INEZ FREIRE RAGUENET

UM MODELO DE COMPOSIÇÃO DE DETECTORES DE INTRUSÃO HETEROGÊNEOS BASEADO EM CONJUNTOS DIFUSOS

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática

CURITIBA

2007

INEZ FREIRE RAGUENET

UM MODELO DE COMPOSIÇÃO DE DETECTORES DE INTRUSÃO HETEROGÊNEOS BASEADO EM CONJUNTOS DIFUSOS

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática

Área de Concentração: *Ciência da Computação*
Orientador: Prof. Dr. Carlos Alberto Maziero

CURITIBA
2007

Raguenet, Inez Freire
R145m Um modelo de composição de detectores de intrusão heterogêneos baseado
2007 em conjuntos difusos / Inez Freire Raguenet ; orientador, Carlos Alberto Maziero. – 2007.
90 f. : il. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2007
Inclui bibliografia

1. Informática. 2. Detecção de intrusão. 3. Modelagem matemática. 4. Conjuntos difusos. I. Maziero, Carlos Alberto. II. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática. III. Título.

CDD 20. ed. – 004

Sumário

Sumário ..... **iv**
Lista de Figuras ..... **vii**
Lista de Tabelas ..... **ix**
Lista de Abreviaturas ..... **xi**
Resumo ..... **xiii**
Abstract ..... **xiv**

Capítulo 1 Introdução **1**
1.1 Objetivos **1**
1.2 Metodologia **2**
1.3 Estrutura do Texto **3**
Capítulo 2 Detecção de Intrusão **5**
2.1 Propriedades Fundamentais de Segurança **6**
2.2 Definições **7**
2.3 Fragilidades **8**
2.4 Arquitetura de um Sistema de Detecção de Intrusão **10**
2.5 Classificação dos IDSs **12**
2.5.1 Quanto à Origem dos Dados (caixa "Audit Source Location") 13
2.5.2 Quanto à Forma de Detecção (caixa "Detection Method") 14
2.5.3 Quanto à Periodiocidade (caixa "Usage Frequency") 15
2.6 Características Desejáveis **16**
2.7 Falsos Positivos e Falsos Negativos **17**
2.8 Escalabilidade de IDS **18**
2.9 Conclusão **20**
Capítulo 3 Composição de Detectores de Intrusão **21**

3.1 IDS Distribuídos 21
3.2 Integração de Alertas..... 24
3.2.1 Scyllarus..... 25
3.2.2 ACCs..... 25
3.2.3 IDMEF/IDXP 26
3.3 Diversidade de Projetos 27
3.4 Diversidade de Projetos em Detecção de Intrusão..... 28
3.5 Composição de IDSs 31
3.6 Conclusão 34
Capítulo 4 Um Modelo Matemático para a Composição de Detectores de Intrusão 35
4.1 Modelagem usando Teoria dos Conjuntos..... 36
4.2 Atribuição do Grau de Importância 39
4.4 Conclusão 44
Capítulo 5 Avaliação do Modelo 45
5.1 As Curvas ROC 45
5.2 Ambiente de Testes 48
5.3 Descrição dos Experimentos..... 50
5.4 Testes Realizados 50
5.5 Análise dos Resultados Obtidos 56
5.6 Trabalhos Correlatos..... 61
5.7 Conclusão 64
Capítulo 6 Conclusão..... 67
6.2 Contribuição 68
6.3 Trabalho futuros..... 69
Referências..... 70

Apêndices

Apêndice A – Cálculo dos Valores-Limite *Li* e *Ls* Referentes aos Experimentos do Capítulo 5..... 74
Apêndice B – Comparativo da Taxa de Acertos entre o CIDS e cada IDS 77

Apêndice C – Comparativo entre CIDS e Snort..... 79
Apêndice D – Comparativo entre CIDS e KFSensor 82
Apêndice E – Comparativo entre CIDS e HoneyBOT 85
Apêndice F – Comparativo entre CIDS e X-Ray..... 88

Lista de Figuras

Figura 2.1 IDS típico [Debar 1999] 10
Figura 2.2 - Estrutura conceitual de um IDS [Verwoerd 2002] 11
Figura 2.3 – Características de um IDS [Debar 1999] 12
Figura 2.4 – Modelo do IDS baseado em anomalias [Maxion 2005] 15
Figura 2.5 - Falsos Positivos e Negativos 17
Figura 2.6 - Espalhador de Tráfego [Bezerra de Mello 2004] 19
Figura 3.1 - IDS Distribuído 22
Figura 3.2 - Modelo de Classes da Mensagem IDMEF [Zaraska 2003] 27
Figura 3.3 – Arquitetura do sistema HACQIT [Reynolds 2003] 30
Figura 3.4 - Análise 31
Figura 3.5 - Consolidação 32
Figura 3.6 - Endosso 32
Figura 3.7 - Confirmação 32
Figura 3.8 – Arquitetura do gerente [Yu-Sung 2003] 34
Figura 4.1 – Diagrama de Venn para Modelagem com um Detector 37
Figura 4.2 - Modelagem com Dois Detectores 38
Figura 4.3 – Gráfico da variação do Grau de Importância de um ataque detectado por todos os detectores de um CIDS 44
Figura 5.1 – Espaço ROC 47
Figura 5.2 – Curva ROC 48
Figura 5.3 – Curva ROC do CIDS 58
Figura 5.4 – Superposição de Curvas ROC para os 4 IDSs e para o CIDS 60
Figura 5.5 – Diagrama de Venn para um sistema baseado em anomalias [Leckie 2004] 62
Figura 5.6 – Modelagem de um ataque usando *spicules* [Vert 1998] 63
Figura A.1 – Traçado da Curva ROC usando somente valores úteis 76
Figura C.1 – Traçado da Curva ROC - Snort 81

Figura D.1 – Traçado da Curva ROC - KFSensor 84
Figura E.1 – Traçado da Curva ROC - HoneyBOT 87
Figura F.1 – Traçado da Curva ROC para o X-Ray..... 90

Lista de Tabelas

Tabela 4.1 – Graus de Importância 42
Tabela 4.2 – Análise do Grau de Importância – CIDS com até 6 detectores 43
Tabela 5.1 – Matriz de Confusão 46
Tabela 5.2 – Ataques deflagrados pelo Nessus 51
Tabela 5.3 – Falsos Positivos capturados pelos detectores 52
Tabela 5.4 – Ataques detectados pelos IDSs 53
Tabela 5.5 – Falsos Positivos detectados pelos IDSs 54
Tabela 5.6 – Grau de Importância dos Eventos 55
Tabela 5.7 – Taxa de Falsos Positivos x Taxa de Verdadeiros Positivos 57
Tabela 5.8 – Graus de Importância para os eventos do CIDS 58
Tabela 5.9 –Segurança e Vivacidade do CIDS 59
Tabela 5.10 –Comparativo no Número de Verdadeiros Positivos 60
Tabela A.1 – Cálculo do Grau de Importância 74
Tabela A.2 – Parâmetros da Curva ROC do CIDS 75
Tabela B.1 – Cálculo do Número de Acertos de cada IDS (VP) 77
Tabela B.2 – Comparativo do Número de Acertos (VP) 78
Tabela C.1 – IDS Snort 79
Tabela C.2 – Parâmetros da Curva ROC do Snort (valores úteis) 80
Tabela C.3 – Comparativo do número de Alarmes Falsos Snort x CIDS 81
Tabela D.1 –KFSensor 82
Tabela D.2 – Parâmetros da Curva ROC do KFSensor (valores úteis) 83
Tabela D.3 – Comparativo do número de Alarmes Falsos KFSensor x CIDS 84
Tabela E.1 –HoneyBOT 85
Tabela E.2 – Parâmetros da Curva ROC do HoneyBOT (valores úteis) 86
Tabela E.3 – Comparativo do número de Alarmes Falsos HoneyBOT x CIDS 87
Tabela F.1 – IDS X-Ray 88
Tabela F.2 – Parâmetros da Curva ROC do XRay (valores úteis) 89

Tabela F.3 – Comparativo do número de Alarmes Falsos X-Ray x CIDS..... 90

Lista de Abreviaturas

ACC	Aggregation and Correlation Component
API	Application programming interface
CAS	Central Analysis Server
CIA	Confidencialidade, Integridade, Autenticidade
CIDF	Common Intrusion Detection Framework
CIDS	Compound Intrusion Detection System
COTS	Commercial-off-the-Shelf
CPU	Central Processing Unit
dIDS	Distributed Intrusion Detection System
DNS	Domain Naming System
DoS	Denial-of-Service
ED	Elementary Detector
FN	Falsos Negativos
FP	Falsos Positivos
FTP	File Transfer Protocol
HIDS	Host-based Intrusion Detection System
ICMP	Internet Control Message Protocol
IDMEF	Intrusion Detection Message Exchange Format
IDS	Intrusion Detection System
IDWG	Intrusion Detection Working Group
IDXP	Intrusion Detection Exchanging Protocol
IESG	The Internet Engineering Steering Group
IETF	Internet Engineering Task Force
IIS	Internet Information Services
IP	Internet Protocol
ISCS	Intruder System Call Sequences
ISP	Internet Services Provider
MAC (1)	Mediator/Adapter/Controller
MAC (2)	Media Access Control

MIME	Multipurpose Internet Mail Extensions
NIDS	Network-Based Intrusion Detection System
NSCS	Normal System Call Sequences
NVS	N-Version Software
ROC	Receiver Operating Characteristic
RPC	Remote Procedure Call
SDDI	Sistema Difuso de Detecção de Intrusão
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
TCP	Transmission Control Protocol
TEC	Tivoli Enterprise Console
UDP	User Datagram Protocol
VN	Verdadeiros Negativos
VP	Verdadeiros Positivos
XML	Extensible Markup Language

Resumo

A capacidade de detecção de um IDS depende de diversos fatores, incluindo sua arquitetura interna e os algoritmos utilizados. Assim, detectores distintos poderão apresentar comportamentos distintos quando submetidos ao mesmo fluxo de eventos. A teoria de diversidade de projetos vem sendo usada com sucesso na área de tolerância a faltas e também pode trazer benefícios na área de detecção de intrusão. O objetivo deste trabalho é propor uma modelagem matemática baseada na teoria de conjuntos difusos para a composição de detectores de intrusão heterogêneos que analisam o mesmo fluxo de eventos. Com esse modelo, mostrou-se que é possível combinar os resultados individuais de cada detector em um detector composto capaz de gerar resultados de melhor qualidade.

Abstract

The performance of an intrusion detector depends on several factors, like its internal architecture and the algorithms employed. Thus, distinct detectors can behave distinctly when submitted to the same event flow. The project diversity theory has been successfully used in the fault tolerance domain, and can bring benefits to the intrusion detection area. The objective of this work is to propose a mathematical model, based on the fuzzy set theory, for the composition of heterogeneous intrusion detectors analyzing the same event flow. This model shows how to combine the individual detectors' results into a global result with better quality.

Capítulo 1

Introdução

Pesquisas recentes mostram que o conceito de diversidade de projetos pode ser uma tendência na criação de sistemas mais seguros [Littlewood 2004]. Porém, a falta de estudos específicos do desempenho de um sistema redundante baseado em diversidade de projetos, em confronto com um sistema simples, sem redundância e de projeto único, não permite afirmar que um sistema composto por vários sub-sistemas distintos pode ser melhor do que um sistema de projeto único.

Para analisar a situação acima, escolheu-se a área de segurança, mais especificamente, a área de detecção de intrusão. Nesta área, os sistemas de detecção de intrusão (IDS), apesar de já apresentarem versões cujos projetos se diferenciem, ainda não existe uma formalização do conceito de diversidade de projetos e tampouco uma modelagem para representar diversos sistemas de detecção trabalhando em conjunto.

1.1 Objetivos

Este trabalho visou, a princípio, criar um modelo matemático capaz de representar não só um IDS, mas *n* detectores funcionando paralelamente, sobre um mesmo conjunto de eventos. O modelo poderia apresentar, inclusive, resultados numéricos com relação ao "desempenho" do sistema de detectores paralelos (ou IDS-composto, ou CIDS), em confronto com o desempenho de cada IDS escolhido para o CIDS. Ao final, se o projeto fosse bem sucedido, seria possível, por exemplo, dizer que "o CIDS é melhor do que um IDS".

Portanto, o objetivo deste trabalho é o de criar uma modelagem matemática para o funcionamento de um sistema de detecção de intrusão (IDS) genérico. O modelo deve ser capaz de mostrar o funcionamento de um IDS pela caracterização de todos os eventos inerentes a este sistema como, por exemplo, a identificação correta ou errônea de eventos normais e ataques .

A partir desta modelagem proposta, o trabalho tem os seguintes objetivos específicos:

- quantificar o desempenho de um IDS, baseando-se a análise na quantidade de acertos e erros que o IDS gera;
- criar um IDS-composto (CIDS) usando-se o conceito de diversidade de projetos;
- analisar, numericamente, o desempenho do IDS-composto (CIDS) e comparar os resultados obtidos em cada IDS individual usado no CIDS e os resultados do CIDS propriamente dito;
- verificar se, de acordo com as pesquisas atuais, os resultados de um sistema composto baseado em diversidade de projetos podem ser melhores do que os resultados dos sistemas individuais que o compõem.

1.2 Metodologia

A metodologia empregada para o desenvolvimento do trabalho contou com:

- Levantamento dos principais conceitos de segurança e de detecção de intrusão;
- Levantamento do estado da arte em detecção de intrusão e diversidade de projetos;
- Estudo da Álgebra Fuzzy e determinação da validade do uso da Teoria dos Conjuntos Difusos na modelagem;
- Desenvolvimento de um modelo matemático para modelagem dos IDSs usando a Teoria dos Conjuntos e posterior extensão para a Teoria dos Conjuntos Difusos;
- Elaboração de testes em laboratório em ambiente aberto (internet) para observação das diferenças entre o uso de um IDS e o uso de mais de um IDS simultaneamente;

- Elaboração de testes em laboratório em ambiente fechado para observação das diferenças entre o uso de um IDS e o uso de mais de um IDS simultaneamente em ataques simulados;
- Estudo da teoria das Curvas ROC como elemento de análise comparativa na avaliação de um IDS de acordo com sua calibragem;
- Análise de resultados dos testes feitos com IDSs individuais em ambiente fechado e aplicação da teoria das Curvas ROC para verificação do melhor ponto de calibragem para um CIDS composto pelos IDSs testados;
- Comparação dos resultados de desempenho do CIDS com cada IDS individual;
- Conclusões

1.3 Estrutura do Texto

Este trabalho foi desenvolvido em 6 capítulos, brevemente descritos a seguir. O Capítulo 2 descreve os conceitos básicos de segurança, de detecção de intrusão e de sistemas de detecção de intrusão (IDS). Também trata de alguns problemas na área de detecção de intrusão como escalabilidade e alarmes falsos. O Capítulo 3 faz uma introdução ao assunto principal da dissertação, apresentando as técnicas de uso de mais de um detector de intrusão, como os dIDS (IDSs distribuídos), os mecanismos de integração de alertas e o conceito de diversidade de projetos na detecção de intrusão, finalizando com a introdução do assunto da composição de IDSs, os CIDS. O Capítulo 4 apresenta uma proposta de modelagem matemática para IDSs, estensível para CIDS, o conceito de fuzzificação, a proposta do uso do "grau de importância" para um evento detectado por um IDS e algumas proposições que serão aprofundadas no Capítulo 5. Esse, por sua vez, apresenta uma avaliação prática do modelo proposto através da criação e testes de um CIDS em particular. Com os resultados dos testes em laboratório com fluxo controlado de eventos são propostos, através da metodologia de análise por Curvas ROC, os valores-limites de variáveis que foram apresentadas no Capítulo 4. Finalmente, o CIDS foi avaliado e seu desempenho foi comparado com cada um dos IDSs escolhidos. Os resultados estão na seção 5.5. O Capítulo 6 conclui a dissertação e apresenta algumas perspectivas de trabalhos futuros.

Capítulo 2

Detecção de Intrusão

Um dos grandes problemas atuais nos sistemas de informação são as brechas que os sistemas de computação, servidores e computadores em geral, apresentam. Explorando estas brechas, é possível invadi-los, roubar dados ou estabelecer um clima de desordem dentro de um sistema.

De uma forma geral, os termos "ataque" e "invasão" (ou "intrusão") podem se confundir. Por exemplo, [Dasgupta 2001] aceita a definição de "intrusão" como "qualquer conjunto de ações que tente comprometer a integridade, confidencialidade ou disponibilidade de um recurso". No entanto, neste trabalho, serão consideradas definições mais objetivas onde os "ataques" são tentativas de explorar pontos fracos de um sistema, enquanto que as "invasões" podem ser consideradas como ações cometidas em decorrência de ataques bem sucedidos.

Ataques devem ser identificados antes que se tornem uma invasão ou que indisponibilizem o sistema ao qual estão direcionados, e é para isso que existem os programas de detecção de intrusão (IDS – *Intruder Detection System*). A função principal de um IDS é a de monitorar o comportamento de um sistema e identificar possíveis comportamentos anômalos, caracterizando-os como, por exemplo, uma exploração mal-intencionada dos serviços que o sistema presta. Espera-se que o IDS identifique este comportamento suspeito e gere alertas. Uma vez dado o alerta, sistemas envolvidos no ambiente global de segurança podem iniciar ações voltadas para a anulação do ataque ou de suas conseqüências.

2.1 Propriedades Fundamentais de Segurança

A segurança no ambiente da computação é normalmente associada a três propriedades, fundamentais, que englobam os conceitos de *confidencialidade*, *integridade* e *disponibilidade*:

- *confidencialidade* – garantia de que a informação não será acessada por agentes não autorizados;
- *integridade* – garantia de que a informação está protegida contra alterações não-autorizadas.
- *disponibilidade* – garantia de que os sistemas sempre estarão em estado operacional, de que a informação sempre estará disponível para o acesso por agentes autorizados para tal.

Em linhas mais gerais, para garantir a segurança nos sistemas de informação e na comunicação de dados, outros princípios básicos precisam ser respeitados:

- *autenticidade* – garantia da origem da informação
- *irrefutabilidade* (*non-repudiation*) – garantia de que a informação que chega a um destino, de fato, foi transmitida pelo seu remetente; que seu remetente não pode negar o fato de que a transmitiu e que o destinatário não pode negar o fato de que a recebeu. Este conceito pressupõe o conceito de *autenticidade*, a identificação apropriada dos agentes envolvidos.
- *controle de acesso* – garantia de que somente usuários autorizados poderão alocar recursos e serviços aos quais têm direito;

O objetivo final de um sistema de detecção de intrusão é o de estabelecer um ambiente controlado, onde qualquer tentativa de violar as garantias acima seja corretamente identificada. Por exemplo, se o sistema de detecção de intrusão for capaz de alertar para a ocorrência de ataques do tipo *denial-of-service* (DoS) em um serviço, e se os administradores do sistema forem capazes de controlar os ataques, haverá chances de não se violar a garantia da *disponibilidade*.

2.2 Definições

Algumas definições devem ser feitas sobre os termos a serem utilizados nesta dissertação. Os termos mais utilizados, como *ataque*, *atacante*, *vulnerabilidade*, *invasão*, *detecção de invasão* e *sistema de detecção de invasão* podem ser entendidos, conforme [Zamboni 2001], [Allen 2000] e a RFC2828 do IESG [IESG 2007] como:

- **Intrusão ou Invasão:** Qualquer conjunto de ações cuja intenção seja a de comprometer a integridade, confidencialidade ou disponibilidade de um recurso computacional.
- **Ataque**: Ação de um sujeito, o atacante, sobre uma vítima em potencial; um conjunto de eventos sobre um objeto que podem gerar conseqüências negativas sobre a integridade deste objeto.
- **Atacante ou invasor**: Aquele que dirige o ataque à vítima, que resulta em invasão; a entidade, ou pessoa, que realizou um ataque bem sucedido. A vítima pode ser um host, sítio, rede, uma empresa, etc.
- **Vulnerabilidade**: Uma característica ou um conjunto de características de um sistema que permitam a um atacante colocá-lo em um estado de funcionamento não desejado pelos seus responsáveis; características que permitem ao atacante colocar o sistema em risco de se chegar a um estado não desejado de funcionamento, ou, características do sistema que possam colocar sua segurança em risco.
- **Detecção de invasão**: capacidade de identificar tentativas de comprometer a integridade, confidencialidade ou disponibilidade de um recurso computacional .
- **Sistema de detecção de invasão**: Um sistema computacional, possivelmente composto de software e hardware, cuja função seja a de detectar invasões, não necessariamente evitando a invasão, mas obrigatoriamente reportando-a para um operador responsável pelo sistema; ou, combinação de hardware e software para monitoramento e coleta de informações da rede e dos sistemas; estes dados serão analisadas posteriormente para determinar se ocorreu um ataque ou uma invasão. Algums sistemas de detecção podem ser programados para reagir imediatamente a um ataque.

- **Tecnologias para detecção de invasão**: Um termo mais amplo que o anterior, significando a combinação de sistemas de detecção de invasão com analisadores de invasões e outras ferramentas adicionais, como analisadores de pacotes e analisadores de arquivos de log, cujo intuito final é o de gerar indicadores precisos da ocorrência de ataques e invasões.

2.3 Fragilidades

Ao se implementar uma política de proteção contra invasão, deve-se ter em mente que as invasões não ocorrem somente de fora para dentro; grande parte dos estragos feitos por agentes não autorizados podem ter sido deflagrados por indivíduos operando nas próprias instalações. Os casos mais comuns de invasão ocorrem quando um usuário não consegue manter em segredo suas senhas ou permite que outros se utilizem delas para acessar recursos particulares (conta de correio eletrônico, por exemplo). Em casos mais sofisticados, os invasores se utilizam de recursos de engenharia social, falhas no sistema de segurança, falhas do sistema operacional ou ainda de programas desenvolvidos especialmente para invadir e tomar posse de um computador (*rootkits*). Casos de "defacement" (troca de conteúdo de um *website*) são comuns, principalmente quando os servidores web ou as aplicações nele instaladas estão vulneráveis pela falta de aplicação de correções do sistema.

Os principais tipos de invasão se baseiam em:

- **Quebra de sigilo:** Basicamente, pela descoberta dos mecanismos de acesso a algum serviço protegido. Por exemplo, no caso das senhas: muitos usuários de sistemas protegidos por senha usam o próprio nome, ou o nome da esposa/do marido, dos filhos, do cachorro, como senha; muitas vezes, usam a palavra "senha" ou "password", ou simplesmente não usam nada. Os sistemas podem ser configurados para rejeitar a criação de senhas óbvias ou estabelecer critérios para tornar as senhas seguras, porém nem todos os sistemas dispõem destes recursos. Para forçar a quebra de sigilo, também podem ser deflagrados os ataques de força bruta, onde combinações de nomes de usuário e senha são testadas exaustivamente com a ajuda de programas próprios.
- **Vulnerabilidades nos sistemas e nos protocolos**: tipicamente, falhas em um sistema que permitem comprometer, de alguma forma, sua integridade.

A exploração mais comum da vulnerabilidade se faz com a técnica do *buffer overflow*, quando o invasor consegue se aproveitar de falhas em sistemas e acessar áreas privilegiadas de programas em execução na memória de um computador.

