparatórios, ou seja, alertas que indicam a preparação ou início de um ataque [23]. Nesse caso, os alertas precisam ter pelo menos um dos atributos iguais e o *timestamp* deve ter uma sequência.

A correlação utilizando o método baseado em regras funciona sobre de um conjunto de regras ou casos presentes em uma base de conhecimento de cenários de ataque [23]. Quando o alerta é gerado, a correlação procura na base de dados de conhecimento cenários

que envolvem esse tipo de alerta. A base de conhecimento geralmente é montada utilizando mineração de dados ou heurísticas definidas por especialistas [22].

2.2.3 Redução de Alertas Falsos-Positivos

Alertas falsos-positivos são aqueles gerados por um IDS sobre uma atividade normal, onde, porém, considerada pelo IDS como maliciosa [25]. Esses alertas, além de tornar mais complexa a realização de análises, reduzem o desempenho do IDS, sendo, portanto, considerado um grande problema na maioria das organizações [26]. Posto isso, reduzir o número dessas ocorrências é de suma importância e pode auxiliar no dia-a-dia ao lidar com os alertas [6].

Para realizar reduções de alertas, algumas técnicas de aprendizado de máquina estão presentes na literatura, como, por exemplo, a Árvore de Decisão e Máquina de Vetor de Suporte [18]. Ambas técnicas utilizam o aprendizado de máquina supervisionado, dessa forma, necessitam de dados rotulados para treinar os algoritmos, e rotular manualmente um conjunto de dados é uma tarefa bastante árdua.

Árvore de Decisão (*Decision Tree*) é baseada em uma árvore binária e que possui dois tipos de nós, o nó folha e os nós internos. Os nós folha contém rótulos de classe ou previsões enquanto os nós internos contêm funções para dividir uma amostra original [27]. Essa técnica é baseada no conceito *top down* e o conceito de divisão e conquista [2].

A Máquina de Vetor de Suporte (SVM, do inglês *Support Vector Machine*) é usada em aprendizado de máquina para resolver problemas de regressão e classificação [28]. Para ambas as técnicas, tanto a árvore de decisão quanto a SVM, é preciso preparar um conjunto de dados com vários alertas verdadeiros e falsos. A partir desse conjunto de dados, os modelos são treinados e vão aprendendo. Após o treinamento, é possível classificar novos alertas em falsos ou verdadeiros.

3 CLUSTERIZAÇÃO DE ALARMES BASEADA NO MÉTODO *ATTRIBUTE-ORIENTED INDUCTION*

3.1 Contextualização

Como explanado na seção 2.1, um IDS é um dispositivo que dispara alertas quando detecta algum comportamento anômalo na rede ou quando algum comportamento viola regras ou assinaturas pré-estabelecidas pelo administrador de redes. No entanto, os IDS podem disparar centenas de alertas diariamente, fazendo com que acumule um volume muito grande de alertas e por consequência acabe dificultando o trabalho do analista de segurança.

A geração de grandes volumes de alertas pode acontecer por diferentes motivos, podendo ser pelo fato do IDS estar mal configurado, o que pode gerar muitos falsos positivos. Outro exemplo é a detecção de passos que um atacante realiza em busca de concretizar um ataque, ou seja, cada passo que ele realiza faz com que o IDS identifique e gere um alerta. Dessa maneira, para uma tentativa de ataque, o IDS pode gerar vários alertas, e consequentemente, se a rede ou sistema monitorado recebe várias tentativas de ataques, o IDS tem o potencial de gerar um volume gigantesco de alertas.

Em [29], Julisch et al. propõem um método que será utilizado como a base do presente estudo, que busca solucionar o problema de tratar o volume excessivo de alertas e trazer cenários e padrões de alertas que chamam mais atenção do analista de segurança, permitindo uma melhor compreensão da rede a partir de um grande volume de alertas que não trazem tanta semântica. Com o conhecimento que o analista de segurança obtém analisando os alertas, ele pode conseguir criar regras e visualizar situações específicas que estão acontecendo na rede, e que precisam de atenção, podendo ser tratadas de forma mais eficaz. Um dos exemplos para isso seria descobrir padrões de comportamentos de alertas falsos positivos, havendo a possibilidade da criação de regras para filtros desses padrões detectados. No restante deste capítulo será apresentado o método desenvolvido para clusterizar e agrupar alertas com base na proposta de Julisch et al.

Posto isso, duas estratégias foram definidas para alcançar o objetivo desse trabalho. A primeira delas é dividir o conjunto de alertas em dois novos conjuntos A e B, onde essa divisão é realizada com base em informações de uma *blocklist*. Uma *blocklist* é uma lista pública de endereços IP que já tem um histórico de envolvimentos em diversos ataques e IPs que foram reportados várias vezes. Essa divisão ocorre da seguinte forma: os alertas que possuem algum de seus endereços IP (origem ou destino) na *blocklist* serão inseridos ao novo conjunto A e alertas que não possuem IPs presente na *blocklist* vão ser inseridos no novo conjunto B.

Posteriormente é realizada a aplicação do algoritmo de clusterização sobre os conjuntos de dados A e B, buscando reduzir o volume de alertas de ambos. Além da redução dos dados, a aplicação do algoritmo de clusterização também ajudará a criar cenários e situações que vão indicar padrões a respeito dos alertas presentes em cada um dos conjuntos de alertas e possibilitando uma visão mais alto nível. Sendo assim, o algoritmo irá gerar conjuntos resultantes contendo grupos (ou *clusters*) apresentando esses padrões.

Sabendo disso, o conjunto de dados A será utilizado como referência para indicar possíveis cenários que devem ser priorizados ao analisar outros alertas, já que, os alertas presentes no conjunto A são alertas que possuem endereços IP já envolvidos em outros incidentes que foram reportados por outras instituições. Sendo assim, a estratégia é mapear as situações predominantes do conjunto A e buscar cenários similares no conjunto B almejando situações onde os alertas envolvidos podem ser priorizadas ao realizar uma análise mais aprofundada.

3.2 Algoritmo de Clusterização

O algoritmo de *conceptual clustering* desenvolvido no trabalho de Julisch et al. [29] tem como base um algoritmo clássico de *Attribute-Oriented Induction* (AOI), no qual os autores realizaram algumas modificações. O algoritmo objetiva realizar uma redução da grande quantidade de alertas presentes em um conjunto de dados, gerando um resultado com menor volume e mantendo o máximo possível das informações originais.

Para isso, a ideia principal do algoritmo é percorrer todo o conjunto de alertas agrupando aqueles que possuem os mesmos valores em todos os atributos, formando *clusters*. Após o agrupamento, os *clusters* que atingirem um tamanho mínimo requerido são removidos do conjunto de dados atual e transferidos para outro conjunto resultante. Os *clusters* que não permaneceram passam por um processo de generalização, onde seus valores são substituídos por outros mais abstratos definidos por uma hierarquia de generalização. Isso faz com que o algoritmo consiga seguir agrupando valores que antes da generalização não poderiam ser agrupados, reduzindo uma grande quantidade de alertas em uma quantidade inferior de *clusters*.

3.2.1 Formalização e Representação dos Dados

Para trazer uma melhor compreensão do processo, a formalização e representação dos alertas é importante. A representação dos alertas é feita em formato de tabelas, semelhante a uma representação de um banco de dados relacional conforme ilustrado na Tabela 1. Nessa representação dos alertas, existem as colunas que representam os atributos de alertas, como, por exemplo, endereços IP de origem e destino, porta de destino e assinatura. Cada linha, por sua vez, representa, inicialmente, um alerta.

IP_origem	IP_destino	Porta_destino	Assinatura
ip1	ipA	80	TOR
ip2	ipA	443	DNS
ip3	ip1	443	DNS
ip4	ipB	80	TOR
...	...	...	...
ipA	ip2	443	DNS
ipB	ip6	80	TOR
...	...	...	...
ipZ	ip3	80	P2P

Tabela 1 – Exemplo de representação do conjunto de alertas

Cada um dos atributos podem ser representados por A_i, onde i é número natural, $A_1, A_2, A_3, ..., A_n$. Exemplificando a partir da Tabela 1, o atributo IP_origem pode ser representado por A_1, IP_destino por A_2, e assim por diante. Cada um desses atributos possui alguns valores, que podem ser representados por D_{A_1}, que significa "Domínio do atributo 1". A Tabela 2 apresenta um exemplo com o domínio do atributo IP_origem, ou seja, com os valores de IP de origem encontrados nos alertas analisados.

IP_origem
ip1
ip2
ip3
ip4
...
ipA
ipB
...
ipZ

Tabela 2 – Exemplo de atributo com seu domínio de valores

3.2.2 Generalização de Dados

A generalização dos dados é realizada com base em uma hierarquia de generalização, que deve ser definida para cada um dos atributos. A hierarquia de generalização define a cadeia de como uma informação muito específica deve ser generalizada para um nível mais abstrato. Um exemplo disso são *timestamps* de alarmes, que ao invés de informar o momento exato, podem ser generalizados para o dia da semana.

