Universidade da Beira Interior
Faculdade de Engenharia
Departamento de Informática

Segurança de Sistemas Informáticos
Guia para Aula Laboratorial 8
2º Ciclo em Engenharia Informática
2º Ciclo em Eng. Eletrotécnica e de Computadores
2º Ciclo em Matemática e Aplicações

Computer Systems Security
Guide for Laboratory Class 8
M.Sc. in Computer Science and Engineering
M.Sc. in Electrical and Computer Engineering
M.Sc. in Mathematics and Applications

Sumário
Exercícios de fortalecimento do conhecimento em mecanismos de controlo de acesso em sistemas operativos, nomeadamente no controlo de acesso discricionário de um sistema operativo Unix-*like*.

Summary
Exercises for strengthening the knowledge on access control mechanisms in operating systems, particularly on the discretionary access control of Unix-like operating systems.

Pré-requisitos:
Este enunciado pressupõe o acesso a uma máquina com sistema operativo Linux. Em particular, os exercícios foram testados no sistema operativo Fedora. A execução de todos os exercícios pressupõe que o executante tem acesso ao sistema operativo com pelo menos dois utilizadores diferentes, sendo um desses utilizadores o `root`.

1 Verificação das Permissões de Acesso
Verification of Access Rights

As tarefas desta secção pressupõem que o executante faz o *login* no sistema operativo (e, potencialmente, no ambiente gráfico, e.g., Gnome) com uma combinação *username*/*password* de um utilizador normal, i.e., um utilizador sem privilégios de `root` (e.g., `Aluno Aluno`).

Tarefa 1 *Task 1*

Abra um terminal e emita o seguinte comando:
```
> who
```

Q1.: Para que serve este comando?

Serve para ver utilizadores do sistema.

Q2.: Quantos utilizadores estão ligados no seu sistema atualmente?
☐ 0,5 ☐ 0,99 ☒ 1 ☐ 1,11 ☐ π ☐ $\sqrt{2}$

Tarefa 2 *Task 2*

Navegue até à diretoria do Ambiente de Trabalho (se ainda lá não estiver) e crie uma diretoria chamada `ThisDirIsMine`. Precisa de um comando parecido com:
```
> mkdir ThisDirIsMine
```

Procure criar também a diretoria `ThisDirIsMineAlso` usando um comando parecido com o anterior, mas precedido de `sudo`. Caso este comando não seja bem sucedido, prossiga para a próxima tarefa. Caso seja, prossiga para a sua sucessora, estudando a próxima tarefa por auto-recreação.

Tarefa 3 *Task 3*

Caso não tenha conseguido executar o comando `sudo`, é porque não está na lista de utilizadores a quem é permitido usar esse comando ou porque o comando não está instalado. Siga perentoriamente os passos seguintes para viabilizar esse comando para o utilizador em uso **(considere que o utilizador que está a utilizar é, sem perda de generalidade, denotado por `my_user`)**:

- Se o comando não estiver instalado (i.e., se obter uma mensagem de `> bash: sudo: command not found`), pode começar por tentar instalá-lo com a

sequência de comandos seguintes:
`> su root` (seguido de palavra-passe do `root`)
`> dnf install sudo`

- Se o `sudo` já estiver instalado, proceda da seguinte forma:
  `> su root` (seguido de palavra-passe do `root`);
  `> nano /etc/sudoers`
  Procure a linha que diz
  `root ALL=(ALL) ALL`
  e adicione a linha
  `my_user ALL=(ALL) ALL`
  imediatamente a seguir. Saia do ficheiro e grave as alterações. Procure perceber o que acabou de fazer, lendo a documentação que antecede as linhas antes mencionadas.

Depois de ter efetuado estes passos, já deve ser capaz de fazer `sudo` com o utilizador normal. Saia do utilizador `root` emitindo a seguinte linha no terminal:
`> exit`

Volte para a tarefa anterior e termine o que não tinha conseguido terminar antes.

Tarefa 4 *Task 4*

Deve ter conseguido criar com sucesso duas diretorias diferentes. **Q3.: As diretorias pertencem ambas ao utilizador com permissões normais?**
☐ Sim, pertencem. ☒ Não, nem pensar.

Q4.: Qual dos seguintes comandos lhe permite ver as informações das diretorias (e ficheiros)?
☐ `lsof` ☐ `ps -u` ☐ `ls` ☐ `cd` ☒ `ls -l`
Nota: tome as providências que achar necessárias para responder às duas questões anteriores corretamente.

Tarefa 5 *Task 5*

Como utilizador normal, entre na diretoria `ThisDirIsMineAlso` (`> cd ThisDirIsMineAlso`) e crie o ficheiro `thisFileIsMine.txt` lá dentro.

Q5.: O que significam as letras `c` e `d` do comando `> cd ThisDirIsMineAlso` que utilizou antes?

