Ciências ULisboa | Informática

di.ciencias.ulisboa.pt

/difcul

Segurança e Confiabilidade

2020/2021

Trabalho 1 – 1ª fase

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

A primeira fase do projeto tem como objetivo fundamental a construção de uma aplicação distribuída a ser executada numa *sandbox*, focando-se essencialmente nas funcionalidades da aplicação. O trabalho consiste na concretização do sistema ***SeiTchiz*** uma versão portuguesa do *Instagram*), que é um sistema do tipo cliente-servidor que permite que os utilizadores (clientes) utilizem um servidor central para partilhar fotografias e comunicar com outros utilizadores. O sistema suporta dois modos de funcionamento. No modo **mural** (*feed*), os utilizadores colocam fotografias no seu perfil público guardado no servidor e podem seguir quaisquer outros utilizadores, vendo no seu mural as fotografias que estes publicaram. O serviço também permite a colocação de *likes* nestas fotografias. No modo **conversação**, podem enviar mensagens para grupos privados de utilizadores e ler as mensagens enviadas para os grupos a que pertencem. Cada utilizador possui uma conta no servidor, pode seguir qualquer outro utilizador livremente, e pode pertencer a vários grupos de acesso privado.

Na segunda fase do projeto serão adicionadas várias funcionalidades de segurança. O enunciado desta segunda fase será oportunamente divulgado.

2 Arquitetura do Sistema

O trabalho consiste no desenvolvimento de dois programas:

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

A aplicação é distribuída, de forma a que o servidor execute numa máquina e um número ilimitado de clientes possam 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:

   SeiTchizServer 45678

   - 45678 identifica o porto (TCP) para aceitar ligações de clientes.

2. O cliente pode ser utilizado com as seguintes opções:

SeiTchiz <serverAddress> <clientID> [password]

Em que:

- <serverAddress> identifica o servidor (*hostname* ou endereço IP e porto; por exemplo 1.2.3.4). Por omissão, a conexão é no porto 45678 (porto onde escuta o servidor).
- <clientID> identifica o utilizador local (cliente). Caso o utilizador não esteja registado no servidor, efetua o seu registo, ou seja, adiciona este *clientID* e a respetiva password ao ficheiro de utilizadores do servidor.
- [password] – password utilizada para autenticar o cliente. Caso a password não seja dada na linha de comando, deve ser pedida ao utilizador pela aplicação.

Se o cliente não se conseguir autenticar no servidor (i.e., se o par *clientID/password* enviados não corresponderem aos que estão guardados no ficheiro de utilizadores do servidor), a aplicação deverá terminar, assinalando o erro correspondente. Caso contrário, a aplicação deverá apresentar um menu disponibilizando os seguintes comandos. Qualquer comando pode ser invocado por extenso ou pela letra que está a negrito (por exemplo, ***follow*** ou ***f***):

- **f**ollow <userID> – adiciona o cliente (clientID) à lista de seguidores do utilizador *userID*. Se o cliente já for seguidor de *userID* ou se o utilizador *userID* não existir deve ser assinalado um erro.
- **u**nfollow <userID> - remove o cliente (clientID) da lista de seguidores do utilizador *userID*. Se o cliente não for seguidor de *userID* ou se o utilizador *userID* não existir deve ser assinalado um erro.
- **v**iewfollowers – obtém a lista de seguidores do cliente ou assinala um erro caso o cliente não tenha seguidores.
- **p**ost <photo> – envia uma fotografia (*photo*) para o perfil do cliente armazenado no servidor. Assume-se que o conteúdo do ficheiro enviado é válido, ou seja, uma fotografia (não é necessário fazer nenhuma verificação).
- **w**all <nPhotos> - recebe as *nPhotos* fotografias mais recentes que se encontram nos perfis dos utilizadores seguidos, bem como o número de *likes* de cada fotografia (mostrando tudo no mural). Se existirem menos de *nPhotos* disponíveis, recebe as que existirem (ou assinala um erro se não existir nenhuma).
- **l**ike <photoID> – coloca um *like* na fotografia *photoID*. Todas as fotografias têm um identificador único atribuído pelo servidor. Os clientes obtêm os identificadores das fotografias através do comando **w**all. Se a fotografia não existir deve ser assinalado um erro.
- **n**ewgroup <groupID> – cria um grupo privado, cujo dono (*owner*) será o cliente que o criou. Se o grupo já existir assinala um erro.
- **a**ddu <userID> <groupID> – adiciona o utilizador *userID* como membro do grupo indicado. Se *userID* já pertencer ao grupo ou se o grupo não existir deve ser assinalado um erro. Apenas os donos dos grupos podem adicionar utilizadores aos seus grupos, pelo que deverá ser assinalado um erro caso o cliente não seja dono do grupo.
- **r**emoveu <userID> <groupID> – remove o utilizador *userID* do grupo indicado. Se *userID* não pertencer ao grupo ou o grupo não existir deve ser assinalado um erro. Apenas os donos dos grupos podem remover utilizadores dos seus grupos, pelo que deverá ser

assinalado um erro caso o cliente não seja dono do grupo.