A definição da hierarquia de generalização pode ser estabelecida pelo usuário do algoritmo, possibilitando realizar uma escolha que ofereça melhor compreensão e interpretabilidade dos agrupamentos de alertas. Além disso, a hierarquia de generalização pode

ser representada em forma de árvore n-ária conforme ilustra a Figura 3 com um exemplo de hierarquia de generalização de endereços IP.

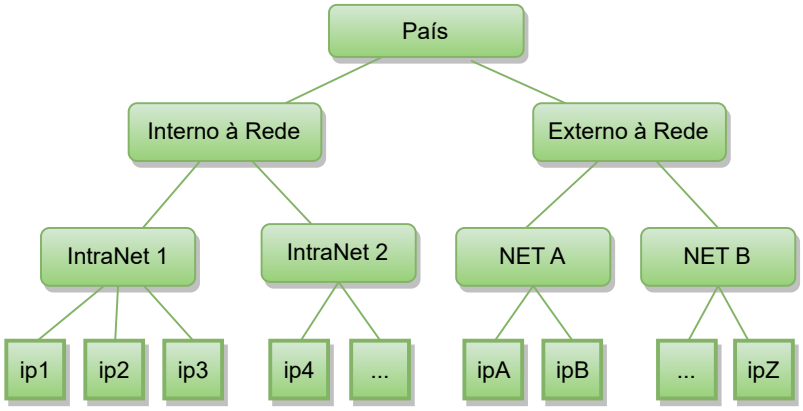

Figura 3 – Exemplo de hierarquia de generalização para endereços IP

Na representação da hierarquia de generalização em forma de árvore, as folhas simbolizam os dados mais específicos, enquanto as raízes representam dados mais genéricos em relação ao valor da folha. Na árvore de hierarquia de generalização para endereço IP apresentada pela Figura 3, o *ip1*, por exemplo, pode ser generalizado como um IP pertencente à *IntraNet 1*, ou um IP que pertence à rede interna da organização. O *ipA* por sua vez, pode ser generalizado como um IP pertencente à *NET A*, e posteriormente, caso necessário pode ser representado como um IP pertencente à rede externa da organização.

A generalização dos dados realiza uma abstração da informação original passando para um nível superior, por conseguinte, em alguns casos podem ocorrer um problema chamado supergeneralização. Esta é uma situação que implica em perdas significativas de informações semânticas do conjunto de dados original, resultante do fato que o grau de abstração foi tão alto que o dado não traz mais significado para a análise. Para reduzir as chances disto acontecer, existem algumas estratégias aplicadas ao algoritmo AOI clássico, fazendo com que ele generalize as informações somente quando for necessário, em busca de melhores resultados.

3.2.3 Pseudo-código do Algoritmo de Clusterização

Para melhor compreensão do algoritmo de clusterização apresentado pela proposta desse trabalho, essa seção apresenta um pseudo-código detalhando como é realizado o processo de clusterização.

Algorithm 1 Algoritmo de Clusterização utilizando *Attribute-Oriented Induction*

Entrada: Conjunto de Alertas δ, uma Hierarquia de Generalização β e um valor inteiro para *min_size*
Saída: Um novo conjunto de alertas clusterizados

1: **para todo** alerta α em δ **faça:**
2: $\alpha.Count \leftarrow 1$
3: **enquanto** δ não vazio **faça**
4: agrupa os alertas que possuem os mesmos valores em todos os atributos
5: **se** δ tem cluster σ onde $\sigma.Count \geq$ min_size **então**
6: **reporta** σ onde $\sigma.Count \geq$ min_size
7: $\delta \leftarrow \sigma$ onde $\sigma.Count <$ min_size
8: $\delta \leftarrow$ remove a generalização de todos os alarmes de δ
9: **para todo** alerta α em δ **faça:**
10: $\alpha.Count \leftarrow 1$
11: **fim se**
12: $A_S \leftarrow$ escolhe qual atributo será generalizado
13: **para todo** alerta α em δ **faça:**
14: $\alpha.A_S \leftarrow \beta.valor$ pai de A_S em β
15: **fim enquanto**

As entradas que o algoritmo necessita são: o conjunto de alertas δ no formato descrito na seção 3.2.1, uma hierarquia de generalização para cada um dos atributos $(A_1, A_2, A_3, ..., A_n)$ e um atributo *min_size*, o qual é um número inteiro que define a quantidade mínima de alertas que o algoritmo deve conseguir agrupar em cada *cluster* que ele irá formar. Antes de processar os dados, é importante adicionar ao conjunto de alertas um atributo denominado *count*, que irá informar a quantidade de alertas que estão agrupados naquele *cluster* em específico. Portanto, os atributos podem ser representados da seguinte forma com a adição do novo atributo: $(A_1, A_2, A_3, ..., A_n, C)$.

O primeiro passo do algoritmo — descrito nas linhas 1 e 2 do Algoritmo 1 — é percorrer todos os alertas presentes no conjunto de alertas por meio de um laço de repetição e atribuir o valor 1 para o atributo *count* de cada um, tornado-os agora em um *cluster* de tamanho 1. A partir de agora, todos os alertas são *clusters* e o algoritmo encara o conjunto de alertas como um conjunto de *clusters* que no processo ainda podem ser reunidos para formar *clusters* maiores.

Na linha 3 é iniciado o laço de repetição que envolve todo o processo de clusterização. Ele mantém o processamento do algoritmo executando até que o conjunto de alertas de entrada encontre-se vazio. Posteriormente, na linha 4 ocorre um agrupamento de *clusters* que possuem os mesmo valores para todos os atributos. Nesse passo, o algoritmo realiza um laço de repetição que percorre todos os *clusters*, onde em cada iteração será selecionado um *cluster* α. Em seguida, é verificado no restante do conjunto se existe um *cluster* α' igual a α. No momento que α' é encontrado, ele é removido do conjunto

e o atributo *count* de α recebe a soma de mais 1 ao seu valor, buscando simular um agrupamento e representar que o *cluster* α' está contido no *cluster* α.

A Tabela 3 ilustra um cenário em que alguns *clusters* se repetem, mais especificamente os *clusters* da primeira, quarta e quinta linha (em verde). Os *clusters* da segunda e sétima linha (em azul) também são iguais. Quando o algoritmo está no primeiro laço de repetição, ou seja, na primeira linha da tabela, ele irá analisar todas as outras linhas buscando *clusters* que são iguais ao da primeira, e quando encontrar algum remove-o do conjunto e soma 1 ao valor de *count* do *cluster* da primeira linha, que está sendo usado como base para verificação no momento. O mesmo acontece para a segunda linha, que possui valores iguais à sétima linha, removendo o *cluster* da sétima linha e soma 1 ao valor de *count* da segunda linha. O resultado do agrupamento é a Tabela 4

IP_origem	IP_destino	Porta_destino	Assinatura	count
ip1	ipA	80	TOR	1
ip2	ipA	53	DNS	1
...	...	...	...	...
ip1	ipA	80	TOR	1
ip1	ipA	80	TOR	1
...	...	...	...	...
ip2	ipA	53	DNS	1
ipA	ip4	80	TOR	1
ipB	ip4	53	DNS	1
ipZ	ip4	53	DNS	1

Tabela 3 – Exemplo de conjunto de *clusters*

IP_origem	IP_destino	Porta_destino	Assinatura	count
ip1	ipA	80	TOR	3
ip2	ipA	53	DNS	2
...	...	...	...	...
ipA	ip4	80	TOR	1
ipB	ip4	53	DNS	1
ipZ	ip4	53	DNS	1

Tabela 4 – Exemplo de conjunto de *clusters* após receber agrupamento

Prosseguindo no Algoritmo, da linha 9 até a 16, ocorre uma verificação se existe algum *cluster* que possui um valor de *count* maior ou igual ao valor definido para *min_size*. O intuito dessa verificação é checar se o *cluster* conseguiu atingir o tamanho mínimo para ser transferido para um conjunto resultante. Então, aqueles que atingem o tamanho mínimo são reportados e o restante permanece no conjunto inicial. Logo após, é feita a remoção da generalização existente nos *clusters* que permaneceram. Esses alertas, vão

receber seus valores-base sem generalização (caso tenham sofrido algum tipo de generalização anteriormente) e novamente todos os *clusters* vão receber valor 1 para *count*. O intuito de remover a generalização existente é uma das estratégias para evitar supergeneralização dos valores *clusters*.

Após transferir os *clusters*, o próximo passo do algoritmo é aplicar a generalização de valores. Porém, é importante enfatizar que o algoritmo generaliza somente os valores de um atributo por vez. Portanto, na linha 17 do algoritmo é realizada a seleção de qual atributo generalizar com base em uma heurística apresentada por Julisch et al. em [29], cujo intuito é reduzir a supergeneralização dos valores.