- **Captura de tráfego de rede**: Programas de captura – *sniffers* – são utilizados para ver todo o tráfego em um segmento da rede, principalmente em redes Ethernet. A verificação do conteúdo dos pacotes não-criptografados capturados em uma sessão de *sniffing* pode trazer informações críticas, como chaves de acesso a serviços ou outros dados úteis para uma invasão.
- **Configuração inadequada**: Instalações pelo *default* e administradores preguiçosos ou mal-treinados criam o ambiente propício para invasão. Diversos sistemas se instalam com uma configuração aberta para facilitar seu uso e os administradores do sistema deixam como está por simples preguiça ou desinformação. Os exemplos mais contundentes são os do servidor SQL da Microsoft e o servidor MySQL AB que, na instalação, criam seus administradores, "sa" e "root" respectivamente, com a senha em branco. Em outras situações os administradores, inadvertidamente, instalam serviços desnecessários nos servidores, criando novas portas de entradas para os invasores. A falta de segurança de uma rede é medida pela seu ponto mais fraco: uma vulnerabilidade em um *host* desprotegido pode abrir toda a rede para o invasor.

Atualmente, os ataques mais comuns podem ser classificados como:

- **DoS**: ataques que visam interromper os serviços oferecidos por um host, normalmente se aproveitando de toda a capacidade útil de um certo serviço oferecido pelo host.
- **Hijack**: ataques que visam o "seqüestro" de um serviço, desviando-o de suas funções normais para atender aos serviços programados pelo invasor. Por exemplo, a exploração de servidores SMTP abertos para *relay* para remeter spam, ou a reprogramação das páginas de pesquisa de um *browser*.

- **Exploits**: exploração das vulnerabilidades de um sistema operacional ou de um serviço; na maioria das vezes, permite que o invasor tome posse do computador e atue como super-usuário no sistema.

2.4 Arquitetura de um Sistema de Detecção de Intrusão

Os IDSs nasceram como sistemas monolíticos, com análise periódica de lotes de arquivos de logs, evoluindo para sistemas distribuídos operando em tempo-real. A visão mais simples do IDS o define como um detector que processa informações provenientes do sistema que deve ser protegido. Este detector usa, basicamente, três tipos de informações: dados provenientes de bases de conhecimento de ataques (arquivo de assinaturas, por exemplo), informações relativas ao estado do sistema no momento e dados provenientes de sistemas de auditoria que identificam o que está ocorrendo dentro do sistema [Debar 1999]. A Figura 2.1 exibe a arquitetura de um IDS típico.

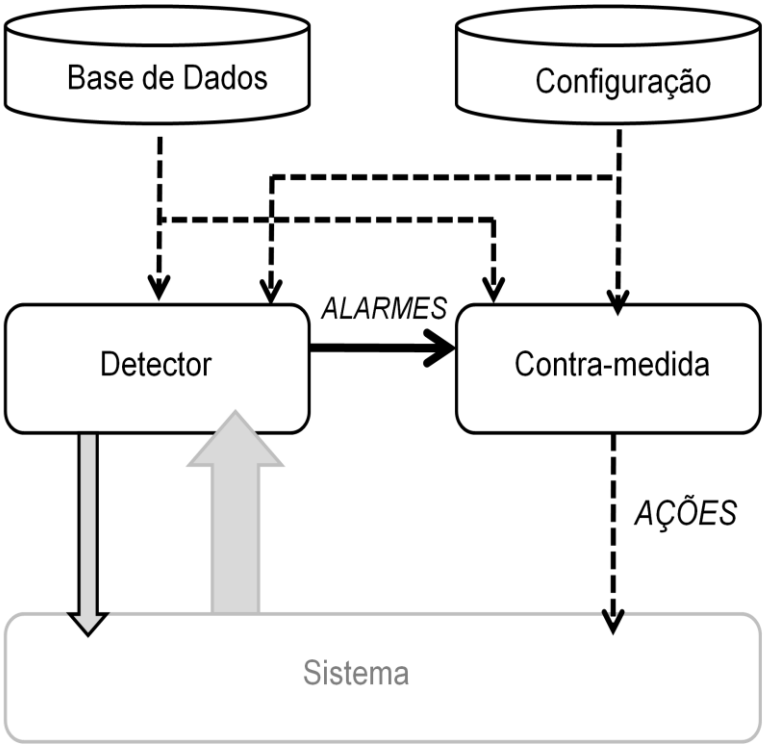

Figura 2.1 IDS típico [Debar 1999]

Atualmente, os IDSs podem ser desmembrados em 4 módulos, cada um com uma tarefa específica: o sensor, o monitor, o acionador (*resolver*), e o controlador [Verwoerd 2002], conforme ilustrado na figura 2.2.

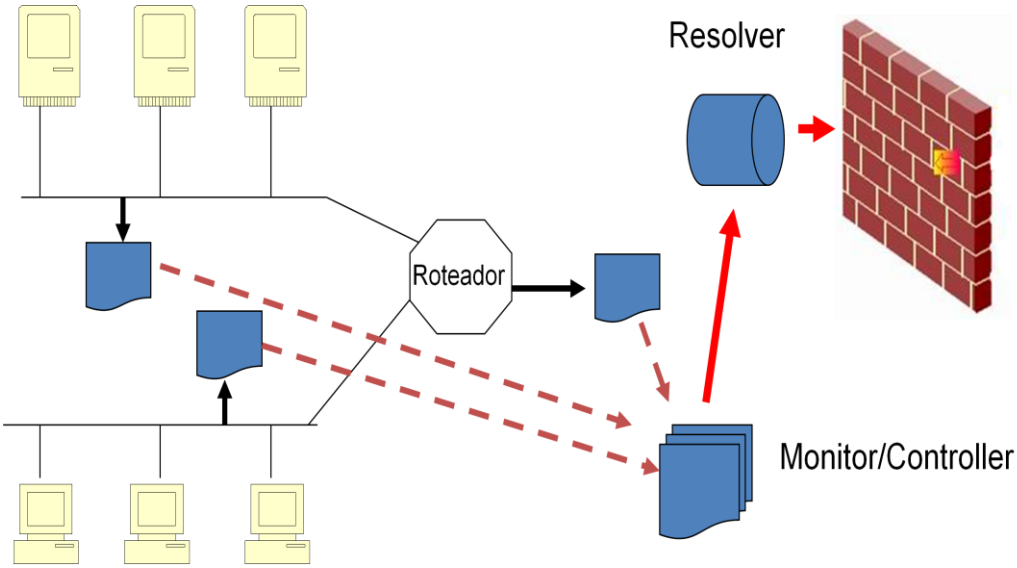

Figura 2.2 - Estrutura conceitual de um IDS [Verwoerd 2002]

- *Sensor*: responsável pela coleta de dados e implementado de acordo com o sistema no qual está sendo usado. Coleta dados de tráfego de rede, dos logs de sistema, etc,. e os traduz para eventos que o monitor irá interpretar;
- *Monitor*: é o componente principal do IDS, recebe os eventos provenientes dos sensores e os processa de acordo com modelos de comportamento conhecidos pelo IDS. Estes eventos podem gerar atualizações nos modelos ou disparar alertas, estes últimos encaminhados para monitores de mais alto nível ou para os *resolvers*;
- *Resolver*: este módulo recebe os dados considerados suspeitos pelos monitores e decide a ação ser tomada – geração de registros, alteração do comportamento de componentes de mais baixo nível, reconfiguração de componentes de segurança (novas regras no firewall, por exemplo) ou notificação dos operadores;
- *Controlador*: centralizador de operações de administração, configuração e atualização dos demais componentes do IDS.

2.5 Classificação dos IDSs

Quandos os primeiros sistemas de detecção de invasão foram desenvolvidos, os sistemas-alvo eram *mainframes* e todos os usuários eram considerados locais, ou seja, estavam conectados diretamente ao sistema. Estes IDSs se apresentavam sob a forma de simples analisadores dos dados coletados pelos sistemas de auditoria dos *mainframes*, os *audit trails*, e relatavam os eventos considerados suspeitos dentre os registros analisados. Com o desenvolvimento das redes e o conseqüente espalhamento dos usuários e dos sistemas, os IDS foram modificados de modo a conseguir capturar os detalhes inerentes às redes, como a localização de um certo usuário e o tráfego gerado por ele. O advento da Internet criou uma necessidade maior de monitoramento dos ataques à rede como um todo. Ataques que se utilizam das técnicas de *spoofing*, por exemplo, não são facilmente detectados somente pelo exame dos logs de auditoria dos hosts [Debar 1999].

De um modo geral, os IDSs se classificam de acordo com 5 conceitos básicos, ilustrados na Figura 2.3. Alguns destes conceitos são explicados a seguir.

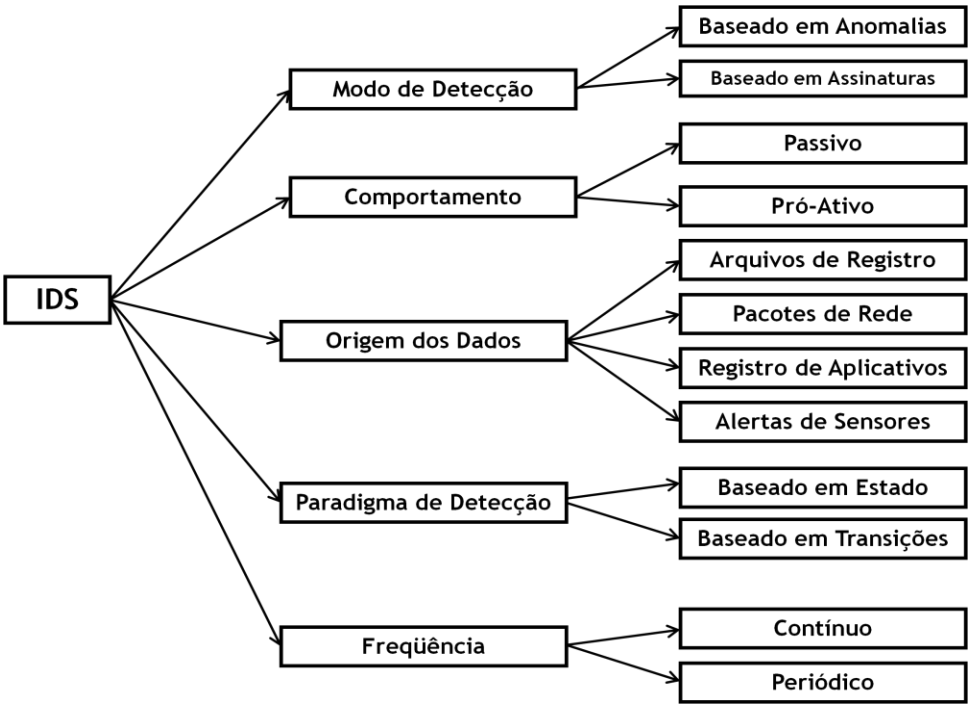

Figura 2.3 – Características de um IDS [Debar 1999]

2.5.1 Quanto à Origem dos Dados (caixa "Audit Source Location")

Conforme a origem da coleta dos dados, se feita nos logs de sistema e de auditoria de um host (baseados em host), ou nos pacotes capturados na rede (baseados em rede), ou nos logs de serviços ou aplicativos disponíveis em um host (baseado em aplicações).

- **Baseado em host (HIDS)**: um HIDS tem a responsabilidade de proteger um host específico analisando, principalmente, os logs de sistema e de auditoria, que dão informações sobre as atividades do sistema deste determinado host. A avaliação do conteúdo destes logs pode indicar um comportamento suspeito ou divergente das políticas de segurança vigentes. Basicamente, um HIDS pode atuar monitorando as atividades relacionadas com as tentativas de conexão ao host através da rede para detectar varreduras de portas (*portscan*) ou ainda monitorar as conexões não usuais de usuários ao host (log-in, log-out), as atividades de usuários privilegiados (root), a integridade do sistema de arquivos e o estado de arquivos de sistema.
- **Baseado em rede (NIDS)**: o NIDS analisa dados relacionados ao tráfego da rede local, capturado em uma interface de rede operando em modo promíscuo. A análise de pacotes é mais comumente utilizada nos NIDSs baseados em arquivos de assinaturas. As "assinaturas" são seqüências conhecidas de pacotes ou comandos que caracterizam os ataques mais conhecidos e são utilizadas para comparar e analisar o tráfego capturado. Também podem ser utilizadas as informações provenientes dos contadores SNMP, que são úteis quando já se conhece o comportamento "normal" do ambiente já que alterações substanciais nos contadores podem caracterizar um ataque.
- **Baseado em aplicações:** estes tipos de IDS são totalmente voltados para o aplicativo em si e analisam, basicament, as informações contidas nos registros (*logs*) dos aplicativos. Como exemplo, um IDS para serviços de FTP analisaria os logs de conexão no servidor FTP e seria capaz de alertar, por exemplo, para uma tentativa de DoS ao identificar a um número excessivo de tentativas de conexão geradas por um mesmo usuário ou um

único endereço IP. Estes IDSs são considerados mais completos e precisos, porém vulneráveis a ataques provenientes de outras fontes que não o aplicativo em si, por exemplo, por um *buffer overflow* no host que o hospeda.

2.5.2 Quanto à Forma de Detecção (caixa "Detection Method")

A função básica de um IDS é a de destacar as prováveis atividades anormais dentro do sistema que monitora, podendo fazer isto de dois modos:

- **Por anomalias**: neste modo, os IDSs operam criando um perfil do comportamento normal do sistema e analisando o comportamento corrente de acordo com os padrões da normalidade; traços de anormalidade são reconhecidos como possíveis ataques. Dependendo da abrangência do perfil criado, estes IDSs podem não ser muito precisos na sua detecção, gerando um alto número de falsos positivos; porém, têm a vantagem de serem capazes de detectar um ataque até então desconhecido pela comunidade simplesmente pelo reconhecimento do comportamento anormal gerado por este evento. [Maxion 2005] apresenta um modelo genérico de funcionamento do IDS baseado em anomalias, ilustrado na Figura 2.4. Este modelo mostra, inclusive, como ataques podem não ser detectados por um IDS baseado em anomalias.

- **Por base de conhecimento (assinaturas):** neste modo, os IDSs se utilizam de uma base de conhecimento ou arquivos de assinaturas, que contêm os padrões de funcionamento dos ataques conhecidos. Normalmente, funcionam em tempo real, verificando, por exemplo, o tráfego de entrada em um host, pacote a pacote. Estes IDSs são mais precisos do que os baseados em anomalias, gerando um número menor de falsos positivos porém não são capazes de detectar novos ataques (ainda não documentados) e dependem da atualização constante dos arquivos de assinaturas.

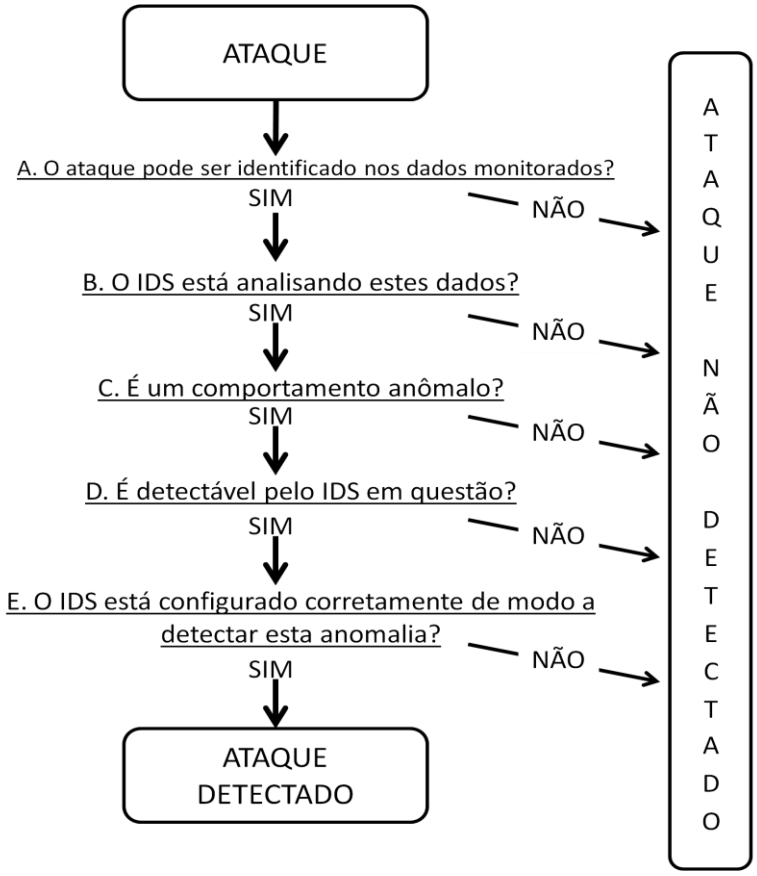

Figura 2.4 – Modelo do IDS baseado em anomalias [Maxion 2005]

2.5.3 Quanto à Periodiocidade (caixa "Usage Frequency")

Os IDS podem operar analisando os dados coletados continuamente ou periodicamente, como detalhado a seguir:

- **Tempo real**: neste modo, o IDS monitora os eventos do sistema em tempo real, mais comumente, o tráfego destinado a um host. Os algoritmos aplicados nestes IDSs, devido ao alto volume de processamento, são limitados a procedimentos rápidos e eficientes, sujeitos a algum tipo de perda, já que o processamento de todo o tráfego pode ser uma tarefa impossível de se realizar.
- **Por análise de logs e atributos**: os IDSs que operam neste modo analisam periodicamente os logs de sistema ou os atributos de segurança de um host em busca de entradas que possam identificar um ataque ou invasão. Existem algumas vantagens em se gerar e manter estes logs de auditoria

como, por exemplo, a identificação de falhas no sistema ou de usuários que estejam praticando atividades incompatíveis com os seus privilégios. A grande desvantagem é que o sistema de auditoria pode ser alvo de ataques bem sucedidos e perder sua integridade**.**

2.6 Características Desejáveis

Algumas características são desejáveis em um sistema de detecção de invasão, de acordo com [Crosbie 1999] :

- Deve estar continuamente em execução e não necessitar de intervenção de operadores humanos;
- Deve ser tolerante a falhas (*fault tolerant*), capaz de se reiniciar após *crashes*, retornando ao último estado consistente de operação;
- Deve ser resistente a tentativas de alteração no seu funcionamento por agentes não autorizados, dificultando alterações em seu funcionamento e monitorando acessos não-autorizados;
- Não deve causar impacto no sistema onde atua (*overhead* de CPU e memória, por exemplo);
- Deve ser configurável de acordo com as diretivas de segurança dos sistemas onde atua;
- Deve ser de fácil implementação, portável para outras plataformas e sistemas operacionais e de fácil entendimento e operação.
- Deve se adaptar a novos padrões de comportamento dentro do mesmo ambiente, reconhecendo novas aplicações, troca de atividades ou novos recursos.
- Deve ser capaz de detectar ataques, não classificando atividades legítimas como ataques (falsos positivos) e não falhando na identificação dos verdadeiros ataques (falsos negativos), sendo genérico o suficiente para detectar o maior número possível de ataques e gerar o alerta o mais rapidamente possível.

2.7 Falsos Positivos e Falsos Negativos

A maioria dos programas de detecção de tentativas de invasão nada mais faz do que comparar, sem espírito crítico, os dados coletados na rede ou no host com um perfil considerado normal (ou anormal). Por espírito crítico entendemos que esta análise é feita pacote a pacote, ou linha a linha de um log, sem considerar o contexto no qual os dados estão sendo analisados. Caberá a um operador, ou ao *resolver*, determinar se o alerta gerado pelo IDS realmente configura um ataque ou não. Este tipo de funcionamento do IDS pode deixar de fora diversos eventos que, de fato, configuram um ataque, como também pode incluir no alerta, por falha de configuração ou operação, acessos legítimos considerados como ataques. Estes casos limites são conhecidos como *falsos negativos* (ataques não percebidos e descartados) e *falsos positivos* (acessos legítimos identificados como ataques).

A Figura 2.5 mostra, graficamente, o universo de eventos que podem ser detectados por um IDS. Os eventos problemáticos estão nos quadrantes 3 e 4 da figura, respectivamente os falsos negativos e falsos positivos.

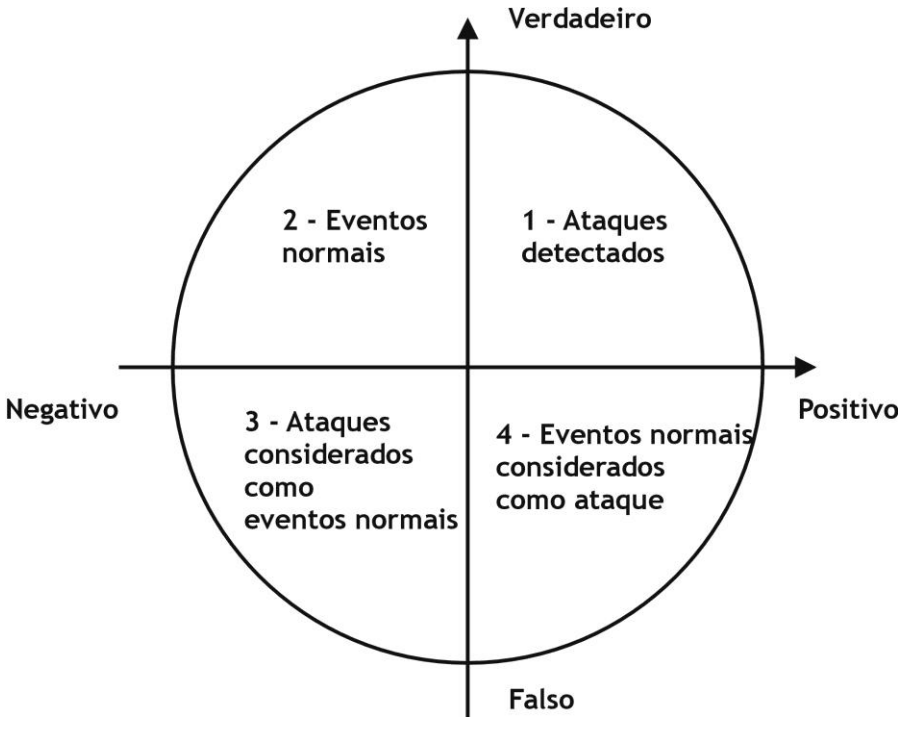

Figura 2.5 - Falsos Positivos e Negativos

No entanto, a questão do evento registrado ser legítimo ou não pode depender de cada contexto. Por exemplo, digamos que um IDS é configurado especificamente para gerar alarmes ao detectar atividades de varredura de portas. Os alarmes poderão:

- ser considerados falsos se os hosts de destino ao ataque não existem na rede (falso positivo), sendo descartados pelo *resolver*;
- ser considerados verdadeiros, pois o administrador da rede sabe que, um invasor, para detectar hosts vulnerávies, vai fazer varreduras em todo o *range* de endereços IP, que inclui diversos endereços inexistentes ou inativos na sua rede.

Algumas características especiais ajudam a diferenciar os falsos positivos de um verdadeiro positivo, como sugerido em [KFSensor 2003] :

- *pela quantidade de eventos*: se uma grande quantidade de eventos sucessivos é registrada na mesma porta, é provável que exista um mecanismo analisando as vulnerabilidades de algum serviço que o host possa dispor naquela porta; requisições mais espaçadas podem indicar uma atividade humana; um longo intervalo entre os eventos pode indicar um mecanismo automático, que o faz deliberadamente para enganar o IDS baseados em rede;
- *pelas portas*: tentativas de conexão feitas, a partir de uma mesma origem, em mais de uma porta de um mesmo host podem indicar uma varredura de sistema, cujo intuito é examinar todos os serviços que o host disponibliza.

2.8 Escalabilidade de IDS

Os desenvolvedores dos programas para identificação de intrusão, na sua maioria, têm o objetivo comum de detectar o maior número possível de eventos perigosos. Porém este objetivo, hoje, é dificultado pelos empecilhos físicos que não permitem o funcionamento dos programas conforme projetados. Os NIDSs, que se baseiam na captura do tráfego de rede, por exemplo, quando em funcionamento em uma rede de alto volume de tráfego e alta capacidade de transferência, são susceptíveis à perda de pacotes pela impossibilidade física de uma

interface de rede, operando em modo promíscuo, capturar todo o tráfego de um barramento. Os IDSs baseados em host, de maneira análoga, também podem ter seu funcionamento comprometido pela obrigatoriedade de processar rapidamente extensos arquivos de registros (logs), armazenados e gerenciados pelo host onde este IDS é executado.

