MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
SECRETARIA DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
CURSO DE MESTRADO EM SISTEMAS E COMPUTAÇÃO

FÁBIO DE OLIVEIRA FAGUNDES

Sistema de Monitoramento Passivo de Segurança de Grid Computacional

Rio de Janeiro
2005

INSTITUTO MILITAR DE ENGENHARIA

FÁBIO DE OLIVEIRA FAGUNDES

SISTEMA DE MONITORAMENTO PASSIVO DE SEGURANÇA DE GRID COMPUTACIONAL

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Mestre em Ciências em Sistemas e Computação.

Orientador: Bruno R. Schulze, D.Sc.
Co-orientador: Jauvane C. de Oliveira, Ph.D.

Rio de Janeiro

2005

C2005

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80 – Praia Vermelha
Rio de Janeiro - RJ CEP: 22290-270

004.6 Fagundes, Fábio de Oliveira
F151 **Sistema de Monitoramento Passivo de Segurança de Grid Computacional** / Fábio de Oliveira Fagundes. – Rio de Janeiro : Instituto Militar de Engenharia, 2005.
119 p. : il.

Dissertação (mestrado) - Instituto Militar de Engenharia – Rio de Janeiro, 2005.

1. Segurança da Informação. 2. Computação em Grade. 3. Verificação de Aderência à Norma NBR ISO/IEC 17799. I. Instituto Militar de Engenharia. II. Título

CDD 004.6

INSTITUTO MILITAR DE ENGENHARIA

FÁBIO DE OLIVEIRA FAGUNDES

SISTEMA DE MONITORAMENTO PASSIVO DE SEGURANÇA DE GRID COMPUTACIONAL

Dissertação de Mestrado apresentada no Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Mestre em Ciências em Sistemas e Computação.

Orientador: Prof. Bruno Richard Schulze – D.Sc.

Co-orientador: Prof. Jauvane Cavalcante de Oliveira – Ph.D.

Aprovada em 12 de setembro de 2005 pela seguinte Banca Examinadora:

_____

Prof. Bruno R. Schulze – D.Sc. do LNCC - Presidente

_____

Prof. Jauvane C. de Oliveira – Ph.D. do LNCC

_____

Prof. Antônio Tadeu Azevedo Gomes – D.Sc do LNCC

_____

Prof. Paulo Fernando Ferreira Rosa – Ph.D. do IME

Rio de Janeiro

2005

À minha querida mãe Madalena, ao meu querido pai Edivaldo, ao meu irmão Flávio e à minha namorada Camila Bayma, que sempre me apoiaram em todos os momentos.

AGRADECIMENTOS

A Deus, pela dádiva da vida e por me permitir chegar até aqui.

Agradeço à minha mãe Madalena, ao meu pai Edivaldo e ao meu irmão Flávio, pelo apoio, paciência e entusiasmo.

Agradeço, ao meu grande amor, minha namorada e amiga Camila P. Bayma pela paciência, apoio e carinho.

Ao Departamento de Engenharia de Sistemas do Instituto Militar de Engenharia pela excelência do ensino e oportunidade.

Aos funcionários do Instituto Militar de Engenharia, pelo apoio e pela colaboração nos momentos que mais precisei.

Ao Laboratório Nacional de Computação Científica, pelo apoio à pesquisa realizada.

À CAPES, por financiar a pesquisa no Brasil.

Ao Laboratório Nacional de Computação Científica pelo apoio à pesquisa e ao projeto GIGA Integridade RNP/FINEP/FUNTTEL 2438 pelo apoio financeiro.

Aos amigos e amigas do Laboratório Virtual – LNCC e Laboratório ACIMA – LNCC, pela sua contribuição para que o trabalho fosse realizado.

Ao amigo Luís Rodrigo de Oliveira Gonçalves, pelo excelente e inspirador trabalho com o Tamanduá Mirim.

Aos amigos do Rotaract Club Rio de Janeiro Tijuca, especialmente Fernanda Alcantara, Fernando Egrejas e Thiago Pereira, pelo apoio durante a defesa.

Aos amigos da Gisplan Tecnologia da Geoinformação, especialmente aos seus diretores Marcus Silva e Antônio Machado e Silva, por todo o apoio e compreensão durante os trinta meses do curso.

A todos que de alguma forma contribuíram para a concretização dessa dissertação.

Aos membros da banca examinadora, professores Antônio Tadeu Azevedo Gomes e Paulo Fernando Ferreira Rosa, por aceitarem o convite para compor a banca e nos honrar com as suas presenças, sugestões e correções.

Aos meus orientadores professores Bruno R. Schulze e Jauvane C. de Oliveira, por compartilharem seus conhecimentos, experiência, determinação e,

principalmente, pelo contínuo interesse em apoiar-me durante toda a jornada desse trabalho. Sua orientação, paciência e opiniões foram imprescindíveis.

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
LNCC	Laboratório Nacional de Computação Científica
OGSA	Open Grid Services Architecture
OGSI	Open Grid Services Infrastructure
RNP	Rede Nacional de Ensino e Pesquisa
SGBD	Sistema Gerenciador de Banco de Dados
UDDI	Universal Description, Discover and Integration
UML	Unified Modelling Language
URI	Uniform Resource Identifier
WSDL	Web Services Description Language

RESUMO

Grids computacionais ou ambientes distribuídos em larga escala têm o potencial de se tornarem plataformas poderosas utilizadas pela comunidade de computação distribuída, tanto científica quanto comercial, para a execução de aplicações de grande importância e alto teor computacional.

Estes ambientes computacionais podem estar constituídos por computadores geograficamente dispersos e sob diferentes domínios administrativos, em um ambiente tão vasto, segurança certamente é uma preocupação.

Um dos problemas envolvendo a adoção e a implementação de políticas ou normas de segurança para sistemas de informação consiste em verificar se as regras efetivamente implementadas estão em conformidade com aquelas definidas nos respectivos padrões adotados.

Este trabalho estende e adapta o modelo de verificação automática de aderência proposto por (GONÇALVES, 2005) para as particularidades de grids computacionais. O Sistema de Monitoramento Passivo de Segurança de Grid Computacional - SiMPaSeG é então apresentado como uma solução que aproveita-se do ambiente de execução remota habilitado pelo Middleware da grid - o Globus Toolkit - para verificar se os controles adotados estão efetivamente implementados e se são eficazes.

Esses controles são recomendações sobre medidas e procedimentos que podem ser adotados de modo a salvaguardar a informação. A Norma Internacional de Segurança da Informação ISO/IEC 17799:2005 é a mais importante e internacionalmente adotado padrão de segurança.

ABSTRACT

Large-scale distributed computing environments, or computational grids have the potential to become a powerful and widely adopted platform, for both the scientific and commercial communities, to solve hard computational problems.

A computational grid may have geographically dispersed computers that are usually under distinct administrative domains, and in such a vast territory, security is indeed a concern.

The current work extends the automated checking model presented in (GONÇALVES, 2005) to the grids requirements. The Grid Security Passive Monitoring System - SiMPaSeG (in Portuguese) is then presented as a tool designed to take advantange on the remote job execution environment enabled by the grid middleware – the Globus Toolkit – in order to check if the adopted security controls have actually been implemented and if they are effective.

These controls are recommendations of procedures and measures to safeguard the information. The most important and internationally adopted standard is the ISO/IEC 17799:2005.

1. INTRODUÇÃO

Este trabalho trata do problema da análise de segurança de uma grid computacional e propõe um modelo de verificação automática do nível de aderência desse ambiente à Norma Nacional de Segurança da Informação NBR ISO/IEC 17799:2005.

1.1 Motivação

Grids Computacionais têm o potencial de se tornarem plataformas poderosas utilizadas pela comunidade de computação distribuída, tanto científica quanto comercial, para a execução de aplicações de grande escala e demanda computacional. Seu objetivo é agregar uma coleção de recursos distribuídos, heterogêneos e compartilhados conectados via rede capaz de oferecer capacidade computacional para tais aplicações. Contudo, permanece como desafio a exploração do desempenho de tais recursos, devido principalmente ao comportamento dinâmico e instável deste ambiente.

A existência de redes de longa distância de alta velocidade a baixo custo tem encorajado o desenvolvimento de aplicações que requerem recursos não disponíveis localmente, tirando vantagem de recursos distribuídos geograficamente. Isso representa uma abertura para novos ramos de pesquisa que previamente encontravam-se limitados e sem exploração por razões econômicas e práticas.

Numa grid se pode identificar três camadas distintas, que se comportam como um sistema computacional único e bem integrado:

- Infra-estrutura - componentes de software e hardware, integrados por uma rede física.
- Middleware - camada oferecendo transparência dos recursos disponíveis, com ferramentas para o gerenciamento e controle da infra-estrutura para as aplicações em grid.
- Aplicações - desenvolvidas e otimizadas para tirar vantagem dos recursos distribuídos e do comportamento dinâmico da grid.

Apesar da grande preocupação com segurança no projeto e desenvolvimento de alguns middlewares para grids, para que possam servir ao propósito da grid computacional, um nó ao disponibilizar ciclos de CPU a usuários da grid, estará inevitavelmente permitindo a execução de código arbitrário no seu sistema operacional (SO). Estes processos podem fazer mau uso dos recursos locais e inclusive explorar falhas de segurança existentes no SO para realizar escalação de privilégios. Além disso, a grande complexidade da instalação do middleware, aliada à pouca cultura de segurança, podem levar a configurações de sistema operacional não-padronizadas e pouco seguras.

A Norma Nacional de Segurança da Informação NBR ISO/IEC 17799:2005 enumera uma série de controles que podem ser implementados para garantir a confidencialidade, a integridade e a disponibilidade das informações. Estes controles devem ser revisados regularmente com auxílio de pacote de software automatizado que gere um relatório técnico (NBR17799, 2005).

Os pacotes de softwares automatizados podem ser divididos em duas categorias: os softwares baseados em sistemas especialistas que fornecem apoio à análise de riscos, à seleção de controles e à preparação para auditoria através do uso de formulários; e os que verificam ativamente a conformidade do ambiente sob análise. Do primeiro grupo destacam-se o COBRA e o Proteus Enterprise, enquanto do segundo temos o Check-up Tool e o Tamanduá Mirim. Entretanto, nenhuma dessas ferramentas é apropriada para análise de grids computacionais.

Como não há ferramentas de verificação de segurança de grids computacionais, será apresentado o Sistema de Monitoramento Passivo de Segurança de Grid Computacional (SiMPaSeG), uma solução para verificação automática do nível de aderência de um ambiente de grid à Norma Nacional de Segurança da Informação NBR ISO/IEC 17799:2005, O SiMPaSeG estende e adapta o modelo do Tamanduá Mirim para grids computacionais.

O presente trabalho está sendo aplicado no Laboratório Nacional de Computação Científica (LNCC) dentro do contexto do Projeto InteGridade, que envolve uma grid de desenvolvimento (INTEGRIDADE, 2005) sobre a rede Gigabit experimental do Projeto Giga da Rede Nacional de Ensino e Pesquisa (RNP). Ainda não há políticas de segurança definidas para esta grid, e, portanto não há verificação dos mecanismos de segurança que, eventualmente, estejam sendo adotados.

1.2 Objetivos

São objetivos do presente trabalho:

- Criação de um mecanismo de verificação automática de conformidade de um ambiente distribuído de uma grid computacional à Norma Nacional de Segurança da Informação - NBR ISO/IEC 17799:2005;
- A implementação proposta deve ser multiplataforma e fazer uso de ferramentas de software livre e gratuito;
- A implementação deve possuir interface simples que permita utilização por pessoas sem conhecimento técnico em segurança e grids;
- Servir de ferramenta de diagnóstico e correção de falhas de segurança no ambiente.

1.3 Organização do Documento

O restante deste trabalho está organizado nos seguintes capítulos:

- No capítulo 2 são apresentados e discutidos alguns trabalhos existentes na literatura sobre segurança em grid computacional, relacionados com o presente trabalho;
- No capítulo 3 são apresentados alguns conceitos e tecnologias importantes, tais como: middleware para sistemas distribuídos, cluster, grid, segurança da informação, norma brasileira de segurança da informação, e Globus Toolkit;
- No capítulo 4 é apresentada a proposta para verificação de segurança em grids computacionais. O objetivo é apresentar as motivações, bem como identificar as suas limitações.
- No capítulo 5 é apresentada a implementação do trabalho.
- Na conclusão é apresentado um resumo das contribuições deste trabalho e trabalhos futuros.

2 TRABALHOS RELACIONADOS

Neste capítulo são apresentados trabalhos relacionados com os modelos e propostas utilizadas neste trabalho, com enfoque na abordagem do problema de verificação de segurança dos sistemas apresentados, bem como de suas limitações.

2.1 Introdução

Na literatura, são encontradas algumas propostas de modelos para análise de aderência de sistemas computacionais à Norma Internacional de Segurança da Informação ISO/IEC 17799:2005. Estes modelos e as ferramentas que os implementam, como o COBRA, o Proteus Enterprise, o Check-up Tool e o Tamanduá Mirim, auxiliam na análise de risco e na verificação dos controles da Norma adotados.

Os dois primeiros usam questionários eletrônicos para auxiliar na análise de risco e escolha dos controles, porém não verificam se o ambiente computacional informado corresponde ao real e nem se os controles aplicados são eficazes. O Check-up Tool e o Tamanduá Mirim adotam uma abordagem ativa podendo realizar o levantamento automático do ambiente computacional e a verificação automática de controles da Norma. O Sistema de Monitoramento Passivo de Segurança de Grid Computacional - SiMPaSeG aplica a verificação automática utilizando-se da infra-estrutura disponível da grid computacional.

2.2 COBRA – Consultative, Objective and Bi-Functional Risk Analysis

O COBRA surgiu em 1991 como resposta à crescente necessidade de gestão e verificação do nível de segurança das informações por parte das empresas. É uma ferramenta para DOS ou Microsoft Windows baseada em questionários que utiliza os princípios de sistema especialista e uma extensa base de conhecimento (COBRA,

2005).

A ferramenta vem sendo usada por consultores de segurança e administradores de rede ou gerentes de Tecnologia da Informação (TI) no processo de homologação de um ambiente à ISO/IEC 17799:2005. Suas funcionalidades possibilitam:

- Identificar ameaças, vulnerabilidades e exposição do sistema;
- Mensurar o grau de risco real para cada área ou aspecto do sistema, e relacioná-lo diretamente ao potencial impacto nos negócios;
- Oferecer soluções e recomendações detalhadas a fim de reduzir os riscos;
- Gerar relatórios técnicos e executivos.

A análise de risco de segurança será tão boa quanto as informações fornecidas à ferramenta, pois a mesma não possui mecanismos de verificação do sistema real por inconsistências nos dados informados: os questionários podem ser gerados automaticamente, mas para isto é necessário completar um pré-questionário de "business/impact".

2.3 Proteus Enterprise

O Proteus Enterprise é uma suíte de ferramentas online para análise, gerenciamento e verificação de aderência a normas como: BS7799, ISO/IEC 17799, Basiléia II etc (PROTEUS, 2005).

O Proteus é capaz de gerar relatórios flexíveis em diferentes formatos digitais. Seus relatórios não apenas mostram as deficiências encontradas, mas também informam as implicações e auxiliam na priorização das ações corretivas.

Assim como o COBRA, baseia-se apenas em formulários e por este motivo, a análise de riscos e de aderência será tão boa quanto as informações fornecidas pelos usuários.

2.4 Check-up Tool

O Check-up Tool é uma ferramenta desenvolvida pela Módulo para análise de riscos para ativos tecnológicos e não-tecnológicos. Apoia a implementação dos requisitos de certificação BS 7799, COBIT, de agências reguladoras, Sarbanes-Oxley Act e Basiléia II através da implementação de coletores automáticos para o ambiente Microsoft (Windows, SQL Server, IIS etc) (CHECKUPTOOL, 2005).

Assim como o COBRA e o Proteus, o Check-up Tool também usa questionários eletrônicos e é capaz de gerar automaticamente relatórios, gráficos e estatísticas. Suas funcionalidades, que aliam uma grande base de conhecimento a coletores automáticos, permitem a verificação de conformidade de ambientes MS Windows a diferentes normas de segurança.

2.5 Tamanduá Mirim

O Tamanduá Mirim (GONÇALVES, 2005) é uma ferramenta totalmente desenvolvida em software livre para análise automática do nível de aderência de sistemas computacionais à Norma Nacional de Segurança da Informação NBR ISO/IEC 17799:2005. Essa análise difere da abordagem do sistema denominado de COBRA, que se baseia apenas no uso de questionários eletrônicos, pois coleta várias informações sobre o ambiente de rede, de forma automática, antes de iniciar o processo de verificação da aderência do mesmo à norma de segurança.

A arquitetura modularizada permite a implementação de seus componentes para diferentes plataformas de hardware e software. Além da varredura de rede e identificação do sistema operacional dos nós da rede, o Tamanduá possui quatro componentes básicos acionados de forma seqüencial:

- Módulo de Coleta de Dados: realiza o levantamento do perfil dos nós que devem ser verificados. Em cada nó, esse levantamento determina quais scripts para coleta de informações sobre o nó devem ser executados.
- Módulo Verificador Primário: controla o processo de coleta das informações e realiza a verificação da aderência do ambiente à NBR ISO/IEC 17799:2005. É

responsável pelo envio aos nós de todo e qualquer componente de software necessário, pelo envio dos scripts de verificação e pela gerência da análise.

- Módulo Verificador Secundário: instalado nos nós que estão conectados a mais de um segmento de rede. Sua função é realizar a análise dos segmentos de rede que não podem ser verificados diretamente pelo Módulo Primário.
- Repositório de Dados: responsável pelo armazenamento de todos os dados e informações necessárias para o funcionamento dos demais componentes, bem como pelo armazenamento dos scripts de verificação e das informações coletadas sobre o ambiente. A FIG. 2.1 mostra a tela de abertura do Tamanduá Mirim.

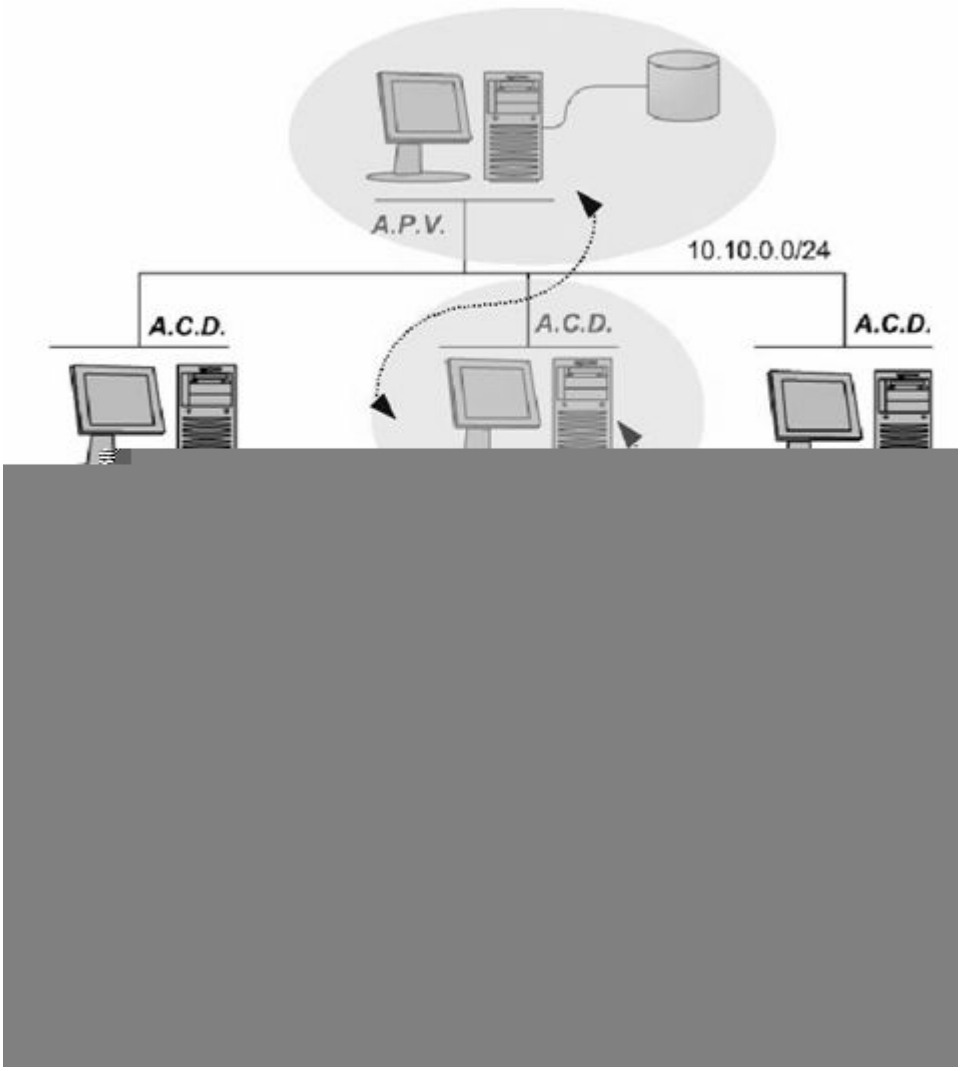

FIG. 2.1 Arquitetura do Tamanduá Mirim

Por seu modelo usar uma arquitetura modular que pode facilmente ser estendida

e reimplementada usando software livre, o Tamanduá Mirim foi escolhido para servir de base para construção do SiMPaSeG.

2.6 SiMPaSeG - Sistema de Monitoramento Passivo de Segurança de Grid Computacional

Do ponto de vista de segurança, os usuários, as aplicações ou o middleware da grid – ou alguma combinação dos 3 – deve ser confiável. Segundo (BUTT, ADABALA, KAPADIA, FIGUEIREDO & FORTES, 2003), o melhor caminho é o de se confiar na segurança das aplicações, visto que os usuários da grid, devido à sobrecarga do gerenciamento das contas ou do gerenciamento dos certificados (BECKLESA, WELCHB & BASNEY, 2005), os usuários da grid são geralmente mapeados para contas compartilhadas por diferentes usuários legítimos e que ainda algumas vezes compartilham certificados digitais, levando a não identificação precisa dos usuários.

Em (BUTT, ADABALA, KAPADIA, FIGUEIREDO & FORTES, 2003), é mostrado ainda que as verificações de segurança das aplicações, sejam elas baseadas no código-fonte, em tempo de compilação, em tempo de “linkagem“ ou de execução – atualmente implementadas na maioria dos ambientes de grid computacional são inadequadas. É proposta então uma técnica de monitoramento em tempo de execução projetada em dois níveis: no primeiro, um shell restrito controla a habilidade do usuário de livremente usar os recursos do sistema operacional durante uma sessão interativa, enquanto no segundo há um módulo de monitoração de chamadas do sistema operacional (system-calls) que determina se o mecanismo de chamada de sistema do kernel deverá ou não permitir que certa chamada monitorada se complete.

O ambiente de grid computacional baseado no Globus Toolkit reconhecidamente não possui mecanismos de autorização de baixa granularidade (KEAHEY & WELCH, 2002). Como o GT também não se preocupa com a segurança das aplicações ao não implementar nada semelhante ao proposto acima, ou já em funcionamento, como no Purdue University Network Computing Hubs (PUNCH) (BUTT, ADABALA, KAPADIA, FIGUEIREDO & FORTES, 2003), a grid, assim implementada, torna-se

um ambiente ideal para usuários mal-intencionados fazerem mau uso dos recursos computacionais e até mesmo praticarem tentativas de escalação de privilégios.

Desse modo, a fim de diminuir a superfície de ataques e os riscos associados à falta de padronização e de preocupação com segurança na configuração dos nós que fazem parte da grid, é, proposto um sistema que verificará, usando as credenciais de um usuário legítimo e não-privilegiado da grid computacional, o nível de aderência de cada nó à Norma.

O Sistema de Monitoramento Passivo de Segurança de Grid Computacional – SiMPaSeG é portanto, um software instalado no portal web de uma grid computacional que permitirá a enumeração dos nós, verificação de controles da Norma e geração de relatórios a partir de um ponto único de acesso.

O SiMPaSeG implementa o APV e o ACD dentro do portal web de submissão da grid.

3 CONCEITOS E TECNOLOGIAS

Neste capítulo são apresentados alguns conceitos e tecnologias importantes com relação à presente dissertação, tais como: middleware para sistemas distribuídos, cluster, grid, segurança da informação, Norma Brasileira de Segurança da Informação e Globus Toolkit.

3.1 Middleware para Sistemas Distribuídos

Um sistema distribuído é uma coleção de computadores independentes que pode se apresentar ao usuário como um único sistema coerente. As diferenças entre esses computadores e como eles se comunicam são transparentes ao usuário.

Outras importantes características de sistemas distribuídos são que usuários e aplicações devem poder interagir com eles de uma forma uniforme e consistente, independente de onde e quando a interação ocorrer; devem ser ainda relativamente fáceis de expandir e escalar, ao mesmo tempo em que deve estar continuamente disponível, embora alguma de suas partes possam estar temporariamente desligadas e terem sido substituídas por outras.

