Módulo 7

Segurança

Lição 4

Segurança em Java

Versão 1.0 - Jan/2008

Autor
Aldwin Lee
Cheryl Lorica

Equipe
Rommel Feria
John Paul Petines

Necessidades para os Exercícios

Sistemas Operacionais Suportados

NetBeans IDE 5.5 para os seguintes sistemas operacionais:

- Microsoft Windows XP Profissional SP2 ou superior
- Mac OS X 10.4.5 ou superior
- Red Hat Fedora Core 3
- Solaris™ 10 Operating System (SPARC® e x86/x64 Platform Edition)

NetBeans Enterprise Pack, poderá ser executado nas seguintes plataformas:

- Microsoft Windows 2000 Profissional SP4
- Solaris™ 8 OS (SPARC e x86/x64 Platform Edition) e Solaris 9 OS (SPARC e x86/x64 Platform Edition)
- Várias outras distribuições Linux

Configuração Mínima de Hardware

Nota: IDE NetBeans com resolução de tela em 1024x768 pixel

Sistema Operacional	Processador	Memória	HD Livre
Microsoft Windows	500 MHz Intel Pentium III workstation ou equivalente	512 MB	850 MB
Linux	500 MHz Intel Pentium III workstation ou equivalente	512 MB	450 MB
Solaris OS (SPARC)	UltraSPARC II 450 MHz	512 MB	450 MB
Solaris OS (x86/x64 Platform Edition)	AMD Opteron 100 Série 1.8 GHz	512 MB	450 MB
Mac OS X	PowerPC G4	512 MB	450 MB

Configuração Recomendada de Hardware

Sistema Operacional	Processador	Memória	HD Livre
Microsoft Windows	1.4 GHz Intel Pentium III workstation ou equivalente	1 GB	1 GB
Linux	1.4 GHz Intel Pentium III workstation ou equivalente	1 GB	850 MB
Solaris OS (SPARC)	UltraSPARC IIIi 1 GHz	1 GB	850 MB
Solaris OS (x86/x64 Platform Edition)	AMD Opteron 100 Series 1.8 GHz	1 GB	850 MB
Mac OS X	PowerPC G5	1 GB	850 MB

Requerimentos de Software

NetBeans Enterprise Pack 5.5 executando sobre Java 2 Platform Standard Edition Development Kit 5.0 ou superior (JDK 5.0, versão 1.5.0_01 ou superior), contemplando a Java Runtime Environment, ferramentas de desenvolvimento para compilar, depurar, e executar aplicações escritas em linguagem Java. Sun Java System Application Server Platform Edition 9.

- Para **Solaris**, **Windows**, e **Linux**, os arquivos da JDK podem ser obtidos para sua plataforma em http://java.sun.com/j2se/1.5.0/download.html
- Para **Mac OS X**, Java 2 Plataform Standard Edition (J2SE) 5.0 Release 4, pode ser obtida diretamente da Apple's Developer Connection, no endereço: http://developer.apple.com/java (é necessário registrar o download da JDK).

Para mais informações: http://www.netbeans.org/community/releases/55/relnotes.html

Colaboradores que auxiliaram no processo de tradução e revisão

Aécio Júnior
Alexandre Mori
Alexis da Rocha Silva
Angelo de Oliveira
Bruno da Silva Bonfim
Denis Mitsuo Nakasaki
Emanoel Tadeu da Silva Freitas
Guilherme da Silveira Elias
Leandro Souza de Jesus
Lucas Vinícius Bibiano Thomé
Luiz Fernandes de Oliveira Junior
Maria Carolina Ferreira da Silva
Massimiliano Giroldi
Paulo Oliveira Sampaio Reis
Ronie Dotzlaw

Auxiliadores especiais

Revisão Geral do texto para os seguintes Países:

- **Brasil** – Tiago Flach e Vinícius G. Ribeiro (Especialista em Segurança)
- **Guiné Bissau** – Alfredo Cá, Bunene Sisse e Buon Olossato Quebi – ONG Asas de Socorro

Coordenação do DFJUG

- **Daniel deOliveira** – JUGLeader responsável pelos acordos de parcerias
- **Luci Campos -** Idealizadora do DFJUG responsável pelo apoio social
- **Fernando Anselmo -** Coordenador responsável pelo processo de tradução e revisão, disponibilização dos materiais e inserção de novos módulos
- **Rodrigo Nunes -** Coordenador responsável pela parte multimídia
- **Sérgio Gomes Veloso -** Coordenador responsável pelo ambiente JEDI™ (Moodle)

Agradecimento Especial

John Paul Petines – Criador da Iniciativa JEDI™
Rommel Feria – Criador da Iniciativa JEDI™

1. Objetivos

Segurança na plataforma Java foi inicialmente concebida para proteger as informações em um computador de serem acessadas ou modificadas (incluindo alterações que um programa malicioso poderia introduzir).

