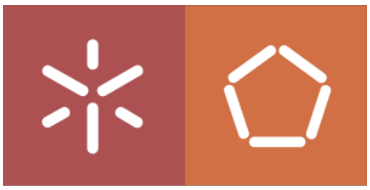

Universidade do Minho
Escola de Engenharia

Universidade do Minho
Mestrado Integrado em Engenharia Informática

SSI

Segurança de Sistemas Informáticos
Trabalho Prático 3

Janeiro 2020

Tânia Filipa Amorim Rocha
A85176

Maria Miguel Albuquerque Regueiras
A85242

1 Introdução

Nos sistemas clássicos Unix, existem vários mecanismos de proteção do sistema de ficheiros que este possui. Uma delas é a existência de permissões e classificação de utilizadores. Estas permissões são relativas a três ações que os utilizadores de cada classe podem realizar sobre ficheiros: ler, escrever e executar.

Existem 3 classes de utilizadores:

- **Owner**: todos os ficheiros e diretorias pertencem a um dono que os cria. Este define a *user class* de cada um.
- **Owner Group**: grupo de donos como indicado pelo nome, o dono pode pertencer a um grupo de outros donos.
- **Others**: conjunto de outros utilizadores que não são donos dos ficheiros.

O trabalho proposto pela equipa docente tem na sua base a manipulação de permissões recorrendo a estes conceitos para complementar estes mecanismos de controlo de acesso a ficheiros. Assim, foi desenvolvida uma maneira de verificar identidades de utilizadores e autorizar, ou não, a leitura de ficheiros.

De seguida é apresentado como instalar o software necessário para o programa funcionar devidamente bem como o processo de desenvolvimento do mesmo. Por fim, realizaram-se testes ilustrativos para demonstrar que o programa cumpre o que foi definido.

2 Manual de Instruções

Para permitir a utilização deste este *filesystem* a qualquer utilizador, são apresentados a seguir os passos para a instalação das ferramentas necessárias, compilação e utilização do sistema:

- É necessária a instalação da biblioteca **libfuse** mais recente, neste caso a versão *libfuse 3.10.1*. Para isso é preciso também fazer o download do *zip*, existente no repositório `https://github.com/libfuse/libfuse/releases/tag/fuse-3.10.1`, seguindo os passos existentes no ficheiro README aí presente.

  Nota: É necessário que exista uma instalação previa de **pyhton3**, **pytest**, **pip3**, **meson** e **ninja**, ferramentas necessárias à instalação do *libfuse*.

- Antes de se compilar seja o que for, é aconselhado inserir um registo no ficheiro *utilizadores.txt* com o formato "[nome] [número de telemóvel]". Tal é necessário visto que o grupo utilizou os seus dados pessoais, e para os proteger entendeu-se que seria melhor serem inseridos por quem quiser testar.
- Na directoria do projeto onde se encontra o ficheiro *passthrough.py* deve-se compilar o mesmo com o seguinte comando:

  ```
  python3 passthrough.py [directoryToBeMounted] [directoryToBeUsedAsMountpoint]
  ```

  Onde [**directoryToBeMounted**] representa o *path* para a directoria que contém o ficheiro a ser executado e [**directoryToBeUsedAsMountpoint**] representa o *path* de uma diretoria onde queremos que o *filesystem* seja montado (preferencialmente vazia).
- Após executar o *passthrough.py* com as regras especificadas, pode então ser iniciado outro cliente num terminal à parte que irá então aceder à directoria utilizada para a montagem do sistema. Aqui poderá tentar abrir o ficheiro *utilizadores.txt* que activará no terminal do servidor as medidas de segurança e autenticação para permitir a abertura do ficheiro.
- Conforme o que o servidor questionar, devem ser inseridas as informações respectivas. O nome do utilizador, que o servidor vai verificar se é um dos donos do ficheiro e no caso de o ser, enviará um SMS para o mesmo que deverá então inserir o *token* recebido no servidor quando este o pedir.
- Se todas as etapas forem seguidas, o utilizador for válido e o token/código estiver correto, o ficheiro abrir-se-á.

3 Desenvolvimento

No desenvolvimento do programa, após se instalar as ferramentas mencionadas, passou-se à anáilse do ficheiro *passthrough.py*. Escolheu-se **python** como a linguagem a usar pela sua simplicidade e por já existirem recursos on-line que facilitavam o processo de desenvolvimento (`http://thepythoncorner.com/dev/writing-a-fuse-filesystem-in-python/`).

