



Universidade do Minho
Escola de Engenharia

MESTRADO INTEGRADO EM ENGENHARIA DE TELECOMUNICAÇÕES E INFORMÁTICA

SEGURANÇA EM REDES DE COMPUTADORES

ACCESS CONTROL

TRABALHO PRÁTICO Nº2

Grupo 9:

David José Ressurreição Alves - A79625

Cláudia Cristiana de Amorim Dias - A78232

Guimarães, 15 de Março de 2019

Índice

1	Introdução	4
2	O modelo Bell-LaPadula	5
2.1	Contextualização histórica	5
2.2	Características do modelo	5
2.3	Desvantagens do modelo	5
2.4	MAC- <i>Mandatory Access Control</i>	6
2.4.1	Conceção da <i>Lattice</i>	6
2.5	DAC- <i>Discretionary Access Control</i>	11
2.6	Possibilidade de fraude por parte de um aluno	12
3	Processo automático de implementação do modelo	13
4	Conclusão	19
5	Referências	20

Lista de Figuras

1	<i>Lattice</i> do modelo de controlo de acesso desenvolvido.	6
2	Esquema das permissões dos níveis de acesso.	10
3	Esquema das permissões do exemplo criado	13
4	Adicionar utilizador aluno.	14
5	Adicionar utilizador professor.	14
6	Adicionar utilizador segurança.	14
7	Criação dos grupos.	14
8	Associação grupo-utilizador.	14
9	Definição da ACL para o utilizador segurança.	15
10	Definição da ACL para o utilizador professor.	15
11	Definição da ACL para o utilizador reitor.	16
12	Ficheiro de pauta-exemplo.	16
13	Verificação das permissões da pauta-exemplo.	17
14	Verificação de que segurança não consegue ler nada do professor.	17
15	Pauta do professor aberta pelo Reitor.	17
16	Simulação de ataque.	18
17	Verificação do ataque efetuado.	18

Lista de Tabelas

1	Entidades definidas para o nível <i>Strictly Confidential</i>	7
2	Entidades definidas para o nível <i>Confidential</i>	8
3	Entidades definidas para o nível <i>Public</i>	9
4	Matriz DAC.	11

1. Introdução

O controlo de acesso numa organização é algo imperativo no que diz respeito à proteção dos dados privados da mesma. Desenvolveram-se por isso, ao longo do tempo, vários mecanismos de acesso a dados secretos das organizações. Precisamente em relação ao controlo de acesso em organizações, implementaram-se os processos de autenticação (determina quem pode aceder ao sistema da organização), autorização (define quais os direitos e permissões de quem está dentro da organização) e auditoria (permite, através da análise do uso dos dados da organização, revelar a origem e tipologia da utilização dos mesmos).

No âmbito da unidade curricular de Segurança em Redes de Computadores, foi-nos por isso proposto que desenvolvêssemos um modelo de controlo de acesso, num contexto universitário, baseado no modelo de Bell-LaPadula. Será discutido se existe a possibilidade de um aluno enganar um professor e apresentado um possível modo de implementação deste modelo.

2. O modelo Bell-LaPadula

2.1. Contextualização histórica

Este modelo foi inicialmente desenvolvido para o departamento de defesa dos Estados Unidos, com o objetivo de controlar o acesso a documentos internos, criando assim uma estrutura de segurança multinível (MLS, do inglês, *Multi Level Security*), para que se pudessem manter os dados em segredo e compartilhá-los apenas com quem tivesse autorização para recebê-los, concentrando-se assim mais na confidencialidade, não havendo nele uma distinção entre proteção e segurança [1]. No caso concreto do departamento de defesa dos Estados Unidos, os níveis de confidencialidade definidos eram: *Unclassified* (nível mais baixo), de seguida, *Classified*, *Secret* e *Top Secret* (nível mais alto).

2.2. Características do modelo

O modelo Bell-LaPadula caracteriza-se fundamentalmente pelo facto de não permitir que algum membro pertencente a um dado nível de confidencialidade possa escrever para níveis abaixo dele, só podendo escrever para níveis superiores a ele (propriedade *write up* ou *don't read up*), bem como que, qualquer membro pertencente a um determinado nível de segurança só possa ler dados de níveis abaixo dele (propriedade *read down* ou *don't write down*). Além disso, este modelo define através de uma matriz de acesso, as permissões que cada entidade tem para cada objeto de dados (propriedade de *segurança descricionária*).

