Ciências ULisboa

Engenharia Informática
Segurança e Confiabilidade- 2020/2021

Trabalho 1
Fase 2

Grupo SegC-034

Diogo Pinto - 52763
Francisco Ramalho - 53472
João Funenga - 53504

Como compilar o projeto

Compilar as classes relativas ao servidor:

javac -d bin src/Server/SeiTchizServer.java src/Server/CatalogoClientes.java src/Server/CatalogoGrupos.java src/Server/Cliente.java src/Server/Grupo.java src/Server/Mensagem.java src/Server/Photo.java src/Server/Autenticacao.java

Compilar a classe relativa ao cliente:

javac -d bin src/Client/SeiTchiz.java

Como executar o projeto com os ficheiros de permissões

Exemplo de comandos para execução do servidor:

Executar ficheiro .class:

java -cp bin -Djava.security.manager -Djava.security.policy==server.policy Server.SeiTchizServer 45678 servidor servidor

Executar ficheiro .jar:

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

Exemplo de comandos para execução do cliente:

Executar o ficheiro .class:

java -cp bin -Djava.security.manager -Djava.security.policy==client.policy Client.SeiTchiz 127.0.0.1:45678 truststore_client clientid clientid clientid

java -cp bin -Djava.security.manager -Djava.security.policy==client.policy Client.SeiTchiz 127.0.0.1:45678 truststore_client manelteste manelteste manelteste

java -cp bin -Djava.security.manager -Djava.security.policy==client.policy Client.SeiTchiz 127.0.0.1:45678 truststore_client mantorras mantorras mantorras

Executar ficheiro .jar:

java -cp bin -Djava.security.manager -Djava.security.policy==client.policy -jar SeiTchiz.jar 127.0.0.1:45678 truststore_client clientid clientid clientid

java -cp bin -Djava.security.manager -Djava.security.policy==client.policy -jar SeiTchiz.jar 127.0.0.1:45678 truststore_client manelteste manelteste manelteste

java -cp bin -Djava.security.manager -Djava.security.policy==client.policy -jar SeiTchiz.jar 127.0.0.1:45678 truststore_client mantorras mantorras mantorras

Passwords das keystores dos utilizadores correspondentes

Username - Password

clientid – clientid

maneltest – maneltest

mantorras – mantorras

Truststore Client - Password

truststore_client – servidor

Decisões de desenho das soluções implementadas

Para organização do trabalho relativamente a cliente-servidor separámos o código de modo a ter os ficheiros relacionados ao servidor dentro da pasta ./data/ e os ficheiros locais do cliente em pastas na ./root/.

Dentro do ./data/ temos 3 pastas que contêm o que o próprio nome indica mais a keystore do servidor.

Primeiro, na pasta relativamente a ficheiros de grupo temos subpastas para cada grupo individual em que cada um tem os ficheiros de: histórico de mensagens, caixa de mensagens, e membros de cada grupo. As keys dos grupos que servirão para cifrar e decifrar as mensagens (que será feito do lado do cliente) encontram-se fora do domínio do servidor, na root, em ./GroupKeys/, cuja pasta tem um ficheiro por grupo com os membros e chaves associadas para cada momento em que houve mudança de chave. Como os ficheiros relativos ao servidor têm de ser protegidos, ciframos os mesmos para termos apenas ficheiros com a extensão .cif que não podem ser lidos por intrusos. No entanto, para realizarmos a cifra, usamos um ficheiro auxiliar com extensão .txt para escrever a informação que queremos que depois irá ser cifrado num .cif e eliminado o ficheiro temporário .txt que fora criado. Para além disto, como a cifra destes ficheiros é feita com chave simétrica, é feito o wrap da chave e guardada num ficheiro.

Na próxima pasta, temos os ficheiros acerca do utilizador que são mantidos no servidor. É criada uma subpasta para cada um dos membros que contém: Fotografias postadas para o sistema, ficheiro acerca de meta-dados do utilizador (a quem o utilizador segue, quem o segue, a que grupos pertence, que fotografias postou, respetivamente) e o ficheiro com a chave que será usada para decifrar o ficheiro dos meta-dados.

Na terceira pasta temos os ficheiros acerca dos meta-dados do servidor (nomes de todos os grupos criados, informação acerca das fotografias publicadas e todos os utilizadores registados na aplicação) bem como os ficheiros com as chaves respetivas.

Na root do projeto existem as pastas relativas ao lado do cliente. Primeiramente a pasta GroupKeys que contém as chaves de um determinado grupo (já falado acima), a pasta Keystores que contém as keystores de cada um dos clientes, a pasta PubKeys que contém os certificados dos clientes e a pasta Truststores que tem duas subpastas, do cliente e do servidor. A do servidor tem o seu certificado e a do cliente tem o certificado exportado para a keystore (que apenas contém o certificado do servidor).

Por último, relativamente às fotografias temos 2 pastas do lado do cliente, a pasta ./test/ que contém as fotografias que os utilizadores irão publicar e a pasta ./wall que contém uma pasta por cada cliente que por sua vez contém as fotografias devolvidas pelo servidor quando esse mesmo utilizador realiza o comando wall.

Decidimos guardar as keystores deste modo para que tudo o que se encontre dentro da pasta ./data seja relacionado com o servidor, e os clientes não têm acesso e tudo o que se encontra fora não esteja relacionado com o servidor.

Limitações do trabalho

As fotografias usadas para o comando Post devem estar dentro da pasta "test" na raiz do projeto (devido a permissões dos policies).

A mensagem enviada para os grupos não deve conter "%%" (é uma palavra reservada para tratamento de ficheiros).

A pasta bin não pode ser apagada, apenas devem ser apagados os seus conteúdos.

O nome do utilizador ou do grupo não devem conter ":" (é um caracter reservado para tratamento de ficheiros).