Posteriormente, nas linhas 18 até 20, o algoritmo realiza o processo de generalização de todos os valores do atributo selecionado anteriormente e representado por A_S. Portanto, na linha 18 é definido o laço de repetição que percorre todos os *clusters* no conjunto. Para cada um destes *clusters*, seu valor no atributo A_S será substituído pelo valor superior definido na hierarquia de generalização, independente do nível de generalização que já tenha recebido. Para exemplificar, a partir das Tabelas 4 e 5 é possível observar o antes e depois do conjunto de *clusters* após receber a generalização sobre os valores da coluna IP_origem, considerando a hierarquia de generalização ilustrada na Figura 3.

IP_origem	IP_destino	Porta_destino	Assinatura	count
IntraNet 1	ipA	80	TOR	3
IntraNet 1	ipA	53	DNS	2
...	...	...	...	...
NET A	ip4	80	TOR	1
NET A	ip4	53	DNS	1
NET B	ip4	53	DNS	1

Tabela 5 – Exemplo de conjunto de *clusters* após receber generalização

4 ESTUDO DE CASO

Buscando aplicar e validar o estudo deste trabalho, foi realizado um estudo de caso em uma organização acadêmica que precisa lidar frequentemente com um grande volume de alertas. Portanto, foi fornecida uma amostra de alertas geradas por um de seus IDS. O estudo de caso deste trabalho está dividido em três partes:

- Descrição dos dados que foram utilizados na aplicação
- Descrição da aplicação e particularidades definidas para o algoritmo utilizado
- Resultados e discussões obtidas através das análises realizadas

4.1 Pré-Processamento

O conjunto de dados que nos foi apresentado continha todos os atributos necessários para a execução da proposta. Os alertas relativos a ele foram coletados por um dos sensores que a organização tem implantado. Juntamente com o conjunto de alertas, foi fornecida uma *blocklist*, que são listas públicas que podem ser fornecidas por diferentes instituições que tem reconhecida reputação e atuam na área de segurança. No nosso caso, foi utilizada uma *blocklist* disponibilizada pelo Serpro, contendo um total 1.876 endereços IP distintos que já possuem históricos de envolvimento com tentativas de ataques.

O conjunto de dados possui um total de 17.094 alertas gerados pelo IDS no mesmo dia, mais especificamente em um intervalo de 35 minutos. Inicialmente, continha um total de 32 atributos, porém, somente 5 atributos foram utilizados. Os atributos selecionados foram:

- *eventType*: esse atributo informa a direção na qual o ataque está acontecendo. Ele pode ser preenchido com o valor “Origem”, onde nesse caso está informando que o possível ataque está acontecendo das máquinas internas para externas à rede. Outro valor que pode estar presente é “Destino”, significando que o possível ataque tem como origem uma máquina externa à rede e o destino está sendo uma máquina interna da rede. A Figura 4 traz uma ilustração dos fluxos.
- *source_ip*: endereço IP de onde está partindo a suspeita de ataque.
- *dst_ip*: endereço IP que está sofrendo o possível ataque.
- *dst_port*: porta na qual o alvo está recebendo o tráfego suspeito.
- *signature:* assinatura que descreve o padrão de um evento que fez disparar o alarme.

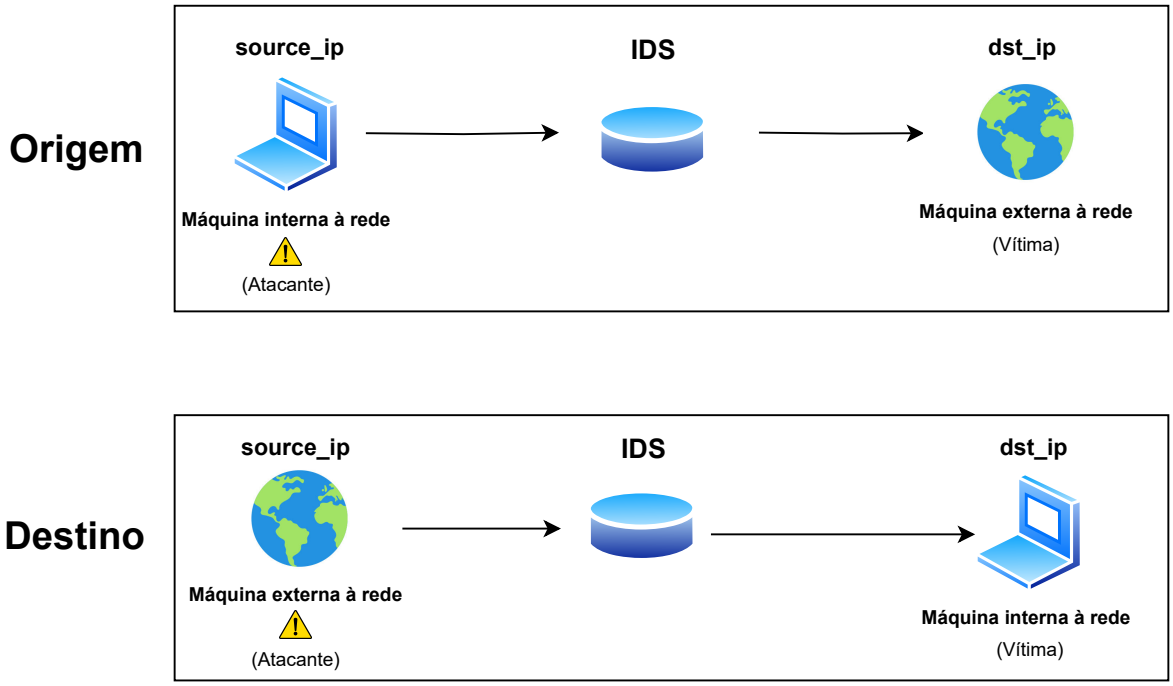

Figura 4 – Direções do ataque descritas pelo atributo *eventType*

Com esses atributos selecionados, outra tarefa de pré-processamento foi criar dois conjuntos de dados chamados *in_blocklist* e *not_in_blocklist*. Estes dois conjuntos foram criados e populados a partir de uma análise de quais alarmes continham seu(s) endereços IP presente(s) na *blocklist*, verificando tanto o atributo *source_ip*, quanto *dst_ip*. Sendo assim, todos os alertas que tinham IP(s) na *blocklist* foram copiados para um conjunto novo de dados e os alarmes que não tinham nenhum dos seus IP(s) na *blocklist* foram copiados para outro conjunto de dados como ilustra a figura 5.

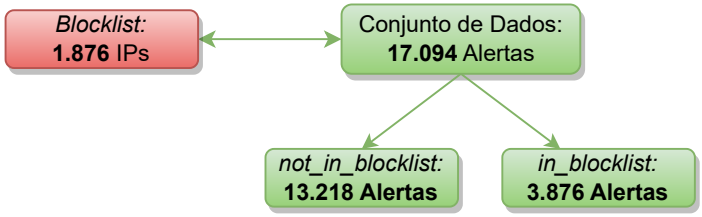

Figura 5 – Pré-processamento de divisão de alertas com base na *blocklist*

O intuito deste pré-processamento foi aplicar o algoritmo em cada um dos conjuntos de dados para notar semelhanças de situações e cenários entre ambos. Além disso, oferece a possibilidade de analisar padrões de comportamentos para definir regras, já que o conjunto de dados que possui endereços IP presentes na *blocklist* são altamente suspeitos.

Por fim, devido a sigilosidade das informações do conjunto de dados, foi necessário realizar um processamento de anonimização dessas informações. Desta forma, valores reais de IP e assinaturas foram substituídos por outras informações mais genéricas que não comprometessem a privacidade da instituição que forneceu os dados.

4.2 Aplicação do Estudo de Caso

A aplicação do estudo de caso que a atual seção apresenta foi realizada com base na proposta apresentada no Capítulo 3. Para tanto, foi necessário definir uma hierarquia de generalização que se comportaria melhor para o conjunto de dados usado no presente estudo de caso. Ao realizar a aplicação do estudo de caso, foi notado que alguns *clusters* não conseguiam agrupar uma quantidade mínima de alertas conforme especificado pelo parâmetro *min_size*, fazendo com que o algoritmo nunca parasse de executar enquanto buscava atingir esse valor mínimo. Por conta disso, foi necessário realizar um ajuste para o algoritmo finalizar quando não fosse mais possível realizar generalização e agrupamentos mesmo que os *clusters* não alcançassem a quantidade mínima de alertas definida.

A hierarquia de generalização foi definida para cada um dos atributos — exceto o atributo *'eventType'* — onde cada um deles possuem níveis de generalizações diferentes. A figura 6 ilustra a hierarquia de generalização definida para os IPs.

