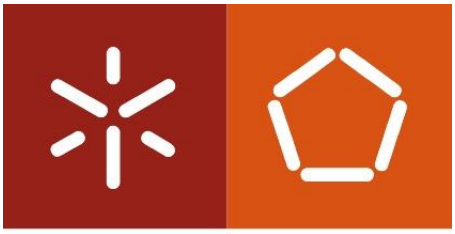

Universidade do Minho

Mestrado Integrado em Engenharia Informática

Segurança de Sistemas Informáticos

Relatório do Trabalho Prático 3

Grupo 11
Lázaro Pinheiro (A86788)
Luis Ramos (A83930)

28 de janeiro de 2021

Conteúdo

Lista de Figuras

1 Introdução

O presente relatório desenvolve-se no âmbito da Unidade Curricular Segurança de Sistemas Informáticos, tendo como principal objectivo reforçar o nível de segurança de controlo de acesso de um sistema de ficheiros tradicional do sistema operativo *Linux*.

Para isso será aplicado um mecanismo adicional de autorização de operações de abertura de ficheiros, baseado em *libfuse*.

O utilizador que tentar aceder a determinado ficheiro, terá de colocar o seu nome de utilizador, e, caso esteja registado, irá receber uma mensagem de texto no seu telemóvel com um código gerado automaticamente. Introduzindo o código correctamente, o acesso ao ficheiro será concebido.

De notar que o código tem apenas uma duração de 30 segundos, sendo que depois expira, tornando-se assim inválido.

2 Diagrama Modelo

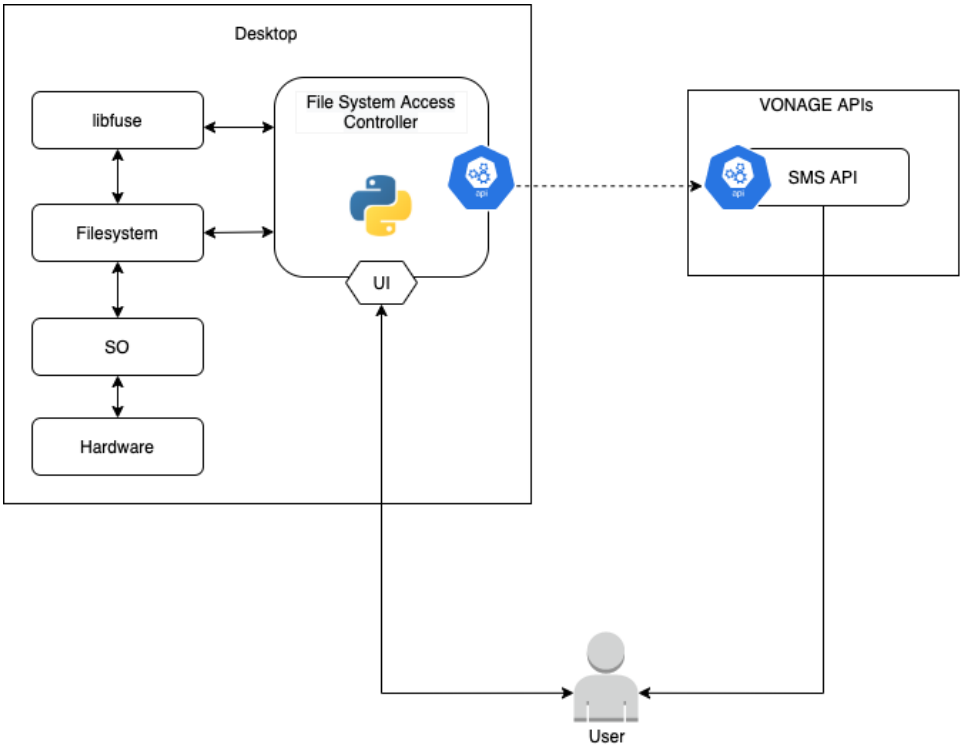

Figura 1: Diagrama Modelo do Sistema

3 Implementação

3.1 Arquitetura e Estrutura

Como era requisito, todo este projecto tem por base a biblioteca *libfuse* e foi puramente desenvolvido sobre a linguagem de programação *Python*. Assim sendo, foram implementadas algumas alterações à função de **open()**, de modo a criar um controlo de acesso de ficheiros.

Este é constituído por várias fases que serão referidas ao pormenor ao longo deste relatório.

De salientar que o acesso aos ficheiros apenas é permitido a quem tenha permissões para tal.

3.2 Verificação do Utilizador

Recorreu-se a um ficheiro de texto(*resources/users.txt*) para guardar todos os utilizadores com permissões de acesso aos ficheiros. Para cada utilizador é associado um contacto, neste caso, um número de telemóvel.

De notar que foi garantida toda a segurança para que este ficheiro não pudesse ser alterado ou roubado, como iremos referir mais abaixo neste relatório.

Quando um utilizador pretende aceder a um ficheiro é invocado o método **open()**, e é lhe solicitado o seu *username* de acesso. É verificado se o mesmo existe no ficheiro de texto. Caso exista, segue-se para o ponto seguinte da geração de um *token* e envio de uma mensagem com o mesmo. Caso contrário, é negado, ao utilizador, o acesso ao ficheiro.

3.3 Geração do Token

Um *token* consaiste num conjunto de letras(maiúsculas e minúsculas) com formato, ordem aleatória e tamanho predefinido. Note-se que o tamanho do *token*, pode ser alterado pelo administrador do sistema, através do ficheiro de configurações(*resources/config.json*).

O grupo achou por bem colocar apenas 6 caracteres, dando segurança suficiente e não sendo trabalhoso para o utilizador de o escrever.

