



Universidade do Minho
Escola de Engenharia

Mestrado em Engenharia de Telecomunicações e Informática

Unidade Curricular de Cibersegurança

Docente: Henrique Santos

TP 2: Modelação de Controlo de Acesso

Bárbara Fonseca PG53677

Bruno Santos A93087

Camila Pinto PG53712

Eduarda Dinis PG53793

Gonçalo Dias PG53833

Guimarães, fevereiro de 2024

Índice de conteúdos

Índice de conteúdos	ii
Lista de Tabelas	iii
Lista de Figuras	iv
Lista de acrónimos e siglas	v
1. Introdução	1
2. Modelo Bell-LaPadula	2
3. Conceção da <i>Lattice</i>	4
4. Possibilidade da ocorrência de fraude	9
5. Processo Automático de Implementação	10
6. Conclusão.....	16
Referências	18

Lista de Tabelas

Tabela 1. Entidades do nível <i>Strictly Confidential</i>	5
Tabela 2. Entidades do nível <i>Confidential</i>	6
Tabela 3. Entidades do nível <i>Public</i>	7
Tabela 4. Tabela de permissões.	8

Lista de Figuras

Figura 1. Conceção da <i>lattice</i>	4
Figura 2. Criação dos níveis de acesso.	10
Figura 3. Associação dos utilizadores aos grupos de acesso.	10
Figura 4. Estabelecimento das permissões e verificação para o reitor.	11
Figura 5. Estabelecimento das permissões e verificação para o professor.	11
Figura 6. Estabelecimento das permissões e verificação para o funcionário.....	12
Figura 7. Exemplo de teste: funcionário não consegue aceder ao conteúdo do professor.	12
Figura 8. Exemplo de teste: professor consegue aceder ao conteúdo do funcionário.	12
Figura 9. Criação do ficheiro <i>notasciber.txt</i>	13
Figura 10. Conteúdo do ficheiro <i>notasciber.txt</i>	13
Figura 11. Permissões associadas ao ficheiro.	13
Figura 12. Exemplo de teste: reitor tem acesso à leitura do ficheiro.	14
Figura 13. Exemplo de teste: reitor não tem acesso de escrita no ficheiro.	14
Figura 14. Tentativa de ataque <i>blindwrite</i>	15
Figura 15. Sucesso na tentativa de ataque.	15

Lista de acrónimos e siglas

ACL	Access Control List
AS	Academic Services
DAC	Discretionary Control Access
MAC	Mandatory Control Access
RBAC	Role Based Control Access
RO	Read Only
ScS	Scientific Services
WO	Write Only

1.Introdução

O presente relatório está a ser desenvolvido no âmbito da Unidade Curricular de Cibersegurança do 2º semestre do 1ºano do curso de Mestrado em Engenharia Telecomunicações e Informática.

O controlo de acesso é uma componente vital para a segurança em redes, uma vez que visa garantir a segurança, confidencialidade e integridade de sistemas de informação e recursos. Os métodos utilizados para o controlo incluem autenticação, autorização e auditoria.

Neste segundo trabalho prático, com o tema Modelação de Controlo de Acesso, pretende-se desenvolver um sistema de controlo de acesso baseado no método Bell-LaPadula, inserido num contexto universitário. Para tal, será feito um estudo sobre esse modelo, discutido se existe a possibilidade de um aluno fazer “batota” com um professor e apresentado um possível modo de implementação deste modelo.

2. Modelo Bell-LaPadula

O modelo Bell-LaPadula é um modelo de controlo de acesso semiformal ou formal, orientado para a política de confidencialidade, caracterizado por garantir a confidencialidade e ser estático, uma vez que os níveis de segurança (*labels*) nunca mudam [1]. Este modelo é baseado numa estrutura de segurança multinível (MLS- *Multi Level Security*), para que seja possível manter os dados em segredo e compartilhados apenas com quem tem autorização para recebê-los, concentrando-se assim mais na confidencialidade, não havendo nele uma distinção entre proteção e segurança [2].