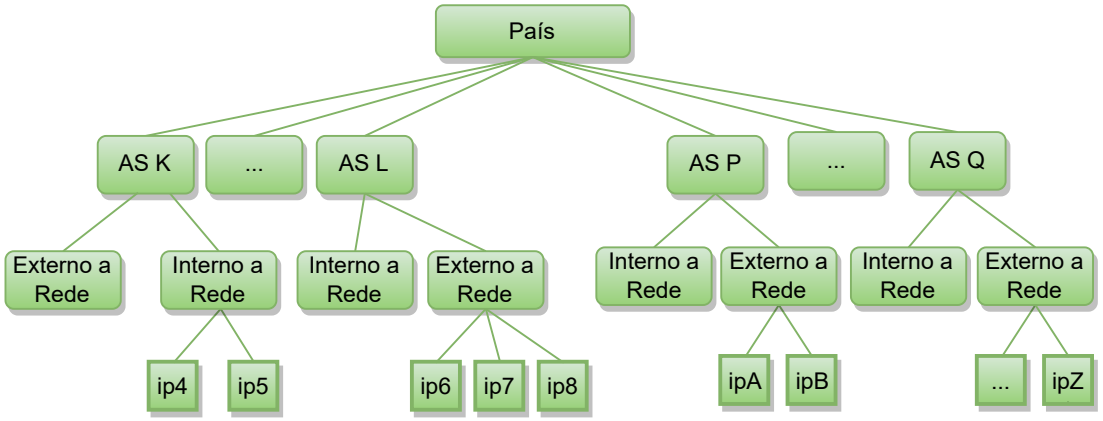

Figura 6 – Hierarquia de generalização para endereços IP utilizada na aplicação do estudo de caso

Para os atributos de IP — tanto o de origem quanto de destino — foram definidos três níveis de generalização. A partir do endereço IP, o primeiro nível de abstração definido é verificar se o IP é de uma máquina interna ou externa a rede, transformando o IP em um desses dois valores. O próximo nível de abstração substitui o valor atual que o alerta possui no momento pelo ASN e as informações do AS correspondente para aquele IP. E por

fim, o último nível de abstração se aplica ao país no qual aquele IP pertence, realizando a substituição do AS atual no valor do alerta, pelo nome do país daquele IP e ASN.

Para realizar a generalização dos endereços IP, foi necessário montar uma base de dados que, para cada IP, foi associado seu ASN/AS juntamente com seu país, para poder então fazer a consulta quando fosse necessário no algoritmo. Para obter os dados necessários de cada IP, foi utilizado uma *API* chamada *"IP Geolocation API"*, que dado o endereço de IP na requisição, as informações públicas necessárias para criar a base de dados eram retornadas.

A hierarquia de generalização das portas e das assinaturas apresentam menos níveis. Para as portas, foi definido apenas um nível de generalização, separando em duas faixas de portas: "menores que 1023" e "maiores que 1023". As portas menores que 1023 são as portas classificadas como *well-known ports*, ou seja, portas mais conhecidas e utilizadas para alguns serviços padrão como FTP, *Telnet*, SSH, entre outras. Já entre as portas com número acima de 1023, podem ser encontradas portas registradas para empresas específicas, softwares específicos, entre outros.

Por fim, para criar a generalização das assinaturas, foram realizadas as análises das presentes no conjunto de dados em busca de um padrão. Ao final foi observado que dentre essas assinaturas, algumas não se encaixavam em nenhum grupo, sendo assim, generalizá-las não iria surtir nenhum efeito. No entanto, para outras assinaturas, foram observados alguns padrões que contribuem para a generalização das mesmas e consequentemente contribuindo com a análise final, como, por exemplo, assinaturas que referenciam ameaças de DNS. A Figura 7 ilustra os diferentes níveis de generalização definidos.

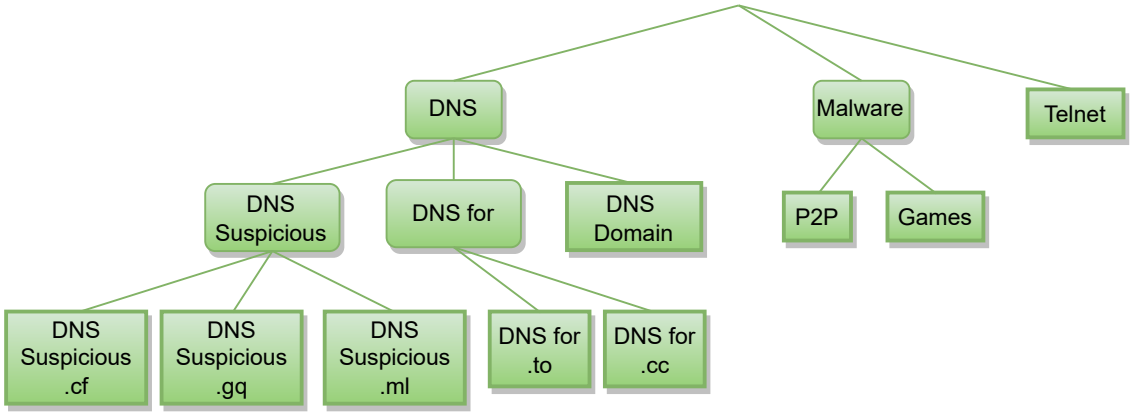

Figura 7 – Hierarquia de generalização para as assinaturas

Após definir as hierarquias de generalização, o outro parâmetro que o algoritmo necessita é um valor para *min_size* — explicado melhor na seção 4.2.3 — para definir o tamanho mínimo de cada *cluster*. Com esse objetivo, foram escolhidos 4 valores distintos:

10, 50, 100 e 250. Ao utilizar esta faixa de valores, podemos entender como a clusterização opera dependendo da variação do *min_size*, o que é importante para auxiliar na escolha dos valores mais adequados para esse parâmetro.

4.3 Resultados e Discussão

Os resultados apresentados nesta seção estão organizados e divididos por diferentes valores de *min_size*. O intuito de expor os resultados utilizando diferentes valores para *min_size* para os mesmos conjuntos de entrada é enfatizar o impacto que esse parâmetro tem sobre os resultados. Além disso, como mencionado na seção 4.1, foi realizado um pré-processamento do conjunto de alertas original, dividindo-o em dois outros conjuntos de alertas chamados *in_blocklist* e *not_in_blocklist*. Os resultados de ambos conjuntos gerados pelo algoritmo serão apresentados em sequência.

4.3.1 Resultados Obtidos

Considerando que as tabelas de resultados possuem muitas linhas, serão apresentadas aqui somente uma amostra delas. Também é importante salientar que cada uma das linhas das tabelas representam um *cluster* e cada uma das colunas representam um atributo. A coluna *count* informa a quantidade de alertas agrupados no *cluster*.

A Tabela 6 apresenta o conjunto de alertas obtidos através do processo de clusterização com *min_size* = 10. Para esse valor, foram obtidos no conjunto de alertas *in_blocklist* um total de 77 *clusters*. Dentre esses *clusters*, 2 deles não conseguiram atingir a quantidade mínima de alertas especificado por *min_size* mesmo com a maior generalização possível.

eventType	**source_ip**	**dst_ip**	**dst_port**	**signature**	**count**
Destino	Externa à Rede	Interno à Rede	22	TOR Traffic group 2	23
Destino	Externa à Rede	Interno à Rede	22	TOR Traffic group 26	23
Destino	Externa à Rede	Interno à Rede	22	TOR Traffic group 27	15
Destino	Externa à Rede	Interno à Rede	22	Tor Relay/Router group 2	29
Destino	Externa à Rede	Interno à Rede	22	Tor Relay/Router group 23	14
Destino	Externa à Rede	Interno à Rede	22	Tor Relay/Router group 26	23
Destino	Externa à Rede	Interno à Rede	80	TOR Traffic group 42	28
Destino	Externa à Rede	Interno à Rede	80	TOR Traffic group 43	17
Destino	Externa à Rede	Interno à Rede	80	TOR Traffic group 47	25
Destino	Externa à Rede	Interno à Rede	443	TOR Traffic group 42	90
Destino	Externa à Rede	Interno à Rede	443	TOR Traffic group 43	41
Destino	Externa à Rede	Interno à Rede	443	Tor Relay/Router group 23	12
Destino	Externa à Rede	Interno à Rede	443	Tor Relay/Router group 286	19
Destino	Externa à Rede	Interno à Rede	1604	SCAN VMware	104
Destino	Externa à Rede	ip849	80	TOR Traffic	24
Destino	Externa à Rede	ip849	80	Tor Relay/Router	28
Destino	Externa à Rede	ip1118	443	Tor Relay/Router	12
Destino	Externa à Rede	ip889	maior que 1023	TOR Traffic	89
Destino	Externa à Rede	ip889	maior que 1023	Tor Relay/Router	106
Destino	Externa à Rede	Interno à Rede	maior que 1023	TOR Traffic group 42	15
Destino	Externa à Rede	Interno à Rede	maior que 1023	Tor Relay/Router group 42	13
Destino	Externa à Rede	Interno à Rede	maior que 1023	TOR Traffic	14
Destino	China	Brazil	menor que 1023	POP3S	1
Destino	Germany	Brazil	menor que 1023	Rede TOR	2

Tabela 6 – Resultado ***in_blocklist*** com IPs anonimizados e *min_size* = 10

Como exemplo de interpretação de um *cluster*, a primeira linha da tabela representa um *cluster* onde todos 23 alertas reportam eventos que tiveram como origem um computador externo à rede e alvo um computador interno à rede. Esses alertas tiveram todos a porta 22 como destino e a assinatura "TOR Traffic group 2".