2.3. Desvantagens do modelo

Apesar de ser funcionalmente vantajoso em termos de confidencialidade este modelo revela algumas desvantagens [2], tais como:

- Complexidade elevada na implementação do modelo na vida real;
- Focado praticamente apenas na confidencialidade, não garantindo a integridade dos documentos;
- Dificuldade em lidar com o facto dos dados poderem sofrer alterações em relação ao nível de confidencialidade a que pertencem (sobretudo porque devido à propriedade *write up*, os dados tendem a subir de nível de confidencialidade, ao longo do tempo).

2.4. MAC-Mandatory Access Control

O *Mandatory Access Control* (em português, controlo de acesso obrigatório), é um tipo de política de controlo de acesso, baseado em rótulos (*labels*), que são definidos pelo sistema (administrador), na qual as entidades (*subjects*) e os objetos de dados recebem uma determinada autorização de segurança (*label*) [3].

As várias labels são então colocadas num diagrama (*lattice*), que define os níveis de segurança que um objeto de dados pode ter, bem como os que uma entidade pode ter acesso.

Este tipo de política deve-se ao facto de, numa organização, o acesso aos dados não ser apenas destinado a um indivíduo, mas sim a vários e com diferentes níveis de acesso.

O MAC torna-se portanto essencial em ambientes onde existe um elevado grau de confidencialidade dos dados. Alguns exemplos de onde a política MAC é implementada são: SELinux e Trusted Solaris.

2.4.1. Conceção da *Lattice*

Para este trabalho prático foi-nos pedida a elaboração da *lattice* de níveis de segurança, na qual são definidos os seguintes níveis de acesso: **Strictly Confidential (SC)**, que é o nível mais secreto, e deve ser colocado no topo da *lattice*, no nível inferior a este, o nível **Confidential (C)** e na base da *lattice*, encontra-se o nível menos secreto de todos, o **Public (P)**.

Tal como é dito no enunciado do problema proposto, à entidade "Alunos", corresponderá a *label* $(C, \{AS\})$ na *lattice*, enquanto que aos "Professores", corresponderá a *label* $(C, \{AS, ScS\})$.

De seguida, apresentamos na figura 1, a *lattice* concebida pelo grupo e posteriormente a justificação da relação *label*-entidades que o grupo associou no âmbito da temática do problema.

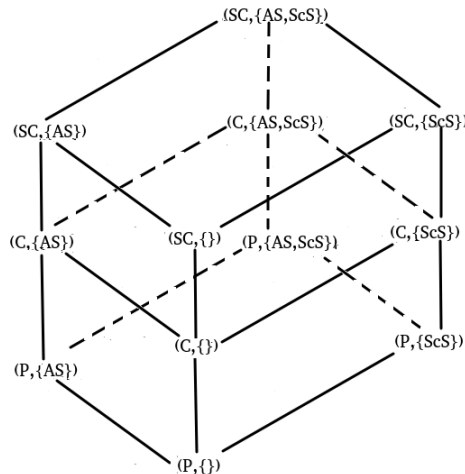


Figura 1: *Lattice* do modelo de controlo de acesso desenvolvido.

Para cada nível de acesso, o grupo apresenta nas tabelas 1 a 3 a atribuição das relações *label*-entidade assim como a justificação das respetivas escolhas:

Tabela 1: Entidades definidas para o nível *Strictly Confidential*.

<i>Label</i>	Entidade
(SC,{AC,ScS})	Reitoria
(SC,{AC})	Funcionários administrativos dos serviços académicos
(SC,{ScS})	Diretores dos departamentos de investigação
(SC,{})	Funcionários da secretaria de cada departamento

A explicação das relação *label*-entidade é a seguinte:

- **(SC,{AC,ScS})**: Nesta *label* escolheu-se a entidade "Reitoria" uma vez que esta é responsável pela gestão da universidade necessitando de gerir informação extremamente confidencial e ler de todos os níveis existentes;
- **(SC,{AC})**: O sujeito "Funcionários administrativos dos serviços académicos" é responsável, por exemplo, pela verificação das pautas lançadas pelos professores. Deste modo, faz todo o sentido que se encontrem neste nível, tendo a permissão de ler as pautas e validar;
- **(SC,{ScS})**: Relativamente à presente *label*, atribuímos os "Diretores dos departamento de investigação" uma vez que estes supervisionam o trabalho e gerem os pedidos efetuados pelos investigadores como por exemplo pedidos de bolsas de investigação, material, etc. Um investigador pode necessitar de um determinado material escrevendo o pedido para os Diretores, estes por sua vez leem o pedido e geram essa informação;
- **(SC,{})**: No que concerne esta *label* decidiu-se colocar a entidade "Funcionários da secretaria de cada departamento" assim pode ler os pedidos que os Professores Convidados efetuam e reservar os recursos necessários, como por exemplo salas.

