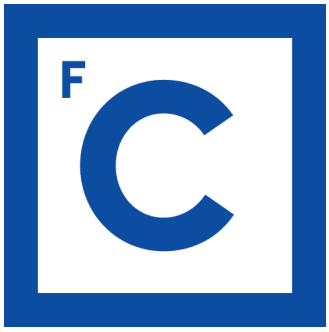
Segurança Informática

2022-2023

Grupo 12

FC45798 Filipa Fernandes
FC51601 José Sousa
FC53746 Andrei Tataru



Relatório da entrega da aplicação myCloud

Objetivos concluidos

A. **Gestão utilizadores no Sistema:**

Quando iniciamos o servidor é criado um ficheiro onde guarda os utilizadores e as suas respectivas passwords, caso este não exista.

B. **Implementação da Autenticação no Cliente:**

O utilizador consegue autenticar-se no cliente através dos parâmetros com flags “-u” e “-p”. Na tentativa de enviar para o servidor qualquer ficheiro, é requirido pelo cliente um valor para estas flags. Caso não receba estes argumentos, o cliente termina a execução. Para se autenticar com sucesso, a password do utilizador tem de corresponder à password guardada no servidor e à password guardada no seu respetivo ficheiro keystore. O servidor valida a autenticação do utilizador recorrendo às classes implementadas no pacote *auth*.

C. **Criação de Utilizadores:**

O cliente consegue criar novos utilizadores através do parâmetro “-au”. Para o registo de um novo utilizador é requerido do cliente um certificado para ser enviado ao servidor. O servidor por sua vez realiza a validação destes pedidos, verificando se o utilizador já existe.

D. **Confidencialidade das passwords:**

O servidor guarda de forma segura e confidencial as passwords dos seus utilizadores através de sínteses e sais. No ato de registo do utilizador é gerado um sal aleatório com 16 bytes de comprimento, recorrendo à classe SecureRandom. O algoritmo escolhido para servir de função de síntese foi o SHA-512 (ainda não quebrado/seguro).

E. **Integridade do ficheiro das passwords:**

Para detetar eventuais ataques à integridade do ficheiro de passwords, implementámos, com o padrão singleton, a classe *Validator* do pacote *auth* é responsável pela verificação do ficheiro das passwords. O construtor desta classe pede ao utilizador uma password de modo a realizar as verificações necessárias durante a execução. A hash gerada é guardada num ficheiro *myCloudServer/passwords.mac*. Se este ficheiro não existir, é pedido ao utilizador se quer criar um novo ficheiro para guardar a hash. Caso o utilizador responda que não, esta funcionalidade é desativada.

F. **Comunicação através de Canais Seguros:**

Para assegurar a confidencialidade da comunicação entre o cliente e o servidor, recorremos ao uso de SocketsSSL do java. Estes sockets utilizam o certificado do servidor guardado na truststore do cliente para assegurar que o servidor é de confiança e também para cifrar a comunicação entre os dois interlocutores.

G. **Guardar e Enviar ficheiros:**

Adaptadas da primeira entrega, as funcionalidades de envio, cifrar e assinatura de ficheiros continuam implementadas. Enquanto na primeira entrega havia apenas um utilizador, agora existem vários utilizadores, sendo necessário implementar um sistema simples de gestão de certificados para a criptografia assimétrica e híbrida. Ao enviar para o servidor, o cliente verifica *a priori* se tem localmente o certificado e caso não o tenha, recebe o certificado do respetivo utilizador recetor. Do mesmo modo, antes de receber do servidor os ficheiros, o cliente pede o nome do utilizador que enviou o ficheiro, verifica se tem o respetivo certificado, caso não o tenha, vai pedi-lo ao servidor.

Melhorias em relação ao primeiro projeto:

Em relação à primeira entrega o projeto foi reestruturado, com o principal foco das alterações sendo a mudança dum estilo de programação mais funcional para orientado a objetos. Foi ainda alterada a comunicação entre o cliente e o servidor: no envio de ficheiros entre os interlocutores, o tamanho do ficheiro é calculado *a priori*, calculando o tamanho resultante após a cifra dos ficheiros enviados, como pedido pela professora.

Melhorias a aplicar:

1. A password utilizada para as verificações, embora esteja definida com os atributos *final* e *private*, continua a estar guardada em texto claro na memória durante a execução. Se um atacante conseguir de alguma forma ler obter um *dump* da memória, vai conseguir ter acesso à password.
2. Optimizações no código e melhor controlo de erros.