Ciências ULisboa | Informática

di.ciencias.ulisboa.pt
/difcul

Segurança e Confiabilidade

2015/2016

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 SSL. 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 de troca de mensagens de texto, fotos e vídeos, **myWhats**, onde diversos clientes utilizam um servidor central para trocarem entre si mensagens de texto, fotos e vídeos. Cada utilizador pode enviar conteúdos para um dos seus contatos ou para grupos criados previamente.

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 *myWhatsServer*, e
- A aplicação cliente *myWhats* 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:

myWhats <localUser> <serverAddress> [-p <password>] [-m <contact> <message> | -f <contact> <file> | -r contact file | -a <user> <group> | -d <user> <group>]

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.

- -m <contact> <message> - envia uma mensagem para o servidor para a partilhar com o *contact*. Um *contact* pode ser um utilizador ou um grupo de utilizadores. Caso o *contact* não exista, devolve erro.
- -f < *contact* > <file> - envia o ficheiro file para o servidor para o partilhar com o *contact*. Um *contact* pode ser um utilizador ou um grupo de utilizadores. Caso o *contact* ou o *file* não existam, devolve erro.
- -r - recebe do servidor a informação sobre a comunicação mais recente trocada com cada um dos seus *contacts*:
    - nome do *contact*: entidade (grupo ou utilizador) parceiro da comunicação;
    - nome do utilizador que deu origem à comunicação (quem enviou a mensagem ou o ficheiro para contact, pode ser o próprio);
    - data e hora da comunicação e
    - mensagem ou nome de ficheiro trocado.
- -r *contact* - recebe do servidor a informação sobre **todas** as comunicações realizadas de/para este *contact* (nome do utilizador que deu origem à comunicação, data e hora da comunicação e mensagem ou nome de ficheiro trocado). Caso o *contact* não exista, devolve erro.
- -r *contact* file - recebe do servidor o conteúdo do ficheiro *file*. Caso o *contact* ou o *file* não existam, devolve erro.
- -a <*user*> <group> - adiciona o utilizador *user* ao grupo. Apenas quem criou o grupo pode adicionar utilizadores. Caso o grupo não exista, cria o grupo. Caso o *user* não exista, devolve erro.
- -d <*user*> <group> - remove o utilizador *user* do grupo. Apenas quem criou o grupo pode remover utilizadores. Caso o grupo ou o *user* não existam, devolve erro. Caso o *user* seja o utilizador que criou o grupo, o grupo deve ser removido assim como todas as mensagens e ficheiros que foram trocados no contexto deste grupo.

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. Este ficheiro **não deve** ser usado para gerir os grupos. Cada mensagem e cada ficheiro devem ser armazenados no servidor separadamente.

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

- O *myWhatsServer* pode esperar e aceitar receber ligações de clientes a partir de qualquer lado, no porto 23456;
- O *myWhatsServer* 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

```
seg000@gcc:~$ myWhats maria 127.0.0.1:23456 -p badpwd -m pedro "olá"
seg000@gcc:~$ myWhats maria 127.0.0.1:23456 -p badpwd -a pedro myfamily
```

```
seg000@gcc:~$ myWhats maria 127.0.0.1:23456 -p badpwd -a sara myfamily
seg000@gcc:~$ myWhats maria 127.0.0.1:23456 -p badpwd -m myfamily "bem vindos"
seg000@gcc:~$ myWhats sara 127.0.0.1:23456 -p badpwdsara -m myfamily "boa bjs"
seg000@gcc:~$ myWhats pedro 127.0.0.1:23456 -p badpwdpedro -f myfamily natal2015.png
seg000@gcc:~$ myWhats maria 127.0.0.1:23456 -p badpwd -r
Contact: Pedro
me: olá
19-02-2016 13:40
Contact: myfamily
pedro: natal2015.png
19-02-2016 14:10
seg000@gcc:~$ myWhats maria 127.0.0.1:23456 -p badpwd -r myfamily
me: bem vindos
19-02-2016 14:00
sara: boa bjs
19-02-2016 14:03
pedro: natal2015.png
19-02-2016 14:10
seg000@gcc:~$ myWhats maria 127.0.0.1:23456 -p badpwd -r myfamily natal2015.png
seg000@gcc:~$ ls
natal2015.png
```

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 da aplicação e indicar os mecanismos de segurança que podem/devem 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 **20 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 página da disciplina submeter o código do trabalho num ficheiro zip, o relatório num ficheiro

pdf e um readme (txt) sobre como executar o trabalho.

Relatório.

Dia **21 de Março**, até as 18:00 horas. A entrega será em papel, no cacifo do professor das TPs.

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.