Soluções para o problema de escalabilidade já foram propostas: em algumas, os sensores são distribuídos em segmentos da rede e analisam somente o tráfego no seu segmento, repassando resultados para um elemento central; em outras, agentes são espalhados em pontos estratégicos da rede, coletam dados relevantes e os repassam para analisadores e correlacionadores de dados, que darão o veredicto final da análise.

Outra solução pesquisada não visa propriamente resolver o problema da escalabilidade, mas o de correlacionamento. Nesta, se aplica a máxima "dividir para conquistar" – em [Bezerra de Mello 2004], os autores propõem um sistema de detecção de intrusão que divide o tráfego total em "porções", para que cada uma seja analisada por um sensor diferente. O elemento que faz esta divisão é chamado de *espalhador* (figura 2.3) e sua função é a de particionar o tráfego de acordo com as sessões TCP detectadas. O *espalhador* escolhe o sensor para o qual direcionará o tráfego usando os mecanismos de distribuição circular (*round-robin*). O sistema então diminui a carga de operações de correlacionamento, distribuindo estas tarefas pelos correlacionadores locais.

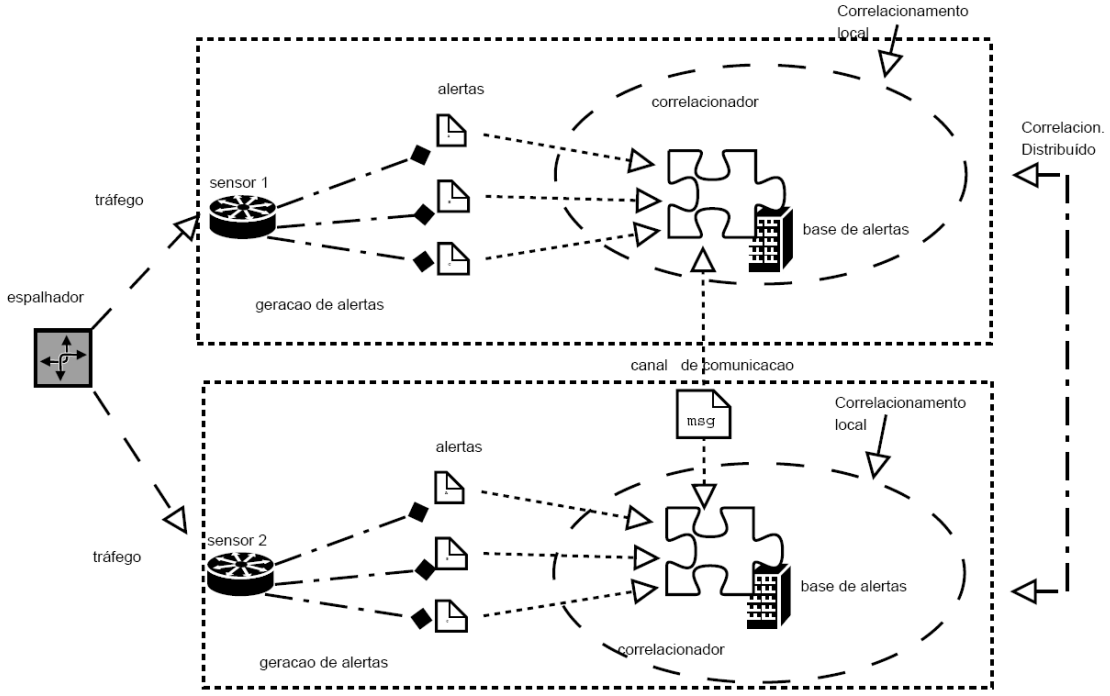

Figura 2.6 - Espalhador de Tráfego [Bezerra de Mello 2004]

2.9 Conclusão

Apesar de todos os esforços em desenvolver sistemas confiáveis, os sub-produtos da Tecnologia da Informação, por serem baseados em dispositivos dependentes de programação, são susceptíveis à apresentação de falhas seu funcionamento. Os programas, por mais que sejam testados e avaliados, sempre apresentam uma ou outra falha, seja de projeto, seja de implementação ou de operação. São estas falhas que admitem a invasão de um sistema com o intuito de furto de informações ou para, simplesmente, efetuar atos de vandalismo.

As invasões, por poderem atuar em vários níveis de um mesmo sistema, desde o nível físico até o da aplicação, podem ser atos extremamente sofisticados e evasivos. Hoje, dentre os grandes desafios dos IDSs, estão os problemas da detecção correta dos ataques, ou seja, como classificar corretamente todas as ações sobre um sistema, sem correr o risco de falsas acusações. Por outro lado, também tem-se o desafio de evitar que ataques passem despercebidos quando os mesmos estão disfarçados de operações “normais” .

O ideal na área de detecção de intrusão seria o de construir IDSs robustos o suficiente que pudessem reagir a ataques maciços e não sucumbir em ambientes com grande volume de tráfego. Em suma, o desafio maior de um projeto de IDSs é o de ser portável, eficiente e escalável o suficiente para operar em ambientes “pesados” e evitar, o mais que puder, a disseminação de alarmes falsos. Na falta de projetos ideais de um IDS que disponha de todas estas características, pode-se optar, ainda, pela composição de detectores de intrusão, assunto que será tratado no Capítulo 3.

Capítulo 3

Composição de Detectores de Intrusão

Observa-se, na prática, que o uso de IDSs individuais e independentes em um sistema mais amplo pode dificultar a identificação de eventos que se configurem como ataques pelo excesso de informação que os detectores podem gerar. Este capítulo mostra alguns aspectos importantes no funcionamento de mais de um IDS em um mesmo ambiente, analisando o mesmo fluxo de dados. Inicia-se com o conceito de IDSs distribuídos (dIDS), a seguir são apresentados os mecanismos de integração de alertas, os problemas de correlacionamento de alertas, a importância da diversidade de projetos e, finalmente, algumas propostas para IDSs compostos (CIDS).

3.1 IDS Distribuídos

Um *Sistema de Detecção de Intrusão Distribuído* (dIDS) é um IDS cujos componentes estão distribuídos em uma ou diversas redes [Fyodor 2000]. Em um dIDS, *sensores* operam em pontos estratégicos da rede, coletando informações e repassando-as para um elemento central de monitoramento e decisão, o *CAS* (*Central Analysis Server*), como ilustrado na figura 3.1.

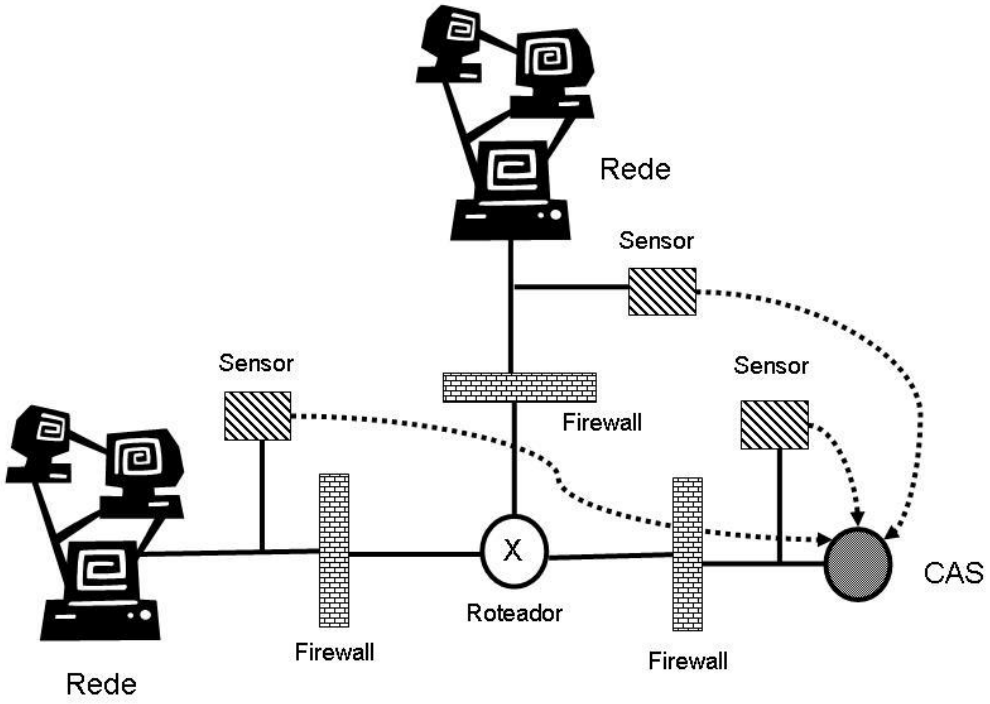

Figura 3.1 - IDS Distribuído

Estes sensores podem ser HIDS, NIDS, ou agentes instalados em equipamentos ativos, como roteadores ou firewalls, desde que providos de suporte à comunicação com o CAS. Sensores podem operar de maneira *ativa* ou *passiva*: na maneira ativa, o sensor é capaz de analisar os dados coletados e optar, dependendo da sua configuração, em repassar ao CAS somente as informações relevantes como, por exemplo, ataques ou anomalias detectadas. Os sensores passivos não processam os dados coletados e os repassam integralmente ao CAS (em forma de *raw data*), deixando para o CAS a responsabilidade de contra-atacar ou de alertar os responsáveis pelo sistema a respeito do ataque.

Ao monitorar vários pontos de uma rede, em especial pontos geograficamente dispersos, o CAS é capaz de consolidar informações que, individualmente, podem não configurar um ataque propriamente dito mas, em conjunto, evidenciam um ataque ou pré-ataque. Por exemplo, ao consolidar informações sobre varreduras de portas detectadas em

diversos sensores de uma rede corporativa, o CAS poderia identificar um ataque eminente a certos serviços desta rede.

Como outra vantagem, temos a possibilidade de disseminação globalizada das informações de sensores, através da transmissão de informações por *agentes* instalados em outros IDSs para um CAS central. Esta possibilidade permitiu a criação de vários serviços, voluntários ou comerciais, de integração de alertas e divulgação de alarmes para ataques conhecidos ou desconhecidos. Serviços cooperativos como o *Internet Storm Center* e *Dshield* (SANS Institute), e o *MyNetWatchman* (mynetwatchman.com), hoje agregam informações enviadas por agentes instalados em ambientes diversos, desde redes corporativas, passando por ISPs, até usuários comuns da internet. Baseados nestas informações, estes dIDS são capazes de manter um mapa global das ameaças, correlacionando e consolidando informações e, ainda, alertando os administradores das redes que colaboram com o serviço através de emails de alerta e de estatísticas e informações atualizadas dos ataques analisados [Robbins 2002].

Contudo, diversos problemas podem ser enumerados nesta abordagem. Em primeiro lugar, os IDSs distribuídos não são totalmente distribuídos uma vez que necessitam de um elemento centralizador, o CAS, que pode ser considerado um ponto de falha. Em segundo lugar, os dIDS que contam com sensores passivos, cujas informações são repassadas ao CAS em aberto (*raw data*), podem ter os dados alterados por invasores a fim de encobrirem seus ataques; mecanismos de criptografia e confiança entre os sensores e o CAS poderiam garantir a integridade das informações, mas acrescentariam uma carga adicional de processamento ao CAS, por si só um mecanismo que precisa processar e decidir rapidamente. Outro ponto fraco é o risco de os próprios dIDS sofrerem ataques DoS se, por exemplo, um atacante gerar uma enorme quantidade de ataques aos sensores, aumentando a quantidade de tráfego transmitido entre os sensores e o CAS e, finalmente, sobrecarregando o CAS a ponto de interromper suas atividades.

Em [Gosh 2004], é apresentado um dIDS baseado em agentes distribuídos, o ABDIAS, cuja arquitetura se propõe a criar um dIDS totalmente descentralizado. O ABDIAS utiliza agentes autônomos, eliminando assim a figura centralizadora do CAS. Todos os agentes trabalham de forma pró-ativa e cooperativa, coletando e analisando os dados. Nesta arquitetura, os agentes são organizados em grupos autônomos (*neighborhoods*) dentro dos quais os agentes se comunicam, só existindo comunicação entre os grupos quando há

problemas de consenso dentro de um grupo. A base de conhecimento dos agentes é baseada no modelo de Bayes, que possibilita a inferência da possibilidade de ocorrência de certos tipos de invasões.

3.2 Integração de Alertas

Atualmente, com a variedade de produtos para detecção de intrusão oferecidos comercialmente ou em plataforma aberta, cada um eficiente na sua especialidade, tem-se uma tendência favorável à instalação de diversos IDSs nas redes onde um cuidado maior na prevenção de ataques é necessário. Já é prática comum a manutenção de sistemas híbridos, compostos por um número de sistemas de detecção, cada um gerando seus logs e relatórios próprios, disponibilizando grandes quantidades de informações, às vezes não tão úteis ou de difícil interpretação.

Acompanhando esta tendência, os administradores se vêem afogados em relatórios dispersos, e dependentes de sua capacidade de organização e observação para agrupar e correlacionar eventos, eliminar alarmes falsos e, finalmente, chegar à conclusão de que realmente *havia* uma brecha e, de fato, ocorreu um ataque bem sucedido. A centralização das informações em um elemento capaz de organizar e correlacionar eventos é o enfoque para resolução deste problema.

Para ajudar nesta tarefa de integração de alertas, pesquisadores vêm desenvolvendo protocolos e ferramentas de integração que sejam capazes de analisar e consolidar os relatórios dos diversos IDSs. Entre eles, temos o SCYLLARUS [Goldman 2001] e a arquitetura baseada em ACCs (*Aggregation and Correlation Component*) [Debar 2001], detalhados a seguir. Nestes dois, a passagem de dados entre os coletores de dados e os analisadores é feita através de implementação própria – o Scyllarus utiliza APIs e o ACC utiliza uma interface chamada *pre-adapter*. Com o intuito de reduzir a necessidade de APIs e interfaces próprias, o IETF patrocina um grupo de pesquisa nesta área, o *Intrusion Detection Working Group* (IDWG), que vem trabalhando nas especificações que vão permitir o transporte de informações entre quaisquer dispositivos de detecção e as unidades de consolidação, a especificação IDMEF/IDXP [Zaraksa 2003].

3.2.1 Scyllarus

É uma arquitetura de integração de IDSs individuais, com o intuito de gerar uma base de informações organizada de acordo com um modelo de segurança bem definido. Neste modelo, os eventos presentes nos relatórios enviados pelos IDSs (chamados de *sensores*) são avaliados, agregados, correlacionados e fundidos através de um processo decisório. Este processo considera os eventos como hipóteses, que podem ser consideradas plausíveis ou não, dependendo da avaliação correlacionada com outros eventos. Hipóteses plausíveis são comparadas com as políticas de segurança pré-definidas, para então concluir se houve ou não um comprometimento dos objetivos de segurança. Com o Scyllarus é possível, por exemplo, determinar que uma varredura de uma certa porta, capturada por um sensor, quando comparada a outros eventos relacionados, como a seqüência de endereços envolvidos e as ocasiões e freqüência em que esta varredura ocorre, nada mais é do que um evento normal na rede, por exemplo, gerado pelo sistema central de atualização dos agentes anti-virus das estações de uma rede. O que seria um alerta de *portscan* é explicado como um evento normal, que não fere os objetivos de segurança.

3.2.2 ACCs

A arquitetura de ACCs (*Aggregation and Correlation Component*) foi desenvolvida visando implementar uma console de gerenciamento de detecção de intrusões em conjunto com o *Tivoli Enterprise Console* (TEC) [Tivoli 2000]. Nesta arquitetura, dois elementos básicos são definidos: os *probes* (IDSs) e os ACCs. A função dos ACCs é a de correlacionar os relatórios dos *probes* e oferecer ao administrador uma visão condensada dos eventos relacionados à segurança. A possibilidade de implementar ACCs hierarquicamente resolve alguns problemas de escalabilidade. Os ACCs se utilizam de um algoritmo de agregação e correlação cujo objetivo é o de formar grupos de alertas e mostrar, ao operador, eventos relacionados ao invés dos eventos isolados. Regras e critérios definem, nos ACCs, as "situações" – um conjunto de alertas que têm características comuns, por exemplo, eventos duplicados ou gerados como conseqüências de outros. Baseando-se nas situações determinadas, a console de gerenciamento somente dispara alarmes quando uma certa "situação" é deflagrada.

3.2.3 IDMEF/IDXP

A intenção do grupo de trabalho IDWG é a de desenvolver um *standard* em protocolos e formatos que permitam padronização dos dados gerados pelos programas de detecção de intrusão e seu transporte para unidades genéricas de monitoramento e consolidação. Os protocolos desenvolvidos pelo grupo visam atender às seguintes demandas:

- transporte dados entre os NIDS e as estações de monitoramento;
- desenvolvimento de uma base de dados padronizada;
- desenvolvimento de ferramentas para correlacionamento entre produtos diferentes;
- desenvolvimento de uma linguagem comum para os grupos de discussão.

A implementação comercial desta padronização ainda é incipiente. Até fevereiro de 2005, havia disponível somente um *plug-in* para o Snort v2.0, atualizado em 26 de dezembro de 2004. Os fabricantes do Prelude, por exemplo, alegam que o uso do XML gera *overhead* e implementam o IDMEF segundo orientação própria, com estruturas em linguagem C [Zaraksa 2003].

Em março de 2003, a especificação IDMEF incluía as seguintes padronizações:

- **IDMEF**: *Intrusion Detection Messagem Exchange Format*. Define a estrutura lógica da mensagem. Baseia-se no conceito de classes e sub-classes. Define as classes Alert e Heartbeat, que se subdividem em sub-classes, incluindo, entre outras, os dados do evento registrado (ataque), como hosts ou serviços de origem e destino (Source, Target), horário da ocorrência (DetectTime) e classificação com relação a ataques conhecidos ou não (Classification). A estrutura completa é exibida na figura 3.2.
- **IDXP**: *Intrusion Detection Exchange Protocol*. É a especificação para a implementação do protocolo de transporte, o BEEP. Os *profiles* do IDXP servem para especificar os parâmetros de criação do canal de comunicação para a transferência dos dados IDMEF e incluem o suporte para autenticação entre origem e destino, confidencialidade, integridade, proteção contra DoS e contra duplicação de mensagens.

- **BEEP**: *Blocks Extensible Exchange Protocol*. Protocolo para transferência de dados, orientado à conexão, que opera sobre o TCP. É responsável por estabelecer uma sessão entre os hosts TCP que pretendem trocar informações. Suporta até 257 canais em um única sessão, cada um estabelecendo seu próprio *profile*, de modo que cada conexão TCP possa ser utilizada para transferência de diversos tipos de dados. As mensagens são do tipo MIME e estruturadas em XML.

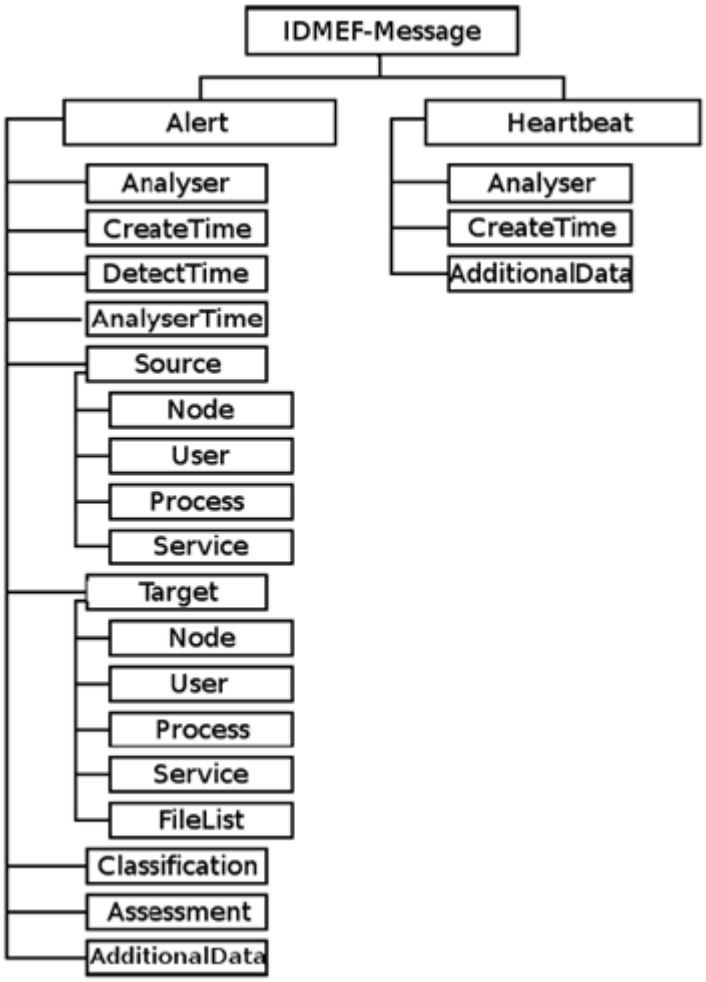

Figura 3.2 - Modelo de Classes da Mensagem IDMEF [Zaraska 2003]

3.3 Diversidade de Projetos

Como todos os outros sistemas de informática, os de detecção de invasão estão sujeitos a falhas, seja por erro no projeto, na configuração ou na falta de manutenção, ou ainda por interferência externa. Nos sistemas distribuídos, a forma mais natural de se contornar o

problema de interrupção dos serviços por falha é pela implementação das técnicas de tolerância a faltas, replicando-se os hosts de serviço e implementando-se técnicas de detecção de falhas e redistribuição de serviços [Avizienis 2000].

As técnicas de diversidade de projetos, apesar de terem alguma semelhança com as de tolerância a faltas, visam um objetivo diferente. Na diversidade de projetos, a idéia principal é a de se ter vários hosts de serviços com implementações diferentes de uma mesma especificação de programa, operando paralelamente, não para redistribuição de carga em caso de falha, mas para comparações de resultados. Um algoritmo eficiente de decisão deve ser capaz de analisar os resultados de cada implementação e de decidir pelo mais confiável.

Como exemplo, em [Avizienis 1992], são apresentadas as especificações do modelo NVS (*N-Version Software*) para diversidade de projetos. Neste modelo, os programas que operam em paralelo devem ter equivalência funcional, mas serem totalmente independentes, tanto no desenvolvimento quanto na manutenção. A intenção é a de minimizar a probabilidade de se obterem erros iguais em todas as implementações, responsabilizando o algoritmo de decisão pelo descarte de resultados divergentes e apresentação de um único resultado.

3.4 Diversidade de Projetos em Detecção de Intrusão

A questão da aplicação da diversidade de projetos em IDSs é discutida em [LittleWood 2004]. Nesta discussão, é proposta a aplicação dos conceitos de redundância e de diversidade de projetos no âmbito da segurança (entendida como o trio *safety*, *security*, *reliability*), mas não de forma livre e desordenada: é necessário avaliar, talvez com modelos probabilísticos, a melhor maneira de compor um ambiente redundante com versões independentes de sistemas. Segundo [LittleWood 2004], um projeto de redundância é intrinsicamente relacionado a um projeto de diversidade de projetos, pois não se atinge o objetivo da redundância se houver a possibilidade de as falhas de um sistema serem replicadas ao replicar-se o sistema. No que toca os IDSs especificamente, esta discussão deixou algumas questões pendentes como uma metodologia eficiente para avaliação do desempenho de um IDS, outra para a comparação de desempenhos entre IDSs diferentes, e uma terceira para decidir a melhor maneira de combinar IDSs diferentes. Outra questão pendente é a da metodologia para a avaliação do desempenho de um IDS obtido pela combinação de IDSs diferentes. A afirmação mais importante em [LittleWood 2004], do ponto de vista do trabalho

apresentado aqui, é que eles apontam a necessidade de uma abordagem matemática para resolver estas questões e sugerem a criação de uma nova metodologia que possibilite a avaliação de IDSs compostos e heterogêneos.