Observou-se que o ficheiro *passthrough.py* definia uma classe Passthrough que continha definições de funções típicas do sistema operativo. Entendeu-se que se devia dar override das funções necessárias para abrir ficheiros, ou seja, a função **open**, e defini-la de forma a que se pudesse criar o mecanismo de autenticação.

Para se implementar a autenticação, o programa solicita a identificação do utilizador sempre que se tente abrir um ficheiro onde as permissões não o abranjam. No nosso caso, logo na main define-se o ficheiro *utilizadores.txt* com as permissões 000, indicando que ninguém o pode ler. Assim, qualquer utilizador (exceto a root) que lhe tente aceder terá de se identificar.

Após a solicitação do nome do utilizador, altera-se de novo as permissões deste para 400 para que seja possível consultá-lo para comparar o nome inserido e passar à verificação do mesmo. Se este estiver registado, o seu registo contém o seu nome e número de telemóvel o qual será usado para enviar um SMS com o *token* gerado aleatóriamente. Após o seu envio, o ficheiro retoma a ter permissões 000 para evitar falhas e tentativas de acesso entretanto.

Uma vez enviado o SMS, o programa inicia uma contagem decrescente de 30 segundos, período no qual o utilizador deve inserir o *token* recebido. Isto é feito através de um *signal* que será ativado no fim deste tempo. Se o *token* for inserido dentro do tempo, o signal é desativado, caso contrário e se der timeout este é acionado e o programa termina com o erro "Token nao chegou a tempo".

Quando há sucesso na operação, a permissão é alterada para 444 (todos as classes podem ler) e o utilizador consegue terminar a ação que pretendia fazer. Caso o token seja inválido este é confrontado com a mensagem "Código incorreto!", o alarme é desativado e não é possível tentar inserir o *token* corretamente, sendo necessário solicitar o acesso outra vez. Pode acontecer de haver uma falha no envio, nesse caso escrevendo a mensagem "Erro no envio da mensagem"ou ainda o utilizador não existir com a mensagem "Utilizador desconhecido!".

4 Testes

Com o objetivo de testar o sistema foram testados alguns casos distintos:

- Utilizador Correto e Token Correto após a tentativa de acesso ao ficheiro *utilizadores.txt*

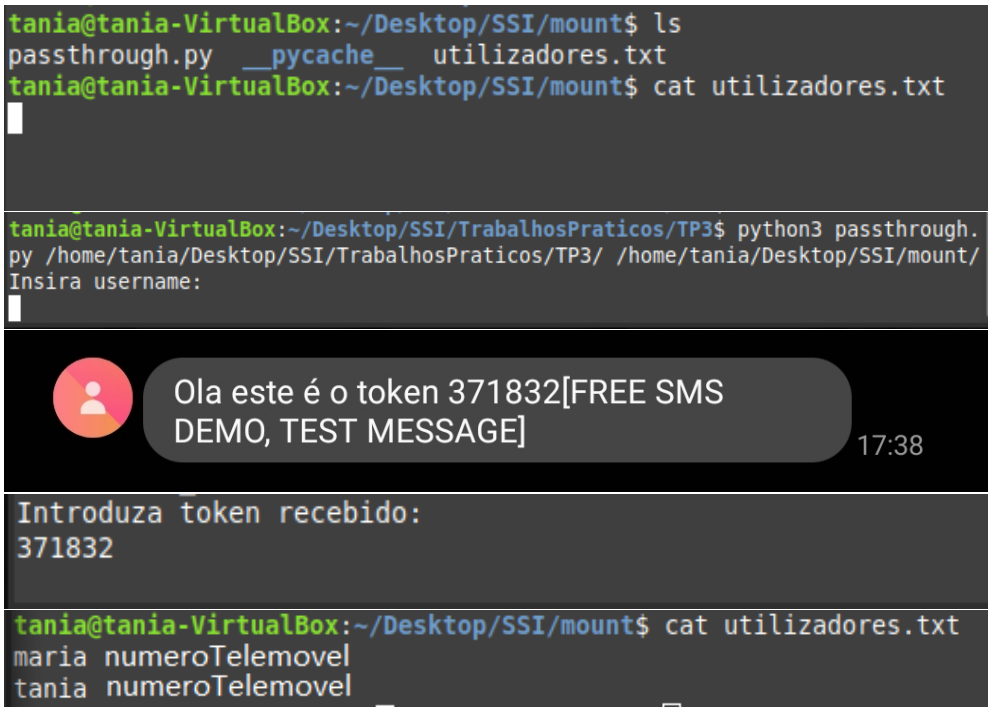

Figura 1: Teste 1

- Utilizador Correto e Token errado após a tentativa de acesso ao ficheiro *utilizadores.txt*