Tabela 2: Entidades definidas para o nível *Confidential*.

<i>Label</i>	Entidade
$(C, \{AC, ScS\})$	Professores/Professores efetivos
$(C, \{AC\})$	Alunos
$(C, \{ScS\})$	Professores investigadores
$(C, \{\})$	Professores convidados

A explicação das relação *label*-entidade é a seguinte:

- $(C, \{AC, ScS\})$: Aqui atribuímos a entidade "Professores/Professores Efetivos", uma vez que o mesmo era previamente proposto no enunciado deste trabalho prático;
- $(C, \{AC\})$: Para esta *label*, atribuímos a entidade de "Alunos", dado que o mesmo era proposto no enunciado do trabalho prático;
- $(C, \{ScS\})$: Definimos aqui, a entidade "Professores investigadores", dado que estes são um tipo de investigadores que tem que apresentar (escrever) o trabalho que desenvolve, junto do diretor do seu departamento científico. Por sua vez estes também têm a possibilidade de ler o trabalho dos Investigadores-bolseiros.
- $(C, \{\})$: Atribuímos a entidade "Professores convidados", uma vez que, este tipo de professores, apenas se poderá encontrar na universidade, ocasionalmente, tendo funções que não obrigam à avaliação de alunos nem ao conhecimento dos dados internos da universidade (por exemplo, poderá considerar-se um professor convidado, alguém que visite a universidade para dar uma palestra sobre determinado assunto).

Tabela 3: Entidades definidas para o nível *Public*.

<i>Label</i>	<i>Entidade</i>
$(P, \{AC, ScS\})$	Seguranças
$(P, \{AC\})$	Funcionários de limpeza dos serviços académicos
$(P, \{ScS\})$	Investigadores bolsseiros
$(P, \{\})$	Alunos visitantes

A explicação das relação *label*-entidade é a seguinte:

- $(P, \{AC, ScS\})$: Aqui definimos a entidade "Seguranças", dado que estes precisam de aceder a dados públicos da universidade (como por exemplo, a localização dos vários departamentos) e precisam de registar (escrever) para ambos os serviços (académicos e científicos), as horas em que vigiaram determinado departamento (por exemplo);
- $(P, \{AC\})$: Definimos para esta *label*, a entidade "Funcionários de limpeza dos serviços académicos", porque têm que ter acesso aos dados públicos dos serviços académicos (como por exemplo, a localização dos departamentos nos quais vão efectuar serviço), e têm que poder escrever para os mesmos o registo dos seus serviços efetuados (por exemplo).
- $(P, \{ScS\})$: Em relação a esta *label* definimos o sujeito "Investigadores bolsseiros", uma vez que os investigadores-bolsseiros estão presentes na universidade precisando apenas de terem acesso aos dados públicos dos serviços científicos, no entanto precisam de escrever os seus trabalhos para os professores investigadores que os acompanham, sendo que, alternativamente, também podem tornar públicos, os seus trabalhos.
- $(P, \{\})$: Associámos esta *label* com a entidade "Alunos visitantes", dado que este tipo de alunos, apenas está presente na universidade esporadicamente, não precisando de ter acesso a dados internos da mesma (um exemplo, serão os alunos que pretendam ser estudantes da universidade e a visitam de modo a conhecê-la).

De acordo com a *lattice* desenvolvida, podemos esquematizar aquilo que cada entidade, em cada nível de acesso pode fazer. Esse esquema está representado na figura 2.

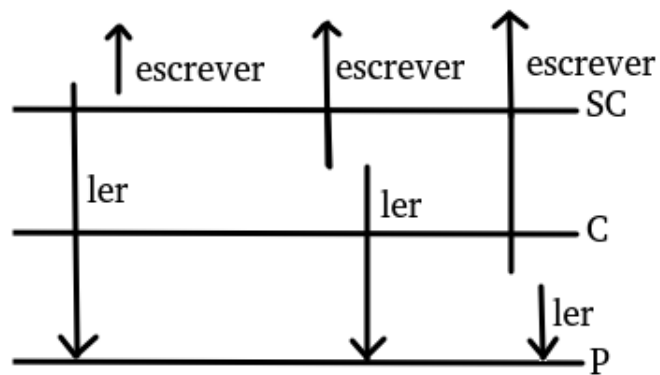


Figura 2: Esquema das permissões dos níveis de acesso.