Desta forma, o modelo é caracterizado por classificar os objetos e sujeitos em níveis de segurança (p.e. *Unclassified* (nível mais baixo), *Classified*, *Secret* e *Top Secret* (nível mais alto), evitando assim que a informação de um nível superior seja acedida por um sujeito de um nível inferior [3].

O modelo assenta em três regras fundamentais, duas de controlo de acesso obrigatório (MAC – *Mandatory Control Access*) e uma de controlo de acesso discricionário (DAC – *Discretionary Control Access*), tais são [4]:

- Um *subject* de um dado nível de segurança não pode ler dados de um *object* com nível de segurança mais alto (propriedade *don't read up*).
- Um *subject* de um dado nível de segurança não deve escrever para um *object* num nível inferior de segurança (propriedade *don't write down*).
- Utilizar uma matriz de acesso para especificar o controlo de acesso discricionário.

Por exemplo, considerando o contexto universitário, um estudante (*subject*) com um nível de segurança mais baixo, não deve poder aceder aos resultados de avaliações dos professores (*object*), que têm um nível de segurança mais alto (propriedade *don't read up*). Da mesma forma, um professor (*subject*) que tem acesso a informações sobre o conteúdo do exame final (*object*) não pode modificar ou escrever notas um

ficheiro de notas de um aluno, com nível de segurança mais baixo (propriedade *don't write down*).

Algumas desvantagens deste modelo centram-se no facto de ser muito orientado para a confidencialidade, não garantindo a integridade; possuir elevada complexidade na implementação do modelo na vida real e não fornecer nenhum método para a gestão das classificações: assume que todos os dados são atribuídos com uma classificação e que a classificação dos dados nunca irá mudar [5].

3. Conceção da *Lattice*

Para este trabalho prático foi-nos pedida a elaboração da *lattice* na qual são definidos os seguintes três 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). O modelo apresenta duas categorias que permitem identificar os serviços presentes num contexto universitário: os Serviços Académicos (AS - *Academic Services*), responsáveis pela gestão das aulas, das salas, ect, e Serviços Científicos (ScS - *Scientific Services*), que abordam tudo relacionado com a área de investigação.

Os níveis de segurança combinados com as categorias formam as *labels* que nesta situação são doze, em que cada uma tem dominância sob a inferior. Tal como é referido no enunciado, à entidade “Alunos” será atribuída a *label* (C, {AS}), por sua vez, à entidade “Docentes” será atribuída 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 do problema.

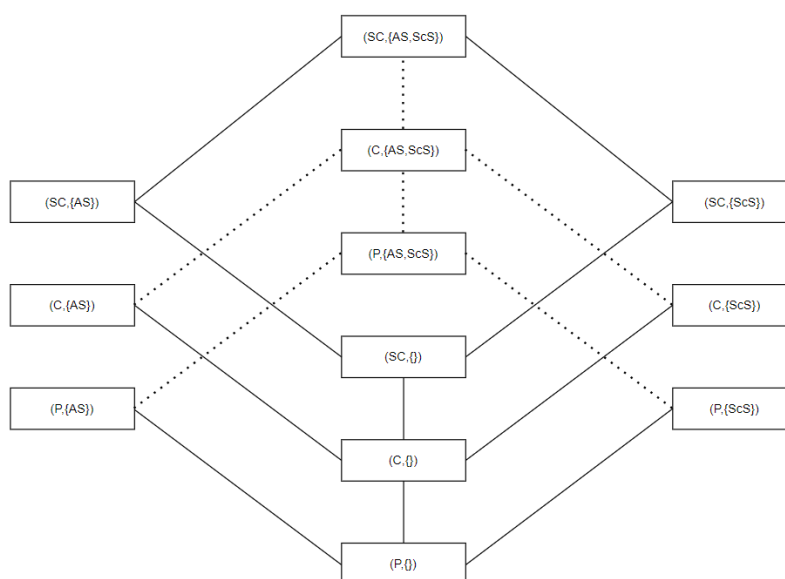


Figura 1. Conceção da *lattice*.

