UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

UMA PROPOSTA DE APLICAÇÃO PARALELA DE TÉCNICAS DISTINTAS DE DETECÇÃO DE INTRUSÃO EM AMBIENTES DE GRID

KLEBER MAGNO MACIEL VIEIRA

FLORIANÓPOLIS-SC
2007

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

UMA PROPOSTA DE APLICAÇÃO PARALELA DE TÉCNICAS DISTINTAS DE DETECÇÃO DE INTRUSÃO EM AMBIENTES DE GRID

Dissertação apresentada ao Programa de Pós Graduação em Ciências da Computação da Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Computação

Mestrando: Kleber Magno Maciel Vieira

Orientador: Prof. Dr. Carlos Becker Westphall

Área de Concentração: Ciências da Computação

Florianópolis - SC
2007

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

KLEBER MAGNO MACIEL VIEIRA

UMA PROPOSTA DE APLICAÇÃO PARALELA DE TÉCNICAS DISTINTAS DE DETECÇÃO DE INTRUSÃO EM AMBIENTES DE GRID

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação - Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

_____

Prof. Dr. Rogério Cid Bastos
Coordenador do Programa de Pós-Graduação em Ciência da Computação

Banca Examinadora:

_____

Prof. Dr. Carlos Becker Westphall
Orientador

_____

Prof. Dr. João Bosco Mangueira Sobral

_____

Prof. Dr. Mário Antonio Ribeiro Dantas

_____

Prof. Dr. Bruno Richard Schulze

"Why does this magnificent applied science, which saves work and makes life easier, bring us little happiness? The simple answer runs: because we have not yet learned to make sensible use of it." (Albert Einstein)

"The pursuit of truth and beauty is a sphere of activity in which we are permitted to remain children all our lives."

AGRADECIMENTOS

Agradeço a Agredeço em especial ao Prof. Dr. Carlos Becker Westphall, pelas contribuições prestadas como meu orientador no desenvolvimento deste trabalho. Agradeço também a todos os professores deste programa de pós-gradução, por contribuírem na minha formação durante o período em que estive nesta instituição. Agradeço também a ajuda oferecida pelo Alexandre Schulter, Fernando Koch, Hung Ruo Han e todos os colegas do LRG.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

IDS	*Intrusion Detection System*
GIDS	*Grid Intrusion Detection System*
BD	Banco de Dados
HIDS	*Host Intrusion Detection System*
NIDS	*Network Intrusion Detection System*
DoS	*Denial Of Service*
LVQ	*Learning Vector Quantization*
API	*Application Program Interface*

RESUMO

Fornecer segurança em um sistema distribuído requer mais que autenticação de usuário através de senha ou certificado digital e sigilo na transmissão de informações. É preciso fazer o controle rigoroso das tarefas que estão sendo executadas para evitar que usuários maliciosos quebrem as políticas do grid, para que o uso de senhas roubadas possa ser identificado e, também, que ataques conhecidos sejam detectados rapidamente. Neste trabalho é apresentada uma proposta de detecção de intrusão em grid que aplica paralelamente duas técnicas distintas de detecção de intrusão sobre os dados de auditoria colhidos do middleware. É realizada uma analise de anomalia para verificar se as ações correspondem ao perfil de comportamento conhecido e uma analise de conhecimento que verifica quebras na política de segurança e padrões de ataques conhecidos. Desta forma foi proposto um padrão para descrever regras e políticas. A proposta foi validada e mensurada visando observar a sua exatidão com a quantidade de falsos positivos e falsos negativos e performance para conhecer o custo de processamento.

Palavras-Chaves: Grid, Gerencia de Redes, Detecção de Intrusão.

ABSTRACT

Providing security in a distributed system requires more than user authentication through passwords or digital certificates and confidentiality in data transmission. Rigorous control of the tasks being executed is needed in order to prevent malicious users from breaking grid policies, to identify the use of stolen passwords, and, also, to make possible the rapid detection of known attacks. In this work a solution for grid intrusion detection is presented. In its analysis of audit data collected from the middleware, two intrusion detection techniques are applied. An analysis for anomaly detection is performed to verify if user actions correspond to a known behavior profile, and a knowledge analysis is performed to verify security policy violations and known attack patterns. The proposed approach was evaluated and its performance was benchmarked to study the resulting detection accuracy in terms of false positives and false negatives, and observe the computational cost.

Keywords: Intrusion Detection System, Security, GRID

1 INTRODUÇÃO

1.1 Tema e Problema da Pesquisa

O aumento progressivo das necessidades por capacidade de processamento e recursos computacionais resultou no compartilhamento em grande escala entre instituições diferentes que possuem afinidades entre si e que podem, em momentos de ociosidade dos recursos, ajudar a suprir as necessidades de outra organização. Desta forma, em meados da década de 90, começou a surgir computação em *grid*, um conceito criado para definir um ramo de pesquisa dentro da computação distribuída.

Foster (2000), define Grid como um sistema heterogêneo com uma grande diversidade de hardware e software, distribuído geograficamente, possuindo dinamicidade onde recursos conectam e desconectam do *Grid*, ao longo do tempo, com um controle distribuído de políticas em que cada organização é responsável por seu domínio.

Neste trabalho se considera que os serviços de um grid são vulneráveis a ataques e ferramentas de autenticação e políticas de segurança são passiveis de falhas. Engenharias sociais e *exploits* abrem espaços para que invasores façam uso do sistema se passando por usuários legítimos. Desta forma, um *Grid* por sua grande capacidade de recursos disponíveis, agrega um atrativo para invasores. Devido a este fator se torna necessária à utilização de técnicas de monitoramento. Estas técnicas podem ser contempladas por IDSs (*Intrusion Detection Systems*) que investigam configurações, log, tráfego de rede e ações de usuários, procurando por comportamentos típicos de ataques (DEBAR 1999).

Para utilizar um IDS de forma eficiente em um ambiente de grid é necessário que cada nodo seja monitorado por um sistema de detecção de intrusão e em caso de ataque um alerta é enviado aos demais nodos. Desta forma, um IDS funciona de forma distribuída. Para alcançar essa distribuição dos IDS entre os nodos é necessário ter controle permissões para efetuar as manutenções e atualizações, compatibilidade com *hosts* heterogêneos e mecanismos de comunicação. Como essas características são

típicas de *Grids* **(**FOSTER 2000) se pode dividir o problema, deixando os aspectos próprios de *grid* para os *middlewares,* enfocando-se somente a detecção de intrusão.

Um ataque em um sistema de Grid pode ser silencioso para um IDS de rede uma vez que a comunicação entre os nodos pode ser criptografado e invisível para um IDS de host uma vez que ataques de grid não necessariamente alteram o S.O. do nodo ou a relação de usuários. Desta forma os IDSs tradicionais (AXELSSON, 1999) não conseguem identificar atividades suspeitas. O Fato de que a comunicação entre os nodos do grid poder ser criptografado, o que justifica pesquisa de um IDS que seja incorporado à arquitetura do Grid. Sendo assim, os sensores incorporados ao middleware teriam acesso ás mensagens não cifradas, ao contrário de sensores de rede, que teriam acesso a pacotes de rede. Invasores podem explorar uma falha no Grid para adicionar serviços não autorizados, como por exemplo, enviar relatórios contendo dados confidenciais para alguém sem autorização.

Este trabalho propõe um serviço de *grid* que foca em detectar ataques específicos desta arquitetura. A arquitetura precisa ser capaz de interceptar e coletar as mensagens do nodo do *grid* bem como dados de log para serem analisados por técnicas de detecção de intrusão (DEBAR 1999). Pretende-se nesta pesquisa fornecer as respostas para as seguintes perguntas:

Quais métodos utilizar para fornecer uma maior abrangência em detecção de intrusão em ambientes de Grid?

Para responder esta pergunta serão examinados os métodos utilizados para implementar *IDS* em *Grid*, fundamentando-se em pesquisas, existentes na literatura, realizadas sobre cenários similares. Neste trabalho será apresentada uma arquitetura abrangente, contemplando detecção baseada em comportamento e em conhecimento, em um middleware de grid o GRID-M (ROLIM 2007). Serão apresentadas umas séries de experimentos para mensurar e validar a qualidade dos serviços.
Com essa pergunta surgem outras questões:

- *O que é considerado um ataque?*

Quando se fala em ataque se imagina logo um *hacker* tentando invadir o sistema, porém, a investigação desse conceito nos revela o que o IDS deve procurar ao auditar um ambiente a procura de intruso.

- *Como garantir que as políticas de segurança estejam realmente sendo cumpridas?*

É preciso conhecer as técnicas de monitoramento e segurança de sistemas distribuídos. Uma busca na literatura revela o que é mais adequado em um ambiente distribuído.

- *Como um IDS detecta um ataque?*

Para desenvolver um sistema de detecção de intrusão é necessário conhecer as técnicas de detecção existentes na literatura e selecionar o modelo adequado para o ambiente que será trabalhado.

- *Qual é o estado da arte em detecção de intrusão em Grid?*

A literatura apresenta alguns trabalhos relacionados à detecção de intrusão em Grid. Conhecer experiências similares possibilita trabalhar na vanguarda da tecnologia.

- *Como contribuir para o incremento do estado da arte?*

Realizada a revisão da literatura de fundamentação teórica, examinando estudos correlacionados, busca-se trazer uma contribuição em relação aos trabalhos existentes, em função de um aumento da abrangência da detecção de invasão em grid.

- *Como coletar os dados para fazer uma auditoria em IDS no Grid?*

Um sistema de IDS necessita de uma entrada de dados para poder processar a análise porque coletar os dados corretos é fundamental para encontrar um ataque.

- *Como detectar um desvio de comportamento de um usuário do sistema?*

Para responder a esta questão é necessário buscar na literatura de IDS técnicas de análise de comportamento.

- *Como detectar ataques conhecidos em um ambiente de Grid?*

É preciso buscar na literatura técnica de análise de conhecimento para responder a esta questão, correlacionada com a anterior, porque com a junção das técnicas de desvio de comportamento e de conhecimento pretende-se definir as condições para um monitoramento mais abrangente dos ambientes de grid

- *Um monitoramento contínuo é muito oneroso para o Hardware?*

É necessário mensurar o desempenho da proposta para conhecer a sua viabilidade em termos de custos.

Esta pesquisa fica limitada ao estudo das técnicas de detecção de ataques em grid, não abrangendo sistemas de alertas e respostas a ataques.

1.2 Motivação

Segundo a 9ª Pesquisa nacional sobre Segurança da Informação, realizada no ano de 2005, conforme Figura 1, o percentual de ataques causados por funcionários representa 23% das ocorrências, são pessoas que conhecem o sistema e possuem algum tipo de acesso. Desta forma, não precisam se preocupar em utilizar técnicas para burlar o *firewall* e encontram poucas dificuldades para terem acesso ao sistema. Ex-funcionários e prestadores de serviços representam 8% da ameaça, ou seja, 31% dos ataques ocorrem a partir de pessoas que, de alguma forma, possuem conhecimento do sistema. Invasores externos, como *hackers* procuram fazer ataques através de cavalos de tróia, ou varredores de portas, para tornar o serviço indisponível, ou ainda, alocar

recurso para distribuir músicas e conteúdo ilegal. Desta forma, vemos que uma proteção paralela ao *firewall* é necessária. O IDS contempla um monitoramento dinâmico das ações no sistema e decide se elas representam uma ameaça ou não.

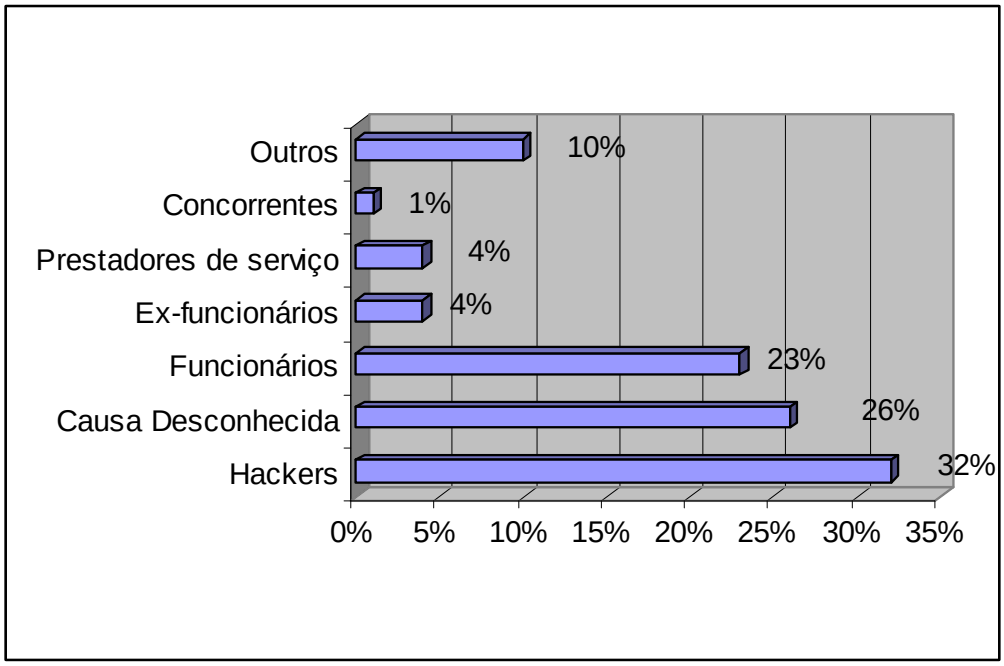


Figura 1 Principais responsáveis por ataques e invasões (Modulo 2003)

A figura 2 representa o cenário de configuração do *Grid* onde todos os nodos possuem o mesmo *middleware* sendo um o nodo principal. Os nodos A, B e D possuem o serviço de proteção *GIDS (Grid Intrusion Detection System*) que faz o monitoramento constante das atividades, detecta ataques, relatando-os ao nodo principal que poderá desabilitar um serviço, usuário ou nodo até que o administrador tome uma providência. Um dos nodos não está protegido pelo GIDS contando apenas com mecanismos básicos de segurança e vulnerável a ataques.