No anexo A pode ser consultado um exemplo da mensagem enviada ao utilizador.

3.4 Envio da SMS

Para se realizar o envio das mensagens com o *token* ao utilizador que requer acesso a um determinado ficheiro, recorreu-se à **API** disponibilizada pela **VONAGE**[1] a qual demonstrou-se simplificada, flexível e sem muitas burocracias, pontos estes que levaram os elementos do grupo a eleger.

Contudo, foi apenas reclamada uma conta grátis, obtendo assim apenas um crédito de 2€. Isto revela-se um entrave, uma vez que permite um reduzido número de utilizações. Importa ainda referir que como mecanismo de segurança, uma vez que é uma conta grátis, todos os utilizadores aos quais se pretende enviar mensagens têm de constar na sua *Whitelist*, ou seja, antes de serem enviadas mensagens pelo sistema desenvolvido, para o número de telemóvel em questão, é necessário registar o mesmo no *website* do fornecedor de comunicações, o qual envia um *PIN* que deverá ser inserido num prazo de 5 minutos no *website* do mesmo.

3.5 Autenticação do Utilizador

Após envio da mensagem, o utilizador tem apenas 30 segundos para inserir o *token* recebido.

Se esse tempo exceder, então o código expira e é negado, ao utilizador, o acesso ao ficheiro. Caso contrário, é necessário verificar se o *token* inserido pelo utilizador corresponde ao *token* gerado e lhe enviado. No caso de corresponderem, é permitido o acesso ao ficheiro, caso contrário é negado.

3.6 Segurança e Vulnerabilidades

Quanto à segurança e às vulnerabilidades da aplicação há vários aspectos em ter em conta.

Recorrendo à abstração das ameaças, tentando encaixar-las num sistema de classes desenvolvido pela Microsoft intitulado por **STRIDE**(acrónimo de cada componente apresentadas sucintamente de seguida), ou seja, um modelo de ameaças usado para ajudar a raciocinar e encontrar ameaças num sistema.

- *Spoofing* - Alguém que se faça passar por outra pessoa, não obterá acesso aos ficheiros requisitados, uma vez que o sistema tem como medida de segurança o envio de um *token* para o número de telemóvel associado a um determinado utilizador;

- *Tampering* - Um utilizador que não tenha permissões para aceder a um ficheiro não o poderá alterar. Note-se que também não poderá alterar o ficheiro de utilizadores(correspondência nome de utilizador para número de telemóvel) e o ficheiro de configurações da aplicação, dado que as suas permissões são 000 e apenas são alteradas pela aplicação para efetuar leituras para 400;

- *Repudiation* - O uso do ficheiro de utilizadores, confere o não repúdio dos mesmos, uma vez que qualquer ação efetuada sobre os ficheiros em causa apenas será possível por um utilizador que exista nesse mesmo ficheiro;

- *Information Disclousure* - Os ficheiros apenas são acedidos a utilizadores com permissões. O sistema não garante que informação presente nestes ficheiros seja divulgado por quem tem permissões para a consultar;
- *Denial of Service* - O sistema não é vulnerável a pequenos ataques, uma vez que atende um de cada vez;
- *Elevation of Privilege* - Um atacante com permissões de super utilizador facilmente consegue alterar os dados presentes no ficheiro dos utilizadores, assim sendo, coloca todo o sistema vulnerável às ameaças supra-citadas.

3.7 Dependências e Compilação

Para automatizar o processo de instalação e compilação do projecto, o grupo decidiu criar um ficheiro(*setup/setup.sh*), o qual consiste numa *shell script* que determina qual o Sistema Operativo do *host*, determina qual a versão do gestor de pacotes padrão escritos em *Python*(*pip*) presente e dependendo deste instala o *VONAGE*. Este ficheiro é sempre executado automaticamente no inicio da execução da aplicação desenvolvida pelo grupo.

Por forma a executar a aplicação, dirija-se para a diretoria do projeto e execute o seguinte comando no seu terminal:

```
(python/python3) main.py "diretoria fonte" "diretoria destino"
```

Por forma a facilitar a demonstração do sistema o grupo preparou um vídeo, no qual pretende exemplificar o uso implícito do método **open()**, bem como de possíveis "exceções"que possam ocorrer. Este vídeo encontra-se apenas disponível para o acesso a partir do seguinte *url*: https://www.youtube.com/watch?v=WPRZ6-OFGVM&feature=youtu.be

4 Conclusão

Culminada a elaboração deste trabalho prático, importa referir que a execução do mesmo permitiu aos elementos do grupo compreender o funcionamento dos mecanismos de controlo de acesso a ficheiros.

A execução desta fase do trabalho prático permitiu uma melhor consolidação dos construtos teóricos através da utilização da biblioteca *libfuse*.

No ímpeto geral o desenvolvimento desta fase do trabalho decorreu como planeado, alcançando os objectivos delineados pelo enunciado.

Referências

[1] *VONAGE Web Site*
https://www.vonage.com

Appendices

A Exemplo de Mensagem Enviada

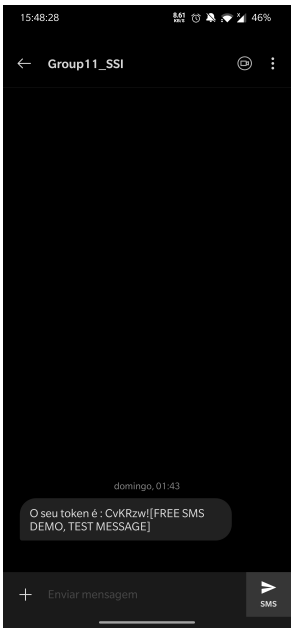

Figura 2: Exemplo da mensagem recebida pelo utilizador