A dominância refere-se à relação de controlo de acesso que determina que *subjects* podem aceder a determinados *objects*. Em termos práticos, um sujeito A domina um sujeito B se o nível de segurança de A for maior ou igual ao nível de segurança B. Além disto, se um sujeito A domina B e um sujeito B domina C, então tendo em conta a propriedade transitiva, A domina C.

O conceito de hierarquia, refere-se à relação entre os níveis de segurança e é utilizada para aplicar as restrições de acesso. Desta forma, um *subject* com maior nível de segurança pode aceder a informação de um nível inferior, mas o contrário não pode ocorrer.

Neste contexto e segundo a Figura 1, SC (nível mais secreto) tem dominância sobre o nível C e o nível C sobre o nível P (nível menos secreto), logo segundo a propriedade transitiva, SC domina P. Da mesma forma, SC é hierarquicamente superior a C e C é hierarquicamente superior a P ($P < C < SC$), logo SC pode aceder às informações de C e P, mas C e P não podem aceder a informações de SC (nível superior), e assim sucessivamente.

De acordo com a figura anterior, foram atribuídos *subjects* a cada nível de acesso, apresentados nas Tabela 1, Tabela 2 e Tabela 3, juntamente com a explicação do raciocínio.

Tabela 1. Entidades do nível *Strictly Confidential*.

<i>Label</i>	<i>Subject</i>
(SC, {AS, ScS})	Reitoria
(SC, {AS})	Serviços de gestão académica
(SC, {ScS})	Coordenadores de Investigação
(SC, {})	Equipa de Presidência das escolas

- A reitoria, (SC, {AS, ScS}), é o órgão com mais poder dentro da universidade, sendo responsável por garantir a gestão dos serviços académicos, a atribuição de bolsas e projetos para investigação, entre outras informações confidenciais, sendo capaz de

aceder a todos os ficheiros. Deste modo, deve ser o *subject* com maior hierarquia, dominando todas as outras *labels*.

- Os serviços de gestão académica, (SC, {AS}), são responsáveis por garantir a confidencialidade de dados de alunos, professores, infraestruturas, horários, projetos e avaliações. Desta forma, deve obter um nível mais alto pois deve ter permissões para validar pautas, consultar dados para validação de estatutos e atribuição de certificados, validação de inscrições entre outras funcionalidades.

- Os coordenadores de investigação, (SC, {ScS}), são responsáveis pela gestão de toda a logística associada às bolsas de investigação, controlam e supervisionam os trabalhos dos mentorados, validam pedidos de bolsas, acesso a material, acesso a laboratórios, etc. Um investigador pode necessitar de um determinado material escrevendo o pedido para os Coordenadores, estes por sua vez leem o pedido e geram essa informação.

- A equipa de presidência das escolas, é responsável pela gestão administrativa e orçamentária, manutenção e desenvolvimento dos laboratórios, contratação de professores, gerir a acreditação e bom funcionamento dos cursos. Por exemplo, são responsáveis por analisar, aceitar ou negar um pedido dos diretores de departamento.

Tabela 2. Entidades do nível *Confidential*.

<i>Label</i>	<i>Subject</i>
(C, {AS,ScS})	Docentes
(C, {AS})	Alunos
(C, {ScS})	Investigadores
(C, {})	Diretores de departamento

- As duas primeiras *labels* foram definidas consoante o enunciado.
- Os investigadores, (C, {ScS}), são responsáveis por investigar e redigir artigos científicos sobre o processo das suas investigações, para tal podem necessitar de

recursos como material de investigação, acesso a laboratórios, bolsas de investigação, etc, os quais podem solicitar aos seus coordenadores.

