Relatório do Projeto da Fase 2 de Segurança e Confiabilidade 20/21

Grupo35

Martim Silva 51304

Francisco Freire 52177

David Rodrigues 53307

//---------------------------------------------------------------------------------------------------------------------------

INSTRUÇÕES PARA COMPILAR E CORRER:

* FAZER OS JARS:

1.Compilar ficheiros ATUAIS para a pasta bin

2.right click no package do server ou cliente

3.export -> java -> jarfile

4.verificar se o package correto e o que esta selecionado

5.selecionar a opcao export java source files and resources(3a opcao)

6.selecionar o path na root do projeto e o nome do jar é SeiTchizServer para servidor e SeiTchiz para o cliente

7.Fazer Next e nao fazer nada na pagina que aparece

8.Fazer Next de novo e em "Select the class of the application entry point" fazer browse

e meter a classe que tem o main

* PARA COMPILAR SERVIDOR

javac -d bin src/server/SeiTchizServer.java src/communication/Com.java src/security/Security.java

* PARA COMPILAR CLIENTE

javac -d bin src/client/SeiTchiz.java src/client/ClientStub.java src/communication/Com.java src/security/Security.java

* PARA CORRER SERVIDOR COM POLICIES

java -cp bin -Djava.security.manager -Djava.security.policy==server.policy server.SeiTchizServer 45678 serverKeyStore passserver

* PARA CORRER CLIENTE COM POLICIES

java -cp bin -Djava.security.manager -Djava.security.policy==client.policy client.SeiTchiz localhost ts\_client 1KS passclient1 client1

java -cp bin -Djava.security.manager -Djava.security.policy==client.policy client.SeiTchiz localhost ts\_client 2KS passclient2 client2

java -cp bin -Djava.security.manager -Djava.security.policy==client.policy client.SeiTchiz localhost ts\_client 3KS passclient3 client3

java -cp bin -Djava.security.manager -Djava.security.policy==client.policy client.SeiTchiz localhost ts\_client 4KS passclient4 client4

* PARA CORRER SERVIDOR COM POLICIES POR JAR

java -cp bin -Djava.security.manager -Djava.security.policy==server.policy -jar SeiTchizServer.jar 45678 serverKeyStore passserver

* PARA CORRER CLIENTE COM POLICIES POR JAR

java -cp bin -Djava.security.manager -Djava.security.policy==client.policy -jar SeiTchiz.jar localhost ts\_client 1KS passclient1 client1

java -cp bin -Djava.security.manager -Djava.security.policy==client.policy -jar SeiTchiz.jar localhost ts\_client 2KS passclient2 client2

java -cp bin -Djava.security.manager -Djava.security.policy==client.policy -jar SeiTchiz.jar localhost ts\_client 3KS passclient3 client3

java -cp bin -Djava.security.manager -Djava.security.policy==client.policy -jar SeiTchiz.jar localhost ts\_client 4KS passclient4 client4

//---------------------------------------------------------------------------------------------------------------------------

LIMITAÇÕES DO TRABALHO

O único argumento que deve ser passado obrigatoriamente para o servidor correr e 45678.

O serverAddress passado pelo cliente como argumento pode ser apenas <endereço IP> (por ex:localhost) ou <endereço IP>:<porto 45678> (por ex: localhost:45678).

Username e passwords passados como argumento em SeiTchiz não devem conter espaços nem dois pontos(:) nem hífens(-) nem forward slashes(/).

Nomes de grupos não devem conter espaços nem dois pontos(:) nem hífens(-) nem forward slashes(/).

Mensagens não devem conter dois pontos(:) nem hífens(-).

UserIDs inseridos nos comandos que recebem userIDs não devem conter espaços nem dois pontos(:) nem hífens(-) nem forward slashes(/).

Um user que já esteja logged on nao deve fazer login enquanto a sessão inicial não tenha sido terminada.

groupIDs inseridos nos comandos que recebem userIDs não devem conter espaços nem dois pontos(:) nem hífens(-) nem forward slashes(/).

As fotos de stock que se podem partilhar são apenas as que se encontram no ficheiro Fotos na root do projeto, ou seja o argumento <photo> de post deve ser foto<1 ou 2 ou 3 ou 4>.jpg

A pasta bin não deve ser apagada apenas os seus conteúdos podem ser apagados.

A pasta files pode ser apagada para dar um "restart" do servidor e todos os seus conteúdos

Como não foi dito no enunciado não foi implementado o impedimento de um utilizador dar múltiplos likes a mesma fotografia nem um utilizador poder dar like a sua própria fotografia.

Para interromper o funcionamento de um cliente usar a opção s ou stop

//---------------------------------------------------------------------------------------------------------------------------

CRIAR CHAVES E KEYSTORES E RESPETIVAS PASSWORDS

* Comandos de chaves assimétricas usados

keytool -genkeypair -alias serverKeyStore -keyalg RSA -keysize 2048 -storetype JKS -keystore keystores/serverKeyStore

\*Password correspondente: passserver

keytool -genkeypair -alias 1KS -keyalg RSA -keysize 2048 -storetype JKS -keystore keystores/1KS

\*Password correspondente: passclient1

keytool -genkeypair -alias 2KS -keyalg RSA -keysize 2048 -storetype JKS -keystore keystores/2KS

\*Password correspondente: passclient2

keytool -genkeypair -alias 3KS -keyalg RSA -keysize 2048 -storetype JKS -keystore keystores/3KS

\*Password correspondente: passclient3

keytool -genkeypair -alias 4KS -keyalg RSA -keysize 2048 -storetype JKS -keystore keystores/4KS

\*Password correspondente: passclient4

* Comandos de chaves simétricas usados

keytool -genseckey -alias serverKey -storetype JCEKS -keystore ServerKeyStore

* Comandos de verificar chaves

keytool -list -keystore keystores.serverKeyStore

keytool -list -keystore keystores.1KS

keytool -list -keystore keystores.2KS

keytool -list -keystore keystores.3KS

keytool -list -keystore keystores.4KS

* Tabela de keystore, aliases e passwords a usar

(Server): serverKeyStore serverKeyStore passserver

(client1): 1KS 1KS passclient1

(client2): 2KS 2KS passclient2

(client3): 3KS 3KS passclient3

(client4): 4KS 4KS passclient4

<keystore> <alias> <password>

NOTA

Os userIDs que usámos para cada cliente foram respetivamente client1, client2, client3 e client4.

//---------------------------------------------------------------------------------------------------------------------------

Decisões de desenho das soluções implementadas, nomeadamente em termos de arquitetura de software, de segurança, de desempenho e de funcionalidade