Ciências ULisboa | Informática

di.ciencias.ulisboa.pt

/difcul

Segurança e Confiabilidade

2016/2017

Trabalho 1

1 Objectivos

A parte prática da disciplina de Segurança e Confiabilidade pretende familiarizar os alunos com alguns dos problemas envolvidos na programação de aplicações distribuídas seguras, nomeadamente a gestão de chaves criptográficas, a geração de sínteses seguras, cifras e assinaturas digitais, e a utilização de canais seguros à base do protocolo TLS. O trabalho será realizado utilizando a linguagem de programação Java e a API de segurança do Java.

A primeira fase do projecto tem como objectivo fundamental a construção de uma aplicação distribuída básica a ser executada numa *sandbox*. O trabalho consiste na concretização de um sistema **simplificado** de controlo de versões de ficheiros, designado por **myGit**, onde diversos clientes utilizam um servidor central para armazenar as versões mais recentes dos seus ficheiros e partilhá-los com outros clientes.

Na segunda fase do projecto serão adicionadas várias funcionalidades de segurança. E finalmente na terceira fase do projecto serão configurados mecanismos de segurança ao nível do servidor: *firewall* e detecção de intrusões.

2 Arquitectura do Sistema

O trabalho consiste no desenvolvimento de dois programas:

- O servidor *myGitServer*, e
- A aplicação cliente *myGit* que acede ao servidor via *sockets* TCP.

A aplicação é distribuída de forma que o servidor fica numa máquina e um número não limitado de clientes podem ser executados em máquinas diferentes na Internet.

3 Funcionalidades

O sistema tem os seguintes requisitos:

1. O servidor recebe na linha de comandos a seguinte informação:
   - Porto (TCP) para aceitar ligações de clientes.
2. O cliente pode ser utilizado com as seguintes opções:

myGit -init <rep_name>

myGit <localUser> <serverAddress> [-p <password>]

myGit <localUser> <serverAddress> [-p <password>] -push <rep_name>

myGit <localUser> <serverAddress> [-p <password>] -push <file_name>

myGit <localUser> <serverAddress> [-p <password>] -pull <file_name>

myGit <localUser> <serverAddress> [-p <password>] -pull <rep_name>

myGit <localUser> <serverAddress> [-p <password>] -share <rep_name> <userId>

myGit <localUser> <serverAddress> [-p <password>] -remove <rep_name> <userId>

Em que:

- localUser identifica o utilizador local. Caso o utilizador não esteja registado no servidor, efetua o seu registo, ou seja, adiciona este utilizador ao ficheiro das *passwords*.
- serverAddress identifica o servidor (*hostname* ou endereço IP e porto; por exemplo 127.0.0.1:23456),
- -p <password> - *password* utilizada para autenticar o utilizador local. Caso a *password* não seja dada na linha de comando, deve ser pedida posteriormente ao utilizador. Obs: esta opção pretende facilitar a fase de desenvolvimento da aplicação.
- -init – cria o repositório localmente (equivale a criar uma diretoria). Os alunos devem definir a localização no sistema de ficheiros para os repositórios.
- <localUser> <serverAddress> [-p <password>] – criar um novo utilizador, caso ainda não exista.
- -push <file_name> - caso a versão local do ficheiro seja mais recente do que a do servidor, envia-a para o servidor. O servidor mantém várias versões do mesmo ficheiro. Para distinguirem as várias versões do mesmo ficheiro, os alunos podem optar pela nomenclatura que preferirem.
- -push <rep_name> - envia para o servidor todos os ficheiros do repositório cuja versão local seja mais recente do que a do servidor. Ficheiros novos no repositório local (que não existem no servidor) também devem ser enviados. Notar que caso o utilizador tenha apagado um ficheiro do repositório local, este também deve ser "**eliminado"** do repositório no servidor, não existindo assim uma versão atual, mas mantendo, no entanto, as anteriores versões. Ver exemplo de utilização.
- -pull <file_name> - caso a versão local do ficheiro seja menos recente do que a do servidor, recebe-a do servidor.
- -pull <rep_name> - recebe do servidor todos os ficheiros cuja versão local seja menos recente do que a do servidor. Ficheiros inexistentes localmente também devem ser copiados do servidor. Notar que se no repositório local existirem ficheiros que não estão presentes no servidor, o programa cliente deve **apenas avisar** que existem esses ficheiros através da sua listagem dos seus nomes no ecrã.
- -share <rep_name> <userId> - permite ao utilizador partilhar o repositório com outro utilizador. Os repositórios só podem ser partilhados pelos utilizadores que os criaram.
- -remove <rep_name> <userId> - permite ao utilizador (que criou o repositório) retirar o acesso previamente dado a outro utilizador.

O servidor mantém um ficheiro com os utilizadores do sistema e respetivas *passwords*. Este ficheiro deve ser um **ficheiro de texto**. Cada linha tem um *user* e uma *password* separados pelo caracter dois pontos.

O **servidor deve correr numa *sandbox*** que limite o seu acesso à rede e ao sistema de ficheiros.

