



Universidade do Minho
Escola de Engenharia

Universidade do Minho
Mestrado Integrado em Engenharia Informática
Tecnologia de Segurança
TP 3 Autorizaçãoo de Operações ao nível do Sistema de Ficheiros

Mateus da Silva Ferreira
(pg37159@alunos.uminho.pt)
Ricardo Cunha Dias
(pg39295@alunos.uminho.pt)

16 de Junho de 2020

Conteúdo

1	Introdução	3
2	Solução desenvolvida	4
2.1	Dependências	4
2.2	Arquitetura	5
2.3	Inicialização e utilização	5
2.3.1	Inicialização	5
2.3.2	Utilização	5
3	Conclusão	6
4	Bibliografia	6

Lista de Figuras

2.1 Arquitetura da solução 5

Introdução

Este trabalho prático consiste na implementação de um sistema de controlo de acesso de ficheiros, com o intuito de complementar os mecanismos de controlo já existentes no sistema operativo *Linux*. Para tal, foi utilizado a biblioteca libfuse, onde foi criado um novo sistema de ficheiro.

O sistema desenvolvido tenta emular um sistema ficheiros, portanto o utilizador pode caso tenha permissões navegar nos directórios e fazer determinadas operações, porém no caso das operações *write* e *read*, estão a ser monitorizadas pelo sistema e caso este utilizador tente fazer operações destas num ficheiro do qual não tem autorização o dono desse ficheiro vai ser informado via correio eletrónico, desse acesso.

Caso o dono deseje que esse utilizador possa de facto fazer operações no ficheiro, ele pode fornecer o mesmo nível de acessos que os utilizadores do seu grupo por um determinado período de tempo via um *link* que lhe foi enviado junto com o *email*

Solução desenvolvida

Para desenvolver uma solução à questão levantada, foi utilizado o Python-fuse, onde o script de exemplo *xmp.py* foi alterado para corresponder aos nossos objectivos. Para gerar a integração, enviar *emails* aos utilizadores do sistema, guardar informações relativas aos utilizadores, foi criado um script em Python (*rest.py*) que utiliza a biblioteca feita para Python *flask* para criar uma API REST e a base dados *nosql* Mongo.

2.1 Dependências

Para testar e utilizar as funcionalidades implementadas neste projeto, é preciso instalar as seguintes ferramentas:

- *libfuse*: Biblioteca principal, responsável pela criação do sistema de ficheiro. A documentação oficial com as instruções de instalação do *libfuse* encontra-se em <https://github.com/libfuse/libfuse/blob/master/README.md>.
- *python-fuse*: Interface Python para o libfuse que deve ser instalada de acordo com as instruções encontradas em <https://github.com/libfuse/python-fuse/blob/master/INSTALL>.
- Por ter sido feito em Python, não há necessidade de compilar o código. Para correr o projeto pela primeira vez, deve-se executar o script de instalação *install.sh* que irá correr a:
 1. Instalação da base de dados MongoDB;
 2. Instalação dos módulos Python necessários;

2.2 Arquitetura

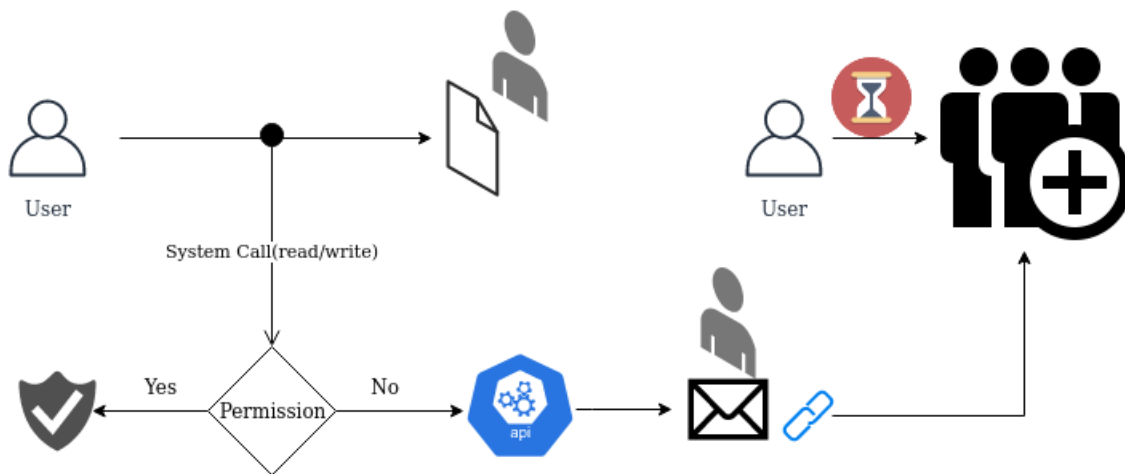


Figura 2.1: Arquitetura da solução

2.3 Inicialização e utilização

2.3.1 Inicialização

Antes de executar a shell script *run.sh* o utilizador deve criar primeiro os ficheiros que deseja que sejam replicados no directório *sourcepoint* ou ele próprio cria-los em tempo de execução. De seguida, deve-se correr o shell script *run.sh*, presente no ficheiro zip anexado a este relatório, que irá correr os seguintes scripts em Python listados em baixo:

1. O script *rest.py*, que inicializará o serviço *rest*;
2. O script *xmp.py*, presente no ficheiro *example/*, que executa o nosso sistema de ficheiros e replica os ficheiros do directório *sourcepoint* para o directório *mountpoint*

2.3.2 Utilização

Após aceder aos ficheiros mencionados anteriormente e ter os respectivos serviços necessários a correr, será criado o sistema de ficheiros dentro do directório *mountpoint*, que é um espelho do directório *sourcepoint*.

Feito isto, caso um utilizador tente aceder um ficheiro que ele não tenha permissão, será enviado uma mensagem via correio eletrónico ao dono do ficheiro alertando sobre a tentativa de acesso e disponibilizando um *link* para autorizar esse acesso durante um determinado período de tempo.

Ao aceder a esse *link*, o novo utilizador é inserido no grupo do dono do ficheiro por um determinado período de tempo que está predefinido como 10 minutos porém o sistema já tem a funcionalidade de introduzir um *Timer* mas devido a falta de tempo não foi totalmente desenvolvido. Ao atingir o tempo predeterminado, "chama" uma outra função que irá retirar o utilizador do grupo do dono do ficheiro.

Conclusão

Após a conclusão deste trabalho prático foi possível, além de aprofundar o nosso conhecimento relativo ao sistema de ficheiros linux e seus mecanismos de segurança, como também, ampliar o nosso conhecimento em ferramentas de acesso remoto. Os maiores desafios encontrados durante este trabalho foram o entendimento e utilização da biblioteca libfuse e a sua integração com o Python tal como a compreensão do funcionamento das diferentes permissões do sistema linux.

Bibliografia

<https://github.com/libfuse/libfuse>

<https://github.com/libfuse/python-fuse>

<https://palletsprojects.com/p/flask/>

<https://www.mongodb.com/>

<https://homepages.uc.edu/~thomam/Intro.Unix.Text/File.System.html>