

Hendrick Felipe Scheifer

João Victor Briganti

Luiz Gustavo Takeda

Segurança em Sistemas Operacionais GNU/Linux

Relatório técnico de atividade prática solicitado pelo professor Rodrigo Campiolo na disciplina de Sistemas Operacionais do Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Universidade Tecnológica Federal do Paraná – UTFPR

Departamento Acadêmico de Computação – DACOM

Bacharelado em Ciência da Computação – BCC

Campo Mourão

Dezembro / 2024

Resumo

Este trabalho teve como objetivo compreender e aplicar conceitos e recursos de segurança em Sistemas Operacionais GNU/Linux. Para isso, foi utilizado um sistema operacional GNU/Linux em uma máquina virtual executada no hipervisor VirtualBox. As atividades incluíram a configuração de mecanismos fundamentais de segurança no sistema operacional, a análise de arquivos essenciais para o gerenciamento desses recursos e a proteção de dados sensíveis. O trabalho foi concluído com êxito, resultando em uma compreensão prática e teórica sólida sobre segurança básica em um sistema GNU/Linux. Os resultados destacam a relevância da aplicação de medidas de segurança para proteger informações e recursos do sistema operacional, especialmente dados sensíveis, reforçando a importância de uma abordagem estruturada e prática.

Palavras-chave: virtualização; gerenciamento de segurança; proteção de dados.

Sumário

1	Introdução	4
2	Objetivos	4
3	Fundamentação	4
4	Materiais	5
5	Procedimentos e Resultados	6
5.1	Configuração do nível de segurança das senhas	6
5.2	Criação de grupos e usuários	8
5.2.1	Configuração <i>GROUPHOMES</i>	8
5.2.2	Criação de grupos	9
5.2.3	Cadastro de usuários	10
5.2.4	Remoção dos usuários do grupo <i>alunos</i> e do grupo <i>sudo</i>	11
5.3	Configuração de permissões para arquivos	12
5.3.1	Criação de arquivo e alteração de permissões	13
5.3.2	Alteração de dono e grupo do arquivo	14
5.4	Gerenciamento de autenticação	15
5.4.1	Observação de autenticações	15
5.4.2	Obrigatoriedade de autenticação	16
5.4.3	Arquivo <i>/etc/shadow</i>	16
5.4.4	Arquivo <i>/etc/passwd</i>	18
5.5	Gerenciamento de registros	19
5.5.1	Arquivos do diretório <i>/var/log/</i>	19
5.5.2	Configuração do arquivo <i>/etc/logrotate.conf</i>	19
5.5.3	Identificação de serviços ativos	20
5.6	Outros mecanismos de segurança	21
5.6.1	SELinux	21
5.6.2	<i>Pluggable Authentication Modules</i>	21
6	Discussão dos Resultados	22
7	Conclusões	22
8	Referências	22

1 Introdução

A segurança é um aspecto indispensável em um sistema operacional, o qual diz respeito a proteção dos recursos do computador contra ameaças externas, como acesso não autorizado, destruição ou alteração maliciosa e introdução acidental de inconsistências ([SILBERSCHATZ, 2015](#)).

O estudo e aplicação da segurança em sistemas operacionais resulta em melhor consistência no funcionamento, assim como garante a devida privacidade ao usuário. Este trabalho tem como foco aspectos de segurança em sistemas operacionais GNU/Linux, as próximas seções falarão sobre os objetivos da atividade, fundamentação essencial para a compreensão, procedimentos realizados e resultados obtidos.

2 Objetivos

Os objetivos deste trabalho são a compreensão de princípios básicos da segurança de Sistemas Operacionais, sendo eles a identificação e análise dos arquivos críticos para configuração de segurança, e comandos essenciais para implementação das configurações de segurança no sistema.