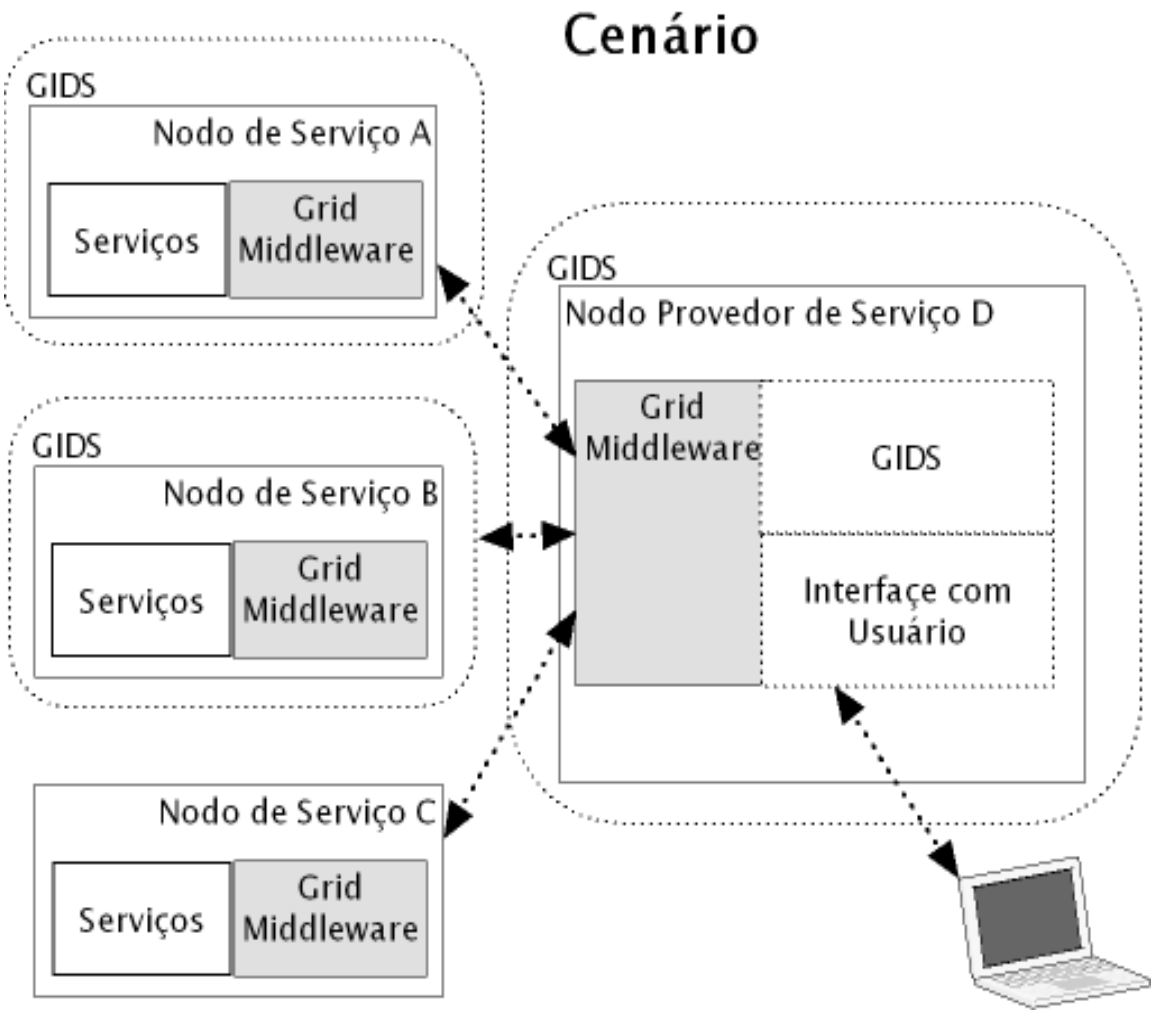


Figura 2 Cenário do GIDS no ambiente de Grid

Os cinco cenários a seguir ilustram o funcionamento do sistema de detecção de intrusão como um serviço do grid (GIDS):

- A fim de obter privilégios no sistema um intruso rouba a senha de um usuário descuidado. O GIDS realiza uma análise de comportamento, compara o comportamento atual do usuário com o seu histórico e descobre uma atividade anormal, indicando uma invasão;

- Explorando uma brecha no sistema de autenticação um usuário sem permissão pode obter acesso ao *grid*. Com um monitoramento freqüente das políticas, o GIDS descobre que um usuário não é autorizado a executar aquela ação e seu acesso é bloqueado.
- Aplicações maliciosas que ocupam o sistema e impedem seu legítimo uso. Através do monitoramento das atividades dos nodos o *GIDS* gera um alerta acusando a aplicação de abuso dos recursos;
- Uma aplicação maliciosa pode usar uma brecha na segurança do *grid* para repassar mensagens com valores fraudados. O mecanismo de detecção de intrusão analisa as mensagens e descobre que muitos valores estão fora do intervalo especificado nas políticas.

1.3 Justficativa

Analisando a literatura existente podemos perceber que autores como Choon (2003), Fang-Yie (2005), Schulter (2005), Tolba (2004), Kenny (2005) estão voltados para pesquisas em detecção de ataques em ambientes de grid. Kenny e Fang-Yie, investigam NIDS (*Network Intrusion Detection System*) para detectar ataques de rede no ambiente de Grid.

Choon (2003), indica uma arquitetura para IDS em grid. Tolba (2004), utiliza técnicas de IA para reconhecer ataques no ambiente de GRID. Shulter (2005), propõe uma arquitetura para integrar NIDS e HIDS (*Host Intrusion Detection System*) no GRID.

Esta situação permite a possibilidade de contribuição para que se possa pesquisar a captura de dados relevantes a uma detecção de intrusão e identificar ataques através desses dados. Desta forma este trabalho busca na literatura formas de descrever os ataques e técnicas de IDS e busca a possibilidade de se criar uma abrangência maior no sistema de detecção de intrusão em ambientes de grid através da composição de duas técnicas de IDS. Provendo mecanismos para aplicação paralela de técnicas distintas de detecção de intrusão em um grid.

Assim, a importância desta pesquisa é a proposta de combinação de duas técnicas distintas de detecção de intrusão. Uma das técnicas é analise de intrusão baseada em conhecimento que busca encontrar ataques típicos de grid em uma base de ataques. Para se ter sucesso com esta técnica é necessária uma forma de notação de ataques em grid e uma coleção de ataques conhecidos. A tarefa de encontrar novos ataques é custosa e requer um ambiente em produção o que justifica o uso de uma segunda técnica. A segunda técnica é analise de intrusão baseada em comportamento (também conhecida como baseada em anomalia) capaz de encontrar desvios de conduta que indicam um a taque desconhecido. Ambas as técnicas necessitam de um o sistema de auditoria que captura os dados para serem analisados. Desta forma se consegue fornecer uma maior cobertura na descoberta de ataques, aumentando a abrangência de IDS para Grid.

1.4 Organização do Trabalho

Este trabalho esta dividido da seguinte forma:

- O primeiro capítulo apresenta uma introdução ao tema, contexto e problema da pesquisa, colocando suas questões e objetivos, bem como, sua justificativa de realização e a organização do estudo.
- O segundo capítulo apresenta a motivação que impulsiona este tipo de pesquisa, a definição de ataque em ambiente de *grid*, os sistemas e técnicas de detecção de intrusão e os trabalhos relacionados que possam contribuir com o estudo proposto no capítulo e efetivado no capítulo 4.
- O terceiro capítulo é referente a esta proposta de pesquisa e a caracterização de seu objeto de estudo tendo em vista as soluções propostas no capítulo 4, em termos de sistemas mais abrangentes de detecção de intrusão em grid, a partir da composição de duas técnicas de detecção.
- O quarto capítulo apresenta a solução desenvolvida e os resultados obtidos que introduzem a idéia de composição das técnicas de análise do comportamento do usuário e a técnica do conhecimento para que se

possa obter uma maior abrangência no monitoramento em ambientes de grid;

- O quinto capítulo aporta as considerações finais, mostrando como e onde foram alcançados os objetivos, descreve as limitações da pesquisa e propõe novas direções de estudo em função da complementação do problema enfocado.

2 ATAQUE E DETECÇÃO DE INTRUSÃO

Neste capítulo serão apresentados o porquê da necessidade de um sistema seguro e quais são as perdas financeiras que a falta de segurança pode proporcionar. Será apresentada a definição de ataque e se verá que o mau uso por parte de um usuário legitimo, também, é considerado uma intrusão. Quais são as origens dos dados de auditoria de um sistema de detecção de intrusão e suas técnicas de defesa. Será vista a fundamentação teórica dos sistemas de detecção de intrusão como se classificam e quais são as técnicas de detecção. Também, será abordado o estado da arte nos sistemas de detecção de intrusão em ambientes de *Grid* e como incrementar sua eficácia.

2.1 Motivação

Segundo a pesquisa Modulo (2003), 42% das empresas brasileiras já sofreram algum tipo de ataque nos últimos seis meses, grande parte proveniente de funcionários, ainda, sim, em 26% dos casos não se consegue identificar de onde partiram os ataques.

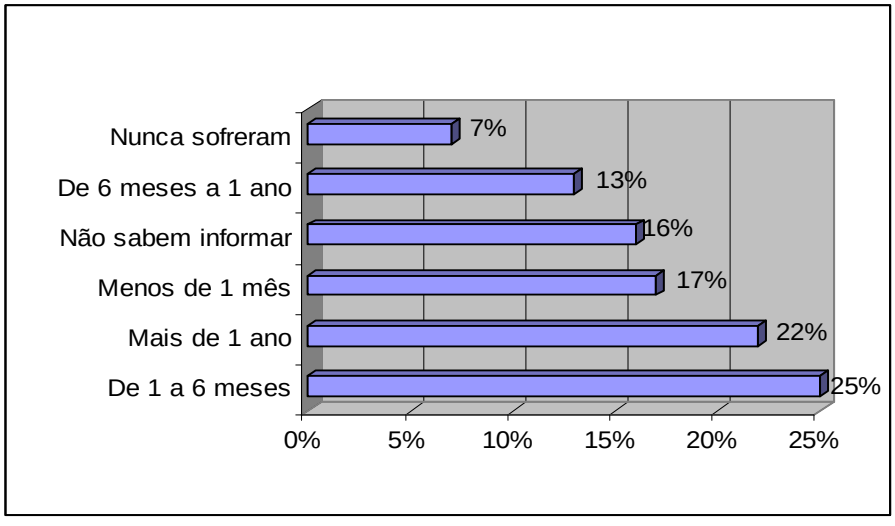


Figura 3 Empresas que já sofreram ataques (Modulo 2003)

As perdas financeiras são grandes, 35% das empresas brasileiras já sofreram perdas decorrentes de ataques, sendo que 22% afirmam que tiveram perdas de até R$ 50 mil. Este fato demanda investimentos na segurança nos sistemas. Essas empresas possuem sistemas de controle de política, *firewall* e antivírus, mas, são necessárias ferramentas que completem esses mecanismos de segurança. Seria necessário um monitoramento ostensivo de todas as ações executadas.

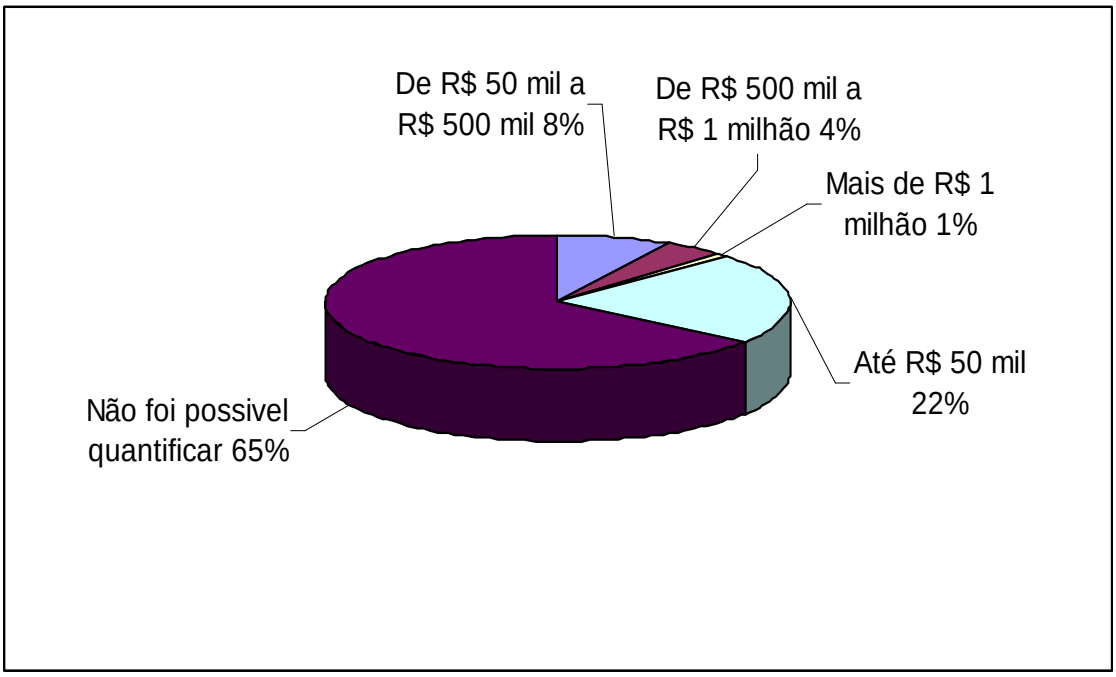


Figura 4 Perdas Financeiras (Modulo 2003)

2.2 Definição de Ataque

Considera-se que ataque é qualquer atividade maliciosa direcionada a um sistema ou serviço podendo ou não torná-lo indisponível. Exemplos de ataques são vírus, cavalos de tróia, vermes, *ping* da morte, entre outros. Porém, nem todos os ataques são tão notórios, pode-se mencionar alguns mais sutis que se não forem

monitorados só serão descobertos depois que ocorrer algum dano como, engenharia social, ou uma má utilização de privilégios.

Kendall (1999), define ataques sobre acesso ilegítimo e negação de acesso:

- Engenharia Social: Um *hacker* pode conseguir acessar um sistema enganando um usuário autorizado e o fazendo passar informações criticas para acessar um sistema ou entregar ao usuário algum cavalo de tróia para capturar informações como senha e endereços privados;
- Explorar falhas no sistema (*exploits*): Falhas de programação podem abrir espaço para que *hackers* acessem o sistema sem autorização. Problemas como *buffer overflow* pode fazer sistemas executarem código malicioso;
- Mau uso: Um usuário legítimo pode executar uma quantidade enorme de processos e consumir todos os recursos do sistema. Mesmo tendo acesso ao sistema, estas práticas fazem que se já considerado um ataque, pois, ira indisponibilizar o sistema. Além disso, alguém que possui acesso ao sistema pode utilizar seus privilégios para executar ações impróprias como acessar jogos ou conteúdo proibido pelas políticas do sistema;
- DoS: Ataques de negação de serviço é quando um intruso tenta tornar um serviço indisponível e para isso utiliza diversas técnicas como sobrecarregar o sistema enviando uma com grande quantidade de solicitações de acesso ou através de ataques coordenados.

Embora um ambiente de grid não crie novos ataques os ataques típicos de grid se diferem dos tradicionais pela maneira como são executados. Feng (2006), relata que um ataque de host ocorrem em três etapas: explorar vulnerabilidades, shell code para obter acesso ao root e atividades maliciosas e um ataque de grid se difere por não possuir a etapa de shell code. Sendo assim Feng define que um ataque de grid é idêntico ao ataque de host. Porem neste trabalho é considerado que um intruso possa corromper o middleware e atacar o grid sem ter que atacar o S.O tornando o IDS de hosts inapto a detectar ataque no ambiente de grid.