Como conseqüência direta do trabalho de [LittleWood 2004], [Maxion 2005] apresenta provas de que a diversidade de projetos na detecção de intrusão deve ser uma forte tendência a ser seguida, pelo menos com os detectores baseados em anomalias. Nesse trabalho, os autores provam que, por haver diferenças nos algoritmos de detecção dos detectores baseados em anomalias, a capacidade de detecção de ataques varia bastante de um IDS para o outro havendo, inclusive, detectores totalmente "cegos" para algumas sequências anômalas utilizadas nos testes controlados. Com os resultados conseguidos nesse estudo, é possível afirmar que é infundado considerar que todos os detectores baseados em anomalias têm igual capacidade de detecção, mesmo quando calibrados de maneira equivalente e analisando o mesmo fluxo de dados.

Conclui-se, portanto, que:

- um IDS único pode não ser eficaz para a segurança de um sistema;
- uma composição de IDSs deve seguir as premissas de redundância e diversidade de projetos;
- para analisar o desempenho de um IDS-composto (CIDS), deve-se desenvolver uma abordagem matemática para sua modelagem e uma metodologia eficaz para a análise do seu desempenho.

Na prática, já existem algumas propostas de diversidade de projetos para IDSs. Em [Reynolds 2003] é apresentado o sistema de detecção de invasão HACQIT, que implementa o conceito de diversidade de projetos para identificar e registrar novos ataques e prover redundância de serviços. No HACQIT (figura 3.3), o sistema de detecção é implementado em um *cluster* de máquinas que incluem duas ou mais implementações do mesmo serviço. No caso específico do artigo citado, os autores usam produtos COTS (*commercial-off-the-shelf*), como os servidores web IIS, da Microsoft, e Apache. Os serviços são implementados em duas máquinas distintas, uma delas é eleita a máquina primária e, a outra, definida como secundária. Além destes dois hosts, é implementado um terceiro, o MAC (Mediator/Adapter/ Controller), responsável em interceptar acessos feitos ao primário e verificar, em sua base de

permissões, se este acesso deve ser bloqueado ou não. A base de permissões contém os registros dos ataques já caracterizados, sendo atualizada dinamicamente. Acessos liberados pelo MAC são replicados no host secundário e as respostas de ambos servidores são comparadas. Havendo discrepâncias entre as respostas, o MAC considera que houve uma falha. Dependendo da extensão da falha, o servidor primário pode ser considerado como atacado, colocado em indisponibilidade, sendo o secundário elevado ao posto de primário. O ataque é então registrado na base de ataques do MAC.

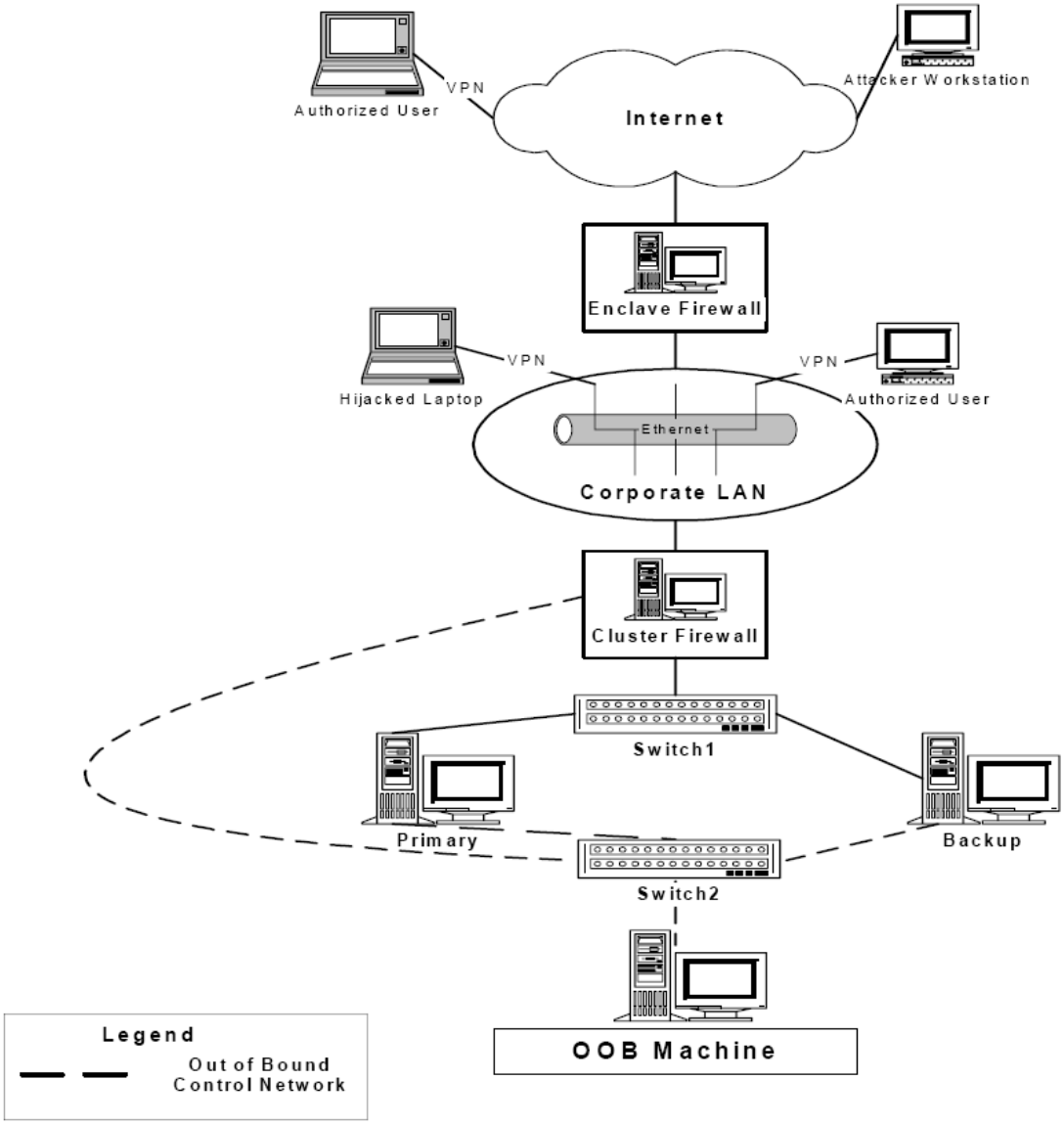

Figura 3.3 – Arquitetura do sistema HACQIT [Reynolds 2003]

3.5 Composição de IDSs

A composição de IDSs é um recurso que visa, principalmente, aumentar a precisão dos resultados obtidos por IDSs originalmente projetados para operar de forma independente. As maneiras com que estes detectores ou seus resultados são combinados pode variar de projeto para projeto, mas o objetivo final é sempre o mesmo: aproveitar o que cada IDS tem de melhor e desprezar o que cada IDS tem de pior. Em última essência, o objetivo de um IDS Composto (CIDS) é o de diminuir a quantidade de alarmes falsos que os IDSs individuais geram. Um CIDS bem projetado deve, idealmente, ser mais preciso e seguro do que cada um dos IDSs individuais que o compõem.

Muitas são as propostas de CIDS. Por exemplo, [Kahn 1998] apresenta um paradigma para a cooperação e interoperabilidade de detectores, denominado *Common Intrusion Detection Framework* (CIDF). O projeto CIDF prega a padronização de formatos, protocolos e arquiteturas para possibilitar a troca de informações relevantes entre os IDSs. A filosofia básica deste projeto sugere que dois sistemas de detecção independentes operam colaborativamente quando são capazes de trocar informações automaticamente entre eles e, assim, atingir resultados globais que nenhum dos dois sistemas atingiria operando sozinho.

A operação colaborativa, segundo [Kahn 1998] pode ser dividida em, pelo menos, 4 aspectos: *análise*, *consolidação*, *endosso* e *confirmação*. Para ilustrar estes aspectos, imagine-se dois IDS heterogêneos, IDS_A e IDS_B, operando de maneira independente, mas analisando o mesmo fluxo de dados:

- No quesito *análise*, o IDS_A seria somente um coletor de dados brutos e o IDS_B analisaria mais detidamente os dados coletados pelo IDS_A (Figura 3.4);

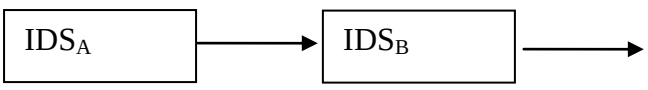

Figura 3.4 - Análise

- No quesito *consolidação*, um terceiro componente, M, mesclaria e consolidaria os dados gerados pelos IDS_A e IDS_B, gerando um conjunto mais consistente de resultados (Figura 3.5);

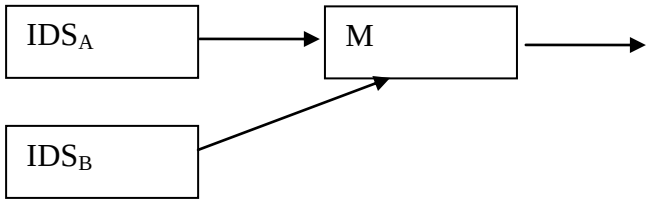

Figura 3.5 - Consolidação

- No quesito *endosso*, um terceiro elemento, J, analisaria e julgaria os dados gerados pelos IDS_A e IDS_B, gerando um conjunto de resultados com maior grau de certeza (basicamente, os resultados em comum dos dois IDSs) (Figura 3.6);

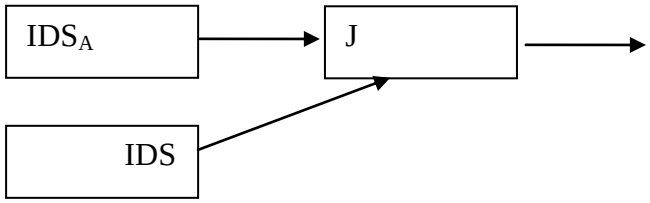

Figura 3.6 - Endosso

- no quesito *confirmação*, o IDS_A, ao identificar um evento, solicitaria a um terceiro elemento, J, a possibilidade de este evento ter sido também detectado pelo IDS_B. Se o IDS_B confirmar a identificação, então o evento é notificado (Figura 3.7).

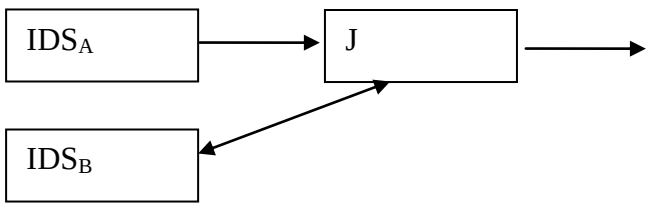

Figura 3.7 - Confirmação

Outra proposta de CIDS é apresentada em [Jianhua 2003], na qual combinam-se não duas arquiteturas diferentes, mas dois *estados* de funcionamento de um sistema: o estado normal e o estado invadido. Neste caso, o "C" do CIDS é dado pela composição de dois estados diferentes do sistema, cada estado denotado por uma base de dados característica. O estado normal é obtido durante o funcionamento de um certo sistema em ambiente controlado, sem a possibilidade de invasões. Todas as chamadas de sistema obtidas neste estado (*normal system call sequences*) são coletadas inseridas em uma base de dados denominada NSCS. Em seguida, o sistema é exposto a um ambiente hostil e as chamadas coletadas neste ambiente, cujas seqüências não estejam na base NSCS e cujas probabilidades de se configurarem como ataques sejam altas (acima de um limite conhecido), são inseridas na base de invasões, a ISCS (*intruder system call sequences*). Com o sistema em funcionamento, o CIDS é acionado e compara cada acesso ao sistema, pelas seqüências de chamadas, com aquelas armazenadas na ISCS e/ou na NSCS, determinando se o acesso ao sistema é legítimo ou não. Os experimentos feitos em [Jianhua 2003] comprovaram que, ao longo do tempo, as bases de acessos normais e de invasões crescem e, consequentemente, o número de alarmes falsos gerados pelo CIDS tende a diminuir.

A proposta de modelo de CIDS apresentada em [Yu-Sung 2003] é semelhante ao modelo proposto em [Kahn 1998] e implementa a colaboração de "detectores elementares" (EDs), cujas orientações são específicas e independentes, ou seja, cada ED é um especialista em um nível específico do sistema. Além disso, os EDs não precisam operar necessariamente no mesmo host. Em particular, o CIDS apresentado por [Yu-Sung 2003] é composto por três EDs: o primeiro opera no nível de rede (Snort), o segundo opera no nível de aplicação (Libsafe) e o terceiro opera no nível de *kernel* do sistema operacional (Sysmon). Os EDs podem monitorar componentes diferentes do sistema e devem comunicar-se com um "gerente" através de filas de mensagens. A função do gerente é a de correlacionar os eventos coletados e calcular a probabilidade dos resultados se configurarem como um ataque. O gerente, por sua vez, é composto por 4 módulos (Figura 3.8), responsáveis pelo correlacionamento dos eventos coletados e análise da probabilidade dos resultados se configurarem como um ataque. Os experimentos de [Yu-Sung 2003] mostraram que este CIDS opera de forma satisfatória, não degradando o sistema como um todo e que a

quantidade de alarmes falsos diminui significativamente se comparada com o funcionamento de cada ED separadamente.

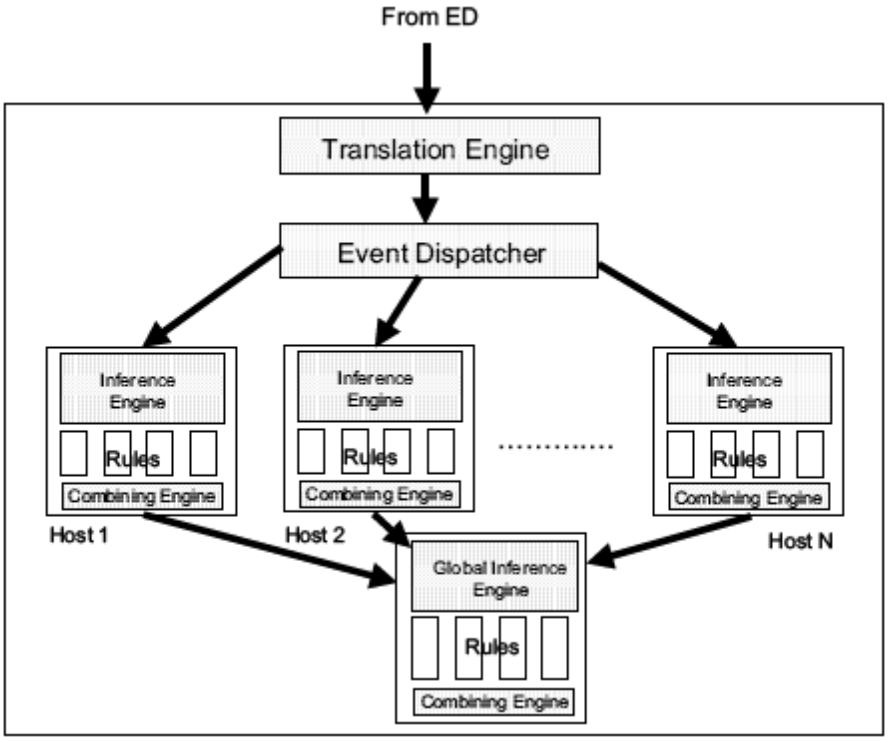

Figura 3.8 – Arquitetura do gerente [Yu-Sung 2003]

3.6 Conclusão

Usar somente um IDS em ambientes mais amplos pode impossibilitar o rastreio e a identificação dos ataques, levando a um problema de classificação exata das informações, o que pode acarretar o desprezo de informações importantes ou o desperdício de tempo na análise de informações irrelevantes. Porém, ao somente se replicar o mesmo IDS em vários pontos de um sistema corre-se o risco gerar uma imensa quantidade de eventos a serem analisados e uma quantidade desagradável de alarme falsos e informações duplicadas. Para contornar este problema, são desenvolvidas técnicas de distribuição de IDSs (dIDS), associadas às técnicas de correlacionamento de alertas e mecanismos de integração de informações entre os detectores. Além disso, as técnicas de diversidade de projetos, quando aplicadas à detecção de invasão, permitem criar sistemas mais abrangentes, robustos e eficientes, como os CIDS – um IDS "maior", composto por detectores heterogêneos, cujas características incluem a intenção de obter sempre o melhor desempenho de cada um dos IDSs individuais.

Capítulo 4

Um Modelo Matemático para a Composição de Detectores de Intrusão

A proposta deste capítulo é a de apresentar um modelo matemático que possibilite a combinação dos resultados de um conjunto de detectores de intrusão individuais heterogêneos. A intenção deste modelo é a de mapear e tratar os resultados dos detectores individuais como também do detector composto (*CIDS – Compound* ou *Composite Intrusion Detection System*) tornando possível a comparação do resultado consolidado com os resultados individuais.

A princípio, o intuito da modelagem era o de apresentar um modelo visual que permitisse exibir todas as possibilidades de resultados em um ambiente no qual um IDS estivesse operando e, depois, estendê-lo para *n* IDSs operando em conjunto. A opção mais natural para este modelo surgiu com a Teoria dos Conjuntos Tradicional já que os elementos envolvidos poderiam ser representados por objetos, conjuntos e relações de pertinência entre eles (Seção 4.1).

Ao longo do trabalho, com a intenção de demonstrar que o CIDS apresenta resultados mais precisos do que qualquer um dos detectores individuais que o compõem, percebeu-se que seria necessário, de alguma maneira, quantificar alguns resultados, ou seja, obter números que pudessem retratar algo mais do que a simples relação de pertinência (“pertence” ou “não pertence”), e que trouxessem valores passíveis de comparação. A solução surgiu com a Teoria de Conjuntos Difusos (Fuzzy, ou Teoria Nebulosa), através da qual é possível atribuir valores

numéricos à relação entre um objeto e um conjunto usando-se uma equação matemática chamada de *função de grau de pertinênci*a.

Alguns trabalhos semelhantes, mas com intenções diferenciadas, foram encontrados durante a pesquisa do modelo aqui apresentado; tais trabalhos são apresentados na Seção 4.3.

4.1 Modelagem usando Teoria dos Conjuntos

A abordagem matemática seguida neste capítulo baseia-se, a princípio, na Teoria dos Conjuntos Tradicional. A intenção desta modelagem é a de tornar possível apontar qualitativamente cada subconjunto de objetos envolvidos no funcionamento de um IDS. Os conjuntos aqui definidos foram selecionados dentre as possibilidades de conjuntos inerentes ao funcionamento de um IDS; seus relacionamentos serão representados nos diagramas a seguir. Uma única abordagem semelhante foi proposta em [Leckie 2002], mas essa se restringe a retratar somente um detector, enquanto a modelagem aqui proposta aqui admite extensão para *n* detectores.

- **Conjunto Universo (U)** – conjunto que compreende todos os eventos possíveis de serem encontrados em qualquer sistema. A princípio, todo tipo de acesso ou operação em um sistema informatizado faz parte do conjunto U.
- **Conjunto de Eventos Normais (N)** – conjunto que compreende os eventos esperados, pressupostos, aceitos e para os quais um sistema está preparado para responder.
- **Conjunto de Eventos Direcionados ao Sistema (T) –** compreende todos os eventos direcionados a um sistema, dentre eles os acessos normais (esperados pelo sistema), os anômalos (classificados como ataques pelos detectores) e os não detectados pelos detectores.
- **Conjunto de Eventos Detectados por um Detector (D) –** compreende todos os eventos detectados por um detector ativado em um sistema.

A primeira modelagem, ilustrada na Figura 4.1, mostra um sistema munido de somente um detector. Neste diagrama, todos os conjuntos propostos serão identificados pelas letras:

- U – o conjunto de todos os eventos possíveis no sistema;
- N – o conjunto de eventos normais, ou seja, aqueles esperados pelo sistema;
- T – o conjunto de eventos direcionados ao sistema;
- D_1 – o conjunto de eventos detectados pelo detector IDS_1 .

Também são identificados, neste diagrama, alguns subconjuntos de interesse. No diagrama da Figura 4.1, é possível identificar:

- em (1), o subconjunto de D_1 que contém os eventos normais observados pelo detector IDS_1 ($D_1 \cap N$);
- em (2), o subconjunto de D_1 que contém os ataques detectados pelo detector IDS_1 ($D_1 – N$);
- em (3), o subconjunto de T que contém os eventos direcionados ao sistema que não foram percebidos pelo detector IDS_1 (T – D1); neste conjunto estão incluídos os falsos negativos (T – D1)-N.

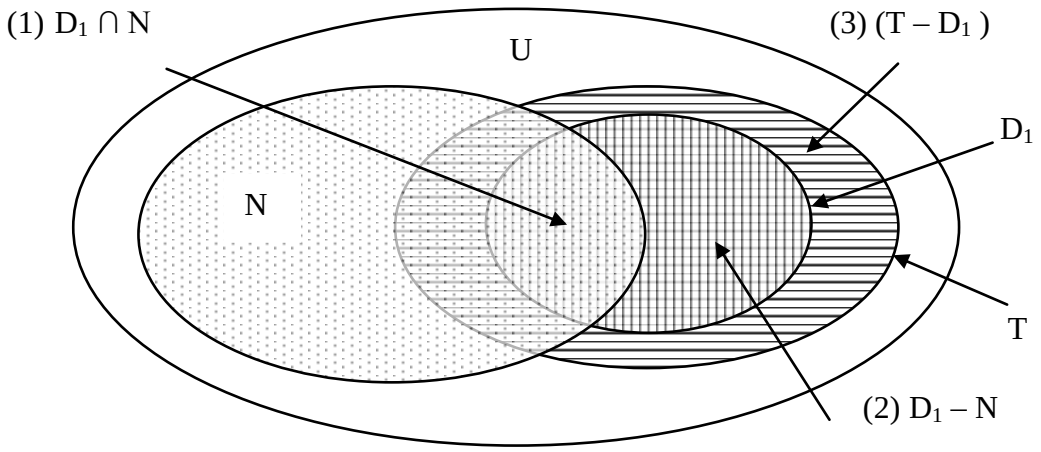

Figura 4.1 – Diagrama de Venn para Modelagem com um Detector

Com o intuito de facilitar as representações gráficas, o modelo será estendido para somente 2 detectores, IDS_1 e IDS_2, como está ilustrado na Figura 4.2. Neste diagrama, o conjunto de eventos detectados pelo detector IDS_1 será identificado pelo conjunto D_1, enquanto que o conjunto de eventos detectados pelo detector IDS_2 será identificado pelo conjunto D_2.

O conjunto final da detecção do CIDS pode ser obtido de duas maneiras: na primeira, mais restritiva e obtida pela intersecção dos conjuntos de resultados, são considerados os ataques comuns, detectados por ambos IDSs; na segunda, mais abrangente e obtida pela união dos conjuntos de resultados, são considerados todos os ataques detectados em cada um dos IDSs.

