Universidade Federal de Pernambuco
Graduação em Ciência da Computação

Centro de Informática
2008.2

ADS-FUSION: FUSÃO DE DADOS PARA DETECÇÃO DE ANOMALIAS BASEADO NA TEORIA DA EVIDÊNCIA DE DEMPSTER-SHAFER

Trabalho de Graduação

BRUNO FILIPE DE OLIVEIRA LINS

Orientador: Prof. PhD Djamel Sadok (jamel@cin.ufpe.br)
Co-orientado: Prof. MSc. Eduardo Feitosa (efeitosa@gprt.ufpe.br)

Recife,
2008

Universidade Federal de Pernambuco
Graduação em Ciência da Computação

Centro de Informática
2008.2

ADS-FUSION: FUSÃO DE DADOS PARA DETECÇÃO DE ANOMALIAS BASEADO NA TEORIA DA EVIDÊNCIA DE DEMPSTER-SHAFER

Trabalho de Graduação

BRUNO FILIPE DE OLIVEIRA LINS

Projeto de Graduação apresentado no Centro de Informática da Universidade Federal de Pernambuco por Bruno Filipe de Oliveira Lins, orientado pelo Prof. PhD. Djamel Sadok, como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação

Orientador: Prof. PhD Djamel Sadok (jamel@cin.ufpe.br)
Co-orientado: Prof. MSc. Eduardo Feitosa (efeitosa@gprt.ufpe.br)

Recife,
2008

FOLHA DE APROVAÇÃO

ADS-FUSION: FUSÃO DE DADOS PARA DETECÇÃO DE ANOMALIAS BASEADO NA TEORIA DA EVIDÊNCIA DE DEMPSTER-SHAFER

BRUNO FILIPE DE OLIVEIRA LINS

APROVADO EM 01 DE DEZEMBRO DE 2008

BANCA EXAMINADORA:

_____

Prof. *Djamel* Fawzi Hadj *Sadok*, PhD –
UFPE (Orientador)

_____

Prof. Nelson Souto Rosa, Dr –
UFPE (Orientador)

AGRADECIMENTOS

Este agradecimento não é apenas dedicado aos que contribuíram para a realização deste trabalho, mas a todos que fizeram parte desses LONGOS anos de graduação. Gostaria de deixar claro que os nomes citados não estão em ordem de preferência ou importância.

Agradeço aos meus pais, Marcilio Severino Lins e Bernardete Rosa de Oliveira Lins, que tanto se esforçaram que eu sempre pudesse ter do melhor. Aos meus irmãos, Ana Carolina de Oliveira Lins e Hugo Leonardo de Oliveira Lins, e meu sobrinho Eduardo Lucas Lins dos Santos, por todos os momentos de alegria (incluindo as brigas) que vivenciamos todos esses anos.

Agradeço a todos os familiares que sempre me apoiaram nessa caminhada, tios, tias, primos e minha querida avó. Agradeço também aos tios, tias, primas e primos postiços que sempre foram de grande importância na minha vida.

Àqueles amigos da UFPE que me acompanharam por esses anos de muita luta e acima de tudo, muitas FARRAS, sem eles, eu provavelmente não estaria onde estou hoje. Agradeço em especial a: Ademir Carvalho (Do Janga), André Guedes (debão), André Schaffer (bahiano), David Levy (xêxinha), Felipe Cavalcanti (moxinho), Filipe Cesar (Gravatá), Fernando Kakimoto (japa), Francisco Magalhães (chicrito) Henrique Seabra (rick), Jesus Sanchez-Palencia (jê aranha), Rebeka Gomes (rosalvão-ou-beka), Rilter Tavares (rilter seabra), Rodrigo Melo (digão), Thiago Lacerda (xeroso) e Viviane Souza (vivi).

Aos amigos do GPRT, Nadia, Manu, Nacha, Rafa, Mestre, Antonello, Kalil, Arthur, Rover, Chacal, Rodrigo Germano, Ramide, Leo, Guto, entre todos os outros (mas não menos importantes) um muito obrigado. Dentre os membros do GPRT não posso deixar de agradecer de uma forma especial a Professora Judith Kelner por todas as oportunidades, ensinamentos e puxões de orelha, o Professor Djamel Sadok, que além de orientador se mostrou um facilitador nestes anos de GPRT e ao Eduardo Feitosa (feitosinha), meu co-orientador e amigo que tanto fez para a realização e conclusão deste trabalho.

Agradeço também a Julia Machado, minha namorada, por todos os felizes momentos juntos e pela compreensão e companheirismo. Você é parte fundamental deste processo de crescimento.

Por fim, peço desculpas aos importantes amigos não citados (Amigos da Católica, do Inocoop e de Gaibu), vocês são sem sobra de duvidas, muito importantes.

RESUMO

Uma breve análise no atual tráfego das redes, em especial a Internet, revela o contínuo crescimento de anomalias de tráfego causadas por atividades de spoofing, spam, proliferação de vírus e worms, e ataques de negação de serviço. A conseqüência é cada vez mais empresas e instituições sofrem perdas financeiras causadas por este tipo de tráfego.

Este trabalho tem por objetivo estudar o uso da teoria da evidência de Dempster-Shafer para elaboração de um mecanismo de fusão de dados capaz de agregar diferentes ferramentas (sensores) de detecção de tráfego anômalo visando aumentar o grau de certeza sobre determinado evento da rede.

A idéia central deste trabalho é implementar um modelo estatístico que seja capaz de lidar com as incertezas das detecções de anomalia minimizando, desta forma, o número de falsos positivos, aumentando a eficiência da detecção.

ABSTRACT

A brief analysis performed on the current network traffic, specifically on the Internet, reveals that traffic anomalies caused by spoofing, spam, viruses and worms dissemination and denial of service attacks are continually increasing. Due to that, lots of companies and institutions are having undesirable expenses, caused by such type of traffic.

This work aims at studying the usage of the Dempster-Shaffer evidence theory, in order to create a mechanism of data fusion able to aggregate distinct anomalous detection tools (sensors), which increases the assurance level of some network event.

As the main idea, this work implements a statistical model that is able to deal with anomalies' detection uncertainties, aiming at reducing the false positive rate and therefore increasing the detection efficiency.

Sumário

Índice de Figuras

Índice de Tabelas

1. INTRODUÇÃO

No atual cenário de segurança das redes de computadores, em especial a Internet, problemas causados por novos ataques, aumento do tráfego e até má configuração dos protocolos de roteamento vem crescendo consideravelmente. Esses problemas são, em geral, conhecidos como anomalias de tráfego [1] e as atividades de detecção e prevenção vão além do tradicional gerenciamento de incidentes de segurança.

Uma anomalia é o desvio de uma condição típica ou normal. Detecção de anomalia parte da premissa de que qualquer coisa fora do comportamento normal é, por definição, anômalo e constitui um ataque [1]. No contexto do tráfego de rede, anomalias podem ser causadas pela proliferação de códigos maliciosos (vírus e worms), ataques de negação de serviço, spam, falhas de roteamento, entre outras. Tais anomalias desperdiçam recursos das redes, causando prejuízos financeiros e degradando o desempenho e a confiabilidade. Muitas vezes, estes prejuízos podem extrapolar os perímetros da rede e afetar parte ou mesmo toda a internet. Desta forma, a identificação eficiente e rápida é um importante desafio para tornar a internet mais segura e eficiente.

Nos últimos anos, enumeras técnicas, ferramentas e soluções de segurança vêm sendo adaptadas, desenvolvidas ou melhoradas com o intuito de detectar, prevenir e minimizar os efeitos causados pelas anomalias do tráfego de rede. Firewalls, regras de filtragem e sistemas antivírus são exemplos de soluções legadas "reprogramadas" para atuar no combate a determinados tipos de anomalias. Contudo, essas soluções têm baixo poder de contra resposta e/ou apresentam resultados não muito eficazes na prevenção de anomalias. Mesmos sistemas especialistas como IDS (*Intrusion Detection Systems*) [2][3][4] e IPS (*Intrusion Prevention Systems*) [5] geram grande quantidade de falsos positivos (tráfego anômalo classificado como normal), inadequação a determinadas anomalias ou mesmo questões de abrangência e escalabilidade.

Atualmente, técnicas para caracterizar o tráfego Internet, métodos para descobrir o tráfego gerado por aplicações, abordagens para desenvolver sistemas de detecção de anomalia mais precisos e soluções específicas para lidar com certos tipos de tráfego não solicitados (por exemplo, spam e P2P) despontam na tentativa de tornar automática, rápida e precisa identificação e redução de anomalias como, por exemplo, técnicas baseadas em modelos

matemáticos [6][7][8] e estatísticos [9][10] e, principalmente, na análise comportamental do tráfego [11][12][13][14].

Contudo, normalmente, essas técnicas operam de forma isolada e não permitem ou não são preparadas para funcionar de modo colaborativo. É neste contexto que técnicas de fusão de dados podem e devem ser usadas para aprimorar e até mesmo agrupar métodos de detecção de anomalias visando uma melhor e mais precisa detecção de anomalias na rede.

1.1. Objetivo

Este trabalho visa comprovar a eficácia da metodologia de fusão de dados na construção de sistemas de detecção de anomalias. Para tanto, será empregada a teoria da evidência de Dempster-Shafer (TDS) para construção de um protótipo de fusão de dados, chamado ADS-Fusion, capaz de tratar as incertezas e imprecisões de métodos estatísticas e baseados no comportamento do tráfego objetivando maximizar o processo de detecção de anomalias.

A idéia central do trabalho é permitir a fusão das saídas de sensores de anomalias distribuídos pela rede visando incrementar a eficiência da detecção de ataques e, conseqüentemente, aumentar a probabilidade de detecção e diminuir o número de alertas falsos.

Para assegurar a robustez, precisão e eficiência do protótipo, serão realizados testes de validação em ambiente controlado com a injeção de anomalias ao trafego normal.

1.2. Estrutura do Documento

O restante deste trabalho está organizado da seguinte forma.

O capítulo 2 introduz os conceitos básicos relacionados à fusão de dados e a teoria da evidência de Dempster-Shafer. No capítulo 3 são apresentados alguns trabalhos relacionados encontrados na literatura que fazem uso da TDS para fusão de dados, apontando suas vantagens e desvantagens.

O capítulo 4 descreve o protótipo ADS-Fusion, seus componentes e seu funcionamento. No capítulo 5 são apresentados testes e resultados. Por fim, conclusões e trabalhos futuros serão expostos no capítulo 6.

2. CONCEITOS BÁSICOS

Este capítulo está dividido em duas seções onde são descritos os conceitos básicos empregados no presente trabalho. A primeira tem como objetivo introduzir conceitos básicos da fusão de dados e a segunda expõe os conceitos fundamentais a compreensão da teoria da evidência de Dempster-Shafer.

2.1. Fusão de Dados

Fusão de dados pode ser definida como o processo de combinar vários dados a fim de produzir informações mais valorosas para o usuário. Os dados podem provir de uma ou mais fontes. As fontes podem ser semelhantes ou desiguais, desde que o sistema tenha a capacidade de lidar com dados heterogêneos ou conflitantes.

Fusão de dados pode ser utilizada em diversas aplicações como, por exemplo, detecção, reconhecimento, identificação, monitoramento, tomada de decisão, etc. Estes objetivos podem ser encontradas em muitos domínios, tal como robótica, medicina, pesquisas espaciais, entre outros.

O uso de um eficiente esquema de fusão pode gerar vantagens significativas como:

- Melhoria na confiança nas decisões devido ao uso de informações extras;
- Melhoria do desempenho de contramedidas;
- Melhoria no desempenho em condições adversas.

Segundo [15], processos de fusão são freqüentemente classificados como de baixo, intermediário ou alto nível de fusão, dependendo do processamento ou fase em que se realiza a fusão.

- **Baixo Nível** - combina várias fontes de dados brutos para produzir novos dados brutos.
- **Nível Intermediário** – combina várias fontes de dados brutos para produzir inferências lógicas ou padrões de características.
- **Alto Nível** – combina dados oriundos de fontes especialistas (*experts*) que geram decisões a serem combinadas ou informações relevantes para a tomada da decisão.

Usar dados de vários sensores combinados para gerar inferências mais precisas pode ser uma ótima solução quando se trabalha com dados imprecisos ou com muitas fontes de análise [16]. Esse tipo de sistema pode ser comparado ao funcionamento normal do ser humano. Vários sensores (tato, olfato, visão, paladar, audição), além da percepção dos fatos, são combinados em nosso cérebro para que possamos tomar nossas decisões e atuar da forma que nosso cérebro considerar adequada.

De forma resumida, a fusão de dados tem como objetivo trabalhar sobre uma vasta gama de dados para associar, correlacionar, estimar e combinar os vários fluxos de dados, gerando, dessa forma, uma informação mais significativa para o estudo.

Muitos são os modelos e arquiteturas propostos para modelar um sistema de fusão de dados. A figura 2.1 ilustra um modelo baseado em [17] que resume as principais funcionalidades dos principais sistemas de fusão de dados.

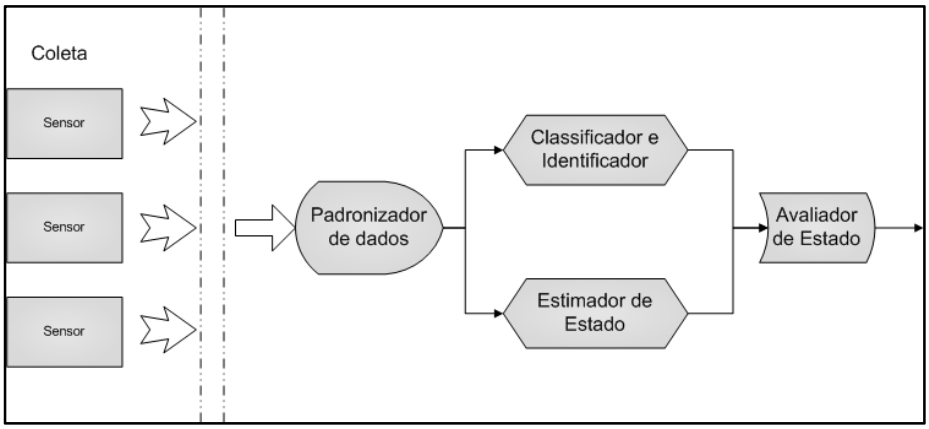

Figura 2.1: Modelo típico de arquitetura de sistema de fusão de dados.

