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

Alexandre Schulter

Um Método para Detecção de Intrusão em Grades Computacionais

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação

Prof. Dr. Carlos Becker Westphall
Orientador

Florianópolis, Fevereiro de 2006

Um Método para Detecção de Intrusão em Grades Computacionais

Alexandre Schulter

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. Raul Sidnei Wazlawick
Coordenador do Programa de Pós-Graduação em Ciência da Computação

Banca Examinadora

_____

Prof. Dr. Carlos Becker Westphall
Orientador

_____

Prof. Dr. Bruno Richard Schulze

_____

Prof. Dra. Carla Merkle Westphall

_____

Prof. Dr. Mário Antônio Ribeiro Dantas

"A formulação de um problema é de longe mais essencial que sua solução, a qual pode ser meramente uma questão de habilidades matemáticas ou experimentais. Para levantar novas perguntas, novas possibilidades, ver velhos problemas sobre um novo ponto de vista requer imaginação criativa e marca os reais avanços na ciência."
(Albert Einstein)

Aos meus pais

Agradecimentos

Gostaria de agradecer ao meu professor orientador Carlos Westphall pela sua ajuda e esclarecimentos. Agradeço também a ajuda oferecida pelo Fernando Koch, os questionamentos do Marcos Assunção e o ambiente motivador proporcionado pelos colegas do LRG: Fábio Navarro, Júlio Reis, Kleber, Leonardo e Rafael Tavares.

Agradeço aos meus pais, meu irmão e minha irmã pelo incentivo e paciência que tiveram comigo. Agradeço especialmente a Elisa por ter me aturado nos períodos difíceis, ter me apoiado e ter sido uma fonte de inspirações.

Também devo lembrar e agradecer todos os amigos que de uma forma ou de outra estiveram presentes nesses últimos dois anos, que por sinal foram inesquecíveis dois anos. Grande parte deles: Arliones, Baggio, Bode, Broca, Cabeça, Christopher, Daniel, Darlan, Fabiana, Gabriela, Geremia, HU, Ivanrs, Joyce, Juninho, Louise, Neves, Parra, Pelotas, Punk, Ronan, Ronaldo, Tarouco, Toni, Valdecir, Veronez, Vô, Washington, Wolfram e o pessoal do Mountain Bike Floripa.

Sumário

Resumo **11**

Abstract **12**

1 Introdução **13**

1.1 Objetivos 15

1.2 Escopo 16

1.3 Organização do Trabalho 17

2 Detecção de Intrusão em Grades Computacionais **19**

2.1 Computação em Grade e Segurança 19

2.1.1 Evolução e Motivações da Computação em Grade 20

2.1.2 Segurança de Grades 23

2.2 A Necessidade da Detecção de Intrusão em Grades 24

2.3 Sistemas de Detecção de Intrusão 26

2.3.1 Características dos IDSs 28

2.3.2 O Papel dos IDSs 30

2.4 Intrusões em Grades 32

2.5 Motivação e Trabalhos Relacionados 35

2.6 Resumo 37

3 Proposta **39**

3.1 Análise do Problema 39

3.1.1 Deficiências das Tecnologias Atuais 39

3.1.2 Requisitos de um GIDS 41

3.2 Método 42

3.2.1 Um Método para Detecção de Intrusão em Grades 42

3.2.2 Um Exemplo de GIDS 43

3.2.3 Considerações 47

3.3 Resumo 48

4 Integração de IDSs para Segurança de Grades Computacionais **50**

4.1 Requisitos para Integração de um GIDS com outros IDSs 50

4.2 Integração Usando os Padrões do IDWG 52

4.2.1 GIDS no Modelo do IDWG 53

4.2.2 Interações do GIDS com outros IDSs 55
4.2.3 Cumprimento dos Requisitos de Integração 57
4.3 Envio de Registros de Auditoria 58
4.4 Resumo 65
5 Estudo de Caso 67
5.1 Ambiente de Simulação 67
5.2 Descrição do Estudo de Caso 69
5.3 Considerações 71
6 Conclusões 72
6.1 Principais Contribuições 76
6.2 Trabalhos Futuros 77
Referências 79
Anexo A – Uma mensagem IDMEF contendo informações de um RUR 88

Lista de Figuras

Figura 1. Cenários de um IDS agindo em um ambiente de grade 15
Figura 2. Escopo desta dissertação 17
Figura 3. Maiores marcos em tecnologias de redes e computação 21
Figura 4. Contagem do número de *hosts* de domínio na Internet 22
Figura 5. Um sistema de detecção de intrusão simples 27
Figura 6. Sofisticação do ataque vs. conhecimento técnico do intruso 32
Figura 7. Integração do GIDS com IDSs de baixo-nível 43
Figura 8. Arquitetura de um exemplo de GIDS 45
Figura 9. Modelo de IDS do IDWG 53
Figura 10. GIDS no modelo do IDWG 54
Figura 11. Coleta de dados e interações dos IDSs 56
Figura 12. Representação simplificada em UML do modelo de dados do IDMEF 61
Figura 13. Fonte e alvo de um mau uso 63
Figura 14. Grade simulada 68
Figura 15. Tarefa Analisadora do GIDS e sua rede neural 69
Figura 16. Erros de detecção em relação ao número de *gridlets* analisados 70

Lista de Tabelas

Tabela 1. Classificação dos sistemas de detecção de intrusão 29
Tabela 2. Conjunto de campos essenciais proposto pelo GGF para o RUR 60
Tabela 3. Mapeamento das informações de um RUR em uma mensagem IDMEF..... 63
Tabela 4. Características do GIDS implementado..... 71

Lista de Siglas

AG	Agente GIDS
BD	Banco de Dados
BRG	*Broker* de Recursos da Grade
CIDF	*Common Intrusion Detection Framework*
DARPA	*Defense Advanced Research Projects Agency*
DTD	*Document Type Definition* XML
GIDA	*Grid Intrusion Detection Architecture*
GIDS	*Grid-based Intrusion Detection System*
GSI	*Grid Security Infrastructure / Globus Security Infrastructure*
HIDS	*Host-based Intrusion Detection System*
HTML	*Hypertext Markup Language*
HTTP	*Hypertext Transfer Protocol*
IDS	*Intrusion Detection System*
IDMEF	*Intrusion Detection Message Exchange Format*
IDXP	*Intrusion Detection Exchange Protocol*
IDWG	*Intrusion Detection Working Group*
IP	*Internet Protocol*
IETF	*Internet Engineering Task Force*
MRG	Medidor de Recursos da Grade
NIDS	*Network-based Intrusion Detection System*
PGIDS	*Performance-based Grid Intrusion Detection System*
PSG	Provedor de Serviços de Grade
RUR	Registro de Uso de Recursos
SNMP	*Simple Network Management Protocol*
SO	Sistema Operacional
TCP	*Transmission Control Protocol*
TLS	*Transport Layer Security*
XML	*Extensible Markup Language*
WWG	*World Wide Grid*
WWW	*World Wide Web*

Resumo

As tecnologias atuais de detecção de intrusão são limitadas na proteção contra ataques que podem violar a segurança de grades computacionais. Este trabalho apresenta o problema da detecção de intrusão em grades, descreve os requisitos para identificar essas violações, propõe um método para detecção de intrusão baseada em grade, descreve um exemplo de uma arquitetura de um sistema de detecção de intrusão (IDS) distribuído e mostra como ele supera as limitações das tecnologias atuais pela integração da detecção dos ataques típicos de computadores *host* e de redes com a detecção de ataques específicos de grade e anomalias de comportamento de usuários. A integração é viabilizada pelo uso de protocolos e formatos de compartilhamento de informações entre IDSs que estão em processo de padronização pelo IETF. É também descrito um caso de uso empregando simulação de uma grade computacional onde usuários que apresentam anomalias comportamentais são identificados por um IDS de grade integrado com outros IDSs que aprendeu previamente a reconhecer mudanças de comportamento que possivelmente são o resultado de uma intrusão.

Palavras-Chaves: grades computacionais, segurança, detecção de intrusão

Abstract

Current intrusion detection technology is limited in providing protection against the attacks that may violate the security of computational grids. This work presents the problem of grid intrusion detection, describes the requirements to identify these violations, proposes a grid-based intrusion detection method, describes an example of a distributed intrusion detection system (IDS) architecture and shows how it overcomes the limitations of current technology by integrating the detection of the typical host and network attacks with the detection of grid-specific attacks and user behavior anomalies. The integration is made possible by the use inter-IDS information sharing protocols and formats that are in standardization process by the IETF. It is also described a case study that makes use of a computational grid simulation in which users who show behavior anomalies are identified by a grid IDS integrated with other IDSs that previously learned to recognize behavior changes that are possibly the result of an intrusion.

Keywords: computational grids, security, intrusion detection

1 Introdução

Grades computacionais estão emergindo como ferramentas para computação distribuída baseada na Internet. Elas facilitam o compartilhamento, troca, descoberta e agregação de recursos heterogêneos distribuídos por múltiplos domínios administrativos. Para alcançar isso, grades precisam de uma infra-estrutura que suporte vários serviços: segurança, acesso uniforme, gerenciamento de recursos, escalonamento, economia computacional e contabilidade (Barmouta & Buyya, 2003, Buyya, 2002, Foster, 2002, Foster & Kesselman, 2003).

As grades de computadores envolvem tanto a disponibilidade de recursos computacionais abundantes quanto o armazenamento de quantidades cada vez maiores de dados valiosos. Tais sistemas de informação dependem fortemente do fornecimento de segurança adequada. Um sistema de segurança amplo, capaz de responder a qualquer ataque aos seus recursos, é indispensável para garantir sua adoção por usuários e provedores de recursos (Navqi & Riguidel, 2005).

Segurança é um dos aspectos mais desafiadores da computação em grade por causa dos seus requisitos inter-domínio únicos (Humphrey et al., 2005). Middleware de grade (Azadzadeh et al., 2005) fornecem serviços de segurança preventivos baseados em criptografia, mas não é realista prevenir absolutamente o aparecimento de brechas de segurança, especialmente em sistema distribuídos complexos como as grades. Mesmo que os serviços de segurança oferecidos por um middleware de grade sejam projetados e implementados cuidadosamente para evitar vulnerabilidades, intrusos podem explorar defeitos em qualquer dos outros componentes envolvidos. Além do mais, uma grade não pode defender-se de senhas roubadas e usuários legítimos que abusam de seus privilégios para executar atividades maliciosas.

Segurança é cheia de deficiências e a única esperança é que elas serão pequenas o suficiente para prevenir que ataques tenham sucesso. Para tornar essas deficiências menores custa tempo, esforço e dinheiro. Além do mais, a implementação de medidas de segurança pode ter um efeito negativo na usabilidade do sistema (Choon & Samsudin, 2003).

Os recursos de uma grade podem ser bastante atrativos (Sardinha et al., 2004) devido às suas capacidades de computação e armazenamento possivelmente elevadas e deveríamos esperar que eles se tornem alvos de atacantes e úteis para intrusos. As deficiências de segurança mencionadas anteriormente sugerem que sistemas de detecção de intrusão (*Intrusion Detection Systems* – IDS) têm um papel importante na gerência de segurança de grades. IDSs são sistemas que detectam violações de segurança em sistemas de informação e respondem enviando notificações de alertas ao gerente de segurança (Allen et al., 2000, Debar et al., 1999). Essas violações podem ser caracterizadas como uso não-autorizado por intrusos ou o abuso de privilégios por usuário legítimos (*insiders*).

IDSs podem ser usados como complementos aos mecanismos de segurança preventivos. Um IDS aumenta o nível de segurança de uma grade, e assim, a disponibilidade, a integridade e a confidencialidade de dados e recursos.

Cenários do funcionamento de um IDS em uma grade são apresentados na Figura 1. O Nodo A e o Nodo B estão protegidos pela infra-estrutura de segurança da grade (*Grid Security Infrastructure* – GSI) e também pelo serviço de detecção de intrusão do IDS. Pela figura, pode-se ilustrar dois cenários onde o IDS age como uma segunda linha de defesa, um *backup* do GSI (Choon & Samsudin, 2003):

- O Intruso 1 ataca o Nodo A pelas deficiências de segurança do GSI. Neste caso, o IDS pode detectar o usuário não autenticado que entrou no Nodo A.
- O segundo cenário é causado pelo descuido de um usuário legítimo na exposição de sua senha durante uma autenticação. O Intruso 2, usando o GSI e a senha do usuário legítimo, mascara-se como um usuário autorizado e acessa o Nodo 2. Isto é mais complicado que o primeiro cenário, pois o intruso acessou a máquina com uma identidade autorizada de usuário. Neste caso, o IDS pode detectar a intrusão checando as atividades do intruso no Nodo 2 e comparar com o perfil do usuário legítimo.

IDSs típicos baseados em *host* e baseados em rede (Debar et al., 1999) podem ser implantados em um ambiente de grade para melhorar sua segurança. No entanto, eles não podem detectar adequadamente intrusões de grade. A detecção dessas intrusões

impõe novos desafios e as tecnologias atuais de detecção de intrusão são limitadas no fornecimento de proteção contra elas. Nesta dissertação é proposto um método de detecção de intrusão em grades que supera essas limitações.

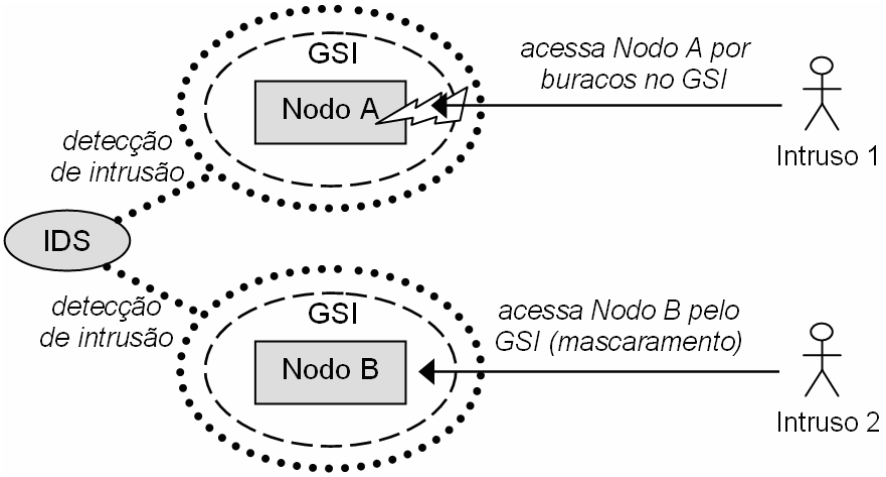

Figura 1. Cenários de um IDS agindo em um ambiente de grade, adaptado de Choon & Samsudin (2003)

1.1 Objetivos

O objetivo geral deste trabalho é responder a seguinte pergunta:

Como elaborar um método de detecção de intrusão em grades computacionais que funcione e que contribua para a área de segurança de sistemas distribuídos?

A partir desta pergunta, várias outras questões surgem. Este trabalho tem como objetivos específicos responder as seguintes questões:

- *O que é uma grade de computadores?*
- *Qual a importância da segurança de grades?*
- *Qual a diferença de um ataque e de uma intrusão?*
- *Usuários legítimos usando uma grade de forma maliciosa são intrusos que devem ser identificados?*

- *Que tipos de ataques e intrusões podem ocorrer em uma grade?*
- *O que diferencia intrusões em grades de intrusões em outros sistemas distribuídos?*
- *O que pode ser afetado em uma grade por causa de uma intrusão?*
- *O que gera a necessidade de um IDS em um ambiente de grade?*
- *Como funcionam os sistemas de detecção de intrusão?*
- *Por que os IDSs existentes não são adequados para proteger grades?*
- *O que é detecção de intrusão em grades?*
- *O que difere um IDS de grade de um IDS típico de host e rede?*
- *O que um IDS de grade precisa para superar as limitações das tecnologias atuais?*
- *Como detectar intrusões em grades?*

1.2 Escopo

Detecção de intrusão em grades de computacionais é um tema que envolve as áreas de sistemas distribuídos e segurança de sistemas de informação. Grades computacionais são uma especialização de sistemas distribuídos e detecção de intrusão é um subconjunto da segurança de sistemas de informação. Sendo assim, detecção de intrusão em grades computacionais é uma intersecção dos dois sub-conjuntos, como podemos ver na Figura 2.

Mais especificamente, nesta dissertação será abordado um método para detecção de intrusão em grades computacionais e a arquitetura de um sistema de detecção de intrusão (IDS) que pode ser usado nesse método. Sendo assim, a criação de uma nova técnica de detecção de intrusão para ser usada pelo IDS está fora do escopo. Já existem muitas técnicas divulgadas que podem ser usadas nesses sistemas, como as técnicas que usam estatísticas, redes neurais ou assinaturas de ataques (Debar, 1999), e não é objetivo deste trabalho criar uma nova ou melhorar o desempenho de alguma já conhecida. Também fogem do escopo a definição de respostas a intrusões de grade (quais as melhores medidas a serem tomadas no caso de uma intrusão detectada), a prevenção contra intrusões por infra-estrutura de segurança (*firewalls*, autenticação,

autorização, controle de acesso, etc.) e a tolerância a intrusões de grade (Sardinha, 2004, Yuanbo, 2005).

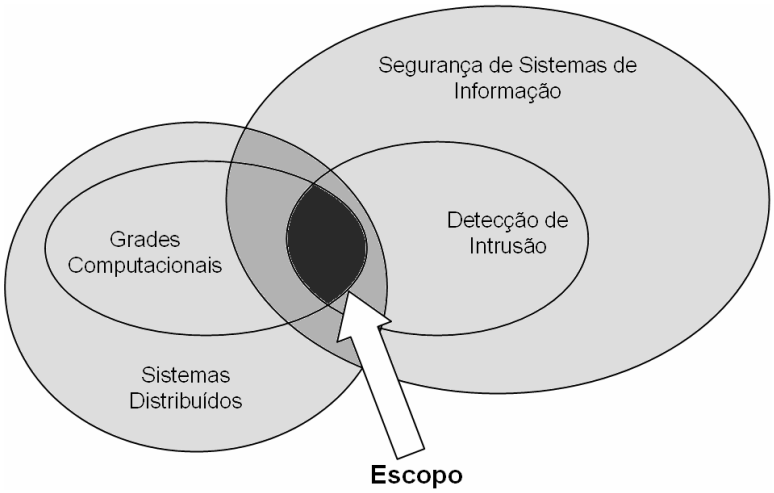

Figura 2. Escopo desta dissertação

1.3 Organização do Trabalho

Este trabalho é organizado da seguinte forma:

- Este primeiro capítulo apresentou uma breve introdução ao problema, os objetivos e o escopo do trabalho.
- O segundo capítulo apresenta uma introdução mais extensa ao problema da detecção de intrusão em grades computacionais. São introduzidos: a computação em grade, sua motivação e o estágio que ela representa na evolução da computação distribuída; a segurança de grades, sua importância e seus desafios; a necessidade do fornecimento de detecção de intrusão em ambientes de computação em grade, trazendo à tona a importância deste trabalho; os sistemas de detecção de intrusão, suas características e o papel que têm na segurança de sistemas de informação; uma classificação de intrusões que podem ocorrer em grades, suas conseqüências e o que diferencia elas de intrusões em outros sistemas distribuídos; e os trabalhos relacionados ao tema, suas características relevantes e deficiências que motivam a proposta deste trabalho.

- O terceiro capítulo aponta as deficiências das tecnologias atuais no fornecimento de proteção a grades contra ataques, lista os requisitos básicos que deveriam ser cumpridos por qualquer IDS de grade e descreve uma proposta de método de detecção de intrusão em grades que cumpre tais requisitos.
- O quarto capítulo apresenta uma solução para a integração de IDSs prevista no método proposto. O uso de protocolos e formatos padronizados é enfatizado.
- No quinto capítulo os mecanismos descritos no Capítulo 4 são validados através de um estudo de caso no qual são realizados experimentos de detecção de intrusão em uma grade computacional simulada.
- O trabalho termina no sexto capítulo com as conclusões finais e uma delineação de trabalhos futuros que podem ser realizados.

2 Detecção de Intrusão em Grades Computacionais

Este capítulo apresenta uma introdução ao problema da detecção de intrusão em grades computacionais. Este tema é relativamente recente e não foram encontradas descrições aprofundadas sobre o problema na literatura.

Inicialmente veremos o que é computação em grade, qual é sua motivação e qual é o estágio que ela representa na evolução da computação distribuída (Seção 2.1). Em seguida veremos qual a importância da segurança de grades, do que consiste e quais são seus desafios. A Seção 2.2 justifica a necessidade do fornecimento de detecção de intrusão em ambientes de computação em grade, trazendo à tona a relevância deste trabalho. Na Seção 2.3 é apresentada uma introdução aos sistemas de detecção de intrusão, suas características e o papel que têm na segurança de sistemas de informação. Na Seção 2.4 são classificadas as intrusões que podem ocorrer em uma grade e são descritas as suas conseqüências e as diferenças que existem entre intrusões em grades e em outros sistemas distribuídos. Na Seção 2.5 são apresentados os trabalhos relacionados ao tema que foram encontrados na literatura, suas características relevantes e suas deficiências que motivam a proposta deste trabalho. Finalmente, na Seção 2.6 é apresentado um resumo deste capítulo.

2.1 Computação em Grade e Segurança

Uma grade de computadores (Buyya, 2002, Foster, 2002, Foster & Kesselman, 2003) é um sistema distribuído que coordena recursos que não estão sujeitos a um controle central. Recursos e usuários que vivem dentro de diferentes domínios administrativos ou de controle podem ser integrados e as questões de segurança, políticas, pagamento e associação são tratadas. Protocolos e interfaces padronizados, abertos e de propósitos gerais são usados para tratar dos problemas de autenticação, autorização e descobrimento e acesso a recursos. Uma grade pode fornecer qualidades de serviço não-triviais relacionadas a, por exemplo, tempo de resposta, vazão,

disponibilidade, confiabilidade, desempenho e segurança e ela pode co-alocar múltiplos tipos de recursos para alcançar as demandas de usuários.