A fim de suportar computadores e redes heterogêneas, um sistema distribuído é geralmente implementado como uma camada lógica entre a camada superior composta por usuários e aplicações e a camada abaixo composta dos sistemas operacionais, e por esse motivo é que tal sistema distribuído é comumente chamado de Middleware.

Num sistema distribuído, aos usuários deve ser fácil acessar recursos remotos e compartilhar recursos com outros usuários de um modo controlado, ao mesmo tempo em que esconde o fato que os processos e recursos estão fisicamente distribuídos em múltiplos computadores, ou seja, deve ser transparente.

Da mesma forma, transparência também se aplica a muitos outros aspectos de um sistema distribuído, como por exemplo: acesso, localização, migração, relocação, replicação, concorrência, falha e persistência.(TANENBAUM & STEEN, 2002)

3.2 Cluster

Um ambiente cluster ou aglomerado computacional é formado por hardware e software homogêneos conectados por uma rede de computadores rápida e dedicada (baixa latência e grande largura de banda, ex.: Gigabit Ethernet e Myrinet) ou mesmo por rede local, onde os computadores ou nós estão localizados fisicamente em um único local e são gerenciados por uma entidade central.

Aglomerados computacionais surgem do avanço das redes de computadores e do aumento da capacidade dos processadores; desta forma, o pico de desempenho do aglomerado, mesmo depois de descontados os tempos de comunicação e transferência de dados entre os nós, muitas vezes iguala ou supera o poder computacional oferecido pelos mais rápidos computadores paralelos. A 25ª lista dos 500 maiores supercomputadores de junho de 2005 mostra a potência dos aglomerados que são representados por 304 sistemas (TOP500, 2005).

3.3 Grid

Ambientes computacionais distribuídos de grande escala ou grid, segundo (FOSTER, 2005), são ambientes capazes de virtualizar recursos computacionais geograficamente distribuídos como se fossem um grande supercomputador. Esses recursos, como processamento, armazenamento, entre outros, podem estar sob o controle de diferentes organizações.

Segundo (BUYYA, 2005), a principal diferença entre grid e cluster está relacionada ao modo como é feito o gerenciamento dos recursos, ou seja, se o compartilhamento de recursos é gerenciado por um único sistema global, sincronizado e centralizado, então é um Cluster.

Em uma grid, os recursos estão distribuídos geograficamente, sendo que os donos desses recursos possuem autonomia para fazer o gerenciamento local. A alocação dos recursos é feita pelos usuários da grid, e os nós executam diferentes tarefas relacionadas a objetivos distintos. Uma grid pode ser encarada como sendo uma evolução do cluster, dado que os recursos deste podem ser compartilhados

pela grid. Assim, uma grid é mais heterogênea, complexa e distribuída.

Ian Foster, em (FOSTER, 2002), elaborou uma Grid Checklist onde define três características básicas para que um sistema computacional possa ser considerado uma grid:

- Recursos coordenados não sujeitos a um controle centralizado: Uma grid integra e coordena recursos e usuários residentes em diferentes domínios administrativos, apesar de poder haver um controle local dentro de uma organização, não pode existir um controle central para toda a grid.
- Utilizar padrões abertos, interfaces e protocolos de propósito geral: Uma grid deve adotar protocolos e interfaces padrões e abertas para alcançar escalabilidade e interoperabilidade entre diferentes padrões de hardware e software, a fim de realizar funções fundamentais como autenticação, autorização, descobrimento de recursos e acesso a eles;
- Prover o mínimo em qualidade de serviços (QoS): Uma grid deve permitir que seus recursos constituintes possam ser usados de forma coordenada de modo a atingir diferentes qualidades de serviço, como por exemplo - tempo de resposta, vazão, disponibilidade, segurança e/ou a co-alocação de múltiplos recursos, para se adequar a demandas complexas do usuário, de modo que a utilidade combinada do sistema seja significativamente maior que a soma das partes.

3.4 Segurança da Informação

Informações podem ser consideradas ativos, e assim sendo, possuem valor para uma organização e consequentemente devem ser protegidos. Essas informações podem existir sob diferentes formas, e quaisquer que sejam as formas que assumam ou meios pelos quais sejam armazenadas e compartilhadas, elas devem ser protegidas adequadamente.

Em geral, segurança da informação é caracterizada pela preservação da confidencialidade, integridade e disponibilidade das informações:

- Confidencialidade visa garantir que as informações somente sejam acessíveis

a pessoas autorizadas;

- Integridade busca salvaguardar a exatidão, completude e métodos de processamento;
- Disponibilidade visa garantir que as informações estejam acessíveis aos usuários autorizados, quando necessário.

A implementação de um conjunto adequado de controles é necessário a fim de se atingir os objetivos de segurança de uma organização.

3.5 Norma Nacional de Segurança da Informação (NBR ISO/IEC 17799)

A NBR ISO/IEC 17799:2005 é baseada na Norma Internacional de Segurança da Informação - ISO/IEC 17799:2005. A NBR ISO/IEC 17799 foi homologada em setembro de 2001 pela Associação Brasileira de Normas Técnicas (ABNT) e foi atualizada em agosto de 2005. Sua publicação coloca o país no conjunto de países que adotam e apóiam o uso dessa norma. Esta versão vem sendo usada em outros países de Língua Portuguesa, entre eles, Portugal e Angola.

A Norma é dividida em 10 macro controles:

- Política de Segurança;
- Segurança Organizacional;
- Classificação e Controle dos Ativos da Informação;
- Segurança em Pessoas;
- Segurança Física e do ambiente;
- Gerenciamento de Operações e Comunicações;
- Controle de Acesso;
- Desenvolvimento da Segurança de Sistemas;
- Gestão da Continuidade do Negócio;
- Conformidade.

Estes macros controles estão subdivididos em vários outros controles, num total de 127 controles de segurança (GONCALVES, 2005). Apenas alguns desses controles podem ser verificados automaticamente por ferramentas como o Tamanduá Mirim ou SiMPaSeG. Tais controles e o modo como são verificados estão

descritos no capítulo 4. Para todos os demais controles consulte o (GONCALVES, 2005, Apêndice B) ou a própria NBR.

Segundo a Norma, deve-se estabelecer os requisitos de segurança baseando-se em três fontes: a primeira, a partir de uma avaliação de riscos; a segunda, são as exigências legais, estatutárias, regulamentadoras e contratuais; enquanto a terceira, vem do conjunto de princípios, objetivos e requisitos para processamento de informações próprio da organização.

Os requisitos de segurança são identificados através de uma avaliação metódica dos riscos de segurança, este sendo uma consideração sistemática de:

- provável prejuízo ao negócio resultante de uma falha de segurança, considerando as conseqüências potenciais de uma perda de confiabilidade, integridade ou disponibilidade das informações e outros ativos;
- a probabilidade realística de tais falhas ocorrerem sob a luz de ameaças e vulnerabilidades prevalecentes, e os controles atualmente implementados.

O processo de avaliar riscos e selecionar controles deve ser executado continuamente até que sejam analisadas todas as partes da organização e sistemas de informação individuais.

Deve-se executar revisões periódicas dos riscos de segurança e dos controles implementados para:

- levar em conta as mudanças nas prioridades e necessidades do negócio;
- considerar novas ameaças e vulnerabilidades;
- confirmar que os controles permanecem eficazes e apropriados.

Um ambiente é aderente à Norma de Segurança da Informação e, portanto, seguro, se o mesmo utiliza os recursos adequados para garantir a confidencialidade, integridade e disponibilidade de suas informações, ou seja, uma vez que os requisitos de segurança tenham sido identificados, devem ser selecionados e implementados controles para garantir que os riscos sejam reduzidos a um nível aceitável.

A seleção desses controles deve ser feita a partir da Norma, de outros conjuntos de controles, ou de novos controles projetados de modo a satisfazer as necessidades específicas da organização. Vai depender também das características do próprio ambiente, como por exemplo: forma e local de armazenamento das informações, valor das informações armazenadas, quem pode acessá-las, quais

servidores estão instalados, que tipo de serviços são disponibilizados aos usuários da rede interna, externa etc, e se considerando os custos de implementação em relação aos riscos que se quer reduzir e aos prejuízos potenciais se ocorrer uma quebra de segurança.

Segundo a Norma, são controles essenciais para uma organização, do ponto de vista legal:

- proteção de dados e privacidade de informações pessoais;
- salvaguarda de registros organizacionais;
- direitos de propriedade intelectual.

E os controles considerados como a melhor prática comum para segurança de informações:

- Documento de política de segurança de informações – descreve quais atividades os usuários estão autorizados a realizar, como e quando podem ser realizadas. É de vital importância que a alta administração apóie o uso da Política e demonstre o seu comprometimento com a aplicação das penalidades cabíveis;
- Alocação de responsabilidades quanto à segurança das informações – este controle visa esclarecer a quem pertence cada ativo da organização, bem como quem deve ser contatado em caso de problemas de segurança relacionados ao ativo em questão;
- Educação e treinamento para segurança das informações – a melhor forma de evitar o mau uso das informações é educar seus usuários, assim é de vital importância que todo e qualquer usuário passe por um treinamento antes de ter acesso às informações contidas no ambiente;
- Relatórios dos incidentes de segurança – estes documentos permitem a criação de uma "base de conhecimento" que poderá ser utilizada para identificar e evitar futuros incidentes de segurança;
- Gerenciamento da continuidade do negócio – este controle diz respeito ao processo de se manter as informações íntegras, sempre acessíveis mesmo quando parte do ambiente estiver comprometido.

3.6 Globus Toolkit

No final de 1994, Rick Stevens, diretor da Mathematics and Computer Science Division do Argonne National Laboratory e Tom DeFanti, diretor do Electronic Visualization Laboratory da Universidade de Illinois em Chicago, propuseram que se estabelecessem links temporários entre 11 redes de pesquisa de alta-velocidade para criar uma grid nacional (o I-WAY), por duas semanas antes e durante a Conferência em Supercomputação de 1995. Um pequeno time liderado por Ian Foster de Argonne criou novos protocolos que permitiram aos usuários do I-WAY executarem aplicações em computadores através dos EUA. Este experimento bem sucedido levou ao financiamento pela Defense Advanced Research Projects Agency (DARPA), e ainda em 1997, foram apresentados à primeira versão do Globus Toolkit.

Desde a versão 1.0 (em 1998) à versão 2.0 (lançada em 2002) e, agora, a versão mais recente 4.0 (lançada em abril de 2005) baseada em novos padrões abertos de serviços em grid, o Globus Toolkit evoluiu rapidamente para se tornar o padrão de facto em computação em grid.

O Globus Toolkit (GT) é um conjunto de ferramentas e bibliotecas de software de código-aberto que dão suporte à arquitetura e às aplicações em grid, permitindo que pessoas compartilhem poder computacional, bases de dados, e outras ferramentas de modo seguro em linha através de diferentes corporações, instituições e limites geográficos, sem contudo, sacrificar a autonomia local. O Toolkit inclui serviços e bibliotecas para segurança, busca de informações, gerenciamento de recursos e de dados, comunicação, detecção de falhas e portabilidade (DANTAS, ALLEMAND & PASSOS, 2003).

O GT é empacotado em um conjunto de componentes que podem ser usados de modo tanto independente quanto conjunto para desenvolver aplicações. Seus serviços centrais, interfaces e protocolos permitem que se acessem recursos remotos como se estivessem localizados em sua própria estação de trabalho ao mesmo tempo em que preservam o controle local a respeito de quem pode usar quais recursos e quando. Dessa forma, sua concepção foi no sentido de remover obstáculos que evitariam colaboração transparente, como incompatibilidades de

formatos de arquivos, computadores, e redes.

3.6.1 Principais Componentes

Os serviços disponibilizados pelo Globus Toolkit foram desenvolvidos na linguagem de programação C, com seu código fonte aberto, o que possibilita a colaboração de modo a validar a implementação, reparar erros ou fazer quaisquer alterações desejadas. O GT, depois de instalado e configurado, propicia transparência através de suas funções e comandos, independente de arquitetura de hardware e sistema operacional.

A seguir, uma breve descrição dos principais serviços do Globus: Grid Resource Allocation and Management (GRAM), Grid Security Infrastructure (GSI) e Monitoring & Discovery Service (MDS).

A FIG. 3.1 (FOSTER, 2005) mostra uma visão geral dos componentes do Globus Toolkit 4.

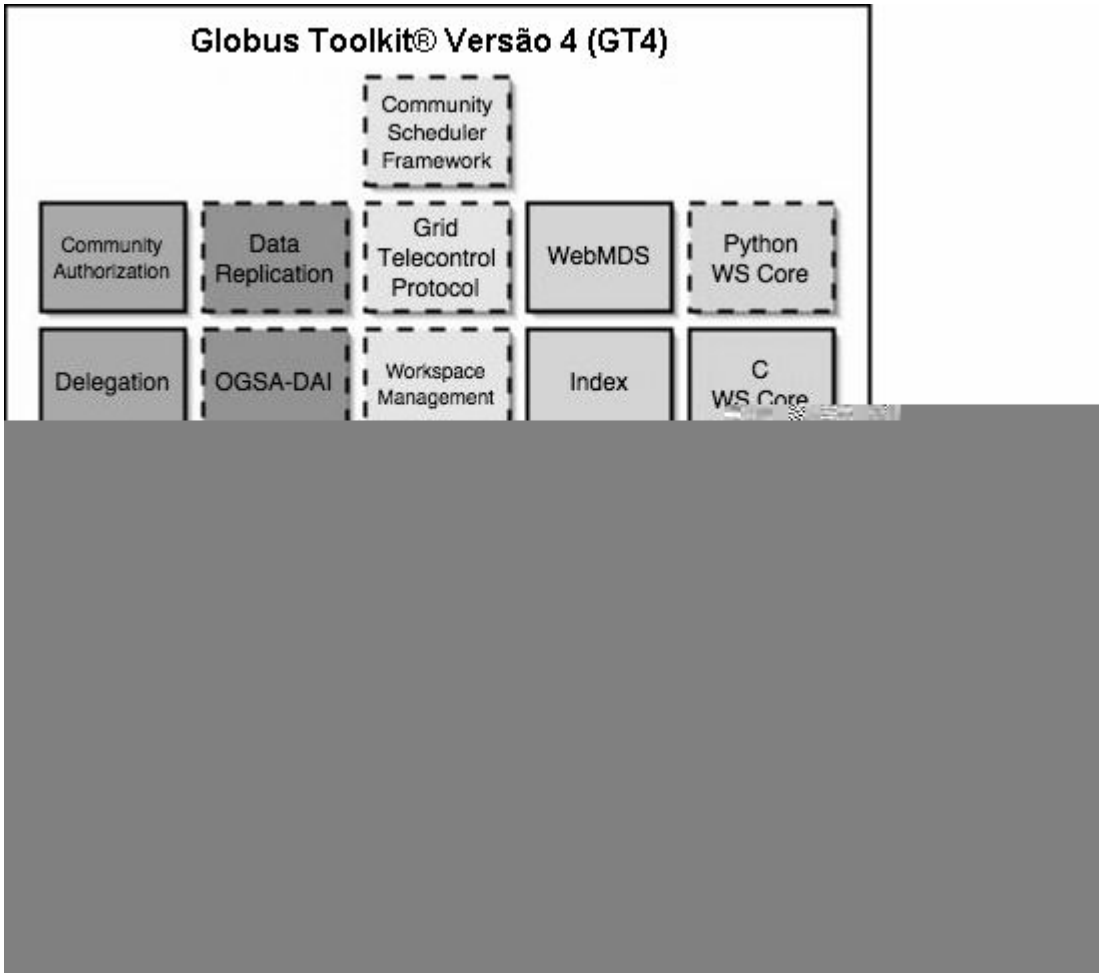

FIG. 3.1 Visão geral dos componentes do Globus Toolkit 4

3.6.2 Grid Information Services

O Grid Information Services provê informações sobre os recursos da grid. Esses serviços incluem o Monitoring and Discovery Service (MDS), composto do Grid Resource Information Service (GRIS) e do Grid Index Information Service (GIIS).

O MDS possibilita o acesso a informações estáticas e dinâmicas sobre a grid e todos os seus componentes: redes, nós, sistemas de armazenagem e instrumentos. Essas informações devem possuir uma forma de acesso uniforme e flexível com manutenção descentralizada. O MDS deve ainda ser configurável e adaptável a ambientes heterogêneos e dinâmicos para que possa acessar múltiplas fontes de informação.

A versão do MDS atualmente utilizada pelo Portal da Grid no qual este trabalho

se insere é o que acompanha a versão 2.4 do Globus Toolkit, a MDS2. Nesta versão, o MDS utiliza o Lightweight Directory Access Protocol (LDAP) como uma interface uniforme para acessar tais informações. O MDS2 foi incluído nas versões mais novas do Globus Toolkit (3.0, 3.2, 4.0 e 4.1) de modo a suportar as implementações legadas. O MDS2 é baseado em OpenLDAP e implementa um mecanismo de redirecionamento com uma combinação de GRIS e GIIS escutando na porta TCP 2135 e respondendo a consultas baseadas na 'string' "Mds-VO-name". Se "Mds-VO-name=local" então se refere ao GRIS e qualquer outra coisa se refere ao GIIS se este existir com aquele nome.

Os dados fornecidos pelos provedores de informações do MDS incluem status de carga de processamento corrente, configuração de CPU, sistema operacional e versão, informações sobre o sistema de arquivos, memória RAM e memória virtual, interfaces de rede, entre outras informações.

A FIG. 3.1 mostra a visão geral dos componente do Globus 4, nesta visão MDS2 está representado como um componente que está obsoleto e que será descontinuado nas próximas versões. Para mais informações sobre o novo MDS4, favor consulte (FOSTER, 2005).

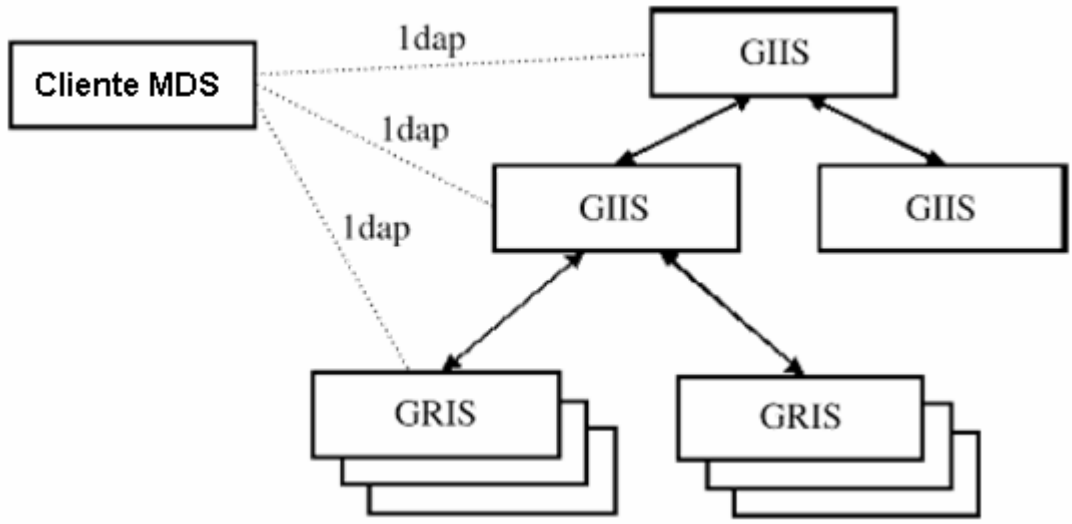

FIG. 3.2 Arquitetura do MDS é uma hierarquia flexível

A FIG. 3.2 (GLOBUSALLIANCE, 2003) mostra que a arquitetura do MDS2 é uma hierarquia flexível, pois podem haver diferentes níveis de GIIS, e qualquer GRIS pode se registrar com qualquer GIIS, e qualquer GIIS pode se registrar com outro, o que torna essa abordagem modular e extensível.

3.6.3 Grid Resource Allocation and Management (GRAM)

O Grid Resource Allocation and Management (GRAM) é o componente do Globus Toolkit responsável pelas interfaces que fornecem os mecanismos básicos para gerenciamento de recursos: inicialização, monitoramento, gerenciamento, escalonamento e coordenação de computação remota.

Os recursos computacionais numa grid computacional são geralmente operados sob o controle de um escalonador que implementa políticas de priorização e alocação de recursos ao mesmo tempo em que otimizam a execução de todas as tarefas submetidas visando eficiência e desempenho. O GRAM não é um escalonador, mas seus serviços estabelecem um protocolo usando mensagem em formato padrão para se comunicar com diferentes escalonadores dos recursos locais.

O GRAM geralmente é instalado juntamente com os serviços de delegação (Delegation) e transferência confiável de arquivos (RFT) para possibilitar apresentação de dados, delegação de credenciais proxy, e gerenciamento e monitoramento de tarefas de uma maneira integrada.

A submissão de tarefas é feita usando a Resource Specification Language (RSL) que permite a definição dos parâmetros de execução como, por exemplo, número de processadores, argumentos e limite de uso ou de tempo de CPU, entre outros.

A FIG 3.3 mostra o envio de uma requisição de tarefa de um usuário a um Gatekeeper e o processo de execução dessa tarefa em um recurso remoto. Essa requisição é feita em RSL ao Gatekeeper, que é o responsável por verificar se o usuário possui autorização aos recursos solicitados . Esta verificação é feita usando um arquivo em disco – grid-mapfile – que possui o mapeamento de cada usuário da grid a um usuário local no recurso solicitado. Após autorizado, o Gatekeeper cria uma instância de Job Manager usando as credencias do usuário local para o qual o usuário da grid foi mapeado. Este Job Manager é quem inicia os processos para realização da tarefa nesse recurso.

Ao contrário de ser uma solução monolítica, o GRAM é baseado numa arquitetura de componentes nos níveis de protocolo e implementação de software.

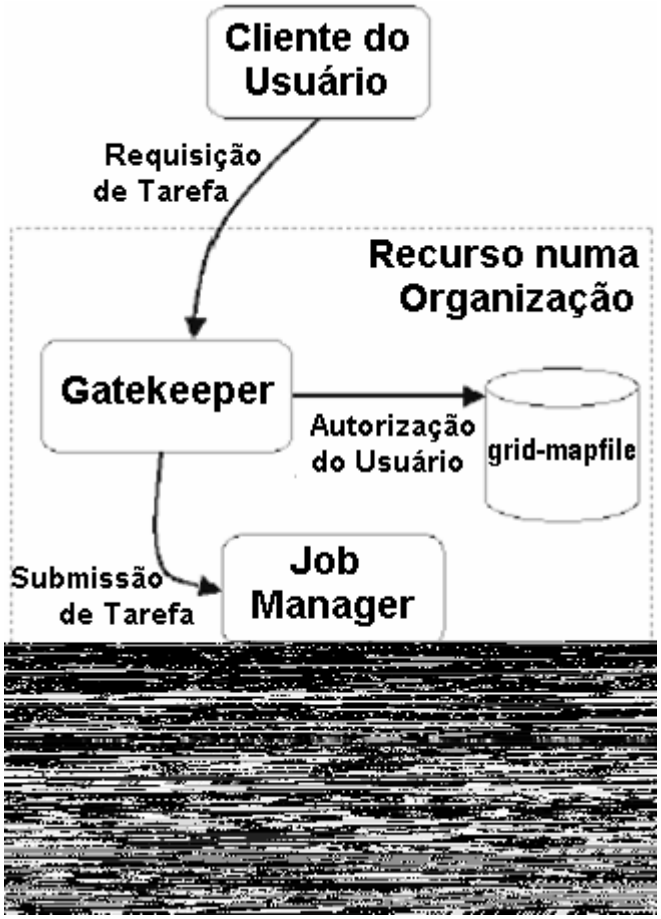

FIG. 3.3 GRAM em ação

3.6.4 Grid Security Infrastructure (GSI)

O Grid Security Infrastructure (GSI) do Globus Toolkit (GT) é a parte deste middleware responsável pelos serviços de segurança fundamentais ao suporte de grid computacionais, para isso usando de criptografia de chave-pública ou assimétrica como base de suas funcionalidades. O GSI implementa diferentes padrões de diferentes entidades padronizadoras em seus componentes Web Services e Não-Web Services (FIG. 3.1).

O suporte a segurança em nível de mensagem é importante por questões de conformidade ao WS-Interoperability Basic Security Profile, porém as implementações atuais apresentam baixo desempenho, motivo pelo qual a segurança em nível de transporte é habilitada por padrão no Globus Toolkit.

O GSI implementa diferentes padrões a fim de fornecer proteção à mensagem

(confidencialidade, integridade, prevenção a replay e não-repúdio), autenticação, delegação e autorização conforme mostra a FIG. 3.4:

- TLS (em nível de transporte) ou WS-Security e WS-SecureConversation (em nível de mensagem) são usados como mecanismos de proteção às mensagens em combinação com o SOAP;
- Certificados X.509 de entidade final ou usuário e senha são usados como credenciais de autenticação;
- Certificados Proxy X.509 e WS-Trust são usados para delegação;
- Diretivas Security Association Markup Language (SAML) são usadas para autorização.

FIG. 3.4 Visão geral do GT4 GSI e dos padrões usados.

A FIG. 3.4 mostra uma visão geral do GT4 Grid Security Infrastructure e dos padrões usados para diferentes funcionalidades. As duas caixas da esquerda mostram a segurança em nível da mensagem, com credenciais X.509 e autenticação por usuário/senha. A caixa da direita mostra a segurança em nível de transporte com credenciais X.509.