- **Coleta de dados:** Conjunto de sensores responsáveis pela coleta dos dados a serem analisados. Podem ser heterogêneos ou não.
- **Padronizador de dados:** Diferentes sensores podem gerar dados com diferenças temporais, unidades de medidas e espaço, necessitando então serem padronizados.
- **Estimador de Estado:** Com base em modelos de comportamento do sistema e baseado nos dados disponibilizados pelos sensores, este módulo funde os dados e estima o estado do sistema.
- **Classificador e Identificador:** Responsável por identificar e classificar os diferentes eventos monitorados.
- **Avaliador de estado:** Maior nível em um sistema de fusão de dados é responsável por determinar o estado do sistema.

2.1.1. Classificação

Segundo [18], os métodos de fusão de dados podem ser classificados em três categorias: **Modelos Físicos**, **Classificação Paramétrica** e **Algoritmos Cognitivos**. Modelos físicos tentam recriar o ambiente observado e, desta forma, fazer previsões adequadas a partir de inferências (modelos) pré-computadas do ambiente real. Outra característica desses modelos é a tentativa de decompor elementos em descrição de componentes a fim de facilitar seu entendimento. O trabalho desenvolvido por Chatzigiannakis et al [19] é um bom exemplo.

Na classificação paramétrica, os algoritmos trabalham basicamente com uma associação direta dos dados para gerar uma classificação dos dados. Esses podem ser divididos em modelos estatísticos como inferências bayesianas e a teoria de Dempster-Shafer e modelos baseados em informações técnicas e teóricas como redes neurais e modelos baseados em entropias.

Por fim, os algoritmos cognitivos são basicamente métodos que tentam simular o comportamento do cérebro humano. Esses métodos são basicamente subdivididos em métodos *experts* (métodos com uma base de conhecimento específica para determinada área) e métodos baseados nos conjuntos de teorias fuzzy. Um modelo expert bastante utilizado, por exemplo, são os NIDS (*Network Intrusion Detection System*).

2.2. Teoria da Evidência de Dempster-Shafer

A teoria da evidência é um dos modelos mais conhecidos para a representação da incerteza em sistemas baseados em conhecimento. Originada dos trabalhos de Arthur Pentland Dempster [20][21] sobre probabilidade inferior e superior, foi refinada e ampliada por Glenn Shafer [22]. Por isso o nome Teoria da evidência de Dempster-Shafer ou TDS.

O foco da TDS é solucionar os problemas encontrados em modelar a incerteza quando se trabalha com métodos meramente probabilísticos. A teoria de Dempster-Shafer mostrasse bastante interessante, pois, diferentemente das teorias probabilísticas bayesianas, não necessita de um conhecimento a priori das distribuições de probabilidade dos elementos estudados, podendo, desta forma, atribuir valores de crença a subconjuntos das possibilidades e não só aos eventos simples.

Outro fator importante é o fato que a crença não atribuída a nenhum evento em particular, é atribuída ao ambiente e não ao restante das evidências. Além disso, é possível combinar funções de crença gerando novas funções de crença, de forma independente a ordem do surgimento de novas evidências exigindo apenas que as hipóteses primitivas sejam mutuamente exclusivas e exaustivas.

Para melhorar o entendimento da teoria da evidência é necessário compreender certos e importantes conceitos como domínio do problema ou frame de discernimento, probabilidade básica atribuída, intervalo de crença, entre outros. Visando simplificar este processo, serão utilizados exemplos simples acompanhados de explanações.

Exemplo 1. Uma determinada empresa adquiriu um sistema de detecção de anomalias (ADS - *Anomaly Detection System*) para, em tempo real, monitorar a rede e alertar sobre possíveis incidentes de segurança. De acordo com suas definições de projeto, o ADS é formado por vários sensores de anomalias e é, assim, capaz de informar o estado atual da rede, através do seguinte nomenclatura:

- Estado Normal {N}
- Rede sob ataque TCP {T}
- Rede sob ataque UDP {U}
- Rede sob ataque ICMP {I}

É óbvio supor que existem muitos outros problemas e que esta qualificação pouco ajudaria a equipe de segurança da rede em um caso real, mas para efeito de simplicidade, apenas esses quatro estados da rede serão avaliados.

2.2.1. Frame de Discernimento

A TDS pressupõe um conjunto de hipóteses primitivas chamado de **Frame de Discernimento**, **Quadro de Discernimento** ou **Domínio do problema**, denotado por Θ. Para que um conjunto de hipóteses seja considerado um frame de discernimento é necessário que o mesmo apresente algumas características fundamentais:

1 Θ deve ser exaustivo, ou seja, deve conter todas as possíveis hipóteses primitivas.

2 Todas as hipóteses pertencentes ao Θ devem ser mutuamente exclusivas.

Usando o exemplo 1, o frame de discernimento é o seguinte:

Θ = {Normal {N}, Ataque TCP {T}, Ataque UDP {U}, Ataque ICMP {I}}

Como mencionado anteriormente, a TDS permite que a partir de uma única coleção de evidências, vários conjuntos alternativos de hipóteses possam ser derivados (Figura 2.2). Para cada um desses conjuntos é associado um intervalo de confiança chamado de **intervalo de crença**.

Supondo que o ADS seja capaz de identificar ataques simultâneos (ataques compostos), de acordo com o exemplo 1, o ataque seria expresso, por exemplo, pelo conjunto {Ataque TCP, Ataque UDP}.

Supondo ainda que o sistema de detecção observe uma evidência que confirme com um determinado grau um diagnóstico de ataque composto, o conjunto {Ataque TCP, Ataque UDP} receberá certa quantidade de crença proporcional ao grau de confirmação observado da evidência.

Supondo agora que uma nova evidência foi percebida e que ela caracteriza que a rede não está em estado normal {N}, ou seja, essa evidência confirma o conjunto {Ataque TCP, Ataque UDP, Ataque ICMP}. Cada subconjunto, gerado pela disjunção dos elementos de Θ pode ser interpretado como uma nova hipótese, dando origem à $2^{|\Theta|}$possíveis hipóteses, como ilustrado na figura 2.2.

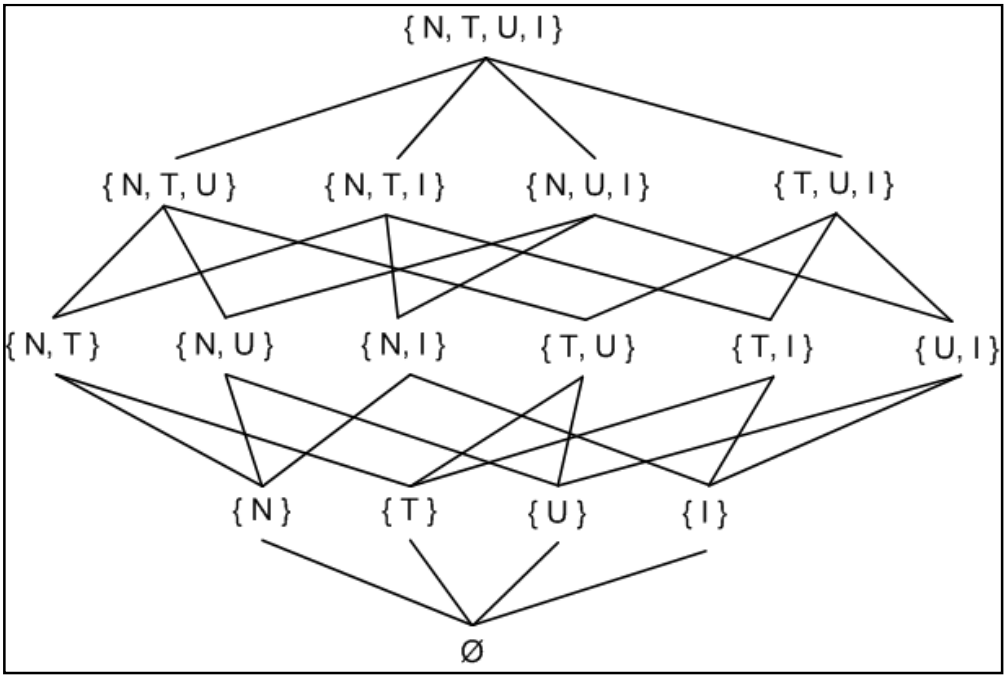

Figura 2.2: Conjunto de todas as possíveis hipóteses do conjunto $\Theta = \{N, T, U, I\}$.

É importante salientar que na maioria dos casos, nem todos os $2^{|\Theta|}$ são de interesse para o caso estudado. Além do fato de que seguindo as propriedades de exaustão e mutua exclusão pode-se então, encontrar um único subconjunto com a resposta correta.

2.2.2. Função de Massa

Na TDS, a indicação de crença em determinada hipótese, dada um evidência, é associada a um valor no intervalo [0,1]. A relevância dada a cada um dos elementos do domínio do problema é representada por uma função denominada de **atribuição de probabilidade básica** (bpa - *basic probability assignment*) ou **função de massa** e notada por m. A função de massa representa a quantidade total de crença na evidência que remete um grupo de hipóteses.

Diferente da probabilidade básica onde a função de densidade associa a cada uma das hipóteses primitivas um número do intervalo $[0,1]$ de maneira que a soma totalize 1, na TDS, a função de massa associa para qualquer subconjunto de Θ e não somente as hipóteses primitivas um valor no intervalo de $[0,1]$ de maneira que a soma dessas atribuições seja 1.

Desta forma, m permite atribuir certa quantidade de crença a qualquer elemento representado na figura 2.1 e não apenas aos elementos primitivos $\{N\}, \{T\}, \{U\}, \{I\}$. A função $m(Ev)$ representa a medida da crença atribuída à evidência Ev, onde Ev pertence a $2^{|\Theta|}$. Essa crença não pode ser subdividida entre os subconjuntos de Ev e não inclui parte da crença atribuída a seus subconjuntos.

A crença básica que não foi atribuída aos subconjuntos de $2^{|\Theta|}$chamada de **crença não atribuída**, notada por $m(\Theta)$ representa a crença que "sobra". Se $m(Ev) = \text{x}$ e m não atribui crença a mais nenhum subconjunto próprio de Θ, então $m(\Theta) = 1 - \text{x}$. O resto da crença é todo atribuído a Θ e não a negação de Ev, como no modelo de Bayes.

Formalmente, para que uma função $m(Ev)$ seja uma *bpa* ela necessita satisfazer:

$$m(\phi) = 0$$

$$m(Ev) \geq 0, \forall\, Ev\ \in 2^{|\Theta|}$$

$$\sum_{Ev\,\epsilon\, 2^{|\Theta|}} m(Ev) \subset 1$$

Exemplo 2. A fim de deixar mais claro, uma possível forma de distribuição de probabilidade seria:

a) $\text{m}(\{\text{N}\}) = 0{,}2$
b) $\text{m}(\{\text{T}\}) = 0{,}4$
c) $\text{m}(\{\text{U}\}) = 0{,}3$
d) $\text{m}(\{\text{I}\}) = 0{,}1$
e) $\text{m}(\text{Ev}) = 0, \forall\, \text{Ev}\ \in 2^{|\Theta|},\ \text{Ev} \neq \{\text{N}\}, \text{Ev} \neq \{\text{T}\}, \text{Ev} \neq \{\text{U}\}, \text{Ev} \neq \{\text{i}\}$

Exemplo 3. Considerando que o ADS, em determinado momento, classifique o estado de nossa rede como NORMAL e suporte essa evidência, chamada de E_1, com 60%. Dada tal evidência nossa função de massa calcula:

a) $\mathrm{m}(\{\mathrm{N}\}) = 0{,}6$

b) $\mathrm{m}(\Theta) = 1 - \mathrm{m}(\{\mathrm{N}\}) = 1 - 0{,}6 = 0{,}4$

Percebe-se que Θ contém {N} e também seu complemento ({T}, {U}, {I}). Diferente da probabilidade Bayesiana, em TDS não se pode atribuir o restante da evidência ao complemento de E_1. Em outras palavras, não se pode afirmar que a crença não atribuída a E_1 é atribuída a $\overline{E_1}$, pois não existe nenhuma evidência que comprove isso. Sabe-se apenas que é possível suportar E_1 com uma confiança de 0,6.

2.2.3. Função de Crença

Função de crença, denotada por $bel()$, mede o total de crença atribuída a um determinado subconjunto de Θ. Na prática, $bel(Ev)$ é a soma das probabilidades básicas atribuídas a todos os subconjuntos de Ev de Θ. Para obter o total de crença atribuída Ev_1, deve-se adicionar a $m(E_{v1})$ os valores de $m(Ev_2)$, para todo subconjunto próprio Ev_2 de Ev_1.

É importante ressaltar que a TDS permite a representação de um grau de crença zero a todo subconjunto de possibilidades do quadro de discernimento, tendo então:

a) $\mathrm{m}(\Theta) = 1$

b) $\mathrm{m}(\mathrm{Ev}) = 0\,, \forall\, \mathrm{Ev} \neq \Theta$

Esse tipo de representação é conhecido como **Função de crença vacuosa**.

Em uma TDS os principais elementos de estudos são aqueles subconjuntos de Θ cuja probabilidade básica não nula. A esses elementos dar-se o nome de **elementos focais** da função de crença sobre 2^{Θ}. A união de todos os elementos focais, para uma função de crença, é chamada de núcleo.

