



Universidade do Minho
Escola de Engenharia

TECNOLOGIA DA SEGURANÇA

TP3

Autorização de Operações ao nível
do Sistema de Ficheiros

Grupo 3 - Diogo Araújo A78485; Diogo Nogueira A78957;

Conteúdo

1. Contextualização.....	3
2. Arquitetura da Solução	3
3. Implementação.....	5
4. Execução/Teste do Programa.....	6
5. Conclusões e Críticas Finais	7

1. Contextualização

A ideia deste último trabalho prático, surge no âmbito da Unidade Curricular Tecnologia da Segurança e a sua ideia base passar por demonstrar as decisões realizadas ao nível da autorização de operações por parte de um Sistema de Ficheiros. Com o auxílio da biblioteca *libfuse*, o objetivo passa então por criar um mecanismo de autorização de abertura de ficheiros que complemente os mecanismos de controlo de acesso de um Sistema de Ficheiros tradicional.

A biblioteca *libfuse* é a grande aliada em todo este processo, dado que através dela conseguimos criar um Sistema de Ficheiros próprio sem existir uma necessidade interna de alterar o sistema de ficheiros predefinido pelo *Linux* em si.

Assim, com o nosso Sistema de Ficheiros criado e o mecanismo de autorizações totalmente implementado, espera-se que sempre que alguém requisita uma operação de abertura de um ficheiro – através do recurso interno à *system call open()* – exista um envio de um “token de segurança” que permita confirmar a autorização do utilizador em causa. Caso este *token* seja colocado no prazo máximo de 30 segundos, a autorização é dada, sendo o ficheiro corretamente aberto.

2. Arquitetura da Solução

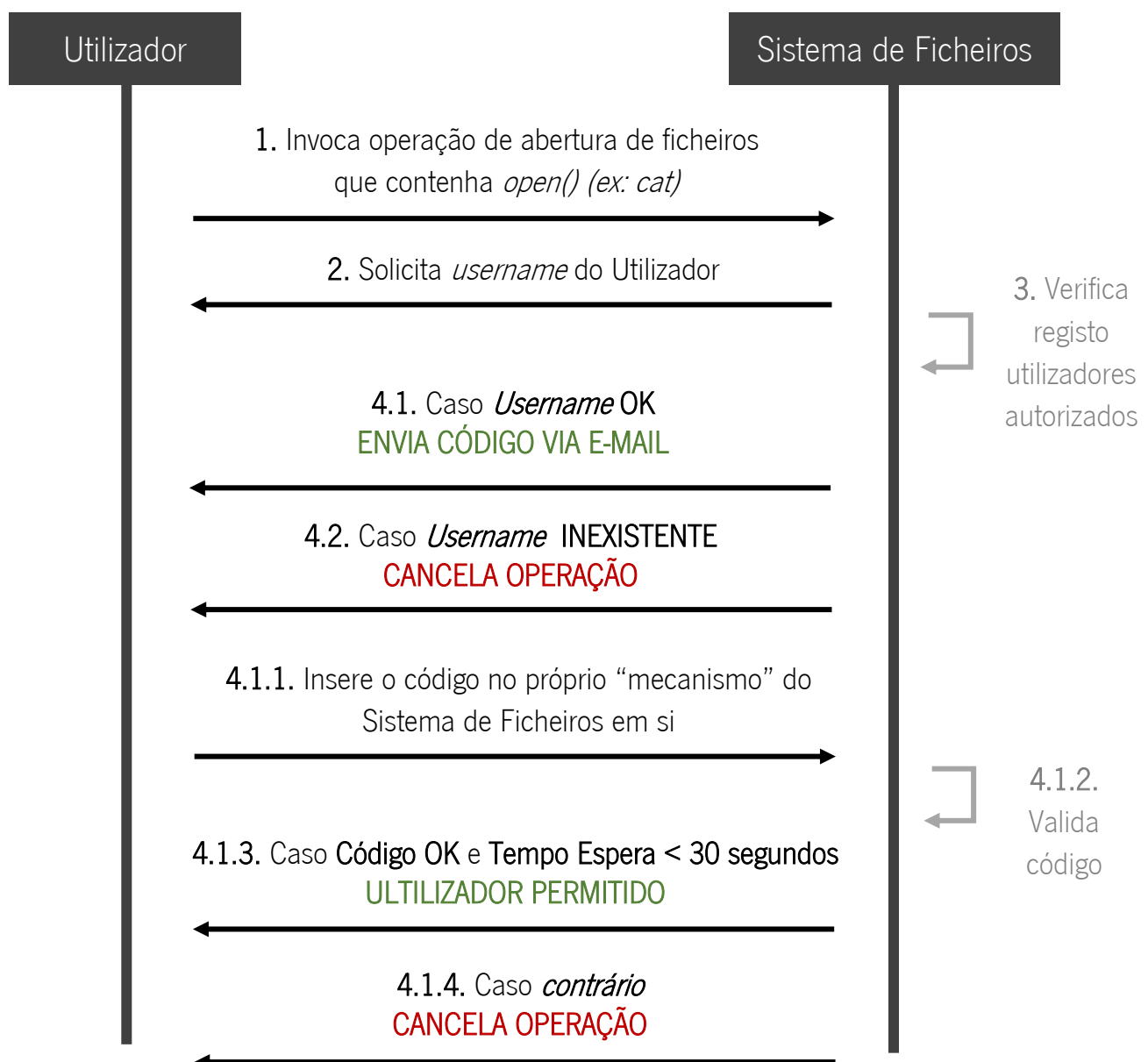
O desenho/esboço inicial da possível Arquitetura da Solução surge como uma medida de boa prática ao nível do projeto, tendo em conta que nos ajuda a perceber quais as alterações a serem feitas e de que forma será então possível estabelecer este controlo de segurança.