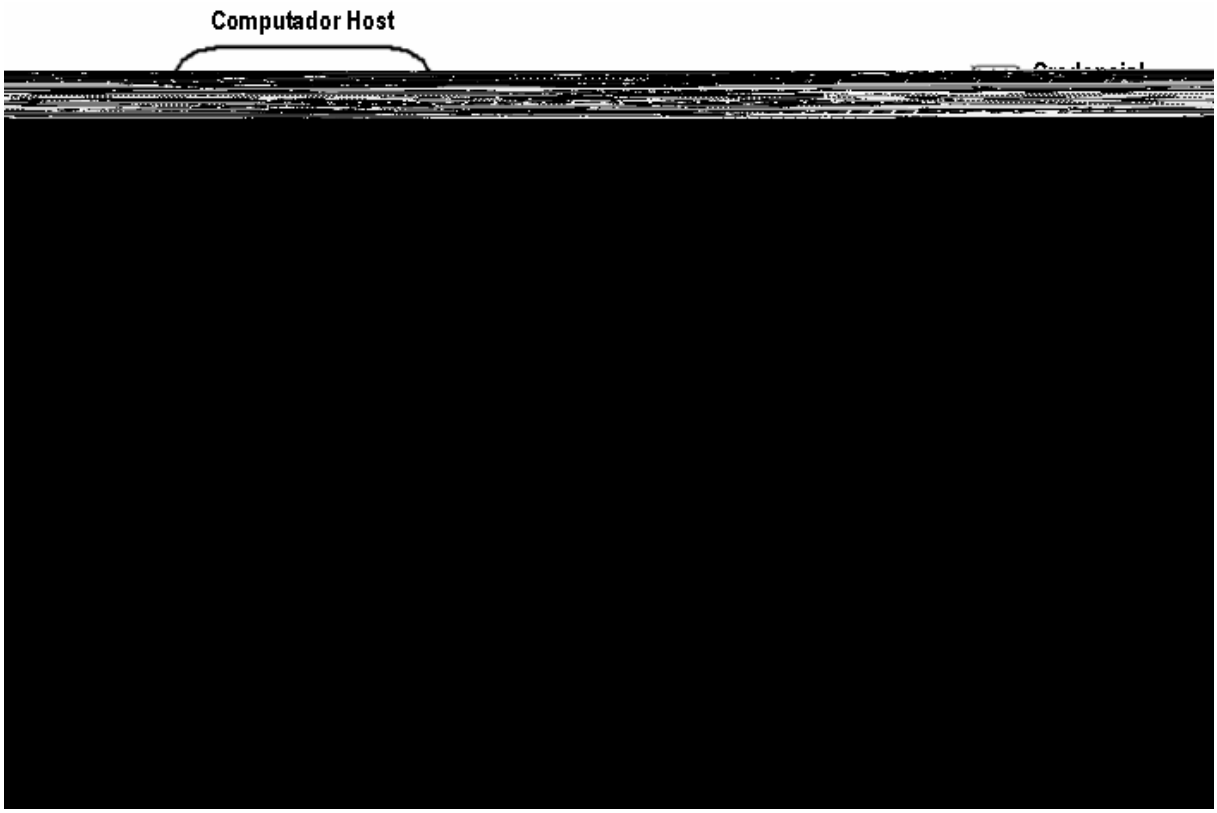

FIG. 3.5 GSI em ação.

A FIG. 3.5 mostra o GT4 Grid Security Infrastructure em ação: um usuário usa seu certificado X.509 para criar um proxy que agirá em seu nome para execução de processos em um sítio remoto. O certificado do usuário (C_U) é usado na criação de um proxy que servirá de mecanismo para delegação de credenciais do usuário. Após criado, o proxy do usuário aloca recursos em um sítio remoto através da criação de um proxy nesse recurso usando-se o certificado (C_{EP}). Com o mapeamento das credenciais do usuário da grid para credenciais locais através do grid-mapfile, os processos remotos já podem ser criados em nome do usuário.

4 PROPOSTA

Neste capítulo é apresentada uma proposta para verificação de segurança em grids computacionais. O objetivo é apresentar as motivações e a arquitetura proposta, bem como identificar as suas limitações.

4.1 Introdução

Uma vez apresentados os desafios de segurança num ambiente de grid computacional, cabe mostrar uma arquitetura para análise de conformidade de ambientes computacionais distribuídos de grande escala, ou grids. Esta arquitetura irá permitir a atenuação dos riscos pela verificação periódica dos mecanismos que implementam a política de segurança.

4.2 Rede GIGA

A Rede GIGA, uma rede óptica experimental com mais de 735km de extensão e capacidade de 2,5 Gbps extensíveis a 10 Gbps, é um dos objetivos do Projeto GIGA (GIGA, 2005). A rede utiliza tecnologias ópticas e protocolo Internet para interligar 17 entidades de pesquisa nas cidades de Campinas, São Paulo, São José dos Campos, Cachoeira Paulista, Rio de Janeiro e Petrópolis, conforme mostra a FIG. 4.1.

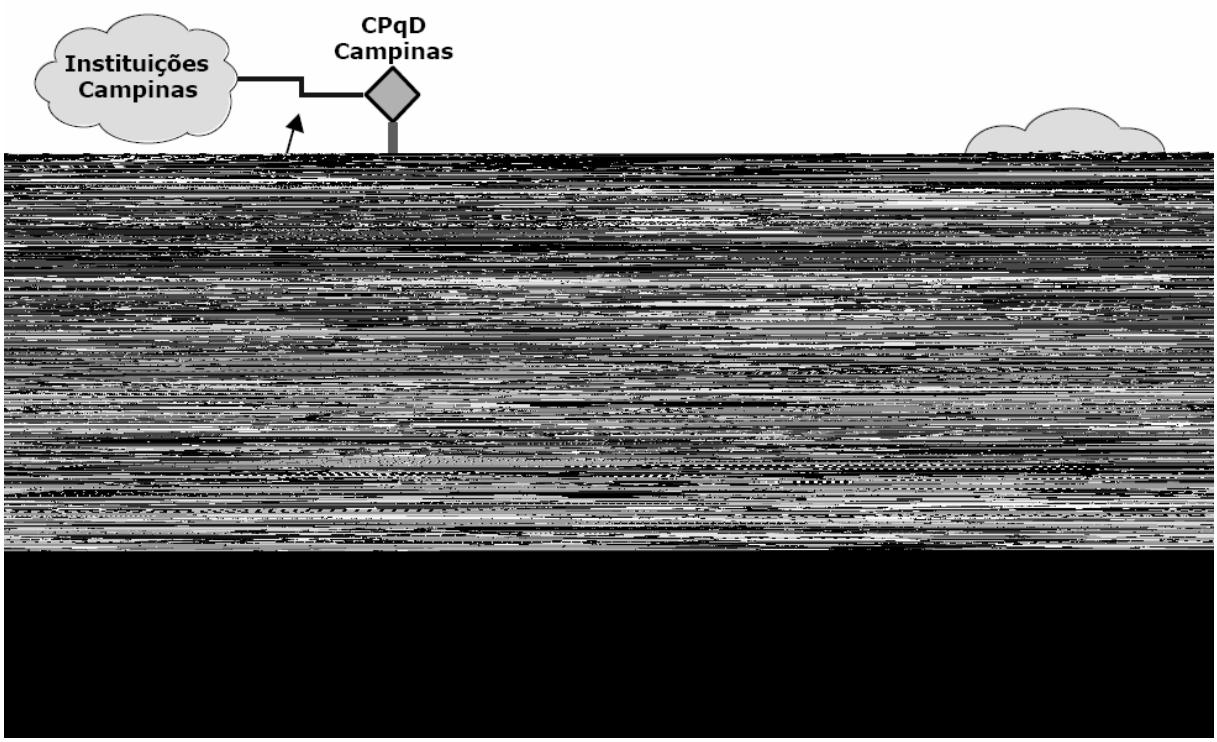

FIG. 4.1 Diagrama físico da Rede Giga

O Projeto GIGA é desenvolvido em parceria com a RNP (Rede Nacional de Ensino e Pesquisa) e com a Fundação CPqD (Centro de Pesquisa e Desenvolvimento em Telecomunicações), com financiamento do FUNTTEL (Fundo para o Desenvolvimento Tecnológico das Telecomunicações) e apoio da FINEP (Financiadora de Estudos e Projetos). O Projeto GIGA possui 33 subprojetos aprovados, dentre os quais encontra-se o subprojeto "InteGridade: Desenvolvimentos em Middleware para Grids Computacionais sobre a Rede Giga", para o qual este trabalho é contribuição.

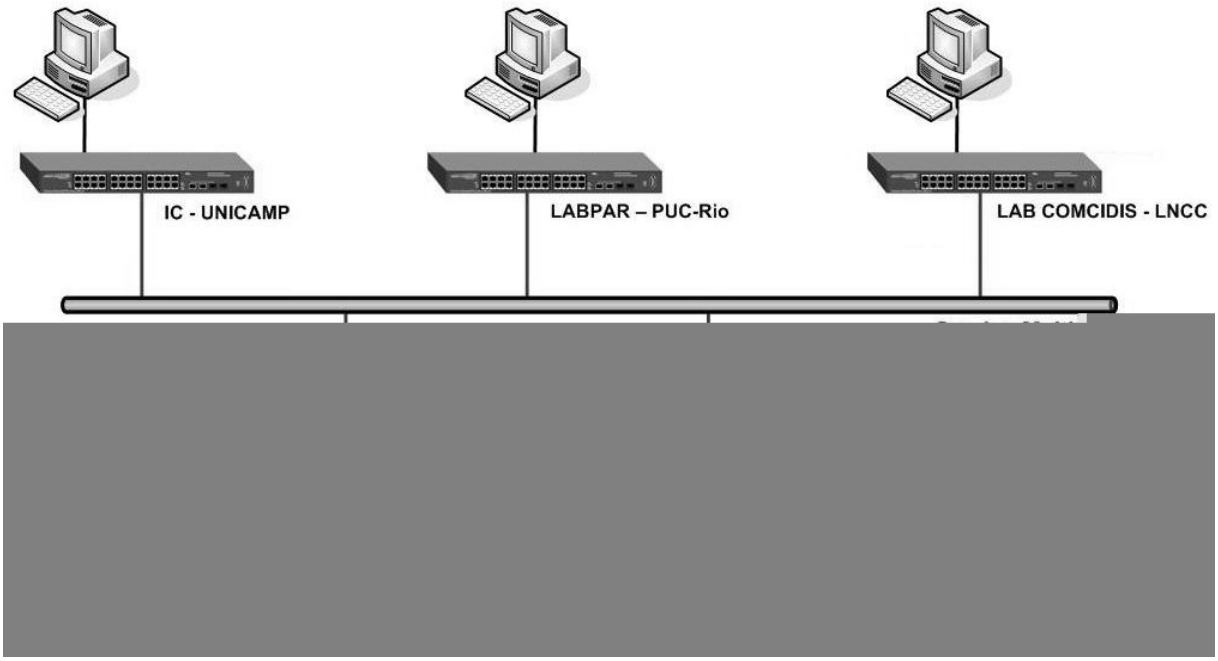

FIG. 4.2 Diagrama lógico de endereçamento IP da Rede Giga

O Projeto InteGridade - Grid Sinergia define uma grid computacional de desenvolvimento sobre a Rede Giga, com a participação das seguintes instituições: ComCIDis/LNCC, IC/UFF, LABPAR/PUC-RJ, CAT/CBPF, IC/UNICAMP e INF/UFRGS, esta última conectada via Internet. A FIG. 4.2 mostra o diagrama lógico da Rede Giga, no que concerne ao Projeto InteGridade.

4.3 Problema de Segurança

A grid computacional faz uso de uma rede de computadores. Realizar a análise de riscos e verificar se este ambiente atende aos controles da NBR implementados não é uma tarefa fácil. Portanto, ferramentas como o "Consultative, Objective And Bi-Functional Risk Analysis" (COBRA), "Check-up Tool" da Módulo e o "Tamanduá Mirim" foram criadas. O primeiro faz uso de questionários eletrônicos para geração de relatórios, o segundo implementa verificação da aderência aos requisitos das normas BS 7799, CobiT (COBIT, 2005), Sarbanes-Oxley (SARBOX, 2005) e Basiléia II (BASILÉIA-II, 2005) através de uma estrutura centralizada para plataforma Microsoft Windows, enquanto o último implementa uma verificação automática da aderência aos requisitos da NBR usando uma arquitetura de software livre.

4.4 SiMPaSeG

O Sistema de Monitoramento Passivo de Segurança de Grid Computacional (SiMPaSeG) tem como objetivo o desenvolvimento e a implantação de ferramentas para verificação de conformidade de um ambiente de grid computacional aos mecanismos que implementam a política de segurança neste ambiente.

Estes mecanismos são implementados tendo-se como base os seguintes controles da NBR ISO/IEC-17799:

- 3.1.1 - Inventário dos ativos;
- 4.3.2 - Reportando pontos fracos na segurança;
- 4.3.3 - Reportando mau funcionamento de softwares;
- 7.2.1 - Registro de usuário;
- 7.2.2 - Gerência de Privilégios;
- 7.2.3 - Gerenciamento de Senha dos Usuários;
- 7.3.1 - Uso de senhas;
- 7.4.3 - Autenticação para conexão externa do usuário;
- 7.5.2 - Procedimentos de entrada no sistema;
- 7.5.3 - Identificação e autenticação do usuário;
- 7.5.4 - Sistema de gerenciamento de senhas;
- 7.5.5 - Uso de programas utilitários;
- 7.5.7 - Desconexão do Terminal por Inatividade;
- 7.5.8 - Limitação do Tempo de Conexão;
- 7.7.3 - Sincronização de relógios;

O ambiente de grid computacional baseia-se no Globus Toolkit (GT) e seu acesso muitas vezes é facilitado pela criação e uso de um Portal Web. Este portal, instalado num servidor de aplicações J2EE, pode ser acessado por navegador web de modo a auxiliar na criação do proxy para autenticação dos processos na grid, na submissão de trabalhos sequenciais e distribuídos, entre outras funcionalidades. Esta implementação do SiMPaSeG é parte integrante do portal do Projeto InteGridade.

4.4.1 Portal Web da Grid

O Portal Web da grid é um programa servidor instalado em um servidor de aplicações como o Apache Tomcat ou Oracle 9iAS que fornece acesso a recursos e serviços da grid computacional a partir de um navegador web. Este portal permite o acesso seguro e personalizado de usuários autenticados aos serviços da grid. Entre estes serviços, se destacam a submissão de tarefas sequenciais, distribuídas e paralelas para execução na grid, a transferência confiável de arquivos, a compilação de aplicações paralelas e distribuídas, e a divulgação de informações sobre a tecnologia de grid computacional e de uso.

Este portal vem sendo desenvolvido usando a linguagem de programação Java e, mais recentemente, a linguagem de programação LUA (FIGUEIREDO, IERUSALIMSCHY & CELES, 1994).

A arquitetura deste portal é muito semelhante ao descrito em (NOVOTNY, 2000), ou seja, faz uso do padrão de projeto MVC (Model-View-Controller) que separa uma aplicação em três componentes distintos: modelo de negócio, interface de usuário e controle lógico, a fim de minimizar os impactos das modificações de um componente nos demais; utiliza a Application Programming Interface (API) Java 2 Enterprise Edition (J2EE) como estrutura básica para páginas dinâmicas programadas como Java Servlets e Java Servlet Pages (JSP), e o Java Commodity Grid Toolkits (Java CoG Kit) para interação com o Globus Toolkit (GT); e ainda utiliza o protocolo HTTPS de modo a garantir a confidencialidade e integridade da comunicação entre o portal e os navegadores web.

4.4.1.1 Acesso ao portal/grid computacional

O acesso ao portal é controlado por um mecanismo de autenticação usando o Java CoG Kit. Um usuário somente tem acesso ao portal após informar um nome de usuário e senha válidos e capazes de se autenticar no Globus Security Infrastructure (GSI) e gerar um proxy que vai servir de mecanismo para delegação de credencial a fim de reduzir o número de vezes que um usuário deve fornecer sua senha. Este proxy consiste num novo certificado e chave particular que permite que o portal atue

em nome do usuário nas demais interações com a grid.

4.4.2 Módulos do SiMPaSeG

O SiMPaSeG possui 4 módulos:

- Varredura da grid: responsável pelo levantamento das informações publicadas no serviço de informações do Globus (MDS) e pela identificação dos nós ativos da grid computacional;
- Enumeração de Portas e Fingerprint do sistema operacional: responsável por identificar as portas TCP/IP abertas e o sistema operacional dos nós ativos;
- Verificação de conformidade: responsável pela execução dos scripts e demais componentes de software necessários para verificação de aderência à NBR;
- Geração de Relatórios e Envio de Alertas: responsável pelo processamento, análise e apresentação das informações coletas sobre os controles verificados.

4.4.2.1 Varredura da grid

A verificação de conformidade é iniciada com a enumeração de todos os nós ativos da grid computacional. Esta abordagem difere da abordagem do Tamanduá Mirim na medida em que estes recursos computacionais podem estar espalhados em diferentes domínios administrativos e consequentemente pertencerem a diferentes redes e sub-redes IP, ao mesmo tempo em que outros computadores nestas sub-redes serão ignorados por este mecanismo se estes não pertencerem à grid.

Esta varredura ocorre em duas etapas:

- Consulta a servidores Globus MDS disponíveis na grid pelas informações de registro dos nós. Estas informações incluem FQDN, endereço IP, características do hardware - como processador e memória - e do sistema operacional instalado. Os servidores MDS (GIIS e GRIS) possuem distribuição hierárquica e permitem, em geral, a enumeração anônima dos recursos, e se a enumeração anônima estiver indisponível, o proxy do usuário do portal será usado para consultar.
- Verificação dos nós ativos da grid por varredura TCP aos nós listados na etapa anterior a portas TCP de serviços do Globus, ou seja, um nó é dito ativo se for possível estabelecer uma conexão TCP entre o portal e serviços do Globus neste nó.

4.4.2.2 Enumeração de Portas e Fingerprint do sistema operacional

As informações sobre os nós ativos que foram obtidas pelo módulo anterior são validadas neste módulo a fim de assegurar o envio dos mecanismos de verificação de conformidade apropriados para a arquitetura e sistema operacionais de cada um dos nós.

Novamente, este processo ocorre em duas etapas:

- O aplicativo ***nmap*** é usado para enumerar todas as portas TCP abertas nos nós ativos da grid.
- Um perfil de cada nó ativo é obtido pela submissão de um aplicativo usando uma RSL do GRAM. Estas informações são complementares àquelas já obtidas através do MDS.

4.4.2.3 Verificação de conformidade

De posse das informações dos nós ativos da grid, suas arquiteturas de hardware

e sistemas operacionais, é possível selecionar os controles e componentes de softwares adequados a cada nó.

A verificação de aderência pode ser realizada em qualquer um ou em todos os nós ativos da grid. Esta verificação é realizada por scripts submetidos através do Portal Web da Grid usando as credenciais do usuário do sistema, ou seja, devido a restrições impostas pelo GSI – que impede o mapeamento de um certificado a um usuário superprivilegiado (UID 0 ou root) – esta verificação ocorre sob a óptica limitada de um usuário comum.

4.4.2.4 Relatórios e Geração de alertas

As informações geradas por qualquer um dos módulos pode ser sumarizada em formato de relatório e disponibilizado ao usuário do sistema para posterior análise. Informações sobre vulnerabilidades e inconformidades devem ser fornecidas em forma de alertas aos responsáveis pelo nó ou pela organização a que este pertence.

4.4.3 Controles da NBR ISO/IEC 17799:2005 Verificados pelo SiMPaSeG

Nesta seção estão descritos os controles da NBR ISO/IEC 17799:2005 que podem ser verificados automaticamente por uma ferramenta como o Tamanduá Mirim ou SiMPaSeG. Para cada um desses controles serão apresentadas as estratégias usadas na verificação de aderência do ambiente de grid computacional.

Os 10 macro controles existentes nesta norma são:

1. Política de Segurança;
2. Segurança Organizacional;
3. Classificação e Controle dos Ativos da Informação;
4. Segurança em Pessoas;
5. Segurança Física e do ambiente;
6. Gerenciamento de Operações e Comunicações;
7. Controle de Acesso;

8. Desenvolvimento da Segurança de Sistemas;

9. Gestão da Continuidade do Negócio;

10. Conformidade.

O SiMPaSeG verificará a aderência de um ambiente de grid computacional a alguns dos controles pertencentes aos macros controles 3, 4 e 7:

3. Classificação e Controle dos Ativos da Informação;

4. Segurança em Pessoas;

7. Controle de Acesso;

A verificação automática de aderência realizada pelo SiMPaSeG vai abranger os seguintes controles através do uso de 'bash scripts' submetidos a cada um dos nós da grid a partir do Portal Web do Projeto InteGridade ou diretamente por código Java executado neste Portal:

- 4.4.3.1 - 3.1.1 - Inventário dos ativos;
- 4.4.3.2 - 4.3.2 - Reportando pontos fracos na segurança;
- 4.4.3.3 - 4.3.3 - Reportando mau funcionamento de softwares;
- 4.4.3.4 - 7.2.1 - Registro de usuário;
- 4.4.3.5 - 7.2.2 - Gerência de Privilégios;
- 4.4.3.6 - 7.2.3 - Gerenciamento de Senha dos Usuários;
- 4.4.3.7 - 7.3.1 - Uso de senhas;
- 4.4.3.8 - 7.4.3 - Autenticação para conexão externa do usuário;
- 4.4.3.9 - 7.5.2 - Procedimentos de entrada no sistema;
- 4.4.3.10 - 7.5.3 - Identificação e autenticação do usuário;
- 4.4.3.11 - 7.5.4 - Sistema de gerenciamento de senhas;
- 4.4.3.12 - 7.5.5 - Uso de programas utilitários;
- 4.4.3.13 - 7.5.7 - Desconexão do Terminal por Inatividade;
- 4.4.3.14 - 7.5.8 - Limitação do Tempo de Conexão;
- 4.4.3.15 - 7.7.3 - Sincronização de relógios;

Devido aos dois capítulos da NBR anteriores ao "Capítulo 3 - Política de Segurança": "Capítulo 1 - Escopo" e "Capítulo 2 - Termos e Definições", a correspondência entre a numeração dos controles neste trabalho apresentada e a das seções da NBR que os definem é a simples adição de duas unidades, desta

forma, por exemplo, a definição do controle 773 é encontrada na seção 9.7.3 da NBR.

4.4.3.1 3.1.1 - Inventário dos ativos

Este controle visa assegurar que ocorra uma proteção efetiva dos ativos, e para que isso seja possível, é necessário compilar e manter um inventário dos ativos da organização.

A Norma dita que estes ativos devem ter importância e valor determinados a fim de que possam ser adotados níveis de proteção proporcionais a estes, assim como devem ser acordadas e documentadas sua propriedade e classificação de segurança.

O SiMPaSeG pode, portanto, auxiliar na compilação e no monitoramento por mudanças de um inventário de ativos de uma grid computacional, ao mesmo tempo em que será de pouca de ajuda quanto a determinar valor, importância, propriedade e classificação de segurança.

São exemplos de ativos que podem ser inventariados e monitorados pelo SiMPaSeG:

a) ativos de informação: bancos de dados e arquivos de dados;
b) ativos de software: software aplicativo, software básico, ferramentas de desenvolvimento e utilitários;
c) ativos físicos: equipamento de computador (processadores, memória, monitores, laptops, modems, placas de rede), equipamentos de comunicação (roteadores), mídia magnética (discos rígidos), outros equipamentos técnicos (UPS);
d) serviços: serviços de informática e telecomunicações: estado de serviços de rede como SMTP, POP3, SSH, página web institucional e estado de aplicações instaladas em servidores de aplicação como Apache Tomcat, Oracle 9IAS e outros.

4.4.3.2 4.3.2 - Reportando pontos fracos na segurança

Este controle determina que os usuários de serviços de informação sejam obrigados a anotar e reportar quaisquer pontos fracos observados ou suspeitados, ou ameaças, aos sistemas e serviços. A Norma alerta ainda que os usuários devam ser instruídos a não provarem (testarem) um ponto fraco suspeitado, já que testar falhas pode ser interpretado como uma potencial utilização indevida do sistema.

O SiMPaSeG fornece aos seus usuários, usuários autenticados da grid computacional, através de seu módulo para "Geração de Relatórios e Envio de Alertas", um mecanismo para que seja cumprido este controle ao enviar alertas aos responsáveis pelos recursos computacionais relatando as possíveis brechas de segurança encontradas. O SiMPaSeG ainda cumpre este controle ao não tentar provar uma fraqueza encontrada.

4.4.3.3 4.3.3 - Reportando mau funcionamento de softwares

Este controle visa instruir os usuários a não tentar reparar nenhuma ferramenta de software que se mostre defeituosa, deve somente relatar aos responsáveis para que estes executem os procedimentos adequados.