Exemplo 4. Usando:

a) $\mathrm{m}(\{\mathrm{N}\}) = 0{,}3$

b) $\mathrm{m}(\{\mathrm{T}\}) = 0{,}2$

c) $\mathrm{m}(\{\mathrm{U}\}) = 0{,}1$

d) $\mathrm{m}(\{\mathrm{T}, \mathrm{U}\}) = 0{,}2$

e) $\mathrm{m}(\Theta) = 0{,}2$

f) $\mathrm{m}(\mathrm{Ev}) = 0, \forall\, \mathrm{Ev} \in 2^{\Theta}, \mathrm{Ev} \neq \{\mathrm{N}\}, \mathrm{Ev} \neq \{\mathrm{T}\}, \mathrm{Ev} \neq \{\mathrm{U}\}, \mathrm{Ev} \neq \{\mathrm{T}, \mathrm{U}\}$

Então pode-se encontrar a $bel(\{T,U\})$ como sendo $m(\{T,U\}) + m(\{T\}) + m(\{U\})$, logo, $bel(\{T,U\}) = 0{,}1 + 0{,}2 + 0{,}3 = 0{,}6$.

2.2.4. Plausibilidade

A **função de plausibilidade** ou **probabilidade superior**,. $\wp\ell()$, determina a quantidade máxima de crença que pode ser atribuída a um determinado subconjunto de 2^{Θ}. Tem-se que $\wp\ell(Ev)$ representa o mesmo que $1 - bel(\overline{Ev})$. Desde que $bel(Ev) + bel(\overline{Ev}) \leq 1$, tem-se que $bel(Ev) \leq \wp\ell(Ev) \; \forall \; Ev \subseteq \Theta$.

2.2.5. Intervalos de Crença

Como visto anteriormente, a plausibilidade representa o quanto se pode acreditar em uma determinada hipótese e $bel(Ev)$ representa a crença atual em Ev. Sabendo-se que $bel(Ev) \leq \wp\ell(Ev)$ é conveniente representar a crença em Ev como sendo o intervalo representado por $[bel(Ev);\ \wp\ell(Ev)]$. A esse intervalo dar-se o nome de **intervalo de crença**, representado por $\Im(Ev)$.

Tal intervalo exprime a faixa de probabilidade na qual se pode acreditar em uma determinada hipótese Ev, sem correr o risco de graves erros com suposições.

2.2.6. Combinação das Funções de Crença

O processo de acúmulo de evidências usando a TDS requer um modelo que combine o suporte a uma hipótese, ou sua negação, baseando-se em várias observações.

Convencionalmente usa-se a notação $m_{\,1} \oplus m_2$ para indicar os efeitos de combinação entre duas atribuições de probabilidade básica $m_{\,1} \oplus m_2$. De forma semelhante, representam-se as combinações entre funções de crenças $bel(Ev_1)$ e $bel(Ev_2)$ como sendo $bel(Ev_1) \oplus bel(Ev_2)$.

Tendo m_1 e m_2 como duas *bpa* de um frame de discernimento Θ, tem-se então a sua combinação como resultado de uma soma ortogonal e uma normalização, definida por:

$$m(\Theta) = 0$$

$$m_1 \oplus m_2 = X \sum_{\substack{A \cap B = E_v \\ E_v = \Theta}} m_1(A) \cdot \mathrm{m}_2(\mathrm{B}), \forall \, \mathrm{E}v \subseteq \Theta$$

Onde X é a constante de normalização definida por $\frac{1}{1-k}$, e K é a soma dos *bpa* das ocorrências de $\emptyset$. O conjunto vazio aparece sempre que se tenta combinar hipóteses disjuntas, ou seja, ocorrência de hipóteses conflitantes entre si.

Exemplo 5. Fazendo uma pequena modificação no frame de discernimento do exemplo 1, tem-se uma rede que está sempre sobre ataque e apresenta como resultado de seus IDSs os seguintes estados:

- Ataque TCP Syn Flood {Ts}
- IP Scan {IPs}
- Ataque TCP Fin Scan {Tf}
- Port Scan {Ps}

Suponha então que para um determinado instante, observa-se uma m_1 que evidência {Ts, IPs}, com um grau de 0,5. Enquanto que uma m_2 desconfirma a presença de um TCP Syn Flood na rede, ou seja, pelos conceitos da TDS já vistos, confirma {IPs, Tf, Ps} com uma valor de 0,6. Então se pode gerar uma nova estrutura pela combinação de $m_1 \oplus m_2$ representada a seguir:

Tabela 2.1. Rede de crenças para m_1 e m_2

		$\boldsymbol{m_2}$	
		{IPs, Tf, Ps} (0,6)	Θ (0,4)
$\boldsymbol{m_1}$	{Ts , IPs} (0,5)	{IPs} (0,3)	{Ts, IPs} (0,2)
	Θ (0,5)	{IPs, Tf, Ps} (0,3)	Θ (0,2)

Logo,

- $\mathrm{m}_1 \oplus \mathrm{m}_2 (\{\mathrm{IPs}\}) = 0{,}3$
- $\mathrm{m}_1 \oplus \mathrm{m}_2 (\{\mathrm{IPs}, \mathrm{Tf}, \mathrm{Ps}\}) = 0{,}3$
- $\mathrm{m}_2 \oplus \mathrm{m}_2 (\{\mathrm{Ts}, \mathrm{IPs}\}) = 0{,}2$
- $\mathrm{m}_2 \oplus \mathrm{m}_2 = \mathrm{m}_2 (\{\Theta\}) = 0{,}2$
- $\mathrm{m}_2 \oplus \mathrm{m}_2 = 0$, para qualquer outro subconjunto de Θ

Note que para esse exemplo $X = 1$, já que $K = 0$.

Com os valores obtidos na tabela 2.1, pode-se encontrar facilmente $bel_1 \oplus bel_2$, para todos os elementos de Θ Por exemplo:

- $\mathrm{bel}_1 \oplus \mathrm{bel}_2 (\{\mathrm{Ts}, \mathrm{IPs}\}) =$
  $\mathrm{m}_1 \oplus \mathrm{m}_2 (\{\mathrm{Ts}, \mathrm{IPs}\}) + \mathrm{m}_1 \oplus \mathrm{m}_2 (\{\mathrm{Ts}\}) + \mathrm{m}_2 \oplus \mathrm{m}_2 (\{\mathrm{IPs}\})$

- $\text{bel}_1 \oplus \text{bel}_2 (\{\text{Ts}, \text{IPs}\}) = 0{,}2 + 0 + 0{,}3$
- $\text{bel}_1 \oplus \text{bel}_2 (\{\text{Ts}, \text{IPs}\}) = 0{,}5$

Exemplo 6. Suponha que agora no mesmo instante do exemplo 5, tem-se uma nova evidência que suporta {Ts} com grau de 0,8. Pelas normas da TDS deve-se então fazer uma nova rede com a combinação da nova evidência m_3 com a evidência m_4, onde $m4 = m_1 \oplus m_2$, do exemplo anterior.

Tabela 2.2. Rede de crenças para m_3 e m_4

		m_4			
		{IPs} *(0,3)*	*{Ts, IPs}* *(0,2)*	{IPs, Tf, Ps} (0,3)	Θ *(0,2)*
m_3	*{Ts}* *(0,8)*	*Ø* *(0,24)*	*{Ts}* *(0,16)*	*Ø* *(0,24)*	*{Ts}* *(0,16*
	Θ *(0,2)*	*{IPs}* *(0,06)*	*{Ts, IPs}* *(0,04)*	*{* IPs, Tf, Ps *}* *(0,06)*	Θ *(0,04)*

Visto que neste exemplo o Ø aparece duas vezes com o valor de 0,24, então:

- $\text{K} = 0{,}24 + 0{,}24 = 0{,}48$
- $1 - \text{K} = 0{,}52$

Então,

- $\text{m}_3 \oplus \text{m}_4(\{\text{Ts}\}) = \frac{(0{,}16+0{,}16)}{0{,}52} = 0{,}615$
- $\text{m}_3 \oplus \text{m}_4(\{\text{IPs}\}) = \frac{0{,}06}{0{,}52} = 0{,}115$
- $\text{m}_3 \oplus \text{m}_4(\{\text{IPs}, \text{Tf}, \text{Ps}\}) = \frac{0{,}06}{0{,}52} = 0{,}015$
- $\text{m}_3 \oplus \text{m}_4(\{\text{Ts}, \text{IPs}\}) = \frac{0{,}04}{0{,}52} = 0{,}077$
- $\text{m}_3 \oplus \text{m}_4(\Theta) = \frac{0{,}04}{0{,}52} = 0{,}077$
- $\text{m}_3 \oplus \text{m}_4 = 0$, para quaisquer outro subconjunto de Θ

2.2.7. Peso de Conflito

Como visto anteriormente, para efetuar uma normalização defini-se X como sendo $\frac{1}{1-k}$, onde K é a soma dos *bpa*'s de todas as instâncias Ø, ou seja, combinação de ocorrências disjuntas.

O valor $\log\ (x)^{-1}$ corresponde ao **peso de conflito** entre bel_1 e bel_2, denotado $con(bel_1, bel_2)$. Se não existe conflito entre as evidências, como no exemplo 5, então K = 0, logo $con(bel_1, bel_2) = 0$. No entanto, se o caso for o contrário e bel_1 e bel_2 são completamente disjuntas, então K = 1 e $con(bel_1, bel_2) = \infty$. Em tais condições a combinação $bel_1 \oplus bel_2$ não é possível.

Em alguns casos, mesmo o que a combinação $bel_1 \oplus bel_2$ seja possível, o resultado dessa combinação pode ser indesejável utilizando-se da regra de Dempster. Nesses casos, o resultado pode não ser o esperado, chegando, em alguns casos, a contrariar até mesmo o óbvio. Nesses casos, no entanto, o peso de conflito é fundamental. Pode-se verificar que quando o valor do peso de conflito supera um valor aproximado de 0.6, os resultados são geralmente indesejáveis.

3. TRABALHOS RELACIONADOS

Como solução, o uso de elementos sensores estrategicamente ou não distribuídos pelas redes junto a mecanismos de avaliação e tomada de decisão têm sido cada vez mais estudados e implementados na luta contra anomalias de tráfego. A questão é como agregar os mais diversos e diferentes resultados sobre o estado de uma rede de forma a obter uma resposta rápida, concisa e correta.

Visando mostrar a importância do uso da teoria da evidência como técnica de fusão de dados aplicada a detecção de anomalias, esse capítulo apresenta e discute três trabalhos relacionados ao tema.

3.1. A novel approach for a Distributed Denial of Service Detection Engine

A arquitetura proposta por Siartelis et al [17] descreve a utilização da teoria da evidência de Dempster-Shafer na elaboração de um sistemas de detecção de ataques distribuídos. Fazendo uso dos dados fornecidos por múltiplos sensores, este trabalho emprega a TDS como arcabouço para a criação de um mecanismo (*engine*) de fusão de dados multisensor visando a geração de um eficiente sistema de detecção de ataques DDoS.

De modo geral, a arquitetura é formada por um conjunto de diferentes sensores, espalhados em pontos diferentes da rede, operando de forma autônoma, mas que compartilham suas crenças sobre o verdadeiro estado da rede, ou seja, se ela está ou não sob ataque. Os autores assumem que a rede pode ser vista como um sistema com comportamento estocástico sem qualquer modelo funcional. Desta forma, é possível tentar inferir sob o estado do sistema sem conhecimento prévio, sendo necessários apenas os dados informados pelos sensores, que podem ter obtido suas "evidências" baseadas em critérios totalmente diferentes. A figura 3.1 ilustra a arquitetura proposta.

A arquitetura permite que ferramentas IDS, detectores de ataques, coletores de tráfego e até mesmo sistema de contabilização e medida de tráfego possam ser usados como sensores. O protótipo apresentado utiliza o IDS SNORT [23] e um coletor de dados SNMP como sensores. O primeiro é um famoso IDS que gera estatísticas do tráfego da rede através de tráfego no formato *pcap*. O outro permite coletar informações como a quantidade de bytes por segundo, pacotes por segundo, número de fluxos ativos e fluxos com falhas.

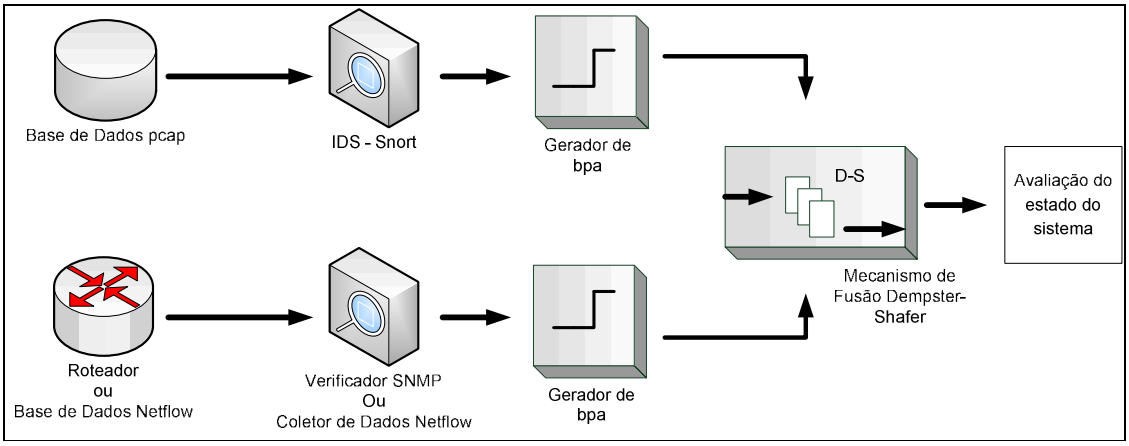

Figura 3.1: Arquitetura para detecção de DDoS usando TDS.

De modo simplificado, uma vez configurado, cada sensor emprega sua própria "inteligência" baseada na técnica de análise que utiliza, para gerar as crenças (*bpa*) sobre o tráfego avaliado. Feito isso, os *bpas* são repassados ao mecanismo de fusão de dados para tomada de decisão.