Essas ações já foram implementadas no modelo Java de segurança desde a versão 1.0. O serviço de **autenticação** foi adicionado em seguida, na versão 1.1, seguido do modelo de **criptografia** disponível na versão 2.0 (como uma extensão) e **auditoria** que pode ser acrescentada por qualquer programa Java, fornecendo uma auditoria completa de gerenciamento de segurança. Provavelmente novos serão serão adicionados no futuro.

Ao final desta lição, o estudante será capaz de:

- Descriminar o que é Segurança
- Compreender o modelo de Segurança implementado em Java
- Obter mais dados sobre a *Sandbox*

2. O que é Segurança?

Entende-se por segurança como um conjunto de serviços ou aspectos que fornecem maior confiança ao uso de um sistema, dentre as quais, destacam-se:

- Proteção contra programas maléficos

Programas não devem ter autorização a executarem processos que podem prejudicar o ambiente de um usuário, tais como cavalos de Tróia e programas prejudiciais que se replicam, como um vírus de computador.

- Bloqueio a programas invasivos

Os programas devem ser impedidos de descobrir informações particulares sobre o computador ou sobre a rede do computador.

- Autenticação

A identidade das partes envolvidas no programa, tanto do autor como a do usuário, devem ser verificadas.

- Criptografia

Os dados que o programa envia e recebe através da rede, como arquivos ou dados, devem ser codificados.

- Auditoria

Operações potencialmente sensíveis sempre devem ser autenticadas.

- Bem-definida

Especificações de segurança devem ser seguidas.

- Verificação

Regras de funcionamento devem ser definidas e confirmadas.

- Bem-comportada

Os programas devem ser impedidos de consumir recursos do sistema em demasia: utilização da CPU por muito tempo, excesso de memória, entre outros

- Padrão de segurança

Os programas devem ser compatíveis com algum padrão de segurança. Pode-se possuir algum tipo de certificação, ou seja, certificar que determinados procedimentos de segurança são seguidos. Por exemplo, certificação C2 ou B1 do governo dos Estados Unidos.

2.1. Compreendendo como funciona a Segurança em Java

O ponto da condução de segurança é o modelo de distribuição de programas Java. Uma das maiores forças de Java é a sua capacidade para trazer programas em uma rede ou executar os programas a partir de outra JVM. Isso é algo a mais que os atuais usuários de computador dentro de um contexto com uma única JVM no navegador. Embora a idéia por trás de código portátil, que está começando a se infiltrar em outras aplicações, tais como, as aplicações baseadas na tecnologia JINI. Conjugado com o aumento da utilização da Internet, Java possui a capacidade de trazer programas para um usuário conforme sua necessidade. Essa característica de programas já desenvolvidos serem facilmente distribuídos tem sido a forte razão para a rápida aceitação e implantação de sistemas Java.

Em última análise, Java é ou não seguro? Este é um julgamento subjetivo e individual no qual os usuários terão de fazer com base nas suas próprias exigências. Para se ter um sistema livre de vírus, que possa autenticar corretamente ou encriptar seus dados, é exigido que todas as operações sejam auditadas. Desta forma, é necessário construir um auditoria nas aplicações e, em seguida, na plataforma Java.

Uma visão muito pragmática da segurança pode ser então elaborada: a questão não é se um

sistema que necessita de uma determinada característica é qualificado como "não seguro", de acordo com a definição de segurança. A questão é saber se Java possui as características que atendem às suas necessidades.

3. Implementação Segura em Java

Segurança em Java é fornecida através de algumas partes do núcleo de sua plataforma, além da biblioteca *Java Cryptography Extension* (JCE) e *Java Secure Sockets Extension* (JSSE).