O SiMPaSeG, ao inventariar os software instalados no controle 3.1.1, pode identificar softwares instalados com versões desatualizadas e recomendar sua atualização. Apesar de necessariamente não apresentar mau funcionamento, é esperado que as novas versões, em especial as de números de revisão ou compilação maiores incorporem correções de falhas ou bugs do software. (esquema de versionamento: major.minor[.revision[.build]]).

4.4.3.4 7.2.1 - Registro de usuário

Este controle recomenda que deva ser criado um procedimento formal de cadastramento e descadastramento de usuários para a concessão de acesso aos sistemas computacionais.

A Norma faz várias recomendações sobre o processo formal de cadastramento de usuários. Dessas recomendações, as que podem ser verificadas pelo SiMPaSeG são as seguintes:

- usar IDs de usuário exclusivas, de modo que os usuários possam ser relacionados com suas ações e responsabilizados por elas;
- confirmar que o nível de acesso concedido é apropriado para os fins do negócio e consistente com a política de segurança da organização; por exemplo, não compromete a segregação de tarefas;
- manter um registro formal de todas as pessoas cadastradas para usar o serviço;
- remover imediatamente os direitos de acesso de usuários que trocaram de função ou deixaram a organização;
- verificar periodicamente e remover IDs de usuários e contas redundantes;
- assegurar que IDs de usuário redundantes não sejam emitidas para outros usuários.

As verificações quanto à inclusão de cláusulas nos contratos de trabalho e

evento por evento, isto é, o requisito mínimo para seu papel funcional apenas quando necessário.

Idealmente dever-se-ia poder verificar a configuração do mecanismo que permite que usuários e aplicações executem tarefas como super-usuário (sudo), porém tal arquivo de configuração não deve poder ser lido por usuários não-privilegiados. Das verificações sobre 'sudo' implementadas por (GONCALVES, 2005), apenas é verificado a permissão sobre o arquivo de configuração, enquanto as demais verificações irão falhar.

As verificações sobre a gerência de permissões, em que se deve alocar privilégios na base da necessidade de uso e na base de evento por evento, respeitando assim o princípio de privilégios mínimos, assim como os procedimentos formais de autorização, registro e revisão periódica dos privilégios concedidos não

O SiMPaSeG poderá verificar se as senhas associadas com cada produto de sistema, por exemplo sistema operacional, sistema de gerenciamento de banco de dados e cada aplicativo estão armazenadas sob forma protegida. Ainda implementa a verificação de solicitação de alteração da senha temporária e de contas de usuário nunca acessadas.

4.4.3.7 7.3.1 - Uso de senhas

Este controle visa instruir os usuários quanto às boas normas de segurança na seleção e uso de senhas, pois estas são o meio mais comum de validar a identidade do usuário e assim estabelecer direitos de acesso aos serviços ou facilidades de processamento de informações.

Os usuários devem ser aconselhados a:

a) manter confidenciais as senhas;

b) nunca anotarem as senhas em papel;

c) alterar senhas sempre que houver qualquer indicação de possível comprometimento da senha ou do sistema;

d) selecionar senhas de qualidade e que:

   1) sejam fáceis de lembrar;

   2) não sejam baseadas em algo que alguém poderia facilmente deduzir e obter usando informações relacionadas com a pessoa;

   3) sejam isentas de caracteres consecutivos idênticos ou grupos totalmente numéricos ou totalmente alfabéticos.

e) alterar senhas a intervalos regulares ou baseado na quantidade de acessos, e evitar reutilizar ou usar ciclicamente senhas antigas;

f) alterar senhas temporárias no primeiro logon ;

g) não incluir senhas em qualquer processo automático de logon;

h) não compartilhar senhas individuais.

Os usuários que precisarem acessar múltiplos serviços ou plataformas devem ser avisados que eles podem manter senhas comuns e de qualidade para serviços e organizações com equivalente e razoável nível de proteção para senhas armazenadas.

As senhas de acesso a dispositivos que podem ser usados para identificação e autenticação, tais como cartões inteligentes (SmartCards) e Tokens devem observar as recomendações deste controle, assim como a chave privada armazenada em disco usada na autenticação para acesso remoto seguro (SSH - Secure Shell) deve ser protegida por senha e o acesso de leitura deve ser proibido aos demais usuários.

O SiMPaSeG implementa a verificação da política de expiração de senhas, do nível de acesso da chave privada usada para autenticação 'ssh' e da obrigatoriedade dos usuários escolherem somente senhas não-triviais ao alterarem suas senhas.

4.4.3.8 7.4.3 - Autenticação para conexão externa do usuário

Este controle busca garantir a proteção de serviços que utilizam redes através da seleção e adoção de mecanismos apropriados para autenticação de usuários e equipamentos. Esta proteção objetiva o controle de acesso a serviços em redes internas e externas.

O acesso de usuários remotos deve estar sujeito à autenticação, pois as conexões externas apresentam um risco de acesso não autorizado às informações do negócio. Para que os métodos de autenticação selecionados sejam apropriados é necessário que antes seja feita uma avaliação de riscos.

Dentre os métodos de autenticação de usuários remotos temos, por exemplo, uma técnica baseada em criptografia, tokens de hardware ou protocolo tipo “challenge/response”. Deve-se avaliar o uso de linhas privadas dedicadas, de callback ou uma funcionalidade para checar endereços de usuário na rede também podem ser usadas para fornecer garantia da origem das conexões.

A implementação do SiMPaSeG para este controle verifica se os mecanismos adotados para autenticação usam técnicas criptográficas apropriadas, para se preservar a confidencialidade das informações e das senhas dos usuários.

4.4.3.9 7.5.2 - Procedimentos de entrada no sistema

Este controle visa proteger o acesso a serviços de informação através de um processo de logon seguro, ou seja, o procedimento para conectar em um sistema de computador deve ser projetado para minimizar a oportunidade de acessos não autorizados. O procedimento de logon deve, portanto, divulgar o mínimo de informações sobre o sistema, de forma a evitar fornecer assistência desnecessária a um usuário não autorizado.

A NBR recomenda que um bom procedimento de logon deva:

a) não exibir identificadores do sistema ou do aplicativo até que o processo de logon tenha sido completado com sucesso;
b) exibir um aviso genérico de que o computador deve ser acessado apenas por usuários autorizados;
c) não fornecer mensagens de ajuda durante o procedimento de logon que poderiam ajudar um usuário não autorizado;
d) validar as informações de logon apenas depois de terminada toda a entrada de dados. Se ocorrer uma condição de erro, o sistema não deve indicar qual parte dos dados está correta ou incorreta;
e) limitar a quantidade permitida de tentativas fracassadas de logon (recomenda-se três) e considerar:
    1) registrar todas as tentativas fracassadas;
    2) forçar um intervalo de tempo antes que tentativas adicionais de logon sejam permitidas ou rejeitar quaisquer tentativas adicionais sem autorização específica;
    3) desconectar as conexões de links de dados;
f) limitar o tempo mínimo e máximo permitidos para o procedimento de logon. Se excedido, o sistema deve encerrar o logon;
g) exibir as seguintes informações na conclusão de um logon bem-sucedido:
    1) data e hora do logon anterior bem-sucedido;
    2) detalhes de quaisquer tentativas de logon fracassadas desde o último logon bem-sucedido.

O SiMPaSeG implementa a verificação dessas recomendações em cada um dos nós da grid computacional.

4.4.3.10 7.5.3 - IDENTIFICAÇÃO E AUTENTICAÇÃO DO USUÁRIO

Este controle visa garantir a responsabilização dos usuários através de identificadores exclusivos de usuários e mecanismos seguros para identificação e autenticação (I&A).

As recomendações da NBR são para que todos os usuários, incluindo equipes de suporte, administradores de rede, programadores e administradores de BD, tenham identificadores exclusivos (ID de usuários) de modo que as atividades possam ser rastreadas até o responsável individual. Poderá existir somente em circunstâncias excepcionais, o uso de uma ID de usuário compartilhada para um grupo de usuários ou um serviço específico, devendo para isso, existir um benefício claro para o negócio e a aprovação documentada da gerência.

Estas IDs não devem dar nenhuma indicação do nível de privilégio do usuário e, se necessário, devem-se adotar controles adicionais para manter a responsabilização.

Dentre os vários processos de autenticação que podem ser usados para substanciar a identidade alegada de um usuário destacam-se :

- as senhas, um modo muito usual de fornecer identificação e autenticação (I&A) baseado em um segredo que apenas o usuário conhece
- os conseguidos através de meios criptográficos e protocolos de autenticação, como tokens de memória ou smartcards que os usuários portem consigo
- as tecnologias de autenticação biométrica, que usam as características únicas ou atributos de um indivíduo.

A combinação de tecnologias e mecanismos associados de forma segura resultará em autenticação mais poderosa.

O SiMPaSeG verificará se os usuários possuem identificadores exclusivos, e se estão seguros os mecanismos de autenticação conseguidos por meios criptográficos e protocolos de autenticação.

4.4.3.11 7.5.4 - Sistema de gerenciamento de senhas

Neste controle, a NBR recomenda o uso de sistemas de gerenciamento de senhas a fim de prover uma funcionalidade eficaz e interativa que assegure senhas de qualidade.

Na maioria dos casos as senhas são selecionadas e alteradas pelos usuários, mas algumas aplicações podem exigir que senhas sejam atribuídas por uma autoridade independente.

Segundo a Norma, um bom sistema de gerenciamento de senhas deve:

a) obrigar o uso de senhas individuais para manter a responsabilização;
b) onde apropriado, permitir aos usuários selecionar e alterar suas próprias senhas e incluir um procedimento de confirmação para permitir corrigir erros de digitação;
c) obrigar a escolha de senhas de qualidade, como descrito no item 9.3.1;
d) onde usuários alteram suas próprias senhas, obrigar alterações de senha como descrito no item 9.3.1;
e) onde os usuários selecionam as senhas, forçá-los a alterar senhas temporárias no primeiro logon (ver 9.2.3);
f) manter um registro de senhas anteriores dos usuários, por exemplo pelos 12 meses anteriores, e impedir a reutilização;
g) não exibir senhas na tela quando estiverem sendo digitadas;
h) armazenar arquivos de senhas separadamente dos dados das aplicações do sistema;
i) armazenar senhas sob forma criptografada usando um algoritmo de criptografia one-way;
j) alterar as senhas default dos fornecedores em seguida à instalação dos softwares.

O SiMPaSeG fará diversas verificações sobre o sistema de gerenciamento de senhas, tais como mascaramento das senhas ao digitar e política de expiração de senhas, e se as senhas default dos fornecedores de hardware e software foram alteradas.

4.4.3.12 7.5.5 - Uso de programas utilitários

Este controle visa impedir o mau uso de utilitários e aplicativos contra a integridade do sistema. Para isso, deve-se restringir e controlar o uso dos programas utilitários de sistema presentes na maioria das instalações e que podem ser capazes de sobrepujar controles dos sistemas e aplicativos.

Segundo a NBR devem ser adotados os seguintes controles:

a) uso de procedimentos de autenticação para utilitários de sistema;
b) segregar os utilitários dos softwares aplicativos;
c) limitação do uso de utilitários de sistema à quantidade mínima praticável de usuários autorizados e confiáveis;
d) autorização para uso ad hoc de utilitários de sistema;
e) limitação da disponibilidade dos utilitários de sistema, por exemplo pela duração de uma modificação autorizada;
f) registro em log de todo uso de utilitários de sistema;
g) definir e documentar níveis de autorização para utilitários de sistema;
h) remover todos os softwares utilitários e softwares de sistema que sejam desnecessários.

O SiMPaSeG verificará todos os softwares instalados, assim como o controle de acesso a esses com atenção especial com a verificação dos mecanismos que podem ser utilizados para sobrepujar os controles de autorização.

4.4.3.13 7.5.7 - Desconexão do Terminal por Inatividade

Este controle evita que usuários não autorizados venham a utilizar um terminal deixado inativo por um usuário autenticado. Para isso, os terminais inativos devem ser desligados (shut-down) e/ou a sessão do usuário deve ser finalizada após um período definido de inatividade.

O SiMPaSeG verificará o uso de uma política de time-out para as sessões de usuários para diferentes tipos de acesso.

4.4.3.14 7.5.8 - Limitação do Tempo de Conexão

Segundo a NBR, as aplicações de alto risco devem possuir restrições quanto os tempos de conexão, pois limitar o período durante o qual são permitidas conexões de terminal a serviços informatizados reduz a janela de oportunidade para acesso não autorizado. Tal controle deve ser considerado para aplicações sensíveis de computador, especialmente aquelas com terminais instalados em locais de alto risco, tais como áreas públicas ou externas que estão fora do gerenciamento de segurança da organização

O SiMPaSeG verificará a implementação deste controle:

a) o uso de slots de tempo predeterminados, por exemplo para transmissões de arquivo em batch ou sessões interativas normais de curta duração;
b) e a restrição dos horários de conexão às horas normais de expediente se não houver necessidade de horas extras ou operação em horário estendido.

4.4.3.15 7.7.3 - Sincronização de relógios

A sincronização de relógios dos computadores e equipamentos de rede é fundamental para assegurar a exatidão dos logs para auditoria, que podem ser exigidos para investigações ou como prova em casos legais ou disciplinares.

Computadores e dispositivos de comunicação que possuam relógio tempo-real devem ser ajustados para um padrão acordado, por exemplo Tempo Universal Coordenado (UTC) ou tempo padrão local. Deve-se ainda garantir um procedimento para verificar e corrigir qualquer variação significativa, como por exemplo, o uso do Network Time Protocol (NTP) para sincronização com servidores de tempo baseados em relógios atômicos como os operados por redes de satélites e observatórios astronômicos.

O SiMPaSeG verificará a exatidão dos relógios, a configuração e funcionamento dos mecanismos de verificação e correção dos relógios.

5 IMPLEMENTAÇÃO

Neste capítulo é apresentado o processo de desenvolvimento, instalação e utilização do Sistema de Monitoramento Passivo de Segurança de Grid Computacional (SiMPaSeG), que visa analisar o nível de aderência de um ambiente de grid computacional à Norma Nacional de Segurança da Informação NBR ISO/IEC 17799:2005 sob a perspectiva de um usuário não-privilegiado. A implementação é parte do portal de submissão do Projeto InteGridade.

5.1 Objetivo

O objetivo do SiMPaSeG é o de permitir a verificação automática da aderência de um ambiente de grid computacional a 15 controles da NBR ISO/IEC 17799:2005. Os requisitos e a estratégia para verificação, assim como as limitações da implementação, foram descritas para cada um desses controles no capítulo 4.

O modelo de verificação implementado é uma extensão do apresentado por (GONÇALVES, 2005) com as seguintes alterações:

- Foi expandido o modelo de banco de dados para suportar a complexidade da grid computacional
- Foram implementados novos controles: “3.1.1 - Inventário dos ativos”, “4.3.3 - Reportando mau funcionamento de softwares”, “7.3.1 - Uso de senhas” e “7.7.3 - Sincronização de relógios”.
- Demais controles foram expandidos para suportar verificação da grid computacional
- Conceito de ferramenta administrativa foi alterado sob vários aspectos:
    - a verificação da aderência passa a ser executada com os privilégios de usuário autenticado na grid, que obrigatoriamente não pode possuir privilégios de administrador nos nós

- a análise dispensa a instalação prévia de componentes de software
- a análise pode ser iniciada por qualquer usuário da grid, e as máquinas analisadas podem estar em diferentes instituições
- os relatórios estão disponíveis a qualquer usuários da grid

O SiMPaSeg foi inteiramente construído usando software livre, e sua implementação seguiu conceitos de engenharia de software usando processo iterativo com análise e projeto orientados a objetos usando UML.

O ambiente utilizado no desenvolvimento foi constituído do sistema operacional RedHat Linux, das linguagens de programação Java e Shell script, das API's Java J2SE e Java EE, do ambiente integrado de desenvolvimento Eclipse IDE, da ferramenta de construção Apache Ant, do servidor de aplicações Apache Tomcat, do servidor de banco de dados relacional puramente Java HSQLDB, do software de modelagem UML Poseidon for UML Community Edition e do software para modelagem visual de bancos de dados fabForce DBDesigner.

Baseado nas informações do capítulo anterior e de entrevistas com os usuários do portal foram realizadas as etapas de análise e projeto do sistema. Nestas etapas foram gerados vários artefatos, como documento de visão, documentos de casos de uso, diagrama de casos de uso, modelo conceitual e diagrama de classes.

Durante a fase de construção foram detalhados os casos de uso e diagrama de classe, ao mesmo tempo em que o desenvolvimento dos códigos foi apoiado pelo sistema de controle de versão Concurrent Versions System (CVS) e do sistema de gerenciamento de desenvolvimento de softwares Bugzilla.

O desenvolvimento do sistema permitiu validar o funcionamento do modelo adaptado para grid e identificar várias dificuldades na análise automática do ambiente e identificar algumas melhorias que são descritas no capítulo de conclusão.

5.2 Plataforma de desenvolvimento

A plataforma de desenvolvimento é formada por sistema operacional,

compiladores, ambiente de desenvolvimento, ferramentas de apoio e gerenciamento de desenvolvimento, de banco de dados e de ambiente para execução e testes a seguir descritos.

A escolha dessa plataforma foi orientada a softwares livres com o objetivo de garantir a portabilidade, de modo que a solução criada seja independente de plataforma de hardware e software.

Consulte o “Apêndice B – Plataforma de Desenvolvimento do SiMPaSeG” para mais informações.

5.3 Metodologia de desenvolvimento do SiMPaSeG

A metodologia usada no desenvolvimento do SiMPaSeG tem sua origem no Rational Unified Process (RUP). O RUP é um processo iterativo de desenvolvimento de software criado pela Rational Software Corporation, agora subsidiária da IBM.

O RUP usa a abordagem da orientação a objetos em sua concepção, e é

HU–GVGúUúProj2–U7H7VúUHx—7–Pa2GVPxG978xPl2T98OAOKÃc2TAUHx—7–Pe2T8UG99H9PpT2TH9HU8

O propósito deste documento é coletar, analisar e definir necessidades e características de alto nível do SiMPaSeG – Sistema de Monitoramento Passivo de Segurança de Grid Computacional.

O documento foca nos recursos necessários aos administradores e porque estas necessidades existem. O detalhamento de como o SiMPaSeG atenderá estas necessidades serão descritos nos Casos de Uso e em documentos suplementares.

5.3.1.2 Posicionamento

O Projeto SiMPaSeG (Sistema de Monitoramento Passivo de Segurança de Grid Computacional) tem como objetivo a especificação, desenvolvimento e implantação de ferramentas para verificação de conformidade dos mecanismos que implementam a política de segurança da grid computacional. O direcionamento para esse projeto é proveniente da grande necessidade de preparar e monitorar a grid computacional na transição para a fase operacional.

5.3.1.3 Colocação do Problema

- problema da ...
  - Heterogeneidade de arquitetura e sistemas operacionais da grid computacional
  - Instalação não-padronizada dos sistemas operacionais
  - Complexidade da infra-estrutura da grid
  - Configuração sem requisito mínimo de segurança
- causa o impacto ...
  - Torna difícil a verificação não-automatizada por um único especialista
  - Permite que pacotes sejam instalados, configurados e iniciados sem que sejam realmente necessários
  - Grande quantidade de pacotes e grande número de passos para a instalação do Globus torna o processo sujeito a falhas
  - Segurança é a última preocupação do administrador, pois não há cultura de

segurança nessas organizações

- uma solução de sucesso seria...
  - Como a heterogeneidade é intrínseca à grid, a construção do sistema tentaria mapear os requisitos da política de segurança e de boas práticas de administração do sistema operacional do nó, de modo a validar a configuração e integridade do nó.
  - sistema identificaria pacotes instalados, serviços em execução e sua correta configuração.
  - sistema validaria se a configuração da infra-estrutura da grid (Globus e suas dependências) está correta
  - sistema buscaria por falhas de configuração do sistema operacional

5.3.1.4 Ambiente

As ferramentas de acesso e submissão de jobs hoje usadas na grid utilizam um portal Web que utiliza tecnologia de software livre, baseada em Java e na API J2EE da Sun Microsystems.

5.3.1.5 Usuários

Para que o processo de modelagem de requisitos do sistema represente as reais necessidades dos usuários, é necessário que cada diferente organização que compõe a grid computacional tenha um representante definido participando do processo de levantamento de requisitos do sistema.

A tabela a seguir apresenta estes representantes e suas responsabilidades.

Organização	**Representante(s)**
LNCC	BrunoSchulze JauvaneOliveira
UFF	Vinod
Puc-Rio	

5.3.1.6 Visão Geral do SiMPaSeG

Esta seção apresenta uma visão de alto nível das funcionalidades, premissas e dependências e alguns requisitos não funcionais do SiMPaSeG.

5.3.1.6.1 Módulos

SiMPaSeG será implementado em 5 módulos:

- Varredura da rede
- Enumeração de portas
- Verificação de conformidade
- Envio de alertas administrativos
- Geração de Relatórios de conformidade

5.3.1.6.2 Premissas e Dependências

São premissas e dependências para implementação e implantação do SiMPaSeG:

- Acesso aos recursos computacionais: grid computacional, estação para desenvolvimento e licenças de software
- Execução não compartimentalizada (jailed) nos nós da grid
- Integração com o sistema do Portal de Submissão

5.3.1.6.3 Requisitos Não Funcionais

- Sun J2EE Servlet container (Apache Jakarta Tomcat, Oracle 9iAS, etc)
- Sistema operacional *nix
- Grid computacional com infra-estrutura Globus 2.4, 3.2 ou 4.0

- Rede de computadores rodando protocolo TCP/IP
- Java 2 SE SDK versão 1.4.x e Java 2 EE SDK versão 1.3.X ou mais recentes;
- Navegadores Web Internet Explorer 6.x e Mozilla Firefox 1.x ou mais recentes;

5.3.1.6.4 Lista de Funcionalidades

ID	Módulo	Funcionalidades
1.	Varredura da rede	Obter listagem dos nós ativos da grid
2.	Enumerar Portas	Para cada nó realizar a enumeração das portas e serviços a ela associados
3. 4.	Verificar Conformidade	Verificar cada nó quanto à conformidade com a política de segurança Verificar quanto aos requisitos de segurança do Globus
5.	Enviar Alertas Administrativos	Gerar e enviar mensagens de alerta aos administradores responsáveis pelo nó e pela rede em que ele se encontra
6.	Gerar Relatórios de Conformidade	Gerar relatórios de conformidade

A FIG. 5.1 mostra os casos de uso essenciais do SiMPaSeG, atores e sistemas externos.

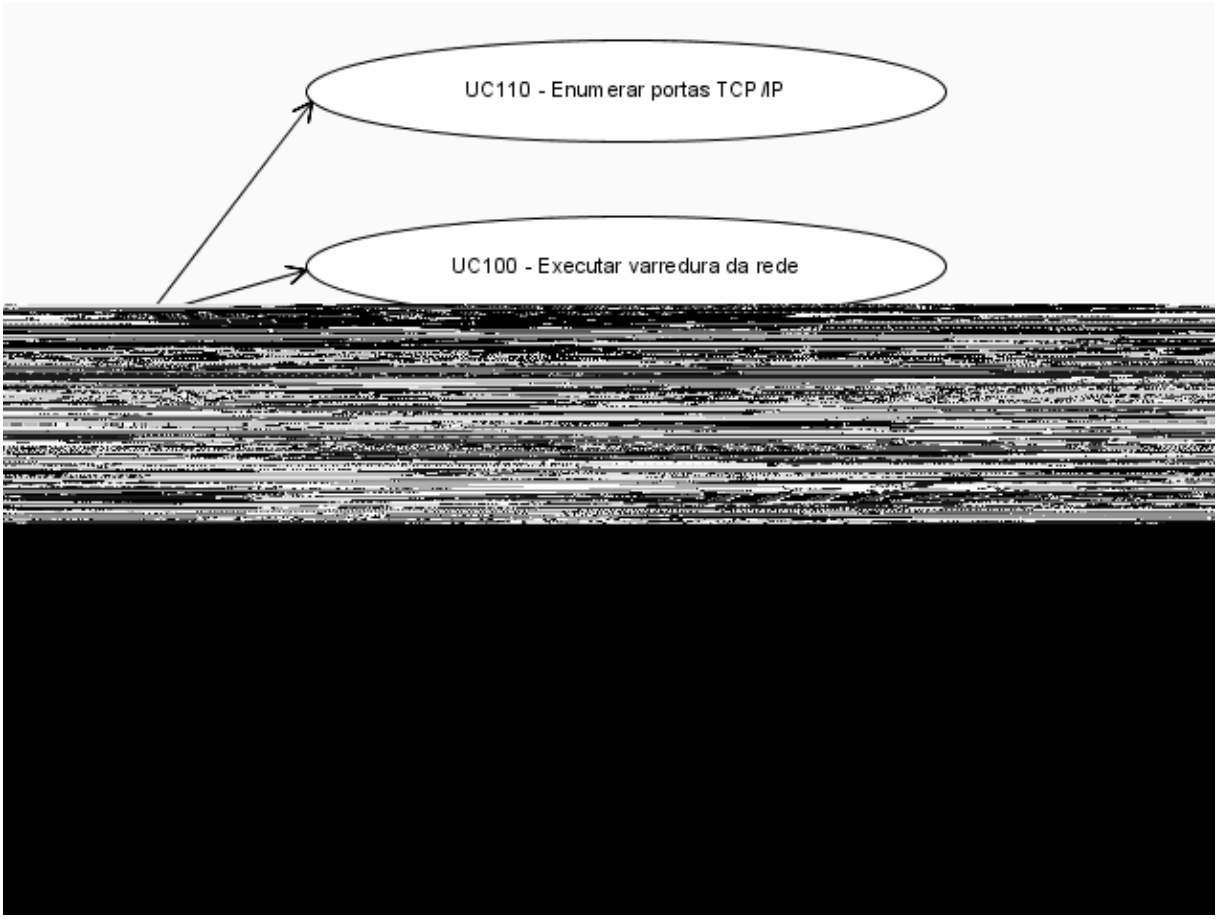

FIG. 5.1 Diagrama de Casos de Uso Essenciais do SiMPaSeG

5.3.2 Elaboração do SiMPaSeG

Nesta fase foram atacados os principais riscos identificados através de prototipação no Portal da grid. Os casos de uso foram detalhados e o modelo conceitual estático foi elaborado. Ao final desta fase, os riscos de arquitetura foram minimizados.