A principal vantagem desta arquitetura é que a TDS não necessita de qualquer conhecimento prévio do comportamento da rede para gerar resultados. Além disso, a possibilidade de usar vários sensores ou filtros, baseados em diferentes mecanismos de classificação é extremamente interessante e tem se mostrado uma tendência na área de segurança. Como desvantagens, é necessário um profundo conhecimento do funcionamento dos sensores para que as crenças geradas nas hipóteses sejam condizentes com o real estado da rede. Outro ponto importante é que como os resultados dependem diretamente do desempenho dos sensores, o uso de sensores muito simples pode acabar por não contribuir para o estabelecimento do estado da rede.

3.2. Anomaly Detection Using the Dempster-Shafer Method

O método proposto por Chen e Aickelin [24] descreve a utilização da fusão de dados em um sistema para detecção de trafego anômalo. Utilizando-se da TDS, os autores desenvolveram um sistema de detecção de anomalias capaz de tratar informações com diferentes características e garantir uma maior eficiência de detecção. O sistema possui um mecanismo capaz de "aprender" as características fundamentais do ambiente e, desta forma, gerar inferências sobre o estado da rede.

Com o conhecimento do padrão de funcionamento da rede, é possível determinar a crença em cada análise gerada por cada um dos vários sensores distribuídos pela rede, mesmo que esses sensores utilizem diferentes metodologias de análises.

A figura 3.2 ilustra a detecção de anomalias baseado em processos de aprendizagem.

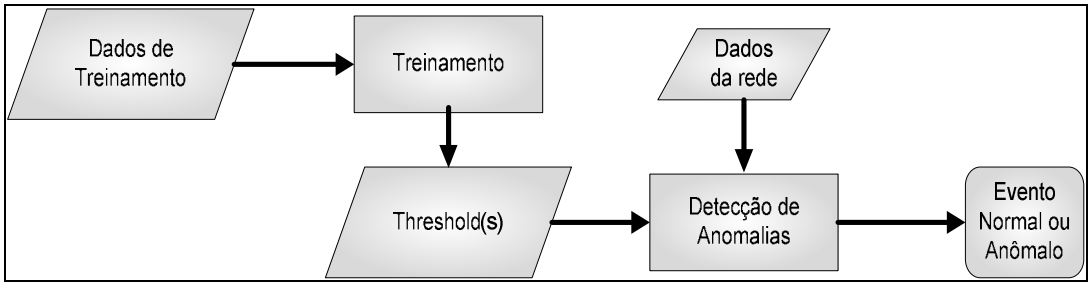

Figura 3.2: Fluxo de dados de um sistema baseado em aprendizagem.

O sistema permite que diferentes tipos de sensores de anomalias possam ser espalhados pela rede de forma a gerar dados que reflitam suas visões em relação ao estado da rede. Diante destes dados módulos específicos são capazes de determinar a crença em cada uma dessas inferências e repassar esta informação para um módulo de combinação de dados, baseado na TDS, que tem o poder de tomar decisões.

Estes módulos específicos são sistemas baseados em aprendizagem que, após uma fase de treinamento, são capazes de estabelecer parâmetros que condizem com as características do ambiente e, desta forma, determinam o bpa de cada dado extraído dos sensores. Na figura 3.3 é possível visualizar a arquitetura geral do sistema proposto.

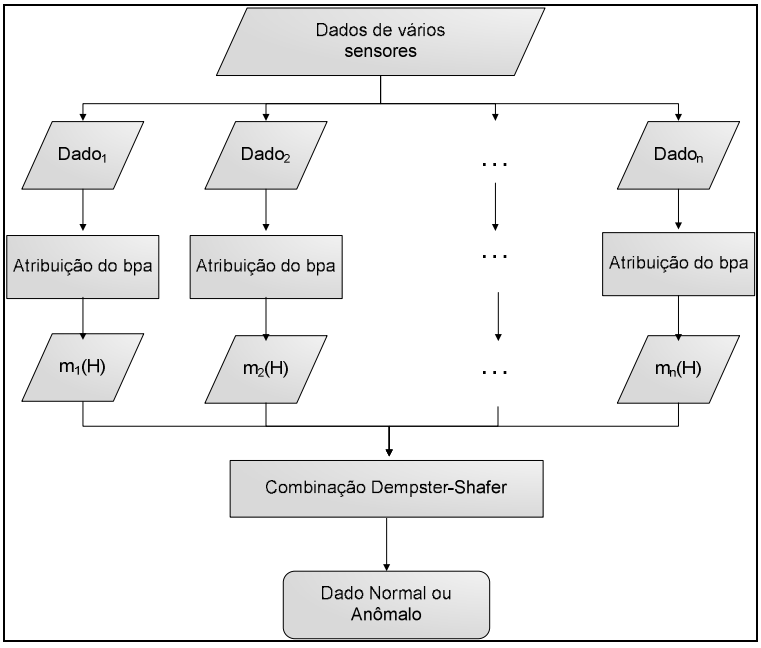

Figura 3.3: Arquitetura do sistema de detecção.

Em resumo, diante de um processo de treinamento o sistema é capaz de estabelecer limiares de padrões dos sensores de detecção, estabelecer a crença nessas análises e com regras de combinações de valores de massa (bpa) identificar o estado da rede.

A solução proposta não necessita de um conhecimento profundo dos mecanismos de análise de dados dos sensores utilizados e capacidades do sistema para lidar com problemas com muitas características inerentes, como acontece em problemas reais. Entretanto, um fator que chama bastante atenção no sistema proposto é o fato de que a geração das funções de crença para a determinação dos bpa é feita de forma adaptativa, necessitando assim de uma grande quantidade de dados para que possa ser efetuada uma fase de treinamento. Cerca de 90% dos dados é utilizada para treinamento e geração das funções de crença, ou seja, nesse método é necessário um grande esforço para que o sistema venha a se tornar eficiente.

3.3. DS Evidence Theory and its Data Fusion Application in Intrusion Detection

O IDSDMF, proposto por Tian et al. [25], é um modelo que descreve o uso de um mecanismo de fusão de dados baseado na teoria da evidência de Dempster-Shafer visando minimizar o número de falsos positivos encontrados nos alertas gerados pelos IDSs espalhados pela rede.

A figura 3.4 ilustra o modelo da rede com o IDSDMF.

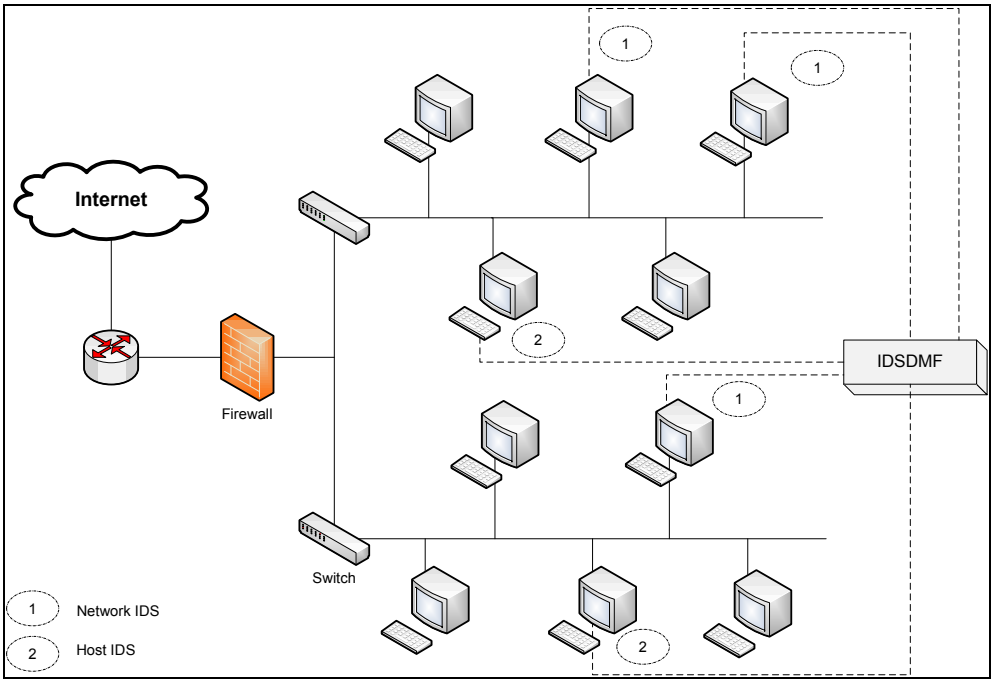

Figura 3.4: Modelo IDSDMF.

Este modelo faz uso de diferentes tipos de IDS como SNORT e E-TRUST, preocupando-se apenas com a padronização nos modelos de alertas gerados [26]. Diante de diferentes alertas gerados por diferentes IDSs, o IDSDFM correlaciona estes dados de acordo com seu grau de similaridade e de acordo com as regra de combinação de massa da TDS define o bpa para cada uma dessas evidências. A figura 3.5 demonstra o processo de correlação entre diferentes eventos, gerados por IDSs distintos.

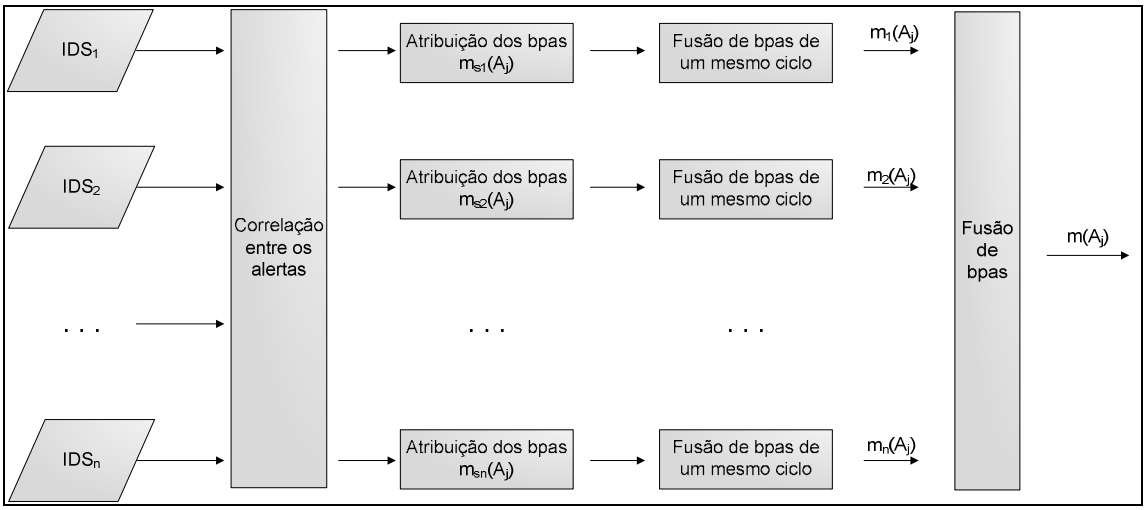

Figura 3.5: Processo de correlação entre alertas.

A principal vantagem desse modelo é a diminuição significativa no número de falsos positivos, além do uso de ferramentas simples e de fácil configuração como IDSs. No entanto, o processo de relação de similaridade entre diferentes alertas pode ser um grande desafio para esta solução.

4. PROJETO E IMPLEMENTAÇÃO DO ADS-FUSION

Este capítulo descreve os componentes do protótipo ADS-Fusion, bem como seu funcionamento e o processo de desenvolvimento. Primeiro, uma visão geral do protótipo é apresentada. Em seguida, cada um dos componentes (módulos) será explicado e, por fim, o processo de funcionamento e integração entre os módulos serão detalhados.

4.1. Protótipo ADS-Fusion

O ADS-Fusion foi projetado com o objetivo de estudar técnicas de detecção de anomalia e desenvolver um sistema capaz de aumentar a eficiência da detecção através da fusão de dados. Na prática representa um software capaz de, baseado em saídas geradas por ferramentas de detecção de anomalias, fundir dados e produzir uma inferência com um grau de certeza maior que as certezas geradas individualmente pelos módulos de detecção.

A idéia central é utilizar a teoria da evidência de Dempster-Shafer para lidar com a incerteza ou imprecisão dos resultados das análises realizadas por sensores espalhados pela rede e, dessa forma, a aumentar o grau de certeza sobre atividades intrusivas e maliciosas, permitindo que as mais corretas e precisar ações sejam tomadas para elevar a segurança da rede.

Semelhante a diversas arquiteturas de segurança, o ADS-Fusion é composto por: um módulo de coleta, sensores de análise ou filtros, e um mecanismo de fusão de dados, responsável por definir o estado real da rede. Afigura 4.1 ilustra a estrutura do ADS-Fusion.

O módulo de coleta é responsável pela monitoração do tráfego da rede e geração de arquivos em formato padronizado. Estes arquivos serão repassados para os diversos sensores e filtros da rede para que os mesmos possam processá-los e inferir o estado da rede. Os sensores são os componentes responsáveis por analisar os dados gerados pelos módulos de coleta e detectar possíveis anomalias existentes no tráfego da rede. Outro papel fundamental destes sensores é definir, a partir de seus mecanismos de classificação, a crença em cada uma das inferências geradas.

Por fim, o mecanismo de fusão que é responsável pela tomada da decisão. Fazendo uso dos recursos das regras de combinação da TDS, correlaciona às diferentes análises dos vários filtros gerando inferências mais precisas e com um maior grau de exatidão.