2.5. DAC-*Discretionary Access Control*

No *Discretionary Access Control* (em português, Controlo de Acesso Discricionário), o proprietário do objeto de dados é quem especifica quais as entidades que podem aceder ao mesmo [3]. Sendo que para um determinado utilizador poder aceder a um determinado objeto, primeiro é verificado quais as permissões que esse mesmo utilizador tem para esse objeto, numa lista de controlo de acesso (CLA, do inglês, *Control Access List*).

Este tipo de política de acesso, está presente na maioria dos sistemas operativos atuais, como por Windows, Linux e Macintosh.

Apesar de ser fácil de implementar, o DAC, tem várias desvantagens [4], tais como:

- Vulnerável a ataques via *trojan* (cavalo de tróia);
- Capacidade e manutenção reduzidas da lista de controlo de acesso.

Para aplicar este tipo de política no trabalho prático, decidimos escolher como entidades, os professores e os alunos, e como objetos de dados, escolhemos as pautas das notas finais dos alunos numa determinada unidade curricular, e os testes e trabalhos realizados por cada aluno na dita unidade curricular, enquanto que no que diz respeito às permissões, temos a permissão de escrita (designada pela letra E), e a permissão de leitura, (designada pela letra L). Na figura seguinte podemos observar a matriz que combina as permissões das entidades professores e alunos, em relação ao objeto de dados referido anteriormente.

Tabela 4: Matriz DAC.

Objeto Entidade	Testes/Trabalhos	Nota Final
Professor	L	L/E
Aluno	L/E	L

2.6. Possibilidade de fraude por parte de um aluno

Segundo o modelo que elaborámos, os professores, uma vez que se encontram num grau hierarquicamente superior (*label* - $(C, \{AS, ScS\})$) podem ler e escrever para todas as entidades que se encontram no nível *Confidential* (C). Tal não se verifica para a entidade "Alunos" uma vez que está localizada num nível hierárquico inferior (*label* - $(C, \{AS\})$) tendo apenas permissão de leitura em níveis de segurança inferiores (*Public*) e de escrita para superiores ($(SC, \{AS\})$).

Contudo, podemos observar que o professor, vai escrever a pauta com as notas dos alunos para um nível de acesso superior, (para os funcionários administrativos dos serviços académicos, $(SC, \{AC\})$, afim de estes validarem as notas e de seguida, delegam para a os funcionários da secretaria do departamento da unidade curricular em causa, que têm a *label* $(SC, \{\})$, a divulgação pública das notas dos alunos (que pode ser feita através do portal académico, por exemplo), ficando as mesmas sob o domínio público, de forma a que os estudantes possam ter acesso às mesmas.

Apesar disto, segundo este modelo existem falhas que podem levar a que um aluno possa alterar a pauta das notas, isto porque, o facto de as notas terem de ficarem sob domínio público, para os alunos poderem ter acesso às mesmas, e uma vez que quer seja um aluno ou outra pessoa qualquer, pode ler e escrever sobre a pauta das notas, isto faz com que, o aluno através de um ataque à base de dados do portal académico (por exemplo), possa adulterar essa mesma pauta. Além disso, ainda existe outro tipo de ataque que poderia ocorrer, chamado *blindwrite*, segundo o qual, o aluno uma vez que as notas são primeiramente publicadas para os funcionários administrativos dos serviços académicos (e visto que os alunos podem apenas escrever para qualquer documento que se encontre nessa *label*), isto faz com um aluno possa escrever dados para a pauta das notas, apesar de não poder ainda estar a ver o seu conteúdo, sendo certo que este seria um ataque complexo de executar caso se quisesse alterar uma nota de aluno concreto, no entanto seria bastante simples de poder apagar todos os dados dessa mesma pauta.



3. Processo automático de implementação do modelo

Nesta secção, tal como pedido no enunciado do problema, iremos elaborar um processo automático de implementação do modelo Bell-LaPadula desenvolvido para o contexto do problema proposto, numa infraestrutura TIC.