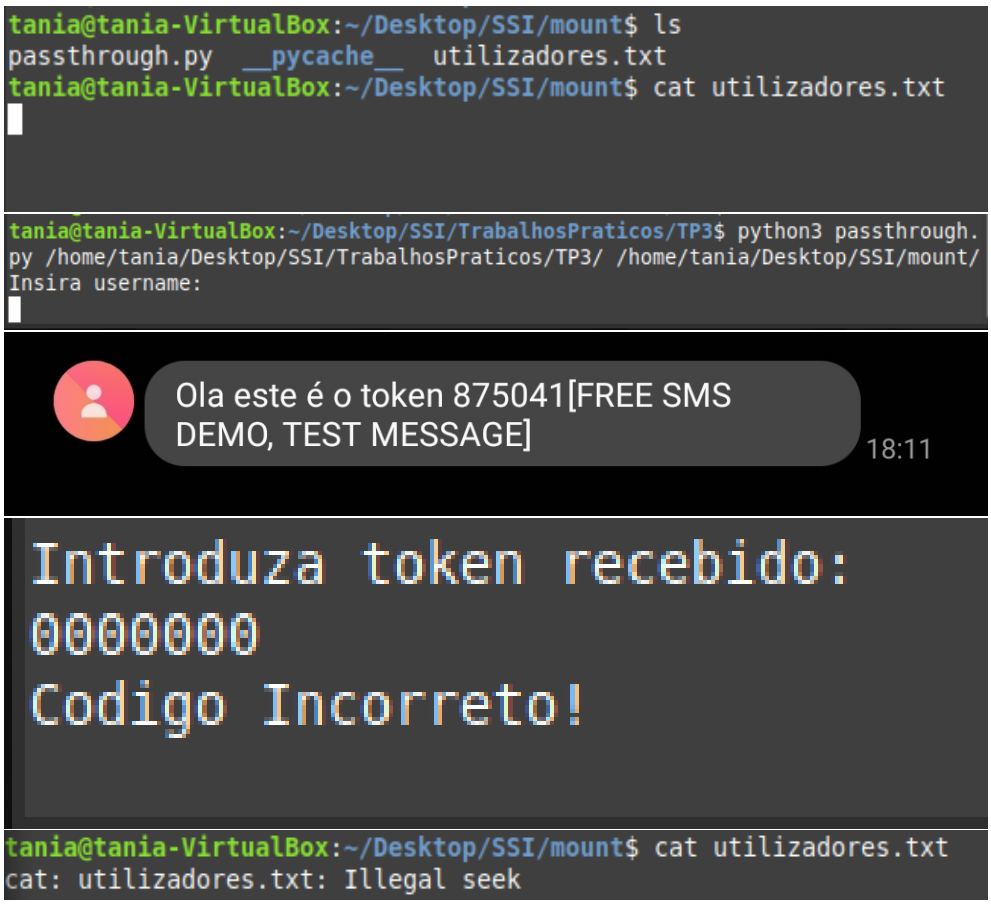

Figura 2: Teste 2

- Utilizador Inexistente na tentativa de acesso ao ficheiro *utilizadores.txt*

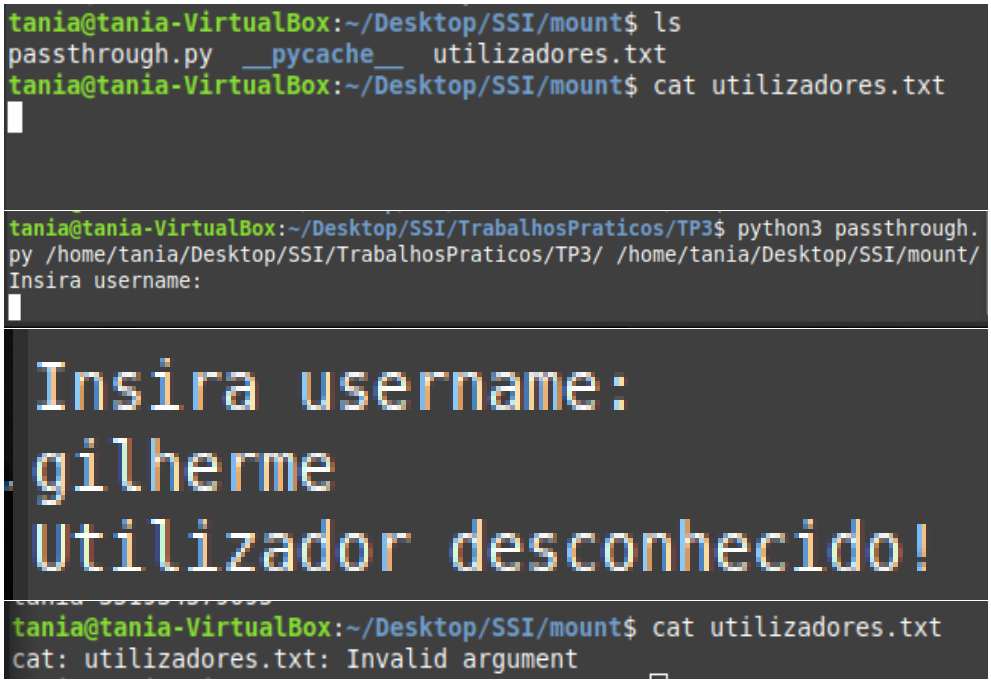

Figura 3: Teste 3

- Tempo limite de espera pelo token ultrapassado após tentativa de acesso ao ficheiro *utilizadores.txt*

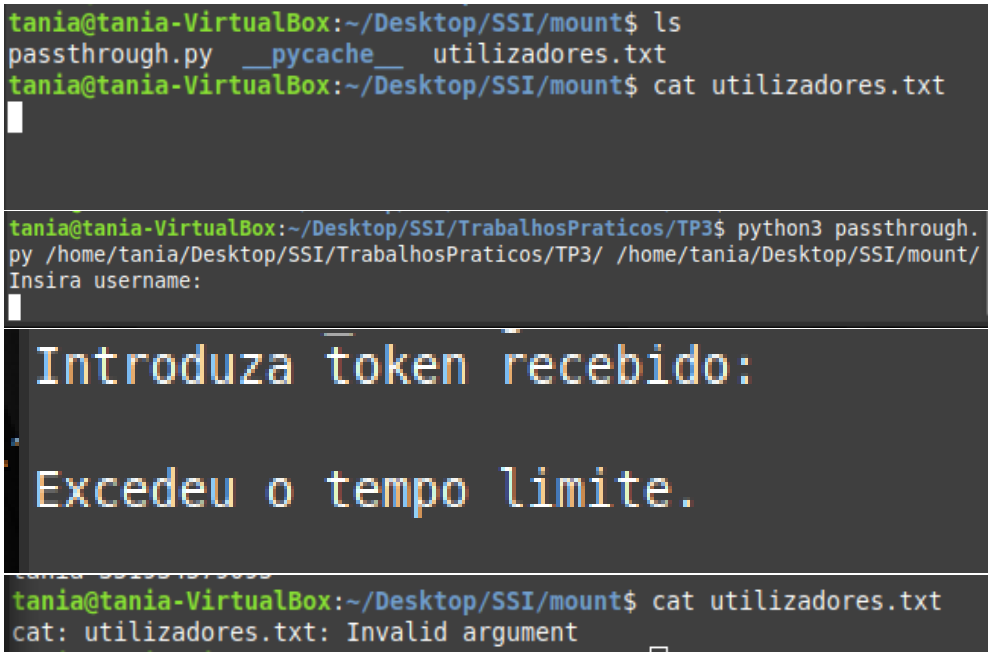

Figura 4: Teste 4

5 Conclusões

Com este relatório foi abordado todo o processo de desenvolvimento assim como os resultados de um programa que permite manipular permissões de forma a criar um mecanismo de autenticação no sistema de ficheiros do Unix.

Aprendeu-se, além de toda a teoria envolvida abordada nas aulas, como utilizar as ferramentas mencionadas anteriormente. Foi ainda uma boa forma de perceber melhor como o sistema de ficheiros funciona e como se pode utilizar o mecanismo de permissões para criar novas funcionalidades.