Ciências ULisboa | Informática

di.ciencias.ulisboa.pt

/difcul

Segurança e Confiabilidade

2017/2018

Trabalho 1

1 Objetivos

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 projeto a desenvolver na disciplina será realizado utilizando a linguagem de programação Java e a API de segurança do Java, e é composto por três fases.

A primeira fase do projeto tem como objetivo fundamental a construção de uma aplicação distribuída básica a ser executada numa *sandbox*. O trabalho consiste na concretização de um sistema de partilha de fotos, ***PhotoShare***, onde diversos clientes utilizam um servidor central para partilharem/armazenarem fotos, comentários e like/dislike das fotos. Cada utilizador partilha as suas fotos e respetivos comentários com os seus seguidores e pode comentar as fotos dos utilizadores que segue, bem como colocar like/dislike nas fotos deles.

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

2 Arquitetura do Sistema

O trabalho consiste no desenvolvimento de dois programas:

- O servidor *PhotoShareServer*, e
- A aplicação cliente *PhotoShare* 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:

    PhotoShare <localUserId> <password> <serverAddress>

    [-a <photos> | -l <userId> | -i <userId> <photo> | -g <userId> |

    -c <comment> <userId> <photo> | -L <userId> <photo> |

    -D <userId> <photo> | -f <followUserIds> | -r <followUserIds>]

Em que:

- <localUserId> identifica este utilizador local. Caso o utilizador não esteja registado no servidor, efetua o seu registo, ou seja, adiciona este utilizador ao ficheiro das passwords.
- <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.
- <serverAddress> identifica o servidor (*hostname* ou endereço IP e porto; por exemplo 127.0.0.1:23456).
- <localUserId> <password> <serverAddress> – criar um novo utilizador, caso ainda não exista.
- -a <photos> - adiciona/copia estas fotos para o servidor. Caso este utilizador já tenha alguma foto com o mesmo nome no servidor, o cliente deve retornar um erro.
- -l <userId> - se o utilizador local fizer parte dos seguidores de *userId*, lista as fotografias do utilizador *userId* indicando o nome da foto e a data de publicação; caso contrário, devolve um erro. Assume-se que se o *userId* for o utilizador local *(localUserId)*, deve-se mostrar a informação do utilizador local.
- -i <userId> <photo> - se o utilizador local fizer parte dos seguidores de *userId*, devolve os comentários e o número de likes e dislikes da fotografia *photo* especificada; caso contrário, devolve um erro. Assume-se que se o *userId* for o utilizador local *(localUserId)*, deve-se mostrar a informação da foto do utilizador local.
- -g <userId> - se o utilizador local fizer parte dos seguidores de *userId*, copia do servidor para o cliente todas as fotos do utilizador *userId* (e os respetivos comentários); caso contrário, devolve um erro.
- -c <comment> <userId> <photo> - se o utilizador local fizer parte dos seguidores de *userId*, adiciona um comentário à fotografia *photo* do utilizador *userId*; caso contrário, devolve um erro.
- -L <userId> <photo> - se o utilizador local fizer parte dos seguidores de *userId*, adiciona um like à fotografia *photo* do utilizador *userId*; caso contrário, devolve um erro;
- -D <userId> <photo> - se o utilizador local fizer parte dos seguidores de *userId*, adiciona um dislike à fotografia *photo* do utilizador *userId*; caso contrário, devolve um erro;
- -f <followUserIds> - adiciona os utilizadores *followUserIds* como seguidores do utilizador local. Se algum dos utilizadores já fizer parte da lista de seguidores deve ser devolvido um erro.
- -r <followUserIds> - remove os utilizadores *followUserIds* como seguidores do utilizador local. Se algum dos utilizadores não fizer parte da lista de seguidores deve ser devolvido um erro.

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 *PhotoShareServer* pode esperar e aceitar receber ligações de clientes a partir de qualquer lado, no porto 23232;

- O *PhotoShareServer* 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 julguem necessárias para o correto funcionamento do sistema.

4 Relatório e discussão

Além do conteúdo habitual de um relatório (tal como a identificação da disciplina, dos elementos do grupo, etc), devem ser apresentados e discutidos os pontos fundamentais do projeto:

1. Indicação dos objetivos **concretizados com êxito e os que não foram.**
2. Explicar a configuração da ***sandbox*** para execução do **servidor** e do **cliente**;
3. Explicar brevemente a organização do software cliente e servidor, por exemplo em termos de classes e *threads*;
4. Explicar brevemente as mensagens trocadas entre o cliente e o servidor e seu formato;
5. 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.**

5 Entrega

- **Código.**

  Dia **23 de Março**, até as 23:59 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 área da disciplina submeter o código do trabalho num ficheiro zip e um readme (txt) sobre como executar o projeto.

- **Relatório.**

  Dia **26 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.