Ciências ULisboa | Informática

di.ciencias.ulisboa.pt

/difcul

Segurança Informática

2020/2021

Trabalho 1

1 Objectivos

A parte prática da disciplina de Segurança Informática 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 trabalho tem como objetivo fundamental a construção de uma aplicação distribuída básica a ser executada numa *sandbox*. O trabalho consiste na concretização de um sistema **simplificado** de registo de informação de utentes de um dado Hospital, designado por **myDoctor.** Cada utente utiliza um servidor central para aceder à sua informação médica, onde os diversos médicos e técnicos registam o resultado dos diferentes exames (imagens médicas – em formato jpeg – e relatórios médicos – em formato pdf).

Na segunda fase do trabalho serão adicionadas várias funcionalidades de segurança. Finalmente, na terceira fase do trabalho, serão configurados mecanismos de segurança ao nível do servidor, nomeadamente uma *firewall* e um sistema de deteção de intrusões.

2 Arquitectura do Sistema

O trabalho consiste no desenvolvimento de dois programas:

- O servidor *myDoctorServer*, e
- A aplicação cliente *myDoctor* 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.

A aplicação cliente terá diferentes níveis de utilização, dependendo das permissões do utilizador:

- Utente – Permite listar os documentos que estão associados à sua ficha pessoal e fazer o *download* de cada um;
- Técnico – Permite listar os utentes e registar novos documentos associados ao utente. Caso já exista um documento com o mesmo nome no servidor, o novo ficheiro deve ser renomeado de forma a incluir um número sequencial (i.e., se o técnico tentar registar o documento com o nome radiografia.jpeg e este já existir no servidor, o ficheiro deve ser renomeado para radiografia_1.jpeg);
- Médico – Permite listar utentes, listar todos os documentos de um utente, fazer o download de documentos e registar novos documentos (conforme descrito para o técnico).
- Admin – Permite criar utilizadores.

De forma a simplificar o trabalho, os alunos devem considerar que a diretoria raiz do trabalho apenas contém diretorias correspondentes a utentes e estas, por sua vez, só incluem ficheiros.

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.
   - O servidor, quando inicializado, deve verificar se existe o ficheiro de passwords no sistema. Caso não exista deve criar automaticamente um utilizador do tipo Admin com o userId igual a 1, o nome de "Administrador_base", a password deve ser pedida e o tipo de utilizador será admin, de acordo com o seguinte exemplo:

     1; Administrador_base*;badpw;admin*

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

   myDoctor -u <userId> -a <serverAddress> [-p <password> | -mu | -md | -mx <userId2> | -d <file> | -du <file> <userid> | -su <file> <userid> | -c <userId> <nome> <tipo de utilizador>

   Em que:
   - -u <userId> identifica o utilizador.
   - -a <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 *useid*. 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.
   - -mu – listar utilizadores. Esta opção é inválida para utentes;
   - -md – listar documentos do próprio utilizador. Esta opção é válida para utentes.
   - -mx <userId2> – listar documentos do utilizador userId2. Esta opção é válida para médicos.

     Caso não seja possível executar a operação de listar, seja por falta de permissões ou por inexistência de informação a listar, o utilizador deve ser informado do erro.
   - -d <file> – fazer o download de <file> do servidor para a máquina do cliente. Esta opção é válida para utentes.
   - -du <file> <userid> – fazer o download de <file> do utilizador <userid> do servidor para a máquina do cliente. Esta opção é válida para médicos.
   - -su <file> <userId> – fazer o upload de <file> para o servidor, guardando-o no diretório correspondente ao <userId>. Esta opção é válida para técnicos e médicos.

- -c <userId> <nome> <password> <tipo_de_utilizador> – Criar um novo utilizador no sistema. Esta opção é válida para os admins.

  Existem quatro tipos de utilizadores: *admin, utente, medico e tecnico*.

  Caso seja introduzido um <userId> já existente, deve ser devolvida uma mensagem de erro e o programa deve terminar.

  O campo <nome> identifica o nome que se pretende para o utilizador e <password> a *password* de acesso ao sistema por parte deste utilizador.

  Com a criação de um novo utilizador do tipo utente, será criada também uma nova diretoria para esse utente.

O servidor mantém um ficheiro com os utilizadores do sistema e respetivas informações. Este ficheiro deve ser um **ficheiro de texto**. Cada linha tem um userId, um nome, uma *password* (guardada em claro nesta fase do trabalho) e o tipo de utilizador, conforme exemplificado se seguida:

1; Administrador_base*;e1nSns0;admin*

2001;Maria Silva;a1b2c;utente

213;Pedro Sousa;l1g2!?cc;medico

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

- O *myDoctorServer* pode esperar e aceitar ligações de clientes a partir de qualquer lado, no porto **23456**;
- O *myDoctor* pode ler e escrever ficheiros do seu repositório e pode ligar-se ao servidor.

O **cliente também deve ser executado numa *sandbox***.

O grupo pode adicionar outras políticas que julgue necessárias para o correto funcionamento do sistema.

4 Exemplo de utilização

```
Obs: as mensagens são exemplificativas.
$ java myDoctor -u 1 -a 127.0.0.1:23456 -p badpwd -c 201 Jose thepwd utente
O utilizador Jose com o ID 201 vai ser criado
O utilizador Jose foi criado
$ java myDoctor -u 1 -a 127.0.0.1:23456 -p badpwd -c 1001 Maria qqpwd tecnico
O utilizador Maria com o ID 1001 vai ser criado
O utilizador Maria foi criado
$ java myDoctor -u 1 -a 127.0.0.1:23456 -p badpwd -c 11 Pedro aapwd medico
O utilizador Pedro com o ID 11 vai ser criado
O utilizador Pedro foi criado
$ java myDoctor -u 1001 -a 127.0.0.1:23456 -p qqpwd -su radiografia.jpeg 201
O ficheiro radiografia.jpeg foi enviado para o servidor e ficou associado ao utilizador
```

```
Jose com o id 201
$ java myDoctor -u 11 -a 127.0.0.1:23456 -p aapwd -mu
1 Administrador_base
201 Jose utente
1001 Maria tecnico
11 Pedro medico
$ java myDoctor -u 11 -a 127.0.0.1:23456 -p aapwd -su relatório.pdf 201
O ficheiro relatório.pdf foi enviado para o servidor e ficou associado ao utilizador
Jose com o id 201
$ java myDoctor -u 11 -a 127.0.0.1:23456 -p aapwd -mx 201
radiografia.jpeg
relatório.pdf
$ java myDoctor -u 201 -a 127.0.0.1:23456 -p thepwd -md
radiografia.jpeg
relatório.pdf
$ java myDoctor -u 201 -a 127.0.0.1:23456 -p thepwd -d relatório.pdf
O ficheiro relatório.pdf foi recebido pelo cliente.
```

5 Relatório

O relatório deve incluir:

- informação sobre os objetivos concretizados com êxito e os objetivos que não foram concretizados.
- explicar a configuração da ***sandbox*** **para execução do servidor e do cliente**;
- 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 3 páginas**.**

6 Entrega

Código:

Dia **14 de Março**, até às 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 **15 de Março**, até às 12:00 horas, em **pdf na página da disciplina**.

Atenção: O código e o relatório têm atividades de entrega distintas 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.