UNIVERSIDADE FEDERAL DO ABC

TUTORIAL PARA IMPLANTAÇÃO DE UMA APLICAÇÃO JAVA WEB EM UM SERVIDOR REMOTO

Orientador Prof. Dr. André Guilherme Ribeiro Balan

UNIVERSIDADE FEDERAL DO ABC

Santo André, São Paulo
12 de Janeiro de 2015

Conteúdo

Capítulo 1

Introdução

O servidor no qual essa aplicação vai rodar é o Debian Wheezy Server, sem interface gráfica. Nesse servidor estão instalados:o servidor de SSH, o MySQL 5.6 e o servidor de aplicações JBoss 7.1. A aplicação foi desenvolvida em Java Web utilizando a IDE Netbeans.

1.1 Servidor SSH

Um servidor SSH utiliza o protocolo **secure shell**, que é um protocolo para realização de login remoto seguro e outros serviços seguros de rede através de uma rede insegura. [2] Esse protocolo possui três componentes principais:

- O Protocolo da camada de transporte [SSH-TRANS]: fornece autenticação, confidencialidade e integridade do servidor e opcionalmente compressão. Normalmente, a camada de transporte será executada através de uma conexão TCP/IP, mas também pode ser usada em qualquer outro fluxo de dados confiável.
- O protocolo de autenticação de usuário [SSH-USERAUTH]: autentica o lado do cliente (client-side) do usuário para o servidor. É executado sobre o protocolo da camada de transporte.
- O protocolo de conexão [SSH-CONNECT]: faz a multiplexação de túnel criptografado em vários canais lógicos. Esse protocolo roda sobre o protocolo de autenticação do usuário.

O cliente envia uma solicitação de algum serviço após o estabelecimento de uma conexão de camada de transporte segura foi estabelecida. Um segunda solicitação de serviço é enviada após a autenticação do usuário ser completada. Isso permite que novos protocolos sejam definidos e coexistam com os protocolos apresentados acima. [2]

Capítulo 2

Passos para implantação da aplicação no servidor

Nesse capítulo serão apresentados todos os passos para a implantação de uma aplicação web, juntamente com sua base de dados, em um servidor remoto que faz autenticação via SSH.

2.1 Exportação do banco de dados MySQL

Para que o programa rode no servidor remoto com o banco de dados atualizado (com as mesmas informações do banco armazenado onde a aplicação está rodando atualmente) é necessário exportar esse banco para um arquivo .sql que será importado em um banco dentro do servidor. Isso é feito através do comando:

```
mysqldump -u [USERNAME] -p [DATABASE-NAME] > database.sql
```

Onde [USERNAME] é o nome de usuário do servidor MySQL que tem acesso à base de dados que se deseja exportar, [DATABASE-NAME] é o nome da base de dados e database é o nome do arquivo .sql que conterá as informações do banco. Após a inserção desse comando, será requisitada a senha do usuário.

2.2 Ajustes na configuração da aplicação

No arquivo de configuração do Hibernate (hibernate.cfg.xml), nas seguintes linhas:

```
1 <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/
     NomeBanco?autoReconnect=true</property>
2 <property name="hibernate.connection.username">usuario</property>
3 <property name="hibernate.connection.password">senha</property>
```

É necessário deixar o **NomeBanco** igual ao banco de dados que será criado no servidor para armazenar as informações da aplicação, bem como o nome de **usuário** e a **senha** do usuário que será criado no servidor MySQL do servidor remoto para fazer o acesso ao banco. Por questões de segurança, não é recomendado utilizar o usuário root.

Caso tenha sido criado um domínio personalizado para a aplicação no servidor (diferente do endereço web padrão definido

2.3 Criação de um arquivo .war da aplicação em Java EE

Um arquivo WAR (Web Application Archive) é um arquivo JAR usado para distribuir recursos como JavaServer Pages, Servlets Java, classes Java, arquivos XML, entre outros, que constituem uma aplicação web.

No Netbeans, para gerar esse arquivo, basta clicar com o botão direito no nome do projeto → Limpar e construir. O arquivo será gerado na pasta **dist** do projeto.

2.4 Criação de um par de chaves RSA

A autenticação no servidor via SSH será feita através de um par de chaves RSA, por ser um método mais seguro do que fazer o login por meio de um usuário e uma senha.

O RSA é um sistema criptográfico de chave pública desenvolvido em 1977 pelos professores do MIT Ronald **R**ivest e Adi **S**hamir e e pelo professor do USC Leonard **A**dleman. Esse algoritmo gera um par de chaves pública e privada através de números primos grandes, sendo intratável obter a chave privada a partir da chave pública. O RSA, assim como os demais algoritmos de chave pública, utiliza uma função unidirecional com segredo, com a qual é tratável calcular y, sendo $y = f(x)$, mas intratável o cálculo de $x = f^{-1}(y)$, a menos que se conheça o segredo. Assim, é permitido que a chave pública seja de conhecimento de qualquer pessoa, pois somente o proprietário do par de chaves, que conhece o segredo, pode calcular a função em ambas as direções, cifrando e decifrando, enquanto os demais apenas conseguirão calcular a função de ciframento. [1]

Para gerar o par de chaves foi utilizado o **puttygen** que é uma ferramenta para gerar e manipular pares de chave SSH pública e privada. Ele é parte do **PuTTY**, um dos programas de cliente mais populares de SSH. Os passos necessários são descritos abaixo [1]:

1. Instalar o PuTTY, através do comando:

   ```
   sudo apt-get install putty
   ```

2. Gerar o par de chaves SSH-2 RSA e salvá-lo no formato do Putty:

   ```
   puttygen -t rsa -C "my home key-o mykey.ppk
   ```

   Onde o que está entre aspas ("**my home key**") é um comentário que identifica o par de chaves e o arquivo **mykey.ppk** é o arquivo salvo com o par de chaves. Tanto o comentário quanto o nome do arquivo (mantendo a extensão .ppk) podem ser modificados.