Do modelo de controlo de acesso desenvolvido neste trabalho prático, podemos observar que, caso uma entidade se encontre num nível de acesso superior ou igual ao nível do objeto em análise, então essa entidade tem permissão para ler esse objeto, caso contrário, só terá permissão para escrever.

Para implementar este modelo usámos o sistema operativo Ubuntu, e os comandos *getfacl* (para obter as ACL de cada ficheiro), e *setfacl* (para definir as ACL de cada ficheiro).

Tendo por base, a *lattice* desenvolvida na secção anterior e considerando também um contexto universitário, decidimos definir um cenário no qual existiriam três níveis de acesso, SC (*Strictly Confidential*), C (*Confidential*) e P (*Public*), os quais consideramos como grupos de utilizadores e para esses grupos, os utilizadores criados foram: reitor (que pertence ao grupo de utilizadores SC), professor (que pertence ao grupo de utilizadores C) e segurança (que pertence ao grupo de utilizadores P). Tal como na *lattice* desenvolvida anteriormente, também aqui SC, vai ser o grupo de utilizadores com permissões mais elevadas e P, o grupo com menos permissões. O esquema para este exemplo seria o presente na figura 3.

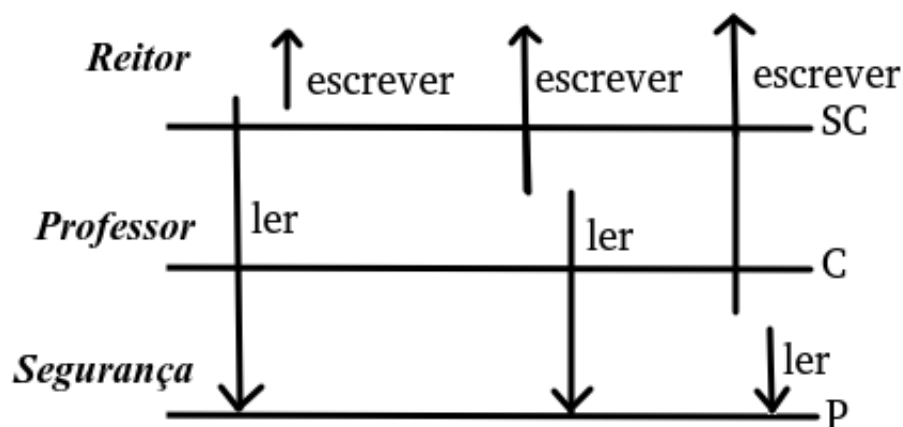


Figura 3: Esquema das permissões do exemplo criado

Posto isto, procedemos implementação do esquema, a qual será descrita nos seguintes procedimentos:

1. Criámos utilizadores com os nomes de reitor, professor e segurança, tal como se pode observar nas figuras 4, 5 e 6.

```
sudo adduser --home /reitor reitor
```

Figura 4: Adicionar utilizador aluno.

```
adduser --home /professor professor
```

Figura 5: Adicionar utilizador professor.

```
sudo adduser --home /seguranca seguranca
```

Figura 6: Adicionar utilizador segurança.

2. Criámos os grupos de utilizadores (SC, C e P), como se observa na figura 7.

```
sudo groupadd SC  
sudo groupadd C  
sudo groupadd P
```

Figura 7: Criação dos grupos.

3. Adicionámos cada utilizador ao grupo respetivo, tal como se pode ver na figura 8.

```
sudo usermod -a -G SC reitor  
sudo usermod -a -G C professor  
sudo usermod -a -G P seguranca
```

Figura 8: Associação grupo-utilizador.

4. Definimos, através do comando *setfacl*, a ACL para o diretório principal do utilizador "segurança", e de seguida através do comando *getfacl* verificámos a mesma, como se mostra na figura 9.

```
ubuntu@ubuntu:/home$ sudo setfacl -d -R -m g:sc:rx segurança/
ubuntu@ubuntu:/home$ sudo setfacl -d -R -m g:c:rx segurança/
ubuntu@ubuntu:/home$ sudo setfacl -d -R -m g:p:rw segurança/
ubuntu@ubuntu:/home$ getfacl segurança/
# file: segurança/
# owner: segurança
# group: p
user::rwx
group::r-x
group:sc:r--
group:c:r--
group:p:rw-
mask::rwx
other::r-x
default:user::rwx
default:group::r-x
default:group:sc:r-x
default:group:c:r-x
default:group:p:rw-
default:mask::rwx
default:other::r-x
```

Figura 9: Definição da ACL para o utilizador segurança.