`c` _____ `d` _____

Q6.: Conseguiu entrar?
☒ Sim. ☐ Não.

Q7.: Conseguiu criar o ficheiro?
☒ Sim. ☐ Não.

Q8.: Por que é que conseguiu entrar nesta diretoria?

Porque Não tem permissões.

Tarefa 6 *Task 6*

Caso não tenha conseguido criar o ficheiro `thisFileIsMine.txt` na tarefa anterior, tente agora novamente, abrindo o editor `nano` com o comando `sudo`:
`> sudo nano thisFileIsMine.txt`
Escreva `Ola` no documento e guarde-o, carregando em `CTRL+X`, `Y` e `enter`.

Q9.: Quem é o dono do ficheiro criado pelo comando anterior?
☐ O utilizador normal.
☒ O utilizador `root`.
☐ Um magnata do petróleo.

Q10.: A que grupo pertence o ficheiro criado anteriormente?
☒ Ao grupo `root`.
☐ A um grupo diferente do especificado na opção anterior, já que `root` é um utilizador, não um grupo.
☐ Ao grupo `users`.
☐ Ao grupo _____.

2 Alteração de Donos e Grupos

Changing Owners and Groups

É frequente a situação em que um utilizador faz o *login* como `root`, faz algum trabalho na máquina, e só no fim se dá conta de que os ficheiros criados não podem ser acedidos no final.

Tarefa 7 *Task 7*

Se está a seguir bem este guia, está neste momento autenticado como utilizador normal. Escreva `> su root` no terminal e insira a palavra-passe, como pedido.

Q11.: Qual a palavra-passe que precisa inserir neste caso?
☒ A palavra-passe de `root`.
☐ A palavra-passe de utilizador normal.

Quando usa o comando `sudo`, o sistema também lhe pede para inserir uma palavra-passe. **Q12.: Qual a palavra-passe que precisa inserir nesse caso?**
☐ *Dah!,* a palavra-passe de `root`.
☒ A palavra-passe do utilizador normal.

Q13.: O que significa / para que serve o comando `su`?

switch user

Tarefa 8 *Task 8*

Navegue até à diretoria `ThisDirIsMine`, criada no início desta aula. Uma vez lá, crie o ficheiro `file.txt` com o seguinte comando:

```
> echo "Este ficheiro devia ser do
utilizador normal." > file.txt
```

Liste e analise as propriedades do ficheiro que acabou de criar. **Q14.: Quando acaba de criar um ficheiro, este tem permissões de execução por defeito?**
☐ Sim. ☒ Não.

Tarefa 9 *Task 9*

Saia do utilizador `root` (e.g., faça *switch user* outra vez ou `exit`) e tente abrir ou alterar o ficheiro. **Q15.: Consegue aceder ao ficheiro?**
☒ Sim. ☐ É óbvio que não vai dar.

Tarefa 10 *Task 10*

Escreva aqui o comando que lhe permite mudar o dono do ficheiro que criou na tarefa anterior para o do utilizador normal.

sudo chown joaom file.txt

Q16.: Consegue emitir o comando que identificou com o utilizador atual?
☒ Sem espinhas.
☐ Não consigo.

Caso tenha respondido *"Não consigo"*, procure resolver o problema e mudar, efetivamente, o dono do ficheiro para o de um utilizador que não `root`.

Q17.: De uma maneira geral, quem pode mudar o `id` do dono de um ficheiro em Unix?
☐ Só o dono.
☒ Só o `root`.
☐ Qualquer utilizador do sistema.
☐ Só o dono e o `root`.
☐ Todos os membros do grupo do utilizador que é dono do ficheiro.

Tarefa 11 *Task 11*

Depois de mudar o dono, verifique de novo as propriedades do ficheiro, nomeadamente o grupo. Tente saber também qual é o grupo a que pertence o utilizador normal que está a utilizar (e.g., usando `> groups my_user`) e escreva aqui o comando que lhe permite mudar o grupo do ficheiro para o do utilizador.

chgrp joaom file.txt

3 Mudar Permissões de Acesso

Changing Access Rights

Quando emite o comando `> ls -l`, aparecem um total de 9 permissões de acesso associadas a cada ficheiro no sistema de ficheiros. Se simbolizarmos com um 1 ou com um 0 o facto de se ter ou não ter cada uma dessas permissões e os escrevermos ordenadamente, ficamos com sequências parecidas com:

```
own  grp  oth
rwx  rwx  rwx
110  100  000
```

,

em que `own, grp, oth` simbolizam o dono, o grupo e todos os outros; `r, w, e x` simbolizam permissões de leitura (r), escrita (w) e execução (x); e `110 100 000` simboliza uma máscara binária que indica que permissões estão ligadas.