No conjunto de dados *in_blocklist*, foi possível notar muitas ocorrências de assinatura que descrevem eventos relacionados ao uso da rede Tor. As ocorrências tiveram predominância nas portas 22, 80 e 443 — portas comumente utilizadas pelos protocolos SSH, HTTP e HTTPS respectivamente — e vindo de um IP externo para um interno à rede. Sendo assim, por ser um padrão de comportamento predominante em conjunto de dados que possui alertas com IP na *blocklist*, pode ser considerado um padrão que merece ser analisado com mais atenção em outros conjuntos de alertas que apresentarem o mesmo padrão entre seus alertas.

Dentre o grupo de ataques que tinham assinaturas mencionando o uso de rede Tor que por si só já é atrativo para atacantes — já que pode ser um meio dos atacantes se esconderem por conta da arquitetura dessa rede — também há eventos direcionados a portas acima de 1023, ou seja, portas que não necessariamente são bem conhecidas. Como são eventos relacionados a protocolos não tão importantes como HTTP ou SSH, isso acaba ficando em outro grupo. Cabe destacar que entre esses eventos temos grande quantidade de alertas voltadas a um endereço IP específico sendo o endereço de *ip889* e isso faz com que esse endereço IP da rede interna também seja um IP que mereça atenção em outras análises já que aparentemente é um alvo bastante perseguido entre os atacantes.

Outro padrão que chama atenção são as ocorrências relacionadas à assinatura *SCANVMware* direcionadas à porta 1604. Esse padrão chama atenção pelo fato do volume de ocorrências desse tipo, representado pelo *cluster* com um dos maiores valores em *count*. Por fim, dois *clusters* que estão presentes e não atingiram o valor mínimo de 10 alertas acabam sofrendo uma generalização maior e traz dados um pouco mais abstratos ao nível de países.

eventType	source_ip	dst_ip	dst_port	signature	count
Destino	ip184	ip215	5992	Tor Relay/Router group 273	100
Destino	ip184	ip215	10567	Tor Relay/Router group 273	31
Destino	ip184	ip215	14069	Tor Relay/Router group 273	91
Destino	ip184	ip215	21287	Tor Relay/Router group 273	16
Destino	ip184	ip215	29462	Tor Relay/Router group 273	48
Destino	ip183	ip215	5925	Tor Relay/Router group 306	26
Destino	ip183	ip215	6469	Tor Relay/Router group 306	38
Destino	ip183	ip215	7581	Tor Relay/Router group 306	16
Destino	ip183	ip215	11609	Tor Relay/Router group 306	17
Destino	ip183	ip215	13212	Tor Relay/Router group 306	16
Destino	ip183	ip215	13432	Tor Relay/Router group 306	84
Destino	ip0	Interno à Rede	22	Tor Relay/Router group 109	2853
Origem	ip96	ip1039	53	DNS Query to	46
Origem	ip96	ip1039	53	DNS Query domain	48
Origem	ip96	ip1026	53	DNS Query cc	34
Origem	ip96	ip1026	53	DNS Query su	26
Origem	ip96	ip1026	53	DNS Query to	169
Origem	ip96	ip1026	53	DNS Query domain	253
Origem	ip96	ip1026	53	DNS Query to .world	14
Origem	ip17	ip1588	53	DNS Query to	28
Origem	ip17	ip773	53	DNS Query cc	12
Origem	ip17	ip1581	53	DNS Query to	40
Origem	ip17	ip745	53	DNS Query cc	12
Origem	ip17	ip1577	53	DNS Query to .world	24
Origem	ip17	ip738	53	DNS Query to	20
Destino	ip143	Interno à Rede	1604	SCAN VMware	1591
Origem	Brazil	Brazil	maior que 1023	Violação de Direitos Autorais/Filmes/Games	2
Origem	Brazil	Canada	menor que 1023	DNS	2
Origem	Brazil	United States	menor que 1023	Malware	1
Destino	Hong Kong	Brazil	menor que 1023	IMAP	1
Destino	Hong Kong	Brazil	menor que 1023	POP3S	1
Destino	Netherlands	Brazil	maior que 1023	Scan on Non-standard Port	3
Destino	Netherlands	Brazil	maior que 1023	Rede TOR	1
Destino	Netherlands	Brazil	menor que 1023	Rede TOR	2
Destino	Portugal	Brazil	menor que 1023	Scan on Non-standard Port	1
Destino	United States	Brazil	maior que 1023	Malware	2

Tabela 7 – Resultado ***not_in_blocklist*** com IPs anonimizados e *min_size* = 10

O conjunto de alertas *not_in_blocklist* é maior, portanto, após a clusterização, foram atingidos o valor de 244 *clusters* — sendo um valor bem inferior ao valor original do conjunto de alertas de entrada — tendo 10 *clusters* que não conseguiram atingir o valor de *min_size*.

Analisando os resultados gerados — descritos na Tabela 7 — é possível notar a presença de muitos *clusters* com assinaturas relacionadas a Tor. Por mais que são em portas diferentes do outro conjunto de *clusters* apresentado anteriormente, ainda sim, aparecem com uma frequência muito grande. Portanto, é um cenário que deve ser priorizado para uma análise mais aprofundada e para entender o porquê da grande quantidade de alarmes desse tipo.

Outro ponto observado entre os resultados foi a alta frequência de *clusters* descrevendo ocorrências com assinaturas relacionadas a DNS. Essas ocorrências partiram de máquinas com IPs específicos de dentro da rede, como, por exemplo, ip96 e ip17. Esses dois endereços IPs aparecem com uma frequência que chama atenção, significando que as máquinas internas à rede que estava utilizando esse IP merecem atenção e podem estar portando algum vírus que possa estar tentando se comunicar com outras máquinas externas.

Por fim, uma situação que também chama atenção é as 1591 ocorrências de alertas com assinaturas relacionadas a *SCAN VMware* na porta 1604, que também foi um dos pontos ressaltados na análise do conjunto de *clusters in_blocklist*.

Após apresentação dos resultados com *min_size* = 10, as Tabelas 8 e 9 contém os resultados dos processamentos *min_size* = 50 para *in_blocklist* e *not_in_blocklist* respectivamente. A partir do conjunto *in_blocklist*, foram obtidos um conjunto resultante com 19 *clusters*, onde dentre eles, apenas 1 não atingiu o valor mínimo de alertas para cada *cluster*.

eventType	**source_ip**	**dst_ip**	**dst_port**	**signature**	**count**
Destino	Externa à Rede	Interno à Rede	22	TOR Traffic group 57	58
Destino	Externa à Rede	Interno à Rede	22	Tor Relay/Router group 57	58
Destino	Externa à Rede	Interno à Rede	443	TOR Traffic group 42	90
Destino	Externa à Rede	Interno à Rede	443	TOR Traffic group 46	61
Destino	Externa à Rede	Interno à Rede	443	TOR Traffic group 47	120
Destino	Externa à Rede	Interno à Rede	443	TOR Traffic group 48	118
Destino	Externa à Rede	Interno à Rede	443	Tor Relay/Router group 287	102
Destino	Externa à Rede	Interno à Rede	443	Tor Relay/Router group 42	87
Destino	Externa à Rede	Interno à Rede	443	Tor Relay/Router group 46	61
Destino	Externa à Rede	Interno à Rede	443	Tor Relay/Router group 47	100
Destino	Externa à Rede	Interno à Rede	443	Tor Relay/Router group 48	126
Destino	Externa à Rede	Interno à Rede	1604	SCAN VMware	104
Destino	Externa à Rede	ip889	maior que 1023	TOR Traffic	89
Destino	Externa à Rede	ip889	maior que 1023	Tor Relay/Router	106
Destino	Externa à Rede	Interno à Rede	maior que 1023	TOR Traffic	72
Destino	Externa à Rede	Interno à Rede	maior que 1023	Tor Relay/Router	78
Destino	Externa à Rede	Interno à Rede	menor que 1023	TOR Traffic	113
Destino	Externa à Rede	Interno à Rede	menor que 1023	Tor Relay/Router	126
Destino	China	Brazil	menor que 1023	POP3S	1

Tabela 8 – Resultado ***in_blocklist*** com IPs anonimizados e *min_size* = 50

Analisando a Tabela 8, mesmo com *clusters* diferentes da análise das tabelas anteriores, é possível destacar a predominância de alertas envolvendo assinaturas Tor. Porém, dessa vez, é importante destacar a predominância da porta 443 entre os *clusters*. A partir de então, esse comportamento de um IP externo à rede, direcionado a um IP interno à rede na porta 443 e relacionado a uma assinatura do tipo Tor é uma situação que merece atenção em uma análise de outros conjuntos de dados.

