Universidade do Minho

Mestrado Integrado em Engenharia Informática

Segurança de Sistemas Informáticos

Autorização de Operações ao nível do Sistema de Ficheiros

Fábio :: A78508

Joel :: A79068

20 de Janeiro de 2019

Conteúdo

1 Introdução **1**

2 Arquitetura da solução **1**

3 Implementação **2**

3.1 Aplicação Web 2

3.2 Sistema de Ficheiros 3

4 Execução **3**

5 Conclusão **4**

1 Introdução

O presente documento, realizado no âmbito da unidade curricular de Segurança de Sistema Informáticos, tem como objetivo apresentar as decisões tomadas na realização do trabalho prático número 3, que aborda a temática *Autorização de Operações ao nível do Sistema de Ficheiros*.

O principal objetivo deste projeto é a elaboração de um mecanismo de autorização de abertura de ficheiros que complemente os mecanismos de controlo de acesso de um sistema de ficheiros tradicional. Para este efeito é proposta a utilização da biblioteca *libufse*. De uma forma genérica esta API permite que utilizadores criem os seus próprios sistemas de ficheiros sem a necessidade de alterações ao nível do *Kernel* do sistema operativo.

Tendo o mecanismo de autorização implementado, um utilizador ao invocar uma operação de abertura de um ficheiro, *open()*, deve retornar sucesso caso o utilizador insira o código de segurança que lhe foi fornecido. Retornará insucesso, caso passem 30 segundos após esta invocação e nenhum código tenha sido inserido, ou o código que porventura inseriu está errado.

2 Arquitetura da solução

Antes de se efetuar a implementação do projeto, foi elaborada uma arquitetura do mecanismo a desenvolver, bem como definidas todas as decisões a tomar aquando da elaboração do trabalho.

Foi definido que, tal como sugerido no enunciado, a troca de ações/mensagens seria desencadeada após o utilizador desejar abrir um determinado ficheiro do sistema de ficheiros criado. Numa primeira instância, determinou-se que o email do utilizador teria de ficar, a partir do momento de criação do sistema, associado a este, para posteriormente ser possível uma comunicação. Posteriormente, estipulou-se que depois de mostrada a intenção do utilizador em abrir um ficheiro, seria enviado um email com um código gerado naquele instante e apresentada uma página *web* que lhe permitisse introduzir o código recebido.

Já a parte de permissão ficaria a cargo do sistema montado, isto é, este é que será responsável por verificar o código inserido bem como o tempo passado até ao momento. Caso o cliente tenha inserido o código corretamente e dentro do tempo previsto (30 segundos) será dada a permissão para a ação. Caso contrário, isto é, código que inseriu é diferente do enviado ou tenham passados mais de 30 segundos, não será dada permissão para o utilizador efetuar a operação desejada.

Deste modo, foi desenvolvido um esquema, representativo da troca de ações cronologicamente entre o utilizador, sistema de ficheiros e a aplicação web, figura 1.

