Lição 2

Sandbox

Objetivos

Ao final desta lição, o estudante será capaz de:

- Identificar o modelo de segurança padrão empregado – Sandbox
- Conhecer os componentes da Sandbox
- Realizar as configurações dos componentes da Sandbox
- Definir os domínios de proteção de sua aplicação
- Aplicar a política de segurança, por intermédio das permissões
- Entender como as classes podem ser assinadas (certificação digital)

Modelo Sandbox

- Tamanhos:
  - Mínima
  - Padrão
  - Restrita
  - Aberta
- Fundamentos:
  - Características de segurança construídas na JVM
  - A arquitetura do carregador de classes
  - O verificador de arquivos de classe
  - O gerenciador de segurança e a API Java

Componentes Principais – Carregador de Classes

- Carrega o código de bytes
- Classes carregadas a partir do sistema de arquivos local tem seu próprio nome de espaço

Componentes Principais – Verificador de Bytecodes

- Checar as seguintes validações:
  - Regras da linguagem de Programação Java
  - Restrições de nome de espaço
  - Estouros de pilha (overflows e underflows)
  - Conversão ilegal de tipos de dados
- Serve para provar:
  - O arquivo de classe tem o formato correto
  - Segurança de classes e métodos finais
  - Cada classe possui uma simples superclasse
  - Não existe conversão ilegal de atributos ou objetos

Componentes Principais – Gerenciador de Segurança

- Por padrão, um gerenciador de segurança não é usado quando um programa está sendo executado. Para habilitar uma aplicação Java para usá-lo, deve ser especificado:

```
-D java.security.manager
```

Elementos da Sandbox Java - Permissões

- java.io.FilePermission
- java.net.SocketPermission
- java.util.PropertyPermission
- java.lang.RuntimePermission
- java.awt.AWTPermission
- java.net.NetPermission
- java.security.SecurityPermission
- java.io.SerializablePermission
- java.lang.reflect.ReflectPermission
- java.security.AllPermission

Elementos da Sandbox Java – Code Sources

- Localizações que indicam onde as classes serão carregadas
- Pode-se associar permissões baseadas na URL
- Combinação de *codebase* e assinante
- Final da URL:
  - URL especifica um arquivo .jar
  - URL termina com uma barra
  - URL termina com um asterisco
  - URL termina com um hífen

Domínios de Proteção – Keystore

- Assinadas através de certificados digitais com a ferramenta *jarsigner*
- Pode ser manipulado através do uso de uma *keystore*
- São administradas através do utilitário *keytool*

Domínios de Proteção – Arquivos de Política de Segurança

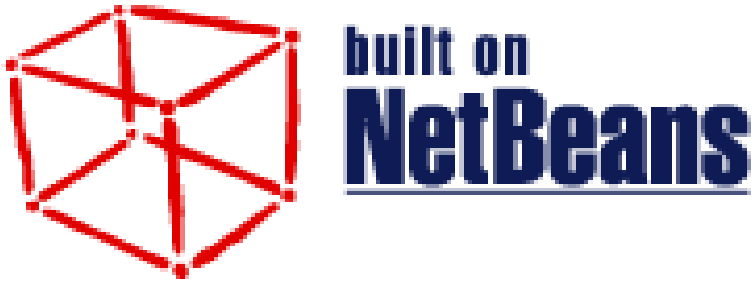
- Administração da *Sandbox* Java é feita listando-se várias permissões
- JVM pode manipular múltiplos arquivos
- São arquivos de texto
- Ferramenta *policytool* é utilizada para administrar
- Código carregado advém de uma única localização
- Permissões concedidas para uma única *Code Source* são concedidas como a união de todas as permissões

Domínios de Proteção – Arquivos de Política de Segurança

- Ferramenta *policytool*
  - Software gráfico para gerenciar o arquivo *java.policy*
- Permissões além dos Arquivos de Política de Segurança
  - Aplicações tem garantia de permissão para o código
  - Permissões a classes

Domínios de Proteção – Arquivo java.policy

- Passaremos agora para o NetBeans



Sumário

- Modelo Sandbox
- Componentes principais
  - Carregador de Classes
  - Verificador de Bytecodes
  - Gerenciador de Segurança
- Elementos da Sandbox Java
- Domínios de Proteção

Parceiros

- **Os seguintes parceiros tornaram JEDI™ possível em Língua Portuguesa:**