2.3 Sistema de Detecção de Intrusão

Um sistema de detecção de intrusão monitora dinamicamente as ações de um sistema ou rede e tenta definir se essas ações representam um ataque ou um uso legitimo do sistema. Alguns IDS trabalham em tempo real e podem ser usados para parar um ataque em progresso, outros trabalham de forma reativa analisando as informações do ataque para reparar danos, entender mecanismos de ataque e reduzir as possibilidades de ataques futuros (KENNY, 2005).

Embora possam existir inúmeras possibilidades na origem dos dados que um IDS utiliza, analisa-se aqui três possibilidades. Muitos sistemas de detecção costumam empregar mais de um tipo:

- Uma fonte de dados é o trafego de rede de todos os dados transmitidos através da tecnologia *ethernet* que são visíveis em um ponto, como em um *uplink* de um *switch.* Desta forma um dispositivo pode ser usado para capturar os pacotes da rede e usar esses dados para analisar a existência de intrusões;
- A segunda fonte é a auditoria de informações de sistemas operacionais. Muitos S.O. oferecem algum tipo de auditoria, porém, as informações podem ser limitadas como “Falha no sistema”;
- A terceira fonte de dados é a auditoria de *Logs*. Monitorando os serviços e as mensagens internas um IDS pode trabalhar de forma direcionada a um determinado ambiente.

2.3.1 Classificação de Sistemas de Detecção de Intrusão

Com os dados capturados é preciso fazer a análise para determinar se existe um ataque ou não.

Debar (1999), classificou os sistemas de IDS se baseando em suas características, conforme são mostradas na figura 5. Essas características podem ser funcionais ou não funcionais:

- *Detecção de Intrusão baseada em Comportamento*: A detecção de intrusão, baseada em comportamento, afirma que é possível detectar um ataque por meio de um desvio na conduta do sistema ou usuário. O modelo do comportamento válido vem de dados coletados em um período de avaliação. O sistema de IDS vai comparar com o modelo ativo, se um desvio for observado, um alarme é gerado. Em outras palavras, se a ação do sistema ou usuário não corresponder ao esperado é considerado um ataque. Entretanto é comum uma certa quantidade de alertas falsos, cujo número vai depender da técnica utilizada. Pode-se enumerar algumas vantagens de uma detecção baseada em comportamento como a descoberta de ataques em falhas, ainda, desconhecidas e uma maior automação do sistema de detecção. Como desvantagem o índice de falso positivo e falso negativo é preocupante. O comportamento do usuário costuma mudar com o tempo e é necessário fazer uma re-análise periódica de suas atividades. O sistema pode detectar um ataque como um comportamento normal. Entre as técnicas de detecção por comportamento se pode encontrar o uso da estatística como a mais comum, utilizando dados como uso e freqüência. Outra técnica, amplamente, difundida é o uso de inteligência artificial (IDRIS, 2005);
- *Método de Detecção de Intrusão baseado em Ataques Conhecidos*: utiliza técnicas sobre o conhecimento acumulado de ataques específicos e vulnerabilidades do sistema. Quando uma tentativa de invasão é encontrada um alarme é gerado. Para se obter uma boa eficiência desta técnica se requer uma constante atualização da base de conhecimento. Uma de suas vantagens é o baixo índice de falsos positivos. Existe uma grande dificuldade de conseguir informações sobre os ataques e requer uma análise cuidadosa de cada vulnerabilidade. Sistemas especialistas contém uma coleção de regras que descrevem ataques;
- *Reação Passiva*, ocorre quando um ataque é detectado e um alarme é gerado, mas, nenhuma contra medida é tomada para impedir o ataque. Em uma reação ativa, também, descrito pela literatura como IPS (intrusion prevention system), quando o sistema detecta um ataque, uma ação é tomada para impedir que o intruso tenha sucesso;

- *Fonte de dados baseada no host HIDS e auditoria em hos:* são apenas um modo de pegar informações sobre a atividade de usuários na máquina. Foi a primeira forma de IDS desenvolvida. Todas as operações do usuário são auditadas. Em um sistema UNIX todos os comandos são capturados, entretanto, é muito difícil realizar uma auditoria contínua porque muitas aplicações não fornecem uma estrutura para a captura e o armazenamento de eventos;
- *Fonte de dados baseada na rede NIDS*, ataques de rede como enganar um DNS, varredura de portas, ataques de DoS, podem ser detectados por HIDS. Também, ferramentas como *sniffer* podem ser utilizadas para procurar ataques. Além disso, um grande número de ataques contra servidores podem ser detectados analisando a carga de dados e procurando por comandos suspeitos;

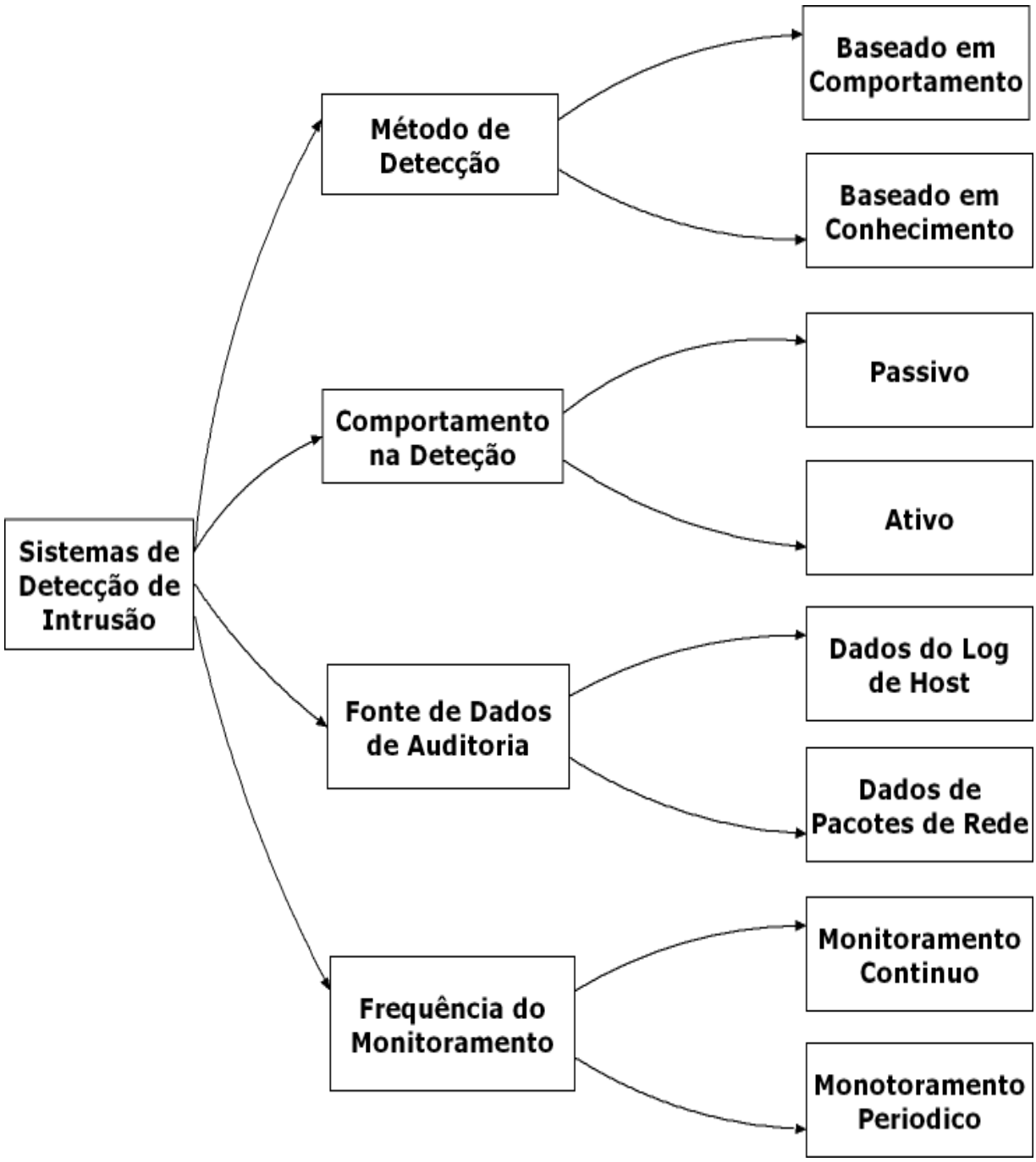


Figura 5 Características de IDS (Debar 1999)

- Como característica não funcional se tem a forma de monitoramento que pode ser contínuo, onde a vigilância é feita em tempo real, ou o monitoramento periódico, onde os *logs* e o estado do sistema são analisados segundo sua freqüência em um período de tempo.

Viu-se as formas de detecção que podem ser baseadas em análise do comportamento ou em ataques conhecidos e, também, a resposta do sistema de IDS que pode tomar uma providência ao ataque, fazendo um bloqueio ou, gerando um alerta. Desta forma, respondemos a pergunta *"Como um IDS detecta um ataque?"*

2.4 Detecção de Intrusão em Grid

Foi realizada uma pesquisa com os trabalhos relacionados com a detecção de intrusão em Grid que representam o estado da arte até o presente momento. Procurou-se destacar quais foram as técnicas utilizadas para se realizar a detecção de intrusão alem das origens dos dados.

2.4.1 Tolba

Tolba (2005), propõe o GIDA (Grid Intrusion Detection Architecture) uma arquitetura de detecção de intrusão em grid dividida em duas partes principais. A primeira é representa o IDA (Intrusion Detection Agent) responsável por coletar informações relevantes a descoberta de ataques. A segunda parte se constitui do IDS (Intrusion Detection Service) serviço responsável por analisar as informações coletadas. O qual utiliza uma rede neural do tipo LVQ (Learning Vector Quantization) para detectar desvios de comportamento que podem caracterizar a ocorrência de uma intrusão. Para validar a arquitetura foi realizados testes em dois estágios. Começando pela coleta de dados em simuladores de grid que representavam usuários em diferentes comportamentos. O autor não utilizou dados provenientes do tráfego de rede e justifica

que pacotes de rede não possuem dados de auto nível como nome de usuário, sendo que as informações realmente interessantes estão no log, no middleware de grade. O segundo estagio foi analisar os dados utilizando uma rede neural do tipo LVQ para encontrar anomalias.

2.4.2 Fang-Yie Leu

Fang-Yie Leu (2005), propõe um sistema de detecção de intrusão de rede. O autor trata de ataques dos tipos lógico, *float* e reflexivo, os quais possuem o intuito de sobrecarregar o sistema criando um grande tráfego na rede. Ficando de fora a detecção com dados de host e ataques próprios de grid. Existe uma preocupação por distribuir o processamento da detecção entre os nodos sugerindo técnicas de balanceamento de carga. E desta forma justifica o uso do grid. A arquitetura propõe uma captura dos pacotes que devem ser encaminhados para uma pilha onde aguardam o escalonamento para a detecção. Nos experimentos são comparados os tempos de detecção utilizando um nodo com o tempo para detectar utilizando o grid.

2.4.3 Fang-Yie Leu

Choon (2003), discute a detecção de intrusão em ambientes de grid sugerindo a arquitetura de um framework conceitual. O framework proposto é dividido de pequenos módulos. O agente do GIDS é um pequeno serviço residente nos nodos do grid para coletar os dados, fazer auditoria e se comunicar com o servidor. O servidor do GIDS representa o núcleo, responsável por processar os dados da auditoria, iniciar o serviço de análise para detectar a intrusão. O Gerenciador do GIDS possui uma interface para fazer a comunicação com o administrador da grade, monitora os servidores de GIDS e gerencia as políticas da grade para que sejam cumpridas. O autor índica a arquitetura para que seja implementado um protótipo. Nenhum experimento é realizado bem como maiores explicações de que dados devem ser auditados e detalhes sobre as técnicas de analise.

2.4.4 Schulter

Schulter (2006), aponta como deficientes os trabalhos de Choon e Tolba por usar apenas técnicas que detectam ataques por anomalia e deixando descoberto uma ampla variedade ataques. O problema de detecção de intrusão em grid é apresentado como deficiente em cobrir ataques de rede e host. Desta forma Schulter define como necessário em um IDS de grid escalabilidade, compatibilidade com o grid, cobertura de detecção de ataques de grid, acesso não autorizado, ataques de host e rede. É proposta uma arquitetura chamada de GIDS (Grid Intrusion Detection Sistem) um IDS de grid que integra IDS de baixo nível, detecção de intrusão em rede e detecção de intrusão em host. Cada nodo possuI um IDS de rede capturando dados de baixo nível e um IDS de host examinando as atividades do nodo, buscando evidencias de ataques, sendo que um IDS de grid é responsável por receber relatórios de cada um desses recursos como apresentado na figura 6 onde é apresenta uma ilustração da arquitetura do GIDS. Os IDS de baixo nível enviam alertas para o GIDS e como os esses IDS são heterogêneos enviam alertas em diferentes formatos desta forma Schulter usa o IDMEF (MCCUBBIN) para integrar as mensagens. Porem a literatura sobre ataques de HIDS e NIDS é bastante avançada essas técnicas não tem como foco cobrir ataques de Grid pois não possuem acesso as informações necessárias.

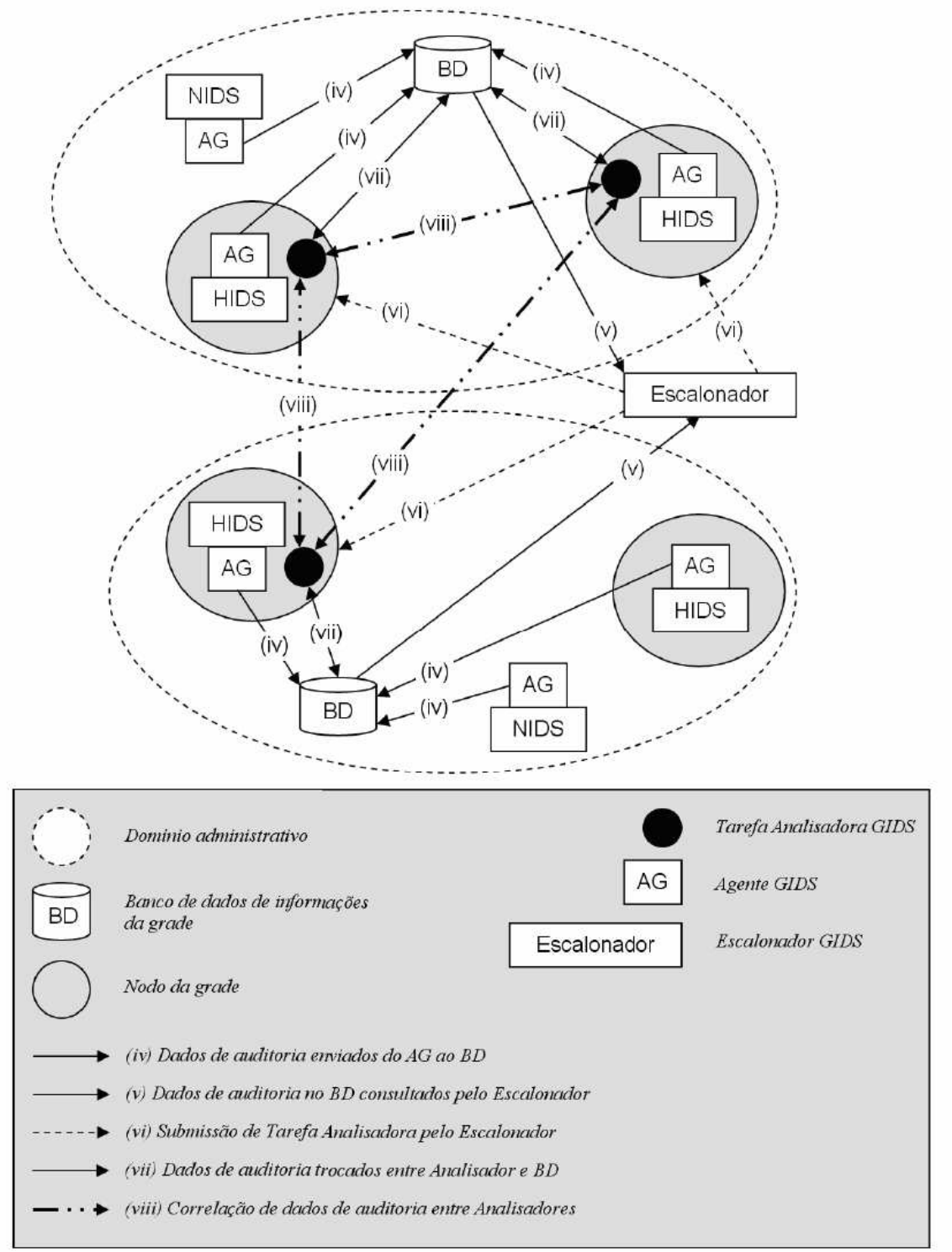


Figura 6 Arquitetura GIDS proposta por Schulter 2006

2.4.5 Kenny

Kenny (2005), propõe o uso de um IDS tradicional de rede (snort) para dar suporte a segurança de cada nodo. O autor descreve um modulo para acoplar esse sistema ao ambiente de grid. Utiliza uma base com assinaturas de ataques de rede e utiliza dados reais para fazer a validação do ambiente.