Aplicações que requerem grande poder computacional, capacidade de armazenamento massivo ou preparativos complexos de compartilhamento de recursos com certos requisitos de qualidade de serviço podem se beneficiar de sistemas de grade. Grupos de instituições e individuais de diferentes domínios podem formar organizações virtuais (Foster et al., 2001) e compartilhar recursos por um objetivo comum. Exemplos de recursos que podem ser compartilhados em um ambiente de grade são: computadores pessoais, computadores de alto desempenho, aplicativos de software (na forma de serviços), dispositivos de armazenamento, dispositivos de computação móvel, sensores e instrumentos científicos, como telescópios e aceleradores de partículas.

2.1.1 Evolução e Motivações da Computação em Grade

A expressão "Computação em Grade" (do inglês *Grid Computing*) surgiu no início da década de 90 como uma metáfora a um poder de computação tão acessível quanto a energia elétrica fornecida por uma grade de distribuição de energia elétrica (do inglês *Electrical Power Grid*). Uma grade computacional não disponibiliza somente poder de processamento aos seus usuários, mas qualquer recurso computacional, da mesma forma que a energia elétrica é disponibilizada em uma grade de distribuição: com acesso transparente, barato, onipresente e seguro (Focke, 2004).

Computação em grade é uma fase evolucionária da computação distribuída, por muitos considerada uma tecnologia revolucionária, mas que é o resultado da sofisticação dos recursos de *software* de computação distribuída, pelo avanço das tecnologias de rede e pelo aumento do desempenho e disponibilidade de computadores (Berman et al., 2003, Focke, 2004). Após a criação da *World Wide Web* – WWW em 1989 (Berners-Lee, 1999), a próxima etapa na evolução da Internet pode ser o *World Wide Grid* – WWG, uma associação de uma grande quantidade de computadores conectados com o objetivo do uso colaborativo de suas capacidades para a resolução dos problemas complexos, sendo esta associação feita usando-se diversas abordagens tecnológicas de computação distribuída.

Vista sob um ângulo de paradigmas de computação distribuída, computação em grade aparece como o quarto paradigma (Focke, 2004). O primeiro marco histórico destes paradigmas foi a interligação de todas as máquinas usando os protocolos TCP (*Transmission Control Protocol*) e IP (*Internet Protocol*). O segundo paradigma era interligar todos os documentos juntos usando a WWW (*World Wide Web*), o HTTP (*Hypertext Transfer Protocol*) e a HTML (*Hypertext Markup Language*). O terceiro paradigma é a ligação de todas as aplicações usando o XML (*eXtensible Markup Language*) e *Web Services*. No entanto, o terceiro paradigma não trata apropriadamente da provisão de recursos e da integração transparente de sistemas. O quarto paradigma é a interligação, baseada em tecnologias e protocolos de grade, das aplicações e recursos computacionais através das fronteiras organizacionais.

Em comparação aos duzentos anos de história da grade de energia elétrica, toda a infra-estrutura de comunicação de computadores tem apenas meio século (Buyya, 2002). A Figura 3 apresenta os maiores marcos dos avanços tecnológicos em comunicação e computação que contribuíram para o surgimento da computação em grade.

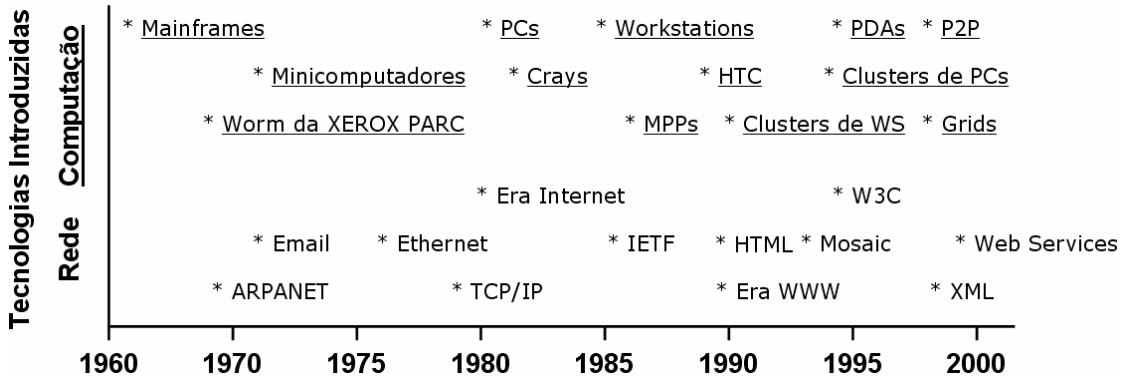

Figura 3. Maiores marcos em tecnologias de redes e computação, adaptado de Buyya (2002)

Os fatores que contribuíram para o surgimento de tecnologias de grade existentes atualmente podem ser divididos em fatores de computação e de rede. Um dos primeiros marcos tecnológicos de rede foi a ARPANET, um modesto projeto da Agência de Projetos de Pesquisa Avançados de Defesa dos Estados Unidos (DARPA) de 1969. Inicialmente apenas 4 nós existiam na rede e em 1970 já eram 30

universidades interligadas. Posteriormente foi transferida a responsabilidade da ARPANET dos interesses militares para os acadêmicos, tornado-se assim o que chamamos hoje de Internet. A expansão da Internet desde então foi surpreendente, como mostra a Figura 4.

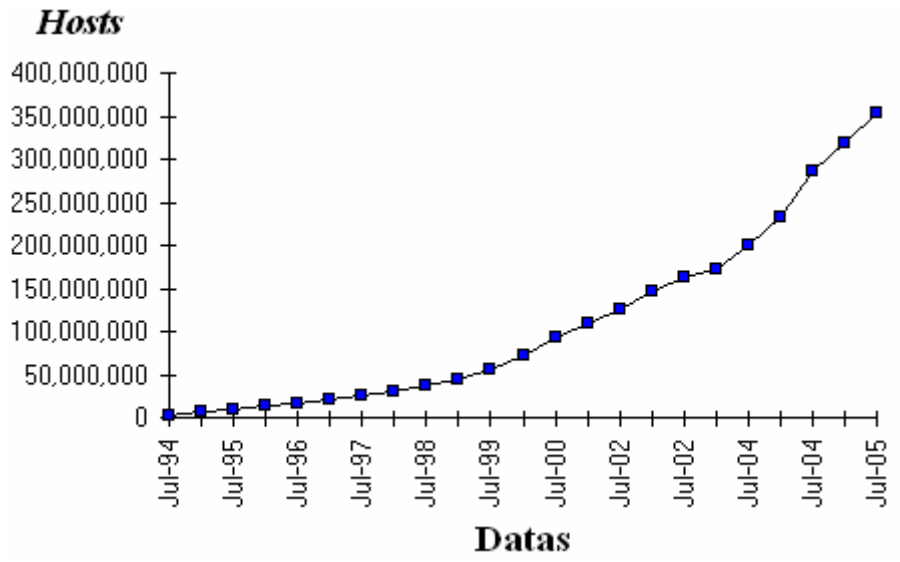

Figura 4. Contagem do número de *hosts* de domínio na Internet, adaptado de ISC (2006)

A evolução das redes de computadores certamente influenciou a evolução da computação, ou seja, contribuiu com os fatores computacionais descritos na Figura 3. Um exemplo disso é a Ethernet (Metcalfe & Boggs, 1976), uma tecnologia de redes locais que possibilitou o arranjo de diversos computadores do tipo PC em *clusters* para processamento de alto desempenho. A existência de *clusters* em várias instituições e da Internet despertaram o interesse para a interligação dessas instituições e seus *clusters* de forma a utilizar o processamento em aplicações de interesse mútuo, sendo estes clusters elementos de um *grid*.

Um exemplo de que o grau de crescimento do poder computacional tem sido exponencial é a regra anunciada por Gordon Moore, conhecida como "Lei de Moore". Em 1965 ele previu que a densidade de transistores em chips semicondutores dobraria a cada 18 meses. Como resultado os microprocessadores têm se tornado menores, mais densos e mais poderosos em termos de instruções por segundo que podem processar.

Hoje em dia esta regra tem se mantido e Gordon Moore está otimista quanto ao assunto, tendo afirmado que a regra se manteria por pelo menos mais dez anos (Moore, 2003). Encontramos computadores pessoais com capacidade de processamento dezenas de vezes maior que supercomputadores de uma década atrás. E isto a um preço acessível, devido ao barateamento dos componentes dos processadores, possibilitando até que sejam agrupados vários processadores, o que deu origem às arquiteturas de processamento paralelo e distribuído atuais.

Apesar do aumento constante da capacidade de microprocessadores, aplicações estão ficando cada vez mais complexas e exigentes em termos de poder de processamento. Grades computacionais são vistas como uma possível solução para esta grande demanda de poder de processamento das aplicações.

2.1.2 Segurança de Grades

Um ambiente seguro para o compartilhamento de recursos é um importante requisito para a aceitação das grades por usuários e provedores de recursos. Segurança e privacidade são preocupações principais nos desenvolvimentos recentes de grades comerciais, já que são componentes chaves para tornar as grades economicamente praticáveis. O escopo das aplicações que executam na grade determina o nível de segurança que deve ser atingido. Aplicações médicas, por exemplo, requerem absoluta confidencialidade, pois a violação de privacidade é irreversível (Navqi & Riguidel, 2005).

À medida que o uso de grades torna-se mais comum, migrando de sistemas experimentais a sistemas de produção, amplifica-se a importância das questões de segurança (Humphrey & Thompson, 2002), assim como a necessidade pelo entendimento e controle da segurança das aplicações de grade.

De diversas formas, argumentavelmente, o desafio mais notável das grades de computadores é o desenvolvimento de um conjunto de mecanismos e políticas amplo para sua segurança. O desafio da segurança de grades fica evidente com seus requisitos inter-domínio únicos (Humphrey et al., 2005). As principais questões estão na autenticação e autorização de usuários a acessar recursos distribuídos e a contenção desses usuários a somente recursos alocados pela grade e não, por exemplo, todos os

recursos de um computador *host*. As atividades que precisam ser seguras em um ambiente de grade são: nomeação (*naming*), autenticação, comunicação, confiança, políticas, autorização e controle de acesso. A confidencialidade e integridade das mensagens que atravessam uma organização virtual e a privacidade dos dados armazenados nos recursos distribuídos precisam ser asseguradas, provedores de recursos precisam acomodar mecanismos de segurança e políticas que não estão estritamente sob seu controle, usuários precisam de nomes definidos globalmente que serão reconhecidos em todos os sítios aos quais eles têm acesso, sítios de grade precisam entrar em relações de confiança com usuários e outros sítios, e usuários precisam de mecanismos para obter autorização de uso de recursos que é consistente através de múltiplos sítios.

Middleware de grade (Asadzadeh et al., 2005) fornecem serviços de segurança baseados em criptografia. O *Globus Security Infrastructure* – GSI (Foster & Kesselman, 1997, Foster et al., 1998), por exemplo, é baseado no *Transport Layer Security* – TLS (Dierks & Allen, 1999), infra-estrutura de chave pública e certificados digitais X.509 (Housley et al., 1999). Ele provê serviços como a inscrição múltipla de um único usuário em múltiplos sítios, criptografia para comunicação segura, autenticação de recursos, delegação de autoridade e confiança. No entanto, serviços adicionais podem ser necessários baseados em experiências reais de grade (Leite et al., 2003) como, por exemplo, detecção de intrusão, a qual aumenta o nível de segurança e, assim, a disponibilidade, a integridade e a confidencialidade de sistemas de grade.

A necessidade do serviço de detecção de intrusão é discutida mais extensivamente na próxima seção.

2.2 A Necessidade da Detecção de Intrusão em Grades

Na evolução das grades computacionais, preocupações com ameaças de segurança foram deixadas de lado no desejo de implementar um sistema computacional para fácil compartilhamento de recursos. Até agora, a tecnologia de grades tem sido pouco utilizada, exceto por certos tipos de usuários, principalmente pesquisadores. Esse público beneficia-se bastante da oportunidade que tem de compartilhar recursos na grade e eles não têm intenção de prejudicar os proprietários dos recursos ou colegas

usuários. Isto com tempo irá mudar, pois o número de pessoas que conhecem as grades está crescendo, assim como os alvos que valem a pena aos intrusos (Navqi & Riguidel, 2005).

Consideráveis pesquisas têm sido conduzidas na área de segurança de sistemas distribuídos (Leite et al., 2003). Mesmo que a evolução desses serviços seja significativa, a evolução de suas contra-partes, como *worms*, vírus e ataques distribuídos por negação-de-serviço (Kendall, 1999), também é. A experiência atual mostra que é muito difícil construir sistemas distribuídos complexos como as grades computacionais de uma maneira totalmente segura, especialmente sendo elas implementadas em uma rede aberta como a Internet.

Os recursos de uma grade podem ser bastante atrativos (Sardinha et al., 2004) devido às suas capacidades de computação e armazenamento possivelmente elevadas e deveríamos esperar que eles se tornem alvos de atacantes e úteis para intrusos. O acesso e o compartilhamento de recursos e a computação colaborativa facilitada por grades amplifica as preocupações sobre intrusões, especialmente em grades de larga-escala (TeraGrid, 2005). Nesse tipo de grade, o considerável poder de computação pode ser usado por um intruso para quebrar senhas, os dispositivos de armazenamento podem ser usados para guardar arquivos ilícitos e as redes com grande largura de banda são ideais para lançar ataques por negação de serviço (Kendall, 1999, Sardinha et al., 2004). Por esses motivos, um princípio de projeto a ser seguido no desenvolvimento de sistemas de grade é assumir que eventualmente alguns deles serão invadidos e construí-los de tal forma que atacantes sejam forçados a empenharem-se significativamente para subverter o sistema.

Não é realista prevenir absolutamente o aparecimento de brechas de segurança, especialmente em sistemas distribuídos complexos como as grades. Mesmo que os serviços de segurança oferecidos por um middleware de grade sejam projetados e implementados cuidadosamente para evitar vulnerabilidades, intrusos podem explorar defeitos em qualquer dos outros componentes envolvidos, como sistemas operacionais, protocolos de comunicação de redes e aplicações que não usam a grade, mas que estão executando no mesmo ambiente. Além do mais, uma grade não pode se defender de senhas roubadas e usuários legítimos que abusam de seus privilégios para executar atividades maliciosas.

Segurança, em geral, pode ser alcançada através da prevenção, preensão, inibição, desvio, detecção e contramedida (Halme & Bauer, 1995). Os serviços de segurança de grades existentes atualmente apenas fornecem prevenção contra violações de segurança. Os defeitos mencionados anteriormente sugerem que sistemas de detecção de intrusão sejam usados como um complemento aos mecanismos de segurança preventivos baseados em criptografia, como uma ferramenta para que os gerentes de segurança aperfeiçoem a segurança geral das grades.

Um IDS de grade eficaz deve detectar ataques nos seus estágios iniciais e despertar alarmes para prevenir maiores danos aos recursos da grade e vazamento de informações sigilosas. Além disso, IDSs podem ajudar na identificação e correção de vulnerabilidades e no arquivamento de informações de segurança, útil quando os gerentes de segurança estão rastreando intrusos e nos seus processos criminais (Sommer, 1999). Esses sistemas são melhor descritos na próxima seção. Na Sub-Seção 2.3.2 são descritas as limitações das infra-estruturas de segurança que apenas fornecem prevenção contra intrusões.

2.3 Sistemas de Detecção de Intrusão

Um IDS é um sistema que detecta, preferivelmente em tempo real, violações de segurança em sistemas de informação e responde enviando notificações de alerta ao gerente de segurança, o qual toma as medidas apropriadas, como desconectar um usuário ou bloquear tráfego de rede. As violações podem ser caracterizadas como uso não-autorizado ou abuso do sistema por usuários legítimos ou intrusos.

IDSs geralmente não tomam medidas preventivas quando um ataque é detectado; eles são agentes reativos ao invés de pró-ativos. Eles fazem o papel de informantes ao invés de oficiais de polícia (Sundaram, 1996). IDSs são similares a alarmes de ladrões e uma analogia pode ser feita (Axelsson, 1999). Os mesmos problemas ocorrem em ambos os sistemas, como os “alarmes falsos” e os intrusos que enganam o sistema. Mas as semelhanças param por aí, pois alarmes de ladrões operam sob uma política de segurança bem mais simples. Tipicamente, não há atividade normal nas premissas do local protegido enquanto o monitoramento está ativado e, assim, qualquer atividade

pode ser considerada suspeita. Se isto fosse verdade em sistemas de computadores e redes de computadores, o problema poderia ser muito mais facilmente resolvido. Infelizmente, sistemas de detecção de intrusão operam em ambientes onde atividades normais acontecem e o problema é ter a capacidade de identificar as poucas atividades maliciosas dentre muitas atividades normais.

Debar et al. (1999) descreve genericamente um IDS como sendo um detector que processa informação vinda do sistema sendo protegido (Figura 5). Esse detector pode lançar sondas para ativar o processo de auditoria e ele usa três tipos de informação: informação de longo-prazo relacionada à técnica usada para detectar intrusões (um banco de conhecimento sobre ataque, por exemplo), informação de configuração sobre o estado atual do sistema e informação de auditoria descrevendo os eventos que estão acontecendo no sistema. A espessura das setas indica a quantidade de informação fluindo entre os componentes.

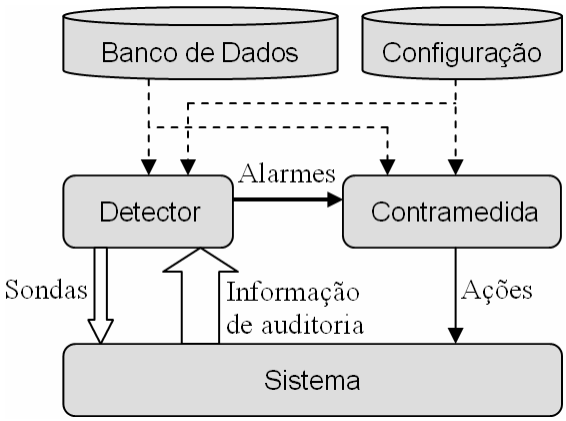

Figura 5. Um sistema de detecção de intrusão simples, adaptado de Debar et al. (1999)

O papel do detector é de eliminar informação não necessária dos registros de auditoria. Ele então apresenta uma visão sintética das ações relacionadas a segurança que foram tomadas durante o uso do sistema ou uma visão sintética do estado de segurança atual do sistema. Uma decisão é então tomada para avaliar a probabilidade de que essas ações ou esse estado podem ser considerados sintomas de intrusão ou vulnerabilidades. O componente de contramedidas pode então tomar ações corretivas tanto para prevenir as ações de serem executadas quanto para mudar o estado do sistema de volta ao estado seguro.

Essa visão genérica é especializada de várias formas pelos muitos IDSs existentes na literatura. Eles podem ser classificados de acordo com uma diversa gama de características, as quais são apresentadas na sub-seção seguinte.

2.3.1 Características dos IDSs

Detecção de intrusão é baseada nas crenças de que o comportamento de um intruso será notavelmente diferente do comportamento de um usuário legítimo e de que muitas ações não-autorizadas são detectáveis (Mukherjee et al., 1994). As técnicas para identificação das intrusões são geralmente classificadas em técnicas baseadas em anomalias e técnicas baseadas em assinaturas, cada uma com suas vantagens e desvantagens (Allen et al., 2000, Axelsson, 1999). Sistemas baseados em anomalias aprendem o comportamento normal ou esperado das entidades (isto é, os usuários) usando o sistema ou rede e tentam identificar divergências de uso. Se o uso está fora de um determinado limiar, uma resposta é disparada. Sistemas baseados em assinaturas ficam atentos a rastros ou padrões conhecidos resultantes do uso malicioso do sistema ou rede que ficam armazenados em um banco de dados de assinaturas de ataques. Dependendo da definição dessas assinaturas, esses sistemas podem às vezes detectar ataques desconhecidos que são similares a ataques conhecidos.

O conceito de detecção de intrusão foi inicialmente proposto por Anderson em 1980, mas não floresceu até 1987 quando Denning publicou seu modelo de detecção de intrusão seminal (Jansen, 2002). Desde então, diversas soluções diferentes ao problema de detecção de intrusão foram desenvolvidas (Allen et al., 2000) e elas podem ser tipicamente caracterizadas como baseadas em *host* ou baseadas em rede. Nos sistemas de detecção de intrusão baseados em *host* (*Host-based Intrusion Detection System* – HIDS), agentes de software localizados em *hosts* individuais coletam e analisam dados de auditoria relacionados ao comportamento dinâmico ou estado dos computadores *host* e nisto estão incluídos: contabilidade de uso de recursos e eventos de segurança potencialmente significativos como, por exemplo, chamadas de sistema e modificações no sistema de arquivos. Intrusões podem também ser detectadas capturando e examinando dados de auditoria de redes, como pacotes de rede e variáveis do *Simple Network Management Protocol* – SNMP (Debar et al., 1999). Sistemas de detecção de

intrusão baseados em redes (*Network-based Intrusion Detection System* – NIDS) podem revelar ataques envolvendo múltiplos *hosts* e ataques à própria rede, os quais podem não ser facilmente reconhecidos examinando-se dados de auditoria de *hosts*.

Além de origens de dados de *hosts* e redes, IDSs também podem examinar dados de auditoria oriundos de registros de aplicações, como no caso de IDSs específicos para servidores de aplicação, e dados oriundos de outros IDSs na forma de alertas. Esses sistemas não identificam ataques diretamente, mas correlacionam informação de várias ferramentas de "baixo-nível".