3 Fundamentação

A segurança da informação possui três princípios fundamentais ([MAZIERO, 2019](#)):

1. **Confidencialidade:** Princípio que estabelece que os recursos do sistema podem ser acessados exclusivamente por usuários devidamente autorizados, restringindo o acesso a informações sensíveis.
2. **Integridade:** Determina que os recursos do sistema podem apenas ser alterados ou destruídos por usuários autorizados.
3. **Disponibilidade:** Garante que os recursos computacionais estejam permanentemente acessíveis aos usuários autorizados, assegurando o funcionamento contínuo dos sistemas.

Toda potencial vulnerabilidade em um sistema computacional representa uma ameaça, enquanto as tentativas efetivas de explorar essas vulnerabilidades configuram ataques. Os ataques podem explorar as vulnerabilidades do sistema de maneira diferente, e, portanto podem ser classificados da seguinte maneira ([SILBERSCHATZ, 2015](#)):

- **Brecha de sigilo:** Acesso não autorizado a dados confidenciais, permitindo que invasores obtenham informações sigilosas para benefício próprio, violando a privacidade dos usuários.
- **Brecha de integridade:** Modificação não autorizada de dados, visando alterar informações para beneficiar o atacante ou prejudicar o proprietário, comprometendo a confiabilidade.
- **Brecha de disponibilidade:** Destruição não autorizada de dados que impede o acesso legítimo aos recursos, restringindo a utilização normal do sistema.
- **Roubo de serviço:** Utilização não autorizada de recursos computacionais, como instalação de um *daemon* para consumo de processamento ou armazenamento sem consentimento.
- **Recusa de serviço:** Impedimento do funcionamento adequado do sistema, prejudicando proprietários ou usuários, podendo ocorrer por falhas no sistema ou pela ação de invasores.

Conforme as vulnerabilidades que trazem ameaças ao sistema se tornam conhecidas, soluções foram desenvolvidas e aplicadas em sistemas operacionais, a seção 5 apresenta a discussão de alguns recursos básicos de segurança implementados para o sistema GNU/Linux e demonstra o uso destes recursos em uma máquina virtual.

4 Materiais

- Especificações do computador utilizado:
 - Modelo: Notebook Dell G15
 - CPU: Intel Core i5-12500H
 - Memória Principal: 16GB RAM
 - Memória Secundária: SSD 1TB NVME
 - Sistema Operacional: Windows 10
- Hipervisor: VirtualBox 7.1.2
- Sistema Operacional utilizado no Hipervisor: Debian *based*
- Núcleo: Linux 4.9.0

5 Procedimentos e Resultados

Este estudo apresenta a implementação de recursos básicos de segurança em um sistema operacional GNU/Linux, utilizando o ambiente de virtualização VirtualBox. O trabalho concentra-se na análise detalhada de arquivos e comandos críticos para a proteção de sistemas GNU/Linux, visando estabelecer práticas fundamentais de segurança da informação.

Após a instalação da distribuição fornecida pelo professor, o usuário “*root*” foi utilizado para a realização das configurações, pois possui privilégios administrativos (NEGUS, 2012).

5.1 Configuração do nível de segurança das senhas

Senhas são recursos de segurança comumente utilizados, devido a sua facilidade de compreensão e utilização. Porém, há a possibilidade de um invasor obter essa senha, por meio processo como engenharia social ou até mesmo métodos de força bruta e utilizado (SILBERSCHATZ, 2015).

Uma solução comumente utilizada para dificultar o ataque e reduzir os riscos de invasão é a utilização de padrões mais complexos de senha, que envolve a exigência de caracteres de diferentes classes para a criação da mesma, além de um número mínimo de caracteres. Para a implementação deste recurso em Sistemas Operacionais GNU/Linux, pode-se utilizar como ferramenta a biblioteca *libpam-quality*, que proporciona recursos de segurança, incluindo a verificação de senhas, esta biblioteca pode ser instalada com o comando `apt install libpam-quality`, após a instalação o arquivo de configuração da qualidade de senhas será criado em `/etc/security/pwquality.conf` (MRAZ, 2020).