Seguindo o raciocínio anterior, a última linha dita que o ficheiro a que esta máscara binária está associada pode ser aberto para leitura e escrita pelo seu dono, apenas lido pelos membros do grupo estipulado e inacessível para todos os outros.

A máscara binária pode ainda ser escrita em octal, para cada grupo de 3 bits. Neste caso, a máscara incluída antes ficaria:

```
own  grp  oth
rwx  rwx  rwx
 6    6    0
```

.

Para mudar os permissões, basta então correr o comando `> chmod` com os permissões especificados em octal (por exemplo) para o ficheiro em questão. **Q18.: Quem é que pode mudar as permissões de um ficheiro ou diretoria?**
☐ Só o dono.
☐ Só o `root`.

☐ Qualquer utilizador do sistema.
☒ Só o dono e o `root`.
☐ Todos os membros do grupo do utilizador que é dono do ficheiro.

Tarefa 12 *Task 12*

Dentro da diretoria `ThisDirIsMine`, crie o ficheiro `script.sh`, abrindo-o com o `nano`, e insira lá o conteúdo incluído a seguir:

```
#!/bin/bash
echo "Hello.  This is my first bash
script...  or maybe not."
```

Tarefa 13 *Task 13*

Verifique as permissões do ficheiro. **Q19.: Consegue executá-lo tal como está?**
☒ Não, nem pensar.
☐ *No prob... Piece of cake.*

Tome as medidas que achar necessárias para tornar este ficheiro executável para o dono, e execute-o.

chmod 722 script.sh

Tarefa 14 *Task 14*

Analise (ou execute) a seguinte cadeia de comandos e responda à questão que lhe é colocada a seguir:

```
> su root
```
(inserir palavra-passe de `root`)
user root

```
> echo "superman" > segredo.txt
```
escreve o stdout para fich

```
> exit
```
sai root

```
> sudo chown my_user segredo.txt
```
muda o dono para user

(inserir palavra-passe de `my_user`)

```
> chmod 000 segredo.txt
```
tirar permissões aos users

```
> chmod u+r segredo.txt
```
damos ao user a permissão de leitura

Q20.: Quem pode ler o ficheiro `segredo.txt`?
☐ Ninguém.
☐ Qualquer utilizador.
☐ Só o `my_user`.
☐ Só o `root`.
☒ O `root` e o `my_user`.

Nota: não se esqueça que `my_user` denota o utilizador com privilégios normais definidos no seu sistema.

Considere comentar o que faz cada um dos comandos indicados em cima nas linhas seguintes:

Comando 1: _____

Comando 2: _____

Comando 3: _____

Comando 4: _____

Comando 5: _____

Comando 6: _____

Tarefa 15 *Task 15*

Como utilizador normal, tente ver o conteúdo do ficheiro `/etc/shadow`.
Q21.: Foi bem sucedido nesta tarefa?
☐ Sim. ☒ Não.

Q22.: Ainda se lembra do que guarda esse ficheiro?
☐ Os nomes de utilizadores do sistema.
☒ O *hash* das palavras-passes dos utilizadores.
☐ As configurações do sistema.
☐ As permissões de acesso aos vários ficheiros no sistema.

Q23.: Quais as permissões de acesso que o ficheiro `/etc/shadow` tem?
☐ 000 ☒ 600 ☐ 060 ☐ 006 ☐ 400 ☐ 040

Q24.: Porque é que, mesmo com as permissões que assinalou antes, consegue abrir o ficheiro para leitura e escrita quando está como `root`?
☐ O Professor está enganado. Não se consegue.
☐ Conseguimos porque as permissões aplicam-se sempre a utilizadores normais. O `root` pode sempre tudo, quer seja o dono ou não de determinado ficheiro.

Ainda como utilizador normal, tente ler o conteúdo do ficheiro `/etc/passwd`. **Q25.: Encontrou informação útil neste ficheiro?**
☐ Apenas números sem significado.
☒ Nomes de utilizador, `id`, grupos e `shells` que são lançadas para cada utilizador.

Tarefa 16 *Task 16*

Por curiosidade, procure saber qual a opção do comando `ls` que lhe permite ver informação relativa ao contexto de segurança implementado pelo mecanismo de controlo de acesso mandatório conhecido por `SeLinux`.

> ls -Z

4 Identificação Associada a Processos

User Identification Associated to Processes

Cada utilizador do sistema operativo tem um IDentificador (ID) associado. Quando executa um processo, esse ID pode ficar associado a essa execução também, ou mudar para o ID do dono do programa executado. Nesse contexto, temos dois identificadores diferentes que importa referir: o *Effective User ID* (EUID), que é o ID com que corre o processo, e o *Real User ID*, que é o ID do utilizador que executa o processo.