- Os diretores de departamento são responsáveis por solicitar a contratação de novos professores, recursos e verbas, portanto devem ser capazes de realizar pedidos à equipa de presidência.

Tabela 3. Entidades do nível *Public*.

<i>Label</i>	<i>Subject</i>
(P, {AS})	Funcionários de limpeza dos serviços académicos
(P, {ScS})	Leitores de artigos científicos

- Os funcionários de limpeza, (P, {AS}), precisam de aceder a dados públicos da universidade (como por exemplo, a localização e horário de funcionamento) e precisam de registar (escrever) as horas e pessoa que realizou o serviço.
- Os leitores de artigos científicos, (P, {ScS}), conseguem aceder a recursos públicos como artigos publicados, dissertações efetuadas, etc, e conseguem ler os mesmos.

Utilizando a *lattice* desenvolvida, podemos construir uma tabela de permissões, sendo a classificação apenas ler (RO – *Read Only*), apenas escrever (WO – *Write Only*).

Tabela 4. Tabela de permissões.

	P {}	P{AS}	P{ScS}	P{AS,ScS}	C {}	C{AS}	C{ScS}	C{AS,ScS}	SC {}	SC{AS}	SC{ScS}	SC{AS,ScS}
P {}	RO/WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO	WO
P{AS}	RO	RO/WO		WO	WO	WO		WO	WO	WO		WO
P{ScS}	RO		RO/WO	WO	WO		WO	WO	WO		WO	WO
P{AS,ScS}	RO	RO	RO	RO/WO	WO	WO	WO	WO	WO	WO	WO	WO
C {}	RO				RO/WO	WO	WO	WO	WO	WO	WO	WO
C{AS}	RO	RO		RO	RO	RO/WO		WO	WO	WO		WO
C{ScS}	RO		RO	RO	RO		RO/WO	WO	WO		WO	WO
C{AS,ScS}	RO	RO	RO	RO	RO	RO	RO	RO/WO	WO	WO	WO	WO
SC {}	RO				RO				RO/WO	WO	WO	WO
SC{AS}	RO	RO		RO	RO	RO		RO	RO	RO/WO	WO	WO
SC{ScS}	RO		RO		RO		RO	RO	RO		RO/WO	WO
SC{AS,ScS}	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO/WO

4. Possibilidade da ocorrência de fraude

Tendo em conta o modelo desenvolvido e a tabela de permissões, verificámos que os docentes, como se encontram num nível hierarquicamente superior (C, {AS, ScS}), podem ler e escrever para todas as entidades que se encontram no nível *Confidential*. No caso dos alunos, (C, {AS}), esta encontra-se num nível hierarquicamente inferior, sendo que apenas pode ler os níveis inferiores e escrever nos níveis superiores.

Desta forma, o professor tem dominância sob os alunos, mas os dois encontram-se no mesmo nível de segurança e ambos têm acesso à mesma categoria “AS”. Tendo isto, surge uma vulnerabilidade que possibilita ao aluno fazer “batota”, assumindo a identidade do professor e enviando ficheiros para um nível de segurança superior, sem violar a propriedade *don't write down*, contando apenas que estes sejam referentes à categoria de serviços académicos.

O professor escreve a pauta dos alunos e encaminha-a para um nível hierárquico superior, os serviços académicos (SC, {AS}), que por sua vez validam a pauta e a disponibilizam para consulta dos alunos. Além disso, a equipa de presidência, (SC, {}), tem a possibilidade de aceder às pautas dos alunos para realizar serviços de acreditação ou avaliação. Desta forma, os alunos sendo hierarquicamente inferiores à equipa de presidência, podem escrever na pauta apesar de não terem acesso à leitura do seu conteúdo, ou até mesmo apagar o conteúdo da mesma, sendo este ataque conhecido como *blindwrite*.

5. Processo Automático de Implementação

Nesta secção, tal como pedido no enunciado do trabalho prático, iremos elaborar um processo automático de implementação do modelo Bell-LaPadula, desenvolvido no contexto universitário. Para implementar este sistema usámos uma máquina virtual com o sistema operativo Ubuntu e SELinux, uma arquitetura de segurança.