O arquivo citado anteriormente contém as configurações da qualidade da senha, como a quantidade mínima de caracteres, de dígitos numéricos, caracteres especiais e de letras minúsculas e maiúsculas. O exemplo de configuração proposto pela atividade diz que a senha deve conter as seguintes exigências:

- Mínimo de 10 caracteres ao todo;
- Mínimo de 2 dígitos numéricos;
- Mínimo de 1 letra maiúscula;
- Mínimo de 1 letra minúscula;
- Mínimo de 1 caractere especial.

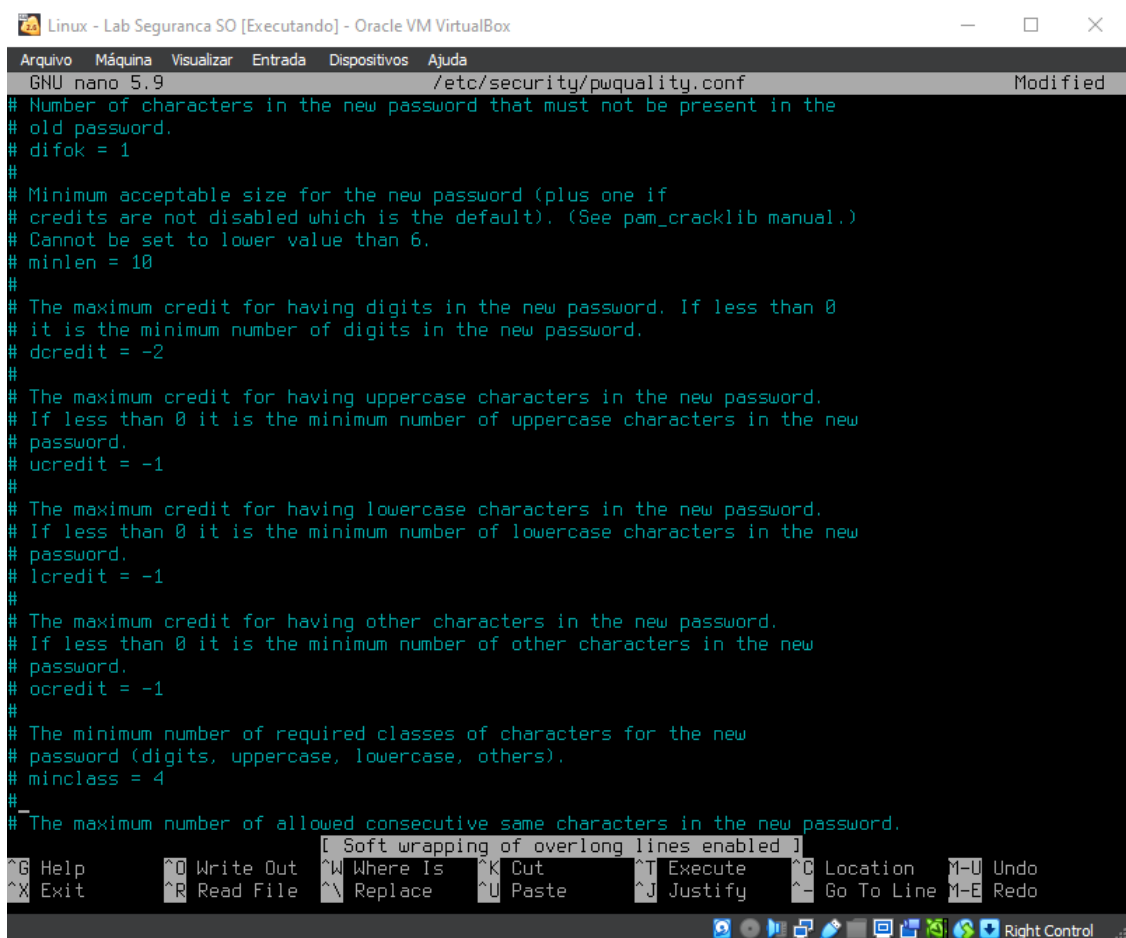
Para a adicionar os requisitos basta modificar o arquivo `/etc/security/pwquality.conf`, o editor de texto *nano*, a partir do comando `nano /etc/security/pwquality.conf`. O arquivo

possui algumas variáveis que controlam os requisitos de senhas, para a implementação proposta anteriormente foram alteradas as seguintes variáveis:

Para modificar os requisitos de senhas, utilizou-se o comando `nano /etc/security/pwquality.conf`, onde o *nano* é um editor de texto e o arquivo `/etc/security/pwquality.conf` permite ajustar os requisitos das senhas do sistema. Para atender os requisitos os seguintes parâmetros foram alterados:

- *minlen* = 10: para especificar que toda senha deve possuir no mínimo 10 caracteres ao todo;
- *dcredit* = -2: exige que senha tenha ao menos dois dígitos numéricos na senha;
- *ucredit* = -1: exige que a senha seja escrita com ao menos uma letra maiúscula;
- *lcredit* = -1: exige que a senha seja escrita com ao menos uma letra minúscula;
- *ocredit* = -1: exige que a senha seja escrito ao menos com um caractere especial;
- *minclass* = 4: determina que o número de classes diferentes que compõe a senha é 4 (dígitos, maiúsculas, minúsculas e caracteres especiais).

A Figura 1 mostra as alterações realizadas no arquivo `/etc/security/pwquality.conf`, citadas anteriormente.



```

Linux - Lab Seguranca SO [Executando] - Oracle VM VirtualBox
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda
GNU nano 5.9 /etc/security/pwquality.conf Modified
# Number of characters in the new password that must not be present in the
# old password.
# difok = 1
#
# Minimum acceptable size for the new password (plus one if
# credits are not disabled which is the default). (See pam_cracklib manual.)
# Cannot be set to lower value than 6.
# minlen = 10
#
# The maximum credit for having digits in the new password. If less than 0
# it is the minimum number of digits in the new password.
# dcredit = -2
#
# The maximum credit for having uppercase characters in the new password.
# If less than 0 it is the minimum number of uppercase characters in the new
# password.
# ucredit = -1
#
# The maximum credit for having lowercase characters in the new password.
# If less than 0 it is the minimum number of lowercase characters in the new
# password.
# lcredit = -1
#
# The maximum credit for having other characters in the new password.
# If less than 0 it is the minimum number of other characters in the new
# password.
# ocredit = -1
#
# The minimum number of required classes of characters for the new
# password (digits, uppercase, lowercase, others).
# minclass = 4
#
# The maximum number of allowed consecutive same characters in the new password.
[ Soft wrapping of overlong lines enabled ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo

```

Figura 1 – Arquivo `/etc/security/pwquality.conf`.

Após salvar as alterações, a criação de senhas só será bem sucedida caso a senha digitada possua os requisitos mínimos.