<table>
<tr><td rowspan="12">Técnica de detecção</td><td rowspan="8">Anomalia</td><td rowspan="4">Auto-aprendizagem</td><td rowspan="2">Não usam séries temporais</td><td>Modelagem por regras</td></tr>
<tr><td>Estatísticas descritivas</td></tr>
<tr><td rowspan="2">Séries temporais</td><td>Redes neurais</td></tr>
<tr><td>Sistemas imunológicos</td></tr>
<tr><td rowspan="4">Programada</td><td rowspan="3">Estatísticas descritivas</td><td>Estatísticas simples</td></tr>
<tr><td>Baseado em regras simples</td></tr>
<tr><td>Limite</td></tr>
<tr><td>Negação de padrão</td><td>Modelagem de séries de estados</td></tr>
<tr><td rowspan="4">Assinatura</td><td rowspan="4">Programada</td><td rowspan="2">Modelagem de estado</td><td>Transição de estado</td></tr>
<tr><td>Rede de Petri</td></tr>
<tr><td colspan="2">Sistemas especialistas</td></tr>
<tr><td colspan="2">Combinação de strings</td></tr>
<tr><td rowspan="3">Comportamento pós-detecção / resposta</td><td rowspan="2">Ativo</td><td colspan="3">Ação de controle sobre o alvo do ataque</td></tr>
<tr><td colspan="3">Ação de controle sobre a origem do ataque</td></tr>
<tr><td colspan="4">Passivo</td></tr>
<tr><td rowspan="4">Origem dos dados</td><td colspan="4">Host</td></tr>
<tr><td colspan="4">Rede</td></tr>
<tr><td colspan="4">Aplicação</td></tr>
<tr><td colspan="4">Alertas de outros IDSs</td></tr>
<tr><td rowspan="2">Tempo de detecção</td><td colspan="4">On-line (tempo real)</td></tr>
<tr><td colspan="4">Off-line</td></tr>
<tr><td rowspan="2">Freqüência de uso</td><td colspan="4">Monitoramento contínuo</td></tr>
<tr><td colspan="4">Análise periódica / batch</td></tr>
<tr><td rowspan="3">Arquitetura de análise</td><td colspan="4">Centralizada</td></tr>
<tr><td rowspan="2">Distribuída</td><td colspan="3">Hierárquica</td></tr>
<tr><td colspan="3">Totalmente distribuída</td></tr>
</table>

Tabela 1. Classificação dos sistemas de detecção de intrusão, adaptado de Brandão (2004)

Muitas outras características podem ser usadas para classificar sistemas de detecção de intrusões. A Tabela 1 foi sintetizada a partir do modelo de classificação de Axelsson (2000), das características de IDSs propostas por Debar et al. (1999) e das

informações de McHugh (2001). Além das técnicas de detecção e origens dos dados, um IDS pode ser caracterizado pelo seu comportamento quanto à intrusão após sua detecção, o tempo de detecção, a freqüência de uso e o tipo de arquitetura.

Um IDS pode funcionar em tempo real ou não (tempo de detecção) e pode funcionar continuamente ou ser ativado somente em determinados momentos (freqüência de uso). A partir do momento que uma intrusão é detectada, o IDS pode agir de forma passiva, enviando um alerta a um operador, ou de forma ativa, tomando alguma ação para bloquear a intrusão ou minimizar suas conseqüências (comportamento pós-detecção). Tanto a coleta dos dados quanto a análise dos mesmos podem ser feitas de forma centralizada e distribuída (arquitetura de análise).

Cabe ressaltar que em geral os IDSs usam exclusivamente uma técnica de detecção e uma origem de dados. No entanto, existem pesquisas que mostram as vantagens de abordagens híbridas, como no caso de IDSs baseados em anomalias com geração automática de assinaturas ou assinaturas pré-programadas e IDSs com múltiplas origens de dados (Axelsson, 1999, Kandula et al., 2002, Porras & Neumann, 1997).

2.3.2 O Papel dos IDSs

A abordagem convencional para segurança de sistemas de computadores e redes é construir um escudo protetor ao seu redor. Esse escudo deve certificar que, através de técnicas de controle de acesso, usuários se identifiquem e se autentiquem sempre que precisarem entrar no sistema e também para prevenir o vazamento de informação. Numa abordagem baseada apenas em prevenção há uma série de limitações (Mukherjee et al., 1994):

- É difícil, talvez impossível, construir um sistema útil totalmente seguro. Isto é, a possível existência de alguma falha de projeto em um sistema com grande número de componentes não pode ser excluída. Também não pode ser excluída a possível ocorrência de falhas administrativas, tanto em sua prática quanto na configuração de equipamentos e definição de políticas.

- Em muitos casos é impraticável se desfazer de toda uma infra-estrutura (possivelmente não-segura) de computadores, redes e software em favor de sistemas novos e seguros.
- A filosofia de segurança baseada em prevenção reprime as atividades de um usuário, sendo sua produtividade inferior em comparação a sistemas onde a operação é mais "aberta".
- Sistemas baseados em criptografia não podem se defender contra chaves perdidas ou roubadas e senhas quebradas.
- Finalmente, um sistema seguro pode ainda ser vulnerável a usuários internos legítimos (*insiders*) fazendo mau uso de seus privilégios.

Sistemas de detecção de intrusão podem ser úteis como complementos aos mecanismos preventivos, detectando e acionando contramedidas sempre que as propriedades de segurança dos cada vez mais complexos sistemas de informação estiverem sob ataque.

Apesar de imaturas e de eficácia limitada, as tecnologias de detecção de intrusão podem ter um papel significante em uma arquitetura de segurança (McHugh et al., 2000). Além de servir como um complemento aos mecanismos de prevenção, um IDS pode fornecer avisos de que o sistema está sob ataque, mesmo não sendo o sistema vulnerável ao ataque. Esses avisos podem ajudar os usuários a modificarem sua postura defensiva para aumentar a resistência ao ataque.

Outro fator que indica a necessidade de mecanismos complementares aos preventivos é o aumento da sofisticação dos ataques e a diminuição dos conhecimentos técnicos necessários para conduzi-los. Segundo Allen et al. (2000), na década de 1980 os intrusos eram especialistas nos sistemas, tinham alto nível de perícia e desenvolviam individualmente seus métodos de invasão. Hoje em dia, qualquer um pode atacar de forma automatizada sítios de Internet usando ferramentas prontamente disponíveis. Como podemos ver na Figura 6, enquanto usuários experientes estão ficando mais espertos e seus ataques mais sofisticados, o conhecimento requerido para que o intruso novato possa lançar um ataque conhecido está diminuindo.

Para ter sucesso, um IDS deve ser projetado de acordo com uma clara definição das ameaças e das possíveis violações de segurança que o sistema monitorado está

sujeito e dos seus alvos potencialmente vulneráveis. Essas definições, no contexto de sistemas de grade, são apresentadas na próxima seção.

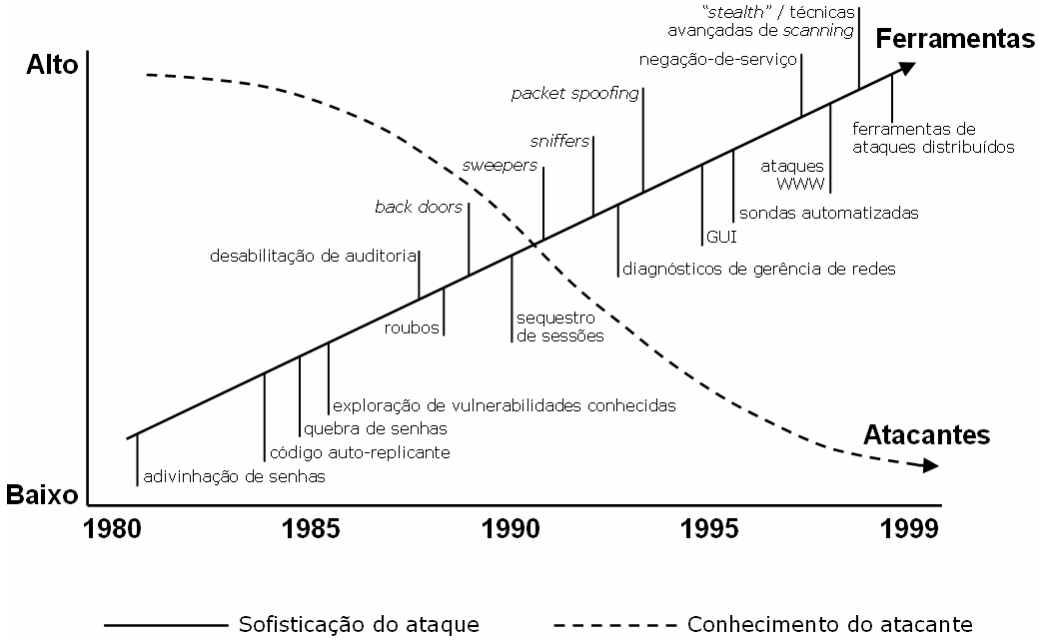

Figura 6. Sofisticação do ataque vs. conhecimento técnico do intruso, adaptado de Allen et al. (2000)

2.4 Intrusões em Grades

Uma grade está sujeita a diversas ameaças de segurança acidentais e intencionais, incluindo ameaças à integridade, confidencialidade e disponibilidade dos seus recursos, dados e infra-estrutura distribuídos por várias localidades. Além disso, quando uma grade com amplo poder computacional e capacidade de armazenamento é usada para propósitos maliciosos por uma entidade mal-intencionada, a própria grade é uma ameaça contra a sociedade (Navqi & Riguidel, 2005). Ameaças intencionais são impostas por intrusos externos e usuários legítimos da grade que abusam de seus privilégios usando a grade para propósitos não-intencionados e isto é considerado, neste trabalho, comportamento intrusivo a ser detectado. Seus possíveis desejos são o comprometimento dos aspectos de segurança de recursos específicos, de domínios

administrativos, de organizações virtuais, de uma grade como um todo ou de sistemas externos, no caso em que a grade em si é a ameaça.

Uma intrusão consiste de um ataque explorando uma deficiência de segurança e uma conseqüente transgressão, a qual resulta na violação da política de segurança explícita ou implícita do sistema (Lindqvist & Jonsson, 1997). Embora uma intrusão conote um ataque bem-sucedido, IDSs também tentam identificar ataques que não resultam em comprometimentos de segurança, ou seja, ataques mal-sucedidos (Allen et al., 2000). "Ataques" e "intrusões" são comumente considerados sinônimos no contexto de detecção de intrusão.

A infra-estrutura de rede de uma grade, sendo um importante componente do ambiente de computação, pode ser o objeto de um ataque. Aplicações de grade executando em *hosts* comprometidos também são uma preocupação de segurança. Neste trabalho, ataques contra qualquer rede ou *host* participando em uma grade são considerados ataques contra aquela grade, porque eles podem afetar diretamente ou indiretamente seus aspectos de segurança. Sistemas de grade são suscetíveis a todos os ataques típicos de redes e computadores, assim como métodos específicos de ataque, por causa dos seus novos protocolos e serviços. Os alvos possivelmente vulneráveis são: a pilha de protocolos; os dispositivos de rede; os processos executando em espaço de núcleo, tais como *daemons* de sistemas operacionais; e os processos executando fora do espaço de núcleo, como os middleware de grade, aplicações de grade e quaisquer aplicações não-grade executando tanto com privilégios de usuário como de super-usuário.

Classificações das típicas intrusões em computadores *host* e intrusões de redes existem na literatura (Kendall, 1999, Shields, 2002, Weaver et al., 2003). Os trabalhos relacionados ao tema, discutidos a seguir na Seção 2.5, não descrevem quais são os tipos de intrusões que podem acontecer em grades. A classificação de intrusões em grades elaborada e adotada neste trabalho é dada a seguir:

(a) *Acesso não-autorizado*: Uma invasão no sistema cometida por alguém que se mascara como um usuário legítimo da grade. Esta intrusão é possível pela obtenção da senha do usuário através do roubo, quebra por força-bruta, adivinhação ou a exposição descuidada da senha pelo próprio usuário. Romper o

serviço de autenticação é outra possibilidade e isso resulta em rastros de ataque deixados na localidade do serviço.

(b) *Mau uso*: Isto pode ser a conseqüência de um (a) acesso não-autorizado ou o abuso de privilégios por um usuário legítimo e geralmente resulta em uma anomalia observável do comportamento do usuário. O que é um mau uso dos recursos de uma grade depende das políticas que foram definidas e estas deveriam considerar utilização agressiva, erros cometidos pelos usuários e comportamento malicioso. Como comportamento malicioso pode-se citar a sondagem de nodos da grade, a submissão, para execução na grade, de tarefas oportunistas que se comportam como *worms* espalhando-se automaticamente, atividades ilegais, etc.

(c) *Ataque de grade*: Ataques executados com a ajuda de ferramentas ou *scripts* de ataque (*exploit scripts*) e que exploram vulnerabilidades específicas existentes nos protocolos, serviços e aplicações de uma grade. Eles podem aparecer na forma de ataques por negação-de-serviço, sondas e *worms* (Kendall, 1999, Shields 2002, Weaver et al., 2003) e podem deixar vestígios em diversas localidades da infra-estrutura da grade. Cabe ressaltar que ainda não foram divulgados trabalhos aprofundados sobre ataques específicos de grade e, até a data atual, não foram encontrados levantamentos ou taxonomias desses ataques.

(d) *Ataques específicos de host ou rede*: enquanto (a), (b) e (c) são intrusões específicas de grade, estes são quaisquer dos ataques típicos a computadores *host* e redes (Kendall, 1999).

Qualquer intrusão tem uma conseqüência. As conseqüências de intrusões de grade são descritas a seguir:

- *Uso não-intencionado de recursos*: Intrusos e usuários legítimos maliciosos consomem recursos e aumentam o uso da rede com tráfego de dados não-desejável que poderiam, do contrário, serem usados para os propósitos intencionados da grade.
- *Perda de disponibilidade*: A prevenção ou atraso do acesso autorizado a recursos da grade.

- *Perda de confidencialidade e/ou integridade*: Quando dados sensíveis de usuários ou sistema são acessados e/ou manipulados. Quando configurações de dispositivos de rede, tais como roteadores, são manipulados, a intrusão também pode resultar em perda de disponibilidade.

Uma das diferenças entre intrusões em grades e em outros sistemas distribuídos está no fato de que suas velocidades, suas conseqüências e seus danos são potencialmente maiores nas grades, já que a agregação massiva de recursos, o amplo acesso de usuário e a alocação de recursos eficiente e automatizada fornecidos por elas podem ser usados por intrusos para sua vantagem própria. Ou seja, caso um intruso consiga acessar a grade e submeter tarefas, suas tarefas maliciosas terão os mesmos benefícios e facilidades que são fornecidos a tarefas não-maliciosas.

Segundo Vijayan (2004), apesar das grades não criarem fundamentalmente novos riscos de segurança, elas servem para amplificar alguns deles. A maioria dos problemas que os usuários têm de lidar em um ambiente de grade são similares aos que eles enfrentam em ambientes não-grade. Mas esses problemas têm uma significância maior nos ambientes de grade por causa das premissas fundamentais das grades: acesso, compartilhamento e computação colaborativa.

2.5 Motivação e Trabalhos Relacionados

Na seção anterior foram descritas quatro tipos de intrusões que podem violar a segurança de grades: (a) acesso não-autorizado, (b) mau uso, (c) ataque de grade e (d) intrusão de *host*/rede. Para evitar as conseqüências não-desejáveis dessas intrusões, HIDS e NIDS típicos (Seção 2.3) podem ser implantados em um ambiente de grade e fornecer proteção contra ataques que exploram vulnerabilidades em seus nodos (*hosts*) e redes. Esta solução não é completa, já que ela fornece proteção contra (d) ataques de *host* e rede, mas não contra (a, b, c) intrusões específicas de grades. Os bancos de dados de assinaturas dos IDSs típicos podem ser atualizados para permitir a identificação de rastros ou vestígios de (a) acessos não-autorizados e (c) ataques de grade deixados em *hosts* e pacotes de rede. Esta também não é uma solução completa, pois (c) ataques de

grade podem deixar seus vestígios em mais de uma localidade e esses ataques podem somente ficar evidentes pela correlação dos vestígios identificados pelos IDSs. Além disso, um HIDS é incapaz de detectar apropriadamente usuários de grade cometendo (b) mau uso, porque eles analisam o comportamento de usuários em seus contextos locais e, já que usuários de grade podem usar múltiplos recursos de diferentes domínios ao mesmo tempo ou consecutivamente, a análise deve ser feita no escopo da grade como um todo. Por esses motivos, uma abordagem diferente ao problema é necessária para superar as deficiências.

A necessidade por sistemas de detecção de intrusão baseados em grades (*Grid-based Intrusion Detection Systems* – GIDS) foi inicialmente mencionada por Liabotis et al. (2003), Talnar et al. (2003) e Leite et al. (2003), apesar de soluções para o problema não terem sido apresentadas. Uma solução eficiente e escalável para armazenar e acessar dados de auditoria de rede coletados por sensores Snort (Snort, 2005) instalados em vários pontos de uma grade foi proposta por Kenny & Coghlan (2005), mas não houve menção sobre como usar os dados para identificar intrusões.

Choon & Samsudin (2003) descreveram uma arquitetura de IDS baseado em grade que consiste de agentes localizados em nodos de uma grade responsáveis por coletar e enviar dados de auditoria de *host* a um servidor de armazenamento e análise, mas visto que é sabido que os IDSs consomem tempo de processamento e espaço de armazenamento consideráveis (Allen et al., 2000), sua solução centralizada é dificilmente escalável com o número de nodos sob análise. O *Grid Intrusion Detection Architecture* (GIDA) proposto por Tolba et al. (2005a, 2005b) resolve o problema da escalabilidade distribuindo o problema da detecção de intrusão em vários servidores de análise.

Tanto o trabalho de Choon & Samsudin quanto o de Tolba et al. concentra-se na detecção de anomalias na interação dos usuários da grade com os recursos, as quais são resultados de (b) mau uso. Mas eles carecem detecção apropriada de (a) acesso não-autorizado e (c) ataques de grade, já que esses podem não causar anomalias e são mais facilmente detectados por sistemas baseados em assinaturas (Seção 2.3). Além disso, nenhuma das duas arquiteturas fornecem proteção contra os (d) ataques típicos de *host* e rede.

Fang-Yie et al. (2005a, 2005b) propôs um IDS chamado *Performance-based Grid Intrusion Detection System* (PGIDS) no qual nodos de uma grade são alocados por meio de balanceamento de carga para analisar tráfego de rede coletado em busca de ataques de negação-de-serviço de rede. O sistema usa os abundantes recursos da grade para detectar pacotes de intrusões, mas ele não detecta ataques contra a própria grade e somente examina a rede, agindo como um NIDS ao invés de um GIDS.

As deficiências das soluções disponíveis para detecção de intrusão em grades e a necessidade do fornecimento desse serviço, como discutido anteriormente na Seção 2.2, motivam a proposição de uma nova abordagem. Uma análise do problema e dos requisitos a serem cumpridos por qualquer GIDS é fornecida no próximo capítulo.

2.6 Resumo

Como visto na Seção 2.1, grades computacionais são ferramentas usadas para facilitar o compartilhamento de recursos distribuídos a serem usados por certas aplicações que requerem grande poder computacional, capacidade de armazenamento massivo ou preparativos complexos de recursos. Entre suas características estão a falta de um controle central, o uso de protocolos e interfaces padronizados e o fornecimento de qualidade de serviço não-trivial. A computação em grade foi motivada pela evolução das tecnologias computacionais e de redes e aparece como uma fase evolucionária da computação distribuída, e não como uma tecnologia revolucionária.

Várias questões são tratadas por uma grade, sendo a segurança uma das principais. O compartilhamento seguro de recursos é um importante requisito para a aceitação das grades por usuários e provedores de recursos e o seu desafio está nos requisitos inter-domínio únicos.

Middleware de grade fornecem serviços preventivos de segurança, mas experiências reais de computação em grade indicam que serviços complementares, como a detecção de intrusão, podem ser necessários. Na Seção 2.2 vimos que as ameaças de segurança a grades estão ganhando mais atenção e que recursos de uma grade podem ser bastante atrativos a pessoas mal-intencionadas, especialmente os recursos de uma grade de larga-escala. Mesmo que os serviços de segurança em

sistemas distribuídos estejam evoluindo, as suas contra-partes (ataques) também estão. Não é realista prevenir absolutamente o aparecimento de brechas de segurança em sistemas distribuídos complexos como as grades e caso o nível de segurança demandado seja alto, fica justificada a necessidade do uso da detecção de intrusão.

Na Seção 2.3 vimos que sistemas de detecção de intrusão (IDS) são ferramentas para detecção de violações de segurança. As técnicas de detecção que esses sistemas usam geralmente são baseadas na identificação de anomalias ou assinaturas de ataques observáreis nos sistemas monitorados. Os IDSs são geralmente classificados pela origem dos dados que analisam. Os dois tipos principais são os IDSs que analisam dados vindos de computadores *host* (HIDS) e IDSs que analisam tráfego de rede (NIDS). No entanto, pesquisas recentes mostram vantagens em IDSs que usam abordagens híbridas, tanto na origens de dados quanto nas técnicas de análise.

Para ter sucesso, um IDS deve ser projetado de acordo com uma clara definição das ameaças e das possíveis violações de segurança que o sistema monitorado está sujeito e dos seus alvos potencialmente vulneráveis. Sendo assim, na Seção 2.4 foi apresentada uma classificação das intrusões que podem ocorrer em uma grade. Vimos que “ataques” e “intrusões” são comumente considerados sinônimos no contexto de detecção de intrusão e que as intrusões de grade podem ser classificadas em: (a) acesso não-autorizado, (b) mau uso, (c) ataque de grade e (d) ataques específicos de *host* ou rede. As conseqüências dessas intrusões são: o uso não-intencionado de recursos, a perda da disponibilidade de recursos e a perda de confidencialidade e/ou integridade de dados de usuários ou de sistema.