Tendo por base a *lattice* desenvolvida, decidimos definir um cenário no qual existem três níveis de acesso (SC, C e P) e três utilizadores: reitor (pertence ao nível SC), professor (pertence ao nível C) e funcionário de limpeza (pertence ao nível P).

Tendo isso, procedemos à implementação do modelo. Primeiro, criamos os três níveis de acesso (grupos), usando o comando: **sudo groupadd [nível]**.

```
camila@camila-VirtualBox:~$ sudo groupadd SC
camila@camila-VirtualBox:~$ sudo groupadd C
camila@camila-VirtualBox:~$ sudo groupadd P
```

Figura 2. Criação dos níveis de acesso.

Definimos os utilizadores do sistema, através do comando **sudo adduser -home /home/ [nome] [nome]**, sendo que neste caso os nomes são reitor (SC), professor(C) e funcionário(P). O comando vai criar o utilizador na diretoria *home* e adicionalmente foi definida uma palavra-passe.

Associamos a cada grupo os respetivos utilizadores:

```
camila@camila-VirtualBox:~$ sudo usermod -g SC reitor
camila@camila-VirtualBox:~$ id reitor
uid=1001(reitor) gid=1005(SC) groups=1005(SC)
camila@camila-VirtualBox:~$ groups reitor
reitor : SC
camila@camila-VirtualBox:~$ sudo usermod -g C professor
camila@camila-VirtualBox:~$ sudo usermod -g P funcionario
```

Figura 3. Associação dos utilizadores aos grupos de acesso.

Ao adicionar um novo utilizador, por padrão, um grupo com o mesmo nome do utilizador é criado automaticamente e definido como principal, para tornar os níveis o grupo principal utilizamos o argumento “-g”.

Recorrendo ao comando *setfacl*, definimos as ACL (*Access Control List*), para cada utilizador, com o intuito de estabelecer as permissões e com o auxílio do comando *getfacl* verificámos a lista das ACLs associadas. Além das permissões de leitura e/ou escrita, é necessário incluir: a permissão de execução ‘x’, o argumento ‘d’ para que novos ficheiros criados dentro do diretório herdem as permissões atribuídas a este, o argumento ‘m’ utilizado para adicionar ou modificar as ACLs existentes; e, por fim, o -R usado para aplicar as modificações ou configurações recursivamente a todos os ficheiros e subdiretórios. As Figura 4, Figura 5, Figura 6 mostram a aplicação destes comandos para reitor, professor e funcionário, respetivamente.

```
camila@camila-VirtualBox:/home$ sudo setfacl -d -R -m g:SC:rxw reitor/
[sudo] password for camila:
camila@camila-VirtualBox:/home$ sudo setfacl -d -R -m g:C:wx reitor/
camila@camila-VirtualBox:/home$ sudo setfacl -d -R -m g:P:wx reitor/
camila@camila-VirtualBox:/home$ getfacl reitor/
# file: reitor/
# owner: reitor
# group: SC
user::rxw
group::r-x
group:SC:rxw
group:C:-wx
group:P:-wx
mask::rxw
other::---
default:user::rxw
default:group::r-x
default:group:SC:rxw
default:group:C:-wx
default:group:P:-wx
default:mask::rxw
default:other::---
```

Figura 4. Estabelecimento das permissões e verificação para o reitor.

```
camila@camila-VirtualBox:/home$ sudo setfacl -d -R -m g:SC:rx professor/
camila@camila-VirtualBox:/home$ sudo setfacl -d -R -m g:C:rxw professor/
camila@camila-VirtualBox:/home$ sudo setfacl -d -R -m g:P:wx professor/
camila@camila-VirtualBox:/home$ getfacl professor/
# file: professor/
# owner: professor
# group: C
user::rxw
group::r-x
group:SC:r-x
group:C:rxw
group:P:-wx
mask::rxw
other::---
default:user::rxw
default:group::r-x
default:group:SC:r-x
default:group:C:rxw
default:group:P:-wx
default:mask::rxw
default:other::---
```