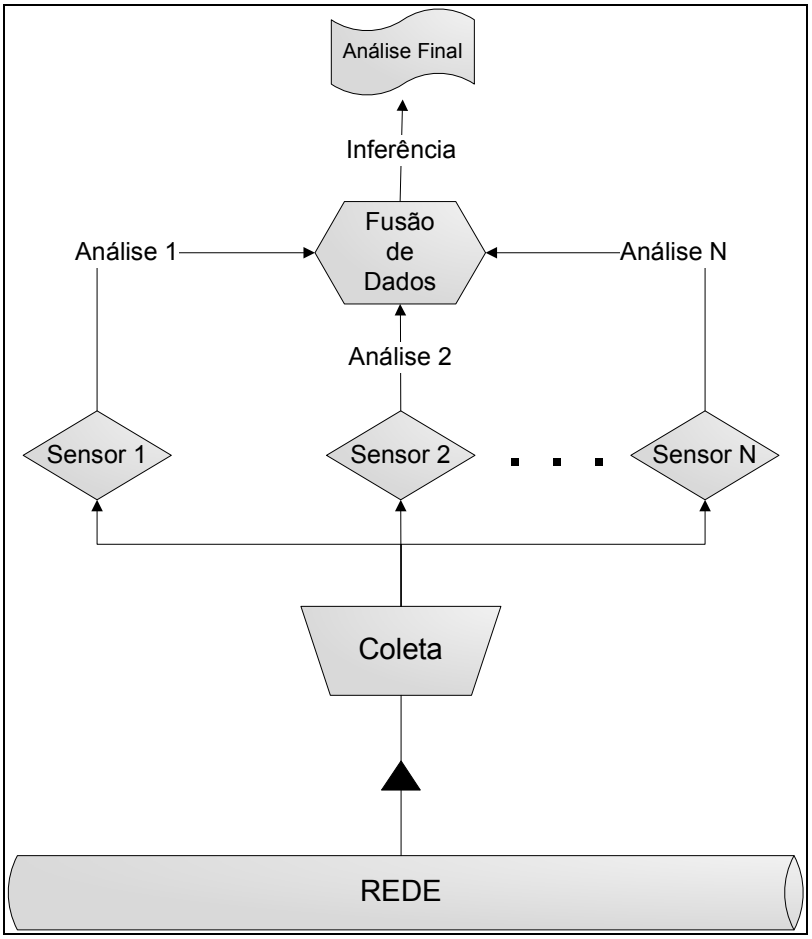

Figura 4.1: Arquitetura do ADS-Fusion.

4.2. Módulo de coleta

Antes de iniciar a explanação sobre o módulo de coleta é necessário afirmar que, por decisão de projeto, a implementação do protótipo só emprega tráfego off-line, isto é, utiliza traces com o tráfego já capturado da rede em formato .PCAP para efetuar todas as análises. O objetivo é justamente facilitar a análise do desempenho do sistema. O módulo de coleta foi todo desenvolvido usando a biblioteca PCAP [27]

A função do módulo de coleta é ler os arquivos de tráfego da rede e gerar as saídas necessárias ao funcionamento dos diversos sensores. Para implementação do ADS-Fusion, foram escolhidos dois sensores (Profiling e TCPModel, explicados posteriormente na seção 4.2) que necessitam de entradas de tráfego especiais. O primeiro utiliza fluxos e o segundo socks. A figura 4.2 ilustra a saída em fluxos gerada pelo módulo de coleta para o sensor Profiling.

```
#ini_sec ini_mic end_sec end_mic src_ip src_port dst_ip dst_port n_bytes n_frames app prot
1224856206,447841,1224856208,697463,192.168.0.66,137,192.168.0.255,137,384,4,0000,17
1224856201,425805,1224856207,785994,192.168.0.159,1900,239.255.255.250,1900,41670,100,0000,17
1224856229,4405,1224856229,4405,192.168.0.125,138,192.168.0.255,138,240,1,0000,17
1224856243,830101,1224856244,603069,192.168.0.139,42757,129.42.58.216,443,1273,7,0000,6
1224856244,7517,1224856244,563433,129.42.58.216,443,192.168.0.139,42757,1890,5,0000,6
1224856249,135561,1224856249,135561,192.168.0.139,60006,192.168.0.105,53,85,1,0000,17
```

Figura 4.2: Saída no formato de fluxos para o Profiling.

As colunas representam, respectivamente, o timestamp do primeiro pacote do fluxo em segundos (ini_sec), em microsegundos (ini_mic), o timestamp do último pacote do fluxo em segundos (end_sec), em microsegundos (end_mic), o endereço IP de origem (src_ip), a porta de origem (src_port), o endereço IP de destino (dst_ip), a porta de destino (dst_port), número de bytes do fluxo (n_bytes), número de pacotes do fluxo (n_frame), aplicação (app) e protocolo de transporte (prot).

4.3. Sensores de Análise

Os sensores são componentes de análise responsáveis pela geração das hipóteses sobre o possível estado real da rede. Estes sensores podem variar de pequenos mecanismos de detecção de anomalias (IDS para host, por exemplo) a complexos sistemas especialistas, como os propostos em [28] e [29].

Para implementação do protótipo foram escolhidas duas soluções para detecção de anomalias no tráfego de rede. A primeira é um método estatístico proposto em [30] denominado TCPModel e a segunda é um método de análise baseado no comportamento da rede, proposto em [13], denominado Profiling. A escolha destes dois sensores de detecção foi baseada em critérios bem simples. O TCPModel foi escolhido por se tratar de um mecanismo especialista em detecção de ataques DDoS. O Profiling, além de detectar um grande número de anomalias, é um sistema baseado em comportamento que não necessita de conhecimento prévio do comportamento da rede.

A utilização dessas diferentes técnicas de análise é utilizada visando uma maior abrangência no acompanhamento do estado da rede estudada. Detalhes sobre os mecanismos utilizados serão explanados a seguir.

4.3.1. TCPModel

O TCPModel [30] é um sistema, desenvolvido em Java, especialista na detecção de ataques DDoS. Baseado no comportamento de troca de mensagens do protocolo TCP entre dois hosts, é capaz de avaliar se existe algum comportamento anormal na rede através da razão entre a taxa de envio e recebimento de pacotes.

A premissa de detecção implementada no TCPModel foi proposta por [28] e para analisar a razão entre entrada e saída de pacotes TCP foi implementado um algoritmo denominado *Threshold Adaptativo*. Quando a razão ultrapassa um determinado limiar considerado normal, um alarme é disparado. Para analisar a entrada e saída de pacotes o TCPModel agrupa os pacotes em fluxos que possuem IP e porta de destino comuns, denominados socks e exemplificados na figura 4.3.

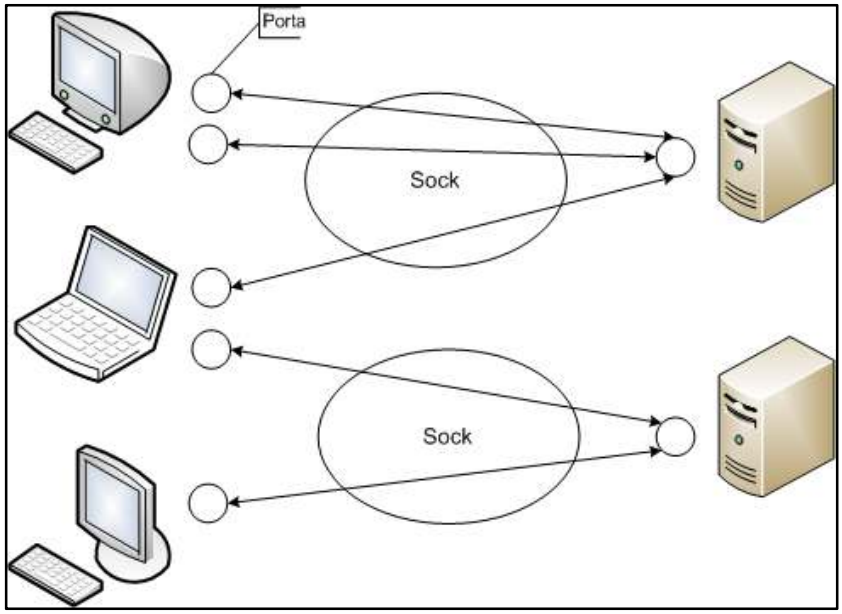

Figura 4.3: Representação de Socks.

Agrupando os pacotes em socks, o TCPModel utiliza-se de um conjunto de métricas para definir o estado normal de uma comunicação de protocolo TCP e é capaz de reconhecer modificações neste padrão. Um fator importante neste método de análise é que por se tratar de um algoritmo adaptativo ele é capaz de se moldar a possíveis modificações no comportamento da rede.

4.3.1.1. Threshold Adaptativo

Considerando X_n a relação entre pacotes enviados e recebidos para um determinado intervalo de tempo *n* dentro de um *sock* qualquer, e μ_{n-1} como sendo a média ou a expectativa do valor para o intervalo anterior nessa mesma média, então pode-se definir a fórmula abaixo como sendo a que verifica se o limiar de troca de pacotes foi excedente ao considerado comum.

$$X_n \geq (\alpha + 1)\bar{\mu}_{n-1}$$

Equação 4.1 – Verificação do limiar do algoritmo *Threshold Adaptativo*.

Na equação 4.1, o parâmetro α define a taxa percentual aplicada ao valor esperado para que se defina um ataque. A aplicação direta dessa equação é extremamente rápida na detecção das anomalias, mas gera um grande número de falsos positivos. A solução encontrada para resolver esse problema foi à generalização da equação 4.1, visando adequá-la ao processo de detecção de anomalias, representada na equação 4.2.

$$f(x) = \begin{cases} 1, & x_n \geq (\alpha + 1)\bar{\mu}_{n-1} \\ 0, & x_n < (\alpha + 1)\bar{\mu}_{n-1} \end{cases}$$

Equação 4.2 – Função de indicação do *Threshold Adaptativo*

A equação 4.2 é utilizada como um indicador para a verificação final do algoritmo, retornando 1 sempre que o limiar for ultrapassado e 0 caso contrário. A verificação final analisa a quantidade de intervalos consecutivos necessários para que uma notificação de ataque seja disparada (Equação 4.3).

$$\sum_{i-n-k+1}^{n} f(x_i) > K$$

Equação 4.3 – Teste de sinalização do *Threshold Adaptativo*

Percebe-se que na equação 4.3 o parâmetro $K > 1$ é o indicador no número de intervalos consecutivos para que o alerta seja disparado. Outro ponto importante do sistema de equações adotado pelo *Threshold Adaptativo* é a forte dependência com os parâmetros α e μ. Se for levado em consideração que estes parâmetros têm uma relação com o estado da rede analisada, faz-se necessário a utilização de um modelo para definição desses valores. O Alisamento exponencial ou EWMA (*Exponentially Weighted Moving Average*) foi à metodologia adotada e pode ser vista em detalhes em [30].

4.3.1.2. Integração ao modelo de fusão de dados

Para integrar o TCPModel ao modelo de fusão de dados proposto, fez-se necessário a criação de um mecanismo capaz de definir valores de crenças (bpa) a cada hipótese resultante do processo de análise.

A geração de bpa pelo TCPModel é bastante simples. Baseia-se na distância entre os valores obtidos na aplicação da equação 4.2. Sempre que o retorno da função for igual a 1, o sistema considera que a rede pode estar sobre ataque e, desta forma, calcula o bpa baseado na distancia de $(\alpha + 1)\bar{\mu}_{n-1}$ para x_n, quanto maior for $(\alpha + 1)\bar{\mu}_{n-1}$ maior a crença no ataque.

A figura 4.4 abaixo contém um trecho da saída gerada pelo TCPModel.

```
58.33.126.229:5576 -> 192.168.0.163:0

Pkt Send: 92 PktRec: 0

TimeSend: 1227656198.961773 TimeRec: 1227656198.961773

threshold: 6.0

State: BAD
```

Figura 4.4: Trecho da análise do TCPModel.

O trecho contém a detecção de uma conexão anômala entre os endereços IP 58.33.126.229 e 192.168.0.163 com um *threshold* de troca de pacotes calculado em 6. Supondo que nesta rede o *threshold* estabelecido como normal tem valor igual a 5, qualquer conexão que ultrapassar este limiar será considerada anômala. A conexão estabelecida acima ultrapassa esse valor e terá seu bpa calculado sobre o percentual da diferença. Fixando a crença do estado normal da rede sempre em 0.5 é possível determinar a crença do ataque como sendo a soma entre a crença normal e o percentual de aumento $(6 \div 5 = 1{,}2\ ;\ o\ que\ representa\ um\ aumento\ percentual\ de\ 20\%)$, obtendo, desta forma, um bpa = 0,6.

4.3.2. Profiling

A metodologia proposta por Xu et al. [13] visa a identificação de anomalias de tráfego. O método faz uso de técnicas de mineração de dados e informação teórica (entropia) para automaticamente descobrir padrões de comportamento significantes no tráfego de dados. A metodologia (aqui denominada de *profiling*) automaticamente descobre comportamentos do tráfego massivo e fornece meios plausíveis para entender e rapidamente reconhecer tráfego anômalo.

Basicamente, examina padrões de comunicação dos computadores (endereços e portas) que são responsáveis por um significativo número de fluxos em um determinado período de tempo.

O processo do profiling basicamente inclui a extração de clusters significativos e a classificação do comportamento deles baseado no relacionamento entre os clusters. Por exemplo, para um dado endereço IP de origem (srcIP) i, o processo do profiling inclui a extração dos fluxos com srcIP i dentro de um cluster (denominado de cluster srcIP) e a caracterização do padrões de comunicação (ou seja, comportamento) usando medições de teoria da informação (entropia) sobre as três dimensões de fluxos restantes, ou seja, endereço IP de destino (dstIP), porta de origem (srcPrt) e porta de destino (dstPrt).

A figura 4.5 ilustra melhor os passos dessa metodologia.