2.4.6 Feng

Feng(2006) , propõe uma arquitetura chamada GHIDS (Grid Host Intrusion Detection System) e defende que a analise de dados de *Host* é vital para proteger o compartilhamento de recursos em um ambiente de grid. O autor afirma que os IDS de *host* não são adequados para o ambiente de grid. A arquitetura proposta usa dados de host para detectar intrusões no grid. Utilizando dados do sistema operacional o sistema procura por ataques de *Host* como acesso de funções sem permissão. A validação é realizada comparando o desempenho da proposta com um IDS de *host* comercial.

A tabela 1, representa o resumo dos trabalhos relacionados e esta organizado por autor e característica. Foi considerada a técnica de captura como HIDS ou NIDS, técnica de análise com base de conhecimento ou com base de comportamento, além da origem dos dados que podem ser dados de rede, host ou do ambiente do grid, e, também, se houve validação. Choon (2003), propõe uma arquitetura e não faz validação.

Fang-Yei Leu (2005) e Kenny (2005), utilizam dados de ataques de rede em ambiente de Grid, porém, suas técnicas não dão suporte a detectar ataques típicos de Grid, nem conseguem capturar dados de alto nível.

Feng(2006), integra um IDS de host em ambiente de grid, relaciona ataques típicos de host não abrangindo ataques contra o middleware de grid.

Tolba (2005) e Schulter (2006) trabalham com IDS de Host, assim, analisam dados de alto nível e tratam de ataques próprios de Grid, ambos utilizam técnicas de análise baseada no comportamento.

Autor	HIDS	NIDS	Dados exclusivos de grid	Base de conhecimento	Base de comportamento	Validação
Tolba	Sim	Não	Sim	Não	Sim usando IA	Sim
Schulter	Sim	Sim	Não	Não	Sim usando IA	Sim
Choon	Não	Sim	N/A	Não	Não	Não
Kenny	Não	Sim	Não	Sim	Não	Sim
Fang-Yie	Não	Sim	Não	Sim	Não	Sim
Feng	Sim	Não	Não	Sim	Não	Sim

Tabela 1 Relação das características dos trabalhos correlatos

Desta forma, verifica-se na literatura uma problematização diferenciada em relação a presente pesquisa, especialmente, no que se refere a abrangência de IDS para Grid investigando uma combinação de duas técnicas distintas de análise e fornecendo uma maior cobertura na descoberta de ataques.

2.6 O projeto

Para alcançar os objetivos deste trabalho, tem-se como requisito a coleta dos dados inerentes à auditoria do ambiente um coletor de eventos que deve ser adicionado ao middleware de grid. Esse coletor será responsável por capturar todas as trocas de mensagens inerentes ao seu nodo e repassá-las ao motor de IDS. Conhecendo esses eventos o núcleo de detecção de intrusão solicita que os módulos devam proceder a sua técnica de análise. Os elementos do problema se referem a:

- Coletar dados inerentes à troca de mensagens no ambiente;
- Elaborar uma arquitetura de interceptação, análise e alerta;
- Solução em um ambiente controlado;

Para resolver esse problema precisaremos de:

- Middleware de grid;
- Permitir a estrutura de coleta de dados;
- Criar um protótipo da proposta;
- Executar o ambiente;

2.7 Sumário do Capítulo

Este capítulo respondeu a questão “O que é considerado ataque?” mostrando os conceitos ataques existentes na literatura. A pergunta “Como um IDS detecta um ataque?” foi respondida fazendo um estudo das técnicas de detecção de intrusão onde encontrarmos as formas de captura de dados que podem ser provenientes da rede ou de dados do sistema. Com esses dados se aplica um método de descoberta que pode ser baseado em ataques já conhecidos ou baseado em comportamento do usuário. A pergunta “Qual é o estado da arte em IDS de Grid?” foi respondida com uma pesquisa e uma análise comparativa foi elaborada para determinar “Como contribuir para o incremento do estado da arte em IDS de Grid?”.

3 PROPOSTA E CARACTERIZAÇÃO DO ESTUDO

Neste capitulo será visto que um IDS de *host* e rede não é suficiente para detectar intrusão em um ambiente de grid é importante que o IDS trabalhe com dados do middleware de grid. As aplicações de Grid se distinguem das demais por possuir características muito próprias como se distribuir entre nodos de ambientes distintos e a identificação do usuário não é a mesma do S.O. (FOSTER, 1998).

Um ataque de Grid também se diferencia dos ataques de host onde o alvo é geralmente o S.O. Neste capítulo será apresentada uma proposta de sistema de detecção de intrusão para Grid. Será definida a arquitetura de um serviço de IDS de Grid, será apresentada a forma como os dados serão coletados e como os nodos interagem entre si para garantir a segurança do conjunto.

Este capítulo esta estruturado da seguinte forma:

- Seção 3.1: apresenta o cenário com um IDS trabalhando em ambiente de grid;
- Seção 3.2: apresenta os componentes do IDS e quais são os seus requisitos;
- Seção 3.3: apresenta o sistema de auditoria, respondendo a questão colocada na introdução "*Como coletar os dados para fazer uma auditoria em IDS no Grid?*";
- Seção 3.4: apresenta a técnica de detecção baseada em comportamento respondendo a pergunta "*Como detectar um desvio de comportamento de um usuário do sistema?*";
- Seção 3.5: será apresentada a forma de detectar ataques conhecidos e uma notação de regra respondendo a pergunta "*Como detectar ataques conhecidos em um ambiente de Grid?*";
- Seção 3.6: apresenta a justificativa da união das duas técnicas;
- A seção 3.7: apresenta a conclusão do capítulo.

3.1 Arquitetura

A detecção de intrusão e o sistema de alerta podem ser distribuídos, cumprindo as necessidades de IDS em Grid. Nesta proposta de arquitetura cada nodo no grid participa da detecção de intrusão e alerta.

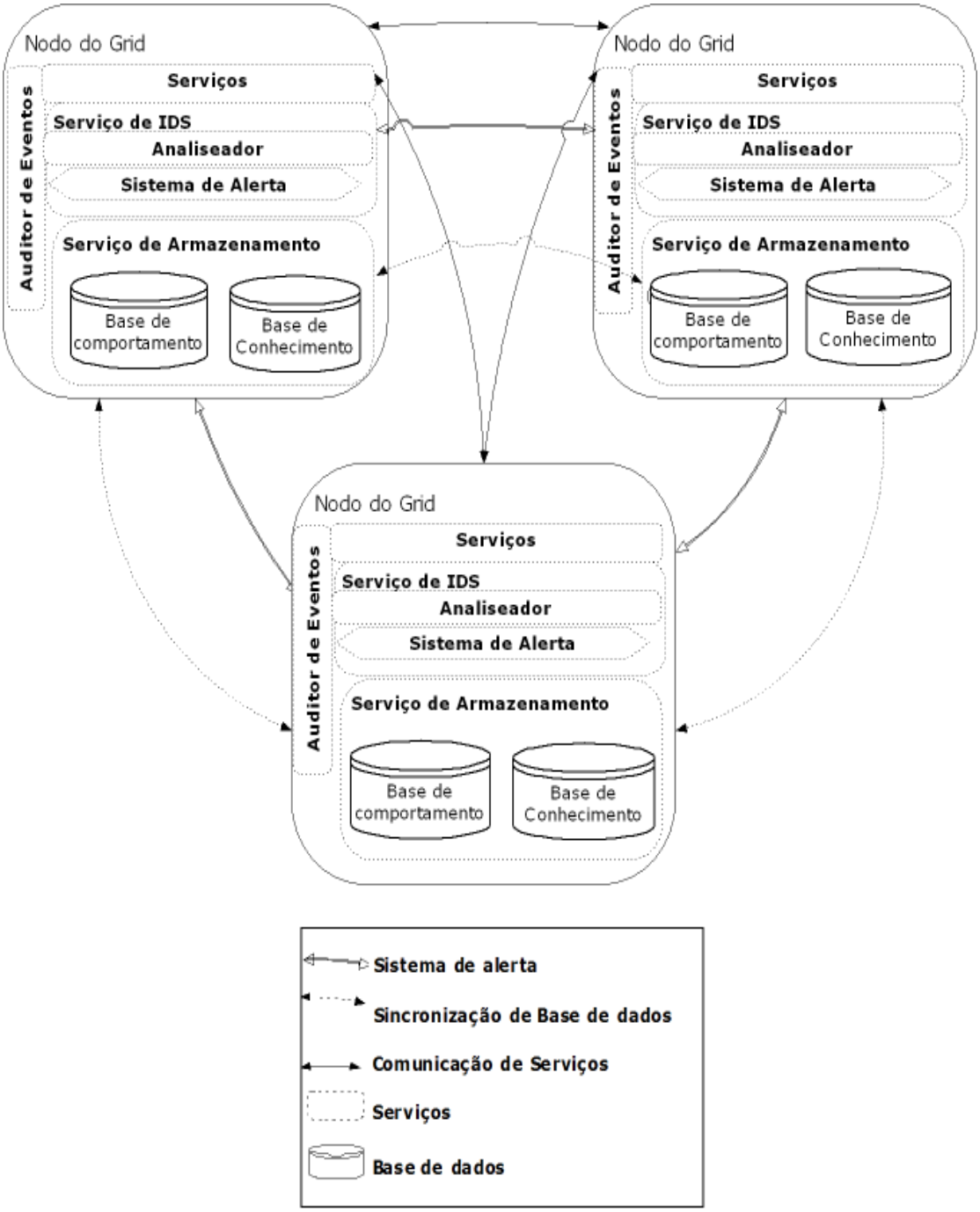


Figura 7 Arquitetura IDS em Grid

Cada nodo é responsável por detectar seus ataques e avisar aos outros nodos quando encontrar uma intrusão local para que se seja tomada uma atitude. Estes IDSs individuais vão coletivamente participar na detecção global. A figura 7, apresenta de uma forma geral o compartilhamento de informações pelo serviço de detecção de intrusão em Grid e os elementos que compõe esta arquitetura. É importante ressaltar que se trata apenas de taques típicos de Grid. A estrutura é formada pelos seguintes elementos:

- *Nodo* é a unidade do Grid composto pelo *middleware* que tem como função homogeneizar o ambiente possibilitando a comunicação entre os dispositivos heterogêneos, fazer o controle de acesso e políticas e fornecer um ambiente de suporte aos serviços;
- *Serviço* utiliza o ambiente do *middleware* para executar sua função, quando é necessário enviar uma mensagem para outro nodo, é através do *middleware* que a comunicação é realizada;
- *Auditor de Eventos* é a peça chave no sistema ele é responsável por capturar dados de varias fontes como o sistema de log, mensagens de serviços e as mensagens entre os nodos.
- *Serviço de IDS* analisa os dados capturados pelo auditor e aplica as técnicas de detecção baseada em comportamento e detecção baseada em conhecimento sendo visto com mais detalhes em 3.3 e 3.4. Em alguma situação de ataque utiliza o middleware para enviar um alerta aos outros nodos. Desta forma, é possível bloquear um ataque em todo o Grid. O middleware, também, é responsável pela sincronização de dados como ataques conhecidos e base de comportamento.
- *Serviço de Armazenamento* contem os dados necessários para se fazer a análise do ataque. É importante que todos os nodos possuam o acesso aos mesmos dados e o ambiente de Grid é responsável pela virtualização do ambiente homogêneo de uma forma transparente tem-se apenas uma única base.

Como foi apresentado no capítulo 2, neste trabalho será proposto um IDS próprio de Grid, desta forma, trata-se de detectar ataques nesse ambiente com um

serviço distribuído entre os nodos e uma sincronização dos dados fornecida pelo middleware.

3.2 Apresentação do IDSService

O IDSService foi concebido para incrementar a segurança do Grid e para isto utiliza duas técnicas, ou métodos, de detecção de intrusão. A primeira é baseada em comportamento e se refere às ações do usuário em relação ao seu comportamento normal e a segunda técnica baseada em conhecimento de ataques anteriores. Ambos os métodos serão vistos mais adiante com maiores detalhes.

As interações entre o Grid e os mecanismos de IDS são representadas na figura 8. Os dados auditados são enviados (i) para o núcleo do IDSService que inicia o processo de análise de comportamento (ii), utilizando os recursos da inteligência artificial para detectar se ocorre, ou não, desvio de conduta.

Com uma base de dados com o histórico de perfil (iii), o analisador consegue determinar a distância desse comportamento com comportamento usual e (iv) envia essa informação para o núcleo do IDSService. O analisador de regras recebe o pacote de auditoria (v) e verifica nas políticas se alguma regra na base (vi) está sendo quebrada. O resultado da análise é retornado para o núcleo do IDSService (vii). Com esses dados de retorno (iv, vii), o IDS compila as probabilidades da ação representar um ataque e caso ocorra uma incidência, um alerta é enviado aos outros nodos.