Numa versão prematura da Arquitetura da Solução tinha ficado definido que sempre que era solicitada a operação de abertura de um ficheiro, era enviado um email para o utilizador que solicitou a operação. Com esse email vinha um código de segurança que deveria ser usado pelo próprio, de modo a se verificar a sua autorização. Esta ideia não era suficiente por si só, uma vez que apenas poderia ser enviado um código via email para utilizadores autorizados pelo “programa” em si. Daí a ideia de existir um ficheiro com todos os utilizadores autorizados e a sua respetiva forma de contacto, para que sempre que um utilizador introduzisse o seu *username*, se fizesse um parse

de todo o ficheiro de permissões até se verificar se efetivamente tem essa possibilidade de abertura.

Numa versão final da Arquitetura da Solução, mudam-se um pouco as coisas. Assim, sempre que é mostrada a intenção de um utilizador em abrir um ficheiro, envia-se um e-mail ao mesmo, que conterá na mesma um código para se usar na validação.

Toda a parte de gestão de permissões fica ao cargo do sistema montado, tendo em conta que este será o grande responsável por verificar o *username* do utilizador e aguardar os 30 segundos até que seja validada a operação requisitada após o envio do e-mail. Todo este pensamento torna-se mais claro se observarmos a troca de ações representada sobre forma de diagrama, através da imagem abaixo anexada.



3. Implementação

No que toca à implementação da solução anteriormente descrita, o grupo recorreu a um Sistema de Ficheiros totalmente modelado na linguagem de programação *Python*. Este *script* usado baseia-se na biblioteca **fusepy** disponibilizada em *Python*, onde se escreve toda uma Classe com as várias chamadas ao sistema pelo Sistema de Ficheiros que são atribuídos para correr diretamente pelo *Operating System* em si.

Em modo de exemplo, e de forma muito simples, para se definir o método *open()* apenas é necessário escrever `open() = os.open`, e dessa forma redireciona-se o trabalho da *system call* para o sistema operativo Unix. Isso reduz muito código que fica visível no exemplo fornecido pelo professor, e ajuda também na compreensão da alteração que é necessária executar neste último projeto.

Com este ponto de partida, alterou-se a função *open()* de modo a ir ao encontro da validação por parte do utilizador que está a invocar a operação. Tal como dito antes, o grupo escolheu a abordagem de envio de e-mail, usando para isso a biblioteca **smtplib** que permite cumprir esse requisito.

Visualmente, o código desenvolvido é muito simples de compreender. Atente-se no excerto abaixo:

```
if(user[0] == username):

    # Tratar de Enviar o SMS.
    # Strip trata de remover espacos e outros caracteres des
necessarios.
    tokenToSend = str(random.randint(100000, 999999))
    sendEmail(user[1].strip(), tokenToSend)

    # Parte de verificacao e possivel timeout.
    timeout = 30

    try:
        signal.signal(signal.SIGALRM, handler)
        signal.alarm(timeout)
        v = verifyToken(tokenToSend)
        signal.alarm(0)
```

```
if v:
    return os.open(full_path, flags)
else: return 0
except IOError:
    print("\nExcedeu o tempo limite.\n")
```

Após se ter verificado a existência do *username* no ficheiro que mapeia os utilizadores permitidos e a sua respetiva forma de contacto, cria-se um *token* que é enviada por e-mail ao utilizador, por meio da função *sendEmail*.

No excerto, valida-se também a existência de um *timeout* que limita a validação do código por exatamente 30 segundos, utilizando para isso as bibliotecas *Python* que nos possibilitam criar sinais e alarmes associados então a um *timeout* de 30 segundos para a confirmação do código enviado por email.

4. Execução/Teste do Programa

Para se testar o programa deve-se primeiro criar duas diretorias. Uma delas será a diretoria reservada para guardar os ficheiros em si. A outra, permitirá montar o novo Sistema de Ficheiros, e nele veremos então os ficheiros dessa diretoria reservado, podendo estes serem manipulados como se estivéssemos no Sistema de Ficheiros original.

Com isto pronto, e a biblioteca fusepy instalada, apenas é necessário correr o seguinte comando:

```
python3 passthrough.py /diretoriaFicheiros
/diretoriaMount
```

5. Conclusões e Críticas Finais

Estando a parte prática da Unidade Curricular totalmente compreendida e trabalhada, é hora de redigir as observações finais. Relativamente ao projeto em si, o grupo não encontrou grandes dúvidas no que toca ao entendimento do sistema FUSE e de que forma poderia ser usado para se criar um Sistema de Ficheiros próprio. Com isso compreendido, apenas foi preciso estudar o ficheiro disponibilizado pelos docentes, ficheiro esse que acabou por nos levar a procurar formas mais simples e intuitivas de se executar os métodos de ficheiros/diretorias que do Sistema Operativo fazem parte.

Tendo o grupo um Sistema de Ficheiros preparado de forma intuitiva sob a forma da linguagem *Python*, o restante do projeto foi reproduzido facilmente, dado que existe por parte de ambos os membros uma boa prática no mundo desta linguagem. Assim, usou-se apenas uma nova biblioteca para que fosse possível enviar os emails para os demais utilizadores autorizados a realização a operação de abertura.

Apesar de se revelar simples de executar, o projeto permitiu desenvolver novos conhecimentos acerca dos Sistemas de Ficheiros, marcando assim a importância na segurança da informação envolvida.