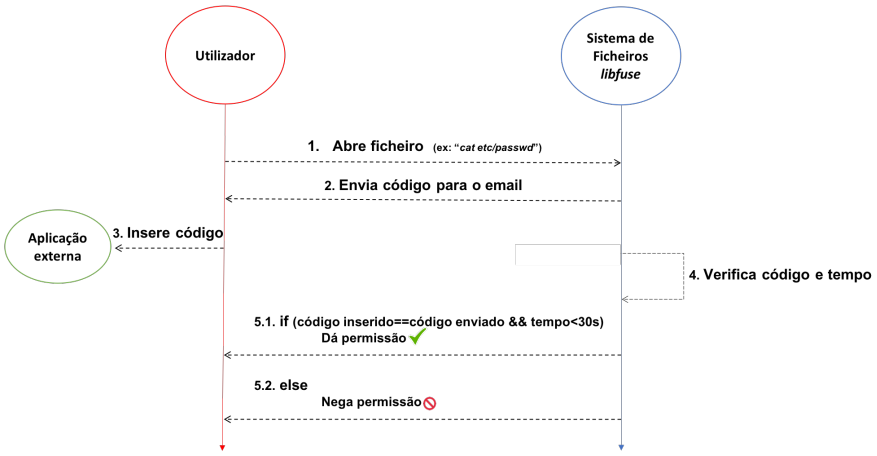

Figura 1: Troca de ações/mensagens despoletadas por abertura de um ficheiro

3 Implementação

3.1 Aplicação Web

De forma a simplificar o processo de inserção do código gerado e enviado ao utilizador, como referido na secção 2, foi desenvolvido um servidor trivial em *NodeJS* que está à escuta na porta 12345 e apenas apresenta uma página ao utilizador 2 bem como guarda o código que o utilizador inseriu num ficheiro, */tmp/validationCode.txt*, que funciona como persistência de dados ao nível do código de validação. É necessário a observação deste ficheiro em duas perspetivas, dado que, como dito anteriormente o sistema de ficheiros *libfuse* é um "espelho"do sistema de ficheiros nativo. Por conseguinte, começando pela perspetiva da ação do *libfuse*, quando este gera um código aleatório e procura verificá-lo no */tmp/validationCode.txt* do sistema de ficheiros *libfuse*, na outra perspetiva, do lado do *servidor*, este insere a informação do código fornecido pelo utilizador, no */tmp/validationCode.txt* nativo. Embora, à primeira vista, pareça que são ficheiros distintos, estes estão estritamente relacionados pelo *libfuse*, o que permite que quando num lado o *libfuse* esteja à espera de algo nesse ficheiro, caso aconteça uma escrita do lado nativo, automaticamente essa informação é transcrita, fornecendo a informação que permite o funcionamento esperado.

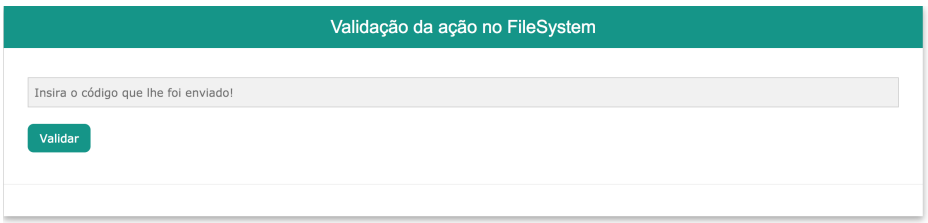

Figura 2: Página *web* apresentada ao utilizador

3.2 Sistema de Ficheiros

Em relação ao sistema de ficheiros, foi usado o exemplo proposto no enunciado, *passthrough.c*, sendo que se procedeu a diversas alterações.

Numa primeira instância foi implementada uma função em *python* que tem o objetivo de enviar para um determinado email um código previamente gerado. De forma a simplificar este processo, decidiu-se criar uma função em C, que tem o código *python* embebido. Para isso foi usada a biblioteca "Python.h"que permite, de uma forma de alto nível, fazer isso.

Todas as outras alterações foram efetuadas na função *xmp_open()*, pois é esta a função acionada sempre que um utilizador deseja abrir um ficheiro.

Inicialmente, nesta função, é invocado o método de envio de email, explicado anteriormente. Seguidamente existe um *alarm(30)*, que tem como objetivo ao fim de 30 segundos despoletar uma ação que mudará o estado do programa (modificando a variável global time_out para 1). Entretanto, durante os 30 segundos, o processo encontra-se num *loop* onde, de 1 em 1 segundo, verifica se foi inserido algum código no ficheiro gerado pela aplicação web. Em caso afirmativo, é testado se o código inserido é igual ao código enviado por email, neste cenário, é dada a permissão ao utilizador para abrir o ficheiro.

```
...
enviar_email(user_email,codigo);

alarm(30);

while(!time_out && inserido == 0){
        FILE *key_file = fopen ("/tmp/validationCode.txt", "r+");
        fscanf(key_file, "%d", &inserido);
        fclose(key_file);
        sleep(1);
}
...
```

Por outro lado, caso passem os 30 segundos, o processo sai do *loop* anterior e não conclui a função com sucesso, ou seja não é dada permissão ao utilizador.

4 Execução

Para compilar e executar o projeto, é necessária a execução dos seguintes comandos:

1. **Instalar *package* do *python*:**
   - *sudo apt-get install python3-dev*;
2. **Pôr o servidor à escuta na porta 12345:**
   - Na diretoria validationServer: *npm start*

3. **Compilar o sistema de ficheiros:**
    - *gcc -Wall passthrough.c ‘pkg-config fuse3 –cflags –libs python3‘ -o passthrough*
4. **Executar o sistema de ficheiros:**
    - *mkdir FileSystem*
    - *./passthrough /FileSystem*
5. **Testar o programa:**
    - Por exemplo, *cat FileSystem/etc/passwd*

5 Conclusão

Terminada a implementação do projeto, é de realçar que apesar das dificuldades encontradas durante o seu desenvolvimento, quase todas as funcionalidades propostas no enunciado foram elaboradas, exceto o registo de todos os utilizadores que poderão aceder ao sistema.

No entanto é de destacar as dificuldades, nomeadamente na ligação de todos os componentes do sistema, a aplicação web, o sistema de ficheiros e a persistência de dados usada, que neste caso foi em ficheiros. Isto despoletou uma maior dispêndio de tempo nestas tarefas o que dificultou a conclusão da totalidade do trabalho.

Em suma, é importante referir que o trabalho permitiu que se desenvolvessem os conhecimentos acerca dos sistemas de ficheiros e sendo totalmente prático, possibilitou que se deparasse com vários erros de baixo nível que tiveram de ser resolvidos.