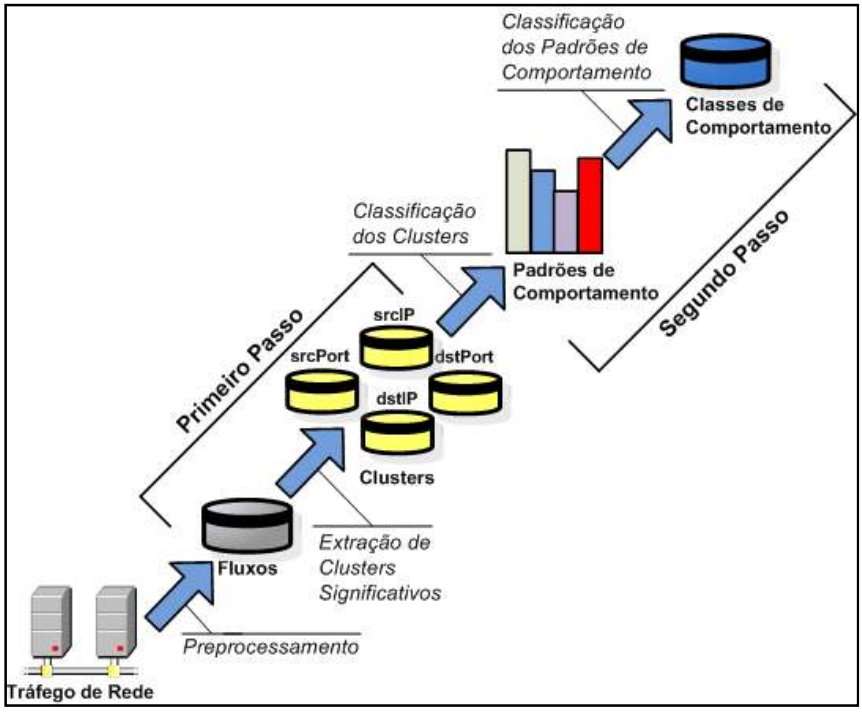

Figura 4.5: Etapas do método Profiling.

A primeira etapa consiste em analisar um conjunto de fluxos baseado nas tuplas bem conhecidas para decidir sobre um cluster de interesse. O objetivo é extrair os clusters significativos de dimensões específicas, isto é, endereços IP de origem e destino e portas de origem e destino. Então, os clusters mais significativos são extraídos de uma dimensão fixa (por exemplo, endereço IP de origem) e o conceito de entropia é usado para medir a quantidade de incerteza relativa (RU - *Relative Uncertainly*) contida nos dados.

A segunda etapa é responsável por descobrir relações entre os clusters, ou seja, encontrar padrões de comportamento comuns para o perfil do tráfego. Visando alcançar este objetivo, a metodologia propõe uma classificação do comportamento baseado nos modelos de comunicação dos computadores de usuários finais e serviços. Desta forma, para cada cluster, uma RU é computada e usada como parâmetro para criar classes de comportamento (BC - *Behavior Classes*). Com essas classes é possível identificar qual delas representa tráfego anômalo ou indesejado.

Visando melhorar o entendimento do processo de definição de classes de comportamento, a seguir serão fornecidos detalhes do procedimento.

4.3.2.1. Definição de Classes de Comportamento

Considerando um cluster de conjunto mais significativo, por exemplo, de srcIP (IP de origem), em um determinado intervalo de tempo, todos os fluxos de cada grupo mais significativo compartilham a mesma chave, ou seja, IP de origem. Enquanto isso, cada uma das suas três dimensões "livres" (no caso IP de destino, porta de origem e porta de destino) pode possuir qualquer valor, logo os fluxos podem possuir diferentes distribuições de probabilidade para cada uma das dimensões livres. Desta forma, pode-se associar uma incerteza relativa (RU).

Para cada cluster extraído em relação a uma dimensão qualquer é definido os elementos X, Y e Z para as suas dimensões "livres" seguindo o padrão da tabela 4.1. Cada conjunto desses gerado pode ser representado pelo vetor RU [RUx, RUy, RUz], onde o RU de cada conjunto pode ser calculado pela fórmula definida em [13].

Tabela 4.1: Convenção das distribuições livres.

Chave do Grupo	Dimensões Livres		
	X	Y	Z
srcIP	srcPort	dstPort	dstIP
dstIP	srcPort	dstPort	srcIP
srcPort	dstPort	srcIP	dstPort
dstPort	srcPort	srcIP	dstPort

Visando uma forma mais conveniente de agrupamento entre os clusters mais significativos, cada dimensão é dividida em três possíveis categorias, rotulada como: 0 (baixo), 1 (médio), 2 (alto), seguindo os seguintes critérios:

$$L(RU) = \begin{cases} 0(baixo), & se\ 0 \leq ru \leq \varepsilon \\ 1(medio), & se\ \varepsilon < ru < 1-\varepsilon \\ 2(alto), & se\ 1-\varepsilon \leq ru \leq 1 \end{cases}$$

Para os valores de ε, [13] define ε = 0,2 para as dimensões srcPort e dstPort, e ε=0,3 para srcIP e dstIP. Tem-se então que este processo de rotulação define 27 diferentes tipos de rótulos (ou diferentes BCs) que podem ser identificados por IDs. Para determinar o ID de um determinado conjunto utiliza-se a equação 4.4:

$$ID = L(RU_x).3^2 + L(RU_y).3^1 + L(RU_z).3^0 \qquad (4)$$
$$ID \in \{0,1,2,3,\ldots,26\}$$

Equação 4.4 – Equação de geração de ID de BCs

Por exemplo, se considerado a análise da chave srcIP e considerando-se que a relação com as chaves srcPort, dstPort, dstIP seja representada pela tupla [0 , 2 , 0], ou seja, relação srcPort = baixa, dstPort = alto e dstIP = baixo. Então o $BC = [\,L(Ru_x),\ L(Ru_y),\ L(Ru_z)]$ pode ser representado como $BC = 0 \cdot 3^2 + 2 \cdot 3^1 + 0 \cdot 3^0 = 0 + 6 + 0 = 6$, então, neste caso BC = 6.

É importante salientar que um mesmo identificador de um BC (BC = 6, como no exemplo anterior) tem significados distintos quando o grupo extraído tem como chave outras chaves de grupo como srcPort ou dstIP.

4.3.2.2. Integração ao modelo de fusão de dados

Depois de agrupados, classificados e estabelecido os BCs dos fluxos, o Profiling detecta a presença de um ataque avaliando os BCs, a freqüência de repetição entre os mesmos e a quantidade de fluxos associados a essa classificação.

Por exemplo, caso o Profiling análise o dstIP como chave de grupo. Esta análise será dividida em interações (espaços predefinidos de tempo). Suponha que na interação #1 os fluxos com IP de destino 10.108.40.X (150 fluxos) sejam classificados com o BC = 24 e que essa classificação para essa chave de grupo represente um ataque DDoS. Na interação #2 o BC para este mesmo IP se manteve e o número de fluxos aumentou para 250, então é possível aumentar a crença nessa inferência. No entanto, se na interação seguinte o BC tivesse alguma alteração, a avaliação teria de ser modificada e a sua crença tem de seguir o aumento ou a diminuição do número de fluxos analisados.

4.4. Mecanismo de Fusão de Dados

O mecanismo de fusão de dados é responsável pela tomada de decisões, definindo quando um determinado evento é classificado como anômalo ou normal. Tem como principal finalidade tratar as incertezas e imprecisões dos métodos de análise. Baseado na teoria da evidência de Dempster-Shafer, este módulo é capaz de manipular as hipóteses geradas pelos diversos filtros encontrados na rede e combinar a crença em cada uma dessas hipóteses inferindo assim o real estado da rede. De um modo geral este é o módulo responsável por agrupar a "visão" da rede pela perspectiva de cada um dos filtros e inferir o comportamento geral baseado na crença individual dos sensores.

Por se tratar de um módulo baseado na teoria da evidencia de Dempster-Shafer, se fez necessário o uso de uma biblioteca capaz de dar suporte as estruturas e algoritmos utilizados no calculo das funções básicas como função de crença $bel()$ e plausibilidade $\wp\ell(Ev)$. Por este motivo foi utilizada a API EvidenZ [31], uma implementação gratuita em C++.

O funcionamento do módulo de fusão de dados pode ser resumido em quatro estágios. O ponto de partida é a **leitura das análises** feitas pelos sensores ativos da rede. Após este estágio é executado um processo de **sincronização** responsável por relacionar os elementos apontados pelos sensores de forma que os relatos de mesmos eventos possam ser combinados. O passo seguinte à sincronização é a **combinação** dos eventos, fazendo uso das regras de combinação de função de crença. Após combinados os eventos é então possível **tomar uma inferência** condizente ao real estado da rede.

A figura 4.6 ilustra o fluxo de eventos do módulo de fusão de dados.

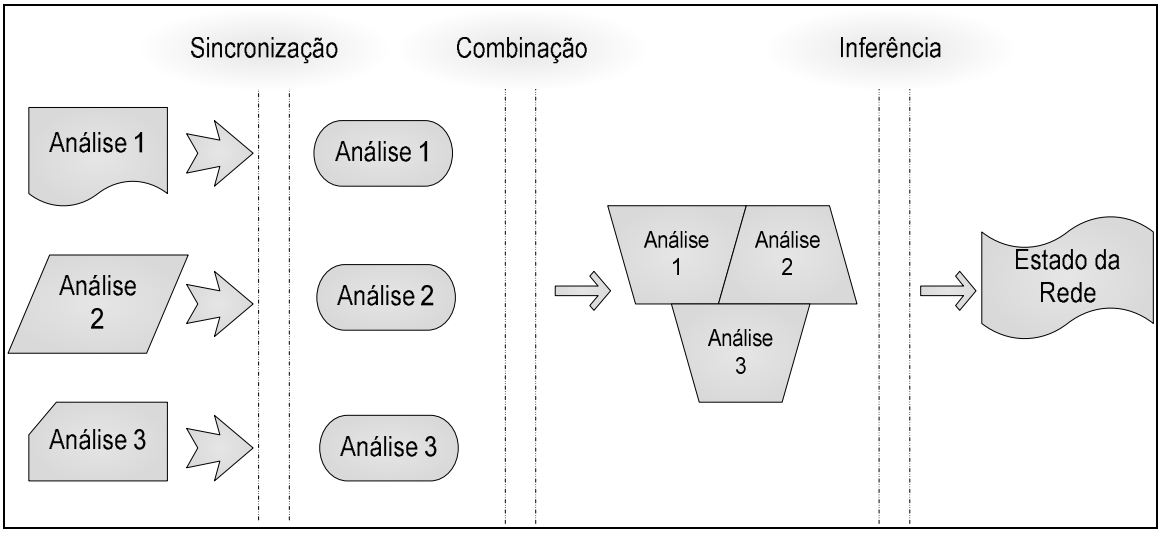

Figura 4.6: Fluxo de processamento do módulo de fusão.

A seguir descrição detalhada dos estágios de funcionamento do módulo de fusão.

- **Leitura das Análises**: Este processo consiste na leitura dos arquivos gerados pelo TCPModel e Profiling. Estes arquivos são subdivididos em interações de análise (subdivisões temporais do período da análise) e contém os elementos analisados e suas devidas classificações.
- **Sincronização:** O processo de sincronização consiste em relacionar os eventos relatados por cada uma das análises. Este processo se faz importante, pois é necessário garantir que cada uma das classificações a serem combinadas condiga ao mesmo período de tempo. Tomando como exemplo uma análise do tráfego de uma rede qualquer coletado no intervalo de dez minutos, se em um instante *t* qualquer, o TCPModel detectar um ataque DDoS a um dos *hosts* da rede. É importante que este evento seja combinado com a análise do Profiling deste mesmo tempo *t*, para que o ADS-Fusion possa gerar uma inferência mais exata. Para que esta sincronização possa ser feita, foram realizadas pequenas modificações na geração dos resultados dos sensores para que os mesmos relacionem as análises uma variável temporal.
- **Combinação:** Sendo o processo central do módulo de fusão de dados, a combinação é utilizada como "pistas" para a geração das inferências do estado real da rede. Para que seja possível combinar as saídas dos filtros, os atributos das saídas são transformados em elementos da EvidenZ, que basicamente contém o estado identificado (NORMAL ou ANÔMALO) e a crença neste estado (bpa).
- **Geração das inferências:** Nessa fase o ADS-Fusion determina se a rede está sob o efeito de uma anomalia ou não. Para tanto é necessária a definição do quadro de discernimento, elemento que contém os possíveis estados da rede, e a hipótese a ser avaliada.

Visando simplificar a validação, o quadro de discernimento gerado contém apenas dois possíveis elementos que representam o estado da rede. Então:

$$\Theta = \{ NORMAL, ANÔMALO \}$$

Considerando que a rede encontra-se por um maior tempo em estado NORMAL, a hipótese a ser questionada será sempre se o estado da rede é NORMAL. Desta forma, com o auxilio da EvidenZ, serão calculados a função de crença, $bel()$, e a plausibilidade, $\wp\ell()$, da hipótese $H = \{NORMAL\}$, lembrando que esse cálculo leva como base a combinação das análises feita na etapa anterior. De posse destes dois elementos é possível determinar o intervalo de crença, $\Im(H)$, que expressa à faixa de valores no qual é possível acreditar na hipótese *H,* ou seja, se é possível acreditar no estado normal da rede.

O trecho de código a seguir é responsável pela combinação e geração das inferências. Em seguida, serão detalhadas partes de seu funcionamento.

```
int main() {
  ...
```

```
  VarFactory<std::string> vf;
  VarT& rede = vf.new_var();
  rede.add_reals((VarT::start(), "NORMAL" , "ANOMALO"));
  ...
```
(1)

```
  VarDomainT vs1(vf, (VarDomainT::start(), &rede ));
  ConfigurationSetT cs1(vs1);
  cs1.add_configs((ConfigurationSetT::start(),
  ConfigurationT(vs1, (ConfigurationT::start(), "NORMAL"))));
  ConfigurationSetT cs1_all(vs1);
  Player p1(0.2);
  PotentialT player1(vs1);
  player1.add(cs1, p1);
  ...
```
(2)

```
  VarDomainT vs2(vf, (VarDomainT::start(),&rede));
  ConfigurationSetT cs2(vs2);
  cs2.add_configs((ConfigurationSetT::start(),
  ConfigurationT(vs2, (ConfigurationT::start(), "ANOMALO"))));
  ConfigurationSetT cs2_all(vs2);
  Player p2(0.7);
  player2.add(cs2, p2);
  ...
```
(3)

```
  ConfigurationSetT csfinal(global_vs);
  ConfigurationT  conffinal(global_vs,  (ConfigurationT::start(),
"NORMAL" ));
  csfinal.add_config(conffinal);
  PotentialT final = player1.get_comb(player2);
  ...
```
(4)

```
  std::cout << final << std::endl;
  std::cout << "Config set " << csfinal
          << "bel = " << final.bel(csfinal, e) << std::endl
          << "pls = " << final.pls(csfinal, e) << std::endl
          << "ign = " << final.ign(csfinal, e) << std::endl
          << "prob = " << final.prob(conffinal, e) << std::endl

          << std::endl;
  return 0; }
```

(5)

Na primeira parte (1) é responsável por definir a estrutura “VarFactory” de forma a tornar possível a definição do quadro de discernimento [`rede.add_reals((VarT::start(), "NORMAL" , "ANOMALO"))]` utilizado no problema. Neste caso é os estado NORMAL e ANOMALO como os possíveis estados da rede.