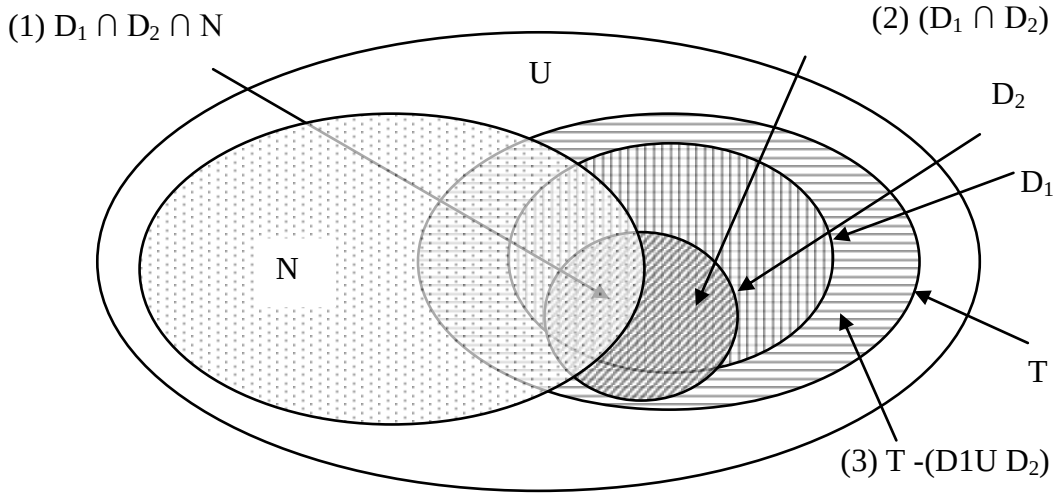

Figura 4.2 - Modelagem com Dois Detectores

A primeira interpretação é chamada de "composição AND"; nela, o conjunto de resultados para CIDS será obtido a partir da interseção dos conjuntos de resultados dos IDSs individuais:

- em (1), os eventos normais observados por ambos ($D_1 \cap D_2 \cap N$);
- em (2), os ataques detectados por ambos (($D_1 \cap D_2$) – N);
- em (3), os ataques não detectados por nenhum dos IDSs (T - (D1 ∪ D2))

Esta abordagem é mais restritiva pois considera como ataques somente os eventos detectados por ambos IDSs. Por um lado, visa diminuir os falsos positivos; por outro lado, corre o risco de descartar eventos importantes que porventura tenham sido detectados por um IDS com maior capacidade de detecção de alguns ataques em particular.

A extensão para *n* detectores pode ser modelada da mesma forma: os eventos normais detectados pelo CIDS serão representado por $(\bigcap_{i=1}^{n} D_i) \cap N$ e os ataques detectados pelo CIDS serão representado por $(\bigcap_{i=1}^{n} D_i)$ - N.

Ao optar pela interseção de todos os conjuntos de detecção, o conjunto de resultados do CIDS $(\bigcap_{i=1}^{n} D_i)$ pode ser "menor" do que alguns dos conjuntos de resultados individuais (D_1, D_2, ..., D_n), já que os ataques que não tenham sido detectados por todos os IDSs serão desconsiderados. Para compensar este fato, já que entende-se que os ataques detectados individualmente não devem ser totalmente ignorados, será apresentada, na próxima seção, uma metodologia que permitirá atribuir um *grau de importância* a cada um dos eventos classificados como ataque por cada um dos componentes do CIDS. Desta maneira, todos os ataques detectados, mesmo aqueles que não estejam presentes em todos os conjuntos de detecção, terão um certo nível de importância na composição do resultado final.

A segunda interpretação é chamada de "composição OR"; nela, o conjunto de resultados do CIDS será obtido pela união dos conjuntos de resultados dos IDSs individuais. No caso do diagrama que representa os dois detectores, o conjunto final de eventos considerados como ataques será $((D_1 \cup D_2) - N)$; para um modelo com *n* detectores, o conjunto final será dado pela união de todos os conjuntos de ataques detectados $(\bigcup_{i=1}^{n} D_i - N)$. Esta abordagem é mais abrangente pois considera os pontos fortes de todos os IDSs da composição, entretanto pode gerar um número maior de falsos positivos.

4.2 Atribuição do Grau de Importância

Como mencionado na seção anterior, a primeira interpretação do modelo para CIDS proposto neste trabalho pode descaracterizar ataques importantes pelo fato de estes ataques não terem sido percebidos por todos os detectores do CIDS; já a segunda interpretação considera uma quantidade maior de eventos, mas aumenta o grau de incerteza do sistema. Portanto, o modelo será adaptado para comportar um pouco de cada uma das considerações acima: da primeira, será aproveitada a maior precisão, da segunda, será aproveitada a maior abrangência. Para tanto, serão acrescentados ao modelo alguns conceitos da Teoria de

Conjuntos Difusos, uma generalização da Teoria dos Conjuntos. Com a "fuzzificação do modelo", ou seja, a transposição do modelo da teoria tradicional para a teoria difusa, as classificações individuais dos detectores serão reconsideradas sob o ponto de vista do entendimento coletivo (do conjunto de detectores) e os todos os ataques detectados terão algum grau de validade no resultado final.

Para evitar o descarte de eventos classificados como ataques por somente alguns dos detectores, será introduzido o conceito de *grau de importância* do evento, uma medida que servirá para graduar a importância de um ataque de acordo com o número de IDSs que o detectam; eventos detectados por um número maior de IDSs terão um grau de importância maior no conjunto de detecção do CIDS, e vice-versa.

Para atribuir-se um valor numérico ao grau de importância de um ataque, será definida uma função denominada *função de presença,* $f_k(\boldsymbol{e})$. Esta função serve para identificar se um evento ***e*** está ou não presente no conjunto D_k de ataques detectados pelo detector IDS_k pertencente a um CIDS com *n* detectores. O somatório *S* de todos os valores da função de presença do evento ***e*** no CIDS será utilizado na composição final do grau de importância deste ataque.

Sejam:

(i) D_k o conjunto de ataques detectados pelo detector IDS_k ,

(ii) ***e*** um evento classificado como ataque por algum dos *n* detectores que compõem o CIDS,

então a *função de presença* $f_k(\boldsymbol{e})$ pode ser definida como

$$f_k(e) = \begin{cases} 1, \text{ se } e \in Dk, \\ 0 \text{ se } e \notin Dk \end{cases} \qquad (1)$$

Para produzir-se o conceito final de *importância do ataque* o modelo original será "fuzzificado", ou seja, adaptado à Teoria de Conjuntos difusos. Em resumo, na teoria de conjuntos difusos um elemento pode pertencer totalmente ou parcialmente a um conjunto difuso. Para definir "o quanto" um elemento pertence a um conjunto, deve ser estabelecida uma *função de grau de pertinência*. Esta função atribuirá, ao elemento ***e***, um valor numérico (um número real), entre 0 e 1, 0 indicando total ausência e 1 indicando total presença do

elemento no conjunto difuso. Outra característica da Teoria de Conjuntos Difusos é a de que qualquer conjunto difuso pode ser totalmente definido pela sua função de grau de pertinência. Para então fazer a *fuzzificação* do modelo, devem ser definidos um conjunto difuso e uma função de grau de pertinência inerentes ao modelo original.

O conjunto difuso escolhido será chamado de "Importância" (I) e a função de grau de pertinência para um certo evento ***e*** (a função $\mu I(\boldsymbol{e})$) medirá o quanto o evento ***e*** é importante. Em outras palavras, $\mu I(\boldsymbol{e})$ medirá o grau de importância do evento ***e*** no contexto do CIDS. Para criar-se a função de grau de pertinência $\mu I(\boldsymbol{e})$, algumas condições foram estabelecidas:

i. os parâmetros de construção da função serão: 1) a quantidade de IDSs que compõem o CIDS e 2) o conjunto dos valores possíveis para o grau de importância de um evento (os números reais entre 0 e 1);
ii. deverá ser uma função exponencial crescente pois se deseja que o crescimento seja rápido nos primeiros termos e que seja possível estabilizar seu crescimento nos termos finais (tenha uma "tendência" para o valor máximo 1);
iii. $\mu I(\boldsymbol{e})$ deverá valor mínimo zero, ou seja, deverá ser zero se o evento ***e*** não for detectado por nenhum dos IDSs do CIDS;
iv. os valores produzidos pela função não devem ser absolutos, ou seja, o grau de importância de um evento ***e*** detectado pelo CIDS deve ser diretamente proporcional ao número de IDSs que o detectam e inversamente proporcional ao número total de IDSs que compõem o CIDS.

Em (2) é apresentada a forma genérica de uma função que obedece a (i), (ii) e (iii) estabelecidos acima e que será adaptada à *fuzzificação* do modelo:

$$f(x) = 1 - \left(\frac{1}{1+x} \right) \qquad (2)$$

Na adaptação, x será substituído pelo somatório das funções de presença $f_k(\boldsymbol{e})$ e as proporções direta e inversas citadas acima em (iv) serão aplicadas, produzindo-se a função de grau de pertinência $\mu I(\boldsymbol{e})$ apresentada em (3).

Sendo n o número de detectores que compõem o CIDS e $S = \sum_{k=1}^{n} f_k(e)$, então

$$\mu\, I(\boldsymbol{e}) = \left(1 - \left(\frac{1}{1+S}\right)\right) * \frac{S}{n} \qquad (3)$$

Na prática, esta *fuzzificação* do modelo foi feita visando possibilitar a criação de alguns pontos de decisão no CIDS:

(i) determinar, em um CIDS com n detectores, qual o número mínimo de detectores do CIDS que precisam acusar um ataque para que este ataque seja considerado "relevante";
(ii) estabelecer o número n de detectores um CIDS deve ter para que seja possível validar um resultado obtido por k detectores ($k \leq n$).

Antes de tratar estas questões, é necessário quantificar a importância de cada ataque detectado, o que pode ser feito através da definição de uma *escala de importância*. Propõe-se estabelecer uma tabela de graus de importância de um ataque (Tabela 4.1), cujos valores limites *Li* (limite inferior) e *Ls* (Limite superior) serão estabelecidos matematicamente na seção 5.

Tabela 4.1 – Graus de Importância

$\mu\, I(\boldsymbol{e})$	Grau de Importância
$0 \leq \mu\, I(\boldsymbol{e}) < Li$	não merece atenção
$Li \leq \mu\, I(\boldsymbol{e}) < Ls$	relevante
$\mu\, I(\boldsymbol{e}) \geq Ls$	muito relevante

A título de ilustração, com base na equação definida em (3), elaborou-se a tabela dos graus de importância para eventos detectados em CIDS com até 6 detectores (Tabela 4.2).

Tabela 4.2 – Análise do Grau de Importância – CIDS com até 6 detectores

Número de IDS que detectaram o ataque	Número de IDS que compõem o CIDS					
	1	2	3	4	5	6
1	0,50	0,25	0,17	0,13	0,10	0,08
2	----	0,67	0,44	0,33	0,27	0,22
3	----	----	0,75	0,56	0,45	0,38
4	----	----	----	0,80	0,64	0,53
5	----	----	----	----	0,83	0,69
6	----	----	----	----	----	0,86

A análise das colunas da Tabela 4.2 mostra qual o número mínimo de IDSs que precisam detectar um ataque para que o CIDS considere este ataque relevante. Por exemplo, se Li = 0,30 e Ls = 0,40, um CIDS composto por 3 detectores já levanta um alerta relevante se os ataques forem percebidos por 2 dos seus detectores; se o CIDS for composto por 6 detectores, este mesmo alerta será levantado se 3 dos detectores perceberem o ataque.

A análise das linhas da Tabela 4.2 mostra a importância que é atribuída a um ataque de acordo com o número de alertas obtido no CIDS; por exemplo, , se Li = 0,30 e Ls = 0,40 então ataques detectados por 2 detectores de um CIDS podem ser "relevantes" (se o CIDS tiver 2, 3 ou 4 IDSs) ou do tipo que "merecem atenção" (se o CIDS tiver mais de 5 IDSs).

O gráfico da Figura 4.3, que mostra a variação do o grau de importância de um evento quando detectado por todos os detectores do CIDS, foi traçado para mostrar que a função de grau de pertinência comporta-se exponencialmente como esperado.

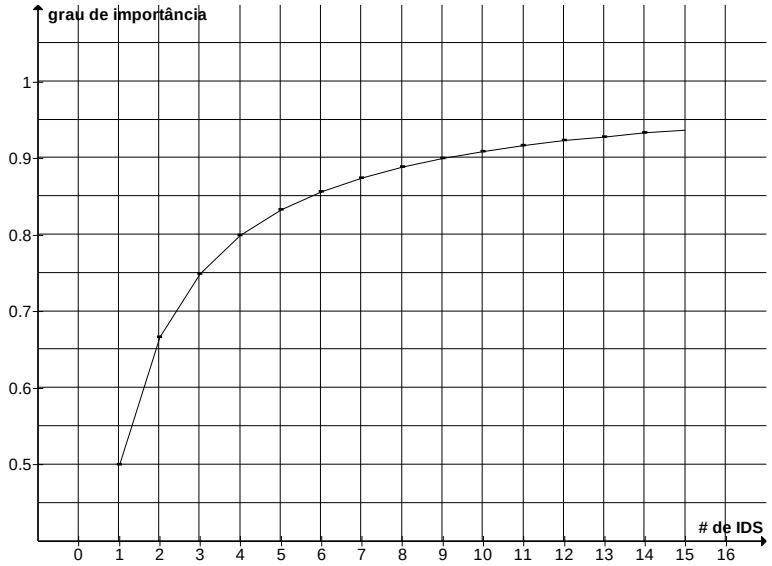

Figura 4.3 – Gráfico da variação do Grau de Importância de um ataque detectado por todos os detectores de um CIDS

4.4 Conclusão

O modelo matemático baseado na Teoria dos Conjuntos é suficiente para representar qualitativamente os eventos envolvidos em um sistema composto por um ou mais detectores, porém a análise quantitativa requer a atribuição de valores à qualidade dos eventos detectados de modo a permitir a comparação dos desempenhos de um CIDS com IDS individualmente. Para tal, optou-se pela fuzzificação do modelo e a construção de uma nova função de grau de pertinência que pudesse definir com maior precisão o grau de importância de cada evento detectado pelo CIDS. Ainda não se estabeleceram os valores limites (*Li* e *Ls*) que determinam o grau de importância dos eventos, no entanto o Capítulo 5 mostra uma situação onde é possível calcular os valores *Li* e *Ls* a partir dos dados coletados nos testes de laboratório.

Capítulo 5

Avaliação do Modelo

Neste capítulo, o modelo proposto no capítulo anterior é avaliado mediante testes executados em um laboratório em ambiente controlado. Dois são os objetivos dos testes: o primeiro consiste em, escolhidos 4 detectores diferentes e que possam garantir o conceito de diversidade de projeto, avaliar o desempenho de cada um dos detectores individualmente, considerando tanto sua capacidade de detectar os ataques simulados como o registro errôneo de eventos normais (falsos positivos). O segundo objetivo é, através da análise ROC (*Receiver Operation Characteristics*) do experimento, estabelecer os valores limites *Li* e *Ls* apresentados no Capítulo 4 e comparar o desempenho do CIDS composto pelos 4 IDSs escolhidos com o desempenho individual de cada um dos IDSs.

5.1 As Curvas ROC

A teoria das curvas ROC (*Receiver Operation Characteristics*) surgiu na década de 1940, mais especificamente após o ataque a Pearl Harbour durante a Segunda Grande Guerra. Este episódio tomou de surpresa os operadores de radares da força militar americana por não terem sido capazes de distinguir, nos sinais emitidos pelos radares, a iminência do ataque inimigo. Tais sinais, por terem sido confundidos com sinais "normais" e ruídos, foram desprezados pelos operadores dos radares [Green 1966].

A análise com curvas ROC é uma técnica de visualização, organização e seleção de classificadores de acordo com o seu desempenho [Fawcett 2004]. Com as curvas ROC, conceitos como "sensibilidade" e "precisão" foram trazidos da forma abstrata para uma forma concreta (numérica) e aplicados aos classificadores de forma geral. Estes conceitos traduzem

matematicamente os relacionamentos entre a calibragem de um classificador e o número de alarmes verdadeiros e de alarmes falsos que o classificador gera.

Em particular, podem ser analisados os classificadores binários, ou seja, aqueles que classificam eventos mapeando-os para os elementos de um conjunto de duas classes possíveis {p,n}, normalmente referentes aos casos de "positivo" e "negativo". Cada evento mapeado corretamente pelo classificador determina um resultado "verdadeiro" (positivo ou negativo); em contrapartida, cada evento mapeado erroneamente determina um resultado "falso" (positivo ou negativo) para o classificador. As possibilidades de geração dos verdadeiros e falsos podem ser resumidas na *tabela de contingência* ou *matriz de confusão* do classificador (Tabela 5.1) [Fawcett 2004]:

Tabela 5.1 – Matriz de Confusão

		Situação real	
		p	n
Classificação atribuída	p	VP	FP
	n	FN	VN

O desempenho de um classificador pode ser calculado através de algumas equações, como mostradas a seguir [Dasgupta 2001]. Considerando o número total de positivos como P= VP+FN e o número total de negativos como N = VN + FP:

- Taxa de Verdadeiros Positivos = VP/P
- Taxa de Falsos Positivos = FP/N ou (1-Especificidade)
- Sensibilidade = Taxa de Verdadeiros Positivos
- Especificidade = VN/N

Em geral, a curva ROC de um classificador é determinada pela seqüência dos pares ordenados (Taxa de Falsos Positivos, Taxa de Verdadeiros Positivos) obtidos com a variação de algum parâmetro no classificador que pode torná-lo mais ou menos "sensível" ou "específico" [Metz 1978]. A cada ajuste do classificador, obtém-se um par ordenado que é plotado no *espaço ROC* do gráfico (Figura 5.1). O mesmo classificador ajustado de maneiras diferentes pode gerar pontos distintos no espaço ROC. Os classificadores que geram pontos

acima da diagonal do quadrado (pontos A e C na Figura 5.1) apresentam uma melhor relação custo-benefício (falsos positivos x verdadeiros positivos) do que aqueles que geram pontos abaixo da diagonal (ponto B). O classificador ideal seria aquele que gerasse um ponto na coordenada (0,1), ou seja, que apresentasse taxa 0 de falsos positivos e 1 de verdadeiros positivos. A diagonal do quadrado mostra a situação em que classificadores agem aleatoriamente, onde pode ser esperados resultados iguais nas taxas de verdadeiro e falso positivos.

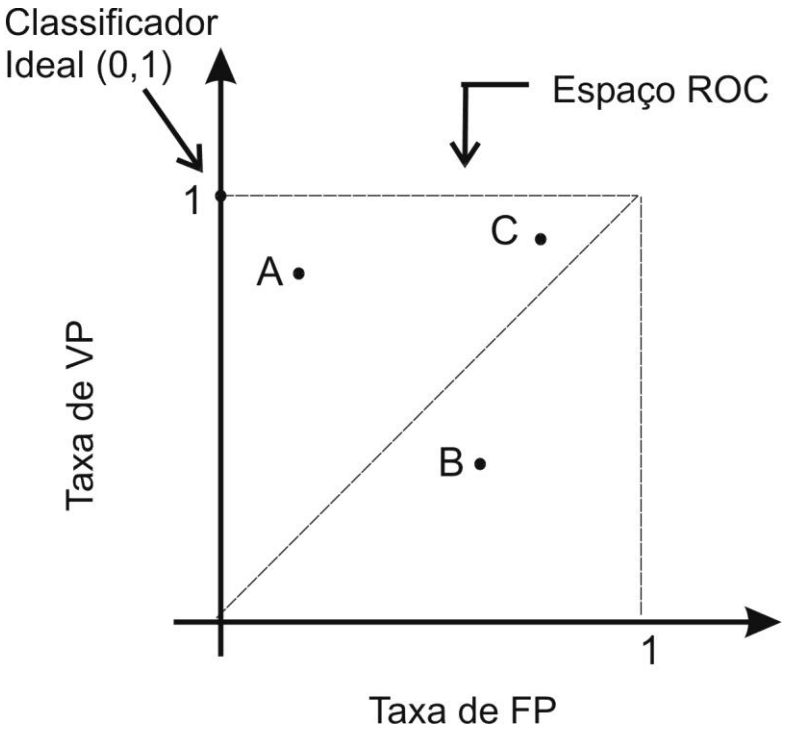

Figura 5.1 – Espaço ROC

Classificadores não-discretos (ou probabilísticos) são aqueles nos quais pode ser obtido mais de um ponto no espaço ROC. A sequência de pontos no espaço ROC é determinada pela variação de algum parâmetro do classificador que pode torná-lo mais conservador, gerando pontos mais próximos ao eixo X, ou mais liberal, gerando pontos mais próximos da diagonal e longe do eixo X (sempre à esquerda da diagonal). Um classificador não-discreto pode gerar uma curva ROC semelhante à da Figura 5.2. Informalmente, o melhor desempenho deste classificador é obtido com o ajuste correspondente ao ponto mais a

"noroeste" da curva (mais acima e mais à esquerda), indicado pela flecha na Figura 5.2, ou seja, o ponto que estiver à menor distância linear do comportamento ideal (0,1).

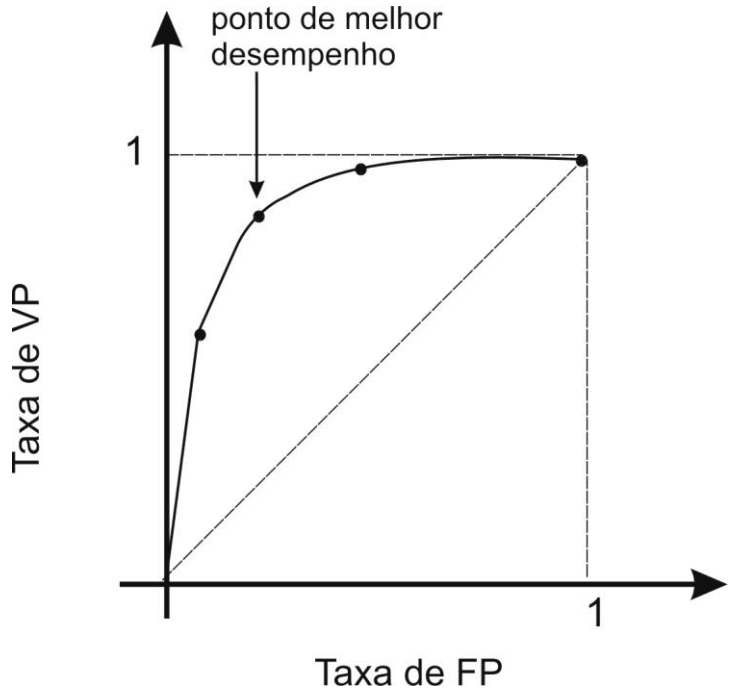

Figura 5.2 – Curva ROC

Neste trabalho, uma curva ROC será traçada de acordo com a variação do valor limite do grau de importância dos eventos detectados pelo CIDS. O melhor ponto da curva será utilizado para identificar quais são os valores *Li* e *Ls* (limite inferior e limite superior) que tornam mais precisos os resultados do CIDS composto por 4 detectores.

5.2 Ambiente de Testes

Nos testes de laboratório foram utilizados 4 computadores interligados por um hub, 4 programas de detecção de intrusão e 1 programa de simulação de ataques. Todos os programas de detecção de intrusão e sistems operacionais foram atualizados com as correções e assinaturas disponíveis até a data do experimento.

O programa escolhido para deflagrar os ataques foi o Tenable Nessus 3.0.4 (Southwest Research Institute, SwRI), uma ferramenta de auditoria cuja função principal é a de descobrir prováveis vulnerabilidades nos hosts para os quais os ataques são direcionados. O uso do

tradicional conjunto de testes composto pelo "DARPA Data Set" [Darpa 1998] e o aplicativo TCPReplay [SourceForge 2006], apesar de mais indicado, não se aplicou nos experimentos deste trabalho. Após alguns testes, percebeu-se que somente o detector baseado em rede do experimento (no caso, o Snort) foi capaz de perceber os ataques presentes no tráfego de rede simulado pelo TCPReplay enquanto que os demais detectores (todos baseados em host) somente detectariam os ataques cujos pacotes se destinassem explicitamente ao endereço lógico e físico (endereço IP e MAC) do host no qual o HIDS estava sendo executado. Foi necessário, portanto, optar pelo simulador Nessus que possibilitou atribuir, para cada ataque, um endereço IP explícito para o host de destino.