Em comparação com outros sistemas distribuídos, intrusões em grades têm velocidade, conseqüências e danos potencialmente maiores, já que os mesmos benefícios e facilidades que são fornecidos a tarefas não-maliciosas serão fornecidos a usuários e suas tarefas maliciosas submetidas à grade. Dessa forma, ao invés de criar novos riscos, as grades amplificam os riscos de segurança já conhecidos.

Na Seção 2.5 foram descritos os trabalhos relacionados ao tema de detecção de intrusão em grades computacionais. Como vimos, todas as soluções existentes na literatura apresentam deficiências e não fornecem proteção contra todas as intrusões (a), (b), (c) e (d); fato que motiva a proposição de uma nova abordagem neste trabalho, descrita no capítulo seguinte.

3 Proposta

Neste capítulo são apresentadas uma análise do problema, que define as deficiências das tecnologias atuais no fornecimento de detecção de intrusão em grades computacionais e os requisitos necessários para superá-las, e uma proposta de um método para detecção de intrusão que supera as limitações existentes.

3.1 Análise do Problema

Um IDS não tem conhecimento da motivação ou das ferramentas empregadas por um intruso para conduzir seu ataque. Para ser bem-sucedido, um IDS precisa focar na avaliação das informações que estão facilmente disponíveis, isto é, informações observáveis nos alvos (Undercoffer et al., 2004). Detecção de intrusão em grades é um processo que envolve a coleta de informações de segurança observáveis em suas redes e nodos (computadores *host*) e a identificação, baseada na avaliação e correlação dos dados coletados, de ataques contra todos os alvos possivelmente vulneráveis, assim como de anomalias na interação de usuários de grade com recursos.

3.1.1 Deficiências das Tecnologias Atuais

Uma análise dos dados de auditoria no escopo local de um nodo ou de uma rede para detecção de intrusão em uma grade não é adequado. Segundo Kannadiga & Zulkernine (2005), um atacante pode invadir múltiplos nodos de um sistema distribuído enquanto mantém um nível de atividade intrusiva baixo o suficiente para não ser identificado por monitores de intrusão executando em cada um desses nodos. No entanto, o nível agregado de atividade intrusiva determinado pela soma dos níveis de atividades intrusivas nos nodos individuais pode ser alto o suficiente para gerar um alerta.

Como discutido na Seção 2.5, as tecnologias atuais de detecção de intrusão falham em fornecer proteção contra todas as intrusões que podem violar a segurança de

uma grade. A implantação dos típicos NIDS e HIDS em um ambiente de grade melhora a segurança, mas não é uma solução completa porque eles não têm a capacidade de detectar apropriadamente intrusões específicas de grade.

HIDS e NIDS analisam dados de auditoria nos escopos locais dos *hosts* e redes onde foram implantados e isto não é suficiente para detectar intrusões em uma grade, já que os usuários de uma grade podem estar usando vários *hosts* e redes ao mesmo tempo ou consecutivamente e ataques de grade podem deixar rastros em várias localidades da grade. Isto fica mais claro com os seguintes exemplos:

- *Um usuário conhecido há muito tempo tem o comportamento típico de submeter aplicações a uma grade que utilizam de 10 a 20% da capacidade computacional de 5 hosts quaisquer de seus 100. Em um determinado momento este usuário submete uma aplicação que utiliza 22% da capacidade computacional de 90 hosts. Os HIDS instalados em cada host aceitariam o pequeno desvio de comportamento local, mas o GIDS alertaria que a anomalia pode ser resultado de uma intrusão, já que o usuário passou a usar quase todos os nodos da grade e normalmente usava só 5.*

- *Um determinado ataque de grade tem como característica o uso dos recursos do tipo A e B que estão disponíveis no host 1 e o recurso do tipo C que está disponível no host 2. Como os rastros ou vestígios do ataque ficam registrados nas variáveis de sistema dos hosts 1 e 2, para identificar o ataque o GIDS deve correlacionar as variáveis de sistema de ambos os hosts, algo que um HIDS ou um NIDS não foi projetado para fazer.*

Um nodo pode não estar dedicado à grade, sendo seus recursos compartilhados entre os usuários locais e os usuários de grade externos alocados a recursos que pertencem a aquele nodo. Portanto, a interação ou o comportamento dos usuários devem ser examinados no escopo dos nodos individuais, no caso de usuários locais, e no escopo da grade como um todo, no caso de usuários da grade.

Também como visto na Seção 2.5, as arquiteturas de GIDS disponíveis (Choon & Samsudin, 2003, Tolba et al., 2005a, Tolba et al., 2005b) são projetadas para detectar

apropriadamente anomalias de comportamento de usuários de uma grade, mas são deficientes na detecção de ataques a computadores *host* e redes da grade e ataques específicos de grade que não causam anomalias.

3.1.2 Requisitos de um GIDS

Este trabalho convenciona que os seguintes três requisitos básicos devem ser satisfeitos por um sistema de detecção de intrusão baseado em grade:

(x) *Abrangência*: Deve fornecer detecção de (a) acesso não-autorizado, (b) mau uso, (c) ataque de grade e (d) ataque de *host*/rede (Seção 2.4).
(y) *Escalabilidade*: Deve ser escalável com o número de recursos e usuários da grade.
(z) *Compatibilidade com a grade*: Deve ajustar-se ao ambiente de grade e beneficiar-se do mesmo.

Enquanto as soluções atuais para o problema da detecção de intrusão em grades têm o intuito de satisfazer os requisitos de (y) escalabilidade (Kenny & Coghlan, 2005, Tolba et al., 2005a, Tolba et al., 2005b) e (z) compatibilidade com a grade (Choon & Samsudin, 2003, Tolba et al., 2005a, Tolba et al., 2005b), elas carecem na (x) abrangência de intrusões.

Outros requisitos também importantes são:

- *Heterogeneidade*: Deve lidar com vários tipos de recursos.
- *Flexibilidade*: Deve permitir o uso de diferentes técnicas de detecção de intrusão (Sub-Seção 2.3.1).

Na próxima seção é proposta uma abordagem ao problema que satisfaz esses requisitos.

3.2 Método

Nas seções seguintes são apresentadas uma proposta de método para detecção de intrusão em grades computacionais e um exemplo de GIDS que pode ser usado no método. Em seguida é mostrado como esse sistema cumpre os requisitos básicos de detecção de intrusão em grades delineados na seção anterior. Finalmente são apresentadas algumas considerações sobre o método e a arquitetura do exemplo.

3.2.1 Um Método para Detecção de Intrusão em Grades

Detecção de intrusão é uma disciplina imatura que ainda está por estabelecer um *framework* aceito por todos (Allen et al., 2000). Novas abordagens de IDS são publicadas com freqüência e existem eventos especializados no assunto (RAID, 2004).

Para detecção de intrusão em grades computacionais proponho um novo método onde o GIDS é um componente de alto-nível que utiliza a funcionalidade de HIDS e NIDS – chamados aqui de IDSs de baixo nível – fornecida através de comunicação inter-IDS. Isso torna possível o re-uso de software de detecção de intrusão já existente, evitando re-implementação de funcionalidade.

A integração do GIDS com os IDSs de baixo-nível é o ponto central deste método e é ilustrada pela Figura 7. Nesse método, para alcançar o nível de segurança desejado para uma grade, HIDS e/ou NIDS são instalados em certos nodos e domínios de rede da grade e trabalham de forma integrada com o GIDS enviando informações relevantes para detecção de intrusões.

Para alcançar o nível máximo de segurança, ou seja, para alcançar uma proteção em todos os alvos potencialmente vulneráveis de uma grade contra todos os tipos de intrusões descritos na Seção 2.4, cada nodo e cada domínio de rede da grade devem ter um IDS de baixo nível instalado. Os vários NIDS localizados em cada domínio de rede capturam dados de auditoria de rede e procuram por anomalias de protocolo e rastros de ataques existentes nos pacotes de rede. Cada nodo da grade tem um HIDS instalado que coleta e examina dados de auditoria de *host* para identificar evidências deixadas por ataques e anomalias de uso de recursos causadas por usuários locais. O GIDS usa dados

de auditoria (i) enviados pelos IDSs de baixo-nível para identificar ataques de grade e para comparar o comportamento de usuários da grade com seus perfis históricos previamente construídos. O gerente de segurança da grade é (ii) alertado sempre que uma intrusão de grade for detectada pelo GIDS ou um alerta for (iii) enviado pelos IDSs de baixo-nível.

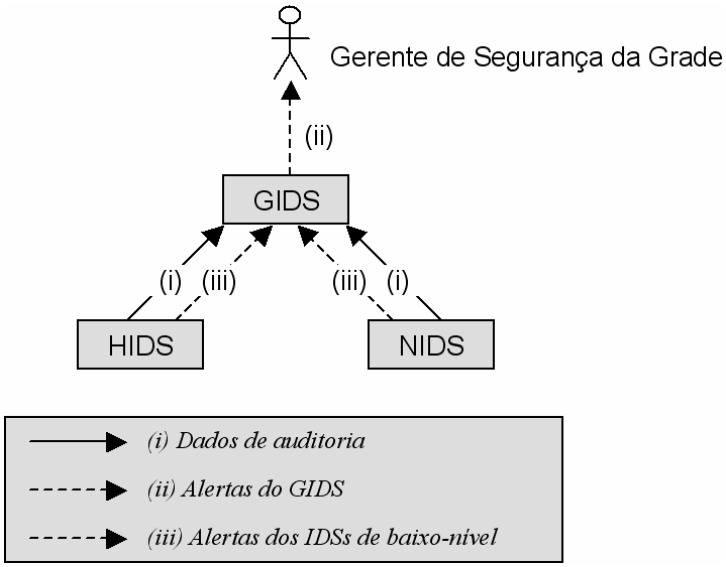

Figura 7. Integração do GIDS com IDSs de baixo-nível

No próximo capítulo é apresentada uma solução para a integração dos IDSs que é viabilizada por comunicação inter-IDS padronizada (Wood & Erlinger, 2002, Feinstein et al., 2002, Curry & Debar, 2005). Estando a integração viabilizada, resta sabermos como que um GIDS integrado com IDSs de baixo nível pode cumprir os requisitos básicos de detecção de intrusão em grades delineados na Sub-Seção 3.1.2: (x) abrangência, (y) escalabilidade e (z) compatibilidade com a grade. Na sub-seção seguinte é descrito um exemplo de GIDS que pode ser usado neste método para cumprir tais requisitos.

3.2.2 Um Exemplo de GIDS

Na Figura 7 da sub-seção anterior o HIDS e o NIDS foram representados de forma abstrata, já que suas arquiteturas variam bastante (Debar et al., 1999). A Figura 8

mostra a arquitetura de um exemplo de GIDS que serve como uma visão mais real do GIDS abstrato representado na Figura 7.

Neste exemplo, o GIDS é composto de Agentes, Analisadores e um Escalonador. A organização dos componentes HIDS e NIDS é ilustrativa e as informações de auditoria que eles (i) enviam aos Agentes GIDS são (iv) armazenadas em Bancos de Dados de Informações da Grade. Cada vez que um usuário acessa a grade, o Escalonador GIDS (v) consulta seu perfil armazenado em um banco de dados e, dependendo do poder de computação demandado para a análise dos dados de auditoria, (vi) submete Tarefas Analisadoras a nodos com recursos computacionais disponíveis. Elas são submetidas a nodos que tem também localizações propícias, onde as transferências de dados possivelmente volumosas entre tarefas e bancos de dados não resultem em sobrecargas na rede.

As tarefas (vii) interagem com os bancos de dados para analisar comportamento dos usuários e atualizar seus perfis e são também responsáveis por (viii) correlacionarem os (iv) dados de auditoria armazenados para identificar ataques de grade.

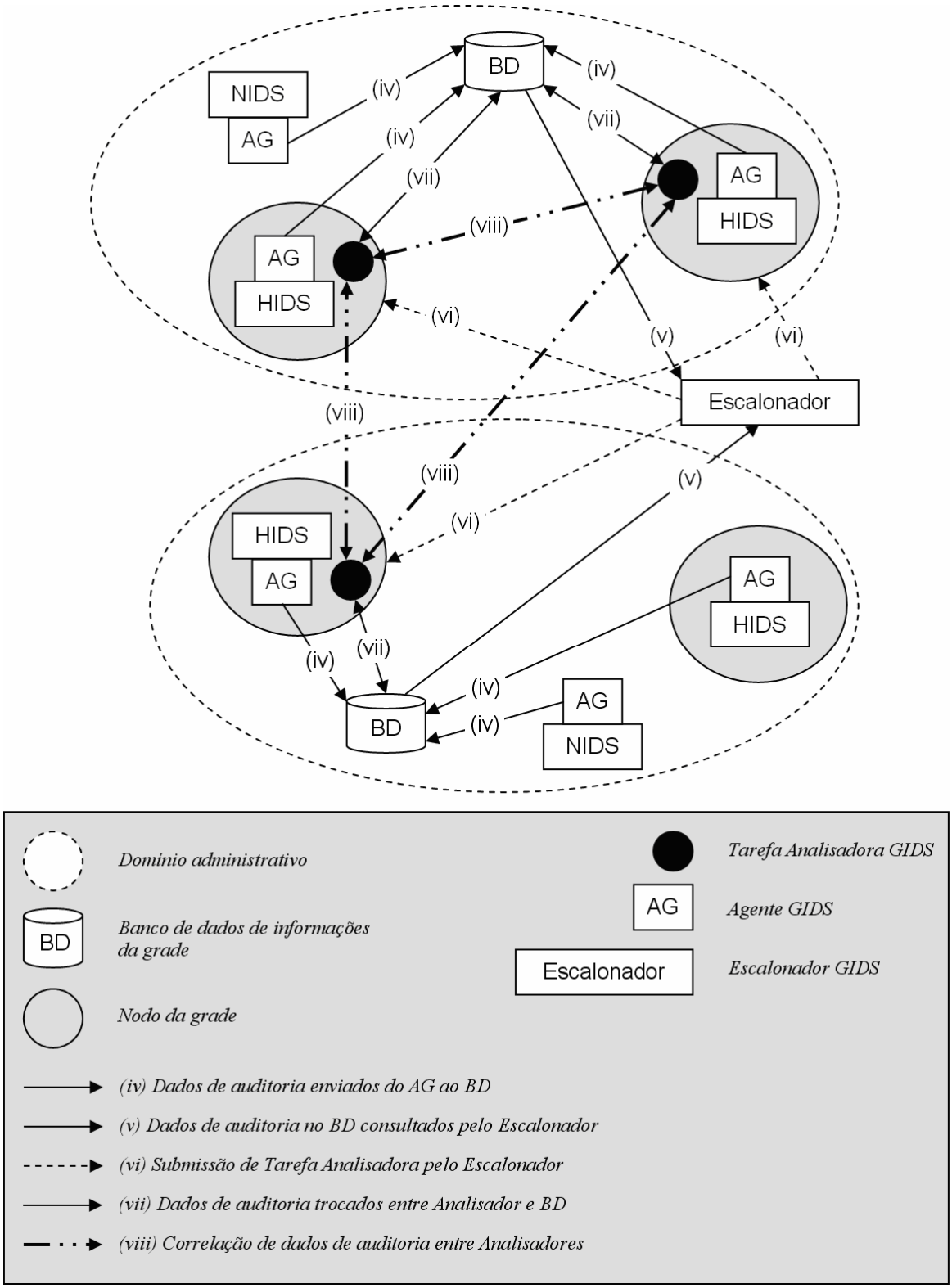

Figura 8. Arquitetura de um exemplo de GIDS

Para mostrar como esse GIDS satisfaz o requisito de (x) abrangência, considere um cenário onde uma grade é protegida por ele e um intruso segue esses passos:

(1) O intruso lança um ataque de *buffer overflow* (Kendall, 1999) contra um processo de um sistema operacional (SO) executando em um nodo da grade. O ataque é bem-sucedido e ele então adquire a capacidade de executar código arbitrário.
(2) Agora com privilégios de super-usuário do SO, ele executa um *script* de ataque que explora uma vulnerabilidade que permite ele personificar (Kendall, 1999) um usuário com privilégios de usuários de grade, ganhando acesso facilitado à muitos nodos da grade.
(3) Continuando sua atividade maliciosa, ele usa vários nodos da grade para executar uma aplicação distribuída.
(4) A aplicação dispara um ataque coordenado de negação-de-serviço de rede contra um alvo externo à grade.

O primeiro passo caracteriza uma (d) intrusão de *host* detectável por um HIDS. Supondo que ela não é detectada, o intruso prossegue ao segundo passo, o qual caracteriza um (c) ataque de grade e um conseqüente (a) acesso não-autorizado, ambos detectáveis pelo GIDS pela análise dos dados de auditoria e alertas enviados pelos HIDS e NIDS. Se não impedido nesse ponto, o intruso chega ao terceiro passo, onde o GIDS compara o comportamento de sua aplicação com o perfil histórico do usuário que ele personificou para poder identificar (b) mau uso. Se de alguma maneira o GIDS falha em identificar uma anomalia de comportamento, no quarto passo o NIDS é responsável por detectar os vestígios do (d) ataque de negação-de-serviço de rede. Conclusivamente, neste cenário o sistema cobre as intrusões (a), (b), (c) e (d), satisfazendo o requisito de (x) abrangência.

O sistema é projetado para distribuir o problema da detecção de intrusão dentre seus componentes de forma a alcançar (y) escalabilidade e, já que o sistema se beneficia da grade consumindo seus recursos, ele alcança (z) compatibilidade com a grade.

O método não impõe o uso de uma técnica de detecção de intrusão específica e as tarefas analisadoras podem fazer análise sobre os dados de auditoria usando qualquer

técnica como, por exemplo, estatísticas ou redes neurais. Desta forma, o exemplo também cumpre o requisito de flexibilidade.

Quanto ao requisito de heterogeneidade, os Agentes GIDS é que são responsáveis por coletar dados de auditoria junto aos recursos e IDSs de baixo nível e esconder a heterogeneidade deles do resto do sistema.

3.2.3 Considerações

Um GIDS pode fazer sua análise de dados de auditoria de forma centralizada em um servidor dedicado a esta tarefa ou pode distribuir a análise, como no exemplo descrito na Sub-Seção 3.2.2, onde a análise é distribuída em tarefas analisadoras. Segundo Kannadiga (2005), os problemas nos modelos de IDS onde a análise é centralizada são:

- A adição de novos *hosts* resulta em um aumento significativo na carga do servidor central. Como resultado, isto torna o IDS não-escalável.
- A comunicação dos coletores de dados com o servidor central pode sobrecarregar partes da rede.
- Alguns dos IDSs centralizados tem componentes específicos de alguma plataforma, enquanto que IDSs distribuídos geralmente tem componentes que funcionam em múltiplas plataformas.

A distribuição dos componentes previne gargalos de desempenho na análise dos dados e pontos de falha únicos. Sendo assim, o Escalonador único do exemplo de GIDS da Sub-Seção 3.2.2 representa um ponto fraco da arquitetura.

Abordagens hierárquicas de detecção de intrusão similares a apresentada na Sub-Seção 3.2.1 já foram desenvolvidas. O DIDS (Snapp, 1991), por exemplo, foi um dos primeiros IDSs hierárquicos e distribuídos a ser publicado. Ele usa o HIDS Haystack (Smaha, 1988) para detectar ataques localmente em *hosts* do ambiente onde foi instalado e o NIDS NSM (Heberlein, 1990) para monitorar a rede. Ambos relatam

informações ao controlador central do DIDS, que faz a análise e correlação das informações.

Assim como o DIDS, que analisa informações enviadas pelo Haystack e pelo NSM, o GIDS também se comporta como uma ferramenta de alto-nível que analisa e correlaciona informações enviadas por ferramentas de mais baixo nível. O DIDS, no entanto, não tem o mesmo objetivo que o GIDS tem de detectar intrusões que podem afetar uma grade e ele concentra-se apenas na proteção de sistemas de informação com múltiplos *hosts* contra os (d) ataques de *host* e rede.

Como um (a) acesso não-autorizado pode ser o resultado de um ataque ao serviço de autenticação de uma grade ou também o resultado de, por exemplo, uma senha roubada, nos próximos capítulos o (a) acesso não autorizado é considerado um (c) ataque de grade, já que o roubo de uma senha dificilmente será identificado por um IDS.

3.3 Resumo

Neste capítulo foi apresentada uma análise do problema que existe no fornecimento de detecção de intrusão em grades computacionais. As soluções disponíveis não fornecem proteção contra todos os tipos de intrusões em grades e foi convencionado que os seguintes requisitos deveriam ser cumpridos por um sistema de detecção de intrusão baseado em grade: abrangência, escalabilidade, compatibilidade com a grade, heterogeneidade e flexibilidade.

Em seguida, foi descrita a proposta de um método de detecção de intrusão em grades computacionais onde o GIDS é um componente de alto nível que trabalha de forma integrada com HIDS e NIDS analisando e correlacionando dados de auditoria e alertas enviados por eles, chamados aqui de IDSs de baixo nível. Nesse método, o nível de segurança desejado é alcançado pela instalação desses IDSs em nodos e domínios de rede da grade. Para alcançar o nível máximo de segurança, cada nodo e cada rede deve ter um desses IDSs instalado.

Foi também descrita a arquitetura de um exemplo de GIDS que pode ser usado no método e foi mostrado como que ele cumpre os requisitos básicos de detecção de intrusão em grades computacionais. O GIDS descrito distribui o problema da detecção

de intrusão dentre seus componentes e usa os próprios recursos da grade para satisfazer os requisitos de escalabilidade e compatibilidade e, com a ajuda de um cenário fictício de intrusões, foi mostrado como que ele fornece proteção contra todas elas, satisfazendo o requisito de abrangência.

Como o método não impõe o uso de uma técnica de detecção específica, o GIDS também cumpre o requisito de flexibilidade. E como os Agentes do GIDS podem lidar com vários tipos de recursos e IDSs, escondendo esses detalhes do resto do sistema, o exemplo cumpre o requisito de heterogeneidade.