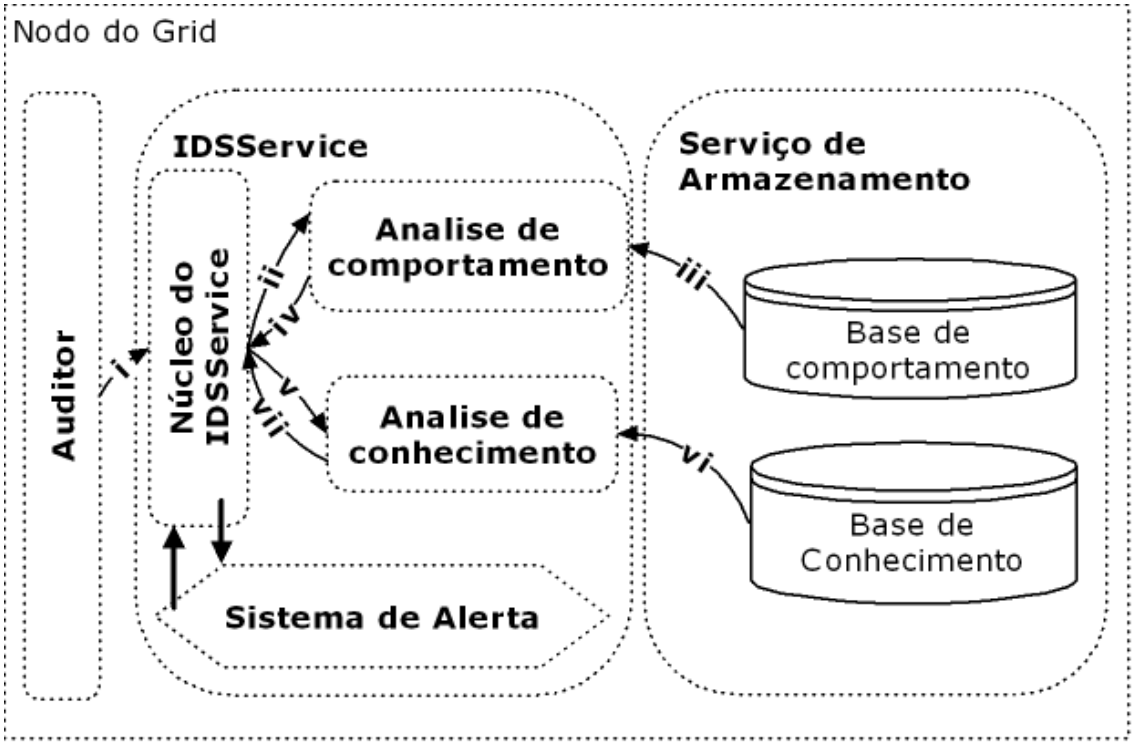


Figura 8 Modelo funcional IDSService

3.3 Auditor de Eventos

Para realizar a analise de intrusão são necessários dados que informem o comportamento do ambiente e suas trocas de mensagem. Desta forma é proposto um auditor de eventos que é composto por dois componentes que permitem o monitoramento de dados que são acessados pelos analisadores. O primeiro componente monitora a troca de mensagem entre os nodos. Mesmo capturando informações de comunicação esse IDS não se caracteriza como NIDS, pois não abranger dados da rede apenas informações do nodo.

O segundo componente monitora o sistema de log do middleware. A cada ação do nodo um log é criado contendo informações como o tipo de ação (exemplos: erro, alerta ou aviso), o evento que gerou o log e a mensagem. Com esses dados é possível analisar se esta ocorrendo uma intrusão. Este trabalho se diferencia dos demais por

identificar e capturar dados do nodo do Grid. Os demais trabalhos relacionados utilizam dados de rede e do host para fazer a detecção. Apenas Tolba (2005) indica o uso de *log* porem não entra em maiores detalhes.

3.4 Análise de Comportamento

Para resolver o problema de detecção de intrusão baseada em comportamento , pode-se encontrar inúmeros métodos como mineração de dados, redes neurais artificiais e sistemas imunológicos artificiais. Neste trabalho foi focado o uso de rede neural artificial do tipo *feed-foward* por que em contraste com os tradicionais métodos sugeridos temos um rápido processamento de informações, uma grande habilidade, a tolerância e o auto aprendizado, todas essas vantagens ajudam a superar os problemas dos IDS (WANG ET AL , 2004).

A rede neural do tipo *feed-foward* é amplamente utilizada para reconhecer padrões de comportamento em diversas áreas (HAYKIN, 2001). No caso de sistemas de detecção de intrusão é necessário reconhecer se o comportamento representa o padrão esperado, caracterizando um uso legitimo, ou, se caracteriza um ataque. Alguns fatores devem ser considerados ao utilizar uma rede deste tipo como:

- O numero de camadas ocultas. De acordo com Kolmogorov (APUD RUMELHART ET AL, 1986) uma rede de três camadas pode expressar o mapeamento de n por m, ou seja, uma camada intermediária é suficiente;
- O tamanho da camada de entrada e de saída. O numero de elementos na entrada vai representar os valores referentes as características que representam o comportamento. A dimensão da saída representa as possibilidades da classificação. No nosso caso é comportamento normal ou ataque;
- Número de neurônios na camada oculta. Ao contrario do tamanho da camada de entrada não existe uma forma determinística para se encontrar esse valor. A medida que os experimentos são realizados diversos valores são testados.

O treinamento é a peça chave da rede *feed-forward*. Somente após ter uma rede treinada corretamente se pode ter a detecção funcionando de forma eficiente. Dado um conjunto de exemplos de intrusão de acordo com o algoritmo de retopropagação, a rede neural ira aprender a identificá-los. Porém, neste trabalho é focado a identificação de padrões comportamentais de usuários e o reconhecimento de desvios de comportamento. Utilizando esta estratégia se tem uma maior cobertura dos ataques desconhecidos, uma vez que o ambiente de Grid, até este momento, é considerado um ambiente recente onde poucos ataques foram identificados.

3.5 Análise de Conhecimento

Detecção baseada no conhecimento é a técnica mais usada em detecção de intrusão porque possui um baixo índice de alarmes falsos, um índice de muito elevado de acertos, porém, não pode detectar padrões de ataques desconhecidos. É baseado em regras que monitoram um fluxo de eventos e agem como um algoritmo que procura uma característica maliciosa em um comportamento. Usando um sistema especializado é possível descrever o comportamento malicioso em uma regra. Esta escolha facilita a evolução do componente porque uma nova regra pode ser adicionada sem mudar as existentes, ao contrario do que acontece com um sistema de análise de comportamento, em que não é possível adicionar uma nova característica no comportamento sem alterar todo aprendizado anterior.

A produção de regras é o elemento chave desta técnica através do qual o sistema especialista é capaz de reconhecer um ataque no ambiente. A elaboração de regras consiste em definir uma condição que representa o ataque.

3.5.1 Sistema de Regras

Para resolver o problema da representação de regras é preciso uma forma de notação que possua algumas características: suportar uma coleção de regras que represente ataque e violação de políticas, ser de fácil administração e compreensão para configurar e ser transparente para o usuário ate que uma detecção seja encontrada.

Esta seção descreve uma estrutura de regra e como ela se aplica sendo capaz de representar os elementos auditados do Grid e seus ataques. O sistema de regras adotado neste projeto é baseado em XML (BRAY, 2005). Especificar as regras neste formato permite ao administrador descrever os elementos e seus valores quando caracterizam uma intrusão.

A estrutura de regras é composta pelos seguintes componentes:

- *Alert***:** Inicia o inicio de uma regra e possui o atributo *source* que representa a origem do dado auditado. O atributo pode conter o valor *log* quando se quer utiliza dados das anotações do sistema ou *message* que são as mensagens que o nodo recebe e envia;
- *identification*: Representa um valor único com a identificação da regra;
- *RuleInformation***:** Responsável por uma informação descritiva da regra ou descrição do evento intrusivo podendo informar os sistemas afetados;
- *Version:* Representa a versão da regra. Auxilia no controle de alterações;
- *Name:* Representa o nome da regra;
- *Credits:* Identifica o autor ou a origem da regra;
- *Element:* representa o elemento que pertence ao pacote de auditoria. Possui um atributo *value* que se refere ao nome do elemento e um valor para caracterizar a ocorrência do ataque. Uma regra pode ter mais de um *Element* para restringir a ocorrência;
- *Comprises:* representa uma função lógica que proíbe um valor especifico. Possui um atributo com o elemento onde sera verificados o valor, o conteúdo proibido e a ocorrência.
- *Range:* representa uma função lógica que proíbe um intervalo de dados. Possui um atributo com nome do elemento que deve ter o valor verificado e possui duas sub-tag;
    - *MoreThan***:** possui um atributo que define o limite superior do intervalo de dados;
    - *LessThan:* possui um atributo que define o limite inferior do intervalo de dados;

Dessa forma, uma regra sempre começa por uma tag *Alert* com o atributo da origem dos dados. Seguida de um ou mais tag *Element* que possui um valor necessário, não suficiente para indicar o ataque. Por ultimo, a função lógica que pode ser *Comprise* ou *Range*. Para uma maior compreensão das regras a tabela 4 , representa cinco exemplos:

Uma notação foi elaborada para definir regras e ataques conhecidos.

- A regra número 1 detecta um ataque nos dados de auditoria provenientes da comunicação. Como regra necessária o elemento *service* precisa ter o valor *WEB*. O conteúdo proibido é um texto que contenha XXX em qualquer lugar da mensagem;
- A regra 2 verifica se o valore do elemento *example* esta dentro do intervalo aberto de dois a dez;
- A regras 3 verifica se o intervalo de dados do elemento *priority* é inferior a 6 no pacote de auditoria da comunicação;
- A regra numero 5 verifica os dados proveniente do log que possui os elementos necessários porém não suficiente: *Event* com valor HTTPSending, *Element* com valor http-server e o elemento *Message* com o valor proibido Games.

A políticas de valores mínimos e máximos podem revelar aplicações com erro semântico como dados inválidos ou um ataques de DoS. Os valores proibidos podem ser conteúdo impróprio ou scripts maliciosos sendo transmitidos pela rede. Dessa forma pode-se criar inúmeras regras protegendo o ambiente com grande rigor.

Id	Regra
01	<alert source=”communication”> <identification>001</identification> <ruleInformation> This rule verify the inappropriate use </ruleInformation> <version>1.0.0</version> <name>inappropriateUse</name> <credits>LRG</credits> <element value="service">web</element> <comprises>XXX</comprises> </alert>
02	<alert source=”communication”> <identification>002</identification> <ruleInformation>False range in example </ruleInformation> <version>1.0.1</version> <name>exampleRange</name> <credits>LRG</credits> <range value="example"> <morethan value=”2”/> <lessthan value=”10”/> </range> </alert>

03	<alert source=”communication”> <identification>003</identification> <ruleInformation>This rule verify priority range </ruleInformation> <version>2.5.0</version> <name>priorityRange</name> <credits>LRG</credits> <range value="priority"> <morethan value=”6”/> </range> </alert>
04	<alert source=”log”> <identification>004</identification> <ruleInformation>This rule verify game use </ruleInformation> <version>1.2.3</version> <name>gameUse</name> <credits>LRG</credits> <element value="Event">HTTPSending</element> <element value="Element">htttp-server</element> <comprise value="Message">Games</comprise> </alert>

Tabela 2 Exemplo de Regras

3.6 Ampliação da cobertura de ataques com a integração de técnicas

As duas técnicas de detecção de intrusão possuem características distintas. Enquanto a detecção de intrusão baseada em conhecimento possui um alto índice de acerto para ataques conhecidos, sua deficiência em novos ataques é complementada com a técnica de detecção baseada em comportamento cuja capacidade em encontrar desvios de conduta é significativamente importante em ambientes onde não se deve aceitar abusos de privilégios e as ações precisam ser minuciosamente investigadas. Como o volume de dados em um ambiente de grid costuma ser altíssimo não cabe ao administrador observar as ações individuais de cada usuário, mas apenas os alertas que o sistema de IDS fornece. Desta forma, as duas técnicas contemplam ataques distintos com uma pequena quantia em comum. Como ilustrado na figura 9.

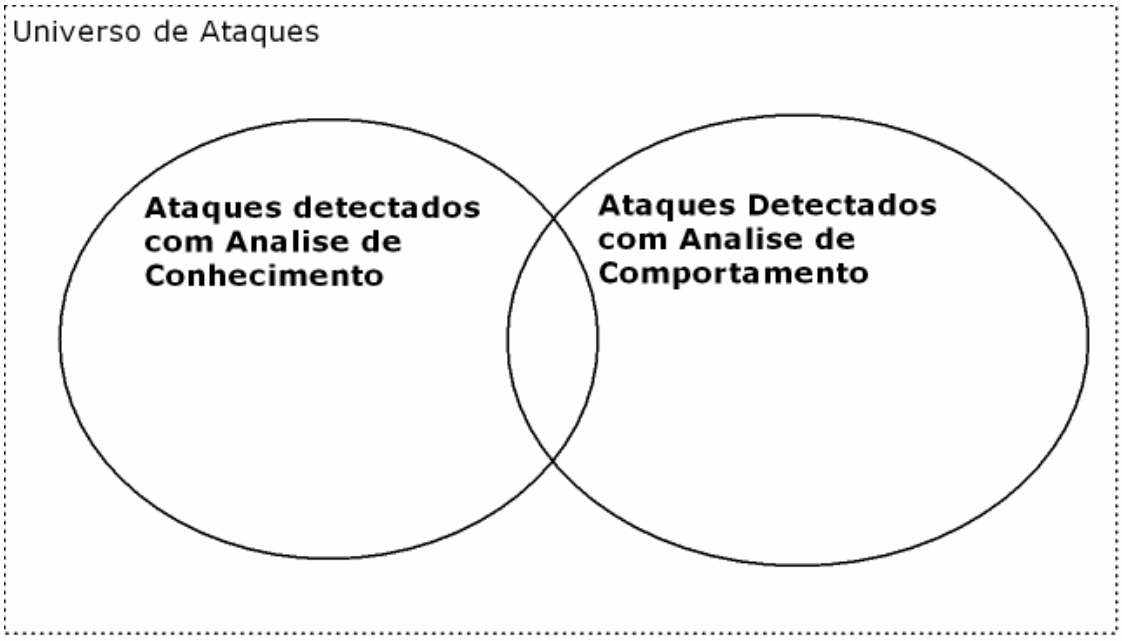


Figura 9 Contemplação das técnicas de defesa em um universo de ataques

3.7 Sumário do Capítulo

Neste capítulo foi apresentada a arquitetura de detecção de intrusão em grid onde cada nodo é responsável por detectar seus ataques e informar ao grid um alerta caso encontre um ataque. Um serviço de IDS foi proposto contendo auditor de eventos, dois sistemas de análise de ataques. O auditor de eventos é responsável por capturar dados do log do nodo e suas informações de comunicação e responde a pergunta "*Como coletar os dados para fazer uma auditoria em IDS no Grid?*". Um dos sistemas de análise utiliza a técnica de análise baseada em conhecimento onde o conhecimento de ataques é representado por um conjunto de regras e responde a pergunta "*Como detectar ataques conhecidos em um ambiente de Grid?*". Uma forma de notação de regras que leva em conta os dados de auditoria do grid foi proposto. Porém, esta técnica possui algumas desvantagens como a dificuldade de se criar novas regras a ineficiência por ataques desconhecido. Desta forma a segunda técnica detecção baseada em comportamento contempla os ataques desconhecidos e abusos de privilégios e responde a pergunta "*Como detectar um desvio de comportamento de um usuário do sistema?*".