Computadores:

- P750SRV: Pentium III 750 MHz, 512 MB RAM, Windows 2000 Server (atacante / detector);
- AthlonSRV: AMD Athlon 1,7 GHz, 512 MB RAM, Windows 2000 Server (detector / atacante);
- P3SRV: Pentium IV 2,0 GHz, 1 GB RAM, Windows 2003 Server e serviços web (IIS 6.0) e DNS – gerador de tráfego de fundo;
- P4SRV: Pentium IV 2,0 GHz, 1 GB RAM, Windows 2003 Server e serviços de correio eletrônico (POP3, SMTP, IMAP e WebMail) e DNS – gerador de tráfego de fundo.

Programas de detecção de intrusão:

- Snort 2.4.3 build 26 (SourceFire Network Security)
- KFSensor versão 4.2.0 *trial* (KeyFocus Ltd.)
- HoneyBOT versão 0.1.2 *trial* (Atomic Software Solutions)
- Ground Zero X-Ray (Ground Zero Security Research and Software Development)

Estes programas foram escolhidos pela sua popularidade, facilidade de instalação e por disporem de versões *freeware* ou de demonstração.

Simulação de ataques:

- Programa: Tenable Nessus 3.0.4 Build W306 (Southwest Research Institute)
- Ataques: 25 ataques escolhidos dentre DoS, BackDoors, e exploradores de vulnerabilidades para WebServers, FTP, RPC dentre outros, identificados na Tabela 5.2.

5.3 Descrição dos Experimentos

Para a análise dos eventos identificados por cada um dos detectores, todos os computadores foram conectados a um hub em uma rede física e lógica (172.16.0.0/16) separada da rede de serviços (192.168.0.0/24). Para garantir resultados mais precisos, os experimentos foram executados na modalidade de "turno e returno", ou seja, no primeiro turno um computador (P750SRV) foi o disparador de ataques e o outro computador (ATHLONSRV) executou cada um dos IDSs a ser analisado; no returno ocorreu o inverso. Cada um dos 25 ataques foi disparado individualmente sobre cada um dos IDS separadamente e os resultados foram registrados. O tráfego de fundo foi cuidadosamente controlado para se obter uma análise confiável da ocorrência de falsos positivos.

Para uma maior precisão na identificação de falsos positivos, foram executadas, em separado, algumas operações envolvendo consultas a servidores DNS, navegação de páginas web, consultas a parâmetros SNMP, transferência de arquivos usando programas de FTP, transferência de arquivos usando o Windows Explorer (através de compartilhamento de pastas do Windows) e acessos à console dos servidores via TELNET. É importante ressaltar que a contagem dos falsos positivos foi feita pelas diferentes classes de alarmes falsos gerados em cada um dos detectores (eventos NETBIOS, ICMP, etc.) e não pelo número total de alarmes falsos registrados no log de cada um dos detectores.

5.4 Testes Realizados

Nos testes de avaliação dos detectores, foram disparados do computador atacante, tendo como alvo o computador atacado, os 25 ataques, um por um. A cada ataque, o efeito sobre o IDS era analisado e, após a verificação do efeito do ataque sobre o detector, o log era movido para uma área separada de armazenamento e o programa era reiniciado. Em alguns casos, como no KFSensor, era necessário reiniciar o computador atacado ou, pelo menos,

aguardar o tempo necessário para o programa desbloquear as portas atacadas. Em outros casos, quando o detector não identificava o ataque na primeira rodada, o mesmo era repetido 3 vezes; se houvesse pelo menos uma identificação, o resultado era considerado como Verdadeiro Positivo.

Nos testes de análise de Falsos Positivos, foram feitas tentativas "normais" de acessos a serviços existentes e inexistentes no computador de destino. Neste trabalho, considerou-se que um acesso a um serviço disponível ou não no computador de destino não deve ser considerado como ataque a não ser que este acesso ultrapasse os limites mínimos estabelecidos pelo parâmetros de configuração originais do serviço ou do programa de detecção. Portanto, estabeleceu-se que se um detector gerasse um alarme durante uma tentativa de acesso normal, este alarme seria considerado Falso Positivo.

Os 25 ataques estão listados na Tabela 5.2 a seguir. Durante os testes, houve um contratempo: dois dos ataques do tipo BackDoor foram totalmente ignorados por *todos* os detectores. Este detalhe criou a necessidade de se fazerem duas baterias de análises numéricas: na primeira, todos os 25 ataques foram considerados e, na segunda, somente os 23 ataques detectados foram incluídos nas análises. Ao final dos cálculos, percebeu-se que considerar ou não os dois ataques ignorados não fez nenhuma diferença nos números finais. Por uma questão de clareza optou-se, portanto, em mostrar os cálculos neste trabalho considerando-se somente os 23 ataques detectados.

Tabela 5.2 – Ataques deflagrados pelo Nessus

Categoria Nessus	Ataque	Descrição
BackDoors	BackOrifice	Testa a presença do cavalo-de-Tróia BackOrifice
	BugBear	Testa a presença do *worm*, *backdoor* e *keylogger* BugBear
	DeepThroat	Testa a presença do *backdoor* DeepThroat
	FingerBackdoor	Testa a reação ao comando cmd_rootsh@target
	IISPossbileCompromise	Testa a presença de arquivos alterados no IIS
	PortalOfDoom	Testa a presença do *backdoor* PortalOfDoom
	Sygate BackDoor	Testa a presença do *remote controller* Sygate
	MyDoom	Testa a presença do *backdoor* MyDoom
DoS	IISFPDoS	Testa uma vulnerabilidade do Microsoft FrontPage (ref. MS00-100)

	PHPImageFile	Testa vulnerabilidade nas rotinas *php_handle_iff* e *php_handle_jpeg*
	Winlogon.exe DoS	Testa vulnerabilidade da rotina *winlogon.exe*
	Personal Web Sharing	Tenta "derrubar" o serviço Personal Web Sharing
Useless Services	NetStat	Verifica se o Netstat está em execução
	Windows Terminal Service	Verifica se o serviço de terminais do Windows está ativo
	Telnet	Verifica se o serviço de Telnet está ativo
	WriteSrv	Verifica se o serviço WriteSRV está ativo
WebServers	FrontPage Passwordless	Verifica se servidor FrontPage está desprotegido (sem senha)
	IISRemoteCommExecution	Testa vulnerabilidade no Microsoft IIS (ref. MS01-044)
Windows	CyDoor detection	Verifica se o host remoto está executando o programa CYDOOR
	GatorDetection+Gain	Verifica se o host remoto está executando o programa GATOR
	I-Nav ActiveX BufferOverf.	Tenta atacar uma vulnerabilidade no ActiveX
	IE VersionCheck	Verifica se o host remoto está executando uma versão não mais suportada do Internet Explorer
FTP	FTP Shell DoS Vuln.	Testa vulnerablidades do serviço FTP
RPC	RPC Port Mapper	Testa a presença do serviço RPC
Remote File Access	AliBaBa Path Climbing	Testa a possibilidade de acessar pastas de um servidor WEB usando URLs com comandos GET

A Tabela 5.3 mostra os eventos "normais" registrados como ataque pelos detectores (Falsos Positivos).

Tabela 5.3 – Falsos Positivos capturados pelos detectores

Protocolo	Porta
UDP	38293
	137
	138
	1027

	2967
TCP	3089
	3090
	3088
	139
ICMP	---

As Tabelas 5.4 e 5.5 mostram todos os eventos detectados pelos 4 IDSs testados. O registro positivo de um evento por um detector é identificado pela presença do sinal "√" no cruzamento da linha correspondente ao evento com a coluna correspondente ao detector; já o registro negativo é identificado pela célula vazia na posição correspondente.

Tabela 5.4 – Ataques detectados pelos IDSs

Categoria Nessus	Ataque	Snort	KFSensor	HoneyBot	X-Ray
BackDoors	BackOrifice	√	√	√	
	BugBear	√	√	√	
	DeepThroat (descartado)				
	FingerBackdoor (descartado)				
	IISPossbileCompromise	√	√	√	
	PortalOfDoom	√			
	Sygate BackDoor	√	√		
	MyDoom	√	√	√	
DoS	IISFPDoS	√		√	
	PFPImageFile	√		√	
	Winlogo.exe Dos	√	√	√	
	Personal Web Sharing	√	√	√	√
Useless Services	NetStat		√	√	
	Windows Terminal Service	√	√		
	Telnet		√	√	
	WriteSrv			√	

WebServers	FrontPage Passwordless	√		√	
	IISRemoteCommExecution	√	√	√	
Windows	CyDoor detection	√	√		
	GatorDetection+Gain	√	√		√
	I-Nav ActiveX BufferOverf.	√	√		
	IE VersionCheck	√	√		
FTP	FTP Shell DoS Vuln.		√		
RPC	RPC Port Mapper	√	√	√	
Remote File Access	AliBaBa Port Climbing	√	√	√	

Tabela 5.5 – Falsos Positivos detectados pelos IDSs

Falsos Positivos	Snort	KFSensor	HoneyBot	X-Ray
UDP 38293			√	
TCP 3089	√			
TCP 3090	√			
TCP 3088	√			
UDP 137		√		
UDP 138		√		
UDP 1027		√		
UDP 2967		√		
TCP 139	√	√		
ICMP	√	√		

Os resultados obtidos nos testes individuais permitem o cálculo do grau de importância de cada evento registrado nas Tabelas 5.4 e 5.5. Como foi proposto na Seção 4, para cada evento ***e*** das Tabelas 5.4 e 5.5, o cálculo do grau de importância do evento será feito usando-se $n = 4$, $S = \sum_{k=1}^{n} f_k(e)$ e $\mu\, I(\boldsymbol{e}) = \left(1 - \left(\frac{1}{1+S}\right)\right) * \frac{S}{n}$.

Os resultados obtidos estão na Tabela 5.6, apresentados em ordem crescente de acordo com o grau de importância dos eventos, desprezados os dois eventos ignorados por todos os detectores.

Tabela 5.6 – Grau de Importância dos Eventos

Ataque	Grau de Importância
WriteSrv	0,125
FTP Shell DoS Vuln.	0,125
PortalOfDoom	0,125
NetStat	0,333
Telnet	0,333
IISFPDoS	0,333
PFPImageFile	0,333
FrontPage Passwordless	0,333
Sygate BackDoor	0,333
Windows Terminal Service	0,333
CyDoor detection	0,333
I-Nav ActiveX BufferOverf.	0,333
IE VersionCheck	0,333
GatorDetection+Gain	0,563
BackOrifice	0,563
BugBear	0,563
IISPossbileCompromise	0,563
MyDoom	0,563
Winlogo.exe Dos	0,563
IISRemoteCommExecution	0,563
RPC Port Mapper	0,563
AliBaBa Port Climbing	0,563
Personal Web Sharing	0,800
Falsos Positivos	
UDP 38293	0,125
TCP 3089	0,125
TCP 3090	0,125

TCP 3088	0,125
UDP 137	0,125
UDP 138	0,125
UDP 1027	0,125
UDP 2967	0,125
TCP 139	0,333
ICMP	0,333

5.5 Análise dos Resultados Obtidos

A Tabela 5.7 mostra o cálculo dos pontos da curva ROC e a Figura 5.3 mostra a curva ROC do CIDS testado, considerando os 23 ataques válidos. Aqui, os Falsos Positivos/Negativos e Verdadeiros Positivos/Negativos são recontados sob a ótica do CIDS, sendo:

- FP-CIDS – falsos positivos do CIDS: todos os eventos normais com grau ≥ $\mu I(\boldsymbol{e})$;
- VP-CIDS – verdadeiros positivos do CIDS: todos os ataques com grau ≥ $\mu I(\boldsymbol{e})$;
- FN-CIDS – falsos negativos do CIDS: todos os ataques com grau < $\mu I(\boldsymbol{e})$;
- VN-CIDS – verdadeiros negativos do CIDS: todos os eventos normais com grau < $\mu I(\boldsymbol{e})$.

As taxas de Falsos Positivos e de Verdadeiros Positivos serão calculadas de acordo com o que foi apresentado na seção 5.2 e plotadas no espaço ROC (Figura 5.3).

Tabela 5.7 – Taxa de Falsos Positivos x Taxa de Verdadeiros Positivos

	μ I(e)	FP-CIDS	VP-CIDS	FN-CIDS	VN-CIDS	Taxa FP	Taxa VP
1	0,805	0	0	23	10	0,000	0,000
2	0,800	0	1	22	10	0,000	0,043
3	0,570	0	1	22	10	0,000	0,043
4	0,563	0	1	22	10	0,000	0,043
5	0,335	0	10	13	10	0,000	0,435
6	0,330	2	20	3	8	0,200	0,870
7	0,130	2	20	3	8	0,200	0,870
8	0,125	10	23	0	0	1,000	1,000
9	0,000	10	23	0	0	1,000	1,000

Ao examinar a Tabela 5.7, é possível perceber que alguns pontos apresentam coordenadas repetidas (Taxa FP, Taxa VP). Portanto, ao traçar a Curva ROC, estes pontos serão representados no Espaço ROC da Figura 5.3 da seguinte forma:

- as coordenadas da linha 1 são representadas pelo ponto A,
- as coordenadas das linhas 2, 3 e 4 são representadas pelo ponto B,
- as coordenadas da linhas 5 são representadas pelo ponto C,
- as coordenadas das linhas 6 e 7 são representadas pelo ponto D,
- as coordenadas das linhas 8 e 9 são representadas pelo ponto E.

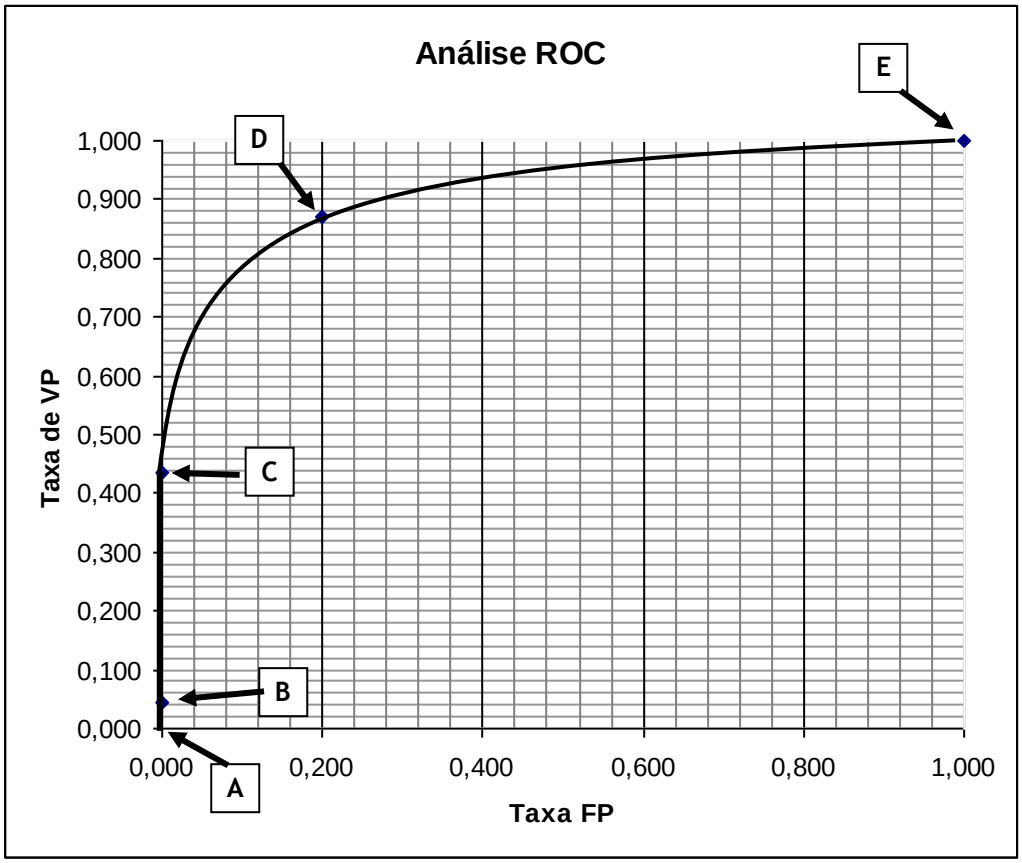

Figura 5.3 – Curva ROC do CIDS

O "melhor" ponto de uma Curva ROC é aquele onde se tem a melhor relação entre a Taxa de FP e a Taxa de VP; no caso desta curva, temos dois "melhores pontos", ambos no ponto "D" do Espaço ROC: o primeiro é onde μ I(***e***) é 0,13 (linha 7 da Tabela 5.7), e o segundo é onde μI(***e***) é 0,33 (linha 6 da Tabela 5.7). Portanto, estabeleceremos os valores *Li* e *Ls* necessários para completar a Tabela 4.1 usando *Li* = 0,13 e *Ls* = 0,33 e definiremos a Tabela de Graus de Importância para este CIDS (Tabela 5.8).

Tabela 5.8 – Graus de Importância para os eventos do CIDS

μI(*e*)	Grau de Importância
$0 \leq \mu I(\boldsymbol{e}) < 0{,}13$	não merece atenção
$0{,}13 \leq \mu I(\boldsymbol{e}) < 0{,}33$	relevante
$\mu I(\boldsymbol{e}) \geq 0{,}33$	muito relevante

O intuito deste trabalho é o de comparar o desempenho de um CIDS com o desempenho individual de cada IDS em um mesmo contexto, dentro de um mesmo conjunto de parâmetros de comparação. Em outras palavras, deseja-se atribuir ao CIDS uma propriedade específica: ser mais preciso do que qualquer um dos IDSs utilizados na sua composição. Para tal, serão considerados como fundamentos os conceitos de *segurança* e *vivacidade* [Lamport 1997]. De forma resumida espera-se que, no quesito "segurança", o CIDS não seja pior do que cada um dos IDSs testados e que, no quesito "vivacidade", o CIDS traga, de alguma forma, algum resultado melhor do que cada um dos IDSs testados. Resultados melhores ou piores de um IDS são analisados, em última instância, pela quantidade de alarmes falsos que ele apresenta durante o seu funcionamento. Comparar, separadamente, as taxas de Falsos Positivos e Falsos Negativos não comprova a superioridade do CIDS mas, na soma de todos os alarmes falsos, o CIDS leva vantagem sobre todos os IDSs testados. A Tabela 5.9 comprova que, na soma dos Falsos Positivos (FP) e Falsos Negativos (FN), o CIDS sempre gera um número *menor* de alarmes falsos.

Tabela 5.9 –Segurança e Vivacidade do CIDS

Detector	FN	FP	Total
CIDS	3	2	5
Snort	4	5	9
KFSensor	5	6	11
HoneyBOT	8	1	9
X-Ray	21	0	21

O experimento também trouxe outro resultado interessante para o CIDS: não só o CIDS gerou um número menor de alarmes falsos, mas também o seu número de "acertos" (verdadeiros positivos) foi maior do que o número de acertos de cada um dos IDSs isoladamente (Tabela 5.10).

Tabela 5.10 –Comparativo no Número de Verdadeiros Positivos

Detector	VP
CIDS	20
Snort	19
KFSensor	18
HoneyBOT	15
X-Ray	2

Para complementar visualmente a comparação entre o CIDS e os IDSs individuais, foram traçadas as Curvas ROC que comparam os melhores pontos de operação de cada um dos IDS testados e a Curva ROC do CIDS (Figura 5.4). É possível observar que a curva do CIDS heterogêneo é a mais externa (mais à esquerda) e denota, segundo a teoria das Curvas ROC, um comportamento mais preciso do classificador heterogêneo em relação aos demais classificadores. As curvas ROC também mostram que, se fosse o caso de se optar por um IDS individual, a melhor escolha seria usar o HoneyBOT.

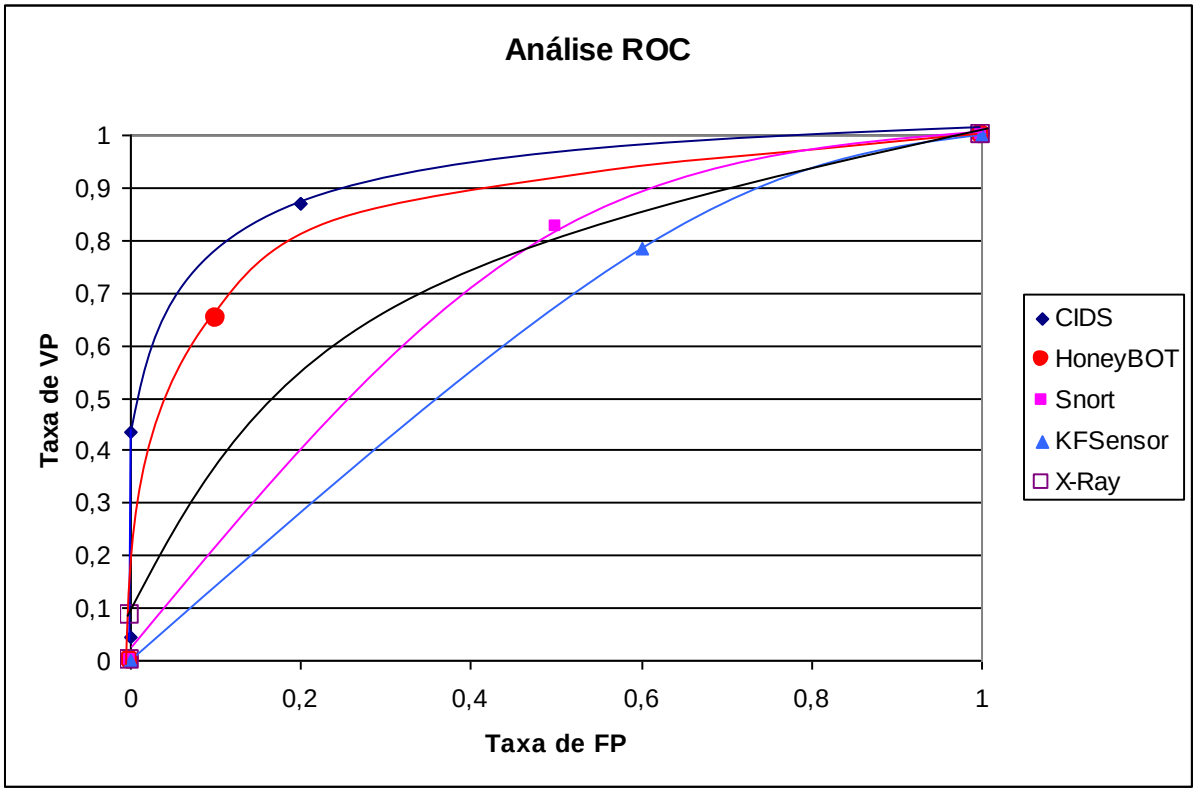

Figura 5.4 – Superposição de Curvas ROC para os 4 IDSs e para o CIDS

Outros aspectos do experimento não podem deixar de ser notados:

- *Com relação a diversidade de hardware e software*: os experimentos mostraram que o mesmo IDS instalado em dois computadores independentes, porém configurados de forma semelhante, produz resultados diferentes. Conclui-se, portanto, que o comportamento de um mesmo IDS não pode ser replicado em hardwares diferentes, ou seja, o mesmo programa de detecção de intrusão sendo executado simultaneamente em dois computadores diferentes, mesmo analisando o mesmo fluxo de dados, pode trazer resultados diferentes na detecção de intrusão;
- *Com relação ao programa de detecção de intrusão "X-Ray"*: o programa X-Ray é, caracteristicamente, um programa de vivacidade reduzida, mas extremamente seguro. Explica-se: dos 23 ataques deflagrados contra o X-Ray, somente 2 foram detectados e, em contrapartida, nenhum alarme falso foi gerado por ele. A princípio, considerou-se descartar o X-Ray dos experimentos, por apresentar uma taxa tão baixa de verdadeiros positivos. Porém, ao analisar mais detidamente o comportamento do CIDS e comparando-se sua eficiência ao incluir e ao excluir o X-Ray, percebeu-se que o quesito "segurança" do X-Ray contribuiu positivamente para o resultado do CIDS como um todo;
- *Independência do modelo proposto com relação à tecnologia adotada*: Mesmo sem ainda estar provada formalmente a adequação do modelo para classificadores binários, observou-se que este modelo pode ser aplicado a qualquer tipo de IDS, independentemente da sua tecnologia.