5.2 Criação de grupos e usuários

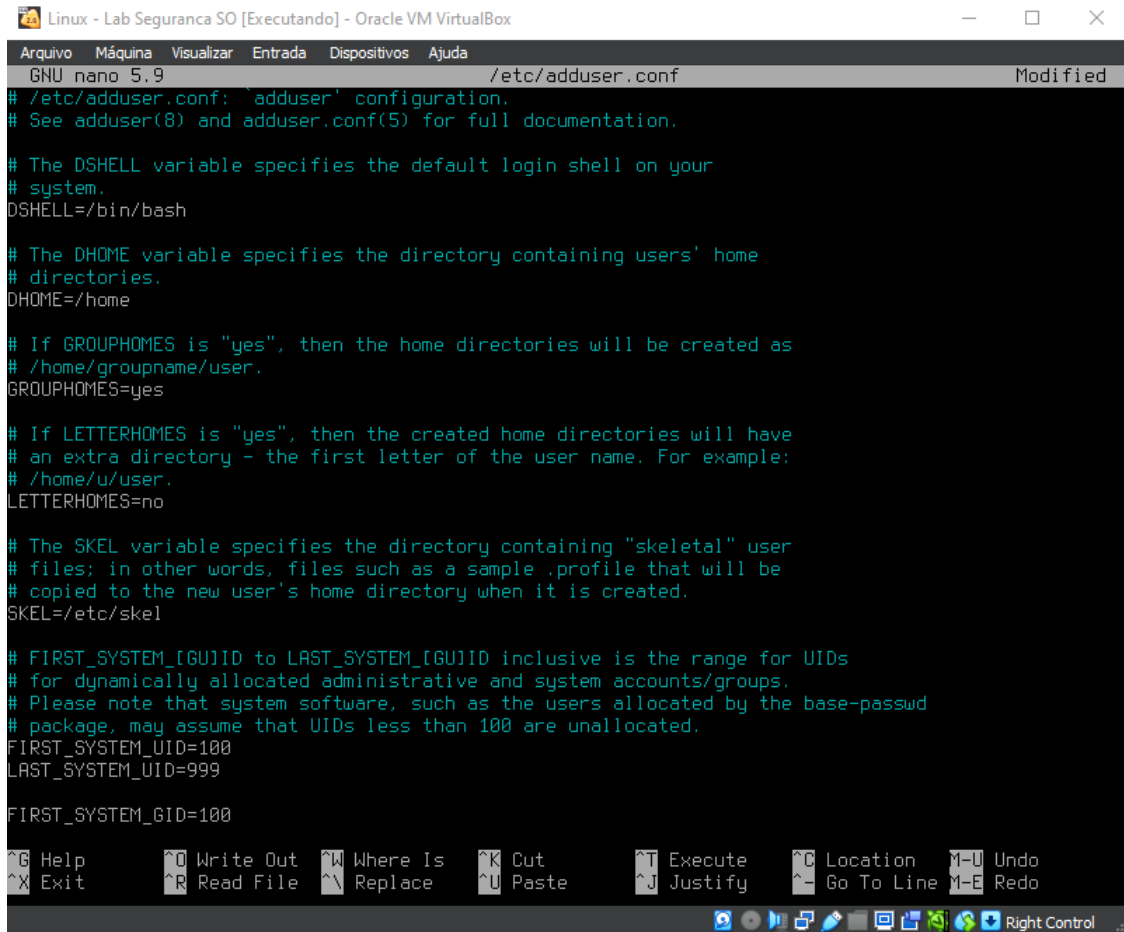
A próxima etapa envolve a criação de grupos e usuários para o sistema. Grupos são comumente criados em sistemas para que usuários com diferentes privilégios ou funções no sistema possam ser gerenciados de forma conjunta. A seguir serão descritos alguns passos essenciais para que esta criação aconteça de forma correta e que não comprometa a segurança do sistema.

5.2.1 Configuração *GROUPHOMES*

O parâmetro *GROUPHOMES* encontrado no arquivo de configuração `/etc/adduser.conf`, é utilizado para definir o comportamento padrão na criação de diretórios “home” de novos usuários, por meio do comando `adduser`. Quando este parâmetro está ativado, os diretórios pessoais dos usuários serão organizados em subdiretórios de um diretório geral do grupo. Por exemplo, o usuário “Aluno” do grupo “alunos” terá seu diretório pessoal em

`/home/alunos/Aluno/` ao invés de apenas `/home/Aluno/`. Esta configuração auxilia na segurança de modo geral, pois permite um maior controle e proteção dos dados armazenados em um sistema multiusuário (SILVA, 2020).