- **g**info [groupID] – se *groupID* não for especificado, mostra uma lista dos grupos de que o cliente é dono, e uma lista dos grupos a que pertence. Caso não seja dono de nenhum grupo ou não seja membro de nenhum grupo, esses factos deverão ser assinalados. Se for especificado o *groupID,* mostra o dono do grupo e uma lista dos membros do grupo. Se o *groupID* especificado não existir ou se o cliente não for dono nem membro do grupo, deve ser assinalado um erro.
- **m**sg <groupID> <msg> – envia uma mensagem (*msg*) para o grupo *groupID*, que ficará guardada numa caixa de mensagens do grupo, no servidor. A mensagem ficará acessível aos membros do grupo através do comando **c**ollect. Se o grupo não existir ou o cliente não pertencer ao grupo deve ser assinalado um erro.
- **c**ollect <groupID> – recebe todas as mensagens que tenham sigo enviadas para o grupo *groupID* e que o cliente ainda não tenha recebido. Por exemplo, se a caixa de mensagens do grupo tem 3 mensagens (m1, m2, m3), se o utilizador u1 já recebeu as mensagens m1 e m2 e o utilizador u2 ainda não recebeu nenhuma, então a execução do comando pelo utilizador u1 retornará apenas m3, mas se for executado pelo utilizador u2 retornará as 3 mensagens. Se não existir nenhuma nova mensagem, esse facto deverá ser assinalado. Os utilizadores apenas têm acesso às mensagens enviadas depois da sua entrada no grupo. Quando uma mensagem é lida por todos os utilizadores, esta é removida da caixa de mensagens e colocada num histórico do grupo. Os donos dos grupos contam como membros para efeito da remoção de mensagens da caixa de mensagens, ou seja, os donos também recebem as mensagens que eles próprios enviaram. Se o grupo não existir ou o cliente não pertencer ao grupo deve ser assinalado um erro.
- **h**istory <groupID> – mostra o histórico das mensagens do grupo indicado que o cliente já leu anteriormente. Se o grupo não existir ou o cliente não pertencer ao grupo deve ser assinalado 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*, o nome do user, e uma *password* separados pelo caracter dois pontos (<user>:<nome user>:<password>). As mensagens e as fotografias também têm de ser guardadas em disco no servidor.

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

- O *SeiTchizServer* pode esperar e aceitar receber ligações de clientes a partir de qualquer lado, no porto 45678;
- O *SeiTchizServer* 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 Entrega

Dia **05 de março**, até às 23:55 horas. O código desta primeira fase do trabalho deve ser entregue de acordo com as seguintes regras:

- O trabalho é realizado em grupos de 3 alunos e os grupos têm de ser formados atempadamente, com alunos pertencentes à mesma turma TP. O corpo docente reserva-se o direito de agrupar alunos que até ao prazo não tenham grupo completo.
- A formação de grupos é feita na página da disciplina no Moodle, até ao dia 19 de fevereiro

(os alunos podem dirigir-se à secretaria para pedir a mudança de turma TP, sendo a mudança apenas possível se a turma tiver vagas). Devem também informar o corpo docente da mudança para que esta seja também feita no Moodle.

- Para entregar o trabalho, é necessário criar um **ficheiro zip** com o nome **SegC-grupoXX-proj1.zip**, onde **XX** é o número do grupo, contendo:
  - ✓ o código do trabalho;
  - ✓ os ficheiros jar (cliente e servidor) para execução do projeto;
  - ✓ um readme (txt) sobre **como compilar** o projeto e **executá-lo** com os ficheiros de permissões (cliente e servidor) e as limitações do trabalho.
- O ficheiro zip é submetido através da atividade disponibilizada para esse efeito na página da disciplina no Moodle. Apenas um dos elementos do grupo deve submeter. Se existirem várias submissões do mesmo grupo, será considerada a mais recente.

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