Foi apontado que há vantagens na coleta e análise de dados distribuída do exemplo de GIDS. Se o sistema fosse centralizado, poderiam aparecer problemas de escalabilidade e sobrecarga de rede. Uma abordagem centralizada também pode resultar em um ponto central de falha, apesar de que no GIDS há um ponto de falha único: o escalonador de tarefas analisadoras.

Finalmente, coube ressaltar que abordagens hierárquicas de detecção de intrusão similares a abordagem do método já foram desenvolvidas. No entanto, com outros objetivos.

Para viabilizar a integração do GIDS com os IDSs de baixo nível, determinados protocolos, formatos e mecanismos podem ser utilizados. No próximo capítulo é descrita uma solução que faz uso deles.

4 Integração de IDSs para Segurança de Grades Computacionais

Não existe um padrão para construção de sistemas de detecção de intrusão e vários caminhos podem ser seguidos para que a proposta do método de detecção de intrusão em grades computacionais descrito no capítulo anterior seja implementada. O objetivo deste capítulo é apresentar uma das possíveis soluções, a qual é focada no uso de protocolos e formatos oriundos de esforços internacionais de padronização amplamente divulgados pela comunidade de pesquisa em segurança de redes e computadores.

O método prevê a integração de vários IDSs de baixo nível e o GIDS. Esta integração é feita através da troca de informações entre os sistemas. Quais informações são enviadas pelos IDSs, quando são enviadas, como são correlacionadas e quais mecanismos são sugeridos para o envio estão entre as questões abordadas nas seções seguintes.

A Seção 4.1 apresenta os requisitos da troca de informações entre os IDSs. A Seção 4.2 descreve como os padrões que estão sendo desenvolvidos pelo *Intrusion Detection Working Group* – IDWG (IDWG, 2005) podem ser usados na integração e a Seção 4.3 detalha o envio dos registros de auditoria usando esses padrões. Finalmente, na Seção 4.4 é apresentado o resumo deste capítulo.

4.1 Requisitos para Integração de um GIDS com outros IDSs

A integração de um GIDS com IDSs de baixo nível (HIDS e NIDS) para detecção de intrusões em grades pode ser alcançada como proposto no Capítulo 3 se os seguintes requisitos forem cumpridos pelos sistemas envolvidos:

(a) Envio de alertas dos IDSs de baixo nível ao GIDS sobre intrusões detectadas localmente;

(b) Envio de alertas dos IDSs de baixo nível ao GIDS sobre rastros ou vestígios de ataques de grade;
(c) Envio de registros de auditoria dos HIDS ao GIDS;
(d) Comunicação padronizada.

Os IDSs de baixo nível devem (a) alertar o GIDS sobre quaisquer intrusões detectadas localmente em seu domínio de atuação, já que o gerente de segurança da grade deve ser informado pelo GIDS de qualquer violação de segurança em redes ou *hosts* que possam afetar a política de segurança da grade. Alertas sobre quaisquer vestígios de ataques que possam ser correlacionados pelo GIDS com outros vestígios a fim de detectar ataques específicos de grade também devem ser (b) enviados pelos IDSs.

Para identificar o mau uso cometido por usuários de uma grade, o GIDS deve analisar o comportamento desses usuários e esta análise é feita sobre dados de uso dos recursos. Fontes de auditoria baseadas em *hosts* são a única forma de recolher informações sobre as atividades dos usuários na grade. Assim, os HIDS são os responsáveis por (c) enviar registros de auditoria ao GIDS utilizando-se de um suporte de gravação não ambígua de identidades de usuários contra o uso de recursos. Os NIDS não participam nisso porque suas fontes de auditoria são as redes.

Como os IDSs de baixo nível podem ser heterogêneos e escreverem seus alertas em diferentes formatos, fica difícil agregar informações mais precisas para processamento automático sem que haja um padrão na comunicação destes alertas. A interoperabilidade entre sistemas de detecção distintos exige que os mesmos "falem a mesma língua", ou seja, possuam formas padronizadas de comunicação e integração (Brandão, 2004, Brandão et al., 2005). Uma das principais características das grades computacionais é o uso de interfaces e protocolos padronizados, abertos e de propósito-geral (Foster, 2002). Por esses motivos, fica determinado aqui o requisito de (d) comunicação padronizada entre os NIDS, HIDS e GIDS.

Na próxima seção é descrito como os padrões do IDWG podem ser utilizados para cumprir os requisitos (a), (b), (c) e (d).

4.2 Integração Usando os Padrões do IDWG

A segurança de sistemas vem ganhando atenção nos últimos anos, principalmente com o crescimento do comércio eletrônico e das fraudes eletrônicas. O incremento no interesse pela segurança dos sistemas também é refletido nos governos e nas organizações de padronização internacionais (Brandão, 2004).

Esforços de padronização para sistemas de detecção de intrusão estão sendo desenvolvidos pelo *Internet Engineering Task Force* – IETF (IETF, 2005) desde o ano de 1999. O grupo de trabalho IDWG do IETF vem desenvolvendo formatos de dados e procedimentos para o compartilhamento de informações de interesse entre diferentes IDSs, componentes e plataformas de gerência em ambientes heterogêneos. O compartilhamento é feito através do envio de alertas formatados de acordo com o *Intrusion Detection Message Exchange Format* (IDMEF) (Curry & Debar, 2005). O formato das mensagens é independente do protocolo de comunicação, mas o *Intrusion Detection Exchange Protocol* (IDXP) (Feinstein et al., 2002) é sugerido pelo IDWG para o envio desses alertas. Recentemente, o formato *Intrusion Detection Response Exchange Format* (IDREF) (Silva & Westphall, 2005) foi proposto para a formatação de mensagens de resposta aos alertas. Tanto as mensagens IDMEF quanto as IDREF são codificadas em *Extensible Markup Language* – XML (Bray, 2005).

O trabalho do IDWG foi construído parcialmente em cima dos desenvolvimentos do *Common Intrusion Detection Framework* (CIDF) (Kahn et al., 1998), o qual começou a ser desenvolvido em 1997, mas está parado desde o início do ano 2000 (Sang-Kil et al., 2003). O abandono do CIDF é o principal motivo para a adoção do IDMEF e do IDXP neste trabalho.

Nas sub-seções seguintes veremos como o GIDS se encaixa no modelo de detecção de intrusão do IDWG e como o IDMEF e o IDXP podem ser usados nas interações entre os IDSs.

4.2.1 GIDS no Modelo do IDWG

No documento de requisitos atual do IDWG (Wood & Erlinger, 2002) é descrita a visão de IDS adotada pelo grupo (Figura 9). Nesta visão, um IDS é composto por um ou mais sensores que relatam informação a um analisador. As (i) atividades realizadas na fonte de dados geram dados brutos, como pacotes de rede, registros de auditoria de sistemas operacionais e registros de aplicações, que são captados pelos sensores. Os (ii) eventos importantes captados pelos sensores são enviados para verificação no analisador. Se os eventos forem sensitivos, isto é, indicarem uma possível intrusão, o analisador gera (iii) alertas IDMEF para o gerenciador.

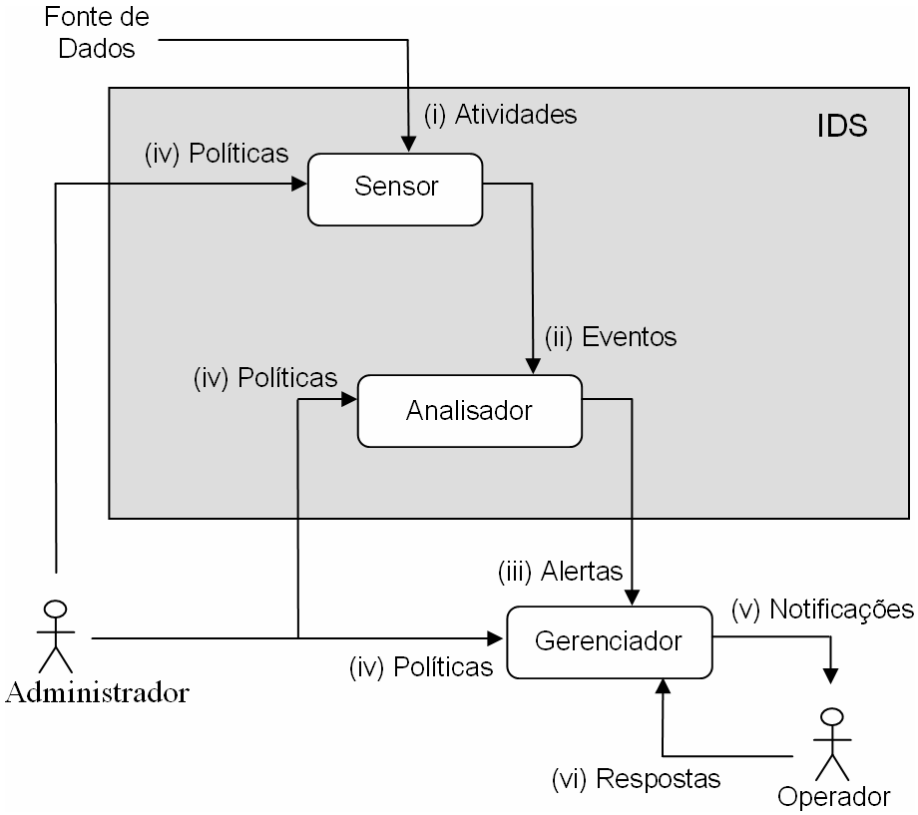

Figura 9. Modelo de IDS do IDWG, adaptado de Wood & Erlinger (2002)

O papel humano em um IDS é cumprido por operadores e administradores. O administrador é responsável por definir as (iv) políticas de segurança do sistema, as quais são aplicadas nos sensores, analisador e gerenciador. Os operadores recebem (v)

notificações sobre alertas do gerenciador, sendo desejável que (vi) respostas sejam acionadas sempre que ocorrerem essas suspeitas de atividades maliciosas.

Cabe ressaltar que nem todos os IDSs têm todos esses componentes descritos. Alguns combinam componentes em um módulo único e outros têm múltiplas instâncias de cada módulo.

Além das fontes de dados descritas anteriormente, um IDS pode também fazer análise de alertas enviados por outros IDSs. Qualquer sistema de detecção de intrusão que gere algum tipo de informação que será correlacionada com outras para detectar novas intrusões é visto como um sensor no modelo do IDWG. Assim, de acordo com o modelo, IDSs de baixo nível, como os HIDS e NIDS, podem ser vistos como sensores de um GIDS.

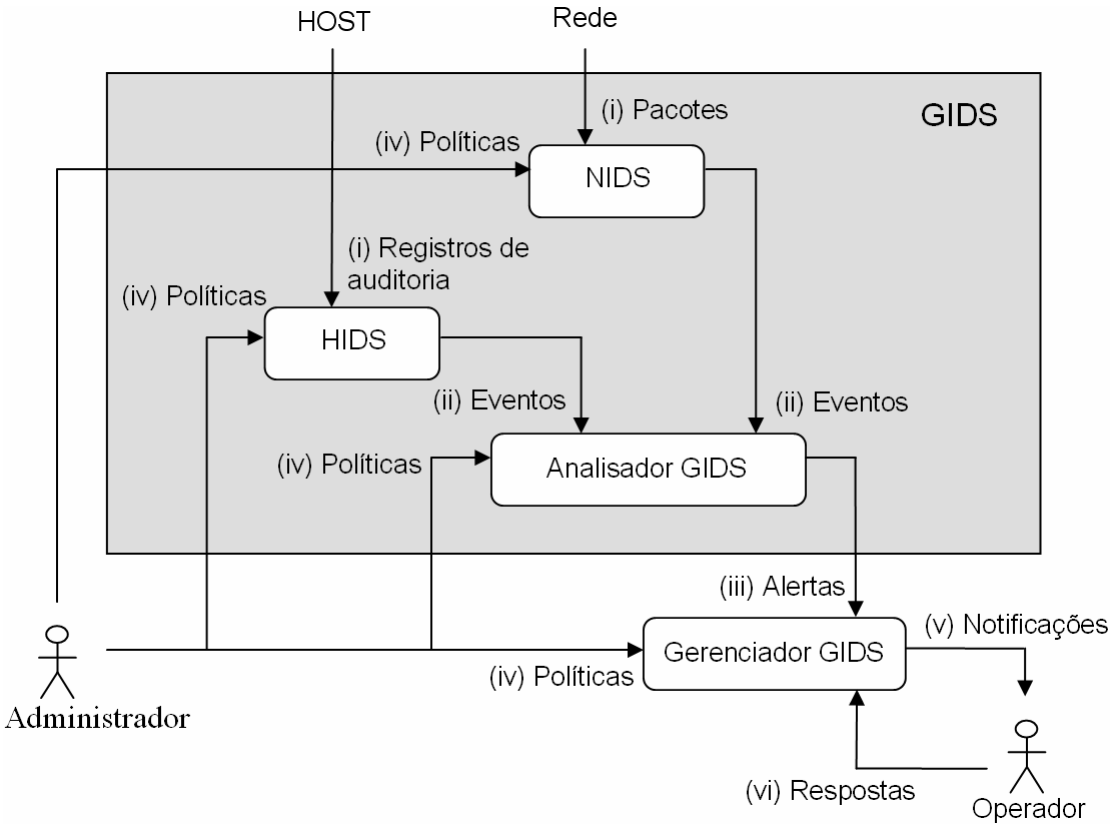

Figura 10. GIDS no modelo do IDWG

Na Figura 10 é apresentado um GIDS segundo a visão de IDS do IDWG. Neste cenário, os IDSs de baixo nível funcionam como sensores captando (i) informação de suas respectivas fontes, processando essas informações com seus próprios analisadores

e gerenciadores e enviando (ii) eventos formatados como alertas IDMEF ao analisador do GIDS. O analisador do GIDS, por sua vez, analisa e correlaciona os alertas recebidos e, quando necessário, gera novos (iii) alertas IDMEF para o gerenciador do GIDS.

O GIDS trabalha de forma integrada aos outros IDSs, consumindo e correlacionando eventos de segurança oriundos dos vários sensores distribuídos pela grade. Além de funcionarem como sensores do GIDS, os IDSs de baixo nível cumprem seu papel detectando intrusões nos seus escopos, analisando eventos de segurança oriundos dos *hosts* e redes onde foram implantados.

4.2.2 Interações do GIDS com outros IDSs

As interações entre os usuários da grade, os recursos, os IDSs de baixo nível e o GIDS são mostradas na Figura 11. O usuário e sua aplicação (i) interagem com o *Broker* de Recursos da Grade (BRG) para negociar os requisitos de processamento da aplicação e outros parâmetros, como por exemplo parâmetros de qualidade de serviço, prazos e custos. O BRG comunica-se com Provedores de Serviço de Grade (PSG) para encontrar nodos adequados onde a aplicação possa ser executada e, então, (ii) submete as tarefas da aplicação do usuário para processamento em nodos do PSG escolhido. Mais informações sobre o funcionamento de um BRG e um PSG podem ser encontradas em Buyya (2002).

Os nodos provêem o serviço do PSG executando as tarefas e o Medidor de Recursos da Grade (MRG) mensura os recursos consumidos enquanto as tarefas são processadas. O MRG (iii) extrai do sistema operacional dados sobre o uso dos recursos por intermédio do *middleware* da grade, como por exemplo os *middleware* Globus e Gridbus (Asadzadeh et al., 2005). Os dados são filtrados pelo MRG e usados na geração de Registros de Uso de Recursos – RUR (Mi-Young, 2005) padronizados e independentes dos SOs, permitindo que sejam utilizados em toda uma grade. Mais informações sobre o funcionamento de um MRG podem ser encontradas em Barmouta & Buyya (2003).

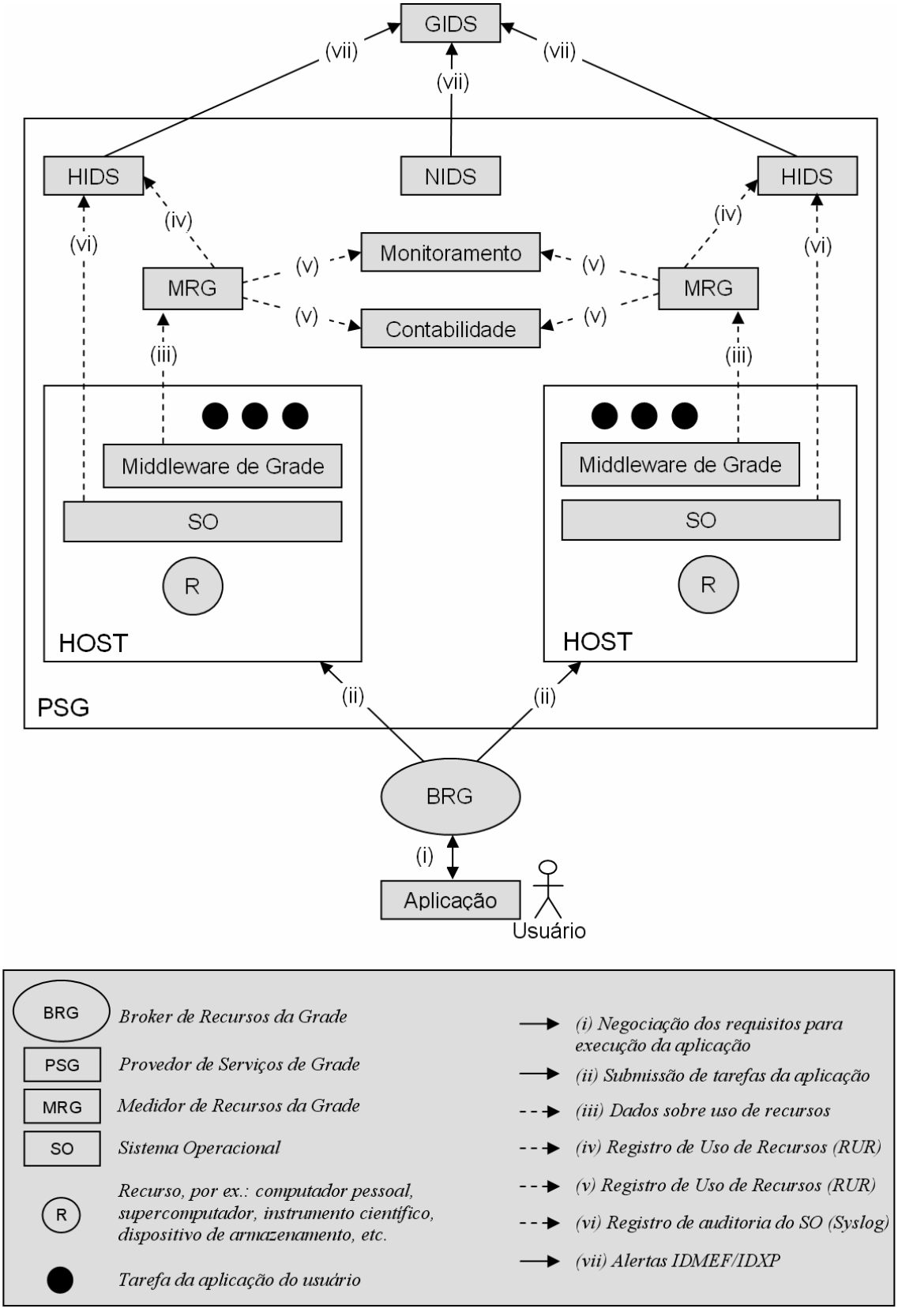

Figura 11. Coleta de dados e interações dos IDSs

Os RURs são enviados ao (iv) HIDS e aos (v) outros sistemas que possivelmente estarão funcionando no PSG, como sistemas de monitoramento (Balaton & Gombas, 2003) e contabilidade (Lim et al., 2005). O HIDS, além de receber dados do MRG, também (vi) recebe registros de auditoria do SO para que possa alertar sobre todos os tipos de intrusões que deixem rastros em *hosts* (Seção 2.4). Cada sistema operacional cria registros de auditoria de sua maneira e uma forma padronizada e popular para o envio destes é o protocolo syslog (Debar et al., 1999).

O GIDS, por sua vez, (vii) recebe e analisa os alertas IDMEF enviados através do protocolo IDXP pelos diversos HIDS e NIDS instalados nos nodos e redes dos PSGs existentes na grade. O conteúdo desses alertas é referente aos ataques típicos de host e rede, vestígios de possíveis ataques específicos de grade e registros de uso de recursos (RURs) de usuários para identificar mau uso.

4.2.3 Cumprimento dos Requisitos de Integração

A interação entre os IDSs, como discutido na sub-seção anterior, cumpre os requisitos (a), (b), (c), e (d) discutidos na Seção 5.1 da seguinte forma:

- Os HIDS e os NIDS alertam o GIDS sobre quaisquer ataques detectados ou vestígios deixados por ataques em nodos ou redes dos PSGs, cumprindo os requisitos (a) e (b). As interações envolvidas nisto são: (vi) e (vii).
- Os HIDS convertem os RURs enviados pelos medidores de recursos em registros de auditoria formatados como alertas IDMEF e os enviam ao GIDS em uma freqüência determinada, cumprindo o requisito (c). As interações envolvidas nisto são: (iii), (iv) e (vii).
- Todos os alertas são formatados de acordo com o IDMEF e enviados pelo protocolo IDXP, os quais ainda não são padrões do IETF, mas que estão em desenvolvimento ativo. O registro de uso de recursos (RUR), usado na comunicação entre MRG e HIDS, é um formato sendo desenvolvido pelo *Global Grid Forum* – GGF (GGF, 2005). O syslog, usado na transmissão de registros de

auditoria pelo sistema operacional, é um protocolo padronizado do IETF que está atualmente sendo revisado (Gerhards, 2005). Todos esses fatores contribuem para o cumprimento do requisito (d).