5.6 Trabalhos Correlatos

Durante a pesquisa foram identificados alguns trabalhos que também se utilizaram de modelos matemáticos para analisar detectores de intrusão, alguns deles baseados na Teoria Difusa. [Leckie 2004] apresenta uma metodologia para ser utilizada em IDSs baseados em anomalias, o BSEADS (Behavioral Secure Enclave Attack Detection System). Esta metodologia analisa diversos aspectos, como tráfego de rede e comportamento de usuários,

combinando estes dados com o que se conhece a respeito de protocolos de segurança e, garantem os autores, o resultado desta análise identifica prováveis comportamentos anômalos. A intenção do BSEADS é a de se implementar um IDS em ambientes com alto índice de criptografia, que seja capaz de analisar o comportamento neste ambiente e detectar anomalias sem a necessidade de analisar o tráfego de rede em si. Ao apresentar o BSEADS, os autores se utilizam de um modelo semelhante ao apresentado aqui, baseado em Teoria dos Conjuntos (figura 5.5) , mas particularizando a representação de eventos em somente um IDS.

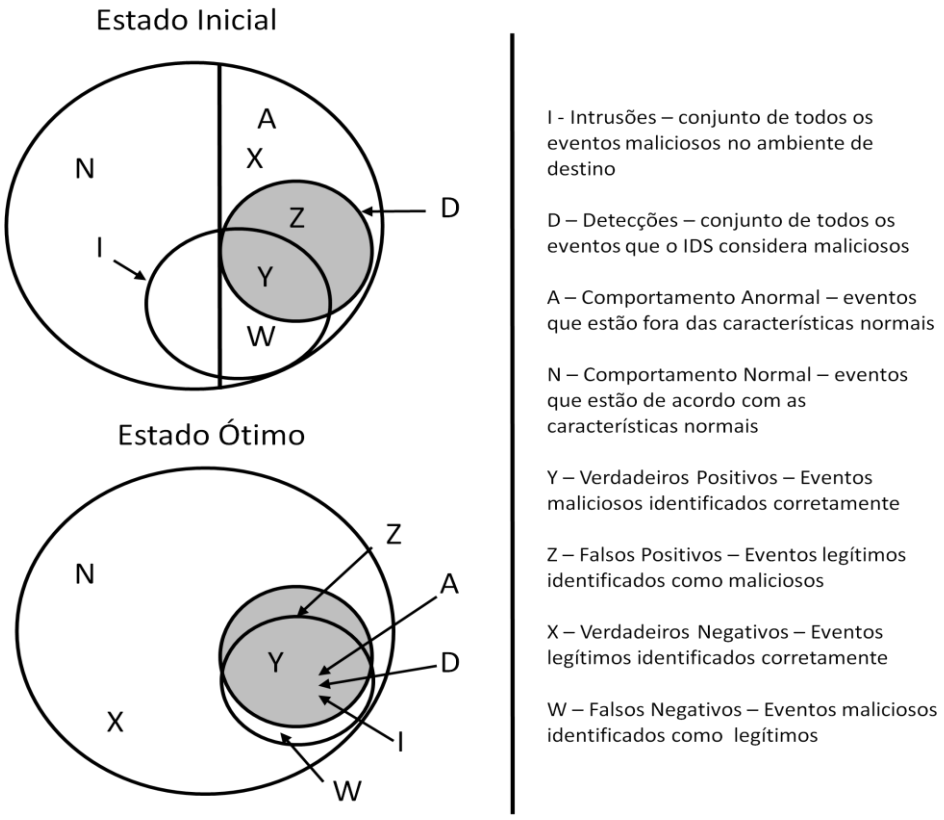

Figura 5.5 – Diagrama de Venn para um sistema baseado em anomalias [Leckie 2004]

[Maia 2004] apresenta uma proposta para um novo IDS, o SDDI (Sistema Difuso de Detecção de Intrusão) que, nas palavras dos autores, se propõe a utilizar "um método de inteligência artificial empregando lógica difusa (Fuzzy Logic) em conjunto com a técnica de detecção por anomalia para a detecção de intrusão do tipo varredura de portas (*port scanning*), muito utilizado nos dias de hoje para verificar a existência de serviços disponíveis e assim

explorar possíveis vulnerabilidades nos sistemas computacionais". O SDDI utiliza um sistema de inferência difusa e três funções de grau de pertinência para analisar as características dos acessos feitos a um sistema e detectar os prováveis ataques. A metodologia apresentada nesse trabalho se aproxima um pouco do assunto desta tese por tratar dos processos de fuzzificação de desfuzzificação no modelo matemático de um IDS. A diferença, porém, é que os autores não criam novos objetos no estudo, mas se utilizam de processos conhecidos da lógica fuzzy, como a função de pertinência triangular, o raciocínio difuso baseado em máximo e mínimo e a desfuzificação baseada na estratégia do centróide. Também observou-se que os autores não tinham, dentre seus objetivos, a extensão do modelo para um conjunto de IDSs heterogêneos.

Na área de modelagem, [Vert 1998] apresenta o Vismath, um modelo geométrico para representação visual da variação do ambiente de processamento de um computador. O modelo, chamado *spicule*, pretende representar vetorialmente cada uma das variáveis monitoradas em um sistema, entre elas taxa de uso da CPU, contagem dos processos de sistema, de usuário e número de arquivos abertos. Estabelecido o modelo do comportamento "normal" de um computador, o monitoramento do *spicule* ao longo do tempo pode identificar, através de mudanças abruptas, um comportamento anormal típico de algumas invasões. A Figura 5.6 mostra dois estados de um computador, o primeiro antes e o segundo após um ataque que, tipicamente, replica (*fork*) processos.

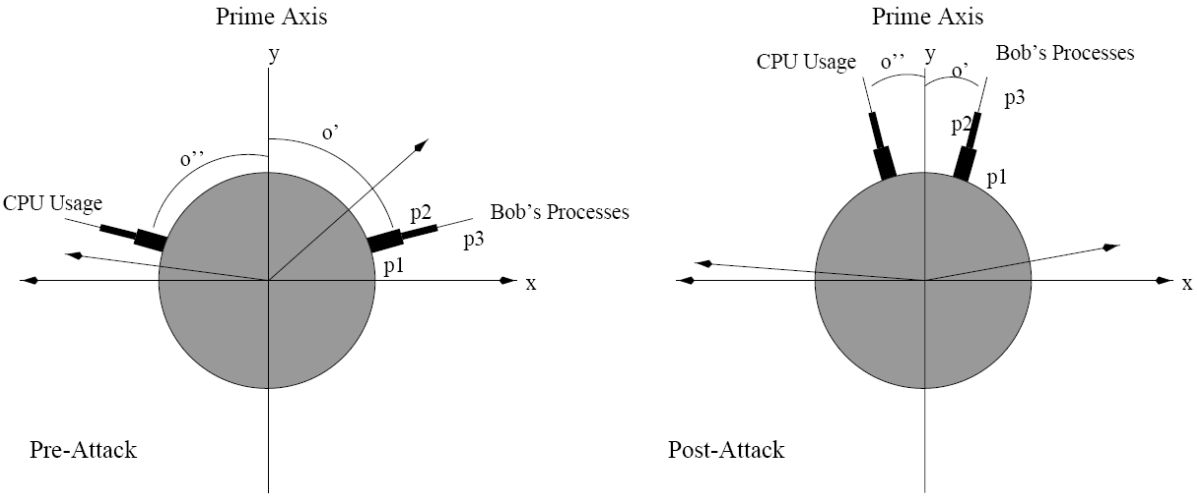

Figura 5.6 – Modelagem de um ataque usando *spicules* [Vert 1998]

[Ulvila 2003] é o trabalho que mais se assemelha ao assunto apresentado nesta tese: nele, os autores apresentam uma forma de avaliar o desempenho de um CIDS usando curvas ROC para, depois, apresentar uma forma alternativa de avaliação, através de uma metodologia baseada em "custos". As diferenças entre [Ulvila 2003] e este trabalho residem, basicamente, em três aspectos. Um primeiro aspecto é evidenciado quando os autores não se preocupam em detalhar a geração das curvas ROC dos IDSs e propõem a geração da Curva ROC do CIDS pela superposição das Curvas ROC dos IDS individuais; a tese aqui apresentada propõe uma modelagem capaz de gerar a Curva ROC do CIDs independentemente das Curvas ROC dos IDSs individuais. Um segundo aspecto se mostra na metodologia de análise dos autores, que desconsidera a diferença de resultados quando há descartes de eventos por um ou mais IDSs do CIDS. Na metodologia de regras de decisão empregada em [Ulvila 2003], os resultados da detecção de um ataque em um CIDS com *n* detectores *podem ser equivalentes* aos resultados gerados por um dos IDS deste CIDS operando isoladamente (isto ocorreria no caso de somente este IDS em particular detectar o ataque em questão). No modelo aqui proposto, numa mesma situação, os resultados de um IDS operando sozinho, refletidos pelo grau de importância do evento, *serão sempre diferentes* daqueles apresentados quando o IDS faz parte de um CIDS (vide Tabela 4.2), mesmo que somente este IDS, quando participando de um CIDS, detecte o ataque. Por último, [Ulvila 2003] afirma que compor o resultado de dois IDSs idênticos pode ser melhor do que compor os resultados de dois IDSs diferentes, ou seja, o CIDS "homogêneo" pode ser melhor do que o CIDS "heterogêneo".

5.7 Conclusão

Neste capítulo, foi apresentada a metodologia para os testes e comparação de desempenho de um CIDS composto por 4 IDSs diferentes, com a intenção de comprovar que o desempenho de um CIDS baseado em diversidade de projetos pode ser melhor do que o desempenho de cada um dos IDSs que o compõem.

A análise comparativa foi feita através de testes em ambiente controlado com cada um dos detectores individualmente, com o intuito de identificar o número de alarmes verdadeiros e falsos que cada um dos IDSs é capaz de gerar sob as mesmas condições de funcionamento. Ao final dos testes, o desempenho geral do CIDS foi avaliado com o apoio do modelo matemático apresentado no Capítulo 4, que possibilitou a avaliação do grau de importância de cada evento detectado pelo CIDS. O modelo apresentado não identificou, a princípio, qual

seria o melhor intervalo numérico para identificar a partir de que grau de importância um evento deveria ser considerado relevante ou não para a análise. Entretanto, a análise do modelo com as curvas ROC foi determinante e apontou, com exatidão, quais os melhores valores-limite para o grau de importância. De posse destes valores, foi possível recalcular as taxas de erros do CIDS e compará-las com as taxas de erros de cada um dos IDS; o resultado mostrou que, neste estudo, o CIDS apresenta uma taxa de erros menor do que qualquer um dos IDSs que o compõem. A comparação da taxa de acertos também se mostrou otimista nos experimentos efetuados: o CIDS também acerta mais do que cada um dos IDSs testados.

Capítulo 6

Conclusão

A intenção principal deste trabalho é a de possibilitar a representação do funcionamento de um sistema de detecção de intrusão (IDS) usando-se uma modelagem matemática. A partir deste modelo, poderíamos experimentar a combinação de IDSs, criar o IDS-composto (CIDS) e analisar e comparar o seu desempenho com o desempenho de cada IDS individual usado na sua composição. O modelo criado a partir da Teoria dos Conjuntos trouxe bons resultados pois possibilitou a representação de *n* detectores em funcionamento, independente do projeto original de cada um deles (se HIDS ou NIDS, ou se baseado em assinaturas ou anomalias, etc.). Porém este modelo não possibilitou fazer-se uma análise numérica de desempenho nem do IDS, nem do CIDS.

A extensão do modelo dentro de uma teoria mais abrangente (a Teoria dos Conjuntos Difusos, ou Teoria Fuzzy) mostrou-se oportuna, mas com um detalhe: não era conveniente usar as funções de grau de pertinência conhecidas, pois elas não representariam bem as nuances de comportamento do CIDS. A definição de uma nova função de grau de pertinência foi um risco pois não havia meios de determinar o que aconteceria com o resultado final na avaliação do desempenho do CIDS usando esta nova função. Proposta a nova função de grau de pertinência, passou-se para as análises práticas em laboratório com alguns produtos disponíveis no mercado e uma bateria de testes controlados. Obviamente, a abrangência dos testes foi bem reduzida e específica, mas trouxe resultados otimistas com relação a um trabalho futuro de generalização do modelo e da metodologia de avaliação.

Em linhas gerais, o trabalho foi bem sucedido considerando-se o intuito de modelar, avaliar e comparar o desempenho de um CIDS baseado em diversidade de projetos. Apesar de as análises comparativas entre CIDS e os IDSs escolhidos para sua composição terem sido feitas para um caso particular, os resultados da comparação foram uma boa surpresa. Intuitivamente, se desejávamos que o CIDS fosse "melhor" do que o uso de somente um IDS, isto foi demonstrado pelo modelo.

6.2 Contribuição

Acreditamos que este trabalho contribuirá para a pesquisa na área de detecção de intrusão nos seguintes aspectos:

- *Maior detalhamento na elaboração de curvas ROC para IDSs e IDSs compostos (CIDS)* – a metodologia apresentada propõe uma nova maneira de elaborar a calibragem de um CIDS com a atribuição informal, a cada um dos IDSs individuais, de um "grau de confiança no IDS" feita de acordo com o número de eventos que cada um deles é capaz de detectar. Com isto, o trabalho propõe um novo conceito e uma nova variável (o "grau de importância" de um evento) que desloca o pólo dos cálculos, relativos ao desempenho de um IDS, do IDS em si para os eventos, sejam eles detectados ou não.
- *Uma metodologia alternativa de análise de IDSs e CIDS independente de teorias probabilísticas* – todos os estudos pesquisados até aqui baseiam a análise do desempenho de IDSs em estudos estatísticos ou probabilísticos. A modelagem proposta aqui preocupou-se em fugir de estudos prontos e criar uma visão nova e independente do que já existe.
- *O incentivo para o aplicação do conceito de diversidade de projetos em detecção de intrusão* – o modelo proposto demonstra que um CIDS pode ter desempenho superior a qualquer um dos IDSs escolhidos para compô-lo. Em [Lippmann 2000] e em [Maxion 2005] fica patente que diferentes sistemas de detecção de intrusão apresentam diferentes falhas na detecção,

o que nos leva a indicar o CIDS como um projeto ideal para minimizar a quantidade de alarmes falsos. Uma versão preliminar desta metodologia foi apresentado em [Raguenet 2006].

6.3 Trabalho futuros

Algumas questões ainda estão pendentes e poderão ser resolvidas em trabalhos futuros:

- Pretende-se executar uma análise mais abrangente do modelo incluindo-se características de detectores baseados em anomalias. Assim, talvez seja possível verificar se a modelagem se estende a CIDS que incluam tanto IDSs baseados em assinaturas quanto IDSs baseados em anomalias.
- O modelo foi testado em um CIDS particular, composto por quatro IDSs baseados em assinaturas, escolhidos dentre os programas disponíveis no mercado entre 2003 e 2006. Gostaríamos de poder demonstrar formalmente e matematicamente a teoria proposta, independentemente dos programas escolhidos, para que a abrangência do modelo seja comprovada e o modelo possa ser estendido para *n* detectores independentes.

Referências

Allen, J. et al. “State of the Practice of Intrusion Detection Technologies”, Technical Report CMU/SEI-99-TR-028 ESC-99-028, January 2000.

Avizienis, A., “Design Diversity and the Immune Paradigm: Conerstones for Information System Survivability”, Third Information Survivability Workshop -- ISW-2000, October 24-26, 2000.

Avizienis, A. e M. R. Lyu “Assuring design diversity in N-version software: a design paradigm for N-version programming”, pages 197–218. In J. F. Meyer and R. D. Schlichting editors, *Dependable Computing for Critical Applications 2*. Springer-Verlag, Wien, New York, 1992.

Bezerra de Mello, T. e Hexsel, R.,“Correlacionamento Distribuído de Alertas em Sistemas de Detecção de Intrusão”, 2004, Departamento de Informática, UFPR, Centro Politécnico - Curitiba, PR, URL http://www.inf.ufpr.br/roberto/sbseg05-nariz.pdf , acessado em 2004.

Crosbie, M e Spafford, G., “Active defense of a computer system using autonomous agents”, Technical Report 95-008, COAST Group, Department of Computer Sciences, Purdue University, West Lafayette, Indiana, February 1995. URL http://www.cerias.purdue.edu/homes/spaf/ tech-reps/9508.ps., acessado em 2004.

Darpa, Intrusion Detection Evaluation Web Page Template, Lincoln Laboratory, Massachusetts Institute of Technology, 1998 Training Data Sets, acessado em 2006.

Dasgupta, D., Gomez, J. “Evolving Fuzzy Classifiers for Intrusion Detection”, in Proceedings for the 2002 IEEE Workshop on Information Assurance, West Point, 2001.

Debar, H, Dacier, M. e Wespi, A., “A Revised Taxonomy for Instrusion Detection Systems”, Reseach Report RZ 3176, October 1999.

Debar, H., Wespi, A., “Aggregation and Correlation of Intruder-Detection Alerts” , 2001, in Recent Advances in Intrusion Detection : 4th International Symposium, RAID 2001 Davis, CA, USA, October 10-12, 2001, Proceedings, p.85.

Denning, D. E. "An Intrusion-Detection Model," sp, p. 118, 1986 IEEE Symposium on Security and Privacy, 1986.

Fawcett, T., "ROC Graphs: Notes and Practical Considerations for Researchers", HP Laboratories, 2004.

Fyodor, Y., "Snortnet A Distributed Intrusion Detection System", 2000, online http://snortnet.scorpions.net/snortnet.pdf.

Ghosh, A. e Sen, S., "Agent-Based Distributed Intrusion Alert System – ABDIAS", in *Lecture Notes in Computer Science*, Vol. 3326/2004, pgs. 240-251, Springer Berlin/Heidelberg.

Green, D.M., Swets, J.M. Swets "Signal detection theory and psychophysics." New York: John Wiley and Sons Inc, 1966

Goldman, R. et al, " Information Modeling for Intrusion Report Aggregation", in DARPA Information Survivability Conference and Exposition (DISCEX II ´01), Volume I, p. 329.

IESG, "The Internet Engineering Steering Group", URL www.ietf.org.

Ilgun, K. et al., "State Transition Analysis: A Rule-Based Intrusion Detection Approach," IEEE Transactions on Software Engineering, vol. 21, no. 3, pp. 181-199, Mar., 1995.

Jianhua, Sun et al, "A Compound Intrusion Detection Model", in *Lecture Notes in Computer Science*, Volume 2836/2003, pgs. 370-381, Springer Berlin / Heidelberg.

Kahn, C. et al, "A Common Intrusion Detection Framework", artigo submetido ao Journal of Computer Security, 1998, como parte do projeto *Common Intrusion Detection Framework* (CIDF), coordenado por Dan Schnackenberg e Brian Tung, do Global Operating Systems Technology Group.

KFSensor, manual que acompanha o produto, 2004, URL http://www.keyfocus.net.

Lamport, L. "Proving the correctness of multiprocess programs". IEEE Trans. Softw. Eng., 3(2):125–143, Mar. 1977.

Leckie, T., "Bayesian Metrics for SEADS", September 3, 2002, URL http://www.cs.fsu.edu/~yasinsac/group/slides/leckie3.pdf.

Leckie, T., Yasinsac, A. "Metadata for Anomaly-Based Security Protocol Attack Deduction," IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 9, pp. 1157-1168, Sept., 2004.

LittleWood, B., Strigini, L., "Redundancy and Diversity in Security", in Computer Security – ESORICS 2004, Lecture Notes in Computer Science, vol. 3193/2004.

Maia, R. B., Soares, A. , Souza Leão, J. L., "Utilização da Lógica Difusa na Detecção da Intrusão", artigo publicado no WorkComp-SUL 2004. http://inf.unisul.br/~ines/workcomp/cd/pdfs/2732.pdf.

Maxion, R, Tan, K., "The Effects of Algorithmic Diversity on Anomaly Detector Performance", in International Conference on Dependable Systems & Networks, 2005, pgs 216-225.

Metz, C.E., "Basic Principles of ROC Analysis", in Seminars in Nuclear Medicine, 8, 283-298, 1978.

Ross, T., "Unix System Security Tools", McGraw-Hill, 1999.

Raguenet, I. F,; Maziero, C., Um Modelo de Composição de Detectores de Intrusão Heterogêneos Baseado em Conjuntos Difusos. In: VI Simpósio Brasileiro de Segurança da Informação e Sistemas, 2006, Santos SP. Anais do VI SBSeg. Porto Alegre RS : Sociedade Brasileira de Computação, 2006. p. 1-14.

Reynolds, J. C. et al, "On-line Intrusion Detection and Attack Prevention Using Diversity, Generate-and-Test, and Generalization", Proceedings of the 36th Hawaii International Conference on System Sciences, 2003.p.8.

Robbins, R., "Distributed Intrusion Detection Systems: An Introduction and Review", 2002, GSEC Practical Assignment, Version 1.4b, Option 1, URL http://www.sans.org/rr/whitepapers/detection/897.php.

SourceForge, Sourceforge.net, TCPReplay, URL http://sourceforge.net/projects/tcpreplay.

Tivoli - IBM International Technical Support Organization, "Early Experiences with Tivoli Enterprise Console 3.7", November 2000. IBM Red Book SG24-6015-00.

Ulvila, J. W. and Gaffney, Jr., J. E., "Evaluation of Intrusion Detection Systems", Journal of Research of the National Institute of Standards and Technology, Volume 108, Number 6, November - December 2003.

Vert,G., Frincke, D. A., McConnell, J. C., "A Visual Mathematical Model for Intrusion Detection", Center for Secure and Dependable Software, Department of Computer Science, University of Idaho, URL http://www.csds.uidaho.edu/ deb/vismath.pdf.

Verwoerd, T,.Hunt R., "Intrusion Detection Techniques and Approaches", University of Canterbutym New Zealand, in Computer Communications Volume 25, Issue 15 , 15 September 2002, Pages 1356-1365.

Yu-Sung Wu et al., “Collaborative Intrusion Detection (CIDS): A Framework for Accurate and Efficient IDS”, in Proceedings of the 19th annual Computer Security Applications Conference (ACSAC 2003), pgs. 234-244.

Zamboni, D. , “ Using Internal Sensors for Computer Instrusion Detection”, A Thesis Submitted to the Faculty of Purdue University, Purdue University, August 2001.

Zaraska, K. "Prelude IDS: current state and development perspectives," Technical Report (2003) for Prelude Hybrid IDS project, URL http://www.prelude-ids.org.