5. Definimos, através do comando *setfacl*, a ACL para o diretório principal do utilizador "professor", e de seguida através do comando *getfacl* verificámos a mesma, como se mostra na figura 10.

```
ubuntu@ubuntu:/home$ sudo setfacl -d -R -m g:sc:rx professor/
ubuntu@ubuntu:/home$ sudo setfacl -d -R -m g:c:rw professor/
ubuntu@ubuntu:/home$ sudo setfacl -d -R -m g:p:wx professor/
ubuntu@ubuntu:/home$ getfacl professor/
# file: professor/
# owner: professor
# group: c
user::rwx
group::r-x
group:sc:r-x
group:c:rw-
group:p:-wx
mask::rwx
other::r-x
default:user::rwx
default:group::r-x
default:group:sc:r-x
default:group:c:rw-
default:group:p:-wx
default:mask::rwx
default:other::r-x
```

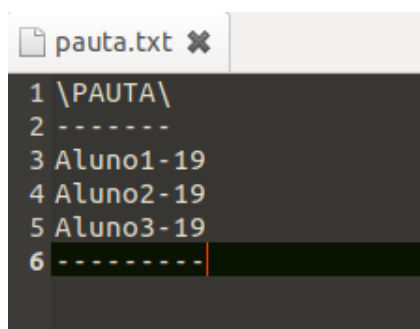
Figura 10: Definição da ACL para o utilizador professor.

6. Definimos, através do comando **setfacl**, a ACL para o diretório principal do utilizador "reitor", e de seguida através do comando **getfacl** verificámos a mesma, como se mostra na figura 11. Além disso é de notar a inclusão da permissão de execução para aqueles grupos que apenas seriam de escrita ou de apenas leitura, isto deve-se ao facto de para poder escrever ou ler para ficheiros dentro de pastas, tem que ser ter permissão de execução ("x"), bem como o facto de no comando **setfacl**, se ter usado o argumento "-d", para que novos ficheiros criados dentro do diretório, herdem automaticamente as permissões atribuídas ao mesmo.

```
ubuntu@ubuntu:/home$ sudo setfacl -d -R -m g:sc:rwX reitor/
ubuntu@ubuntu:/home$ sudo setfacl -d -R -m g:c:wx reitor/
ubuntu@ubuntu:/home$ sudo setfacl -d -R -m g:p:wx reitor/
ubuntu@ubuntu:/home$ getfacl reitor/
# file: reitor/
# owner: reitor
# group: sc
user::rwX
group::r-x
group:sc:rw-
group:c:-w-
group:p:-w-
mask::rwX
other::r-x
default:user::rwX
default:group::r-x
default:group:sc:rwX
default:group:c:-wx
default:group:p:-wx
default:mask::rwX
default:other::r-x
```

Figura 11: Definição da ACL para o utilizador reitor.

7. Fazendo *login* com a conta do professor, criámos um ficheiro de texto com a pauta de hipotéticas notas de alunos-exemplo, como se vê na figura 12.



```
pauta.txt x
1 \PAUTA\
2 -----
3 Aluno1-19
4 Aluno2-19
5 Aluno3-19
6 -----
```

Figura 12: Ficheiro de pauta-exemplo.

8. Verificámos que as permissões foram definidas automaticamente, e como se pode ver na seguinte figura, o ficheiro criado herda do grupo a que o utilizador pertence, as permissões anteriormente definidas.

```
professor@ubuntu:~$ getfacl pauta.txt
# file: pauta.txt
# owner: professor
# group: p
user::rw-
group::r-x                    #effective:r--
group:sc:r-x                  #effective:r--
group:c:rwX                    #effective:rw-
group:p:-wX                    #effective:-w-
mask::rw-
other::r--
```

Figura 13: Verificação das permissões da pauta-exemplo.