   Esse comando pede a inserção de uma senha (segredo) e dessa forma é gerado um arquivo .ppk que pode ser lido com um editor de texto. Esse arquivo possui um cabeçalho contendo

[1] http://linux.die.net/man/1/puttygen

o tipo de criptografia utilizado, o comentário que identifica o par de chave e as linhas pública e privada do par.

3. Converter a chave para o formato OpenSSH de chave privada:

```
puttygen mykey.ppk -O private-openssh -o my-openssh-key
```

Sendo **mykey.ppk** o arquivo gerado anterioremente e **my-openssh-key** o nome que deverá ser dado à chave privada no formato OpenSSH que será criada. Essa conversão é necessária para a autenticação do usuário no servidor.

2.5 Autenticação no servidor

Após a geração do par de chaves, é necessário a criação de um usuário no lado do servidor e a atribuição da chave pública do par gerado anteriormente para esse usuário. Feito isso, para fazer a autenticação no servidor é necessário abrir um terminal no lado do cliente, navegar até a pasta onde está salvo o arquivo de chave privada (**my-openssh-key**) gerado anteriormente e digitar o seguinte comando:

```
ssh -i my-openssh-key username@IPServer -p PNumber
```

- **my-openssh-key**: chave privada no formato OpenSSH
- **username**: nome de usuário registrado no servidor
- **IPServer**: número de IP do servidor
- **PNumber**: número da porta do servidor de SSH

Após a execução desse comando, será pedida a senha da chave privada, que deve ser digitada em até 10 segundos.

2.6 Tranferência do banco de dados para o servidor

É necessário transferir o arquivo .sql gerado pela exportação do banco para o servidor no qual a aplicação será implantada. Isso será feito de maneira segura através do PSCP (**P**utty **S**ecure **C**o**P**y). Para isso, é necessário salvar o arquivo com o par de chaves .ppk e o arquivo com o backup do banco (.sql) em uma mesma pasta, navegar até essa pasta através do terminal e digitar o seguinte comando e informar a senha de chave privada:

```
pscp -i mykey.ppk -P PNumber database.sql username@IPServer:/home/username
```

Isso irá transferir uma cópia do arquivo database.sql para a pasta home do usuário no servidor.

2.6.1 Importação do banco de dados

Feita a transferência do arquivo de *backup*, é necessário fazer a autenticação no servidor como mostrado na sessão 2.5. Após a autenticação no servidor, é possível verificar se o arquivo foi transferido corretamente com o comando **ls** na pasta do usuário. Para importar o arquivo .sql em um banco de dados, primeiramente deve-se criar esse banco no servidor MySQL. Isso será feito através do usuário root. É necessário que esse banco de dados tenha o mesmo nome do banco indicado nas configurações da aplicação (2.2):

```
mysql -u root -p
Enter password: *****
create database NomeBanco;
```

A importação do banco é feita através do comando:

```
source /home/username/database.sql;
```

2.6.2 Criação de um usuário do banco de dados

Como foi mencionado, por questões de segurança, não é recomendado que a aplicação tenha acesso ao banco como usuário root. Para criar um novo usuário para o banco de dados, o seguinte comando é utilizado:

```
create user 'usuario'@'localhost' identified by 'senha';
```

Os campos **usuario** e **senha** devem ser iguais aos informados nas configurações da aplicação. (2.2).

Também devem ser configurados quais privilégios esse usuário tem (a quais bancos ele pode ter acesso e que tipo de acesso: leitura, escrita, leitura e escrita, etc). Para essa aplicação, é suficiente que ele tenha acesso ao banco da aplicação e todas suas tabelas e que ele possa realizar as operações de criar, deletar, inserir e atualizar as tabelas: [2]

```
grant create, drop, delete, insert, select, update on NomeBanco.* to 'usuario'@'localhost';
```

2.7 Tranferência da aplicação para o servidor

Para transferir o arquivo .war para o servidor, primeiro é necessário efetuar o logout e depois realizar os mesmos procedimentos descritos para a transferência do arquivo .sql:

[2] https://www.digitalocean.com/community/tutorials/how-to-create-a-new-user-and-grant-permissions-in-mysql

```
pscp -i mykey.ppk -P PNumber Aplicacao.war username@IPServer:/home/username
```

Após a transferência, deve ser feita a autenticação no servidor (2.5). Uma vez logado no servidor, é necessária a troca do usuário para appserver, utilizando o seguinte comando e informando a senha do appserver:

```
su appserver
```

Feita a troca do usuário, o arquivo .war deve ser movido para a pasta de deployments do Jboss:

```
cp Aplicacao.war /usr/local/share/jboss/standalone/deployments/;
```

Dessa forma, a aplicação já estará rodando no endereço da web destinado a ela.

Bibliografia

[1] Samáris Ramiro Pereira. *O sistema criptográfico de chave pública rsa.* PhD thesis, UNIVERSIDADE CATÓLICA DE SANTOS, 2008.

[2] Tatu Ylonen and Chris Lonvick. The secure shell (ssh) protocol architecture. 2006.