- O *myGitServer* pode esperar e aceitar receber ligações de clientes a partir de qualquer lado, no porto 23456;
- O *myGitServer* pode ler e escrever ficheiros do seu repositório.

O **cliente também deve correr numa *sandbox***. Para além disso, o grupo pode adicionar outras políticas que julgue necessárias para o correto funcionamento do sistema.

4 Exemplo de utilização

Obs: as mensagens são exemplificativas assim como a forma de distinguir versões dos ficheiros.

```
seg000@gcc:~$ java myGit -init myrep
-- O repositório myrep foi criado localmente
seg000@gcc:~$ ls myrep
myrep
seg000@gcc:~$ cp myGit.java myrep
seg000@gcc:~$ java myGit maria 127.0.0.1:23456 -p badpwd -push myrep
-- O utilizador maria vai ser criado
Confirmar password do utilizador maria
bapwd
-- O utilizador maria foi criado
-- O repositório myrep foi criado no servidor
-- O ficheiro myGit.java foi enviado para o servidor
seg000@gcc:~$ java myGit maria 127.0.0.1:23456 -p badpwd -share myrep pedro
Erro: O utilizador pedro não existe
seg000@gcc:~$ java myGit pedro 127.0.0.1:23456 -p badpwd1
-- O utilizador pedro vai ser criado
Confirmar password do utilizador pedro
bapwd1
-- O utilizador pedro foi criado
seg000@gcc:~$ java myGit maria 127.0.0.1:23456 -p badpwd -share myrep pedro
-- O repositório myrep foi partilhado com o utilizador pedro
seg000@gcc:~$ java myGit pedro 127.0.0.1:23456 -p badpwd1 -pull maria/myrep
-- O repositório myrep do utilizador maria foi copiado do servidor
seg000@gcc:~$ echo lixo >> myrep/myGit.java
seg000@gcc:~$ java myGit pedro 127.0.0.1:23456 -p badpwd1 -push maria/myrep/myGit.java
-- O ficheiro myGit.java foi enviado para o servidor
seg000@gcc:~$ ls -l myrep/myGit.java   #repositório local da maria
-rw-r----- 1 seg seg 420 Feb 17 14:42 myGit.java
seg000@gcc:~$ java myGit maria 127.0.0.1:23456 -p badpwd -pull myrep/myGit.java
-- O ficheiro myGit.java foi copiado do servidor
seg000@gcc:~$ ls -l myrep/myGit.java   #repositório local da maria
-rw-r----- 1 seg seg 425 Feb 17 14:52 myGit.java
```

```
No servidor:
seg000@gcc:~$ ls -l myrep/myGit*
-rw-r----- 1 seg seg 420 Feb 17 14:42 myGit.java.1
-rw-r----- 1 seg seg 425 Feb 17 14:52 myGit.java
```

```
seg000@gcc:~$ rm myrep/myGit.java
seg000@gcc:~$ java myGit maria 127.0.0.1:23456 -p badpwd -push myrep
-- O ficheiro myGit.java vai ser eliminado no servidor
```

```
No servidor:
seg000@gcc:~$ ls -l myrep/myGit*
-rw-r----- 1 seg seg 420 Feb 17 14:42 myGit.java.1
-rw-r----- 1 seg seg 425 Feb 17 14:52 myGit.java.2
```

```
seg000@gcc:~$ java myGit pedro 127.0.0.1:23456 -p badpwd1 -pull maria/myrep
-- O ficheiro myGit.java existe localmente mas foi eliminado no servidor
```

5 Relatório e discussão

Além do conteúdo habitual de um relatório (tal como a identificação da disciplina, dos elementos do grupo, dos **objetivos concretizados com êxito e os que não foram**, etc), devem ser apresentados e discutidos os pontos fundamentais do projeto:

- Explicar a configuração da ***sandbox* para execução do servidor e do cliente**;
- Explicar a organização do software cliente e servidor, por exemplo em termos de classes e *threads*;
- Explicar as mensagens trocadas entre o cliente e o servidor e seu formato;
- Identificar os **requisitos de segurança** que se deveria garantir na aplicação e **indicar os mecanismos de segurança** que deveriam ser utilizados de modo a satisfazer esses requisitos.

O relatório deve ter no máximo 5 páginas (sem contar com o código) e **é necessário incluir o código fonte.**

6 Entrega

Código.

Dia **19 de Março**, até as 23:55 horas. O código do trabalho deve ser entregue da seguinte forma:

Os grupos devem inscrever-se atempadamente de acordo com as regras afixadas para o efeito, na página da disciplina.

Na página da disciplina submeter o código do trabalho num ficheiro zip e um readme (txt) sobre como executar o trabalho.

Relatório.

Dia **20 de Março**, até as 18:00 horas. A entrega será em papel, no **cacifo do professor** das TPs e em **pdf na página da disciplina**.

Não serão aceites trabalhos por email nem por qualquer outro meio não definido nesta secção. Se não se verificar algum destes requisitos o trabalho é considerado não entregue.