4 ESTUDO DE CASO

Esta seção explora o comportamento do modelo proposto no capitulo 3 onde foi apresentada a arquitetura abrangente de detecção de intrusão, contemplando detecção baseada em comportamento e em conhecimento. Neste capitulo é apresentada a implementação do protótipo para validação e é examinada a eficiência em exatidão e o desempenho para conhecer a sua viabilidade em termos de custos das técnicas apresentadas.

Este capitulo esta dividido da seguinte forma:

- A seção 4.1 apresenta o ambiente de prototipagem;
- A seção 4.2 apresenta a preparação de dados para a validação;
- A seção 4.3 apresentação da definição de qualidade e as e as métricas de IDS;
- A seção 4.4 mecanismos de captura de dados e auditoria de eventos;
- A seção 4.5 apresenta as simulações do sistema de detecção baseado no comportamento e os dados de desempenho e eficiência;
- A seção 4.6 apresenta as simulações do sistema de detecção baseado no conhecimento e os dados de desempenho deste ambiente;
- A seção 4.7 apresenta as conclusões do capitulo.

4.1 Ambiente de prototipagem

Para validar a arquitetura proposta neste trabalho um protótipo foi implementado como um serviço de detecção de intrusão em grid chamado de IDSService sobre o Grid-M (Rolim 2007). O Grid-M foi utilizado por ser um middleware desenvolvido no LRG e por possuir características como flexibilidade, escalabilidade, confiabilidade, extensabilidade e independência de plataforma. Mesmo sendo um middleware de grid para mobilidade contempla os requisitos necessário para a implementação do sistema de detecção de segurança.

Foi conduzida uma serie de testes e experimentos para avaliar a performance e eficiência das técnicas. As simulações foram executadas em uma arquitetura x86 com um processador Pentium M 1.5 Ghz, 1Gb ram sobre o S.O. Windows XP. A proposta de IDS em grid relatada neste trabalho descreve uma arquitetura onde cada nodo é responsável pela detecção de intrusão e integridade de seus dados e interagindo com os demais nodos informando ataques e bloqueando ações maliciosas desta forma é contemplada a segurança do grupo. Conforme descrido no capítulo 3.1.

4.2 Preparação de dados para a validação

A simulação permite que seja criado um ambiente controlado onde os testes podem ser repetidos mantendo as características necessárias. Para a realização dos experimentos foram criadas tabelas de dados com os elementos de auditoria provenientes tanto do *Log* do ambiente como dos dados capturados da comunicação do nodo.

Para a realização dos testes foram preparados três tipos de dados simulando ações legitimas, anomalias e ataques como segue:

- Dados representando ações legitimas; Para preparar os dados representando as ações legitimas foi executado um conjunto de serviços conhecidos simulando um comportamento usual;
- Dados representando anomalias no comportamento: Para representar as ações anômalas do comportamento procurou-se alterar os serviços e a freqüência de uso. Por exemplo um abuso de privilégios ou comportamento anômalo seria o caso de um departamento de ensino informatizado. Onde as notas são informadas eletronicamente no final de um período é conhecido que pelo menos duas notas em cada 100 vão ser corrigidas devido a algum problema. Desta forma uma correção de 10 notas consecutivas merece uma atenção especial devendo ser confirmada a legitimidade da ação;
- Dados representando violação de regras: Para se preparar o conjunto de dados contendo violação de regras foi criada uma coleção de pacotes de auditoria contendo uma serie de elementos infringindo uma base de regras.

Esse conjunto de dados é extraído da comunicação dos nodos e do log gerado pelo middleware como será visto com maiores detalhes na seção 4.4 onde serão apresentados e cada elemento será explicado.

4.3 Performance e métricas

Para mensurar a eficiência de um sistema de detecção de intrusão pode-se usar os seguintes parâmetros(Debar 1999):

- Exatidão: avalia a adequação da detecção do ataque e a ausência de alarme falso. Um sistema é dito imperfeito quando um IDS alerta com intrusão uma ação legitima no ambiente. A exatidão mede os números de falsos positivos e falsos negativos:
    - Falso positivo é quando o sistema alerta como ataque uma ação legitima no ambiente;
    - Falsos negativos quando ocorre um ataque e não é gerado um alerta desta forma a ação maliciosa passa despercebido.
- Cobertura: esta métrica é a muito difícil de ser aplicada. Para conhecer a cobertura real da técnica seria necessário conhecer todos os ataques existentes. Porem pode-se determinar uma media de ataques que a técnica consegue cobrir.
- Desempenho: o desempenho no IDS é a porcentagem de processamento utilizada para avaliar se um evento é legitimo. Se o desempenho for baixo não se pode utilizar o IDS em tempo real (tempo de execução);

Neste trabalho procura-se medir a exatidão da técnica de analise do comportamento verificando seus alarmes falsos para um conjunto de ataques controlados. Já a técnica de analise do conhecimento não cabe analisar a exatidão uma vez que todos os ataques conhecidos são alertados. A cobertura da técnica de analise do comportamento se refere

as anomalias no comportamento em quanto no técnica de analise do conhecimento ao número de regras registradas em sua base de dados.

4.4 Auditor de Eventos

Todas as requisições recebidas pelo nodo bem como as mensagens enviadas são capturadas pelo auditor de comunicação. A captura desses dados é de fundamental importância para se ter uma analise do comportamento do grid afim de descobrir ataques.

A tabela 3 representa um exemplo de mensagem enviada e uma mensagem recebida. Essas informações são referentes ao grid[1] escolhido para implementar o estudo de caso porem de uma forma geral vale para outras implementações de grid. Através dessas mensagens pode-se obter os seguintes dados:

- *Nome do serviço:* Atributo do serviço que esta sendo auditado. Representado pelo elemento *Service*. Este parâmetro é informado na construção do serviço**;**
- *Nodo de origem**:*** Refere-se ao nome do nodo que da origem a ação ou serviço. Representado pelo elemento *Originator***;**
- *Nodo de destino:* Refere-se ao nome do nodo que tem o destino da ação ou serviço. Representado pelo elemento *destination***;**
- *Identificador da tarefa:* Representa um valor alfa-numérico único para a tarefa. Este valor é criado na inicialização da tarefa. Representado pelo elemento *TaskId*;
- *Prioridade:* Define a prioridade da tarefa. Pode variar de 1 até 5, sendo 5 a prioridade máxima., representada pelo elemento *Priority***;**
- *Parâmetros:* Refere-se aos parâmetros do serviço. Cada serviço possui seus próprios parâmetros como, por exemplo, em um sistema de multiplicação os parâmetros são os valores que vão ser multiplicados. Representado pelo elemento *Parameters***;**
- *Resultado da Tarefa:* Esta informação é própria de cada serviço e se refere ao valor de resposta de um serviço. Representado pelo elemento *Result***;**

[1] Dados referentes ao M*iddleware* Grid-M(**ROLIM 2007**) escolhido para prototipagem do sistema.

Descrição	**Mensagem**
Solicitação de um Serviço	<task> <service>discovery</service> <originator>node-1</originator> <destination>node-4</destination> <taskid>task-1002</taskid> <timestamp>1170953490330</timestamp> <priority>2</priority> <parameters> <discovery-parameters> <service>multiply</service> <requester>node-0</requester> </discovery-parameters> </parameters> </task>
Resultado de uma tarefa	<task-result> <result-code>OK</result-code> <originator>node-4</originator> <destination>node-0</destination> <taskid>task-1000</taskid> <timestamp>1170953492549</timestamp> <parameters> <result>6</result> </parameters> </task-result

Tabela 3 Mensagem do nodo solicitando um serviço

Cada ação realizada pelo nodo um log é gerado informando os métodos e os parâmetros invocados por esta ação.

A tabela 4 apresenta um exemplo do log do sistema onde cada linha representa uma ação ambiente. Com o log é possível identificar o comportamento interno do nodo. Tem-se as seguintes informações:

- *Responsável pela anotação:* Representado pelo elemento *LogName* e geralmente indica o nome do nodo que criou a anotação;
- *Tipo de anotação:* Representado pelo elemento *Type*. Categoriza a anotação em: aviso, alerta ou anotação de erro. Pode ser usado como indicador da importância do registro no log;
- *Evento:* Representado pelo elemento *Event*. Sempre que ocorrer um acontecimento como a inicialização de um serviço, um nodo é adicionado, o recebimento de uma tarefa o sistema de log irá criar uma anotação e será registrado o evento.
- *Element:* Representada pelo elemento *Element.* Indica a classe do sistema que criou a anotação;
- *Método:* Representado pelo elemento *Method* . Todo evento é concebido por uma função ou método no sistema este campo indica qual função ou método criou a anotação;
- *Mensagem:* Representado pelo elemento *Message.* Campo que fornece os detalhes sobre o evento.

Cada um dos elementos provenientes da comunicação: *Service, Originator, destination, TaskId, Priority, Parameters, Result;* e proveniente do sistema de log: *LogName, Type, Event, Element, Method, Message*, são utilizados pelo sistema de auditoria. Com esses dados é possível fazer um monitoramento fino dos eventos que estão ocorrendo no Nodo.

LogName	Type	Event	Element	Method / (Positio)	Message
node-4	Note	HTTPServer	http-server	addServlet(1)	address=/
node-0	Note	RouteAdd	networker	addRoute(1)	host=node-1:url=http:///localhost:8001:metric=1
node-0	Note	DiscoveryTask	node	sendServiceRequest(1)	trying to discovery service multiply requested by node-0
node-0	Note	RouteTableRequested	networker	getRouteTable(1)	returning route table of node-0
node-0	Note	RouteFound	networker	findRoute(1)	host=node-1:url=http://localhost:8001:found in route table
node-0	Note	HTTPSending	http-server	httpClient(1)	url=http://localhost:8001/request:size=328
node-1	Note	http-servlet	HTTPServletReceived	doPost(2)	task=Task(task-1001,discovery,node-0,node-1,...)
node-1	Note	ReceivedDiscoveryTask	node-1	processServiceRequest(0)	node-1 service multiply requested by node-0 not found! forwarding discovery task...
node-1	Note	StoreInTaskStak	networker	StoreTaskStack(1)	task=task-1001:queueSize=1
node-4	Note	ReceivedDiscoveryTask	node-4	processServiceRequest(0)	found service multiply requested by node-0 in this node
node-0	Note	PingNode	networker	ping(1)	sending ping to node-4
node-4	Note	pingReceived	node-4	processServiceRequest(0)	pong reply to node-0
node-4	Note	http-servlet	HTTPServletResponding	doPost(4)	task=Task(task-1005,ping,node-0,node-4,...):taskresult=TaskResult(task-1005,OK,node-4,node-0,...):status=200
node-4	Note	http-servlet	HTTPServletResponding	doPost(4)	task=Task(task-1000,multiply,node-0,node-4,...):taskresult=TaskResult(task-1000,OK,node-4,node-0,...):status=200
node-0	Note	HTTPReceiving	http-server	httpClient(2)	result=200:message=OK:size=235
node-0	Note	HTTPSending	http-server	sendTask(4)	done:taskResult for task-1000 stored: taskResult.resultCode=OK: queuesize=3

Tabela 4 exemplo de log

4.5 Simulações do Sistema de Detecção Baseado no Comportamento

Para realizar a simulação de detecção baseada no comportamento foi necessário capturar os elementos descritos na seção 4.4. Foi considerada a utilização dos dados de auditoria do log e da comunicação porem como os dados que compões o log do sistema, com exceção do elemento *message*, são um conjunto limitado de valores com poucas variações é difícil encontrar padrões que representam um ataque. Desta forma foram explorados os elementos da comunicação para se fazer a validação desta técnica.

Foi necessário criar um vetor onde o valor de cada elemento possui um correspondente numérico único. Podendo assim ser usado como dados de entrada na rede neural. Um exemplo ilustrativo pode ser verificado tabela 5 onde se tem o elemento *service* com o valor *discovery* e esse valor é representado pelo numero 1.

A validação da técnica de detecção baseada em comportamento se deu com a utilização de IA e através de uma rede neural do tipo FeedForward (Idris 2005). No ambiente de simulação foi idealizado uma situação onde se tinha 10 acessos e dos quais 5 eram intrusos e 5 usuários legítimos.

Elemento	Valor	Valor Numérico
Service	Discovery	1
Originator/ destination	node-0	2
Originator/ destination	node-1	3
Originator/ destination	node-2	4
Originator/ destination	node-3	5
parameters	Multiply	6
result-code	OK	7
Parameters	Games	8
Parameters	AGE II	9

Tabela 5 Elementos da auditoria e sua representação numérica

A rede neural foi configurada para responder -1 para ações legitimas e +1 para ataques desta forma os dados obtidos eram em um intervalo de [-1;+1]. Foi considerado que qualquer valore menor e igual a zero representava uma ação legitima e maior que zero representava um ataque. A incerteza da rede pode ser verificada com as saídas próximas de zero.

O treinamento da rede neural iniciou com um conjunto de dados representando dez dias de uso. Porem com esses valores se tinha um numero muito elevado de falsos negativos alem de um índice de incerteza muito elevado. A medida que um período maior de amostras foi sendo adicionado no aprendizado as respostas no ambiente de teste também foram se aproximando do esperado. A figura 10 representa a variação de falsos negativos, ou seja, ataques não são detectados em relação aos dados de entrada.