As duas etapas seguintes (2 e 3) definem o que a API define como “Player”, estruturas que representam as hipóteses geradas pelos filtros. Nestas estruturas são definidas a avaliação do estado da rede e o valor de crença atribuída a cada hipótese.

A quarta parte (4) consiste na geração da hipótese a ser avaliada pelo modelo de fusão e combinação das hipóteses. Como mencionado anteriormente, o ADS-Fusion sempre questionará se a rede encontra-se em estado normal.

O quinta e última parte é caracterizada pela utilização das funções da EvidenZ onde serão calculados as funções de crença, plausibilidade para que o intervalo de crença possa ser definido e o ADS-Fusion possa inferir o estado real da rede.

5. AVALIAÇÃO E RESULTADOS

Esta seção descreve o processo de avaliação de resultados utilizado neste projeto, assim como as métricas de avaliação, descrição do ambiente de teste e ferramentas utilizadas na geração do tráfego analisado.

5.1. Ambiente de Teste

Para realizar os testes do ADS-Fusion foram utilizadas as instalações do GPRT (Grupo de Pesquisas em Redes e Telecomunicações) da Universidade Federal de Pernambuco (UFPE) visando criar um ambiente controlado que se assemelha ao real e capaz de permitir ataques DDoS. Foram utilizados 52 PCs desktops, 2 switches com 24 portas 10/100/1000 Mbps e 2 servidor (processador Athlon XP 4200+ 64bits, com 2Gbytes de RAM e 160Gbytes de HDD), sendo um a implementação do ADS-Fusion). Windows XP e Ubuntu Linux foram os sistemas operacionais utilizados. A figura 5.1 apresenta a topologia da rede do ambiente de teste.

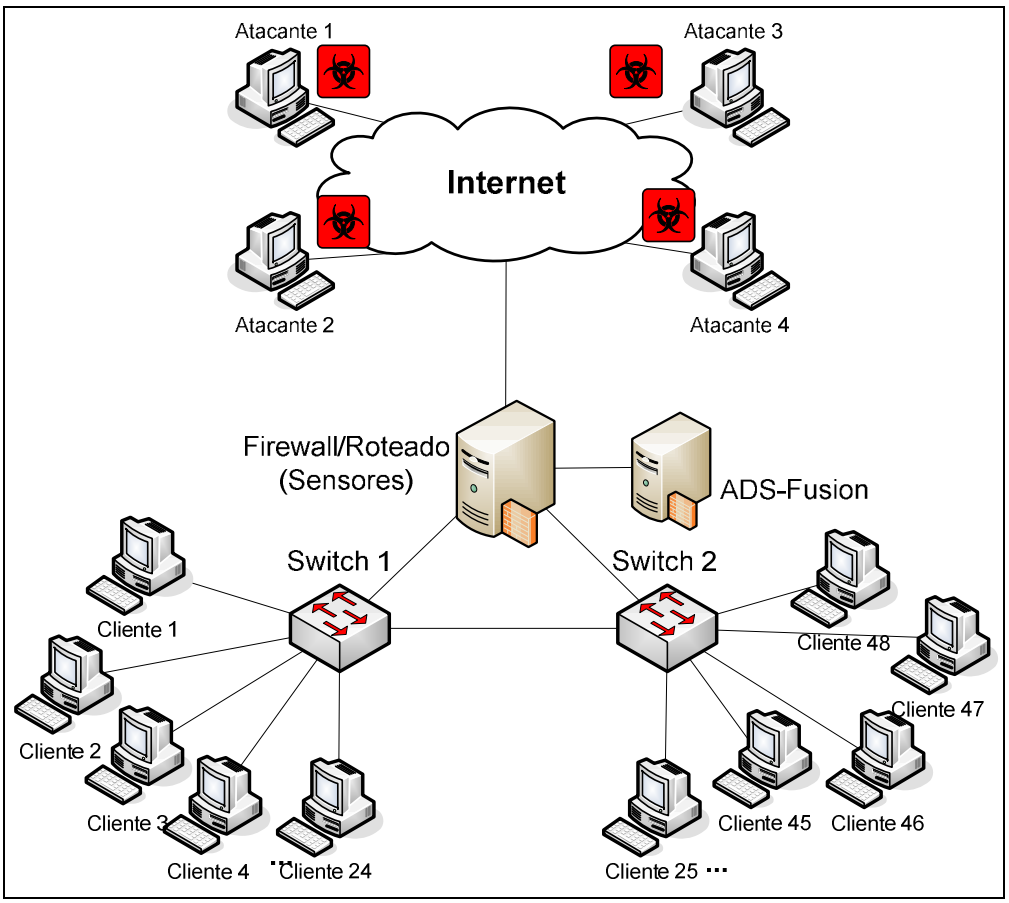

Figura 5.1: Ambiente de Teste.

5.2. Resultados

Uma vez que um dos sensores escolhidos na implementação do protótipo ADS-Fusion é especializado em ataques DDoS baseados no protocolo TCP, a gama de ataques para validação teve que ser reduzida e não incluiu ataques típicos dos protocolos UDP e ICMP.

Os ataques testados foram: TCP SYN flood, tráfego SPAM e um TCP SYN de baixa carga.

5.2.1. Ataque TCP SYN Flood

O ataque TCP SYN flood foi gerado através de um script que emprega a ferramenta Packit [34]. O script gera pacotes com endereço MAC, endereço IP de origem, porta de origem e porta de destino randômico. Cada pacote possui 1500 bytes de tamanho. A taxa de geração de pacotes é escolhida aleatoriamente, podendo variar entre 1000 e 8000 pacotes por segundo.

O ataque ocorreu no dia 20 de Novembro de 2008 entre as 13h50min e 14h00min, horário de um grande número de acessos a rede. Foram gerados dois ataques: o primeiro às 13h55min com duração de 30 segundos e o segundo as 13h58min com duração de 20 segundos.

A figura 5.2 ilustra o cenário de ataque onde é possível notar claramente o aumento no número de fluxos (de aproximadamente 1450 fluxos por segundo para 8500 fluxos por segundo) durante os períodos de ataque.

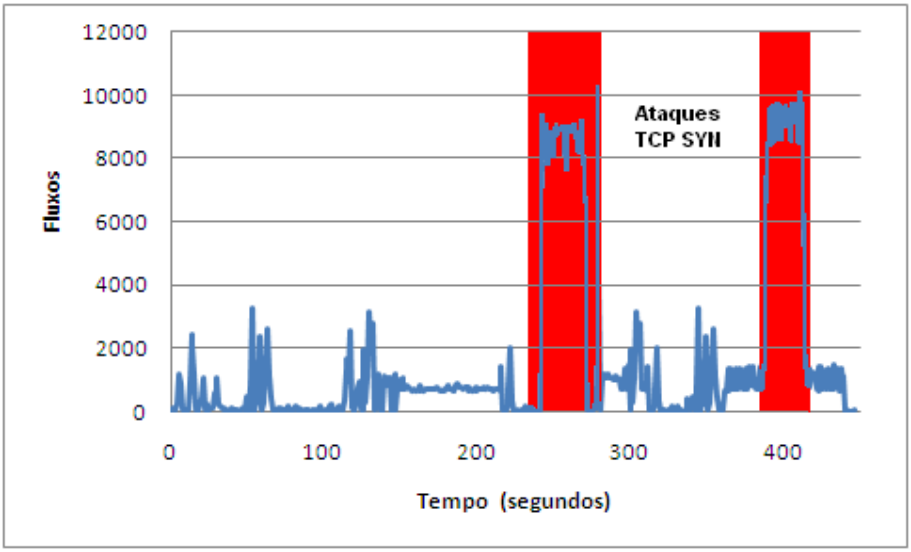

Figura 5.2: Cenário de ataque TCP SYN flood.

O tráfego gerado foi capturado e transformado em fluxos para análise de ambos os sensores (TCPModel e Profiling).

A Figura 5.3 ilustra a detecção de anomalias efetuada por ambos os sensores durante o primeiro ataque TCP SYN flood (às 13h55min, o que corresponde ao intervalo de tempo entre 200 e 300 segundos do trace capturado da Figura 5.2).

É facilmente percebido pela figura acima que ambos os métodos detectam o TCP SYN flood. Contudo, o sensor Profiling apresenta uma resposta mais rápida. No ponto 1 da figura, o TCPModel "demora" a atingir a carga máxima do ataque. No ponto 2, um resquício do ataque (provavelmente pacotes perdidos pela rede) é percebido somente pelo Profiling. Por outro lado, o TCPModel apresenta-se mais estável durante o período do ataque.

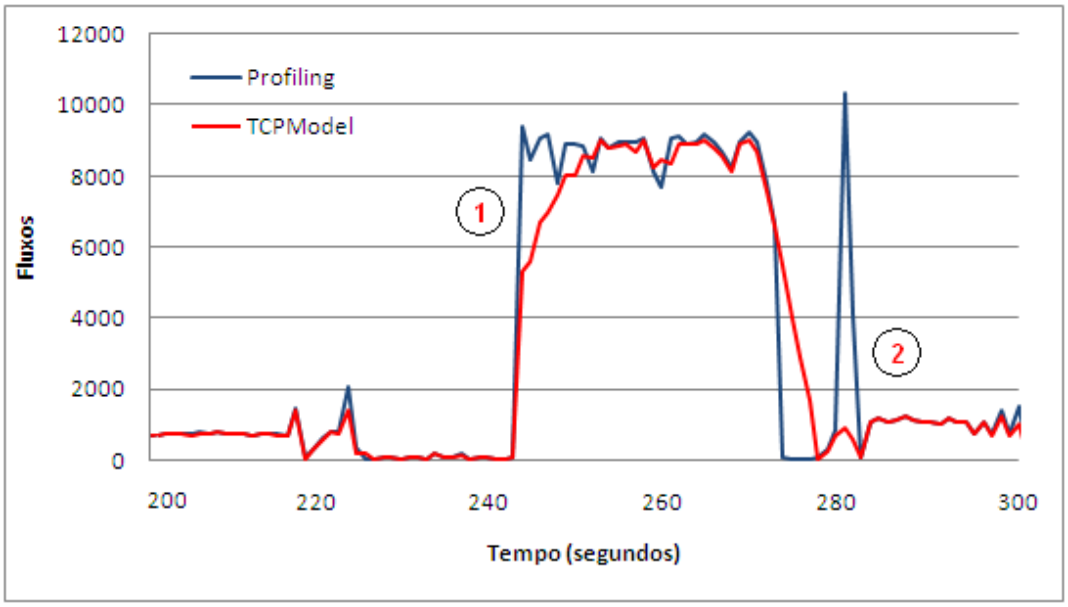

Figura 5.3: Detecção do Profiling e TCPModel.

Uma vez que ambos os sensores foram capazes de detectar o ataque, o ADS-Fusion foi empregado para aumentar a precisão da detecção. Os resultados apresentados na tabela 5.1 exprimem tanto a crença individual dos sensores quanto do mecanismo de fusão sobre o estado NORMAL da rede e não sobre o estado de ataque.

Tabela 5.1: Resultados da Fusão para o ataque TCP SYN Flood.

Tempo	Profiling	TCPModel	ADS-Fusion
200-210	80%	83%	98,3%
210-220	78%	81%	97,9%
220-230	74%	72%	96,36%
230-240	82%	86%	98,74%
240-250	36%	58%	36,09%
250-260	16%	12%	8%
260-270	19%	13%	9,26%
270-280	53%	46%	87,31%
280-290	8%	78%	17,6%
290-300	81%	79%	98%

A primeira coluna representa a linha temporal da análise do primeiro ataques TCP SYN (100 segundos). As coluna do Profiling e do TCPModel exprimem a crença atribuída ao tráfego normal de acordo com as técnicas utilizadas para detecção. Percebe-se que entre os intervalos entre 240-250 e 260-270 (período de duração do ataque), ambos os métodos foram capazes de detectar o ataque e o resultado da fusão de ambas as crenças (coluna ADS-Fusion) apenas corroborou com as técnicas, aumentando a precisão do tráfego avaliado não ser normal.

Os intervalos entre 270-280 e 280-290, como explicado anteriormente, uma grande quantidade de fluxos originados pelo ataque foi detectada pelo sensor Profiling, onde a crença na atividade normal caiu de 53% para 8%, enquanto que o sensor TCPModel não registrou essa anomalia. Para o mecanismo de fusão essas crenças são conflitantes, mas mesmo assim podem ser combinadas. Para o intervalo entre 270-280, a crença na normalidade do tráfego é alta, mas para intervalo seguinte, a crença, influenciada pelo Profling, aumentou a crença em que o tráfego não seja normal.

5.2.2. SPAM

Visando não somente avaliar ataques, foi realizada a injeção de tráfego SMTP visando à criação de "SPAM" para o servidor de email externo. Mais uma vez foi utilizada a ferramenta Packit para gerar tráfego forjado com duração de 60 segundos.

A figura 5.4 mostra que somente o sensor Profiling foi capaz de detectar a anomalia. Tal fato pode ser explicado da seguinte forma: para o TCPModel, o tráfego de e-mail é um tráfego legítimo tendo em vista que faz uso do mecanismo de *handshake* para estabelecer conexões. Por outro lado, o Profiling detecta a anomalia uma vez que o número de fluxos enviados ao mesmo destino cresce consideravelmente.