A FIG.5.2 mostra o modelo conceitual. Este modelo representa os conceitos identificados e seus relacionamentos. Durante a fase de construção, alguns destes conceitos serão mapeados em classes de objetos.

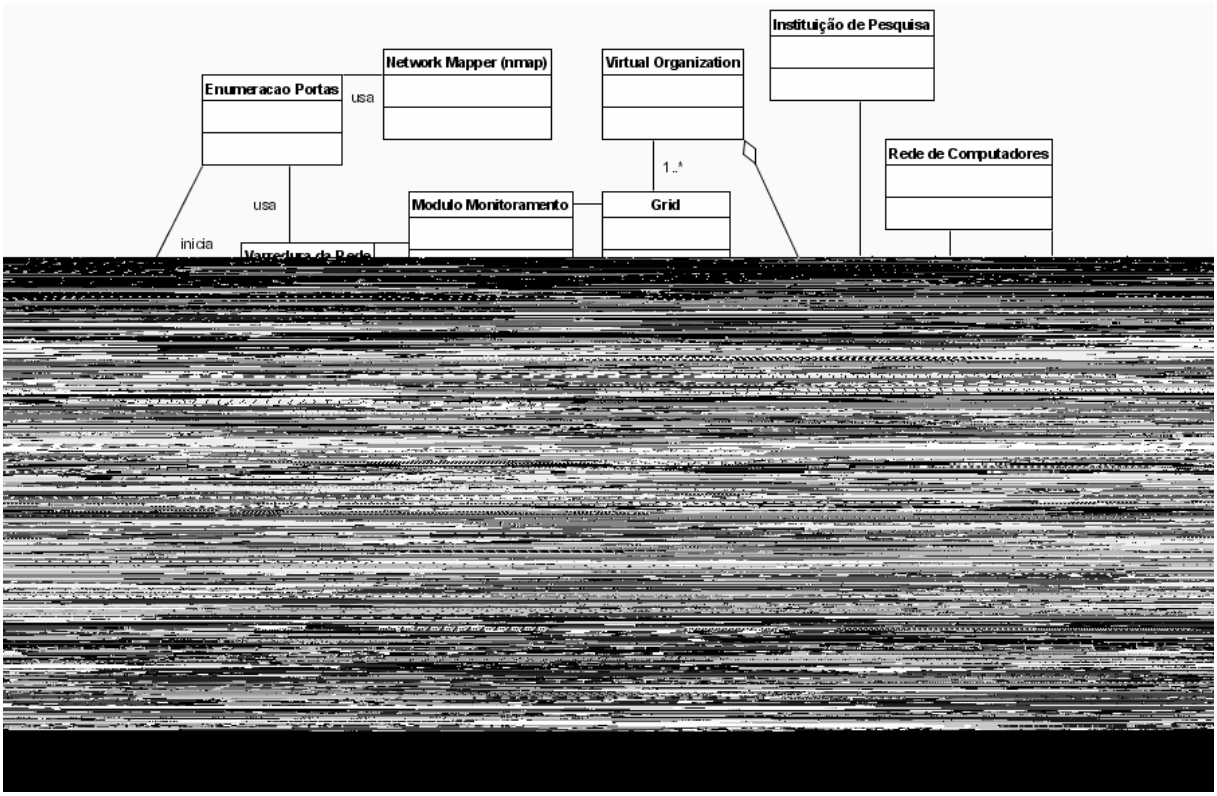

FIG. 5.2 Modelo Conceitual do SiMPaSeG

5.3.2.1 Caso de Uso 100 – Varredura da Rede

5.3.2.1.1 Descrição Sucinta

Este caso de uso descreve como são realizadas as varreduras da rede por nós ativos da grade.

5.3.2.1.2 Pré-condições

- Usuário autenticado pelo portal e com proxy válido
- Usuário com permissão de administrador do portal

5.3.2.1.3 Fluxo Básico

1. O ator seleciona o menu "Monitoramento" no portal
2. Em resposta, o sistema exibe as opções desse menu, entre elas, a opção Varredura da Rede
3. O ator comanda a busca pelos nós ativos
4. O sistema verifica entre os nós registrados na grade, a conectividade TCP para uma relação de portas-padrão do Globus, e baseado nisso, determina os nós ativos e exibe sua lista
5. O ator encerra o caso de uso

5.3.2.1.4 Subfluxo

Não há.

5.3.2.1.5 Fluxo Alternativo

(A-01)Ator seleciona e visualiza os nós ativos:

1. Ator seleciona os nós ativos de interesse e então pode:

- Visualizar os dados no nó.
- Comandar o caso de uso Enumerar Portas - Sistema apresenta as portas TCP abertas nos nós selecionados

(A-02)Impressão

- Usar as funcionalidades do navegador InterG7úPo2T8UG97G8GVP 2THG8GVP 2THGHU

- Lista de nós ativos na grade

5.3.2.2 Caso de Uso 110 – Enumerar Portas

5.3.2.2.1 Descrição Sucinta

Este caso de uso descreve como são realizadas as enumerações de portas nos nós ativos da grade.

5.3.2.2.2 Pré-condições

- Usuário autenticado pelo portal e com proxy válido
- Usuário com permissão de administrador do portal

5.3.2.2.3 Fluxo Básico

1. O ator seleciona o menu "Monitoramento" no portal
2. Em resposta, o sistema exibe as opções desse menu, entre elas, a opção Enumeração de Portas
3. O ator comanda a enumeração de portas
4. O sistema usa o Caso de Uso 100 Varredura da Rede para obter a lista dos nós ativos da rede
5. O sistema aciona o utilitário NMAP (Network mapper) sobre os nós ativos e exibe o resultado do software
6. O ator encerra o caso de uso

5.3.2.2.4 Subfluxo

Não há.

5.3.2.2.5 Fluxo Alternativo

(A-01)Ator seleciona e visualiza o resultado da enumeração:

1. Ator seleciona os nós ativos de interesse e então pode:

- Visualizar os dados no nó.
- Comandar o caso de uso Verificar Conformidade - Sistema apresenta o resultado da análise de conformidade

(A-02)Impressão

- Usar as funcionalidades do navegador Internet, a qualquer momento, para imprimir as informações exibidas pelo sistema

5.3.2.2.6 Pós-condições

- Lista de nós ativos e suas portas TCP abertas

5.3.2.3 Caso de Uso 120 – Verificar Conformidade

5.3.2.3.1 Descrição Sucinta

Este caso de uso descreve como são realizadas as verificações de conformidade nos nós ativos da grade em relação à política de segurança e aos requisitos de segurança do Globus.

5.3.2.3.2 Pré-condições

- Usuário autenticado pelo portal e com proxy válido
- Usuário com permissão de administrador do portal

5.3.2.3.3 Fluxo Básico

1. O ator seleciona o menu "Monitoramento" no portal
2. Em resposta, o sistema exibe as opções desse menu, entre elas, a opção Verificar Conformidade
3. O ator comanda a verificação de conformidade
4. O sistema usa o Caso de Uso 100 Varredura da Rede para obter a lista dos nós ativos da rede (A-01)
5. O ator seleciona os controles/verificações que deseja executar sobre os nós da grade
6. O sistema envia, usando o Globus, os scripts e programas aos nós ativos
7. O sistema analisa o resultado dos scripts (A-02)
8. O ator encerra o caso de uso

5.3.2.3.4 Subfluxo

Não há.

5.3.2.3.5 Fluxo Alternativo

(A-01)Ator seleciona e visualiza o resultado da varredura da rede:

1. Ator seleciona os nós ativos de interesse e então pode:

- Visualizar os dados no nós.
- Comandar o caso de uso Enumerar Portas - Sistema apresenta as portas

TCP abertas nos nós selecionados

- Comandar o caso de uso Verificar Conformidade - Sistema apresenta o resultado da análise de conformidade

(A-02)O sistema comanda-se o UC130 - Enviar Alertas Administrativos

- Os administradores de rede e de host são alertados sobre eventuais problemas de segurança

5.3.2.3.6 Pós-condições

- Dados da verificação coletados e analisados.
- Administradores alertados sobre eventuais problemas de segurança.

5.3.2.4 Caso de Uso 130 – Enviar Alertas Administrativos

5.3.2.4.1 Descrição Sucinta

Este caso de uso descreve como são enviados os alertas administrativos aos administradores de rede e do nó analisado.

5.3.2.4.2 Pré-condições

Dados da verificação de conformidade coletados e analisados

5.3.2.4.3 Fluxo Básico

1. O UC120 envia a identificação da última verificação de conformidade analisada
2. O sistema decide, baseado em regras estabelecidas pelo administrador, se a não-conformidade desta verificação de aderência deve ser informada ou ignorada.

5.3.2.4.4 Subfluxo

Não há.

5.3.2.4.5 Fluxo Alternativo

Não há.

5.3.2.4.6 Pós-condições

- Mensagens de alerta enviados aos Administradores de rede e do nó.

5.3.2.5 Caso de Uso 140 – Gerar Relatórios de Conformidade

5.3.2.5.1 Descrição Sucinta:

Este caso de uso descreve como são regados os relatórios de conformidade.

5.3.2.5.2 Pré-condições

- Usuário autenticado pelo portal e com proxy válido
- Usuário com permissão de administrador do portal

5.3.2.5.3 Fluxo Básico

1. O ator seleciona o menu "Monitoramento" no portal
2. Em resposta, o sistema exibe as opções desse menu, entre elas, a opção “Gerar Relatórios de Conformidade”
3. O ator comanda a geração de relatórios
4. O sistema exibe as opções de relatórios geral, por controle ou por nós
5. O ator seleciona o relatório desejado
6. O sistema exibe o relatório (A-1)
7. O ator encerra o caso de uso

5.3.2.5.4 Subfluxo

Não há.

5.3.2.5.5 Fluxo Alternativo

(A-01)Ator pode:

- Visualizar o dado em tela ou comandar a sua impressão
- Reiniciar o UC para obter outros relatórios

5.3.2.5.6 Pós-condições

Não há.

5.3.3 Construção do SiMPaSeG

Esta fase é a de maior número de iterações, pois os casos de uso foram desenvolvidos individualmente de forma iterativa: foram individualmente e sequencialmente projetados, construídos e integrados ao Portal para testes.

Ao final de cada iteração foi possível verificar a funcionalidade rodando na grid. Projetar com ênfase em testes é importante nesta fase, pois um projeto difícil de testar está muitas vezes relacionado a alto acoplamento ou más práticas. Em desenvolvimento de sistemas há dois tipos de testes comumente utilizados: os testes unitários que servem para testar cada componente individualmente e os testes funcionais que avaliam a interação entre os componentes ou o comportamento do sistema. Nesta fase foram construídos os testes unitários usando JUnit para as classes Java implementadas.

O processo de obtenção dos códigos-fontes no CVS, compilação, testes e instalação do SiMPaSeG foram facilitados pelo uso do Ant. Entretanto não houve testes funcionais automatizados para o sistema.

5.3.3.1 Modelo de Classes

A FIG. 5.3 mostra como o SiMPaSeG se integra ao Portal da grid que implementa a arquitetura MVC (Model-View-Controller). A classe HelperSeguranca é o controlador projetado para o SiMPaSeG, pois esta classe recebe a delegação da FrontController (Portal) para processar as regras de negócio de segurança. A Dispatcher, também parte do Portal, é o que coordena a apresentação das informações aos usuários da grid.

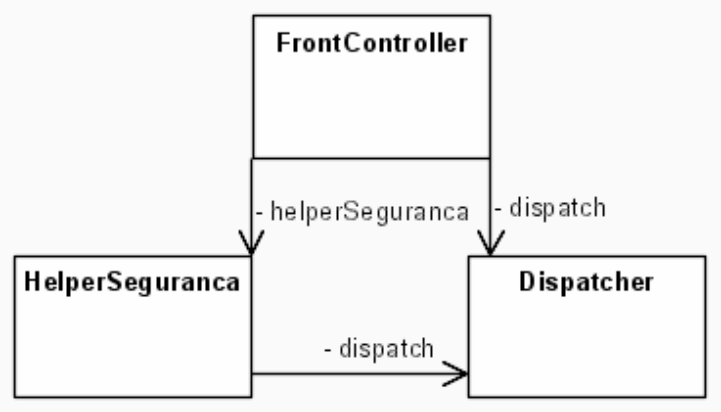

FIG. 5.3 Classe HelperSeguranca e sua integração na arquitetura MVC implementada no Portal da Grade

A FIG. 5.4 mostra o diagrama de classes do pacote de software br.Incc.seguranca. Este pacote contém cinco classes principais detalhadas a seguir:

- HsqldbFacade – componente responsável pela interação com a camada de dados, oferecendo serviços de consulta, inserção, atualização e remoção de dados nas tabelas do sistema;
- NetServices – responsável por serviços de rede como consulta de DNS.
- Controle – classe abstrata responsável pela verificação dos controles de uma norma de segurança;
- Controle311 a Controle773 – classe que especializa controle a fim de implementar o método analisa() que processa os dados da coleta em informações;
- Relatorio – componente responsável pela geração dos relatórios de conformidade.

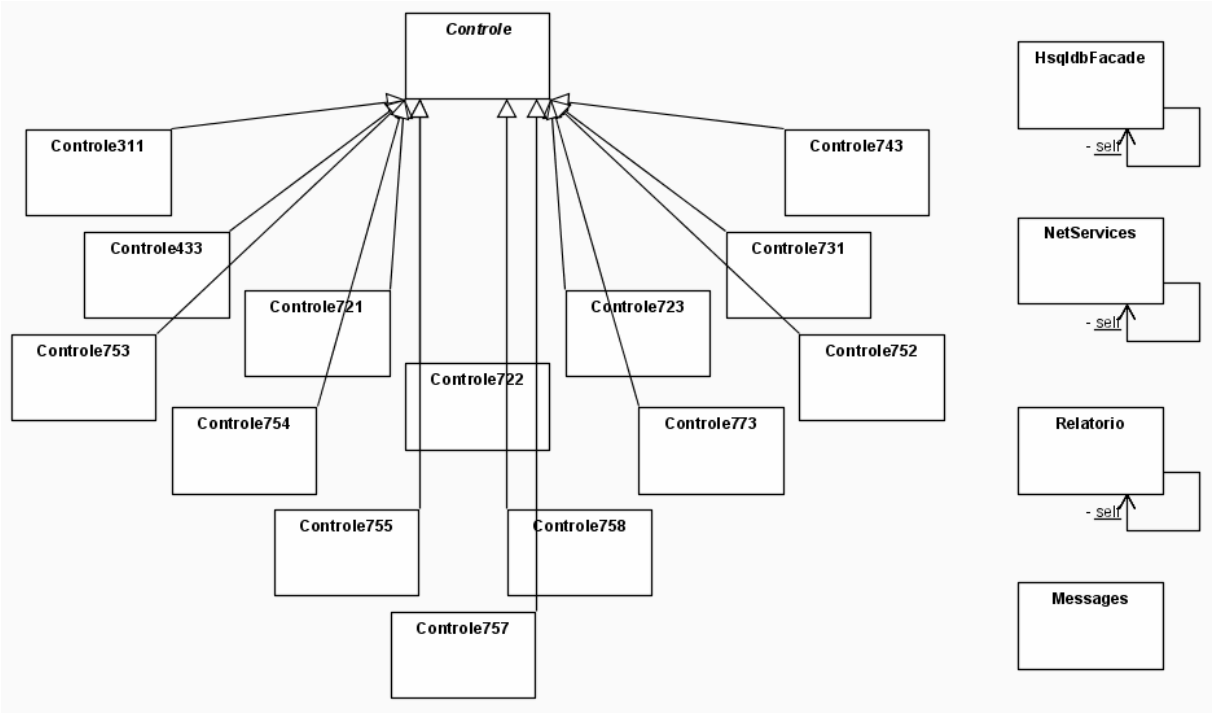

FIG. 5.4 Diagrama de Classes do Pacote br.lncc.seguranca

As FIG. 5.5, 5.6 e 5.7 visam oferecer mais detalhes sobre os atributos e métodos das classes implementadas no SiMPaSeG. Apenas uma classe concreta de verificação de controle é mostrada: a Controle721, as demais especializações da Controle são omitidas.

A FIG. 5.5 mostra a classe HelperSeguranca. Esta classe mantém o estado dos nós sob análise e oferece o acesso às funcionalidades do SiMPaSeG através dos métodos: hostsList() para o UC100, hostsPortEnum() para o UC110, hostsVerify() e analisaControles() para o UC120, relatorioPorControle() e relatorioPorHost() para o UC140.

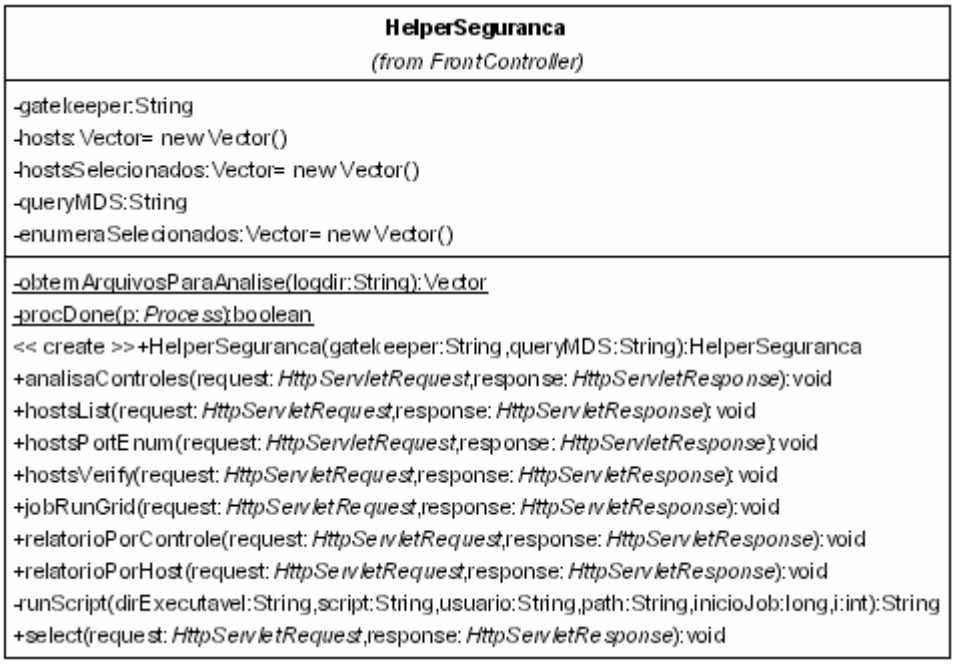

FIG. 5.5 Classe HelperSeguranca – responsável pelo controle do SiMPaSeG

A FIG. 5.6 mostra as classes de serviço HsqldbFacade e NetServices. Ambas as classes implementam o padrão Singleton, restringindo-as a apenas uma única instanciação por aplicação. A HsqldbFacade é responsável pela conexão ao banco de dados e por serviços de consulta através do método query(), inserção, atualização e remoção de registros usando

A FIG. 5.7 mostra as classes de modelo: a classe abstrata Controle, e as classes Controle721 e Relatorio.

A classe Controle é responsável pela análise da verificação de um controle de uma norma. Ela possui todos os atributos e métodos comuns de um Controle, e deixa o método que realiza a análise como abstrato, devendo portanto, ser implementado por cada classe que a estenda. A classe Controle721 e as demais que estendem a Controle implementam a lógica da análise de seu arquivo de coleta de dados nos nós através do método analise().

A classe Relatorio é responsável por gerar os relatórios de conformidade. Eles são gerados exclusivamente pelas informações já armazenadas no banco de dados. Seus métodos são reportByControle() que gera um relatório com as informações da análise agrupadas por cada controle verificado; e reportByHost() que gera um relatório com as informações da análise agrupadas por cada nó da grade.

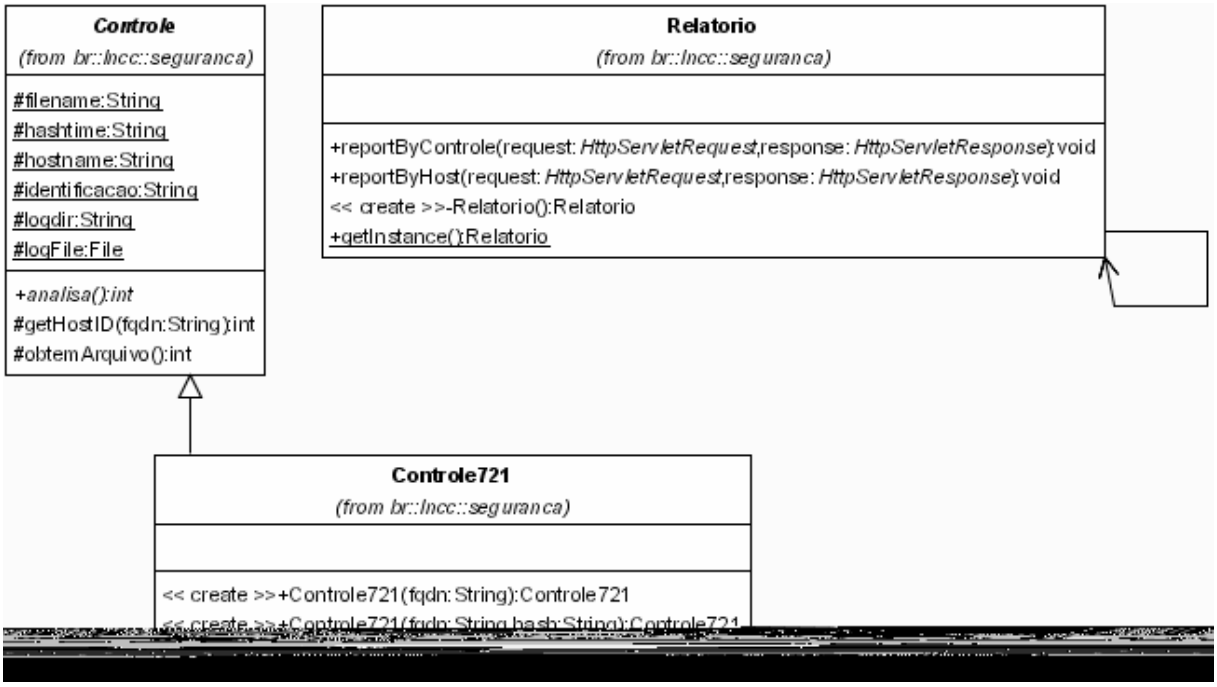

FIG. 5.7 Classes de modelo em detalhe: Controle, Controle721 e Relatorio

5.3.3.2 Modelo de Dados

A FIG. 5.8 mostra o modelo de dados utilizado no Repositório de Dados do

SiMPaSeG. A partir deste modelo, são gerados e mantidos no CVS, os scripts SQL correspondentes para criação de tabelas e carga de dados inicial do repositório para as sintaxes do MySql, SGBD oficial do Tamanduá Mirim, e do HSQLDB, o escolhido para este sistema.

O modelo é dividido em três regiões: SiMPaSeG, Tamanduá e Tamanduá/Administrativo. As duas últimas são as tabelas originais do Tamanduá Mirim resultantes de uma revisão juntamente com seu autor original, e para mais informações sobre estas duas regiões consulte (GONÇALVES, 2005).

A tabela Usuarios da região Tamanduá/Administrativo é a única do modelo que não é utilizada pelo SiMPaSeG, sendo somente usada pelo Tamanduá Mirim para fins de autenticação de usuários.

As tabelas da região Tamanduá são usadas tanto pelo Tamandua Mirim quanto pelo SiMPaSeG e os dados coletados por ambos os sistemas são compatíveis e podem compartilhar uma base de dados comum.