Figura 5. Estabelecimento das permissões e verificação para o professor.


```
camila@camila-VirtualBox:/home$ sudo setfacl -d -R -m g:SC:rx funcionario
camila@camila-VirtualBox:/home$ sudo setfacl -d -R -m g:C:rx funcionario
camila@camila-VirtualBox:/home$ sudo setfacl -d -R -m g:P:rx funcionario
camila@camila-VirtualBox:/home$ getfacl professor/
# file: professor/
# owner: professor
# group: C
user::rwx
group::r-x
group:SC:r-x
group:C:rwx
group:P:-wx
mask::rwx
other::---
default:user::rwx
default:group::r-x
default:group:SC:r-x
default:group:C:rwx
default:group:P:-wx
default:mask::rwx
default:other::---
```

Figura 6. Estabelecimento das permissões e verificação para o funcionário.

Tendo isto, vamos verificar se as permissões estabelecidas estão a funcionar de forma correta. Usando o exemplo do funcionário de limpeza, e de acordo com a *lattice* definida, este não deve ter permissão para ler o conteúdo do diretório do professor. Observando a Figura 7 verificamos que isto está funcional.

```
camila@camila-VirtualBox:/home$ su funcionario
Password:
funcionario@camila-VirtualBox:/home$ ls
camila funcionario professor reitor teste
funcionario@camila-VirtualBox:/home$ cd professor
funcionario@camila-VirtualBox:/home/professor$ ls
ls: cannot open directory '.': Permission denied
funcionario@camila-VirtualBox:/home/professor$
```

Figura 7. Exemplo de teste: funcionário não consegue aceder ao conteúdo do professor.

Adicionalmente, podemos testar se o professor tem acesso ao conteúdo do funcionário, sendo que isso é permitido. Observando a Figura 8 verificamos que isto também está funcional.

```
camila@camila-VirtualBox:/home$ su professor
Password:
professor@camila-VirtualBox:/home$ ls
camila funcionario professor reitor teste
professor@camila-VirtualBox:/home$ cd funcionario
professor@camila-VirtualBox:/home/funcionario$ ls
professor@camila-VirtualBox:/home/funcionario$
```

Figura 8. Exemplo de teste: professor consegue aceder ao conteúdo do funcionário.

Com o intuito de simular um ataque, como descrito na secção “Possibilidade de ocorrência de fraude”, fizemos *login* no utilizador professor, e criamos um ficheiro denominado “notasciber.txt”, como demonstrado na Figura 9.

```
camila@camila-VirtualBox:/home$ su professor
Password:
professor@camila-VirtualBox:/home$ cd
professor@camila-VirtualBox:~$ touch notasciber.txt
professor@camila-VirtualBox:~$ nano notasciber.txt
```

Figura 9. Criação do ficheiro *notasciber.txt*.

Acedendo ao conteúdo, temos:

```
professor@camila-VirtualBox:~$ cat notasciber.txt
\PAUTA\GRUPO 3
Barbara Fonseca- 18
Bruno Santos- 18
Camila Pinto- 18
Eduarda Dinis- 18
Gonçalo Dias- 18
```

Figura 10. Conteúdo do ficheiro *notasciber.txt*.

Recorrendo ao comando *getfacl* e tendo em conta que as permissões do ficheiro foram geradas automaticamente, temos:

```
professor@camila-VirtualBox:~$ getfacl notasciber.txt
# file: notasciber.txt
# owner: professor
# group: C
user::rw-
group::r-x
group:SC:r-x
group:C:rw-
group:P:-wx
mask::rw-
other::--
#effective:r--
#effective:r--
#effective:rw-
#effective:-w-
```

Figura 11. Permissões associadas ao ficheiro.

