Repositório Seguro de Aplicações Baseado em GSS*

Jose de R. B. Pinheiro Júnior, Alexandre César Tavares Vidal, Fabio Kon

[1]Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo
http://gsd.ime.usp.br/integrade

{jrbraga,vidal,kon}@ime.usp.br

Abstract. *Security is a key concept in distributed systems; in Opportunistic Grid Computing, resources, applications, and user data must be protected. This article describes the implementation of the InteGrade secure application repository. First, we present some important security-related issues in Grid Computing and how they are addressed in InteGrade. Next, we describe the InteGrade architecture and explain the implementation of the security protocols for the application repository. Finally, we show some experimental results on the performance of the proposed implementation, and discuss ongoing and future work.*

Resumo. *Segurança é um conceito chave em sistemas distribuídos. Em sistemas de Grades Computacionais Oportunistas, recursos, aplicações, e dados dos usuários devem ter garantias de segurança. Esse artigo descreve a implementação do repositório seguro de aplicações do InteGrade. São apresentados problemas de segurança em Grades de computadores e como eles são abordados no InteGrade. Segue-se uma descrição da arquitetura do InteGrade e a implementação dos protocolos de segurança no repositório de aplicações. São mostrados, por fim, resultados experimentais relacionados ao desempenho da implementação proposta, trabalhos em andamento e idéias para trabalhos futuros.*

1. Introdução

O conceito de Grade Computacional Oportunista (*Opportunistic Computacional Grid*) [Livny et al., 1997] surge frente a possibilidade de permitir que máquinas ociosas, disponibilizadas em uma rede de computadores, colaborem na solução de problemas envolvendo grande demanda de recursos computacionais. A arquitetura da Internet (TCP/IP) firmou-se como o padrão de fato das redes de computadores. Contudo, a pouca preocupação, no surgimento da tecnologia das redes de computadores, com a segurança dos dados transmitidos, [Garfinkel and Spafford, 1996], vem motivando uma demanda por técnicas adicionais para manter a segurança das informações.

A proteção da informação de um sistema possui quatro objetivos fundamentais [Ford, 1994]. A confidencialidade garante a proteção da informação contra o acesso não autorizado. A integridade visa manter a consistência dos dados, impedindo sua modificação de forma intencional ou não. A disponibilidade garante que usuários

*Este trabalho recebeu apoio da RNP/FINEP.

legítimos terão acesso aos recursos a que têm direito. O uso legítimo determina que os recursos somente serão disponíveis a pessoas autorizadas.

Os computadores que participam de Grades computacionais [Berman et al., 2003, Foster and Kesselman, 2003, de Camargo et al., 2004] tornam-se mais vulneráveis a problemas com a segurança (sistemas operacionais e middleware da grade podem não ser seguros o bastante). Este artigo descreve a implementação de segurança para proteção das aplicações executadas no sistema de Grade Computacional InteGrade [Goldchleger et al., 2004] contra usuários maliciosos.

2. Trabalhos Relacionados

O *Globus Toolkit* [Foster and Kesselman, 1997] é uma ferramenta de software baseada em padrões da indústria para construção de sistemas para grade de modo incremental. O Globus disponibiliza um serviço de segurança denominado GSI (*Globus Security Infrastructure*), o qual implementa autenticação única, comunicação segura e delegação. O Legion [Grimshaw et al., 1997] é um middleware que integra diversos recursos computacionais para prover aos usuários, de maneira transparente, a visão de um único e poderoso computador. Um usuário no Legion é representado por um objeto de autenticação que contém a chave secreta criptografada do usuário e informações adicionais. O OurGrid é um sistema de computação em grade baseado em redes *peer-to-peer* cujo foco é a execução de aplicações *bag-of-tasks* [Cirne et al., 2003]. Para impedir que uma aplicação maliciosa use os recursos de uma forma indevida (inclusive a própria rede), o OurGrid isola a execução de aplicações numa máquina virtual (*sandbox*) [Andrade et al., 2005].