A região SiMPaSeG é composta pelas seguintes tabelas:

- Usuarios_Grade – armazena as informações sobre os usuários da grid computacional. Estes são identificados pelo campo “Assunto” de seu certificado X.509
- Nodo_Usuario – tabela de relacionamento N:N entre as tabelas Tamandua.Hosts e Usuarios_Grade
- Compromized_Keys – armazena os hashes das chaves privadas potencialmente comprometidas. Estas chaves são identificadas durante a verificação de conformidade por estarem armazenadas de forma insegura.
- Organizacao_Virtual – armazena as organizações virtuais identificadas na grid. Uma organização virtual reúne nós que podem estar sob diferentes domínios administrativos, mas que compartilham recursos computacionais de forma coordenada.
- Instituicao – armazena o nome das instituições reais que participam da grid.
- VO_Instituicao – mapeia o relacionamento N:N entre instituições reais e virtuais
- MDS – armazena as informações sobre os diferentes servidores GRIS (Grid Resource Index Service) de uma instituição

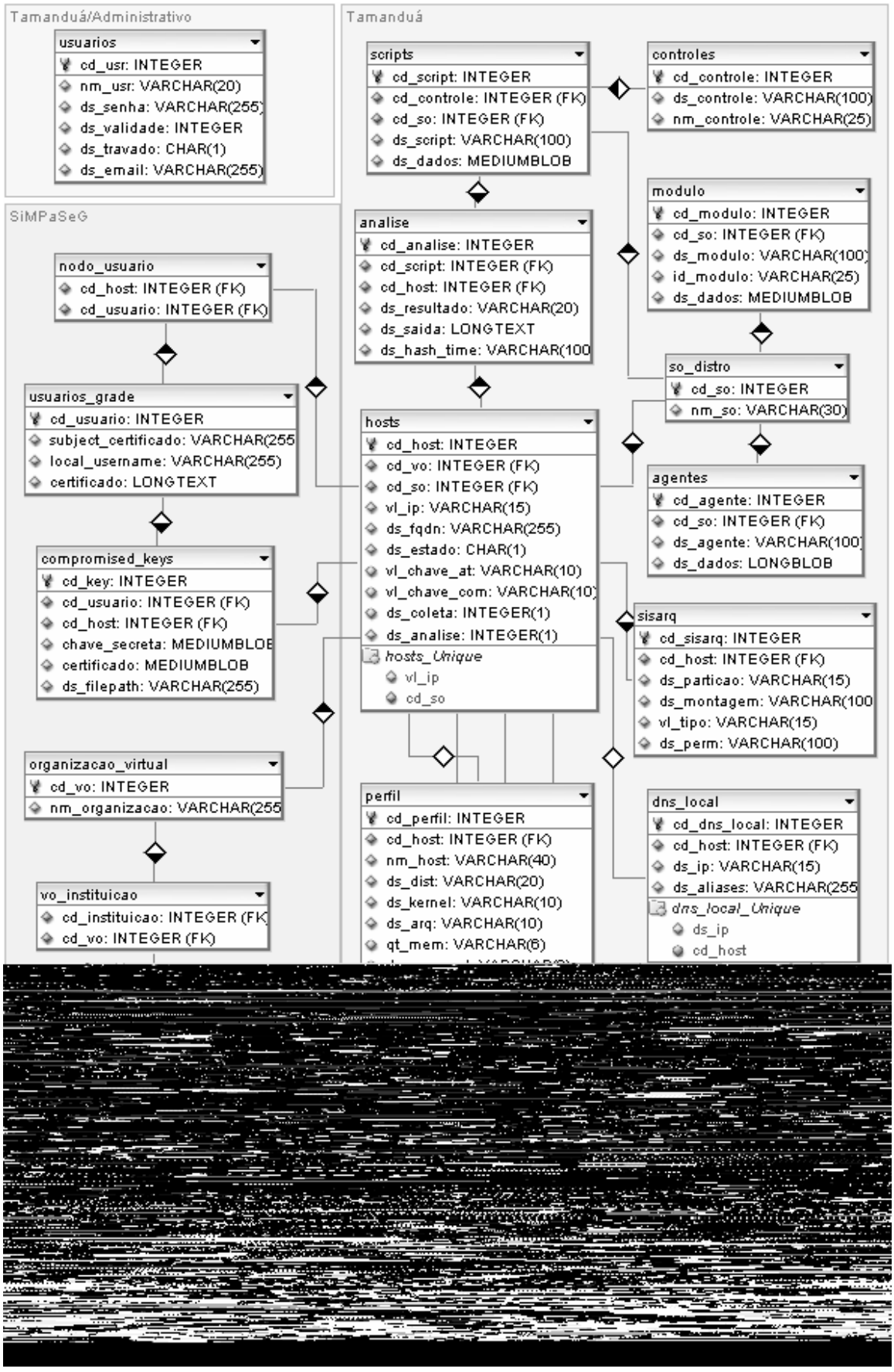

FIG. 5.8 Modelo de Dados do SiMPaSeG

5.3.4 Transição do SiMPaSeG

A fase de transição caracteriza-se pela instalação e testes do sistema no Portal da grid.

A integração do SiMPaSeG ao Portal poderia ter sido mais fácil se todos os desenvolvedores compartilhassem o mesmo repositório CVS; esta limitação está sendo resolvida com o recebimento de recursos computacionais do Projeto Giga e pela padronização de práticas entre os desenvolvedores envolvidos.

5.4 Documentação

A documentação da API das classes do pacote ‘br.lncc.seguranca’, com seus membros e métodos, pode ser gerada automaticamente pelo Ant durante a construção utilizado-se a ferramenta Javadoc.

Os diagramas UML e os demais artefatos de análise e projeto complementam a documentação para fins de expansão e manutenção corretiva do sistema.

5.5 Requisitos para a instalação e uso

Para instalação do SiMPaSeG como parte do Portal da grid deve-se atender aos seguintes requisitos:

- o Apache Ant e as bibliotecas JUnit, Log4J, Java CoG Kit e Java EE devem estar instalados
- os códigos-fontes devem ser compilados com Java 2 SDK 1.4 ou superior
- possuir um servidor de aplicações ou um 'servlet container' compatível com as especificações de Servlet 2.3 e JSP 1.2 ou mais recentes.
- a Java 2 SDK usada para compilar os códigos-fontes não deve ser mais nova que a JRE utilizada no servidor de aplicações
- os arquivos de configuração do SiMPaSeG (*.properties) devem ser editados

de acordo com o ambiente

5.6 Utilização do sistema

O SiMPaSeG é uma contribuição ao Projeto InteGridade - Grid Sinergia, uma grid computacional de desenvolvimento entre as instituições participantes: ComCIDis/LNCC, IC/UFF, LABPAR/PUC-RJ, CAT/CBPF, IC/UNICAMP, conectadas através da Rede Giga e INF/UFRGS. Seus usuários potenciais são os usuários, administradores de host e de rede participantes da grid. Esses usuários, apesar de não terem recebido treinamento durante a fase de transição do RUP, estes terão acesso ao manual online no portal.

As interfaces foram construídas de modo a guiar o usuário para a solução de problemas. A FIG. 5.9 mostra a tela de login no portal.

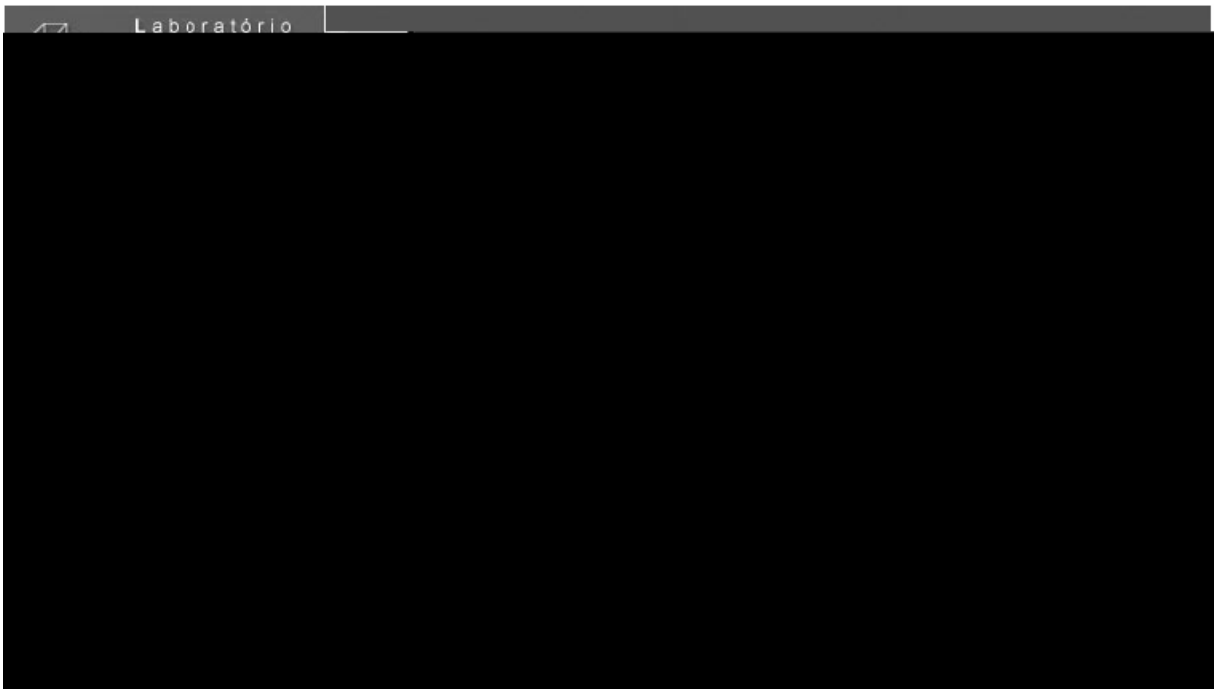

FIG. 5.9 Tela de *Login* do Portal

5.6.1 Acesso ao portal da grid

O acesso ao portal é controlado por um mecanismo de autenticação usando o Java CoG Kit. Um usuário está autenticado, se e somente se, o portal tiver conseguido criar um proxy, que será usado para delegação de credenciais, se utilizando da senha informada pelo usuário para acessar a chave privada associada ao seu certificado.

Na grid, os usuários são identificados através do campo Subject de seus certificados. Este valor é então usado para mapear o usuário da grid para um usuário local ao recurso utilizado. Abaixo, uma entrada no arquivo '/etc/grid-security/grid-mapfile':

"/O=GridBR/OU=Grid Sinergia/OU=lncc.br/CN=Fabio Fagundes" fagundes

5.6.2 Funcionalidades do portal

A FIG. 5.10 - "Tela de Apresentação do Portal" apresenta uma descrição sucinta das funcionalidades implementadas no portal. Este portal é acessível pela URL http://giga01.lncc.br:8080/portal05/jsp/

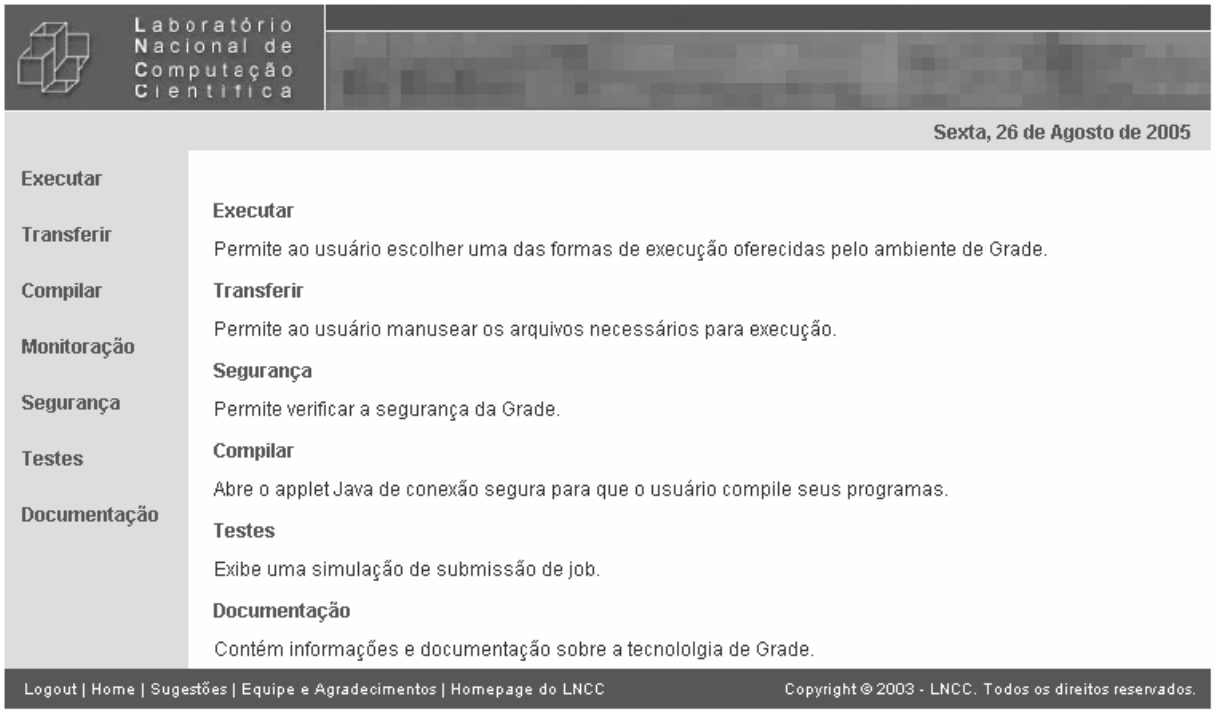

FIG. 5.10 Tela de Apresentação do Portal

- Executar: Permite enviar programas para execução paralela ou seqüencial em um ou mais nós da grid e coletar o resultado através de download pelo portal
- Transferir: Permite transferir arquivos do computador do usuário ao portal, resultados de processamentos anteriores de programas na grade e de arquivos entre nós da grade
- Compilar: Permite a compilação de programas paralelos escritos em diversas linguagens de programação e que utilizem as bibliotecas MPI e MPICH-G2
- Monitoração: Permite acompanhar o nível de utilização de CPU da grid nas instituições participantes, atualmente estão ativas: LNCC, PUC-Rio e UFF. (FIG. 5.11)
- Segurança: Permite acessar as funcionalidades implementadas pelo SiMPaSeG
- Testes: Permite simular a submissão de jobs MPI e MPICH-G2
- Documentação: Permite acesso a uma biblioteca de papers, tutoriais, documentação de software e links sobre o tema "Grid"

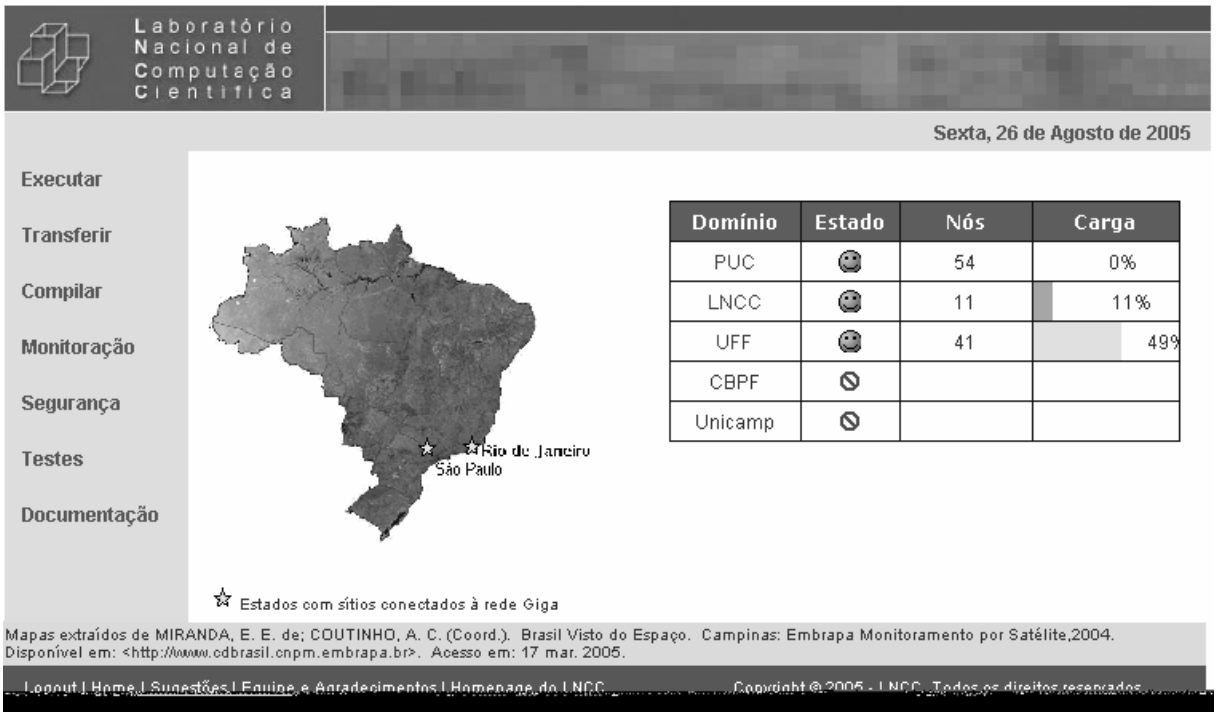

FIG. 5.11 Ferramenta de Monitoramento do estado da *grid*

5.6.3 Apresentação SiMPaSeG

A seguir é mostrado um tour pela implementação atual do SiMPaSeG. As limitações estarão registradas usando a notação BUG seguida de uma numeração seqüencial, do número do caso de uso correspondente e de um comentário.

5.6.3.1 Varredura da rede

O fluxo básico inicia-se como o submenu 'Varredura da rede'. Ao ser clicado, é acionada uma busca no GIS (Grid Information Service) pelos nós ativos da grade. Os nós informados como ativos são verificados quanto à conectividade TCP a uma porta de serviço da grid. A FIG. 5.12 mostra a tela de resposta.

- BUG1 – UC100 - Otimização: A busca MDS deve ser mais rápida, robusta e retornar informações completas
- BUG2 – UC100 – Desejável: Inserir as informações sobre os Nós e Organizações Virtuais (VO) nas tabelas do BD para rastreamento de usuários e autorizações

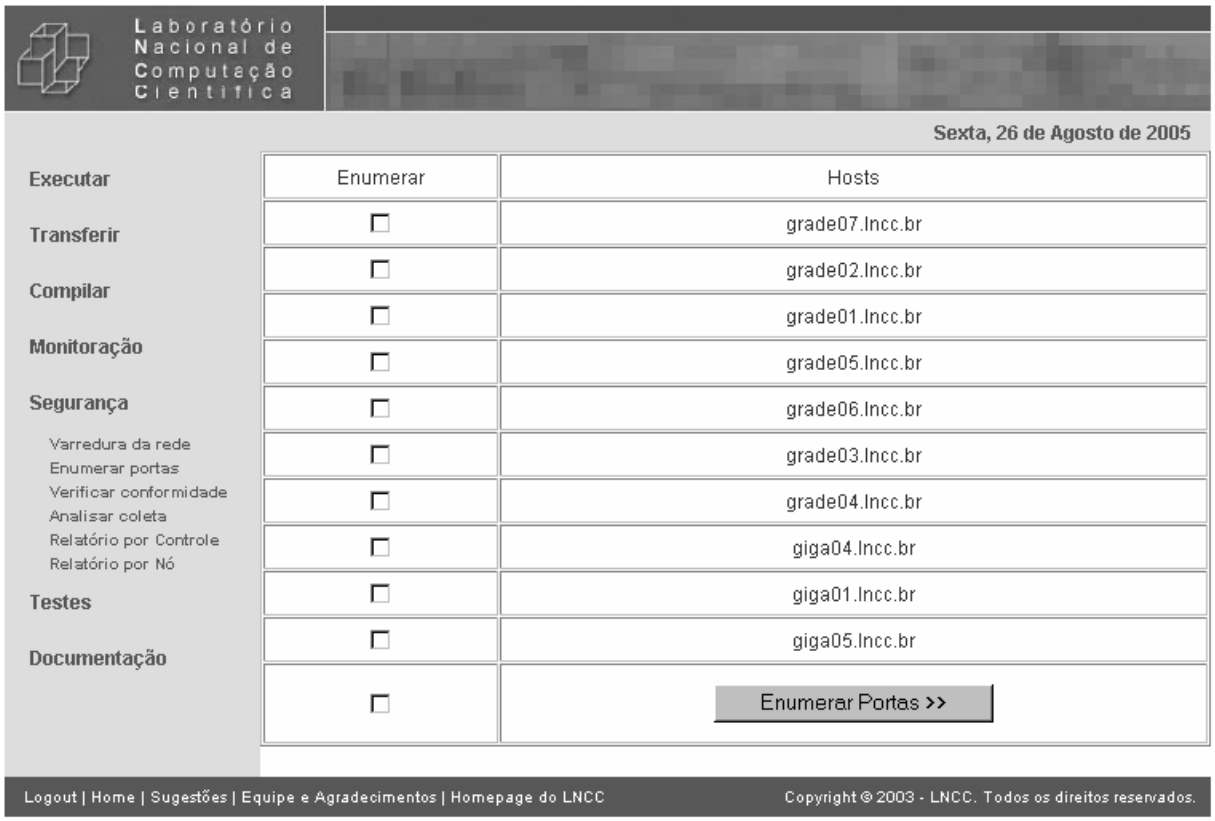

FIG. 5.12 Nós ativos da *grid*

5.6.4 Enumerar portas

Esta funcionalidade inicia o aplicativo nmap

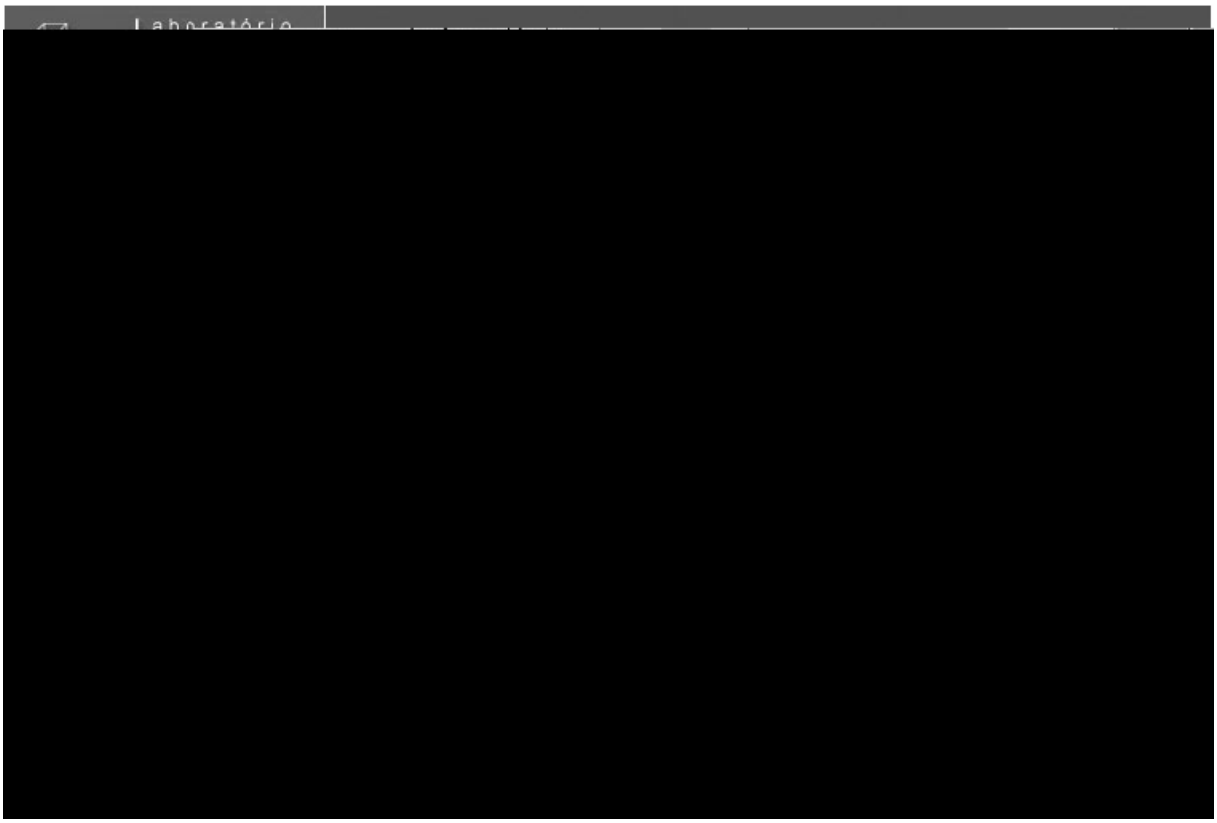

FIG. 5.13 Resultado da Enumeração de Portas

5.6.5 Verificar conformidade

A verificação, como implementada originalmente no Tamanduá Mirim, permitia a escolha dos controles que seriam verificados. Nesta implementação, todos os controles são verificados para um nó selecionado.

- BUG6 – UC110 e UC120 – Projeto: Enumerar portas e Verificar conformidade não deve depender de serem executadas em seqüência. Devem-se obter os nós ativos e executar a operação sobre todos os nós de forma automática. Eventualmente deve-se considerar a solicitação de confirmação do usuário ou mesmo remover essas opções do Menu Segurança
- BUG7 – UC120 – O sistema não acompanha a execução dos Scripts de coleta e não sabe quando o trabalho está feito.
- BUG8 – UC120 – A verificação de conformidade deve ser assíncrona, realizar a análise e enviar os alertas administrativos de modo automático
- BUG9 – UC120 – Remover submenu 'Analisar Coleta' - depende BUG8

- BUG10 – UC120 – Permitir a seleção dos controles que serão verificados
- BUG11 – UC110, UC120 e UC140 – Desejável: Incluir seleção de Hosts por Instituição ou VO para segmentar análises
- BUG12 – UC120 – Verificar conformidade deve ter solução de interatividade para informar ao usuário do status da verificação
- BUG13 – UC120 – Otimização: Verificar conformidade deve usar Threads, ser escalonável para ser escalável; necessariamente deve ser assíncrono

5.6.6 Analisar coleta

Esta funcionalidade foi criada por limitação da implementação corrente que não acompanha se a verificação de conformidade já acabou. O sistema verifica um diretório do sistema de arquivos do portal, atualmente /tmp.