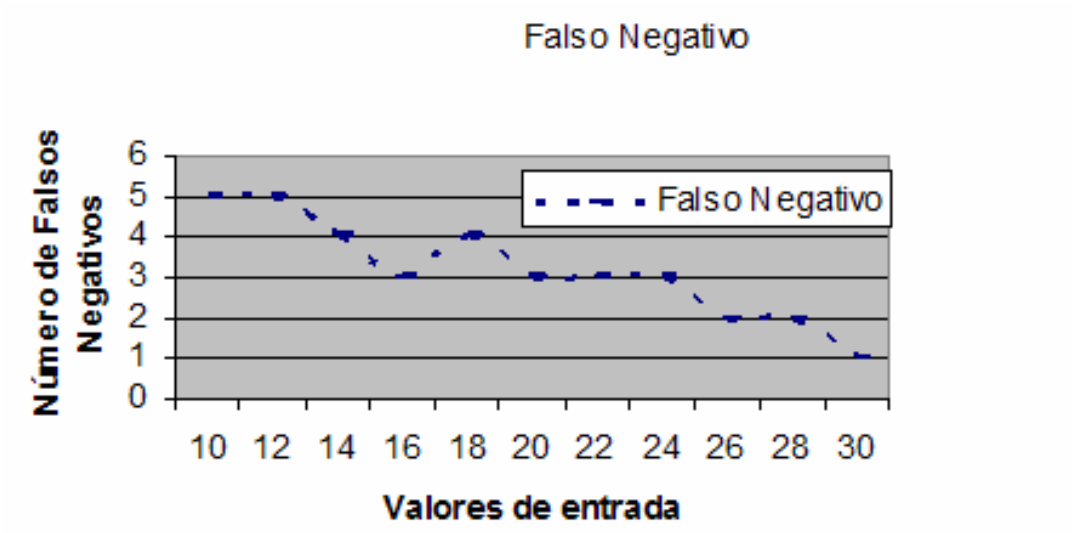


Figura 10 Análise da eficiência da detecção de intrusão baseada no comportamento observando-se o numero de falsos negativos

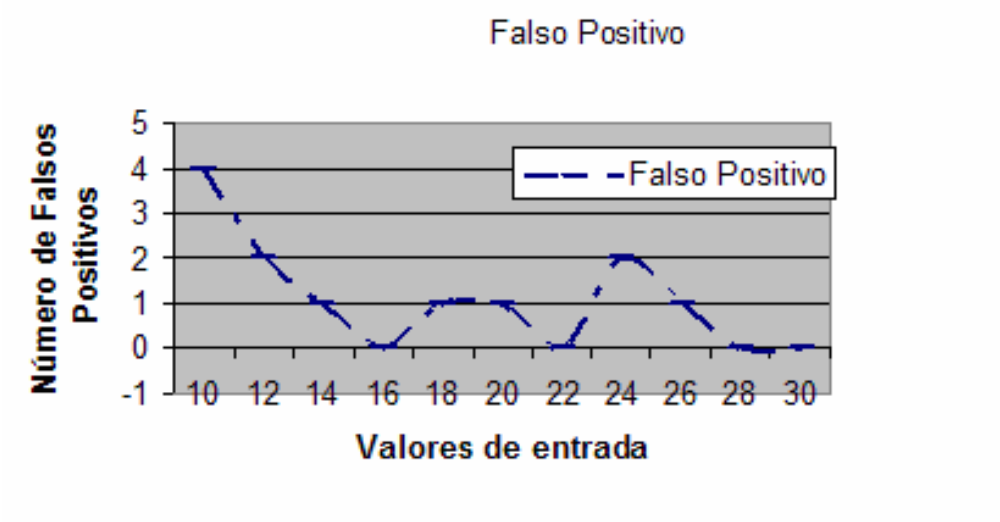
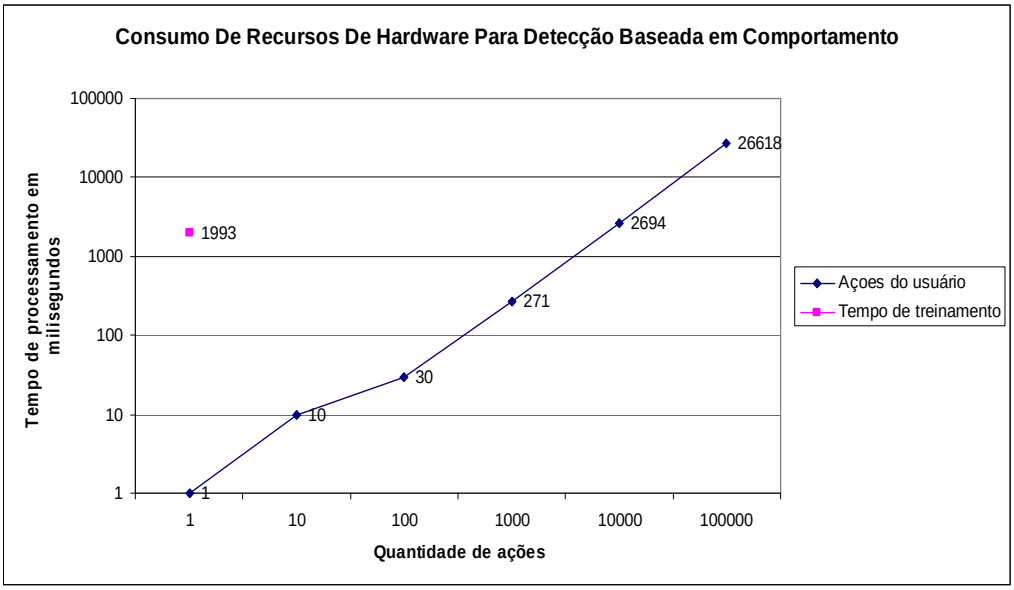


Figura 11 Análise da eficiência da detecção de intrusão baseada no comportamento observando-se o numero de falsos negativos positivos

Nesta ilustração se analisou o número de dias conveniente para determinar o perfil de uso legitimo. Como uma RNA não possui um comportamento determinístico os gráficos nas figuras 10 e 11não representam uma progressão decrescente linear.

A figura 11 representa a variação de falsos positivos em relação aos dados de entrada, ou seja, ações legitimas marcadas com ataques. A rede neural tem a tendência de não marcar com demasias ações legitimas como ataques pode-se perceber que o numero de falsos negativos é maior para a mesma quantidade de dados. Já com dados representando 16 dias de comportamento não se teve alarmes falsos. Porem o índice de incerteza ainda era alto com muitas saídas próximas de zero. Com o aumento de dados no treinamento a rede foi representando suas respostas mais próximas de -1 e +1 ataques ou uso legitimo. Com a base de 28, 29 e 30 dias de dados de entrada o algoritmo mostrou um número muito baixo de falsos positivos porem com um numero elevado de repetições dos testes se observou que a quantidade de falsos positivos pode variar. O que caracteriza a rede neural como se sabe não se pode ter 100% de acertos. É importante destacar que o comportamento legitimo tende a mudar com o tempo de uso do sistema o que gera a necessidade de constantes atualizações na base de dados que compõe o treinamento da rede.



Fonte:

Figura 12 Análise de desempenho de detecção baseada no comportamento

O teste de desempenho foi elaborado para avaliar o custo da técnica de analise baseada em comportamento e verificar qual é o consumo de recurso de hardware. Um teste de carga onde a técnica analisou de 1 a 100.000 ações foi elaborado.

A simulação de 100.000 ações é hipotético superando o volume de dados usual do ambiente e serve para conhecer o comportamento do sistema em um momento de sobrecarga. Uma ação demora 0,000271 em média segundos para ser processada o tempo de resposta 1000 ações 0,27 segundos. Com o aumento de carga o gráfico mostrou que o crescimento do tempo em relação às ações é linear. O tempo de treino para uma base de 30 dias de exemplos demorou 1,993 segundos. Porem essa é uma ação esporádica. A atualização da base de comportamento deve ser planejada de acordo com a rotina do ambiente de execução, pois o comportamento tende a mudar com o tempo.

4.6 Simulações do Sistema de Detecção Baseado no Conhecimento

Para realizar o estudo de casos da detecção baseada em conhecimento um protótipo foi implementado baseado na arquitetura proposta na seção 3.5. Utilizaram-se os dados de auditoria descritos na seção 4.4. Ao contrario do sistema de detecção baseado no comportamento foi utilizado tanto os dados de *log* como os dados provenientes da comunicação para se fazer a analise. Uma serie de regras foram elaboradas para ilustrar políticas de segurança que são analisadas pelo sistema de detecção de intrusão.

Foram coletados os dados de auditoria referentes ao serviço de descoberta de rota, descoberta de serviço, solicitação de serviço, resultado do serviço. Para ilustrar o ataque foram introduzidos dados de auditoria com valores que violavam regras como por exemplo o XML a seguir representa um serviço chamado *Storage* com um parâmetro XXX.

```
<task>
        <service>Storage</service>
        <originator>node-3</originator>
        <destination>node-5</destination>
        <taskid>task-1002</taskid>
        <timestamp>1170953490330</timestamp>
        <priority>2</priority>
        <parameters>
                <service>XXX </service>
                <requester>node-0</requester>
        </parameters>
</task>
```

Uma serie de regras foram elaboradas para testar o desempenho do sistema. Porem o escopo deste trabalho não abrange a descoberta de novos ataques nem a elaboração de uma base de ataque em grid. Desta forma as regras criadas aqui serviram para validar o sistema de detecção e seu desempenho. A seguinte regra caracteriza como ataque

qualquer mensagem do serviço *Storage* com conteúdo XXX. Desta forma o exemplo anterior de comunicação é considerado um ataque.

```
<alert source=”communication”>
        <identification>023</identification>
        <ruleInformation>
                This rule verify the inappropriate use
        </ruleInformation>
        <version>1.0.3</version>
        <name>inappropriateUse</name>
        <credits>LRG</credits>
        <element value="service">web</element>
         <comprises>XXX</comprises>
</alert>
```

O sistema de analise baseado no conhecimento implementado se comporta da seguinte maneira:

1. Ao ser iniciado ocorre a leitura das regras armazenadas em um arquivo XML;
2. Em seguida faz um pré-processamento para organizá-las em uma estrutura de dados;
3. No próximo passo o auditor captura os dados de log e comunicação;
4. É realizado um pré-processamento dos dados para se montar uma estrutura de dados que divide dados de log e dados de comunicação e fornece fácil acesso a cada elemento;
5. Em seguida são verificadas as regras cabíveis ao pacote te auditoria;
6. Caso exista um ataque ou violação é gerado um alerta.

Foi realizado um teste de carga para o algoritmo apresentado. O teste se deu simulando a analise de 10 até 1.000.000 regras para uma ação. A verificação se da basicamente verificando o texto ou campo numérico e comparando com as regras. No núcleo do analisador de conhecimento existem duas funções primaria uma para comparar a presença de conteúdo impróprio e outra para compara intervalo de dados numéricos. O desempenho dos testes em babas as funções foram equivalentes. A

comparação de cem mil regras para uma ação consumiu 0,361 segundos e um milhão de regras 2,7 segundos. O sistema possibilita que seja realizada a analise para detectar intrusão em tempo real até um limite de regras.

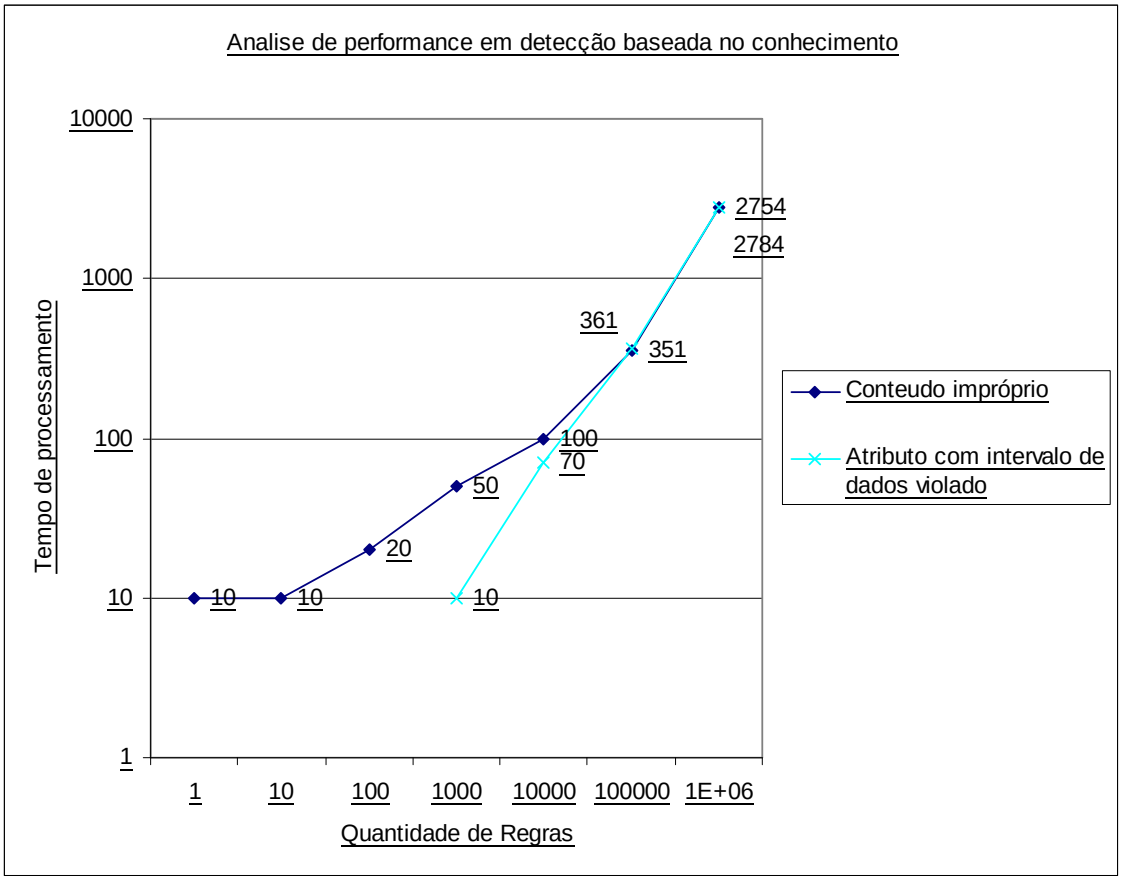


Figura 13 Análise de Performance em Detecção Baseada no Conhecimento

4.7 Sumário do capítulo

Aqui foi apresentada a implementação do modelo proposto no capitulo 3. O estudo de caso serviu para demonstrar o comportamento da plataforma na prática e também como ela pode ser utilizada para solucionar um problema no ambiente de grid.

Foi apresentado o ambiente de prototipagem bem como os dados utilizados nos experimentos. As técnicas foram validadas de acordo com as métricas referentes a cada uma delas. Com a técnica de analise de comportamento os testes de exatidão mostraram um baixo numero de falsos positivos com uma grande quantidade de exemplos na base de comportamento. E um baixo numero de falsos negativos para a mesma quantidade de dados. O experimento demonstrou que o algoritmo consome 2,6 segundos para 10.000 ações o que extrapola o numero de ações executadas no ambiente. A técnica analise do conhecimento mostrou que consome 2,7 segundos para analisar um volume de um milhão de regras. Em síntese este capítulo demonstrou que a arquitetura foi implementada em forma de protótipo. Os dados coletados e as técnicas apresentadas no capitulo 3 possibilitam aumentar a abrangência na detecção de intrusão em um ambiente de grid monitorando de perto as atividades do middleware e focando nos ataques próprios dessa arquitetura.

5 CONCLUSÕES

Neste trabalho, foi proposto um sistema de detecção de intrusão em grid capaz de abranger ataques desconhecidos, como ações maliciosas dos usuários pelo seu desvio de comportamento, e ataques conhecidos, utilizando uma base de regras que sintetizam ataques conhecidos. A solução proposta neste trabalho consiste em um serviço de IDS que captura dados do log e da comunicação do middleware de grid garantindo a segurança dos nodos de forma que cada um contribua para a segurança do grid. Duas técnicas de análise foram utilizadas para garantir a abrangência.

A técnica de detecção baseada em comportamento que utiliza uma rede do tipo *feed-forward* para reconhecer o padrão de comportamento do usuário e alertar em caso de ações anormais, sendo muito eficiente com pequenas tolerâncias. Com esta técnica foi possível desenvolver uma solução com baixo índice de alarmes falsos e um baixo índice de ações não alertadas. Conseguiu-se um desempenho foi satisfatório, uma vez que para um volume elevado de regras foi obtido um tempo de resposta hábil. Porem a necessidade de adicionar ataques conhecidos justificou a utilização de uma segunda técnica.