Outras duas situações foram destacadas na análise das tabelas anteriores e continuam presentes aqui. Uma delas é a presença de 104 alertas envolvendo a assinatura *SCAN VMware*, e a outra é a quantidade de ocorrências partindo de um IP externo à rede sobre o IP *ip889* com diferentes assinaturas envolvendo Tor.

eventType	**source_ip**	**dst_ip**	**dst_port**	**signature**	**count**
Destino	ip184	ip215	5992	Tor Relay/Router group 273	100
Destino	ip183	ip215	5659	Tor Relay/Router group 306	53
Destino	ip183	ip215	13432	Tor Relay/Router group 306	84
Origem	ip96	ip1026	53	DNS Query to	169
Origem	ip215	ip1559	53	DNS Query to	104
Origem	ip215	ip991	53	DNS Query to	68
Origem	ip134	ip1591	53	DNS Query for TOR	106
Origem	ip134	ip868	53	DNS Query for TOR	95
Origem	ip17	ip1582	53	DNS Query to .world	104
Origem	ip17	ip1678	53	DNS Query domain	151
Origem	ip17	ip1146	53	DNS Query domain	170
Origem	ip17	ip2185	53	DNS Query domain	153
Origem	ip17	ip1056	53	DNS Query to	74
Origem	ip17	ip1583	53	DNS Query to .world	78
Destino	ip144	Interno à Rede	1604	SCAN VMware	247
Destino	ip143	Interno à Rede	1604	SCAN VMware	1591
Origem	ip17	Externo à Rede	53	DNS Query cc	202
Origem	ip17	Externo à Rede	53	DNS Query su	159
Origem	ip17	Externo à Rede	53	DNS Query to	497
Origem	ip17	Externo à Rede	53	DNS Susp cf	51
Origem	ip17	Externo à Rede	53	DNS Susp ml	58
Destino	ip0	Interno à Rede	22	Tor Relay/Router group 109	2905
Destino	ip385	ip96	maior que 1023	Tor Relay/Router group 178	86
Destino	ip184	ip215	maior que 1023	Tor Relay/Router group 273	274
Destino	ip183	ip215	maior que 1023	Tor Relay/Router group 306	728
Destino	ip388	ip96	maior que 1023	Tor Relay/Router group 432	70
Origem	Interno à Rede	Externo à Rede	maior que 1023	Bad Login	134
Origem	Brazil	United States	menor que 1023	Rede TOR	14
Destino	United States	Brazil	maior que 1023	Malware	2
Origem	Brazil	United States	menor que 1023	Malware	1

Tabela 9 – Resultado ***not_in_blocklist*** com IPs anonimizados e *min_size* = 50

No resultado gerado para *not_in_blocklist* exibido na Tabela 9 foram obtidos 63 *clusters*, onde dentre esse total, 13 deles ficaram com tamanho inferior ao *min_size*, recebendo um nível maior de generalização para os valores. Uma amostra de exemplos para os *clusters* que atingiram o máximo de generalização estão nas últimas três linhas da Tabela 9. Essas três linhas descrevem eventos pelos países do qual o IP está localizado.

Ao analisar os resultados apresentados na Tabela 9, foi possível notar que com o valor 50 para *min_size*, as predominâncias de alertas com assinaturas relacionadas a Tor e DNS permaneceram. Por mais que os alertas relacionados a Tor ficaram um pouco mais agrupados do que no resultado gerado com *min_size* = 10, ainda sim continuaram predominantes no resultado com *min_size* = 50.

Além dos padrões já enfatizados, há outro *cluster* que chama bastante atenção no conjunto apresentado pela Tabela 9. Ele descreve uma tentativa de ataque vindo de fora para dentro da rede, onde o IP de origem é representado por *ip0*, direcionado a diferentes IPs internos na porta 22 e relacionado a assinatura Tor. Esse *cluster* chama bastante atenção por conta da sua alta frequência, apresentando um total de 2905 ocorrências. Portanto, esse IP *ip0* merece ser priorizado em uma análise mais aprofundada, para entender o porquê de tantas ocorrências partindo dele em uma porta específica e diferentes alvos.

Seguindo a análise, agora são apresentados os resultados obtidos no processamento do conjunto *in_blocklist* com *min_size* = 100. Foram obtidos 8 *clusters*, onde 2 deles não conseguiram atingir o valor mínimo para *min_size*. Considerando valores maiores para *min_size*, a abstração e o volume de *clusters* são menores, porém com maior número de alertas agrupados. Entretanto, ainda sim é possível notar as ocorrências relacionadas a assinaturas de Tor e DNS, sendo mais predominantes entre os *clusters* apresentados pelas Tabelas 10 e 11.

eventType	**source_ip**	**dst_ip**	**dst_port**	**signature**	**count**
Destino	Externa à Rede	Interno à Rede	443	TOR Traffic group 47	504
Destino	Externa à Rede	Interno à Rede	443	TOR Traffic group 48	569
Destino	Externa à Rede	Interno à Rede	443	Tor Relay/Router group 287	438
Destino	Externa à Rede	Interno à Rede	443	Tor Relay/Router group 48	550
Origem	Interno à Rede	Externo à Rede	53	DNS Query to	425
Origem	Interno à Rede	Externo à Rede	53	DNS Query su	269
Origem	Interno à Rede	Externa à Rede	53	DNS Query domain	294
Destino	Externa à Rede	Interno à Rede	menor que 1023	Tor Relay/Router	529
Destino	China	Brazil	menor que 1023	POP3S	1
Destino	Germany	Brazil	maior que 1023	Rede TOR	2

Tabela 10 – Resultado ***in_blocklist*** com IPs anonimizados e *min_size* = 100

eventType	source_ip	dst_ip	dst_port	signature	count
Origem	ip96	ip1026	53	DNS Query to	169
Origem	ip96	ip1026	53	DNS Query domain	253
Origem	ip215	ip1559	53	DNS Query to	104
Origem	ip134	ip1591	53	DNS Query for TOR	106
Destino	ip144	Interno à Rede	1604	SCAN VMware	247
Destino	ip143	Interno à Rede	1604	SCAN VMware	1591
Origem	ip85	ip991	53	DNS Query for TOR	186
Origem	ip17	ip1582	53	DNS Query to .world	104
Origem	ip17	ip1678	53	DNS Query domain	151
Origem	ip17	ip1146	53	DNS Query domain	170
Origem	ip17	ip2185	53	DNS Query domain	153
Origem	ip37	Externo à Rede	53	DNS Query cc	154
Origem	ip134	Externo à Rede	53	DNS Query for TOR	280
Origem	ip17	Externo à Rede	53	DNS Query cc	202
Origem	ip17	Externo à Rede	53	DNS Query su	169
Destino	ip0	Interno à Rede	22	Tor Relay/Router group 109	2905
Destino	ip184	ip215	maior que 1023	Tor Relay/Router group 273	465
Destino	ip183	ip215	maior que 1023	Tor Relay/Router group 306	865
Destino	Externa à Rede	Interno à Rede	13389	Scan on Non-standard Port	101
Origem	Brazil	Australia	menor que 1023	Mining Pool	2
Origem	Brazil	United States	menor que 1023	DNS	3
Origem	Brazil	United States	menor que 1023	Rede TOR	14
Origem	Brazil	Brazil	maior que 1023	Violação de Direitos Autorais/Filmes/Games	2
Origem	Brazil	United States	menor que 1023	Malware	1
Destino	France	Brazil	maior que 1023	Violação de Direitos Autorais/Filmes/Games	2
Destino	Hong Kong	Brazil	menor que 1023	IMAP	1
Destino	Netherlands	Brazil	maior que 1023	SCAN VMware	2
Destino	Portugal	Brazil	menor que 1023	Scan on Non-standard Port	1
Destino	Russia	Brazil	maior que 1023	Scan on Non-standard Port	3
Destino	United States	Brazil	maior que 1023	Malware	2

Tabela 11 – Resultado ***not_in_blocklist*** com IPs anonimizados e *min_size* = 100

Por fim, as Tabelas 12 e 13 apresentam os resultados utilizando o valor de *min_size* = 250, gerando menos *clusters* quando comparado aos resultados anteriores. Para o resultado gerado para *in_blocklist* foram obtidos 9 *clusters*, onde dentre eles 3 não conseguiram atingir a quantidade mínima de alertas no *cluster*.

eventType	source_ip	dst_ip	dst_port	signature	Count
Destino	Externa à Rede	Interno à Rede	22	TOR Traffic	326
Destino	Externa à Rede	Interno à Rede	22	Tor Relay/Router	359
Destino	Externa à Rede	Interno à Rede	80	Tor Relay/Router	286
Destino	Externa à Rede	Interno à Rede	443	TOR Traffic	743
Destino	Externa à Rede	Interno à Rede	443	Tor Relay/Router	944
Destino	Externa à Rede	Interno à Rede	maior que 1023	Rede TOR	370
Destino	China	Brazil	menor que 1023	POP3S	1
Destino	Germany	Brazil	menor que 1023	Rede TOR	1
Destino	Netherlands	Brazil	maior que 1023	SCAN VMware	2