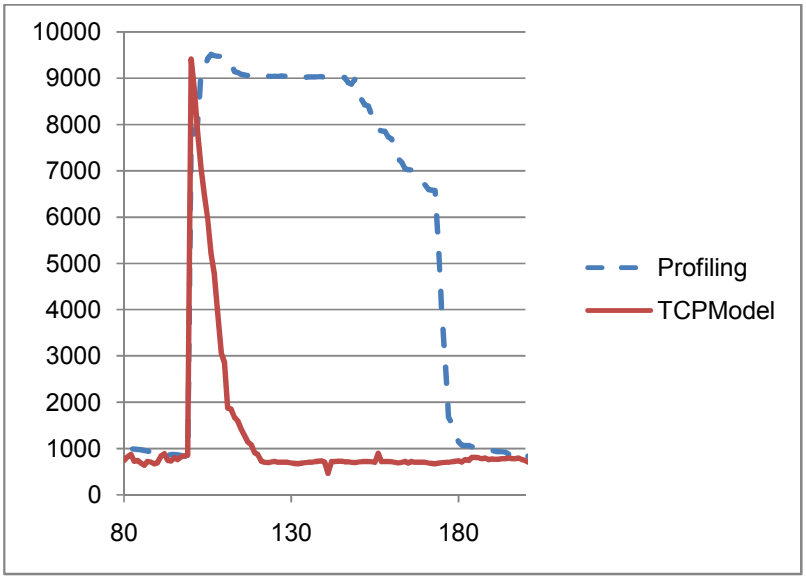

Figura 5.4: Detecção do tráfego de SPAM.

No processo de fusão dos dados, a não adequação do TCPModel influencia diretamente os resultados. A tabela 5.2 demonstra os resultados.

Tabela 5.2: Resultados da Fusão para o tráfego SPAM.

Tempo	Profiling	TCPModel	ADS-Fusion
80-100	80%	81%	98,10%
100-120	19%	34%	13,27%
120-140	22%	81%	21,24%
140-160	37%	78%	23,07%
160-180	41%	82%	44,69%
180-200	78%	80%	97,8%

Os intervalos entre 100-120 a 140-160 representam o ataque e a detecção pelos sensores. Uma vez que o TCPModel não registrou essa anomalia, uma combinação padrão da TDS não seria capaz representar a anomalia. A solução encontrada foi trabalhar com crença especialista, ou seja, foi atribuído um peso maior as evidências geradas pelo Profiling. Na prática, o TCPModel não gerá alertas e, então, sua inferência relacionada a anomalia não entra na combinação.

5.2.3. Ataque TCP SYN de baixa carga

O último ataque testado foi um TCP SYN de baixa carga, ou seja, foi empregada uma taxa de aproximadamente 48 pacotes por minuto. Uma vez que este tipo de ataque é difícil de detectar, pois gera uma pequena quantidade de fluxos, o Profiling não é capaz de detectá-lo.

Por outro lado, o TCPModel consegue perceber o número de pacotes TCP SYN destinados ao mesmo endereço e assim é capaz de detectar o ataque.

A figura 5.5 ilustra a detecção do ataque.

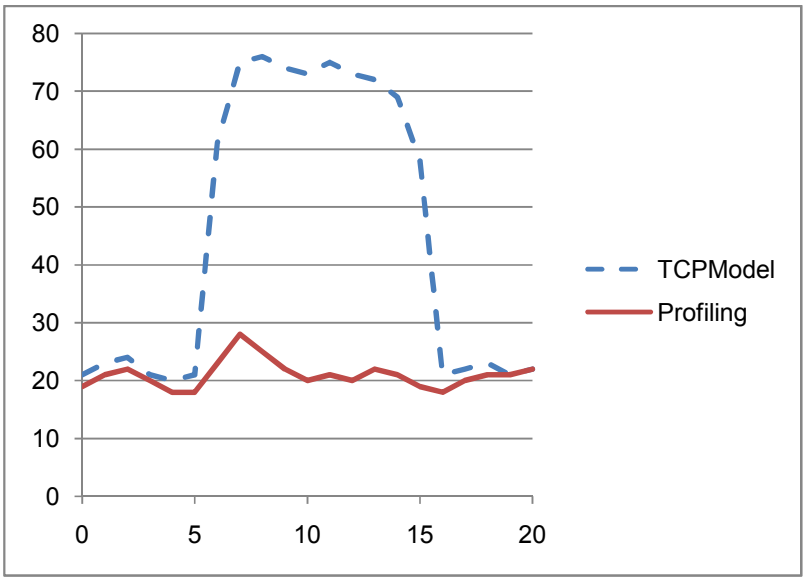

Figura 5.5: TCP SYN de baixa carga.

De modo similar ao SPAM, mais uma vez um dos sensores não foi capaz de detectar o ataque. Novamente, o uso de crença especialista foi utilizado. A tabela 5.2 exemplifica os resultados da combinação.

Tabela 5.3: Resultados da Fusão para o ataque de baixa carga.

Tempo	Profiling	TCPModel	ADS-Fusion
0-5	78%	83%	98,13%
5-10	62%	21%	24,99%
10-15	74%	24%	20,12%
15-20	73%	67%	95,54%

6. CONCLUSÃO

Hoje em dia, anomalias continuam causando inúmeros prejuízos a empresas e instituições. Ataques de negação de serviço, scans, worms, vírus e outros tipos de males ainda geram problemas a milhares de administradores de rede e até mesmo a usuários comuns. Apesar do desenvolvimento e aperfeiçoamento das técnicas de detecção, o número de falso positivo destes mecanismos ainda é preocupante. Desta forma, este estudo contribui com a discussão do uso da teoria da evidência de Dempster-Shafer aplicada como técnica de fusão de dados.

De forma geral este trabalho apresentou uma visão geral do mecanismo de fusão de dados e elementos da teoria da evidência, além de descrever estudos com fusão de dados visando à detecção de anomalias de rede.

Também desenvolveu um protótipo de ferramenta capaz de agregar dados sobre anomalias baseado na inferência de sensores espalhados na rede e inferir o real estado da rede monitorada. A validação do protótipo foi realizada utilizando-se dados sintéticos contendo diversos ataques gerados na rede do GPRT-UFPE.

Apesar dos resultados estarem em estado inicial e ainda existir um longo caminho até uma solução completa e definitiva, percebe-se que é possível e definitivamente positivo o uso da teoria da evidência como técnica de fusão de dados.

6.1. Dificuldades Encontradas

No decorrer da elaboração deste trabalho foram encontrados alguns desafios que, em muitos momentos, tornou lento e cansativo o ritmo de desenvolvimento, mas que ao serem transpostos geraram bons frutos ao trabalho final.

Entre os desafios encontrados podemos citar:

- Problemas no funcionamento da API com elementos da teoria da evidência de Dempster-Shafer. Foi necessário um estudo aprofundado para compreensão e manipulação das bibliotecas. Devido à quase inexistente referência e a baixa qualidade da documentação, a curva de aprendizado foi muito maior do que a planejada. Em compensação, após o total entendimento, que agregou um peso maior ao projeto. Além disso, fomos capazes de contribuir no fórum da ferramenta.
- Outra dificuldade foi à definição de eficientes sensores para a detecção de anomalias capazes de serem adaptados para a geração das crenças.

6.2. Trabalhos Futuros

Pode-se como trabalhos futuros:

- Modificar o ADS-Fusion para ser capaz de operar com coleta e análise de dados em modo on-line, gerando assim alertas em tempo real.
- Implementar outros tipos de regras de conflito para o módulo de fusão TDS como, por exemplo, os propostos em [32][33], visando um melhor tratamento na resolução de conflitos de evidências.
- Desenvolver ou adaptar outros sensores de forma a aumentar a abrangência de anomalias que possam ser detectadas, contribuir com uma maior eficácia do ADS-Fusion.

REFERÊNCIAS

[1] Sundaram, A. (1996) An Introduction to Intrusion Detection. *ACM Crossroads 2.4*.

[2] Anderson, J.P. (1980) *Computer Security Threat Monitoring and Surveillance*. Technical report, Fort Washington, Pennsylvania, Abril.

[3] Lunt, T. F. (1993) A survey of intrusion detection techniques. Em *Computers and Security*, 12, páginas 405-418.

[4] Sherif, J.S., e Dearmond, T.G. (2002) Intrusion detection: systems and models. Em *Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises.*

[5] Debar, H., Dacier, M., e Wespi, A. (1999) Towards a taxonomy of intrusion-detection systems. *Computer Networks* 31.

[6] Gao, J., Hu, G., Yao, X., e Chang, R. (2006) Anomaly Detection of Network Traffic Based on Wavelet Packet. Em *Asia-Pacific Conference on Communications (APPC´06)*.

[7] Ramanathan, A. (2002) Wades: A tool for distributed denial of service attack detection. M.S. thesis, Texas A&M University.

[8] Li, L., e Lee, G. (2005) DDoS attack detection and wavelets. Em *Telecommunication Systems*, Vol. 28, No. 3-4, páginas 435-451, Março.

[9] Abry, P., Borgnat, P., e Dewaele, G. (2007) Sketch based anomaly detection, identification and performance evaluation. Em *IEEE/IPSJ SAINT Measurement Workshop*, páginas 80-84.

[10] Scherrer, A., Larrieu, N., Owezarski, P., Borgnat, P., e Abry, P. (2007) Non-Gaussian and Long Memory Statistical Characterizations for Internet Traffic with Anomalies. *IEEE Transactions on Dependable and Secure Computing*, volume 4, páginas 56-70.

[11] Lakhina, A., Crovella, M., e Diot, C. (2005) Mining anomalies using traffic feature distributions. *ACM SIGCOMM Computer Communication Review,* 35(4), páginas 217-228. ACM Press.

[12] Karagiannis, T., Papagiannaki, K., e Faloutsos, M. (2005) BLINC: Multilevel traffic classification in the dark. *ACM SIGCOMM Computer Communication Review,* 35(4), páginas 229-240. ACM Press.

[13] Xu, K., Zhang, Z-L., e Bhattacharrya, S. (2005) Profiling internet backbone traffic: Behavior models and applications. *ACM SIGCOMM Computer Communication Review,* 35(4), páginas 169-180. ACM Press.

[14] Bernaille, L., Teixeira, R., Akodjenou, I., Soule, A., e Salamatian, K. (2006) Traffic Classification on the Fly. *ACM SIGCOMM Computer. Communication Review,* 36(2), páginas 23-26, Abril. ACM Press.

[15] Dasarthy, Decision Fusion, IEEE Computer Society Press, 1994

[16] Waltz, E., and Llinas, J. (1990) *Multisensor Data Fusion*. Artech House Boston, London.

[17] Siaterlis, C., Maglaris, B., e Roris, P. (2003) A novel approach for a Distributed Denial of Service Detection Engine. Relatório Técnico, Network Management and Optimal Design (NETMODE) Lab; National Technical University of Athens.

[18] Hall, D. (1992) *Mathematical Techniques in Multisensor Data Fusion*. Artech House, Norwood, Massachussets.

[19] Chatzigiannakis, V., Papavassiliou, S., Androulidakis, G., e Maglaris, B. (2006) On the realization of a generalized data fusion and network anomaly detection framework. Em

Fifth International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP'06), Patra, Greece, July.

[20] Dempster, A. P. (1967) Upper and Lower Probabilities Induced by a Multivalued Mapping. Em *Annals Mathematics Statistics*, 38, páginas 325-339.

[21] Dempster, A. P. (1967) Upper and Lower Probability Inferences Based on a Sample from a Finite Univariate Population. Em *Biometrika*, 54, páginas 515-528.

[22] Shafer, G. (1976) *A mathemathical theory of evidence*. Princeton, Princeton University Press.

[23] Caswell, B., e Roesch, M. (2008) Snort: The open source network intrusion detection system. Disponível em: <http://www.snort.org>

[24] Chen, Q., and Aickelin, U. (2006) Anomaly Detection Using the Dempster-Shafer Method. Em *International Conference on Data Mining, DMIN 2006*, Las Vegas, Nevada, USA.

[25] Tian, J., Zhao, W, Du, R., e Zhang, Z. (2005) D-S Evidence Theory and its Data Fusion Application in Intrusion Detection. Em *The Sixth International Conference on Parallel and Distributed Computing Applications and Technologies*. Páginas 115 - 119

[26] Intrusion Detection Working Group, "Intrusion detection message exchange format data model and extensible markup language (XML) document type definition." Internet-Draft, January 2003.

[27] PCAP Public Repository. Disponível em: <http://www.nrg.ee.lbl.gov>

[28] Mirkovic, J., Prier, G., e Reiher, P. (2002) Attacking DDoS at the Source. Em *Proceedings of 10th IEEE International Conference on Network Protocols*, páginas 312-321, Paris, França, Novembro.

[29] Mirkovic, J., Robinson, M., Reiher, P., e Kuenning, G. (2003) Forming Alliance for DDoS Defenses. Em *Proceeding of the New Security Paradigms Workshop (NSPW 2003)*, ACM Press, páginas 11-18, Agosto.

[30] Aschoff, R. R. (2007) ChkModel: *Um Mecanismo de Defesa Contra Ataques DDoS*. Trabalho Final de Graduação. Centro de Informática. Universidade Federal de Pernambuco.

[31] LRDE-EPITA. *EvidenZ*. Disponível em: < http://www.lrde.epita.fr/cgi-bin/twiki/view/Projects/Evidenz>

[32] Sentz, K. e Ferson, S. (2002) *Combination of Evidence in Dempster-Shafer Theory*. Relatório Técnico. Disponível em:
<http://www.sandia.gov/epistemic/Reports/SAND2002-0835.pdf>

[33] Campos, F. F (2005) *Uma extensão a matemática da evidência*. Tese de Doutorado. Centro de Informática. Universidade Federal de Pernambuco.

[34] Intrusense Packit. - Network Injection and Capture. Disponível em: <http://www.intrusense.com/software/packit >