PBOX Client

MIECT

Mário Liberato, Jorge Oliveira

PBOX Client

DETI

MIECT

Mário Liberato, Jorge Oliveira
(84917) mliberato@ua.pt, (84983) jorge.am.oliveira@ua.pt

21-04-2017

Resumo

Este relatório apresenta o trabalho de grupo realizado para criar um PBox Client sendo o objetivo inicial criar uma aplicação que comunique com um servidor. O cliente permite criar caixas, listar caixas, enviar e receber/ler mensagens de uma certa caixa. Foram utilizados Python e Pycharm. O cliente tem duas fases, terminal e gráfica. Fase terminal o utilizador utiliza o terminal de preferência e corre a aplicação a partir daí. A fase gráfica o utilizador tem uma página web onde pode realizar as operações de forma mais simples.

Conteúdo

Capítulo 1

Introdução

O tema do seguinte trabalho é a criação de um programa cliente onde é possível realizar certas operações envolvendo a interação com um servidor. Este servidor contém um modelo de "boxes"com (ou sem) segurança onde é possível deixar mensagens de curto comprimento para consulta do seu criador. Todos os utilizadores têm possibilidade de criar uma **caixa** com um certo nome, também sendo possível existir uma chave pública ao servidor (no caso das caixas seguras). Além da criação de caixas, é possível **listar**,**receber** e **enviar** mensagens para uma certa caixa.

O documento encontra-se dividido em quatro capítulos, sendo que no Capítulo 2 é apresentada a metodologia seguida para a criação do cliente e as funções do mesmo. No Capítulo 3 são apresentados os resultados obtidos no cliente e a respetiva análise. Finalmente, no Capítulo 4 são apresentadas as conclusões do trabalho.

Capítulo 2

Metodologia

Para a criação do cliente foi utilizada essencialmente programação em Python (para isto foram utilizadas ferramentas como IDEs, nomeadamente o PyCharm, Geany e vim) e a interface do cliente foi concebida como uma aplicação web baseada em CherryPy.

Antes da realização de qualquer programação o grupo reuniu-se e discutiu como deveria ser realizado o trabalho, que caminhos seguir até ao produto final. Foi optado ser realizada uma pequena base do cliente onde seria possível listar as caixas disponíveis no servidor, de seguida foram sendo adicionadas as funções de criar caixas, dar segurança às mesma, receber e enviar documentos às caixas. Finalmente a interface gráfica, mais apelativa ao utilizador foi introduzida. Antes de prosseguir, em cada etapa foram realizados testes para determinar erros ou falhas (em especial de segurança) para ser obtida uma experiência sem problemas ou crashes.

2.1 Descrição do cliente

Nesta secção são apresentadas as funções do cliente e como foram adaptadas ao mesmo.

2.1.1 Listagem

É pretendido listar todas as caixas seguras existentes através do envio de uma mensagem **LIST**, sendo que o servidor responderá com todas caixas seguras existentes e, caso existam, as chaves públicas das mesmas. Diversas funções foram concebidas para isto, para, por exemplo, obter o nome de todas as caixas, e mostrá-los ao utilizador.

2.1.2 Criação de caixas

Esta função permite criar uma caixa através do envio da mensagem **CREATE**, a mensagem deverá conter o nome da caixa e o seu timestamp. Ainda é possível

ter uma chave pública e assinatura da mensagem. Para oferecer segurança, se for optado por fornecer uma chave pública essa caixa só pode ser modificada com mensagens seguras.

2.1.3 Envio de documentos para uma caixa

Para o envio de documentos para uma caixa é necessário o envio da mensagem **PUT** contendo uma mensagem de até 65536 octetos. Se a mensagem tiver mais que este comprimento, será cortada para ter a dimensão adequada.

2.1.4 Receção de um documento

A receção de um documento de uma certa caixa é realizado através da mensagem GET contendo o seu timestamp. Foram criadas funções que permitem obter a mensagem mais antiga no servidor, assim como duas implementações de uma função que permite obter todas as mensagens existentes de uma vez. Esta função pode ainda eliminar todas as mensagens existentes ou apenas uma, sendo chamada sem utilizar o valor devolvido, ou usando as duas funções que as chamam sem retornar valores.

2.2 Segurança

Para segurança são utilizadas cifras assimétricas Rivest Shamir Adleman, Iniciais dos apelidos dos fundadores deste algorítmo de criptografia (RSA) com chaves de 2048 bits que são enviadas em formato Privacy-enhanced Electronic Mail (PEM). Também são necessárias Assinaturas com chaves RSA e sínteses-sha1, sendo isto visível na criação de caixas e obtenção dos documentos numa caixa.

2.3 Interface Web

Foi criada uma interface web para o cliente ser mais apelativo e de uso mais simples e familiar ao utilizador.

Capítulo 3

Resultados e Análise

Antes demais será de notar que o grupo fez o trabalho em conjunto com sucessivos commits no git. Inicialmente o trabalho começou a usar o GitHub devido a alguns problemas com a plataforma Code.UA. Acrescenta-se ainda que não foi possível implementar o nível de segurança.

O produto final existe em duas formas: Exclusiva ao terminal e outra com interface gráfica. Note-se que é necessário o **Python 2.7** e dois módulos do mesmo sendo esses o **CherryPy** e **Colorama**. Podem ser obtidos a partir dos seguintes passos (Para Linux/GNU):

1. apt-get install python
2. pip install colorama
3. pip install cherrypy s n tuivers

NOTA: Poderão ser necessárias permissões de administrador.

3.1 Terminal

Para aceder a esta forma, a que deverá consumir menos recursos ao computador, o utilizador deverá aceder ao ficheiro *client.py* através do terminal de preferência.

De seguida o utilizador pode introduzir o comando *help* para obter informação do cliente e comandos para o uso deste. É possível ver e criar caixas tal como enviar, ler e apagar mensagens em caixas. Vistos que o comando *list* mostra todas as listas disponíveis foi necessário implementar o comando *clear* para limpar o terminal (sem fechar o cliente).

Estão disponíveis vários comandos para a mesma função com fim a facilitar a tarefa ao utilizador.

Para executar um comando basta escrever o seu nome que será apresentada informação inicial requerida para a execução do mesmo.

Exemplo de uso:
create

Box Name> *Nome da Caixa*
Creating new box "Nome da Caixa"...
Created box "Nome da Caixa"successfully!
list
Getting information from the server...
Nomes das caixas
Number of Boxes: xxxx
clear

3.2 Interface

Para aceder à interface gráfica deverá ser executado o ficheiro $web_client.py$ e de seguida abrir o browser de preferência com o link *localhost:8080*. Desta forma o utilizador obtém uma página simples no qual pode realizar as 4 operações focadas para este trabalho. Esta interface web foi concebida em HTML para os visuais e aproveita os ficheiros *.py* e *.pyc* usados para o terminal. A finalidade de utilizar esta forma é para fazer uma demonstração rápida do que o cliente consegue fazer de forma simpática ao utilizador.

Capítulo 4

Conclusões

Concluindo o trabalho, foi possível criar um cliente que consegue comunicar com funções de receber e enviar mensagens. Foram incorporados elementos aprendidos em Laboratórios de Informática, tais como: Programação em Python, aplicações web, testes e comunicação entre aplicações. A maior dificuldade no trabalho foi a implementação das cifras o que não foi possível.

Acrónimos

RSA Rivest Shamir Adleman, Iniciais dos apelidos dos fundadores deste algorítmo de criptografia

PEM Privacy-enhanced Electronic Mail