Como proposto no capítulo 3, o GIDS tem a função de avisar o gerente de segurança da grade sobre os ataques típicos de *hosts* e redes, assim como correlacionar dados enviados pelos vários IDSs de baixo nível para detectar mau uso e ataques específicos de grade.

Ataques a *hosts* e redes podem ser detectados localmente pelos HIDS e NIDS usando as técnicas existentes (Allen et al., 2000), sem a participação do GIDS.

O mau uso cometido por um usuário pode ser detectado pela análise e correlação dos registros de auditoria referentes ao seu uso recente dos recursos em comparação com seu perfil histórico de uso da grade.

Alguns dos ataques específicos de grade podem ser detectados localmente pelos IDSs de baixo nível e outros podem somente ficar evidentes pela correlação de vários vestígios identificados por IDSs distintos (Seção 2.5). No entanto, como a pesquisa em detecção de intrusão em grades computacionais é recente, ainda não foram divulgados trabalhos aprofundados sobre ataques específicos de grade (Naqvi & Riguidel, 2005, Tolba et al., 2005b). A determinação dos tipos de ataques específicos de grade e a correlação de dados necessária para detectá-los foge do escopo deste trabalho. Assim, a detecção de ataques específicos de grade não é explorada aqui.

A detecção de ataques a hosts e redes fica a cargo dos HIDS e NIDS e, assim, também não é explorada neste trabalho. Já a detecção de mau uso é responsabilidade do GIDS. Na próxima seção é descrito o uso do IDMEF para o envio de registros de auditoria dos HIDS ao GIDS para a detecção do mau uso.

4.3 Envio de Registros de Auditoria

Para que o analisador do GIDS seja flexível (Capítulo 3), ele deve poder aplicar diversos tipos de técnicas de análise sobre os registros de auditoria para detecção de anomalias de uso como, por exemplo, técnicas que usam redes neurais ou estatística

(Seção 2.3). Isto implica que os registros de auditoria enviados pelos HIDS sejam independentes da técnica de análise a ser usada pelo GIDS.

Os recursos de uma grade computacional geralmente são heterogêneos, podendo variar de computadores de uso geral, como PCs, clusters e supercomputadores; a computadores especializados, como dispositivos de armazenamento, sensores, instrumentos científicos e dispositivos móveis (Seção 2.4). Isto implica que os registros de auditoria sejam também genéricos, abrangendo diversos tipos de recursos.

O comportamento de um usuário pode ser medido ao longo do tempo por vários parâmetros. Exemplos de parâmetros incluem: o horário de início e término de sessões de usuários e parâmetros de consumo dos recursos por aplicações submetidas (Buyya, 2002), como o tempo de CPU, memória principal, armazenamento secundário e canais de entrada/saída utilizados. Outros parâmetros mais especializados podem ser considerados para cada tipo de recurso.

Todos esses parâmetros estão incluídos nos registros de uso de recursos (RUR) gerados pelos MRGs, os quais definem dados para medição de uso de recursos em cenários computacionais elaborados, diferente dos cenários dos esforços iniciais da computação em grade, que eram em sua maioria acadêmicos e exploratórios (Barmouta & Buyya, 2003).

Um RUR é um documento XML que descreve detalhes sobre o uso de recursos por uma tarefa já finalizada que foi submetida para processamento na grade por um determinado usuário. A Tabela 2 apresenta os campos essenciais do RUR agrupados em: detalhes do usuário, detalhes da tarefa e detalhes do recurso. Além dos campos essenciais, novos parâmetros específicos de certos recursos podem ser abrigados, pois a natureza do documento permite que ele seja estendido para isso.

É tarefa dos HIDS obterem os RURs dos MRGs (suas fontes de dados) e gerarem registros de auditoria de cada usuário no formato IDMEF para o GIDS, sendo que é necessário um consenso prévio entre esses IDSs sobre as unidades de medição de cada parâmetro.

A principal aplicação do IDMEF é na comunicação de alertas entre os componentes de análise e o componente de gerenciamento do sistema de detecção de intrusão. Porém, outras aplicações também são possíveis (Brandão, 2004). O IDMEF pode encapsular qualquer tipo de dados, inclusive dados binários. A vantagem é que

várias informações podem ser normalizadas e autenticadas, melhorando a confiabilidade do modelo e facilitando a correlação das mesmas com dados provenientes de outros sensores.

Agrupamento	Campo	Descrição
Detalhes do usuário	Identificação	Identificador único do usuário na grade
	Nome do host / endereço de IP	Nome / endereço de IP do host a partir do qual o usuário submete a tarefa
Detalhes da tarefa	Identificação	Identificador único da tarefa (combinação do identificador local da tarefa no recurso e o identificador de tarefa atribuído pelo BRG)
	Nome da aplicação	Nome da aplicação
	Data/hora de início	A hora em que a tarefa começa a ser executada
	Data/hora de término	A hora em que a execução da tarefa é completada
	Status	Estado da tarefa no seu término
Detalhes do recurso	Identificação	Identificador único do recurso
	Nome do host / endereço de IP	Nome / endereço de IP do host onde a tarefa foi executada
	Tipo de host	Tipo do recurso (opcional)
	Tempo de execução *(walltime)*	Tempo decorrido na execução da tarefa
	Tempo de CPU	Tempo de CPU usado
	Memória principal	Quantidade máxima de memória utilizada
	Armazenamento secundário	Armazenamento em disco utilizado
	Atividade de rede	Quantidade de dados transferidos pela interface de rede do recurso durante a execução da tarefa

Tabela 2. Conjunto de campos essenciais proposto pelo GGF para o RUR, adaptado de Lim et al. (2005)

O modelo de dados de uma mensagem IDMEF consiste de um conjunto de classes que descrevem seus dados de forma segmentada. O modelo é definido através de DTD (*Document Type Definition*) XML (Bray, 2005) e pode ser representado em um diagrama de classes UML (Rumbaugh et al., 1998). As principais classes e associações do modelo são apresentadas na Figura 12.

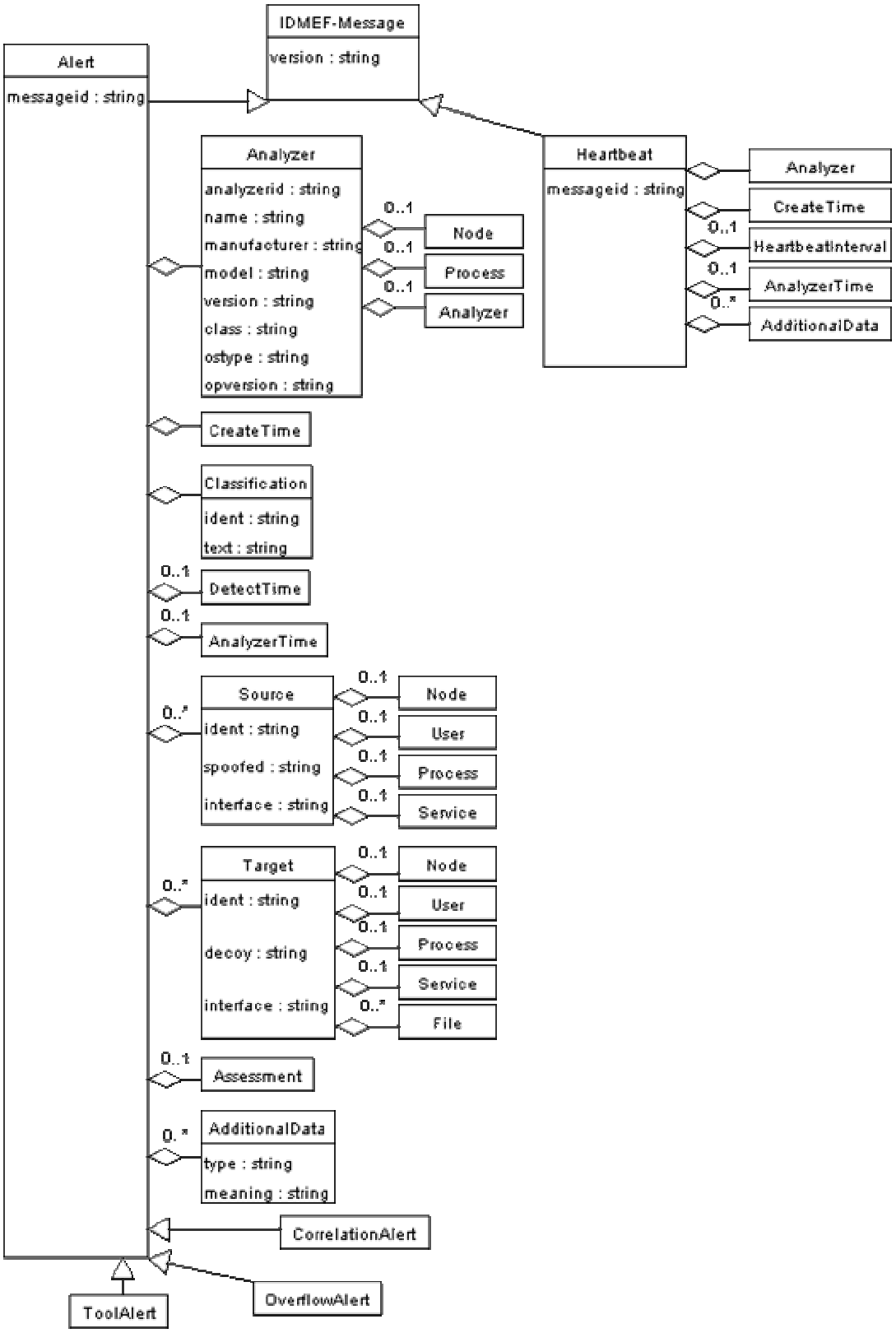

Figura 12. Representação simplificada em UML do modelo de dados do IDMEF, adaptado de Curry & Debar (2005)

A classe raiz do modelo é *IDMEF-Message*, que representa uma mensagem IDMEF e é especializada, na especificação atual do modelo, em duas sub-classes: *Alert* e *Heartbeat*. Mensagens Heartbeat são usadas por analisadores para avisarem frequentemente ao gerenciador que estão vivos. *Alert* é a principal classe do modelo, detalhando um alerta enviado por um analisador. Um alerta deve conter obrigatoriamente a descrição do analisador que o gerou (*Analyzer*), uma classificação indicando do que trata a mensagem (*Classification*) e o momento de sua criação pelo analisador (*CreateTime*). Além do momento no qual a mensagem foi criada, também podem ser descritos os momentos nos quais o evento foi detectado (*DetectTime*) e o alerta foi enviado pelo analisador (*AnalyzerTime*).

As possíveis fontes e possíveis alvos de uma intrusão alertada são opcionalmente descritas pelas classes agregadas *Source* e *Target*. O alerta pode ainda conter informações adicionais que não podem ser representadas no modelo (*AdditionalData*) e também o nível de confiança que o analisador tem sobre os dados avaliados (*Assessment*).

A classe *Alert* pode ser estendida, caso necessário, para incluir novas informações. Atualmente, a classe *Alert* é estendida pelas classes *ToolAlert*, *OverflowAlert* e *CorrelationAlert* para incluir informações adicionais de, respectivamente, ferramentas usadas no ataque, alertas que deveriam ser correlacionados e ataques do tipo *buffer overflow*.

A especificação do IDMEF prevê a sua extensão através de mecanismos de herança e agregação de novas classes. À medida que as extensões amadurecem, elas podem ser incorporadas às versões futuras da especificação. Todavia, para incluir as informações dos registros de auditoria de uso de recursos não é necessária a extensão do modelo. Portanto, a solução apresentada aqui é compatível com a especificação original do IDMEF.

A Tabela 3 demonstra como uma mensagem IDMEF pode incluir informações de um RUR. Cada campo do RUR é mapeado à alguma classe do modelo do IDMEF. Por exemplo, a identificação do usuário pode ser descrita no IDMEF pelos atributos da classe de usuário (*User*) e, por isso, esse campo é mapeado à classe *User* agregada de *Source*, que por sua vez é agregada de *Alert*.

Registro de Uso de Recursos (RUR)		IDMEF	
Agrupamento	Campo	Classe agregada de *Alert*	Classe agregada de *Source/Target*
Detalhes do usuário	Identificação	Source	User
	Nome do host	Source	Node
	Endereço de IP	Source	Node
Detalhes da tarefa	Identificação	Source	Process
	Nome da aplicação	Source	Process
	Data/hora de início	AdditionalData	-
	Data/hora de término	AdditionalData	-
	Status	AdditionalData	-
Detalhes do recurso	Identificação	Target	Node
	Nome do host	Target	Node
	Endereço de IP	Target	Node
	Tipo de host	AdditionalData	-
	Tempo de execução	AdditionalData	-
	Tempo de CPU	AdditionalData	-
	Memória principal	AdditionalData	-
	Armazenamento secundário	AdditionalData	-
	Atividade de rede	AdditionalData	-

Tabela 3. Mapeamento das informações de um RUR em uma mensagem IDMEF

Como visto anteriormente, um alerta IDMEF pode incluir informações sobre as possíveis fontes e alvos de uma intrusão ou ataque. Quando um usuário está fazendo mau uso de uma grade, a fonte da intrusão é ele, sua aplicação maliciosa e as tarefas da aplicação que estão executando na grade. Os nodos onde as tarefas executam são os alvos da intrusão (Figura 13). Desta forma, como descrito na Tabela 3, os detalhes do usuário e da tarefa são incluídos na classe que descreve a fonte (*Source*) e os detalhes do recurso são incluídos na classe do alvo (*Target*).

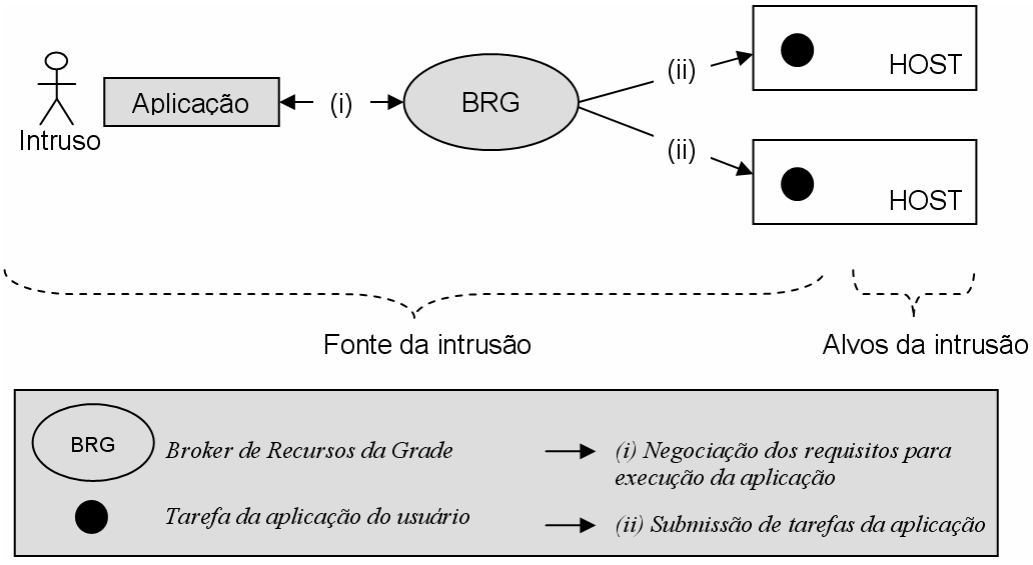

Figura 13. Fonte e alvo de um mau uso

As classes agregadas de *Source* e *Target* (*User*, *Node* e *Process*) têm atributos e classes agregadas que podem ser usadas para descrever os campos do RUR detalhadamente. Maiores informações sobre esses atributos e classes podem ser encontradas na especificação do IDMEF (Curry & Debar, 2005). Todos os campos que não podem ser descritos por *Source* ou *Target*, devido a falta de classes agregadas e atributos apropriados, são incluídos como dados adicionais (*AdditionalData*).

Como podemos ver na Figura 12, a classe *AdditionalData* tem dois atributos: *type* e *meaning*, que descrevem o tipo e o significado do dado adicional. Esses atributos permitem descrever desde dados atômicos, como números inteiros e literais, a estruturas complexas, como dados codificados em XML. Todos os campos do RUR que são mapeados em classes *AdditionalData* são dados atômicos.

Para que o GIDS saiba que uma mensagem IDMEF enviada por um HIDS é um registro de uso de recursos, a mensagem deve ser marcada como tal através da classe *Classification*. E, como o GIDS faz análise dos dados cronologicamente, o HIDS deve fornecer no mínimo a data e hora no qual a mensagem foi criada, através da classe *CreateTime*.

Um exemplo de uma mensagem IDMEF enviada por um HIDS a um GIDS contendo informações mapeadas de um RUR de acordo com a Tabela 3 pode ser visto no Anexo A.

Finalmente, cabe ressaltar que o IDMEF não possui qualquer especificação sobre a segurança das mensagens, deixando essa tarefa para o protocolo de comunicação (Brandão, 2004). Como a segurança de um IDS é crítica e as mensagens podem carregar informações sensíveis, é obrigatório o uso de protocolos que garantam a autenticação mútua entre os envolvidos na comunicação, além da integridade e confidencialidade das mensagens. Portanto, para a comunicação das mensagens IDMEF é sugerido o uso do IDXP, um protocolo de nível de aplicação que usa o *Transport Layer Security* – TLS (Dierks & Allen, 1999) como o protocolo de nível de transporte e infra-estrutura de chave pública e certificados digitais (Housley et al., 1999) para garantir a segurança nas transferências de mensagens.

4.4 Resumo

Neste capítulo foi apresentada uma solução para a viabilização da proposta de método de detecção de intrusão em grades de computadores. Como a proposta prevê a integração de vários IDSs de baixo nível com o GIDS, inicialmente foram descritos os requisitos para que esta integração seja alcançada. São requisitados mecanismos para o envio de alertas e registros de auditoria dos IDSs de baixo nível ao GIDS e o uso de comunicação padronizada, já que a interoperabilidade entre IDSs exige que estes "falem a mesma língua" e o uso de protocolos padronizados é uma característica das grades computacionais.

Em seguida, foi detalhado o uso dos padrões do IDWG na integração dos IDSs, de forma a cumprir os requisitos estabelecidos. O formato de dados IDMEF e o protocolo IDXP, ambos padrões em desenvolvimento pelo IDWG, foram adotados na solução, ao invés do CIDF, que está aparentemente abandonado pelos seus criadores.

Para que o uso dos padrões do IDWG seja justificável, foi mostrado como que o GIDS se encaixa no modelo de detecção de intrusão do IDWG, trabalhando de forma integrada com os outros IDSs. Nesse modelo, os IDSs de baixo nível funcionam tanto como sensores do GIDS, enviando eventos de segurança formatados como alertas IDMEF, quanto como detectores, examinando eventos de segurança em seus escopos locais.

Foi mostrado como acontecem as interações entre o GIDS, os IDSs de baixo nível, usuários e recursos e como essas interações cumprem os requisitos determinados inicialmente. Na comunicação inter-IDS são usados o IDMEF e o IDXP, e na coleta de dados feita pelos IDSs de baixo nível, são usados o formato de dados RUR e o protocolo de comunicação de registros de auditoria syslog.

Os mecanismos de interação apresentados servem para a comunicação ao GIDS de alertas sobre ataques típicos de *hosts* e redes, registros de auditoria sobre a utilização de recursos para verificação do possível mau uso da grade e alertas sobre vestígios de ataques específicos de grade. Como ainda não foram divulgados trabalhos aprofundados sobre ataques específicos de grades, a correlação de dados que possivelmente seria necessária para detectar esses tipos de ataques não foi explorada aqui. A detecção de

ataques típicos de *hosts* e redes também não foi explorada porque isso fica a cargo dos IDSs de baixo nível e o GIDS apenas recebe seus alertas.

A detecção do mau uso é responsabilidade do GIDS. Para verificação do mau uso, diversos parâmetros gerais e específicos podem ser analisados. Esses parâmetros estão todos incluídos nos registros de uso de recursos (RURs) obtidos pelos HIDS através de medidores de recursos da grade (MRG). Foi apresentada uma solução para a inclusão dessas informações em mensagens IDMEF sem a necessidade da extensão do modelo de dados do mesmo, mantendo assim a compatibilidade com sua especificação original. A solução apresentada permite que o analisador do GIDS seja flexível, podendo aplicar diversos tipos de técnicas de análise sobre os dados recebidos.

No próximo capítulo, de forma a validar os mecanismos apresentados aqui, será descrito um estudo de caso de um GIDS que recebe e analisa dados de auditoria enviados como mensagens IDMEF para verificação de mau uso.

5 Estudo de Caso

No capítulo anterior foram descritas soluções para o envio de alertas e registros de auditoria de IDSs de baixo nível a um GIDS usando o formato de dados IDMEF. Um estudo de caso envolvendo o envio e o correlacionamento de alertas sobre vestígios de ataques de grade não pode ser realizado, já que não foram encontrados bancos de dados que compilam assinaturas de tais ataques. A detecção dos ataques típicos de *hosts* e redes não é responsabilidade do GIDS, e um estudo de caso sobre isto também não foi realizado.

Um estudo de caso sobre o envio de registros de auditoria para detecção de mau uso pode ser feito para validar os mecanismos descritos no capítulo anterior. Para isto foi implementado um ambiente de simulação de uma grade computacional com ajuda da ferramenta GridSim (Buyya & Murshed, 2002). Este ambiente é similar ao criado por Tolba et al. (2005a, 2005b).

Simulação foi a única maneira praticável encontrada para analisar o funcionamento de um detector de anomalias em um ambiente de grade com diversos recursos espalhados em diferentes domínio administrativos. Muitos pesquisadores não têm acesso a grades computacionais ou plataformas de testes. Isto se deve ao altos custos e os desafios técnicos e organizacionais necessários para se construir uma grade computacional. Simulação tem ainda vantagem de permitir que experimentos sejam realizados de forma repetitiva usando diferentes combinações e arranjos em um ambiente controlado. De outra forma isso seria custoso e dispendioso em questão de tempo (Tolba, 2005b).