- BUG9 – UC120 – Remover submenu 'Analisar Coleta' - depende BUG8
- BUG14 – UC120 – O local de coleta do resultado das verificações de conformidade deve ser configurável pelo usuário. Deve permitir recuperação e análise em qualquer caminho suportado por RSL ou globus-url-copy
- BUG15 – UC120 – Evitar atualizações desnecessárias dos registros no BD ao verificar antes se os dados da coleta se alteraram desde a última execução (assume que 'Analisar Controles' possa processar coletas em andamento)

5.6.7 Relatórios por Controle e por Nó

Exibe relatório em formato HTML das informações coletadas e analisadas que estão armazenadas no BD. FIG. 5.14 e FIG. 5.15 exibem os relatórios sumarizados por Controles e por nós.

- BUG16 – UC140 – Desejável: Facilitar o envio dos relatório por email com suporte para agendamento
- BUG17 – UC140 – Desejável: Suportar criação de gráficos e séries históricas das informações de verificações armazenadas no BD
- BUG18 – UC140 – Desejável: Suportar geração de relatorio usando diferentes

estilos e formatos digitais de arquivo (PDF, CSV, etc)

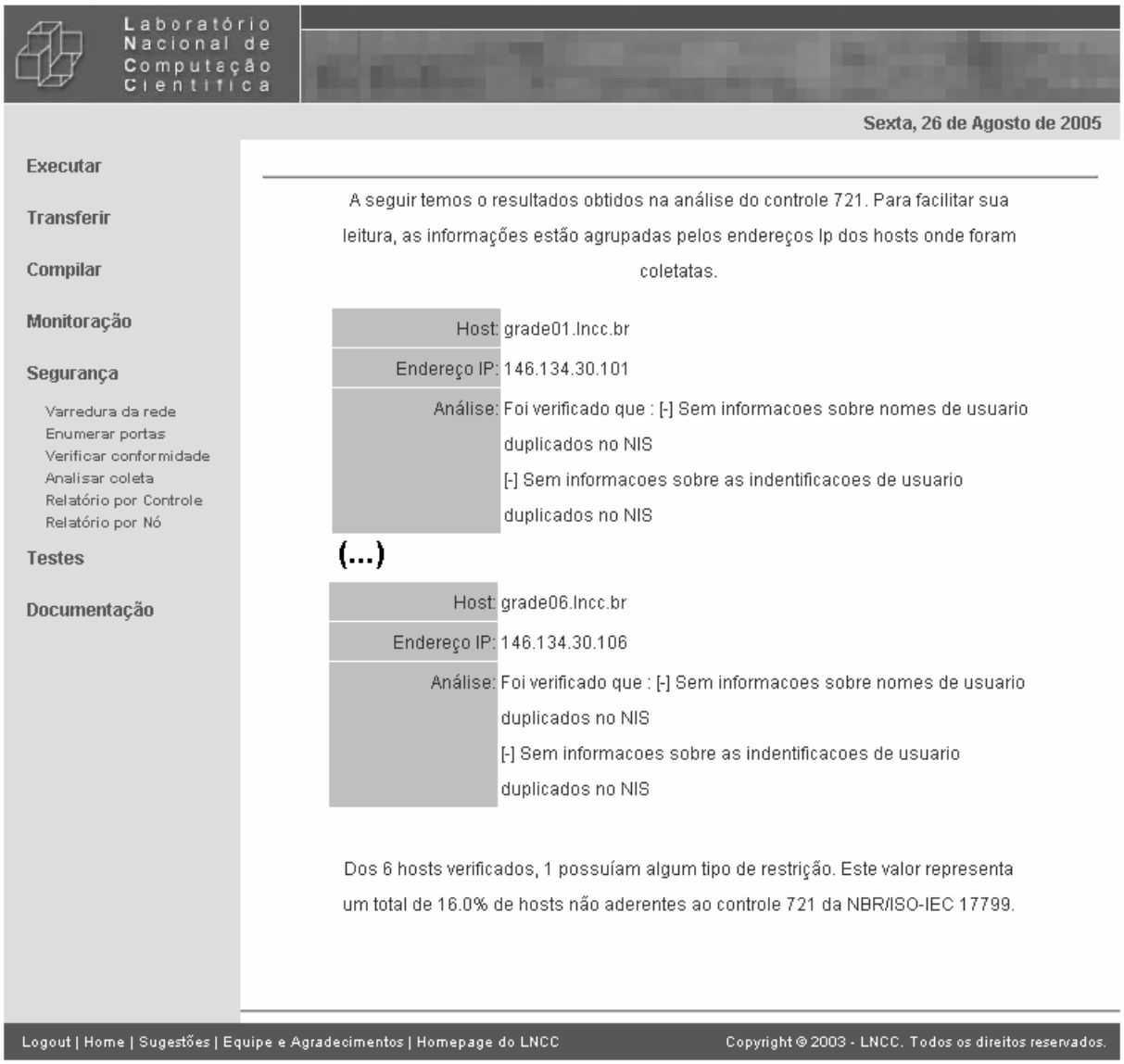

FIG. 5.14 Relatórios por Controle

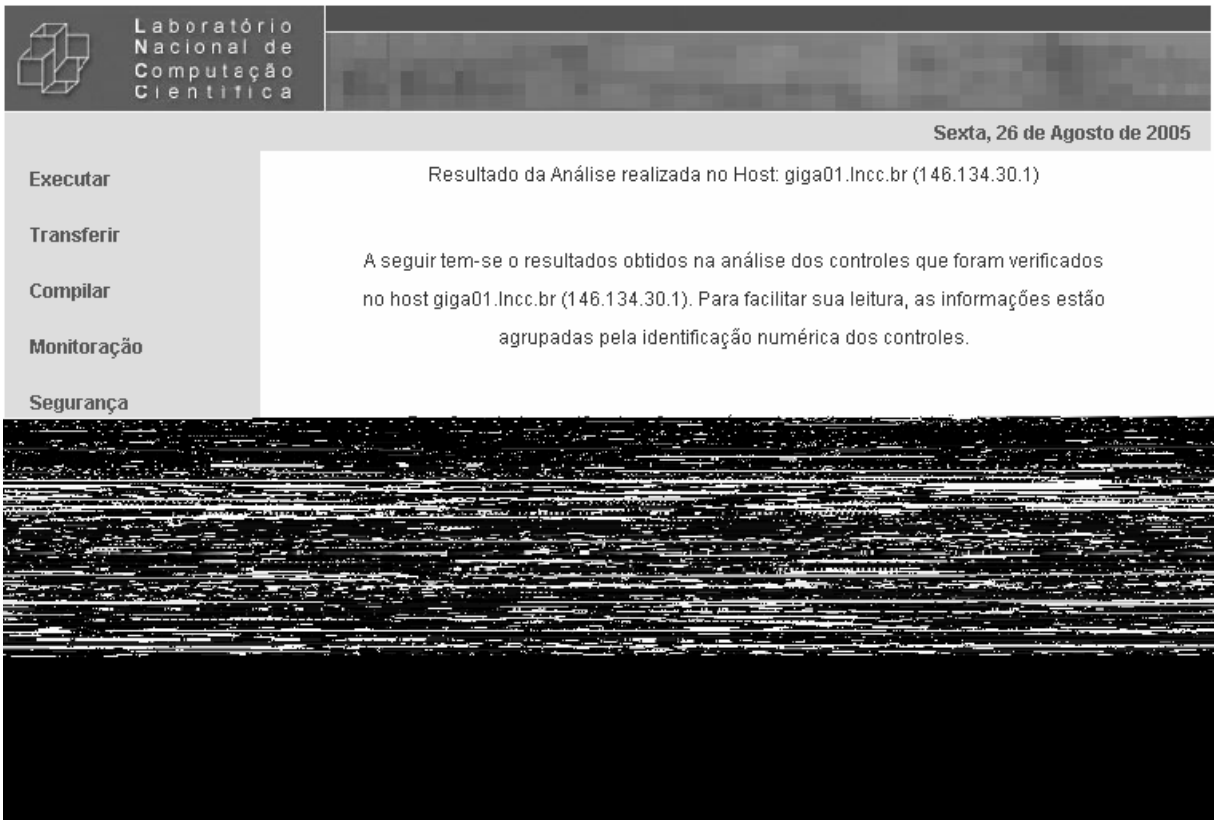

FIG. 5.15 Relatórios por Nó

6 CONCLUSÃO

O principal objetivo da tese, implementar um modelo automático de verificação do nível de aderência de uma grid computacional à Norma Nacional de Segurança da Informação NBR ISO/IEC 17799:2005, foi alcançado, através da construção do SiMPaSeG. Este sistema é baseado no Tamanduá Mirim, e suas funcionalidades permitem a análise de segurança de grids computacionais a partir de um Portal Web de submissão de tarefas de uma grid.

A revisão bibliográfica e dos trabalhos relacionados levou à seleção do protótipo Tamanduá Mirim como uma implementação promissora de um modelo para verificação automática de normas de segurança que possam ser mapeadas em controles. Portanto, o modelo proposto por (GONÇALVES, 2005) foi estendido e adaptado para as particularidades de uma grid computacional. As principais restrições e diferenças entre o modelo aqui proposto para o Tamanduá são que, no SiMPaSeG a verificação dispensa a instalação de software extra nos nós, pois transferir e executar código em nós remotos faz parte da natureza da grid; e que as verificações são executadas como usuário não-privilegiado diferentemente do Tamanduá e ainda pressupõe o acesso não restrito ao sistema de arquivos da máquina anfitriã, ou seja, o usuário local não deve estar jailed e nem chrooted.

A implementação foi realizada de maneira consistente, produzindo modelos visuais nas diferentes fases da análise à construção, o que permitirá o melhor e mais rápido entendimento do sistema por novos desenvolvedores que venham a dar continuidade ao processo de evolução da ferramenta. A construção modularizada permite, assim como o Tamanduá, que novos controles da NBR e de outras normas possam ser verificados automaticamente. As alterações feitas para adaptar o modelo de (GONÇALVES, 2005) à grid, foram feitas de modo a que se preservasse a compatibilidade entre as duas implementações.

6.1 Contribuições

O desenvolvimento do SiMPaSeG e a sua incorporação ao portal da grid resultante do Projeto InteGridade – Grid Sinergia é a principal contribuição a ser destacada, além de se ter dado continuidade, adaptado e estendido o Tamanduá Mirim para a grid. As potencialidades demonstradas pela capacidade de se coletar dados quase que instantaneamente de mais de uma centena de computadores e essas informações servirem de base para a construção de um ambiente computacional científico mais seguro, também merece destaque.

6.2 Sugestões para trabalhos futuros

Como trabalhos futuros surgem a possibilidade da integração desse sistema a uma ferramenta de IDS (Intrusion Detection System) baseada em host (nó) ou de rede; a construção de ferramentas administrativas de grande escala, pois se aproveitando da infra-estrutura de segurança da grade, um administrador poderia realizar correções e instalar patches em um número quase ilimitado de estações quase que instantaneamente; a utilização de ferramentas de configuração ou da tradução dos shell scripts para linguagens orientadas a objetos a fim de facilitar seu desenvolvimento; criar mecanismos de verificação automática dos controles para outras plataformas, como por exemplo: Windows, Solaris, Aix, Qnx, OpenBSD, NetBSD, FreeBSD, entre outras.

7 GLOSSÁRIO DE TERMOS TÉCNICOS E EXPRESSÕES USADAS

- ABNT. Esta é a abreviatura da Associação Brasileira de Normas Técnicas, que é o órgão responsável pela normalização técnica no país, fornecendo a base necessária ao desenvolvimento tecnológico brasileiro. É a única e exclusiva representante no Brasil das seguintes entidades internacionais: International Organization for Standardization - ISO, International Electrotechnical Comission - IEC, COPANT - Comissão Panamericana de Normas Técnicas e Associação Mercosul de Normalização - AMN.

- ACL. Esta é a abreviatura de Access Control List, ou Lista de Controle de Acesso, em português, é um termo em segurança da informação usado para o mecanismo que descreve os direitos de acesso a um recurso computacional.

- API. Esta é a abreviatura de Application Programming Interface, ou seja, é API é um conjunto de definições de como uma parte de um software se comunica com outra. É um método para se obter abstração, usualmente entre softwares de baixo-nível, como o sistema operacional, e de alto-nível , como um programa aplicativo.

- BS 7799. Esta é a norma britânica de segurança da informação e que posteriormente deu origem à ISO/IEC 17799:2005.

- BS 7799-1. Esta é a primeira parte da BS 7799 e aborda um código de práticas para gerência de segurança da informação. Esta norma foi homologada em 1995, revisada em 2005 e é equivalente à ISO/IEC 17799:2005.

- BS 7799-2. Esta é a segunda parte da BS 7799 e é um guia para aplicação da norma. Fornece os fundamentos para auditoria externa e certificação. Esta parte foi homologada em 1999 e revisada em 2002 a fim de se harmonizar com outras normas, como a ISO 9001 e a ISO 14001. Esta norma será substituída pela ISO/IEC 27001:2005.

- BSI. Esta é a abreviatura do BSI British Standards que é o novo nome do British Standards Institute. O BSI possui um Royal Charter para atuar como a organização de normatização do Reino Unido.

- Basiléia II. Esta é a nova versão de um conjunto abrangente de 25 Princípios

2AU8–GPd2T8UG2T8G9rPç2T8UG99H9P52TAUGPd2T8UG99H9G: 0

Basiléia como referência básica para uma supervisão bancária eficaz. Foi originalmente intitulado "Core Principles for Effective Banking Supervision", publicado pelo BIS - Banco de Compensações Internacionais, em setembro de 1997.

- Cobit. Esta é a abreviatura de Control Objectives for Information and related Technology, que é um framework para segurança da informação criado pela ISACA, the Information Systems Audit and Control Association, e pelo ITGI, IT Governance Institute. COBIT possui 34 objetivos de alto nível que cobrem 318 objetivos de controle categorizados em quatro domínios: Planning and Organization, Acquisition and Implementation, Delivery and Support, and Monitor.

- Criptografia. É o ramo da ciência que estuda a proteção da informação por meio de cifras e códigos.

- *Firewall*. Hardware ou software usado para aumentar o nível de segurança de uma rede. Implementa os controles de acesso especificados na política de segurança, controlando o tráfego de rede entre diferentes zonas de confiança.

- IETF. Esta é a abreviatura de Internet Engineering Task Force, uma comunidade internacional aberta composta por projetistas de redes, operadoras de redes, empresas fornecedoras de hardware e software e pesquisadores preocupados com a evolução da arquitetura da Internet. A IETF foi responsável pelo nível de padronização e sucesso da Internet no mesmo período em que a ISO recebia críticas pela falta de agilidade.

- ISO. Esta é a abreviatura da International Organization for Standardization, uma entidade normatizadora internacional composta por representantes das entidades normatizadoras nacionais. É responsável pela emissão de normas e relatórios técnicos e pelos processos de certificação usados em tratados internacionais e pela entidade normatizadoras nacionais.

- ISO/IEC 17799:2005. Norma Internacional de Segurança da Informação. Consulte BS 7799-1.

- ISO/IEC 27001:2005. Norma Internacional de Segurança da Informação. Consulte BS 7799-2.

- NBR. Esta é a abreviatura de Norma Brasileira, neste texto refere-se especificamente à NBR ISO/IEC 17799:2005.

- NBR ISO/IEC 17799:2005. Está é a norma nacional de segurança da informação, cujo conteúdo é uma tradução literal da ISO/IEC 17799:2005.

- NFS. Esta é a abreviatura de Network File System, um sistema de arquivos distribuído largamente usado em redes Unix e Linux.

- NIS. Esta é a abreviatura de Network Information Services, um banco de dados distribuído que pode fornecer informações para autenticação de usuários numa rede Unix ou Linux.

- NMAP. Esta é a abreviatura de Network Mapper, uma ferramenta para varredura e coleta de dados sobre uma rede de computadores ou uma máquina a ela conectada.

- RFC. Esta é a abreviatura de Request For Comments, uma RFC é um documento produzido pela IETF que pode ser utilizado na padronização de serviços e funcionalidades fornecidas por vários protocolos de rede. RFC's, mesmo antes de serem oficialmente adotadas como um padrão Internet (Internet Standard), são largamente utilizadas pela indústria de hardware e de software e pela comunidade de código livre.

- Sarbanes-Oxley. O Sarbanes-Oxley Act foi criado para proteger os investidores ao melhorar a acurácia e confiabilidade dos demonstrativos financeiros das empresas. É oficialmente intitulado "Public Company Accounting Reform and Investor Protection Act of 2002", mas é comumente chamado de SOX ou Sarbox, e foi assinado como lei em 30 de Julho de 2002 pelo governo estadunidense. Esta lei prevê as responsabilidades dos gerentes de TI quanto à implementação de controles e auditorias a fim de proteger os dados financeiros.

- Sniffer. Também chamado de packet sniffer, Ethernet sniffer ou analisador de rede é uma ferramenta capaz de capturar e analisar o tráfego de rede. Pode ser instalado numa máquina, ficando assim restrito aos pacotes trafegados pelo segmento de rede dessa máquina ou mesmo instalado em equipamentos de rede como hubs e switches para assim capturar todos os pacotes que trafegam por estes.

COBRA. **COBRA – Consultative, Objective and Bi-Functional Risk Analysis.** Disponível: http://www.ca-systems.zetnet.co.uk. [capturado em 12 outubro 2005]

COMCIDIS – **Projeto Computação Científica Distribuída (ComCiDis).** Disponível: http://virtual.lncc.br/comcidis/ [capturado em 11 julho 2005]

CONALLEN, J. **Building Web Applications with UML** 2.ed. New York : Addison Wesley October 04, 2002. 496p.

CRAMM. Disponível: http://www.cramm.com. [capturado em 12 outubro 2005]

CSRC – **Computer Security Resource Center (CSRC) - Guidance / Publication / Library.** Disponível: http://csrc.nist.gov/publications/inde

FIGUEIREDO, L. H. de, IERUSALIMSCHY, R., CELES, W. **The design and implementation of a language for extending applications.** Proceedings of XXI Brazilian Seminar on Software and Hardware (1994) 273-283.

FOSTER, I., KESSELMAN, C., NICK, J., TUECKE, S. **The physiology of the grid: An open grid services architecture for distributed systems integration.** In Global Grid Forum, June 2002.

FOSTER, I. **What is the Grid? A Three Point Checklist**. jul 2002. Disponível em: http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf. [capturado em 27 junho 2005]

FOSTER, I., LASZEWSKI, G. von, **Usage of LDAP** in Globus. Disponível: ftp://ftp.globus.org/pub/globus/papers/ldap_in_globus.pdf [capturado em 23 agosto 2005]

FOSTER, I. et al. **GT4 Key Concepts (A Globus Primer).** Disponível: http://globus.org/toolkit/docs/4.0/key/GT4_Primer_0.6.pdf [capturado em 23 agosto 2005]. Draft, Maio 2005.

GIGA – **Projeto GIGA**. Disponível: http://www.rnp.br/pd/giga/ [capturado em 11 julho 2005]

GLOBUSALLIANCE – **Globus Toolkit 2.2 MDS Technology Brief**. Disponível: http://www.globus.org/toolkit/docs/2.4/mds/mdstechnologybrief_draft4.pdf. Draft 4 – January 30, 2003 [capturado em 23 agosto 2005]

GONÇALVES, L. R. de O. **Um Modelo para Verificação, Homologação e Certificação de Aderência à Norma Nacional de Segurança de Informação - NBR ISO/IEC 17799.** Dissertação de Mestrado. Rio de Janeiro : UFRJ, 2005

HSQLDB – **HSQL Database Engine**. Disponível: http://hsqldb.org [capturado em 23 agosto 2005]

IANA – **List of assigned ports**. http://www.iana.org/assignments/port-numbers [capturado em 27 junho 2005]

INTEGRIDADE – **Projeto Integridade – Grid Sinergia**. Disponível: http://virtual.lncc.br/comcidis/Portugues/integridade.html [capturado em 11 julho 2005]

J2EE – **Java 2 Standard Edition**. Disponível: http://java.sun.com/j2ee [capturado em 23 agosto 2005]

J2SE – **Java 2 Standard Edition**. Disponível: http://java.sun.com/j2se [capturado em 23 agosto 2005]

JOSEPH, J., FELLENSTEIN, C. **Grid Computing.** Prentice Hall PTR – 2003

KEAHEY, K., WELCH, V. **Fine-Grain Authorization for Resource Management in the Grid Environment**. Proceedings of Grid2002 Workshop, 2002

KROLL, P., KRUCHTEN, P. **The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP**. New York : Addison Wesley, April 11, 2003. 464p. (FIG. 5.1)

LASZEWSKI, G. von, FOSTER, I., GAWOR, J., LANE, P. **A Java Commodity Grid Kit**. Concurrency and Computation: Practice and Experience, vol. 13, no. 8-9, pp. 643-662, 2001, http:/www.cogkit.org/

LOUKIDES, M. et al. **UNIX Power Tools**. 3rd Edition; O'Reilly – 2002. ISBN 0596003307

MANN, S. e MITCHELL, E. L. **Linux System Security: The Administrator's Guide to Open Source Securiy Tools**. 1st Edition, Prentice Hall PTR, 1999. ISBN 0130158070

MERKLE, R. C., **Protocols for Public Key Cryptosystems**. pp. 122-133 em Proc. 1980 Symp. on Security and Privacy, IEEE Computer Society, Abril 1980

NBR17799 – Associação Brasileira de Normas Técnicas. **NBR ISO/IEC 17799:2005 - Tecnologia da Informação - Código de Práticas para a gestão da Segurança da Informação.** Agosto de 2005.

NOVOTNY, J. **The Grid Portal Development Kit**. Concurrency and Computation: Practice and Experience, Special Issue: Grid Computing Environments, vol. 14, no. 13-15, November-December 2002

POSEIDON – **Poseidon for UML** Disponível: http://www.gentleware.com/index.php?id=poseidon_for_uml [capturado em 23 agosto 2005

QUIGLEY, E. **UNIX Shells by Example**. 3rd Edition; Prentice Hall PTR – 2001. ISBN 013066538X

RISK. **Introduction to Security Risk Analysis.** Disponível: http://www.security-risk-analysis.com/ [capturado em 27 junho 2005]

SARBOX. **U.S. Sarbanes Oxley Act of 2002.** Disponível: http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107_cong_bills&docid=f:h3763enr.tst.pdf. [capturado em 25 outubro 2005]

SARDINHA, L., NEVES, N. F., VERÍSSIMO, P. **Tolerância a Intrusões num Sistema em Grid**. Actas da 7ª Conferência sobre Redes de Computadores (CRC2004), Leiria, Portugal. Outubro 2004.

SCHNEIER, B., **Handbook of Applied Cryptography**, Second Edition. New York. Wiley, 1996. ISBN 0849385237

SECURITYFOCUS – **Security Focus.** Disponível: http://www.securityfocus.com [capturado em 27 junho 2005]

SECURITYREVIEW - **Security Review Products.** Disponível: http://www.securitypolicy.co.uk/secrevie.htm [capturado em 27 junho 2005]

SECURITYSPACE – **Security Space**. Disponível: http://securityspace.com/sspace/. [capturado em 27 junho 2005]

SECURITYTEAM – **Globus Toolkit Version 4 Grid Security Infrastructure: A Standards Perspective** - The Globus Security Team - Version 3 updated July 29, 2005 – Disponível: http://globus.org/toolkit/docs/4.0/security/GT4-GSI-Overview.pdf / [capturado em 11 julho 2005]

SIEVER, E. et al. **Linux, O Guia Essencial.** Segunda Edição. Editora Campus - 2000

STALLINGS, W., **Cryptography and Network Security.** Third Edition," New Jersey. Pearson Education, 2002. ISBN 0130914290

SUBPROJETOS - **Subprojetos de P&D do Projeto Giga.** Disponível: http://www.rnp.br/pd/giga/subprojetos.html [capturado em 11 julho 2005]

SURVEY – **Desktop Linux Market Survey.** Disponível: http://www.desktoplinux.com/articles/AT2127420238.html [capturado em 23 agosto 2005]

TANENBAUM, A. S., STEEN, M. van. **Distributed Systems: Principles and Paradigms,** 1st edition, Prentice Hall; 2002. ISBN 0130888931

TOMCAT – **Apache Tomcat.** Disponível: http://jakarta.apache.org/tomcat/index.html [capturado em 23 agosto 2005]

TOP500 – **25ª Lista dos 500 mais poderosos supercomputadores**. Disponível: http://top500.org/lists/2005/06/ [capturado em 11 julho 2005]

TUECKE, S., CZAJKOWSKI, K., FOSTER, I., FREY, J., GRAHAM, S., KESSELMAN, C., MAQUIRE, T., SANDHOLM, T, SNELLING, D. e VANDERBILT, P. **Open grid services infrastructure (OGSI)**, Junho 2003.