3. O InteGrade

Uma Grade InteGrade se constitui de aglomerados (*clusters*) de computadores organizados de forma hierárquica e escalável. Os nós de um aglomerado do InteGrade podem ser de quatro tipos: (i) o nó gerenciador do aglomerado suporta a coordenação do aglomerado e a comunicação com gerenciadores de outros aglomerados. Estas atividades podem ser distribuídas para balanceamento de carga ou replicadas para tolerância a falhas; (ii) o nó dedicado é exclusivo para execução de aplicações da Grade; (iii) o nó compartilhado disponibiliza tempo ocioso para execução de aplicações dos usuários da grade; (iv) e o nó de usuário pertence ao usuário da grade que submete aplicações à grade.

O módulo *Global Resource Manager* (GRM) executa no nó gerenciador do aglomerado. Ele usa as informações sobre o estado dos demais nós, coletadas e enviadas pelos módulos *Local Resource Manager* (LRMs), para escalonamento das aplicações na grade, com base em seus requisitos e na disponibilidade de recursos. O LRM também é responsável pela execução das aplicações nos nós da Grade. O módulo *Application Submission and Control Tool* (ASCT) permite aos usuários do InteGrade registrar aplicações armazenadas em um repositório e submetê-las para execução na Grade.

O módulo *Application Repository* (*AppRepos*) permite o armazenamento das aplicações da Grade por usuários e administradores. Um esquema de metadados mantém informações sobre as aplicações e suas versões de código binário para diferentes plataformas. Essas informações são utilizadas no escalonamento de uma aplicação quando de sua execução. O módulo de segurança do InteGrade possibilita ao repositório de aplicações

incorporar informações associando permissões de usuários a aplicações registradas, resultando no repositório seguro de aplicações.

4. Implementação

Autenticação, confidencialidade, integridade, autorização e registros de eventos (*logging*) são características da implementação do repositório seguro de aplicações. O repositório seguro de aplicações foi implementado sobre a API GSS, cujos serviços escondem detalhes de segurança da rede. A GSS disponibiliza os conceitos de contexto de segurança – um estado de confiança entre grupo de aplicações, e serviços de segurança (integridade e confidencialidade, na GSS); enquanto o primeiro garante que os dados não serão modificados, o último certifica que eles não serão interceptados. Ao se criar um contexto de segurança, cliente e servidor poderão assinar e criptografar as mensagens que trocarem durante toda a sessão do serviço a ser utilizado.

Os módulos (*Local Security Manager*) (LSM) e (*Global Security Manager*) (GSM) usam a GSS para obter os contextos de segurança entre o repositório de aplicações e os módulos que com ele interagem. O GSM inicia e gerencia os contextos. O repositório usa o GSM (via o LSM) para verificar e assinar os arquivos executáveis das aplicações dos seus clientes, enquanto estes clientes usam o LSM para assinar e verificar arquivos durante o armazenamento e recuperação do repositório. A GSS, tem seus serviços implementados através do Kerberos. A implementação atual do repositório seguro utiliza a versão 5 do Kerberos para a linguagem C e a API GSS Java para a linguagem Java.

No protocolo de armazenamento de uma aplicação do InteGrade, o ASCT usa o LSM para assinar o arquivo e solicitar o seu armazenamento ao repositório de aplicações. O repositório verifica a assinatura do binário pelo LSM (e este usa o GSM), calcula um resumo do binário através de uma função *hash* conhecida e o armazena no sistema de arquivos, e por fim, envia a identificação (ID) assinada da aplicação, que por sua vez também é verificada. No protocolo de recuperação de um arquivo, o LRM usa o ID da aplicação para assinar e indicar ao repositório de aplicações, o arquivo executável desejado. O repositório obtém o arquivo e verifica sua integridade através da função de *hash*. Antes de enviá-lo ao LRM, o repositório assina o binário através do LSM, que repassa essa função para o GSM. Ao receber o arquivo, o LRM verifica sua assinatura e inicia sua execução. Em ambos os protocolos, em caso de falha, a exceção gerada é tratada e registrada em arquivo de *log*.