Tabela 12 – Resultado ***in_blocklist*** com IPs anonimizados e *min_size* = 250

eventType	source_ip	dst_ip	dst_port	signature	count
Origem	ip96	ip1026	53	DNS Query domain	253
Destino	ip143	Interno à Rede	1604	SCAN VMware	1591
Origem	ip134	Externo à Rede	53	DNS Query for TOR	627
Origem	ip17	Externo à Rede	53	DNS Query to	587
Origem	ip17	Externo à Rede	53	DNS Query domain	613
Origem	ip17	Externo à Rede	53	DNS Query to .world	331
Destino	ip0	Interno à Rede	22	Tor Relay/Router group 109	2905
Destino	ip184	ip215	maior que 1023	Tor Relay/Router group 273	465
Destino	ip183	ip215	maior que 1023	Tor Relay/Router group 306	865
Origem	Interno à Rede	Externo à Rede	53	DNS Query to	320
Origem	Brazil	Russia	maior que 1023	Bad Login	134
Origem	Brazil	Australia	menor que 1023	Mining Pool	2
Origem	Brazil	United States	menor que 1023	DNS	13
Origem	Brazil	United States	menor que 1023	Rede TOR	14
Origem	Brazil	Brazil	maior que 1023	Violação de Direitos Autorais/Filmes/Games	4
Origem	Brazil	United States	menor que 1023	Malware	1
Destino	France	Brazil	maior que 1023	Violação de Direitos Autorais/Filmes/Games	12
Destino	Hong Kong	Brazil	menor que 1023	IMAP	2
Destino	Hong Kong	Brazil	menor que 1023	POP3S	2
Destino	Netherlands	Brazil	maior que 1023	SCAN VMware	2
Destino	Netherlands	Brazil	maior que 1023	Rede TOR	209
Destino	Netherlands	Brazil	menor que 1023	Rede TOR	205
Destino	Portugal	Brazil	menor que 1023	Scan on Non-standard Port	1
Destino	Russia	Brazil	maior que 1023	Scan on Non-standard Port	96
Destino	United States	Brazil	maior que 1023	Malware	2

Tabela 13 – Resultado ***not_in_blocklist*** com IPs anonimizados e *min_size* = 250

Para processamentos com valores de *min_size* maiores, são geradas menores quantidades de *clusters* com um maior nível de abstração. Caso o analista de segurança de redes tenha um intuito em aumentar o nível de abstração, por exemplo, ao nível de países, um valor maior de min_size maior pode ajudá-los. Entretando, como pode ser visto ao longo dos resultados, se esse valor continuar aumentando, irá ocorrer uma supergeneralização podendo acarretar perdas de informações.

Importante enfatizar que definir um tamanho mínimo para o *cluster* com o hiperparâmetro *min_size* traz a possibilidade de ajuste e refinamento dos resultados de acordo com o intuito do analista de redes. Com a apresentação dos resultados obtidos, é possível notar a diferença de comportamentos dos resultados quando se utiliza um valor para *min_size* muito baixo ou valores mais altos.

Após analisar as características dos resultados aqui apresentados, vale ressaltar que quanto menor o valor de *min_size*, menor é a generalização e por consequência um volume maior de *clusters*. Isso pelo fato dos *clusters* atingirem o tamanho mínimo muito rápido, sendo transferidos mais rápido. Também para valores mais baixos, a quantidade de *clusters* que não conseguem atingir o número mínimo especificado por *min_size* também é baixa, isso pelo fato de não haver tanta dificuldade para conseguir agrupar essa quantidade de alertas em seus *clusters*.

Em contrapartida, definir valores altos para *min_size*, faz com que a quantidade de *clusters* em geral seja menor, porém a generalização é maior, podendo ocorrer uma supergeneralização dos alertas. Além disso, *clusters* que não conseguem atingir o valor mínimo especificados pelo *min_size* podem ser maiores pela dificuldade em encontrar e agrupar alertas suficientes mesmo com o maior nível de generalização possível. A Figura 8 elucida uma comparação da quantidade de *clusters* formados para diferentes valores de *min_size*.

Vale ressaltar também que, comparando especificamente os resultados para *in_blocklist*, tanto com $min\_size = 10$ quanto com $min\_size = 50$, foi possível notar um ponto importante entre eles. Por mais que o conjunto de *clusters* com $min\_size = 50$ seja menor, não houve perda dos padrões predominantes detectados no conjunto de *clusters* gerados com $min\_size = 10$. Portanto, foi possível manter os padrões predominantes que merecem atenção, mesmo com um volume bem menor de dados a serem analisados.

Desse modo, se o analista de segurança deseja um nível maior de abstração e consequentemente um agrupamento maior dos alertas, então define valores maiores para *min_size*, caso contrário, pode ser definidos valores menores para *min_size* buscando uma generalização menor e deixando as informações mais detalhadas.

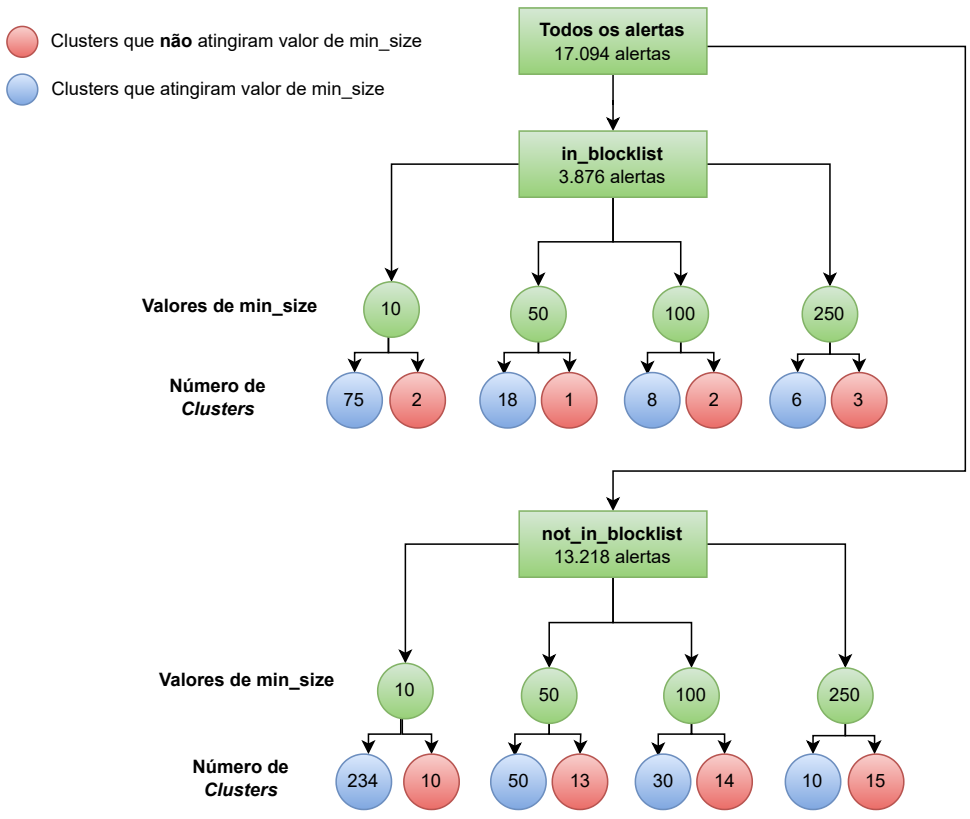

Figura 8 – Diagrama da quantidade de *clusters* gerados para cada valor de *min_size*

4.3.2 Contribuições do Estudo de Caso

Examinando os resultados, foi possível notar algumas contribuições que o presente estudo de caso e aplicação apresenta. Primeiramente, é muito importante destacar a grande redução do volume de alertas como relatado na seção 4.3.1, que apresentou casos com uma população inicial de 13.219 alertas, reduzindo a 244 *clusters* no caso de maior volume ($min\_size = 10$), e 25 *clusters* no caso com menor volume gerado ($min\_size = 250$).

Com a redução do grande volume de alertas e a generalização de valores, é possível entender melhor situações que já estavam presentes entre os dados, porém anteriormente muito mais difíceis de serem visualizadas pelo analista de redes. Entretanto, quando um *cluster* chega a um nível de generalização muito alto, acabam perdendo um pouco de informações, como é o caso de endereços IP generalizados para o país ao qual ele pertence. Em contrapartida, essas perdas de significado acabam facilitando na identificação de padrões.

Sendo assim, após a clusterização, é possível obter uma nova perspectiva sobre os alertas, a qual traz maior facilidade em compreender padrões presentes entre os dados. Como exemplo de padrões, podemos ter padrões de situações críticas de vulnerabilidades não notadas antes, ou até mesmo padrões de alertas que são falsos-positivos e estão somente contribuindo em aumentar o volume de alertas de forma desnecessária.