9 APÊNDICES

9.1 APÊNDICE A – CONCEITOS DE CRIPTOGRAFIA

Neste apêndice estão descritos os conceitos criptográficos necessários para o entendimento dos mecanismos de autenticação e delegação implementados na grid.

9.1.1 Criptografia

Mediante a necessidade de comunicação de forma segura, evitando que mensagens ou que o seu conteúdo, sejam usados por adversários, desenvolveram-se códigos e cifras. Tais adversários, por sua vez, buscam técnicas para obter a informação a partir das comunicações cifradas a que tenham acesso. A evolução das cifras e sua história está relacionada a essa luta intelectual entre criadores de cifras e decifradores ao longo do tempo.

A comunicação secreta pode ser obtida por esteganografia (do grego ***steganos*** que significa coberto) ou criptografia (do grego ***kriptos*** que significa oculto). Na primeira a existência da mensagem é ocultada a fim de evitar sua interceptação, enquanto na última o seu significado está oculto. Dos dois, a criptografia é mais eficaz, pois impede que a informação caia em mãos dos adversários mesmo em casos de interceptação da mensagem, situação em que a esteganografia falha.

A criptografia basicamente é obtida por transposição, na qual ocorre o rearranjo de símbolos da mensagem e/ou por substituição quando símbolos são substituídos por outros, seguindo um código. Nos dias atuais, estes mecanismos são implementados por algoritmos de P-box e S-box respectivamente.

Um sistema criptográfico é uma família de transformações inversíveis de parâmetro único E_K ; K pertencente ao conjunto K : P -> C e com o algoritmo inverso E_K^{-1} ; K pertencente ao conjunto K : C -> P tal que a inversa é única. A chave K é a informação necessária para cifrar e decifrar uma mensagem, usando um método geral preestabelecido, o algoritmo de codificação. A segurança da chave deve ser mantida, assim como o sistema de codificação deve possuir um amplo número de

chaves possíveis(K), de modo que não seja possível obter a mensagem em claro pelo teste de todas elas (técnica conhecida como ataque por forca-bruta). Segundo o Princípio de Kerckhoff, a segurança desse criptosistema não deve depender da manutenção em segredo do criptoalgoritmo, mas apenas do sigilo da chave.

9.1.2 Data Encryption Standard (DES) (FIPS 46-3)

Durante a década de 1960 os computadores se popularizam e em 1973, a National Bureau of Standards dos EUA solicitou propostas para um sistema padrão de cifragem. Em 1976, a versão modificada de 56 bits da cifra Lucifer criada pelo alemão Horst Feistel para IBM foi adotada como Padrão de Cifragem de Dados (DES).

A Lucifer é uma cifra de blocos, pois opera separadamente sobre blocos de 64 bits da mensagem original. As alterações introduzidas pela NSA parecem ter sido introduzidas por 2 razões: proteger o DES contra criptoanálise diferencial (conhecida pela NSA desde 70) e publicada em 1990 por Sean Murphy, Eli Biham e Adi Shamir; e enfraquecê-lo ao adotar chaves de 56 bits contra os 112 bits da Lucifer.

9.1.3 Criptografia Assimétrica ou de Chave Pública

Com a padronização pelo DES, o governo, as empresas e os indivíduos tinham agora uma cifra comum, de fácil implementação, rápida e segura, restando apenas o problema de se trocar por um meio seguro uma chave. Esse problema da distribuição de chaves era tido como dogma até a publicação de "New Directions in Cryptography" por Diffie e Hellman [DIFFIE & HELLMAN, 1976].

Diffie, Hellman e Merkle tinham descoberto as cifras assimétricas baseadas em funções de mão única como as da aritmética modular, lançando o conceito de criptografia de chave pública. Todas as cifras até então criadas eram simétricas, ou seja, o processo de decifrar era o oposto ao de cifrar utilizando-se para isso uma única chave. As assimétricas possuem duas chaves em que as mensagens cifradas por uma delas só podem ser decifradas pela outra.

New Directions derrubava o dogma do problema da troca de chaves, mas coube a Rivest, Shamir e Adleman propor um uma cifra prática no ano seguinte. O sistema RSA, como ficou conhecido, lançou a criptografia de chave pública baseada na intratabilidade da fatoração de grandes números gerados pela multiplicação de dois números primos.

A RSA funciona pela escolha de dois grandes números primos (**p** e **q**) e de um terceiro número **e** escolhido de modo que **(p-1)**, **(q-1)** e **e** não sejam primos relativos. A chave pública composta por **N** (**p** * **q**) e por **e** é publicada. Uma mensagem cifrada **C** [$C = M^e \pmod N$] pode ser decifrada pelo possuidor da chave particular [conhecimento de p e q e cálculo de d pela equação **e** x **d** = 1 (mod (**p**-1) x (**q**-1))] usando $M = C^d \pmod N$.

9.1.4 Advanced Encryption Standard (AES) (FIPS 197)

Em Janeiro de 1997, o National Institute of Standards and Technology (NIST) – instituto subordinado ao Departamento de Comércio dos EUA – anunciou o desenvolvimento de um novo algoritmo padrão de criptografia para substituir, o já não tão seguro Data Encryption Standard (FIPS 46-3).

Após três rodadas, em 26 de novembro de 2001, o NIST anunciou a escolha do algoritmo Rijndael como Advanced Encryption Standard (AES) (FIPS PUB 197) entre os quinze algoritmos submetidos para apreciação pública. O novo padrão, adota Rijndael, uma cifra de blocos simétricos capaz de processar blocos de dados com 128 bits de comprimento, usando chaves criptográficas de 128, 192 ou 256 bits de comprimento.

9.1.5 Assinatura Digital

Assinatura digital é uma primitiva criptográfica fundamental na autenticação, autorização e não-repúdio. Fornece a uma entidade a capacidade de associar a sua identidade a um pedaço de informação, ou seja, no processo de assinatura digital

um usuário ou processo transforma a mensagem e alguma informação secreta de seu conhecimento, num rótulo chamado de assinatura.

São requisitos das assinaturas digitais:

- **B** tem que poder validar a assinatura de **A** em M
- deve ser impossível para qualquer um forjar a assinatura de **A**
- caso **A** não honre a assinatura em M deve ser possível uma terceira pessoal ou juiz resolver a disputa entre **A** e **B**.

Em resumo, assinaturas digitais estabelecem autenticidade do remetente, e podem ser fornecidas por sistemas de chave-pública. No entanto, sistemas convencionais, em princípio, não fornecem essa autenticidade pois **A** e **B** compartilham a chave, de modo que seria impossível resolver uma disputa entre ambos, pois **B** pode facilmente forjar mensagens.

Duas aplicações de assinaturas digitais são propostas por Merkle [Merk80] para proteção de software: a primeira, atualmente amplamente adotada, consiste na distribuição de software assinado a vários nós de uma rede, a fim de que se possa verificar a assinatura dos mesmos antes de sua execução; a segunda, envolve a execução de programas privilegiados em sistemas operacionais, em que um sistema, implementado preferencialmente em hardware, negaria a execução de programas não assinados. Esta segunda proposta será incorporada pela Microsoft Co. em seu novo sistema operacional Longhorn, de modo a não apenas proteger o sistema operacional, mas evitar que o mesmo seja usado para práticas ilícitas como produção, reprodução, transmissão ou recepção de conteúdo copyrighted ou até mesmo classificado como ilegal ou censurado por algum governo autoritário.

9.1.6 Infra-estrutura de Chaves Públicas (PKI) e Certificados de Chaves Públicas

A distribuição de chaves públicas, por não necessitar segredo, é geralmente muito mais fácil que a de chaves simétricas, porém a preservação de sua integridade, e consequentemente da autenticidade, é crítica.

De modo a conferir autenticidade, entidades podem eleger uma terceira parte confiável (TTP em inglês), para que esta, após se certificar que uma chave pública pertence a uma entidade, insira uma assinatura que garanta a integridade da chave.

Um certificado de chave pública é, portanto, dividido em duas partes: a parte de dados e a parte da assinatura. Na primeira temos o nome da entidade, sua chave pública correspondente e demais informações relevantes (como endereço postal ou de rede, período de validade e outros atributos), e

9.2 APÊNDICE B – PLATAFORMA DE DESENVOLVIMENTO DO SIMPASEG

Neste apêndice estão detalhados os softwares usados como plataforma de desenvolvimento do SiMPaSeG.

9.2.1 Red Hat Linux 9 (Linux)

A distribuição Red Hat Linux foi lançada em Novembro de 1994 pela Red Hat Inc. e apesar de não ser tão antiga quanto a Slackware, obteve grande popularidade (com 15% do mercado em 2004 e praticamente empatada Mandrake, SuSE e Debian) (SURVEY,2004) e serviu de base para muitas outras distribuições como Mandriva Linux (originalmente Red Hat Linux with KDE), Yellow Dog Linux (Red Hat Linux com suporte à arquitetura PowerPC) e ASPLinux (Red Hat Linux com melhor suporte a caracteres não-Latinos).

A distribuição introduziu o sistema de gerenciamento de pacotes de software 'RPM Package Manager', originalmente chamado de Red Hat Package Manager. Apesar de sua popularidade em todos os segmentos, a Red Hat Inc. optou por focar no mercado corporativo e descontinuou a distribuição da linha Desktop com o lançamento da versão 9 em abril de 2004.

A escolha do Red Hat Linux 9 como plataforma de desenvolvimento foi devido à facilidade de instalação com seu software Anaconda, ao sistema de gerenciamento de pacotes RPM e principalmente pela sua popularidade entre usuários do Globus Toolkit, evidenciada por ser a única versão binária da instalação do Globus na época.

9.2.2 Java 2 Standard Edition Software Development Kit (Java SDK)

Java é uma linguagem de programação orientada a objetos inicialmente desenvolvida por James Gosling e colegas da Sun Microsystems. A versão 1.0 foi lançada em 1996 e suas especificações, assim como a Java Virtual Machine (JVM) e a Java API são mantidas pela comunidade através do 'Java Community Process' da Sun.

A linguagem foi a escolhida novamente por ser a empregada no restante do Portal. Seria ainda a escolhida por suas características como multiplataforma, orientação a objetos e finalmente, por sua rica API.

9.2.3 Java 2 Enterprise Edition (Java EE)

A "Java 2 Platform, Enterprise Edition", J2EE (até a versão 1.4) e recentemente nomeada Java EE é uma plataforma de programação construída sobre a "Java 2 Platform, Standard Edition (J2SE)" para desenvolvimento e execução de aplicações de arquitetura multi-camadas distribuídas. Essa plataforma é definida por especificação, mas também é considerada informalmente como uma linguagem ou padrão.

A Java EE inclui a especificação de várias API, como por exemplo JDBC, client-side applets, RPC, CORBA, e define como coordená-los entre si. A Java EE inclui Enterprise Java Beans, Servlets, Java Server Pages e muitas tecnologias de Web Services, provendo um framework para desenvolvimento e deploying de serviços web para a plataforma Java.

9.2.4 Java Commodity Grid Toolkits (Java CoG Kit)

O Java Commodity Grid Toolkit permite acesso aos serviços da grid através de framework Java, pois seus components provêem funcionalidades cliente e servidoras limitadas. Um subconjunto do Java CoG Kit é distribuído juntamente com as versões 3.0.2, 3.2, 3.9 e 4.0 do Globus Toolkit.

Os Commodity Grid (CoG) Kits permitem que usuários, desenvolvedores de

aplicações e administradores da grid possam usar, programar e administrar

9.2.7 Eclipse IDE

A Plataforma Eclipse é um ambiente de desenvolvimento integrado (IDE) extensível de código-fonte aberto e livre "para qualquer coisa e para nada em particular". Essa plataforma é um subprojeto do Projeto Eclipse cuja missão é adaptar e evoluir a plataforma e ferramentas associadas para atender às necessidade da comunidade de ferramentas de construção de software e seus usuários, para realizar a visão do Eclipse como uma plataforma da indústria.

A Eclipse pode lidar com qualquer tipo de recurso, como arquivos Java, arquivos C, arquivos do MS Word, arquivos HTML, arquivos JSP, entre outros, de uma maneira genérica mas sem saber o que fazer o que fazer em específico com um arquivo em particular. A fim de que possa apresentar real valor, a plataforma vem acompanhada de ferramentas plug-ins que ensinam a plataforma sobre como trabalhar com estes diferentes tipos de recursos de maneira transparente.

9.2.8 Concurrent Version System (CVS)

CVS é um sistema de controle de versão permitindo o registro da história de alteração dos códigos-fonte e documentos. Um sistema de controle de versão é um importante componente da Gerência de Configuração de Fontes (Source Configuration Management - SCM).

Para o desenvolvimento do SiMPaSeG foi criado um repositório CVS para todos os arquivos do sistema.

9.2.9 Bugzilla

Bugzilla é um sistema de rastreamento de defeitos e/ou incorformidades em software, em inglês: "Bug-Tracking System".

O Bugzilla é código-fonte aberto e livre e suas funcionalidades permitem a indivíduos e a grupos de desenvolvedores manterem um registro do ciclo de vida dos bugs em seus produtos, ou seja, com o bugzilla é possível acompanhar o bug

desde o seu registro, passando pela confirmação de sua existência, pela correção, controle de qualidade e fechamento.

9.2.10 FabForce DBDesigner

O DBDesigner é um sistema visual para projeto de bases de dados que integra projeto de bases de dados, modelagem, criação e manutenção num único transparente ambiente.

Esta ferramenta foi escolhida para auxiliar na construção do SiMPaSeG e também será a ferramenta usada para manutenção e construção das novas versões do Tamanduá-Mirim. O DBDesigner é desenvolvido como projeto de código-fonte aberto e sua licença é a GPL. Dentre suas funcionalidades, destaca-se sua capacidade de gerar um modelo a partir de uma base de dados, funcionalidade conhecida como engenharia reversa, e pelos recursos de sincronização entre modelo e base de dados.

9.2.11 Poseidon for UML Community Edition

O Poseidon for UML Community Edition - uma ferramenta para modelagem utilizando Unified Modelling Language (UML) - é a versão gratuita do Poseidon para uso não-comercial.

O SiMPaSeG foi modelado usando UML para facilitar a estruturação das idéias e a clareza na comunicação entre desenvolvedores.

Este é o único software de código-fonte fechado desta plataforma de desenvolvimento.

9.2.12 HSQLDB

O HSQLDB, previamente conhecido como Hypersonic SQL Database, é um servidor de banco de dados relacional totalmente escrito em Java que pode ser usado em modo stand-alone ou em modo cliente/servidor, neste caso aceitando muitos usuários concorrentes. HSQLDB possui um driver Java Database Connectivity (JDBC) e suporta um rico subconjuntos dos padrões ANSI-92 SQL (Structured Query Language), SQL 99 e 2003.

HSQLDB está sendo usado como banco de dados e mecanismo de persistência em muitos projetos de software livre (por exemplo, HSQLDB é o mecanismo de banco de dados do OpenOffice.org 2.0) e mesmo em projetos e produtos comerciais. A versão corrente do HSQLDB (1.8.0) é extremamente estável e confiável, e é reconhecida por seu pequeno tamanho, capacidade de ser executado inteiramente em memória e sua velocidade.

HSQLDB é um banco de dados transacional, pois implementa as semânticas ACID (Atomicity, Consistency, Isolation, and Durability). Ainda assim, sua simplicidade é obtida pela serialização da execução das sentenças SQL, ou seja, apesar de muitos usuários concorrentes poderem estar conectados ao banco de dados, todas as sentenças SQL são enfileiradas e executadas sequencialmente, uma-a-uma. Desta forma, evitou-se a implementação de complicados mecanismos de sincronização e locking.

O HSQLDB é software livre, com uso e distribuição gratuitos sob licença do Hypersonic SQL Group, licença baseada na licença básica do BSD.

9.3 APÊNDICE C - METODOLOGIA DE DESENVOLVIMENTO DO SIMPASEG

A metodologia usada no desenvolvimento do SiMPaSeG tem sua origem no Rational Unified Process (RUP). O RUP é um processo iterativo de desenvolvimento de software criado pela Rational Software Corporation, agora subsidiária da IBM. Apesar de ser um processo comumente usado em grande times de desenvolvedores, RUP também pode ser usado para pequenos projetos de desenvolvimento devido a ser amplamente customizável.

O RUP usa a abordagem da orientação a objetos em sua concepção, e é projetado e documentado utilizando a notação UML (Unified Modelling Language). RUP utiliza técnicas e práticas comercialmente testadas.

Apesar da liberdade para ser customizado, o RUP define algumas linhas mestras:

- Desenvolvimento iterativo: o desenvolvimento iterativo, em contraposição ao em cascata, permite o desenvolvimento de grandes sistemas em ciclos de desenvolvimento que, através de refinamentos sucessivos, possibilitam um maior entendimento do projeto e mitigação de riscos através da priorização do endereçamento dos itens de alto-risco do projeto e da entrega de um 'release' executável ao final de cada iteração. A FIG. 12.1 ilustra o desenvolvimento iterativo no RUP.
- Gestão de requisitos: prevê a documentação apropriada da funcionalidade, restrições de sistema, restrições de projeto e requisitos de negócio. Os requisitos são capturados através dos Casos de Uso (Use Cases).
- Uso de arquitetura baseada em componentes: O uso de componentes possibilita que um sistema possa facilmente extensível, promovendo a reutilização de software e um entendimento intuitivo.
- Uso de software de modelos visuais: Uso intensivo de UML para construção de modelos visuais a fim de permitir melhor entendimento de um dado problema, desta forma, o RUP consegue uma maneira efetiva de se ter uma

visão geral e compartilhada de uma solução.

- Verificação da qualidade do software: O RUP tenta assegurar a qualidade do software verificando-a na construção de todo o processo e envolvendo todos os membros da equipe de desenvolvimento.
- Gestão e controle de mudanças do software: As mudanças de requisitos são inevitáveis e precisam ser gerenciadas, pois uma pequena mudança pode afetar aplicações de formas inteiramente imprevisíveis. O controle de mudanças é essencial para o sucesso de um projeto.

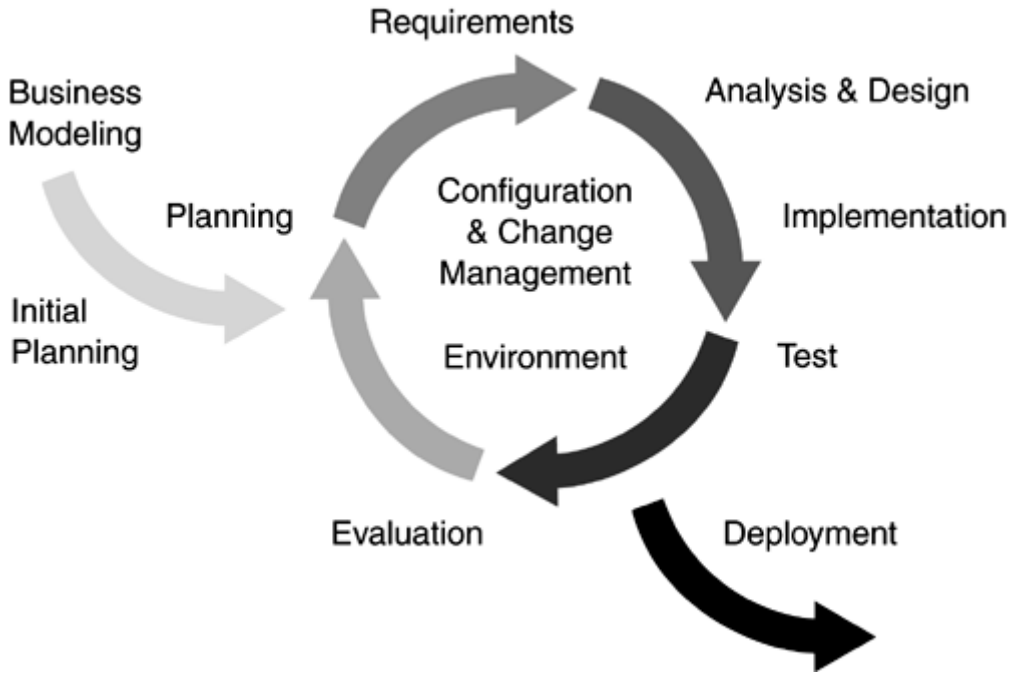

FIG. 12.1 Desenvolvimento iterativo no RUP

O RUP é dividido em 4 fases com objetivos e marco bem definido (FIG. 12.2):

- Fase de Concepção - Nesta fase é definida uma visão e escopo para o produto de software. Os artefatos desta fase são documento de visão, caso de negócio e análise de riscos. Marco: Lifecycle Objective Milestone (LCO)
- Fase de Elaboração - Nesta fase busca-se mitigar os riscos identificados na fase anterior, e definir-se uma arquitetura para o software; para isso, deve-se usar de prototipação sempre que necessário. Os casos de uso de projeto e o planejamento dos desenvolvimento e dos testes são elaborados. Marco:

Lifecycle Architecture Milestone (LCA)

- Fase de Construção - Nesta fase é que são implementados os componentes e outros características do sistema. A codificação é iniciada realmente nesta fase, sendo que ao seu fim, ocorre a entrega da primeira versão do software. Marco: Initial Operational Capability Milestone (IOC)
- Fase de Transição - Nesta fase o produto passa dos desenvolvedores ao usuário final. As atividades desta fase incluem o treinamento dos usuários finais, e os testes de funcionalidade completa, ou testes Beta, são realizados para se verificar se o sistema atende as expectativas do usuário. O produto é ainda verificado contra o nível de qualidade acordado na fase de concepção, e contra os padrões dos usuários finais, se não atender às expectativas, o ciclo inteiro recomeça. Marco: Product Release Milestone (PR)

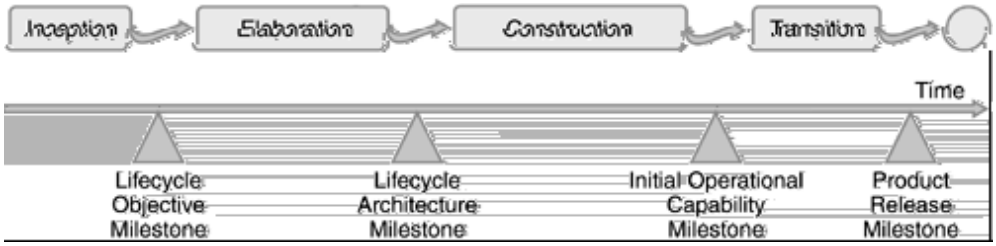

FIG. 12.2 Marcos das fases do RUP

A FIG. 12.3 ilustra bem as duas dimensões do RUP: o aspecto dinâmico (horizontal) expresso em ciclos, fases, iterações e marcos; e o aspecto estático (vertical) expresso nas atividades, disciplinas, artefatos e papéis.

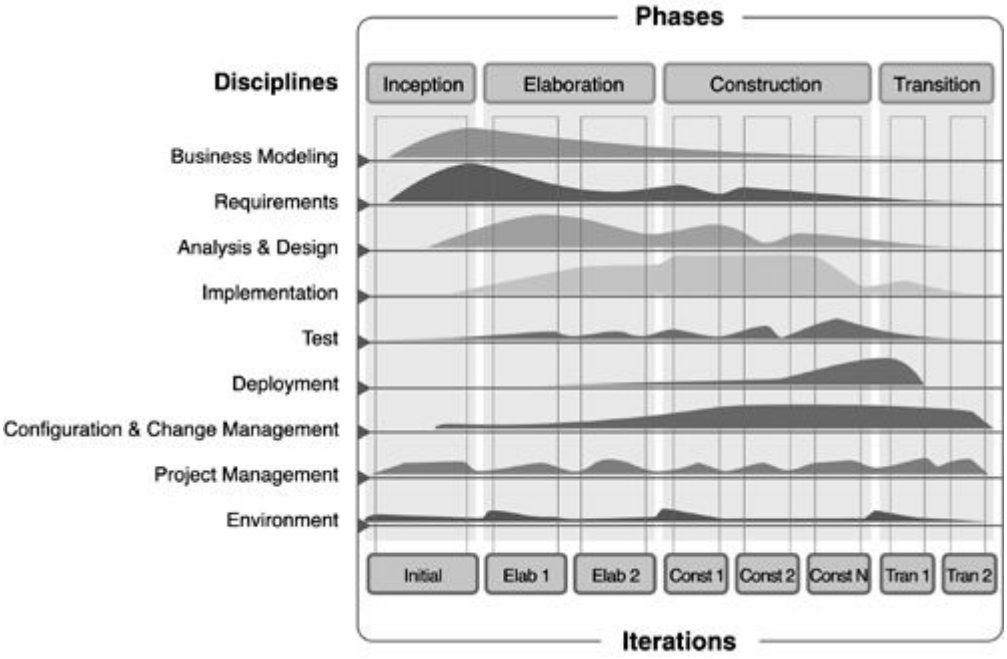

FIG. 12.3 Duas dimensões do RUP