Ao observar a imagem verificámos que as permissões foram herdadas e são idênticas às permissões anteriormente definidas para o diretório.

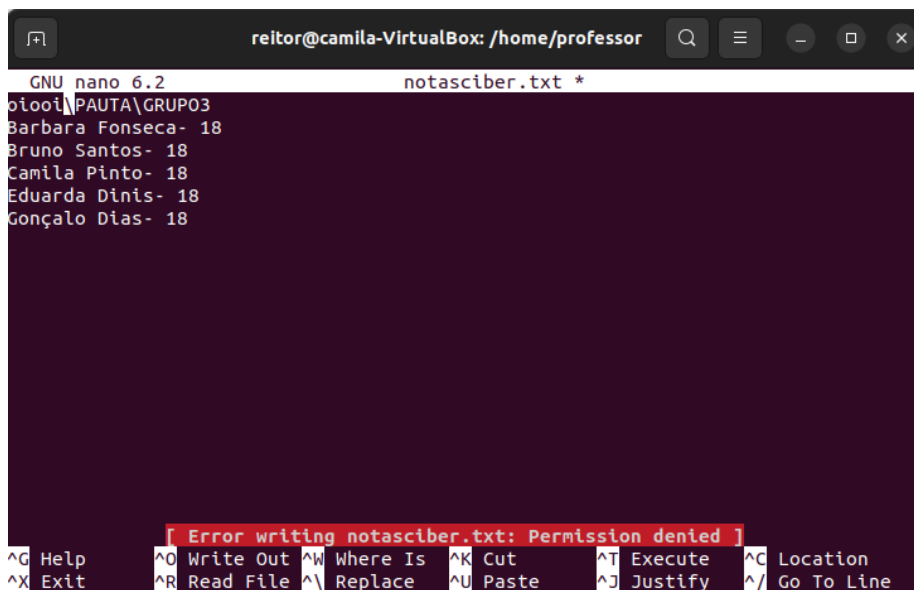
Para realizar uma verificação adicional, vamos testar se o reitor tem acesso a este ficheiro, e se apenas pode ler o mesmo.

Com a Figura 12 , conferimos que o reitor tem permissões de leitura do ficheiro, vamos agora analisar se tem permissões de escrita.

```
camila@camila-VirtualBox:~$ su reitor
Password:
reitor@camila-VirtualBox:/home/camila$ cd
reitor@camila-VirtualBox:~$ cd /home
reitor@camila-VirtualBox:/home$ ls
camila funcionario professor reitor teste
reitor@camila-VirtualBox:/home$ cd professor
reitor@camila-VirtualBox:/home/professor$ ls
notasciber.txt
reitor@camila-VirtualBox:/home/professor$ cat notasciber.txt
\PAUTA\GRUPO3
Barbara Fonseca- 18
Bruno Santos- 18
Camila Pinto- 18
Eduarda Dinis- 18
Gonçalo Dias- 18
reitor@camila-VirtualBox:/home/professor$
```

Figura 12. Exemplo de teste: reitor tem acesso à leitura do ficheiro.

Com a Figura 13, certificámos que o reitor não tem permissões de escrita, e que está tudo bem definido e implementado.



```
reitor@camila-VirtualBox: /home/professor
GNU nano 6.2 notasciber.txt *
oiooi\PAUTA\GRUPO3
Barbara Fonseca- 18
Bruno Santos- 18
Camila Pinto- 18
Eduarda Dinis- 18
Gonçalo Dias- 18
[ Error writing notasciber.txt: Permission denied ]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line
```

Figura 13. Exemplo de teste: reitor não tem acesso de escrita no ficheiro.

Focando na parte de simular um ataque, vamos fazer *login* no funcionário, sabendo que este apesar de não poder ler o ficheiro, consegue escrever no mesmo. Neste exemplo, apaga o conteúdo e substitui o mesmo por uma mensagem.

```
funcionario@camila-VirtualBox:~/home$ echo conseguientrar > /home/professor/notasciber.txt  
funcionario@camila-VirtualBox:~/home$
```

Figura 14. Tentativa de ataque *blindwrite*.