Estas são as facilidades básicas fornecidas pelo núcleo de segurança da plataforma Java:

- Uma política de segurança configurável que permite evitar que programas Java leiam seus arquivos, fazendo ligações de rede para outros hospedeiros, acessando a impressora sem a devida permissão, e assim por diante. Esta política é baseada em *Java Access Control*, que por sua vez depende das classes *Loaders*, *Security Manager* e proteções da linguagem.
- A capacidade de gerar *Message Digests* e obter uma maneira simples (mas não segura) de determinar se os dados que o seu programa lê foram alterados.
- A capacidade de gerar assinaturas digitais para detectar se os dados lidos no programa foram modificados (ou para enviar dados e permitir que o destinatário destes possam detectar se foram modificados durante o transporte).
- Um elemento-chave para administração das chaves necessárias para se obter as assinaturas digitais.
- Um suporte a infra-estrutura extensível.

JCE alavanca o núcleo da plataforma Java de segurança ao fornecer uma variedade de operações criptográficas:

- Criptografia básica e avançada
- Segurança na chave
- Segurança no corpo da mensagem
- Uma alternativa à chave de gerenciamento de sistema

Por último, a **JSSE** fornece *Secure Sockets Layer* (SSL), que auxilia na criptografia dos dados. Para se comunicar com um **servidor SSL** ou **cliente SSL**, podemos utilizar estas bibliotecas de extensão. Se a aplicação escrever no cliente ou no servidor e desejamos aplicar a criptografia então, podemos utilizar essa extensão ou as facilidades da **JCE**.

3.1. Sandbox Java

Este modelo de segurança gira em torno da idéia de uma *Sandbox* (vista em lições anteriores). A idéia central é que um aplicativo possa ser alojado de forma segura e restrita em um determinado computador. Pretende proporcionar um ambiente onde este programa possa ser executado entretanto, controlar sua execução em uma área com certos limites. É possível permitir que este programa tenha acesso somente a determinados recursos do sistema, em geral, certificar-se de que o programa estará confinado dentro de uma *Sandbox*.

A *Sandbox* é encarregada de proteger um determinado número de recursos. Considere os recursos típicos de um computador: o usuário da máquina tem acesso, por exemplo, a muitas coisas, internamente, acesso a memória local (memória RAM) e, externamente, acesso a um servidor de arquivos ou outras máquinas da rede. Para executar *Applets*, também devemos ter acesso ao *WEB Services*, cujo acesso pode ser feito através da rede particular ou da internet.

Dados trafegam através de todo este modelo, a partir da máquina do usuário, por meio da rede e, eventualmente, para o disco rígido. Cada um desses recursos deve ser protegido. Essas proteções da *Sandbox* formam a base do modelo de segurança do Java, tais como:

- Um programa que possui acesso a memória e a CPU, bem como ao servidor WEB onde foi carregado. Isso é muitas vezes é um estado padrão para a *Sandbox*.
- Um programa que possui acesso à CPU, a memória, ao servidor WEB e um conjunto de recursos específicos (tais como, arquivos locais, máquinas locais, entre outros). Uma palavra de processamento de programa, por exemplo, pode ter acesso aos documentos do diretório no sistema de arquivo local, mas não em relação a quaisquer outros arquivos

- Um programa que possui o acesso a todos os recursos seja qual for a máquina servidora que se encontra instalado.

Sandbox não é um uniforme padrão para todos os modelos. A expansão de suas fronteiras é baseada em uma noção de confiança. Em alguns casos, pode-se confiar nos programas Java para acessar o sistema de arquivos, em outros, pode-se confiar o acesso a apenas parte do sistema de arquivos, e ainda, em outros casos, não é possível confiar-lhes nenhum acesso ao sistema de arquivos.

Parceiros que tornaram JEDI™ possível

Instituto CTS
Patrocinador do DFJUG.

Sun Microsystems
Fornecimento de servidor de dados para o armazenamento dos vídeo-aulas.

Java Research and Development Center da Universidade das Filipinas
Criador da Iniciativa JEDI™.

DFJUG
Detentor dos direitos do JEDI™ nos países de língua portuguesa.

Banco do Brasil
Disponibilização de seus *telecentros* para abrigar e difundir a Iniciativa JEDI™.

Politec
Suporte e apoio financeiro e logístico a todo o processo.

Borland
Apoio internacional para que possamos alcançar os outros países de língua portuguesa.

Instituto Gaudium/CNBB
Fornecimento da sua infra-estrutura de hardware de seus servidores para que os milhares de alunos possam acessar o material do curso simultaneamente.