Tarefa 17 *Task 17*

Emita o comando `id` no terminal para saber o ID dos utilizadores que conhece no sistema:

```
> id my_user
> id root
```

Q26.: Sem olhar, indique aqui o ID do utilizador

`root:` 0

Tarefa 18 *Task 18*

Procure saber o que faz o comando `ps`, emitindo o comando no sistema e explorando a sua documentação.

Q27.: Quando o executa como utilizador normal, quantos processos vê em execução?

☐ 0. ☐ 1. ☒ 2. ☐ 3.

Tarefa 19 *Task 19*

Mude de utilizador para `root` (`> su root`) e crie o ficheiro `program.c` com o seguinte conteúdo:

```c
#include <unistd.h>
#include <stdio.h>

void main(){
  while(1){
    sleep(10);
    printf("Ja passaram mais 10 segundos.\n");
  }
}
```

Compile este ficheiro (`> cc program.c`) e execute-o (`> ./aout`) para se certificar que fica a funcionar bem. Verifique também os seus privilégios de execução (`x`) emitindo o comando `> ls -la` na diretoria em que compilou o programa.

Q28.: Quem é o dono do programa?

☒ O `root`. ☐ O `my_user`. ☐ O `bitcho-papao`.

Q29.: Quem pode executar o programa que criou?

☒ Só o `root`.
☐ O `root` e o `my_user`.
☐ Qualquer utilizador.

Tarefa 20 *Task 20*

Ainda como `root`, execute o programa com `> ./aout &` (o `&` é para colocar o programa a correr em segundo plano). Procure a opção do comando `ps` que mostra detalhes dos processo que estão a correr, e use-a para ver qual o UID associado à execução do programa que criou antes.

Nota: quando quiser terminar o processo, emita o seguinte comando no terminal:

```
> killall a.out
```

Tarefa 21 *Task 21*

Volte ao utilizador normal e corra o programa antes criado novamente. **Q30.: Qual o *effective* UID associado a esta execução?**

☐ É o UID do utilizador normal.
☒ É o UID do dono do programa, neste caso o `root`.

Q31.: Qual o *real* UID associado a esta execução?

☐ É o UID do utilizador normal.
☒ É o UID do dono do programa, neste caso o `root`.

Tarefa 22 *Task 22*

Execute o comando incluído a seguir para ajustar a *flag* do `setuid` a 1:

```
> sudo chmod u+s a.out
```

(palavra-passe do utilizador normal)

Q32.: Por que é que que teve de preceder o comando `chmod` com `sudo` para que fosse bem sucedido?

Verifique o sucesso da empreitada emitindo o comando:

```
> ls -l
```

Q33.: Como é que sabe se foi bem sucedido?

☐ Sei, porque o ficheiro `a.out` aparece com uma cor diferente.
☐ Sei, porque aparece um `s` no lugar do `x` na secção de privilégios referente a `others`.
☐ Sei, porque aparece um `s` no lugar do `x` na secção de privilégios referente a `users`.

Tarefa 23 *Task 23*

Execute novamente o programa (`> ./a.out &`) e verifique as propriedades do processo com `> ps -l`

Q34.: Conseguiu verificar as propriedades do processo?
☐ Estranhamente, parece que não tenho o processo a correr. :S
☐ O processo está a correr, mas o comando anterior só mostra processos com *effective* UID igual àquele que estou a executar, e não mostra o `a.out`.
☐ Talvez se executar o comando `> ps -la` ou `> ps -lA`.

Q35.: Qual o *effective* UID associado a esta execução?
☐ É o UID do utilizador normal.
☐ É o UID do dono do programa, neste caso o `root`.

Q36.: Qual o *real* UID associado a esta execução?
☐ É o UID do utilizador normal.
☐ É o UID do dono do programa, neste caso o `root`.

Q37.: Investigue se o comando `chmod 4777` também servia para colocar um programa executável e com `setuid` a 1.
☐ Sim, dava, e faz perfeito sentido na minha cabeça.
☐ Não, não dava. Nem fazia sentido dar.

Tarefa 24 *Task 24*

O comando `passwd` serve para mudar as palavras-passe dos utilizadores do sistema e precisa aceder ao ficheiro `/etc/shadow`. **Q38.: Com que *effective* UID tem de correr este programa para efetuar a sua tarefa com sucesso?**
☐ `root` ☐ `my_user`

Escreva `whereis passwd` no terminal para encontrar as diretorias onde o programa está instalado, e verifique os seus privilégios com `ls -l`. Veja se o que descobre está de acordo com o que respondeu antes.

Tarefa 25 *Task 25*

Considere que criava um ficheiro com chaves RSA que só queria utilizar quando se autenticava no sistema com o seu utilizador normal. **Q39.: Para além das cifrar, que outra medida deveria tomar para as proteger minimamente no sistema?**

Talvez devesse emitir o comando 400 (SÓ LER) para proteger mininimamente o ficheiro no sistema.