Apêndice A – Cálculo dos Valores-Limite *Li* e *Ls* Referentes aos Experimentos do Capítulo 5

Tabela A.1 – Cálculo do Grau de Importância

Detector	Snort	KFSensor	HoneyBOT	XRay	
Ataque					Grau de Importância
WriteSrv			√		0,125
FTP Shell DoS Vuln.		√			0,125
PortalOfDoom	√				0,125
NetStat		√	√		0,333
Telnet		√	√		0,333
IISFPDoS	√		√		0,333
PFPImageFile	√		√		0,333
FrontPage Passwordless	√		√		0,333
Sygate BackDoor	√	√			0,333
Windows Terminal Service	√	√			0,333
CyDoor detection	√	√			0,333
I-Nav ActiveX BufferOverf.	√	√			0,333
IE VersionCheck	√	√			0,333
GatorDetection+Gain	√	√		√	0,563
BackOrifice	√	√	√		0,563
BugBear	√	√	√		0,563
IISPossbileCompromise	√	√	√		0,563
MyDoom	√	√	√		0,563
Winlogo.exe Dos	√	√	√		0,563
IISRemoteCommExecution	√	√	√		0,563
RPC Port Mapper	√	√	√		0,563
AliBaBa Port Climbing	√	√	√		0,563
Personal Web Sharing	√	√	√	√	0,800
Falsos Positivos					

UDP 38293			√		0,125
TCP 3089	√				0,125
TCP 3090	√				0,125
TCP 3088	√				0,125
UDP 137		√			0,125
UDP 138		√			0,125
UDP 1027		√			0,125
UDP 2967		√			0,125
TCP 139	√	√			0,333
ICMP	√	√			0,333

Tabela A.2 – Parâmetros da Curva ROC do CIDS

μ I(e) limite	FP	VP	FN	VN	Taxa de FP	Taxa de VP
0,805	0	0	23	10	0,000	0,000
0,800	0	1	22	10	0,000	0,043
0,570	0	1	22	10	0,000	0,043
0,563	0	1	22	10	0,000	0,043
0,335	0	10	13	10	0,000	0,435
Ls = 0,330	2	20	3	8	0,200	0,870
Li = 0,130	2	20	3	8	0,200	0,870
0,125	10	23	0	0	1,000	1,000
0,000	10	23	0	0	1,000	1,000

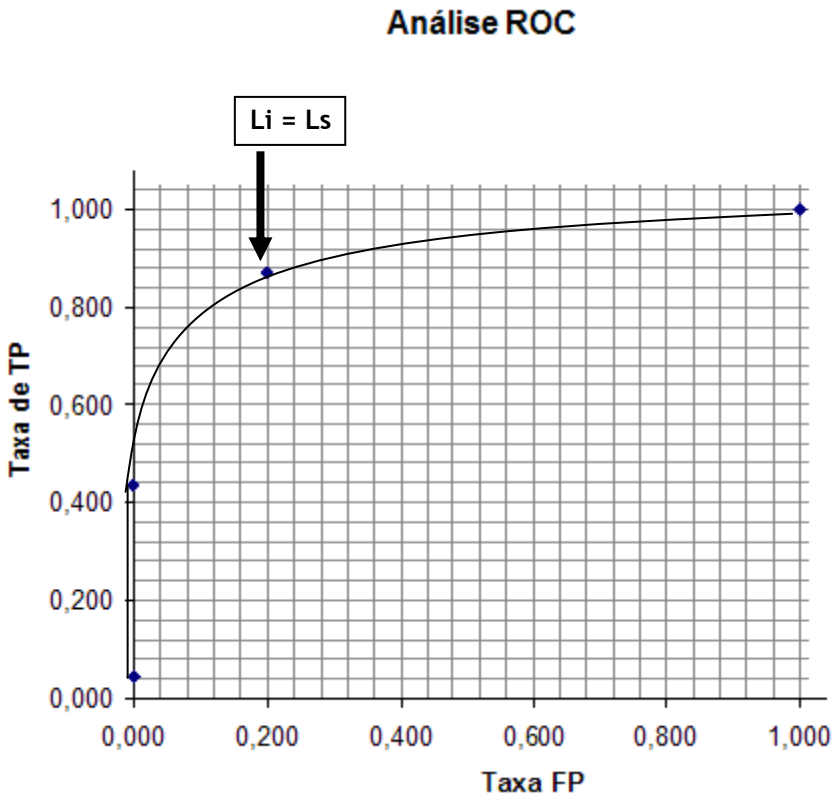

Figura A.1 – Traçado da Curva ROC usando somente valores úteis

Apêndice B – Comparativo da Taxa de Acertos entre o CIDS e cada IDS

Tabela B.1 – Cálculo do Número de Acertos de cada IDS (VP)

Detector	Snort	KFSensor	HoneyBOT	XRay
Ataque				
WriteSrv			√	
FTP Shell DoS Vuln.		√		
PortalOfDoom	√			
NetStat		√	√	
Telnet		√	√	
IISFPDoS	√		√	
PFPImageFile	√		√	
FrontPage Passwordless	√		√	
Sygate BackDoor	√	√		
Windows Terminal Service	√	√		
CyDoor detection	√	√		
I-Nav ActiveX BufferOverf.	√	√		
IE VersionCheck	√	√		
GatorDetection+Gain	√	√		√
BackOrifice	√	√	√	
BugBear	√	√	√	
IISPossbileCompromise	√	√	√	
MyDoom	√	√	√	
Winlogo.exe Dos	√	√	√	
IISRemoteCommExecution	√	√	√	
RPC Port Mapper	√	√	√	
AliBaBa Port Climbing	√	√	√	
Personal Web Sharing	√	√	√	√
Total VP	19	18	15	2

Tabela B.2 – Comparativo do Número de Acertos (VP)

IDS	VP
CIDS	20
Snort	19
KFSensor	18
HoneyBOT	15
XRay	2

Apêndice C – Comparativo entre CIDS e Snort

Tabela C.1 – IDS Snort

Detector	Snort	
Ataque		Grau de Importância
WriteSrv		0,000
FTP Shell DoS Vuln.		0,000
PortalOfDoom	√	0,800
NetStat		0,000
Telnet		0,000
IISFPDoS	√	0,800
PFPImageFile	√	0,800
FrontPage Passwordless	√	0,800
Sygate BackDoor	√	0,800
Windows Terminal Service	√	0,800
CyDoor detection	√	0,800
I-Nav ActiveX BufferOverf.	√	0,800
IE VersionCheck	√	0,800
GatorDetection+Gain	√	0,800
BackOrifice	√	0,800
BugBear	√	0,800
IISPossbileCompromise	√	0,800
MyDoom	√	0,800
Winlogo.exe Dos	√	0,800
IISRemoteCommExecution	√	0,800
RPC Port Mapper	√	0,800
AliBaBa Port Climbing	√	0,800
Personal Web Sharing	√	0,800
Falsos Positivos		
UDP 38293		0,000
TCP 3089	√	0,800

TCP 3090	√	0,800
TCP 3088	√	0,800
UDP 137		0,000
UDP 138		0,000
UDP 1027		0,000
UDP 2967		0,000
TCP 139	√	0,800
ICMP	√	0,800

Tabela C.2 – Parâmetros da Curva ROC do Snort (valores úteis)

μ I(e) limite	FP	VP	FN	VN	Taxa de FP	Taxa de VP
0,805	0	0	23	10	0,000	0,000
Li =Ls = 0,125	5	19	4	5	0,500	0,826
0,000	10	23	0	0	1,000	1,000

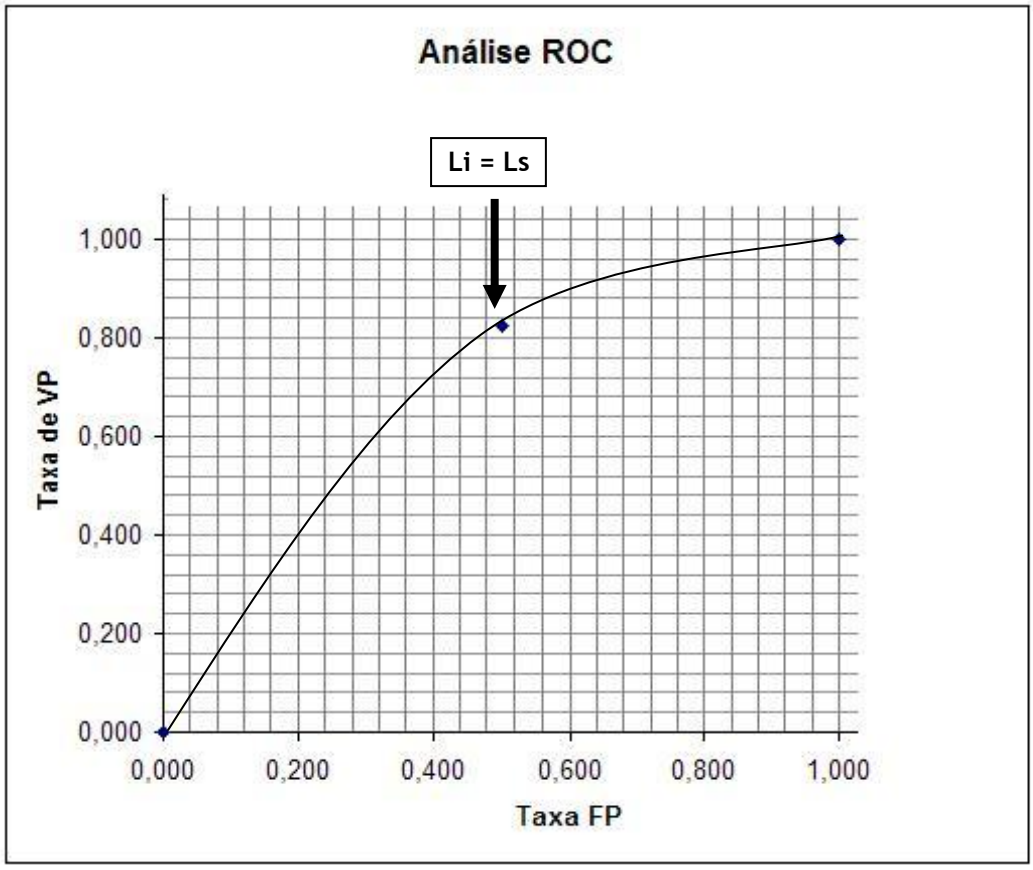

Figura C.1 – Traçado da Curva ROC - Snort

Tabela C.3 – Comparativo do número de Alarmes Falsos Snort x CIDS

Detector	FN	FP	Total
CIDS	3	2	5
Snort	4	5	9

Apêndice D – Comparativo entre CIDS e KFSensor

Tabela D.1 –KFSensor

Detector	KFSensor	
Ataque		Grau de Importância
WriteSrv		0,000
FTP Shell DoS Vuln.	√	0,800
PortalOfDoom		0,000
NetStat	√	0,800
Telnet	√	0,800
IISFPDoS		0,000
PFPImageFile		0,000
FrontPage Passwordless		0,000
Sygate BackDoor	√	0,800
Windows Terminal Service	√	0,800
CyDoor detection	√	0,800
I-Nav ActiveX BufferOverf.	√	0,800
IE VersionCheck	√	0,800
GatorDetection+Gain	√	0,800
BackOrifice	√	0,800
BugBear	√	0,800
IISPossbileCompromise	√	0,800
MyDoom	√	0,800
Winlogo.exe Dos	√	0,800
IISRemoteCommExecution	√	0,800
RPC Port Mapper	√	0,800
AliBaBa Port Climbing	√	0,800
Personal Web Sharing	√	0,800
Falsos Positivos		
UDP 38293		0,000
TCP 3089		0,000

TCP 3090		0,000
TCP 3088		0,000
UDP 137	√	0,800
UDP 138	√	0,800
UDP 1027	√	0,800
UDP 2967	√	0,800
TCP 139	√	0,800
ICMP	√	0,800

Tabela D.2 – Parâmetros da Curva ROC do KFSensor (valores úteis)

μ I(e) limite	FP	VP	FN	VN	Taxa de FP	Taxa de VP
0,805	0	0	23	10	0,000	0,000
Li = Ls = 0,800	6	18	5	4	0,600	0,783
0,000	10	23	0	0	1,000	1,000

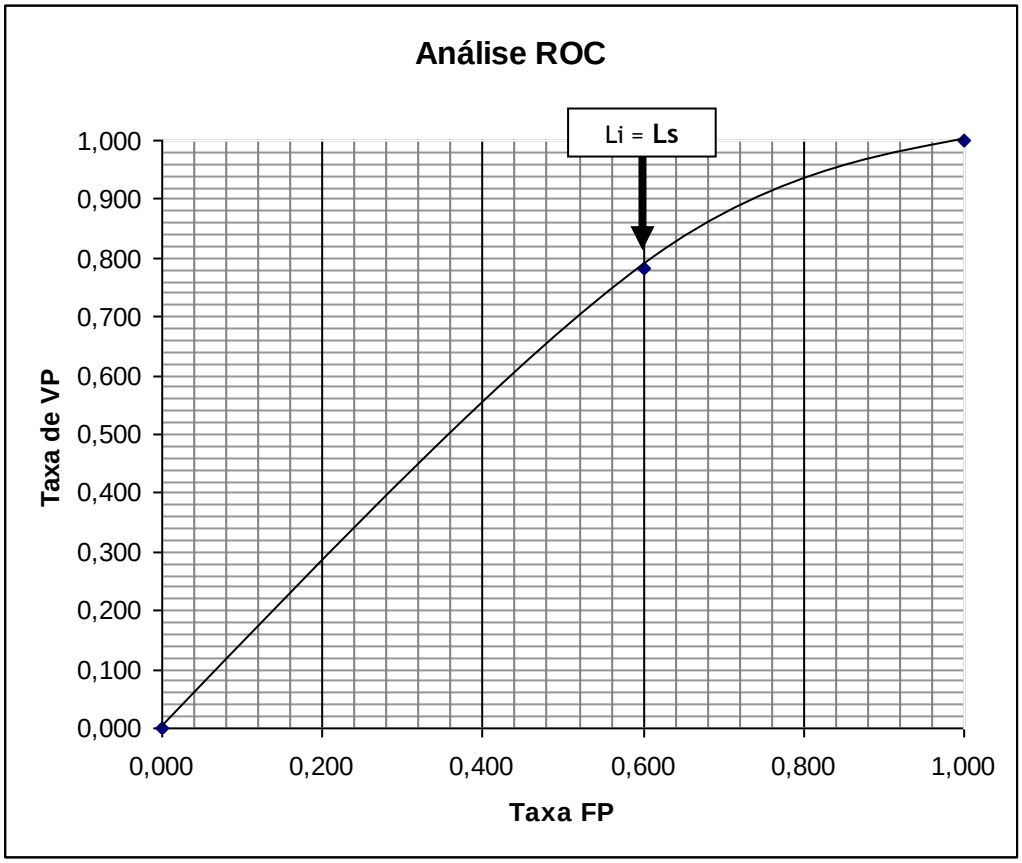

Figura D.1 – Traçado da Curva ROC - KFSensor

Tabela D.3 – Comparativo do número de Alarmes Falsos KFSensor x CIDS

Detector	FN	FP	Total
CIDS	3	2	5
KFSensor	5	6	11

Apêndice E – Comparativo entre CIDS e HoneyBOT

Tabela E.1 –HoneyBOT

Detector	HoneyBOT	
Ataque		Grau de Importância
WriteSrv	√	0,800
FTP Shell DoS Vuln.		0,000
PortalOfDoom		0,000
NetStat	√	0,800
Telnet	√	0,800
IISFPDoS	√	0,800
PFPImageFile	√	0,800
FrontPage Passwordless	√	0,800
Sygate BackDoor		0,000
Windows Terminal Service		0,000
CyDoor detection		0,000
I-Nav ActiveX BufferOverf.		0,000
IE VersionCheck		0,000
GatorDetection+Gain		0,000
BackOrifice	√	0,800
BugBear	√	0,800
IISPossbileCompromise	√	0,800
MyDoom	√	0,800
Winlogo.exe Dos	√	0,800
IISRemoteCommExecution	√	0,800
RPC Port Mapper	√	0,800
AliBaBa Port Climbing	√	0,800
Personal Web Sharing	√	0,800
Falsos Positivos		
UDP 38293	√	0,800
TCP 3089		0,000
TCP 3090		0,000

TCP 3088		0,000
UDP 137		0,000
UDP 138		0,000
UDP 1027		0,000
UDP 2967		0,000
TCP 139		0,000
ICMP		0,000

Tabela E.2 – Parâmetros da Curva ROC do HoneyBOT (valores úteis)

μ I(e) limite	FP	VP	FN	VN	Taxa de FP	Taxa de VP
0,805	0	0	23	10	0,000	0,000
Li = Ls = 0,800	1	15	8	9	0,100	0,652
0,000	10	23	0	0	1,000	1,000

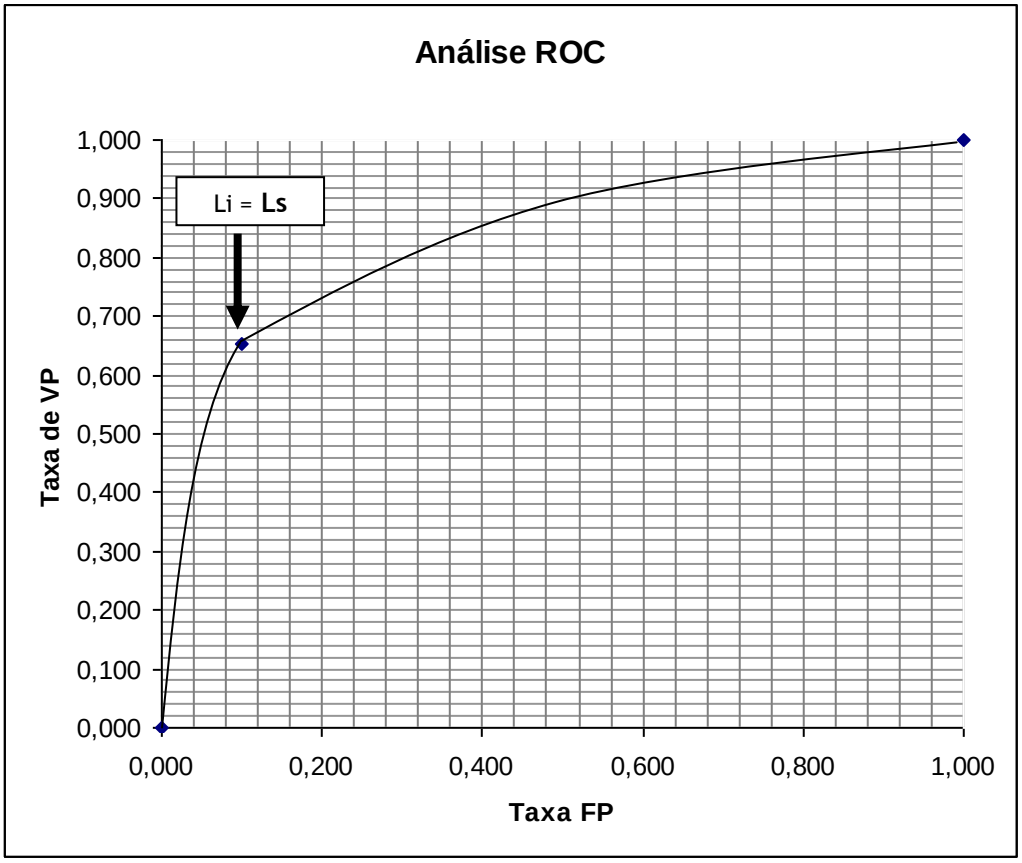

Figura E.1 – Traçado da Curva ROC - HoneyBOT

Tabela E.3 – Comparativo do número de Alarmes Falsos HoneyBOT x CIDS

Detector	FN	FP	Total Falsos
CIDS	3	2	5
HoneyBOT	8	1	9

Apêndice F – Comparativo entre CIDS e X-Ray

Tabela F.1 – IDS X-Ray

Detector	X-Ray	
Ataque		Grau de Importância
WriteSrv		0
FTP Shell DoS Vuln.		0
PortalOfDoom		0
NetStat		0
Telnet		0
IISFPDoS		0
PFPImageFile		0
FrontPage Passwordless		0
Sygate BackDoor		0
Windows Terminal Service		0
CyDoor detection		0
I-Nav ActiveX BufferOverf.		0
IE VersionCheck		0
GatorDetection+Gain	√	0,5
BackOrifice		0
BugBear		0
IISPossbileCompromise		0
MyDoom		0
Winlogo.exe Dos		0
IISRemoteCommExecution		0
RPC Port Mapper		0
AliBaBa Port Climbing		0
Personal Web Sharing	√	0,5
Falsos Positivos		
UDP 38293		0
TCP 3089		0

TCP 3090		0
TCP 3088		0
UDP 137		0
UDP 138		0
UDP 1027		0
UDP 2967		0
TCP 139		0
ICMP		0

Tabela F.2 – Parâmetros da Curva ROC do XRay (valores úteis)

μ I(e) limite	FP	VP	FN	VN	Taxa de FP	Taxa de VP
0,600	0	0	23	10	0,000	0,000
Li = Ls = 0,500	0	2	21	10	0,000	0,087
0,000	10	23	0	0	1,000	1,000

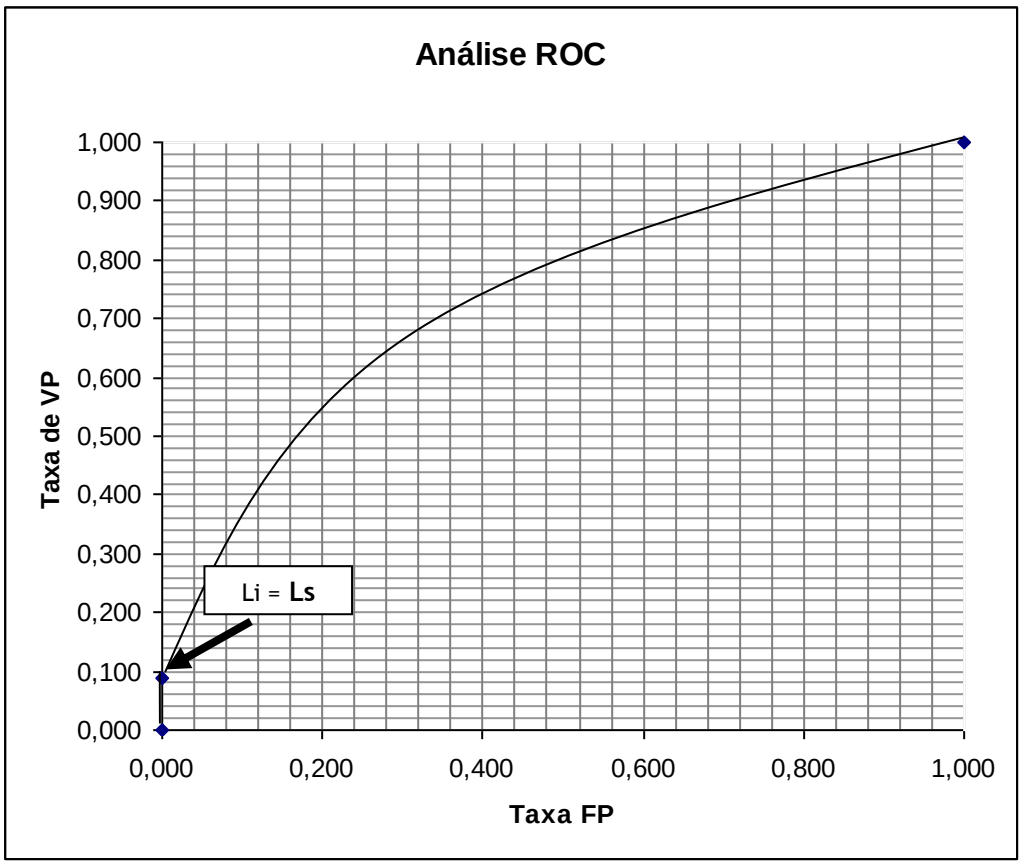

Figura F.1 – Traçado da Curva ROC para o X-Ray

Tabela F.3 – Comparativo do número de Alarmes Falsos X-Ray x CIDS

Detector	FN	FP	Total
CIDS	3	2	5
X-Ray	21	0	21