5.1 Ambiente de Simulação

Como podemos ver na Figura 14, a grade simulada no GridSim consiste de vários *hosts* distribuídos em três domínios administrativos. Dez usuários utilizam os recursos da grade submetendo aplicações que são escalonadas para processamento em

seus *hosts*. Essas aplicações são chamadas de *gridlets* no GridSim e os *hosts* tem capacidades computacionais homogêneas para facilitar a realização dos experimentos.

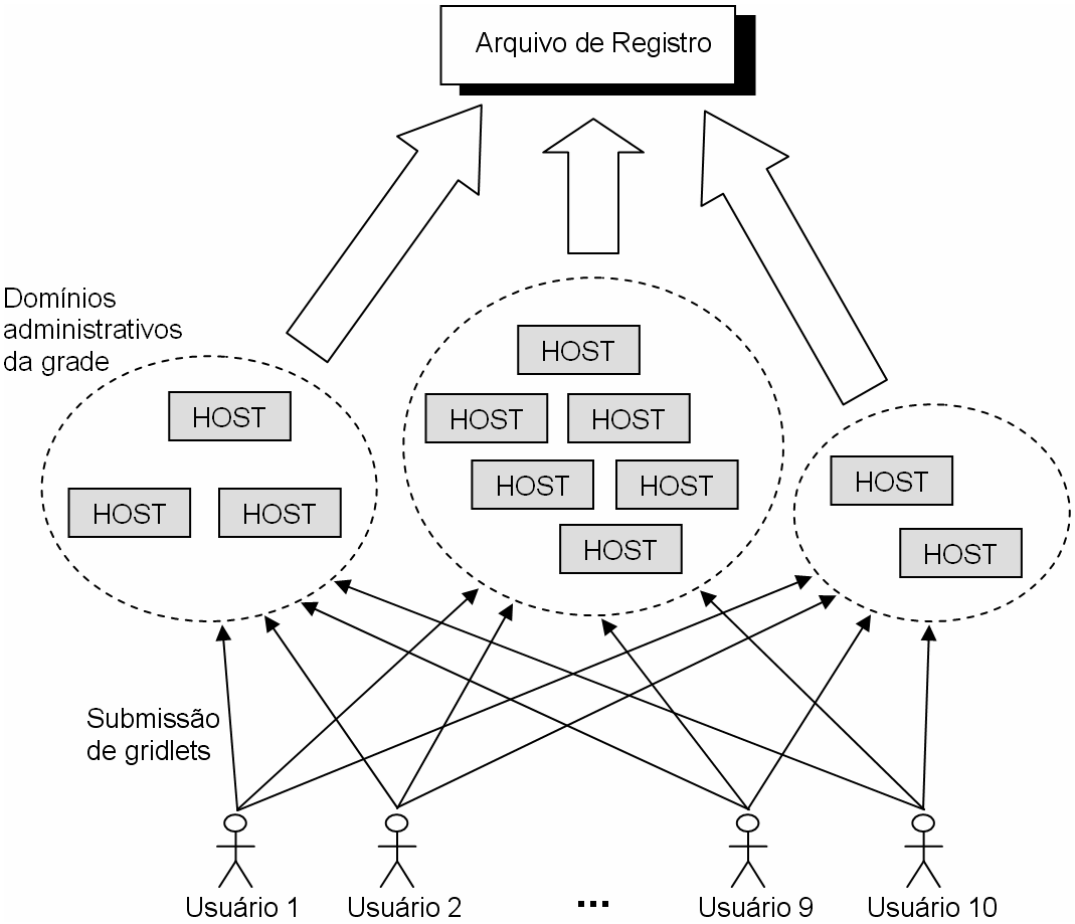

Figura 14. Grade simulada

A simulação foi programada para escrever registros de uso de recursos de cada *gridlet* em um Arquivo de Registro, imitando o mecanismo do MRG (Sub-Seção 4.2.2). As informações desse arquivo são convertidas em mensagens IDMEF – como descrito na Seção 4.3 – usando uma ferramenta implementada com a ajuda de uma biblioteca de manipulação de mensagens IDMEF (McCubbin & Luu, 2002). Essa ferramenta envia as mensagens ao GIDS, simulando o comportamento de um HIDS.

O GIDS é um *script* escrito com a ajuda do Matlab (Matlab, 2006). Esse *script* simula a funcionalidade de uma Tarefa Analisadora (Sub-Seção 3.2.2). A técnica implementada usa uma rede neural artificial com uma regra de aprendizado de

retropropagação (Barreto, 2001) para detectar anomalias no comportamento dos usuários.

O funcionamento da Tarefa Analisadora e da rede neural é descrito na Figura 15. Os dados são pré-processados antes de serem encaminhados à rede neural, já que as mensagens enviadas pelos HIDS precisam ser convertidas em um formato de dados adequado ao Matlab.

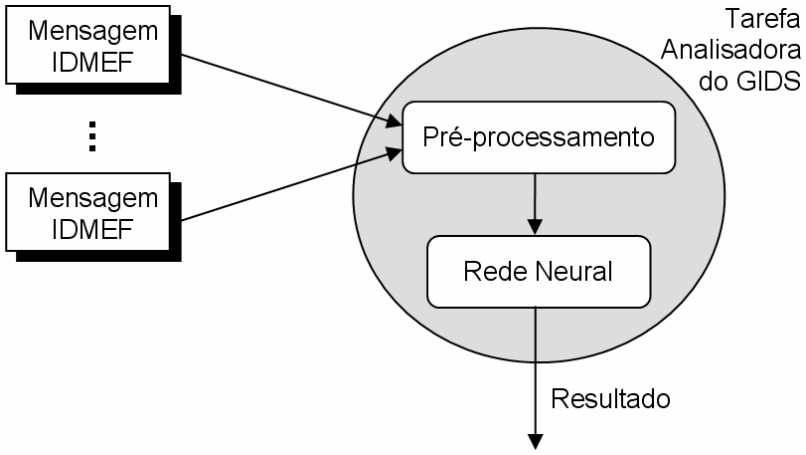

Figura 15. Tarefa Analisadora do GIDS e sua rede neural

Na seção seguinte é apresentado o estudo de caso realizado com o ambiente de simulação descrito aqui.

5.2 Descrição do Estudo de Caso

Primeiramente, a rede neural foi treinada para aprender o que é uma anomalia de comportamento de um usuário. Para isto, foi feita uma simulação na qual cinco do total de dez usuários, a partir de determinado momento, começam a submeter *gridlets* que demandam uma capacidade de processamento bastante diferente do que foi demandado pelos seus últimos *gridlets*. Nesta etapa, a rede neural é informada de quais são esses usuários com comportamento anormal, de modo que ela possa aprender a identificá-los.

Em seguida, foram feitos diversos experimentos para verificar se a rede neural realmente aprendeu a detectar as mudanças bruscas de comportamento dos usuários. Em

cada experimento, dentre os dez usuários havia um número indeterminado deles que apresentavam anomalias de comportamento e foi constatado que a rede neural de fato conseguiu identificar os usuários que são possivelmente intrusos. Entretanto, cabe salientar que um IDS é um sistema no qual a parte humana tem um papel importante no julgamento dos alertas gerados pela parte do software. A ocorrência de um alerta não significa necessariamente que houve uma intrusão.

Tanto na etapa de treino quanto na etapa de experimentação a rede foi alimentada com uma quantidade de parâmetros de entrada. Esses parâmetros foram obtidos dos últimos *gridlets* que o usuário submeteu à grade. Quanto menor o número de *gridlets*, menor a quantidade de informação que a rede tem e maior a dificuldade enfrentada no aprendizado e identificação de anomalias. Na Figura 16 é mostrada a influência do número de *gridlets* na taxa de erros de detecção da rede. Quanto maior o número de *gridlets* analisados, menor o número de falsos alarmes e intrusos não identificados.

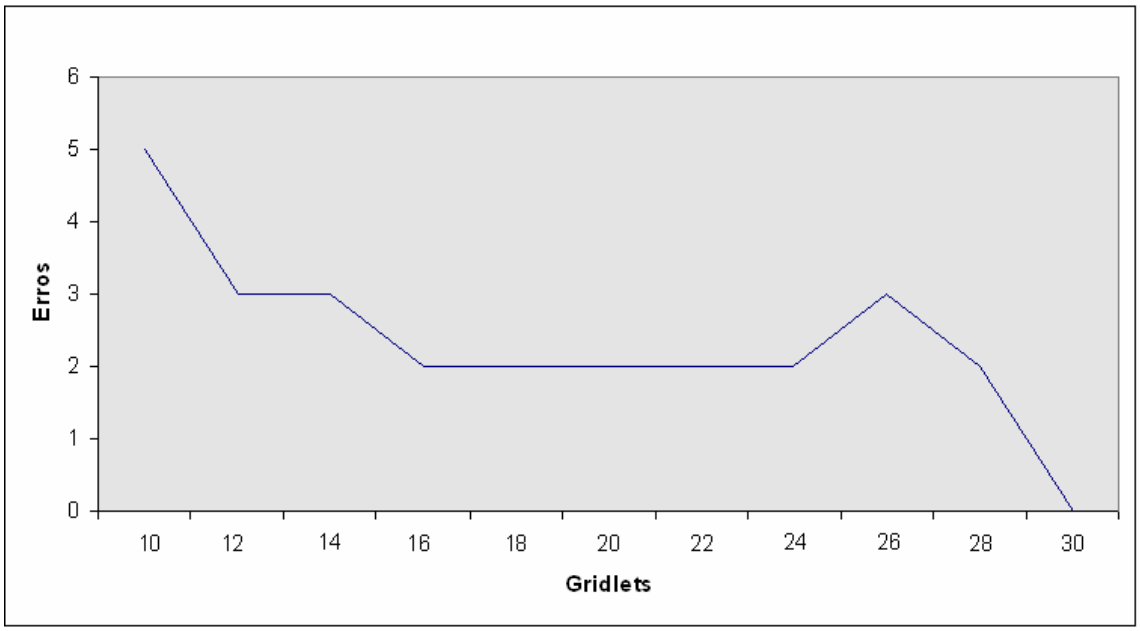

Figura 16. Erros de detecção em relação ao número de *gridlets* analisados

5.3 Considerações

Não é objetivo deste trabalho identificar o melhor tipo de rede neural para resolver o problema. Trabalhos específicos sobre detecção de intrusão usando redes neurais existem na literatura (Allen et al., 2000).

O GIDS implementado e descrito aqui pode ser caracterizado de acordo com a tabela de classificação de IDSs apresentada na Sub-Seção 2.3.1. Essa classificação é demonstrada na Tabela 4.

Técnica de detecção	Anomalia (rede neural)
Comportamento pós-detecção	-
Origem dos dados	*Hosts*
Tempo de detecção	*Off-line*
Freqüência de uso	Análise periódica / *batch*
Arquitetura de análise	Centralizada

Tabela 4. Características do GIDS implementado

6 Conclusões

O propósito geral deste trabalho foi apresentar uma abordagem que supera os pontos fracos das soluções disponíveis ao problema da detecção de intrusão em grades computacionais. Mais especificamente, este trabalho teve como objetivos responder uma série de questões que foram introduzidas no Capítulo 1. As questões e suas respostas estão listadas abaixo, formando um sumário do que foi apresentado neste trabalho:

O que é uma grade de computadores?

Uma grade de computadores é uma ferramenta usada para facilitar o compartilhamento, troca, descoberta e agregação de recursos heterogêneos distribuídos por múltiplos domínios administrativos. Grades podem ser usadas por certas aplicações que requerem grande capacidade computacional, capacidade de armazenamento massivo ou preparativos complexos de recursos. Entre suas características estão a falta de um controle central, o uso de protocolos e interfaces padronizados e o fornecimento de qualidade de serviço não-trivial. A computação em grade foi motivada pela evolução das tecnologias computacionais e de redes e aparece como uma fase evolucionária da computação distribuída, e não como uma tecnologia revolucionária.

Qual a importância da segurança de grades?

Várias questões são tratadas por uma grade, sendo a segurança uma das principais. O compartilhamento seguro de recursos é um importante requisito para a aceitação das grades por usuários e provedores de recursos. Segurança e privacidade são preocupações principais nos desenvolvimentos recentes de grades comerciais, sendo componentes chaves para tornar as grades economicamente praticáveis.

À medida que o uso de grades torna-se mais comum, migrando de sistemas experimentais a sistemas de produção, amplifica-se a importância das questões de segurança (Humphrey & Thompson, 2002), assim como a necessidade pelo entendimento e controle da segurança das aplicações de grade.

Qual a diferença de um ataque e de uma intrusão?

"Ataques" e "intrusões" são comumente considerados sinônimos no contexto de detecção de intrusão.

Usuários legítimos usando uma grade de forma maliciosa são intrusos que devem ser identificados?

Neste trabalho, o comportamento de usuários legítimos da grade que abusam de seus privilégios usando-a para propósitos não intencionados é considerado intrusivo e sua detecção é desejável, já que eles podem representar ameaças intencionais de segurança.

Que tipos de ataques e intrusões podem ocorrer em uma grade?

Sistemas de grade são suscetíveis a todos os ataques típicos de redes e computadores *host*, assim como métodos específicos de ataque, por causa dos seus novos protocolos e serviços. O acesso não-autorizado por invasores que se mascaram como usuários legítimos e o mau uso da grade tanto por usuários legítimos como por intrusos externos também são consideradas intrusões de grade.

Assim, as intrusões que podem ocorrer em uma grade foram classificadas neste trabalho como: (a) acesso não-autorizado, (b) mau uso, (c) ataque de grade e (d) ataques específicos de *host* ou rede.

O que diferencia intrusões em grades de intrusões em outros sistemas distribuídos?

Uma das diferenças entre intrusões em grades e em outros sistemas distribuídos está no fato de suas velocidades, suas conseqüências e seus danos são potencialmente maiores nas grades. Caso um intruso consiga acessar a grade e submeter tarefas, suas tarefas maliciosas terão os mesmos benefícios e facilidades que são fornecidos pela grade a tarefas não-maliciosas.

Segundo Vijayan (2004), apesar de que as grades não criam fundamentalmente novos riscos de segurança, elas servem para amplificar alguns deles. A maioria dos problemas que os usuários têm de lidar em um ambiente de grade são similares aos que eles enfrentam em ambientes não-grade. Mas eles têm uma maior significância no

ambiente de grade por causa da premissa fundamental de grades: acesso, compartilhamento e computação colaborativa.

O que pode ser afetado em uma grade por causa de uma intrusão?

A disponibilidade dos recursos e a confidencialidade e a integridade de dados de usuários e de sistema.

O que gera a necessidade de um IDS em um ambiente de grade?

Middleware de grade fornecem serviços preventivos de segurança, mas experiências reais de computação em grade indicam que serviços complementares, como a detecção de intrusão, podem ser necessários.

As ameaças de segurança a grades estão ganhando mais atenção. Os recursos de grades podem ser bastante atrativos a pessoas mal-intencionadas, especialmente os recursos de uma grade de larga-escala. Mesmo que os serviços de segurança em sistemas distribuídos estejam evoluindo, as suas contra-partes (ataques) também estão.

Não é realista prevenir absolutamente o aparecimento de brechas de segurança em sistemas distribuídos complexos como as grades. Isto ainda é enfatizado quando elas são implementadas em uma rede aberta como a Internet. Mesmo que os serviços de segurança oferecidos por um middleware de grade sejam livres de vulnerabilidades, intrusos podem explorar defeitos em qualquer dos outros componentes envolvidos. Além do mais, uma grade não pode defender-se de senhas roubadas e usuários legítimos que abusam de seus privilégios para executar atividades maliciosas.

Neste contexto, IDSs aparecem como ferramentas que podem melhorar a segurança de uma grade.

Como funcionam os sistemas de detecção de intrusão?

IDSs, essencialmente, são sistemas que coletam e analisam dados de auditoria de uma fonte de eventos de segurança de um sistema de informação em busca de anomalias ou assinaturas de ataques conhecidos.

Por que os IDSs existentes não são adequados para proteger grades?

As tecnologias atuais de detecção de intrusão falham em fornecer proteção contra todas as intrusões que podem violar a segurança de uma grade. Os típicos NIDS e HIDS podem ser implantados em uma grade para melhorar a sua segurança, mas não é uma solução completa porque eles não têm a capacidade de detectar apropriadamente intrusões específicas de grade.

Já foram publicadas arquiteturas de IDSs baseados em grade (GIDS) e foi constatado que esses GIDS são projetados para detectar apropriadamente anomalias de comportamento de usuários de uma grade, mas são deficientes na detecção de ataques a computadores *host* e redes da grade e também são deficientes na detecção de ataques específicos de grade que não causam anomalias.

O que é detecção de intrusão em grades?

Detecção de intrusão em grades é um processo que envolve a coleta de informações de segurança observáveis em suas redes e nodos (computadores *host*) e a identificação, baseada na avaliação e correlação dos dados coletados, de ataques contra todos os alvos possivelmente vulneráveis, assim como de anomalias na interação de usuários de grade com recursos.

O que difere um IDS de grade de um IDS típico de host e rede?

HIDS analisam dados de auditoria obtidos em computadores *host*. Já os NIDS analisam dados de auditoria coletados em uma rede. Um GIDS, por sua vez, analisa ambas as fontes de dados. A diferença está no fato de que para detectar intrusões de grade o GIDS precisa fazer correlação desses dados e o escopo em que ele trabalha é mais abrangente que o escopo de um HIDS ou um NIDS, que geralmente estão limitados a um *host* ou um domínio de rede, respectivamente.

O que um IDS de grade precisa para superar as limitações das tecnologias atuais?

Para superar as limitações das soluções disponíveis, um GIDS precisa abranger as intrusões (a), (b), (c) e (d) e ser escalável com o número de recursos e usuários da grade. Além disso, é desejável que ele seja compatível com a grade, ajustando-se ao

ambiente e beneficiando-se do mesmo, lide com recursos heterogêneos e seja flexível quanto às técnicas de detecção que podem ser aplicadas.

Como detectar intrusões em grades?

Para detectar intrusões em grades foi proposto um método no qual o GIDS é um componente de alto nível que trabalha de forma integrada com HIDS e NIDS analisando e correlacionando dados de auditoria e alertas enviados por eles, chamados aqui de IDSs de baixo nível. Nesse método, o nível de segurança desejado é alcançado pela instalação desses IDSs em nodos e domínios de rede de uma grade. Para alcançar o nível máximo de segurança, cada nodo e cada rede deve ter um desses IDSs instalado.

6.1 Principais Contribuições

Este trabalho tem como principal contribuição o método de detecção de intrusão em grades computacionais apresentado no Capítulo 3. Este método supera as limitações das soluções existentes e tem como vantagem a re-utilização de software de IDS disponível, evitando re-implementação de funcionalidade.

Foi também apresentada a arquitetura de um exemplo de GIDS que pode ser usado nesse método e foi mostrado como ele cumpre os requisitos básicos de um GIDS: abrangência, escalabilidade, compatibilidade com a grade, heterogeneidade e flexibilidade. Mesmo essa arquitetura tendo problemas – como apontado na Sub-Seção 3.2.3 – e não sendo validada neste trabalho, ela é uma contribuição na forma de uma idéia de arquitetura de um GIDS ideal que pode ser melhor desenvolvida em trabalhos futuros.

Outra contribuição do trabalho é a solução apresentada no Capítulo 4 para a integração de IDSs prevista no método. A solução permite que vários IDSs de baixo nível distintos enviem a um GIDS alertas sobre intrusões detectadas localmente, alertas sobre vestígios de ataques de grade e registros de auditoria sobre a utilização de recursos por usuários da grade. Uma vantagem dessa solução é o uso de protocolos e formatos em processo de padronização.

A introdução ao problema da detecção de intrusão em grades feita no Capítulo 2 também pode ser considerada uma contribuição ao conhecimento, já que os poucos trabalhos relacionados ao tema que foram citados na Seção 2.5 não descrevem o problema de forma alguma ou fazem introduções ao mesmo de uma maneira relativamente curta e com pouco embasamento.

Este trabalho gerou duas publicações (Schulter et al., 2006a, Schulter et al., 2006b).

6.2 Trabalhos Futuros

A método proposto neste trabalho consiste basicamente da integração de vários IDSs de baixo nível e um GIDS. Depois da solução de integração de IDSs descrita no Capítulo 4, o problema da detecção de intrusão em grades torna-se agora um problema de adequação de IDSs já existentes para que trabalhem de forma integrada com um GIDS usando os protocolos e formatos sugeridos na solução para detectar as intrusões de grade classificadas na Seção 2.4, sendo isto sugerido para trabalhos futuros.

A arquitetura do exemplo que foi descrito na Sub-Seção 3.2.2 também pode ser melhor desenvolvida e validada. As principais questões que o exemplo desperta e que precisariam ser respondidas são:

- *Como o Escalonador traduz o perfil do usuário em Tarefas Analisadoras e as distribui em nodos da grade para processamento de dados de auditoria?*
- *Como saber qual vai ser a computação demandada a partir dos perfis?*
- *Para onde submeter uma Tarefa Analisadora de forma a evitar sobrecargas na rede devido a transferências volumosas de dados entre componentes do GIDS?*
- *Como Tarefas Analisadoras correlacionam entre si os dados de auditoria que analisam?*
- *Os bancos de dados de auditoria são sincronizados ou independentes?*

A arquitetura tem como objetivo satisfazer os requisitos de abrangência de intrusões, escalabilidade, compatibilidade com a grade, flexibilidade e heterogeneidade.

Além desses, novos requisitos podem ser considerados. Alguns exemplos são: exatidão de detecção, tolerância a falhas, tempo de detecção, desempenho e segurança do próprio GIDS.