Portanto, a partir das análises dos *clusters* gerados utilizando a ideia apresentada nesse trabalho, surge a possibilidade do desenvolvimento de outros trabalhos. Alguns exemplos possíveis são realizar a detecção automática de padrões, fazendo com que seja possível criar novas regras de segurança para aprimorar o sistema gerador de alertas, ou até mesmo criar regras para filtrar e reduzir a quantidade de alertas disparados diariamente. Além disso, abre a possibilidade do desenvolvimento de um painel interativo para ajudar a equipe de segurança de redes.

5 CONCLUSÃO

O IDS é um dos dispositivos que vem ajudando administradores de redes no processo de identificar e combater ataques. No entanto, em vários casos os administradores de redes têm que lidar com um volume enorme de alertas disparados diariamente, dificultando as análises que precisam ser realizadas, e consequentemente atrapalhando em ter uma perspectiva do cenário presente na rede em questão.

O presente trabalho apresentou um estudo sobre clusterização de alertas utilizando hierarquias de generalização, a ideia principal do algoritmo AOI (*Attribute-Oriented Induction*) e heurística. O objetivo foi reduzir o grande volume do conjunto de dados dos alertas, perdendo o mínimo possível de informações, e possibilitando visualizar alguns padrões presentes na rede e facilitando a análise que os administradores de rede farão sobre esses alertas.

Também foi realizado um estudo de caso com um conjunto de dados disponibilizados por uma organização acadêmica. Portanto, com esses dados, foi possível realizar experimentos aplicando algoritmo no contexto dos alertas gerados por um dos pontos dessa organização acadêmica.

A partir dos resultados pode-se concluir que foi possível diminuir o grande volume de dados em um número muito menor utilizando os *clusters* e, além disso, apresentar cenários que podem necessitar de maior atenção em uma análise mais profunda pelo analista de segurança. Outro ponto é a possibilidade de definir diferentes valores para o hiperparâmetro *min_size*, o que pode tornar mais customizável o uso do algoritmo dependendo do quanto deve ser generalizado as informações. Além disso, para valores não extremos de hiperparâmetro, os *clusters* apresentaram um equilíbrio entre uma boa redução dos alertas sem perder informações que o analista de segurança precisa.

Como sugestão de trabalhos futuros, conseguir definir uma hierarquia de generalização para as assinaturas de forma mais dinâmica para poder processar outros conjuntos de dados que tiverem diferentes tipos de assinaturas. Além disso, incluir ao processamento mais atributos, como, por exemplo, *timestamp* ou outro que tiver disponível no conjunto de dados utilizado e que possa trazer mais semântica ainda em situações presentes na rede. Outra sugestão seria o estudo de algum algoritmo para definir um valor de hiperparâmetro automaticamente para o analista de segurança.

REFERÊNCIAS

[1] BHATI, N. S.; KHARI, M. Comparative analysis of classification based intrusion detection techniques. In: IEEE. *2021 5th International Conference on Information Systems and Computer Networks (ISCON)*. [S.l.], 2021. p. 1–6.

[2] GOESCHEL, K. Reducing false positives in intrusion detection systems using data-mining techniques utilizing support vector machines, decision trees, and naive bayes for off-line analysis. In: IEEE. *SoutheastCon 2016*. [S.l.], 2016. p. 1–6.

[3] ALMSEIDIN, M. et al. Evaluation of machine learning algorithms for intrusion detection system. In: IEEE. *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*. [S.l.], 2017. p. 000277–000282.

[4] ABUROMMAN, A. A.; REAZ, M. B. I. Survey of learning methods in intrusion detection systems. In: IEEE. *2016 international conference on advances in electrical, electronic and systems engineering (ICAEES)*. [S.l.], 2016. p. 362–365.

[5] VIJ, C.; SAINI, H. Intrusion detection systems: Conceptual study and review. p. 694–700, 2021.

[6] HU, L. et al. False positive elimination in intrusion detection based on clustering. In: IEEE. *2015 12th International conference on fuzzy systems and knowledge discovery (FSKD)*. [S.l.], 2015. p. 519–523.

[7] KHRAISAT, A. et al. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, Springer, v. 2, n. 1, p. 1–22, 2019.

[8] MALEK, Z. S.; TRIVEDI, B.; SHAH, A. User behavior pattern -signature based intrusion detection. In: *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. [S.l.: s.n.], 2020. p. 549–552.

[9] LI, W.; MENG, W.; KWOK, L. F. Surveying trust-based collaborative intrusion detection: State-of-the-art, challenges and future directions. *IEEE Communications Surveys & Tutorials*, IEEE, v. 24, n. 1, p. 280–305, 2021.

[10] ALSUBHI, K.; AL-SHAER, E.; BOUTABA, R. Alert prioritization in intrusion detection systems. In: IEEE. *NOMS 2008-2008 IEEE Network Operations and Management Symposium*. [S.l.], 2008. p. 33–40.

[11] PIETRASZEK, T. Using adaptive alert classification to reduce false positives in intrusion detection. In: SPRINGER. *International workshop on recent advances in intrusion detection*. [S.l.], 2004. p. 102–124.

[12] NARSINGYANI, D.; KALE, O. Optimizing false positive in anomaly based intrusion detection using genetic algorithm. In: IEEE. *2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE)*. [S.l.], 2015. p. 72–77.

[13] BUCZAK, A. L.; GUVEN, E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications surveys & tutorials*, IEEE, v. 18, n. 2, p. 1153–1176, 2015.

[14] AHMED, L. A. H.; HAMAD, Y. A. M. Machine learning techniques for network-based intrusion detection system: A survey paper. In: IEEE. *2021 National Computing Colleges Conference (NCCC)*. [S.l.], 2021. p. 1–7.

[15] ELSHOUSH, H. T. I. An innovative framework for collaborative intrusion alert correlation. In: IEEE. *2014 Science and Information Conference*. [S.l.], 2014. p. 607–614.

[16] PORRAS, P. A.; FONG, M. W.; VALDES, A. A mission-impact-based approach to infosec alarm correlation. In: SPRINGER. *International Workshop on Recent Advances in Intrusion Detection*. [S.l.], 2002. p. 95–114.

[17] CHAKIR, E. M.; MOUGHIT, M.; KHAMLICHI, Y. I. An efficient method for evaluating alerts of intrusion detection systems. In: *2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*. [S.l.: s.n.], 2017. p. 1–6.

[18] KAWAKANI, C. T. *Geração online de hiperalertas com base no histório de estratégias de ataque*. Dissertação (Mestrado) — Universidade Estadual de Londrina, 2017.

[19] FAN, G.; JIHUA, Y.; MIN, Y. Design and implementation of a distributed ids alert aggregation model. In: *2009 4th International Conference on Computer Science Education*. [S.l.: s.n.], 2009. p. 975–980.

[20] VALDES, A.; SKINNER, K. Probabilistic alert correlation. In: SPRINGER. *Recent Advances in Intrusion Detection: 4th International Symposium, RAID 2001 Davis, CA, USA, October 10–12, 2001 Proceedings 4*. [S.l.], 2001. p. 54–68.

[21] DEBAR, H.; WESPI, A. Aggregation and correlation of intrusion-detection alerts. In: SPRINGER. *Recent Advances in Intrusion Detection: 4th International Symposium, RAID 2001 Davis, CA, USA, October 10–12, 2001 Proceedings 4*. [S.l.], 2001. p. 85–103.

[22] SALAH, S.; MACIÁ-FERNÁNDEZ, G.; DÍAZ-VERDEJO, J. E. A model-based survey of alert correlation techniques. *Computer Networks*, Elsevier, v. 57, n. 5, p. 1289–1317, 2013.

[23] MORAES, E. A. *Aplicação de Aprendizado de Máquina Supervisionado e técnicas de correlação na análise de alertas de instrusão*. Dissertação (Mestrado) — Universidade Estadual de Londrina, 2018.

[24] BENFERHAT, S.; BOUDJELIDA, A.; TABIA, K. Revising the outputs of a decision tree with expert knowledge: Application to intrusion detection and alert correlation. In: *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*. [S.l.: s.n.], 2012. v. 1, p. 452–459.

[25] HUBBALLI, N.; SURYANARAYANAN, V. False alarm minimization techniques in signature-based intrusion detection systems: A survey. *Computer Communications*, Elsevier, v. 49, p. 1–17, 2014.

[26] ABOUABDALLA, O. et al. False positive reduction in intrusion detection system: A survey. In: *2009 2nd IEEE International Conference on Broadband Network Multimedia Technology*. [S.l.: s.n.], 2009. p. 463–466.

[27] MITROFANOV, S.; SEMENKIN, E. An approach to training decision trees with the relearning of nodes. In: *2021 International Conference on Information Technologies (InfoTech)*. [S.l.: s.n.], 2021. p. 1–5.

[28] XIONG, S.-W.; LIU, H.-B.; NIU, X.-X. Fuzzy support vector machines based on fcm clustering. In: *2005 International Conference on Machine Learning and Cybernetics*. [S.l.: s.n.], 2005. v. 5, p. 2608–2613 Vol. 5.

[29] JULISCH, K.; DACIER, M. Mining intrusion detection alarms for actionable knowledge. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2002. p. 366–375.