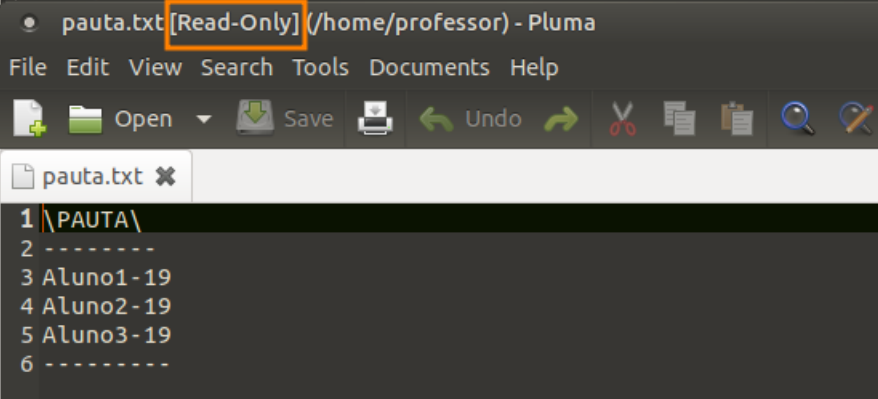
9. Verificámos que o utilizador "segurança", estando no nível de acesso abaixo do professor, não pode ler a pauta (nem sequer consegue ler nada do diretório do professor).

```
seguranca@ubuntu:/home$ ls
professor reitor seguranca ubuntu
seguranca@ubuntu:/home$ cd professor/
bash: cd: professor/: Permission denied
seguranca@ubuntu:/home$
```

Figura 14: Verificação de que segurança não consegue ler nada do professor.

10. Fazendo *login* com a conta do reitor, verificámos de que ele, apenas consegue ler os ficheiros do professor, neste caso em concreto a pauta que o professor elaborou.

```
reitor@ubuntu:/home/professor$ ls
Desktop Downloads pauta.txt Public Templates
Documents Music Pictures snap Videos
reitor@ubuntu:/home/professor$ pluma pauta.txt
```



The screenshot shows the Pluma text editor window titled "pauta.txt [Read-Only] (/home/professor) - Pluma". The editor displays the following text:

```
1 \PAUTA\
2 -----
3 Aluno1-19
4 Aluno2-19
5 Aluno3-19
6 -----
```

Figura 15: Pauta do professor aberta pelo Reitor.

11. Simulámos um ataque *blindwrite* por parte do segurança, no qual este, apesar de não poder ver o conteúdo da pauta, pode escrever para ela, e neste exemplo em concreto apagou todo o seu conteúdo e substituiu o mesmo por uma mensagem..

```
seguranca@ubuntu:/home$ echo pauta_hackeada > /home/professor/pauta.txt
```

Figura 16: Simulação de ataque.

12. Verificámos de que o ataque ocorreu com sucesso.

```
professor@ubuntu:~$ ls
Desktop  Downloads  pauta.txt  Public  Templates
Documents Music      Pictures  snap    Videos
professor@ubuntu:~$ cat pauta.txt
pauta_hackeada
```

Figura 17: Verificação do ataque efetuado.

Uma das principais vantagens deste método implementado, é a de que, a qualquer momento é fácil adicionar um utilizador a um grupo, herdando este, automaticamente, as permissões relativas a esse grupo.

4. Conclusão

Após concluído este trabalho prático, observámos que de facto, este modelo (Bell-LaPadula), atualmente não se adequa em termos de segurança, dado que devido ao crescente número de vulnerabilidades presentes nos sistemas operativos, ou nas redes de computadores, existem várias formas de se poderem escalar níveis de segurança, comprometendo não só a confidencialidade dos dados (que é o que este modelo mais protege), como também a integridade dos mesmos.

Contudo, este trabalho prático também serviu para aumentar-mos o nosso conhecimento na área da segurança informática, bem como melhorar-mos as nossas capacidades de implementação desses mesmos conhecimentos.

5. Referências

- [1] **"Landwehr, Carl (setembro de 1981). «Formal Models for Computer Security»**
- [2] **"Bell-LaPadula - Computer Security - A brief look", Sites.google.com,**
Disponível em: <https://sites.google.com/site/cacsolin/bell-lapadula>. [Acedido em 12 de março de 2019].
- [3] **"Discretionary Access Control vs Mandatory Access Control - jimmyxu101", Sites.google.com,**
Disponível em: <https://sites.google.com/site/jimmyxu101/concepts/accesscontrol>. [Acedido em 13 de março de 2019].
- [4] **"What is Discretionary Access Control (DAC)? - Definition from Techopedia", Techopedia.com,**
Disponível em: <https://www.techopedia.com/definition/229/discretionary-access-control-dac>. [Acedido em 13 de março de 2019].