Com a Figura 15 confirmámos que o ataque ocorreu com sucesso:

```
professor@camila-VirtualBox:~$ ls  
notasciber.txt  
professor@camila-VirtualBox:~$ cat notasciber.txt  
consequientrar  
professor@camila-VirtualBox:~$
```

Figura 15. Sucesso na tentativa de ataque.

6. Conclusão

Fazendo uma retrospectiva e análise do trabalho realizado, notou-se que durante a implementação do modelo Bell-LaPadula, deparámo-nos com desafios relacionados à definição das permissões e à elaboração da *lattice* de segurança. Graças à orientação e colaboração com o professor, conseguimos superar essas dificuldades.

A realização do trabalho prático, foi executada em colaboração com os membros do grupo, cada um contribuindo com a sua opinião crítica durante as reuniões realizadas, o que nos permitiu alcançar um consenso.

Apesar da eficácia do modelo Bell-LaPadula na garantia da confidencialidade dos dados, é essencial reconhecer que ele não aborda diretamente a questão da integridade das informações. Como observado ao longo do trabalho, existe uma relação intrínseca entre confidencialidade e integridade: ao priorizarmos uma, a outra pode ser comprometida. Esta dualidade ressalta a complexidade do controlo de acesso em ambientes onde a sensibilidade dos dados requer um equilíbrio delicado entre os dois princípios.

A dualidade surge porque, ao aumentar as medidas de segurança para garantir a confidencialidade, como restrições mais rigorosas de acesso, pode-se inadvertidamente limitar a capacidade de modificar ou atualizar os dados de forma legítima, comprometendo a integridade. Da mesma forma, medidas para garantir a integridade dos dados, como bloqueios rígidos de modificação, podem restringir o acesso aos dados, comprometendo a confidencialidade.

O modelo Biba, ao contrário do Bell-LaPadula, adota uma abordagem oposta ao controlo de acesso. Enquanto o Bell-LaPadula se concentra na confidencialidade dos dados, o modelo Biba prioriza a integridade das informações. Isso significa que o Biba procura garantir que os dados não sejam modificados de forma não autorizada, garantindo sua precisão e consistência. Ao incorporar os princípios do modelo Biba juntamente com o Bell-LaPadula, podemos criar um sistema de controlo de acesso mais abrangente, capaz de equilibrar efetivamente a confidencialidade e a integridade

das informações em ambientes universitários e além. Essa abordagem combinada oferece uma solução mais completa para as complexidades do controlo de acesso em sistemas de informação, reconhecendo a importância de garantir não apenas a confidencialidade, mas também a integridade dos dados.

Em suma, afirmamos que este trabalho prático provou ser vantajoso para nós, para aumentar o nosso conhecimento nesta área e praticar a implementação de um modelo de controlo de acesso.

Referências

- [1] H. Santos, “Access Control and Authentication”, Segurança em Redes de Computadores, 2024.
- [2] "Landwehr, Carl (setembro de 1981). «Formal Models for Computer Security».
- [3] M. Toapanta, J. Nazareno, R. Tingo, F. Mendoza, A. Orizaga, and E. Mafla, “Analysis of the appropriate security models to apply in a distributed architecture,” IOP Conf. Ser. Mater. Sci. Eng., vol. 423, p. 012165, 2018.
- [4] Wikipedia contributors, “Modelo Bell–LaPadula,” Wikipedia, The Free Encyclopedia. [Online]. Available: https://pt.wikipedia.org/w/index.php?title=Modelo_Bell%E2%80%93LaPadula&oldid=60075511.
- [5] Icet.ac.in. [Online]. Available: <https://www.icet.ac.in/Uploads/Downloads/MOD2.pdf>. [Accessed: 22-Feb-2024].