Realizaram-se experimentos para avaliação de desempenho na transferência de arquivos de diferentes tamanhos entre o repositório e um nó da Grade. Foram usados os algoritmos de criptografia DES triplo e SHA1 e uma amostra com dez binários executáveis de tamanhos distintos. Os tempos de execução obtidos com os módulos de segurança ativados foram maiores que os obtidos com os mecanismos de segurança desativados. Essa diferença deve-se às repetidas operações de criptografia e descriptografia na movimentação de cada arquivo.

5. Conclusões e Trabalhos Futuros

A implementação do repositório seguro de aplicações provê confidencialidade e integridade através de serviços na API GSS. O uso legítimo dos recursos é garantido pela sessão com autenticação prévia dentro de um contexto de segurança associada às

restrições de visão por usuário. O repositório seguro também armazena registros (*logs*) para auxiliar na análise de possíveis incidentes de segurança. O sistema de segurança para Grade que estamos desenvolvendo usa redes de confiança [Ellison et al., 1999] para implementar um mecanismo de segurança totalmente distribuído com suporte a federações de aglomerados InteGrade.

As causas da vulnerabilidade do esquema de autenticação automática na implementação com o Kerberos são duas: a dependência da segurança do sistema de arquivos da máquina onde são executados os serviços e a natureza centralizada do sistema. Características do sistema que podem ser melhoradas ou estendidas futuramente incluem a definição de grupos de usuários e o controle de acesso ao conteúdo do repositório no estilo UNIX para facilitar a visualização e a execução das aplicações.

Referências

Andrade, N., Costa, L., Germóglio, G., and Cirne, W. (2005). Peer-to-peer grid computing with the ourgrid community. In *SBRC 2005 - IV Salão de Ferramentas*, pages 1191–1198. Simpósio Brasileiro de Redes de Computadores (SBRC).

Berman, F., Fox, G., Hey, A. J. G., and Hey, T. (2003). *Grid Computing: Making the Global Infrastructure a Reality*. John Wiley & Sons, Inc.

Cirne, W., Paranhos, D., Costa, L., Santos-Neto, E., Brasileiro, F., Sauvé, J., Silva, F. A. B., Barros, C. O., and Silveira, C. (2003). Running Bag-of-Tasks Applications on Computational Grids: The MyGrid Approach. In *Proceedings of the 2003 International Conference on Parallel Processing*, pages 407–416.

de Camargo, R. Y., Goldchleger, A., Carneiro, M., and Kon, F. (2004). Grid: An Architectural Pattern. In *The 11th Conference on Pattern Languages of Programs (PLoP'2004)*, Monticello, Illinois, USA.

Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B., Bell, S., and Ylonen, T. (1999). SPKI Certificate Theory. Internet RFC #2693.

Ford, W. (1994). *Computer communications security: principles, standard protocols and techniques*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Foster, I. and Kesselman, C. (1997). Globus: A Metacomputing Infrastructure Toolkit. *International Journal of Supercomputer Applications*, 2(11):115–128.

Foster, I. and Kesselman, C. (2003). *The Grid 2: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers Inc.

Garfinkel, S. and Spafford, G. (1996). *Practical UNIX & Internet Security*. O Reilly & Associates, Inc.

Goldchleger, A., Kon, F., Goldman, A., Finger, M., and Bezerra, G. C. (2004). InteGrade: object-oriented Grid middleware leveraging the idle computing power of desktop machines. *Concurrency and Computation: Practice and Experience*, 16(5):449–459.

Grimshaw, A. S., Wulf, W. A., and Team, T. L. (1997). The Legion Vision of a Worldwide Virtual Computer. *Communications of the ACM*, 40(1):39–45.

Livny, M., Basney, J., Raman, R., and Tannenbaum, T. (1997). Mechanisms for High Throughput Computing. *SPEEDUP Journal*, 11(1).