A técnica de detecção baseada em conhecimento consiste na utilização de regras para definir o que é um ataque. Foi criada uma forma de descrever ataques.

Para realizar a análise um sistema de auditoria foi especificado para fornecer a captura de dados de log do ambiente do middleware de Grid e dados da troca de mensagens entre os nodos possibilitando a verificação dos possíveis ataques conhecidos.

Com a utilização dessas duas técnicas sobre esse resultado com uma vigilância intensiva das atividades no grid utilizando os dados de auditoria conseguimos responder a pergunta principal “*Quais métodos utilizar para fornecer uma maior abrangência em detecção de intrusão em ambientes de Grid?*”.

Foi apresentada uma validação da arquitetura onde pode-se confirmar que as técnicas descritas não exigem grande necessidade de recursos de hardware. As duas técnicas apresentaram desempenho satisfatório. onde 10.000 análises da detecção

baseada em comportamento são feitas em 2,5 segundos e com análise de conhecimento 10.000 análises são realizadas em 0,1 segundo.

Uma contribuição que não era esperado no inicio do trabalho e foi obtita com uma deficiência na literatura foi uma forma de descrever regras para ataques em grid.

Utilizar a análise individual de ataques onde cada nodo é responsável por seus dados aditáveis reduz a complexidade do sistema de detecção de intrusão e o volume de dados a serem analisados comparando com uma arquitetura onde todos os dados de auditoria são capturados no grid formando virtualmente um único recurso. A comunicação, sincronização e heterogeneidade da arquitetura para o middleware foram problemas desconsiderados uma vez que o ambiente de grid se preocupa em trata-los.

A redução das falhas de segurança só pode se dar com uma constante atualização do sistema de IDS, para abranger as complexidades de cada serviço do grid. As técnicas de detecção de intrusão demonstraram eficiências em dispositivos de pouco poder computacional não sendo necessário o envio de dados para o processamento em um nodo onde se encontra um recurso de processamento.

5.1 Trabalhos Futuros

Como o desenvolvimento deste trabalho se abriu um leque de possibilidades para contribuições e aperfeiçoamento de detecção de intrusão em grid como:

- Criação de uma base de assinaturas e compartilhar com outros centros de pesquisa;
- Aperfeiçoar e propor o padrão de assinatura de ataques em ambientes de Grid;
- Investigar ataques e incrementar a base de assinaturas de ataques;
- Incorporação de NIDS a estrutura para o monitoramento dos outros serviços do nodo;

- Resposta a ataques, ser capaz de bloquear a origem do ataque impedindo que cause danos no ambiente ou ignorar ataques;
- Tolerância a ataques, explorar a redundância de serviços essenciais e garantir o pleno funcionamento do ambiente enquanto ele esta sendo atacado.

REFERÊNCIAS

AXELSSON, Stefan. *Research in Intrusion-Detection Systems: A Survey.* Technical Report TR-98-17, Department of Computer Engineering, Chalmers University of Technology, Göteborg, Sweden, 93 p. aug. 1999.

BARMOUTA, Alexander; BUYYA, Rajkumar. GridBank: *A Grid Accounting ServicesArchitecture (GASA) for Distributed Systems Sharing and Integration*. In: International Parallel And Distributed Processing Symposium (IPDPS), 17., 2003. Anais… Washington: IEEE Computer Society, p. 254a, 2003.

BRANDÃO, José Eduardo Malta de Sá. *Congregação de Sistemas de Auditoria: Uma Abordagem Orientada a Serviços para Construção de Sistemas de Detecção de Intrusão de Larga Escala.* 2004, 120 f. Qualificação (Doutorado em Engenharia Elétrica)– Programa de Pós-Graduação em Engenharia Elétrica, Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis, SC, 2004.

BRAY, Tim et al. Extensible Markup Language (XML) 1.0 (Third Edition). W3C Recommendation. Disponível em: <http://www.w3.org/TR/REC-xml/>. Acesso em: 10 dez. 2005.

BUYYA, Rajkumar. "Economic-based Distributed Resource Management and Scheduling for Grid Computing". 2002. 180 f. Thesis (Doctor of Philosophy)– School of Computer and Software Engineering, Monash University, Melbourne, Australia, 2002.

BUYYA, Rajkumar; MURSHED, Manzur. GridSim: *"A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid"* Computing. Concurrency and Computation: Practice and Experience (CCPE) Journal, USA, v. 14, n. 13-15, p. 1175-1220, dec. 2002.

CHOON, O. T., SAMSUDIM, A. *"Grid-based Intrusion Detection System"*. The 9 IEEE Asia-Pacific Conference Communications, Setembro 2003.

CHUNG, S. P.; MOK, A. K.; *"The LAIDS/LIDS Framework for Systematic IPS Design"* Proceedings of the Fourth IEEE International Workshop on Information Assurance (IWIA'06), 2006.

DEBAR, H., DACIER, M., WESPI, A., *"Towards a taxonomy of intrusion detection systems,"* Int. J. Computer and Telecommunications Networking, vol. 31, no. 9, pp. 805-822, 1999.

DEPREN., O; Topallar., M.; Anarim., E.; Kemal M.; *An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks*; Expert Systems with Applications p 713-722; 2005.

DUAN, Q.; Hu, C.; We,i H. ;*Enhancing network intrusion detection systems with interval methods;* Symposium on Applied computing; 2005.

FANG-YIE, Leu et al. *Integrating Grid with Intrusion Detection*. In: International Conference On Advanced Information Networking And Applications (AINA), 19., 2005, Taipei, Taiwan. Anais… [S.l.]: IEEE Computer Society, v. 1, p. 304-309, 2005.

FENG, G. ; Dong, X.; Weizhe L.; Chu, L.; Li, J.: GHIDS: *Defenfing Computational Grids against Misusing of Shared Resource*. In:Asia-Pacific Conference on Services Computing (APSCC'06), 2006.

FOSTER, I. and Kesselman, C. The GRID *"A Blueprint for a New Computing Infrastructure.*" Morgan Kaufman Publishers, New York, US, 2000.

FOSTER, I. ;C.Kesselman, G. Tsudik, S. Tuecke.:A Security *Architecture for Computational Grids.* Proc. 5th ACM Conference on Computer and Communications Security Conference, pp. 83-92, 1998.

FOSTER, I.: *What is the Grid? A Three Point Checklist* GRIDtoday, v. 1, n. 6, July 2002.

HEADY, R.; LUGER, G.; MACCABE A.; SERVILLA M.; *The Architecture of a Network Level Intrusion Detection System, Technical Report;* Departament of Computer Science; University of New Mexico; USA; 1990.

HEBERLEIN, L. Todd et al. *A Network Security Monitor.* In: Ieee Symposium On Research In Security And Privacy, 1990, Oakland, CA, USA.Anais… Los Alamitos: IEEE Computer Society, p. 296-304, 1990.

HUMPHREY, M., Thompson, M., and Jackson, K. R. *"Security for Grids,"* in Proc. of the IEEE (Special Issue on Grid Computing), vol. 93, no. 3, pp. 644-652, March 2005.

MCCUBBIN, Chris; LUU, Michael. *JavaIDMEF Message* Implementation v.941beta. 2002. Disponível em: <http://sourceforge.net/projects/javaidmef>. Acesso em: 22 julho. 2006.

IDRIS, N. B.; SHANMUGAM B.; *"Artificial Intelligence Techniques Applied to Intrusion Detection"* IEE Indicon 2005 Conference, India, pp.52-55, 2005.

KANNADIGA, Pradeep; ZULKERNINE, Mohammad. DIDMA: *A Distributed Intrusion Detection System Using Mobile Agents*. In: Acis International Conference On Software Engineering, Artificial Intelligence, Networking And Parallel/Disitributed Computing, 6., 2005, Towson, Maryland, USA. Anais… USA: IEEE Computer Society, p. 238-245. 2005.

KENDALL, Kristopher. A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems. 1999. 124 f. Master Thesis – Department of Electrical Engineering and Computer Science, Massachusets Institute of Technology (MIT), Cambridge, MA, USA, 1999.

KENNY, S. and Coghlan, B. *"Towards a grid-wide intrusion detection system,"* in Proc. European Grid Conference (EGC2005), pp. 275-284, Amsterdam, The Netherlands, February 2005.

KUMAR, S.; Spafford, E; *"A Software Architecture to Support Misuse Intrusion Detection,"* in 18th National Information Security Conference, pp. 194-204, 1995.

MCHUGH, John. *Intrusion and Intrusion Detection*. International Journal of Information Security, v. 1, n. 1, p. 14-35, aug. 2001.

MODULO. *9ª Pesquisa Nacional De Segurança Da Informação. OUTUBRO 2003*

NAESS, E.; FRINCKE, D. A.; McKINNON, A. D.; BAKKEN, D. E.; *"Configurable Middleware-Level Intrusion Detection for Embedded System"* International Conference On Distriuted Computing Workshops (ICDCSW05), 2005.

NAQVI, Syed; RIGUIDEL, Michel. *Threat Model for Grid Security Services*. In: European Grid Confence (EGC), 2005, Amsterdam, The Netherlands. Anais… [S.l.: s.n.], 2005.

NWANZE, N. ; SUMMERVILLE, D. H.; SKORMIN, V. A.; *"Real-Time Identification of Anomalous Packet Payloads for Network Intrusion Detection"* Workshop On Information Assurance And Security United States Military Academy, 2005.

PORRAS, Philip A.; NEUMANN, Peter G.; EMERALD: *Event Monitoring Enabling Responses to Anomalous Live Disturbances.* In: National Information Systems Security Conference, 20., 1997, Baltimore, Maryland, USA. Anais… [S.l.: s.n.]. p. 353-365, 1997.

SANG-KIL, Park et al. *Supporting Interoperability to Heterogeneous IDS in Secure Networking* Framework. In: ASIA-PACIFIC CONFERENCE ON COMMUNICATIONS (APCC), 9., 2003, Penang, Malaysia. Anais… [S.l.: s.n.]. v. 2, p. 844-848, 2003.

SCHULTER, A.; VIEIRA, K.;WESTPHALL, C; "*Defending Grids Against Intrusions*". Lecture Notes in Computer Science; International Workshop on Self-Organizing Systems. Germany IWSOS, 2006;

SCHULTER, Alexandre; NAVARRO, Fabio P.; KOCH, Fernando L.; WESTPHALL, Carlos B.; *"Towards Grid-based Intrusion Detection."* In: IEEE/IFIP NETWORK Operations & Management Symposium, 10., 2006, Vancouver, Canada. April 2006.

HAYKIN, S.*; Redes Neurais* - 2.ed.Princípios e Prática. 2001.

ROLIM. C. O. *;O uso de grade computacional para a integração de gerenciamento de sensores e dispositivos móveis.* Dissertação de Mestrado PPGCC 2007

RUMELHART D E, HINTON G E, WILLIAMS R J. Learning Representation by Backpropagation Errors. Nature, 1986.

SILVA, Paulo Fernando; WESTPHALL, Carlos Becker. *Um Modelo para Interoperabilidade de Respostas em Sistemas de Detecção de Intrusão*. In: Simpósio Brasileiro de Redes de Computadores (Sbrc), 23.,2005, Fortaleza, Ceará, Brasil. Anais... [S.l.: s.n.], 2005.

TOLBA, M. F., ABDEL-WAHAB, M. S., TAHA, I. A., AL-SHISHTAWY, A. M., "GIDA: *Toward Enabling Grid Intrusion Detection Systems*" IEEE/ACM International Symposium On Cluster Computing And The Grid (CCGrid), 5., 2005, Cardiff, UK. Anais… [S.l.:s.n.], 2005.

TOLBA, M. et al. *Distributed Intrusion Detection System for Computational Grids*. In: International Conference On Intelligent Computing And Information Systems, 2., 2005, Cairo, Egypt. Anais… [S.l.]: ACM, 2005.

VIEIRA, K. ; SCHULTER, A.; SILVA, P.; WESTPHALL, C.; *"Detecção de Intrusão Distribuída em Ambientes de Grid"*. In: SSI - Simpósio de Segurança em Informática, 2006, São José dos Campos. Anais, 2006.

VIEIRA, K.; IDSService <http://grid.lrg.ufsc.br/idsservice> acessado em janeiro 2007.

WANG Jing-xin, WANG Zhi-ying, DAI Kui,. *A network intrusion detection system based on the artificial neural networks.* IDS, content filtering, Java, etc.: Proceedings of the 3rd international conference on Information security (InfoSecu '04) November 2004.

WEAVER, Nicholas et al. A Taxonomy of Computer Worms. In: ACM WORKSHOP ON RAPID MALCODE, 1, Washington, DC, USA. Anais… [S.l.]: ACM, 2003.

p. 11-18. Associated with the 10th ACM Conference on Computer and Communications Security. 2003.

ZHANG, Y.; XIONG, Z.; WANG, X.; "*Distributed Intrusion Detection Based On Clusteing"* PROCCEDINGS OF THE FOUTH INTERNATIONAL CONFERENCE ON MACHINE LEARNING And CYBERNETICS, Guangzhou. 2005.

ZHU, P.; GAO, J; JIANG, B.; SONG, H.*;A New Flexible Multi-Agent Approach To Intrusion Detection For Grid.* In: Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian, 13-16 August 2006

Anexos A - Dados da analise de performance de detecção baseada em conhecimento

Quantidade de Regras	Tempo de Processamento	
Conteúdo impróprio		
1	10	Milisegundos
10	10	Milisegundos
100	20	Milisegundos
1000	50	Milisegundos
10000	100	Milisegundos
100000	351	Milisegundos
1000000	2754	Milisegundos
Mensagem com valor violado		
1	0	Milisegundo
10	0	Milisegundo
100	10	Milisegundos
1000	10	Milisegundos
10000	40	Milisegundos
100000	321	Milisegundos
1000000	3085	Milisegundos
Atributo com intervalo de dados violado		
1	10	Milisegundos
10	10	Milisegundos
100	10	Milisegundos
1000	20	Milisegundos
10000	40	Milisegundos
100000	991	Milisegundos
1000000	3895	Milisegundos

Tabela 6 Dados da analse de performance de detecção baseada em conhecimento

Anexos B - Dados da analise de performance em detecção baseada no comportamento

Quantidade de ações	Tempo de processamento	
1	1	Milisegundo
10	10	Milisegundos
100	30	Milisegundos
1000	271	Milisegundos
10000	2694	Milisegundos
100000	26618	Milisegundos

Tabela 7 Dados da analise de performance em detecção baseada no comportamento

Anexo C - Dados da analise de eficiência em detecção baseada em conhecimento

Dados	Falso Positivo	Falso Negativo
10	4	5
12	2	5
14	1	4
16	0	3
18	1	4
20	1	3
22	0	3
24	2	3
26	1	2
28	0	2
30	0	1

Tabela 8 Dados da analise de eficiência em detecção baseada em conhecimento