Para simplificar o problema, foi abstraída a existência de organizações virtuais – OVs (Foster et al., 2001) na grade onde o GIDS atua. Caso existam OVs na grade, outras questões surgem e podem ser respondidas por trabalhos futuros. Uma delas diz respeito à integração de diferentes GIDS, cada um atuando em uma OV distinta, e as relações de confiança que deveriam existir entre tais sistemas.

Uma série de outras atividades podem ser destacadas como bons caminhos a serem seguidos por trabalhos futuros neste tema. Algumas delas estão delineadas abaixo:

- Um estudo sobre quais são as respostas ou contra-medidas mais adequadas para cada tipo de intrusão.
- Um estudo sobre as deficiências inerentes de qualquer IDS, mas no contexto de grades computacionais. Uma relação dessas deficiências foi compilada por Jansen (2002).
- Um estudo sobre a viabilidade da detecção de intrusão em grades "da vida real". Em quais casos compensa haver um IDS, em quais casos não compensa e o porquê disso. Entre os fatores envolvidos estão o custo de recursos, já que IDSs são sabidos consumirem bastante poder de processamento e espaço de armazenamento (Allen et al., 2000), e as conseqüências dos alarmes falsos.
- A proposta de um GIDS que forneça um serviço gerenciado de detecção de intrusão em grades, já que grades computacionais têm como característica o fornecimento de qualidade de serviço (Foster, 2002).

Referências

ALLEN, Julia et al. State of the Practice of Intrusion Detection Technologies. Technical Report CMU/SEI-99-TR-028, Software Engineering Institute, Carnegie Mellon University, jan. 2000.

ASADZADEH, Parvin et al. Global Grids and Software Toolkits: A study of four middleware technologies. **High Performance Computing: Paradigm and Infrastructure**, New Jersey: Wiley, june 2005.

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

AXELSSON, Stefan. Intrusion Detection Systems: A Survey and Taxonomy. Technical Report, Department of Computer Engineering, Chalmers University of Technology, Göteborg, Sweden, mar. 2000.

BALATON, Zoltan; GOMBAS, Gabor. Resource and Job Monitoring in the Grid. In: EUROPAR 2003 INTERNATIONAL CONFERENCE ON PARALLEL AND DISTRIBUTED COMPUTING (Euro-Par), 9., 2003, Klagenfurt, Austria. **Anais…** [S.l.]: Springer, 2003. p. 404-411.

BARMOUTA, Alexander; BUYYA, Rajkumar. GridBank: A Grid Accounting Services Architecture (GASA) for Distributed Systems Sharing and Integration. In: INTERNATIONAL PARALLEL AND DISTRIBUTED PROCESSING SYMPOSIUM (IPDPS), 17., 2003. **Anais…** Washington: IEEE Computer Society, 2003. p. 254a.

BARRETO, Jorge M. **Inteligência Artificial No Limiar do Século XXI.** 3. ed, Florianópolis: O Autor, 2001. 379 p.

BERMAN, Fran; FOX, Geoffrey; HEY, Tony. The Grid: Past, Present, Future. In: BERMAN, Fran; FOX, Geoffrey; HEY, Tony (Editores). **Grid Computing: Making the Global Infrastructure a Reality**. UK: Wiley and Sons, mar. 2003. cap. 1. p. 9-50. A Special Issue of Concurrency and Computation: Practice and Experience.

BERNERS-LEE, Tim. **Weaving the Web: The Past, Present, and Future of the World Wide Web by its Inventor**. UK: Orion, 1999. 283 p.

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.

BRANDÃO, José Eduardo; FRAGA, Joni da Silva; MAFRA, Paulo Manoel. Composição de IDSs Usando *Web Services*. In: SIMPÓSIO BRASILEIRO EM SEGURANÇA DA INFORMAÇÃO E DE SISTEMAS COMPUTACIONAIS (SBSeg), 5., 2005, Florianópolis, SC, Brasil. **Anais...** [S.l.]: SBC, 2005. p. 1-18.

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, Ong Tian; SAMSUDIN, Azman. Grid-based Intrusion Detection System. In: ASIA-PACIFIC CONFERENCE ON COMMUNICATIONS, 9., 2003, Penang, Malaysia. **Anais…** [S.l.: s.n.], 2003. v. 3, p. 1028-1032.

CURRY, David; DEBAR, Hervé. Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition. Intrusion Detection Workgroup (IDWG). **IETF Internet-Draft**, 2005. Disponível em: <http://www.ietf.org/internet-drafts/draft-ietf-idwg-idmef-xml-14.txt>. Acesso em: 10 dez. 2005.

DEBAR, Hervé; DACIER, Marc; WESPI, Andréas. Towards a Taxonomy of Intrusion-Detection Systems. **International Journal of Computer and**

Telecommunications Networking, v. 31, n. 9, p. 805-822, apr. 1999, Special Issue on Computer Network Security

DENNING, Dorothy E.. An Intrusion-Detection Model. **IEEE Transactions on Software Engineering**, USA, v. 13, n. 2, p. 222-232, feb. 1987, Special Issue on Computer Security and Privacy.

DIERKS, Tim; ALLEN, Christopher. The TLS Protocol, Version 1.0, Network Working Group, **IETF RFC-2246**, 1999. Disponível em: <http://www.ietf.org/rfc/rfc2246.txt>. Acesso em: 10 dez. 2005.

FANG-YIE, Leu et al. A Performance-based Grid Intrusion Detection System. In: IEEE INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE (COMPSAC), 29., 2005, Edinburgh, Scotland, UK. **Anais…** [S.l.]: IEEE Computer Society, 2005. p. 525-530.

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, 2005. v. 1, p. 304-309.

FEINSTEIN, Benjamin; MATTHEWS, Gregory; WHITE, John. The Intrusion Detection Exchange Protocol. Intrusion Detection Working Group (IDWG). **IETF Internet-Draft**, 2002. Disponível em: <http://www.ietf.org/internet-drafts/draft-ietf-idwg-beep-idxp-07.txt>. Acesso em: 10 dez. 2005.

FOCKE, Nils. **Introduction to Grid Computing with Globus**. 2004. 106 f. Dissertação (Mestrado em Ciência da Computação)– Programa de Pós-Graduação em Ciência da Computação, Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis, SC, 2004.

FOSTER, Ian; KESSELMAN, Carl. Globus: A Metacomputing Infrastructure Toolkit. **International Journal of Supercomputer Applications and High Performance Computing**, v. 11, n. 2, p. 115-128, summer 1997.

FOSTER, Ian et al. A Security Architecture for Computational Grids. In: ACM CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, 5., 1998, San Francisco, CA, USA. **Anais…** New York: ACM, 1998. p. 83-91.

FOSTER, Ian; KESSELMAN, Carl, TUECKE, Steven. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. **International Journal of Supercomputer Applications,** v. 15, n. 3, p. 200-222, fall 2001.

FOSTER, Ian. What Is The Grid? A Three Point Checklist. **GRIDtoday**, v. 1, n. 6, July 22, 2002.

FOSTER, Ian; KESSELMAN, Carl. **The Grid 2**: Blueprint for a New Computing Infrastructure. 2. ed., San Francisco, CA, USA: Morgan Kaufmann, 2003. 748 p.

GERHARDS, Rainer. The syslog Protocol. syslog Working Group. **IETF Internet-Draft**, 2005. Disponível em: <http://www.ietf.org/internet-drafts/draft-ietf-syslog-protocol-15.txt>. Acesso em: 10 dez. 2005.

GGF. Global Grid Fórum. Sítio da Comunidade. Disponível em: <http://www.ggf.org/>. Acesso em: 10 dez. 2005.

HALME, Lawrence R.; BAUER, Kenneth R. AINT Misbehaving – A Taxonomy of Anti-Intrusion Techniques. In: NATIONAL INFORMATION SYSTEMS SECURITY CONFERENCE, 18., 1995, Baltimore, MD, USA. **Anais…** [S.l.: s.n.], 1995.

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, 1990. p. 296-304.

HOUSLEY, Russel et al. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. **IETF RFC-2459**, 1999. Disponível em: <http://www.ietf.org/rfc/rfc2459.txt>. Acesso em: 10 dez. 2005.

HUMPHREY, Marty; THOMPSON, Mary R.. Security Implications of Typical Grid Computing Usage Scenarios. **Cluster Computing,** USA, v. 5, n. 3, p. 257-264, july 2002.

HUMPHREY, Marty; THOMPSON, Mary; JACKSON, Keith R. Security for Grids. **Proceedings of the IEEE,** v. 93, n. 3, p. 644-652, mar. 2005, Special Issue on Grid Computing.

IDWG. Intrusion Detection Working Group official charter. Disponível em: <http://www.ietf.org/html.charters/idwg-charter.html>. Acesso em: 10 dez. 2005.

IETF. The Internet Engineering Task Force. Sítio da Força Tarefa. Disponível em: <http://www.ietf.org/>. Acesso em: 10 dez. 2005.

ISC. Internet Software Consortium. Internet Domain Survey. Disponível em: <http://www.isc.org/ds>. Acesso em: 9 jan. 2006.

JANSEN, Wayne. Intrusion Detection with Mobile Agents. **Computer Communications**, v. 25, n. 15, p. 1392-1401, oct. 2002. Special Issue on Intrusion Detection.

KAHN, Cliff et al. A Common Intrusion Detection Framework. **Journal of Computer Security**, july 1998.

KANDULA, Srikanth; SINGH, Sankalp; SANGHI, Dheeraj. Argus: A Distributed Network Intrusion Detection System. In: INTERNATIONAL SYSTEM ADMINISTRATION AND NETWORK ENGINEERING CONFERENCE, 3., 2002, Maastricht, The Netherlands. **Anais…** Netherlands: NLUUG, 2002.

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, 2005. p. 238-245.

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, Stuart; COGHLAN, Brian. Towards a Grid-wide Intrusion Detection System. In: EUROPEAN GRID CONFERENCE (EGC), 2005, Amsterdam, The Netherlands. **Anais…** [S.l.]: Springer, 2005. p. 275-284.

LEITE, Ricardo et al. Functionalities in Grid Computing with Active Services. In: INTERNATIONAL WORKSHOP ON MIDDLEWARE FOR GRID COMPUTING (Associated with Middleware 2003), 1., 2003, Rio de Janeiro, RJ, Brazil. **Anais…**Rio de Janeiro: PUC-Rio, 2003. p. 194-199.

LIABOTIS, Ioannis et al. Self-Organising Management of Grid Environments. In: INTERNATIONAL SYMPOSIUM ON TELECOMMUNICATIONS, 2003, Isfahan, Iran. **Anais…** [S.l.: s.n.], 2003.

LIM, Dudy et al. MOGAS: A Multi-Organizational Grid Accounting System. **International Journal on Information Technology**, Singapore, v. 11, n. 4, 2005, Special Issue on Grid Computing 2.

LINDQVIST, Ulf; JONSSON, Erland. How to Systematically Classify Computer Security Intrusions. In: IEEE SYMPOSIUM ON SECURITY AND PRIVACY, 1997, Oakland, CA, USA. **Anais…**Los Alamitos: IEEE Computer Society, 1997. p. 154-163.

MATLAB – The Language of Technical Computing. Disponível em: < http://www.mathworks.com/products/matlab/>. Acesso em: 10 jan. 2006.

MCCUBBIN, Chris; LUU, Michael. JavaIDMEF Message Implementation v.941beta. 2002. Disponível em: <http://sourceforge.net/projects/javaidmef>. Acesso em: 10 jan. 2006.

MCHUGH, John; CHRISTIE, Alan; ALLEN, Julia. Defending Yourself: The Role of Intrusion Detection Systems. **IEEE Software**. Los Alamitos, CA, USA, v. 17, n. 5, p. 42-51, sep./oct. 2000.

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

METCALFE, Robert M.; BOGGS, David R. Ethernet: Distributed Packet Switching for Local Computer Networks. **Communications of the ACM**, v. 19, n. 7, p. 395-404, july 1976.

MI-YOUNG, Koo. Usage Record Fields Survey Results and Proposed Minimum Set. Usage Record (UR) Working Group. **GGF Draft Documents**. Disponível em: <http://www.ggf.org/ggf6/ggf6_wg_papers/res_definition.pdf>. Acesso em: 10 dez. 2005.

MOORE, Gordon. Cramming More Components Onto Integrated Circuits. **Electronics Magazine.** USA, v. 38, n. 8, p. 114-117, apr. 1965.

MOORE, Gordon. No Exponential is Forever… 2003. Disponível em: <ftp://download.intel.com/research/silicon/Gordon_Moore_ISSCC_021003.pdf>. Acesso em: 10 dez. 2005.

MUKHERJEE, Biswanath; HEBERLEIN, L. Todd; LEVITT, Karl N. Network Intrusion Detection. **IEEE Network**, v. 8, n. 3, p. 26-41, may/june 1994.

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.

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.], 1997. p. 353-365.

RAID – INTERNATIONAL SYMPOSIUM ON RECENT ADVANCES IN INTRUSION DETECTION, 7., 2004, Sophia Antipolis, France, USA. **Anais…** [S.l.]: Springer, 2004.

RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady. **The Unified Modeling Language Reference Manual**. 1. ed., USA: Addison-Wesley, 1998. 568 p.

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.], 2003. v. 2, p. 844-848.

SARDINHA, Luis; NEVES, Nuno F.; VERÍSSIMO, Paulo. Tolerating Intrusions in Grid Systems. In: INTERNATIONAL CONFERENCE ON SECURITY AND MANAGEMENT, 2004, Las Vegas, USA. **Anais…** [S.l.]: CSREA, 2004. p. 207-214.

SCHULTER, Alexandre et al. Towards Grid-based Intrusion Detection. In: IEEE/IFIP NETWORK OPERATIONS & MANAGEMENT SYMPOSIUM, 10., 2006, Vancouver, Canada. **Anais…** [S.l.: s.n.], 2006.

SCHULTER, Alexandre et al. A Grid-based Intrusion Detection System. In: INTERNATIONAL CONFERENCE ON SYSTEMS, 1., 2006, Mauritius. **Anais…** Los Alamitos: IEEE Computer Society, 2006. p. 187.

SHIELDS, Clay. What do we mean by Network Denial of Service? In: IEEE WORKSHOP ON INFORMATION ASSURANCE AND SECURITY, 3., 2002, New York, USA. **Anais…** [S.l.: s.n.], 2002.

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.

SMAHA, Stephen. Haystack: An intrusion detection system. In: AEROSPACE COMPUTER SECURITY APPLICATIONS CONFERENCE, 4., 1988, Orlando, FL, USA. **Anais…** Los Alamitos: IEEE Computer Society, 1988. p. 37-44.

SNAPP, Steven R. et al. DIDS (Distributed Intrusion Detection System) – Motivation, Architecture, and an Early Prototype. In: NATIONAL COMPUTER SECURITY CONFERENCE, 14., 1991, Washington, DC, USA. **Anais…** [S.l.: s.n.], 1991. p. 167-176.

SNORT. Sítio do Projeto. Disponível em: <http://www.snort.org/>. Acesso em: 10 dez. 2005.

SOMMER, Peter. Intrusion Detection Systems as Evidence. **International Journal of Computer and Telecommunications Networking,** v. 31, n. 23-24, p. 2477-2487, dec. 1999.

SUNDARAM, Aurobindo. An Introduction to Intrusion Detection. **ACM Crossroads**. New York, NY, USA, v. 2, n. 4, p. 3-7, apr. 1996. Special Issue on Computer Security.

TALNAR, Vanish; BASU, Sujoy; KUMAR, Raj. An Environment for Enabling Interactive Grids. In: IEEE INTERNATIONAL SYMPOSIUM ON HIGH PERFORMANCE DISTRIBUTED COMPUTING, 12., 2003, Seattle, Washington, USA. **Anais…** Washington: IEEE Computer Society, 2003. p. 184-195.

TERAGRID. Sítio do Projeto. Disponível em: <http://www.teragrid.org/>. Acesso em 10 dez. 2005.

TOLBA, M. et al. GIDA: Toward Enabling Grid Intrusion Detection Systems. In: 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.

UNDERCOFFER, Jeffrey et al. A Target-Centric Ontology for Intrusion Detection. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 18., 2004, Acapulco, México. **Anais…** [S.l.: s.n.], 2004.

VIJAYAN, Jaikumar. Guarding the Grid. **Computerworld**, nov. 2004. Disponível em: <http://www.computerworld.com/securitytopics/security/story/0,10801,97815,00.html>. Acesso em: 9 jan. 2006.

WEAVER, Nicholas et al. A Taxonomy of Computer Worms. In: ACM WORKSHOP ON RAPID MALCODE, 1., 2003, Washington, DC, USA. **Anais…** [S.l.]: ACM, 2003. p. 11-18. Associated with the 10th ACM Conference on Computer and Communications Security.

WOOD, Mark; ERLINGER, Michael. Intrusion Detection Message Exchange Requirements. Intrusion Detection Working Group (IDWG). **IETF Internet-Draft**, 2002. Disponível em: <http://www.ietf.org/internet-drafts/draft-ietf-idwg-requirements-10.txt>. Acesso em: 10 dez. 2005.

YUANBO Guo, JIANFENG Ma, YADI Wang. An Intrusion-Resilient Authorization and Authentication Framework for Grid Computing Infrastructure. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE, 5., 2005, Atlanta, GA, USA. **Anais…** [S.l.]: Springer-Verlag, 2005. v. 3516, n. 3, p. 229-236.

Anexo A – Uma mensagem IDMEF contendo informações de um RUR

A mensagem XML abaixo representa um alerta enviado a um GIDS por um HIDS (`grid-domain_a-hids01`) que está localizado em um dos domínios administrativos da grade (`domain_a.grid.com`). Na mensagem está indicado o momento no qual ela foi criada (`idmef:CreateTime`) e sua classificação (`idmef:Classification`) descrevendo que ela corresponde a um Registro de Uso de Recursos – RUR (`Grid Resource Usage Record`). Na classificação também é descrito um endereço de uma página na Web onde mais informações sobre esse tipo de alerta podem ser encontradas (`http://www.gridsecuritysite.com/database/99`).

Como visto na Seção 4.3, os detalhes do usuário e da tarefa são mapeados em Source (`idmef:Source`) e AdditionalData (`idmef:AdditionalData`) e os detalhes do recurso são mapeados em Target (`idmef:Target`) e também AdditionalData.

```
<?xml version="1.0" encoding="UTF-8"?>
<idmef:IDMEF-Message xmlns:idmef="http://iana.org/idmef"
                     version="1.0">
  <idmef:Alert messageid="abc123456789">
    <idmef:Analyzer analyzerid="grid-domain_a-hids01">
      <idmef:Node category="dns">
        <idmef:location>Grid Resource Domain A</idmef:location>
        <idmef:name>domain_a.grid.com</idmef:name>
      </idmef:Node>
    </idmef:Analyzer>
    <idmef:CreateTime ntpstamp="0xbc723b45.0xef449129">
      2006-01-01T11:05:00.0000-03:00
    </idmef:CreateTime>
    <idmef:Classification text="Grid Resource Usage Record">
      <idmef:Reference origin="vendor-specific">
        <idmef:name>99</idmef:name>
        <idmef:url>http://www.gridsecuritysite.com/database/99
                   </idmef:url>
      </idmef:Reference>
    </idmef:Classification>
    <idmef:Source ident="a1b2c3d4">
      <idmef:User ident="grid_id_01" category="application">
        <idmef:UserId ident="grid_id_01" type="current-user">
          <idmef:name>badguy</idmef:name>
        </idmef:UserId>
      </idmef:User>
      <idmef:Node ident="a1b2c3d4-001" category="dns">
        <idmef:name>badguy.example.net</idmef:name>
        <idmef:Address category="ipv4-net-mask">
          <idmef:address>192.0.1.2</idmef:address>
          <idmef:netmask>255.255.255.255</idmef:netmask>
        </idmef:Address>
      </idmef:Node>
      <idmef:Process ident="a1a2-03">
        <idmef:name>malicious_gridlet</idmef:name>
        <idmef:pid>10569</idmef:pid>
      </idmef:Process>
    </idmef:Source>
```

```
    <idmef:Target ident="d1c2b3a4">
      <idmef:Node ident="d1c2b3a4-001" category="dns">
        <idmef:name>host1.domain_a.grid.com</idmef:name>
        <idmef:Address category="ipv4-net-mask">
          <idmef:address>192.0.2.4</idmef:address>
          <idmef:netmask>255.255.255.255</idmef:netmask>
        </idmef:Address>
      </idmef:Node>
    </idmef:Target>
    <idmef:AdditionalData type="date-time"
                          meaning="job-start-date-time">
      <idmef:date-time>2006-01-01T01:00:00-03:00</idmef:date-time>
    </idmef:AdditionalData>
    <idmef:AdditionalData type="date-time"
                          meaning="job-finish-date-time">
      <idmef:date-time>2006-01-01T02:00:00-03:00</idmef:date-time>
    </idmef:AdditionalData>
    <idmef:AdditionalData type="string" meaning="%status">
      <idmef:string>sucessful</idmef:string>
    </idmef:AdditionalData>
    <idmef:AdditionalData type="string" meaning="%hosttype">
      <idmef:string>Sun ULTRA Sparc</idmef:string>
    </idmef:AdditionalData>
    <idmef:AdditionalData type="integer" meaning="%wallclocktime">
      <idmef:integer>5424</idmef:integer>
    </idmef:AdditionalData>
    <idmef:AdditionalData type="integer" meaning="%cputime">
      <idmef:integer>59</idmef:integer>
    </idmef:AdditionalData>
    <idmef:AdditionalData type="integer" meaning="%memory">
      <idmef:integer>123</idmef:integer>
    </idmef:AdditionalData>
    <idmef:AdditionalData type="integer" meaning="%storage">
      <idmef:integer>4576</idmef:integer>
    </idmef:AdditionalData>
    <idmef:AdditionalData type="integer" meaning="%network">
      <idmef:integer>442</idmef:integer>
    </idmef:AdditionalData>
  </idmef:Alert>
</idmef:IDMEF-Message>
```