A figura 2 demonstra a alteração realizada no arquivo de configuração citado anteriormente, esta alteração foi realizada através do editor de texto *nano*. Para a ativação do parâmetro, basta alterar o valor associado à variável *GROUPHOME* para *yes*.



```
Linux - Lab Seguranca SO [Executando] - Oracle VM VirtualBox
GNU nano 5.9 /etc/adduser.conf Modified
# /etc/adduser.conf: 'adduser' configuration.
# See adduser(8) and adduser.conf(5) for full documentation.

# The DSHELL variable specifies the default login shell on your
# system.
DSHELL=/bin/bash

# The DHOME variable specifies the directory containing users' home
# directories.
DHOME=/home

# If GROUPHOMES is "yes", then the home directories will be created as
# /home/groupname/user.
GROUPHOMES=yes

# If LETTERHOMES is "yes", then the created home directories will have
# an extra directory - the first letter of the user name. For example:
# /home/u/user.
LETTERHOMES=no

# The SKEL variable specifies the directory containing "skeletal" user
# files; in other words, files such as a sample .profile that will be
# copied to the new user's home directory when it is created.
SKEL=/etc/skel

# FIRST_SYSTEM_UID to LAST_SYSTEM_UID inclusive is the range for UIDs
# for dynamically allocated administrative and system accounts/groups.
# Please note that system software, such as the users allocated by the base-passwd
# package, may assume that UIDs less than 100 are unallocated.
FIRST_SYSTEM_UID=100
LAST_SYSTEM_UID=999

FIRST_SYSTEM_GID=100

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo
Right Control
```

Figura 2 – Arquivo `/etc/adduser.conf`.

5.2.2 Criação de grupos

Como exemplo para a utilização do sistema, a prática propõe a criação de dois grupos: “alunos” e “professores”. Para criar os grupos, o comando *addgroup* foi utilizado, conforme apresentado na Figura 3. O retorno *Done* após a execução do comando indica que o grupo foi criado com sucesso (SILVA, 2020).

```

root@labsec:~# addgroup alunos
Adding group `alunos' (GID 1001) ...
Done.
root@labsec:~# addgroup professores
Adding group `professores' (GID 1002) ...
Done.
root@labsec:~# _

```

Figura 3 – Comando *addgroup*.

5.2.3 Cadastro de usuários

A atividade propõe o cadastro de um usuário para cada grupo, esta etapa pode ser dividida em dois passos: cadastrar os usuários e adicioná-los aos seus grupos. Para o cadastro do usuário foi utilizado o comando *adduser*, a Figura 4 demonstra o uso do comando com o cadastro do usuário “aluno” no sistema (SILVA, 2020).

```

Linux - Lab Seguranca SO [Executando] - Oracle VM VirtualBox
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda
root@labsec:~# adduser aluno
Adding user `aluno' ...
Adding new group `aluno' (1003) ...
Adding new user `aluno' (1001) with group `aluno' ...
Creating home directory `/home/aluno/aluno' ...
Copying files from `/etc/skel' ...
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
passwd: password updated successfully
Changing the user information for aluno
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
Adding new user `aluno' to extra groups ...
Adding user `aluno' to group `dialout' ...
Adding user `aluno' to group `dip' ...
adduser: The group `fuse' does not exist.
Adding user `aluno' to group `cdrom' ...
Adding user `aluno' to group `audio' ...
Adding user `aluno' to group `video' ...
Adding user `aluno' to group `plugdev' ...
Adding user `aluno' to group `users' ...
Adding user `aluno' to group `floppy' ...
Adding user `aluno' to group `netdev' ...
adduser: The group `scanner' does not exist.
Adding user `aluno' to group `lp' ...
adduser: The group `lpadmin' does not exist.
Adding user `aluno' to group `sudo' ...
adduser: The group `vboxsf' does not exist.
root@labsec:~# usermod -s /bin/bash aluno

```

Figura 4 – Comando *adduser aluno*.

De forma semelhante ao cadastro anterior, a figura 5 apresenta o cadastro do usuário “professor”.

```

Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda
Adding user `professor' ...
Adding new group `professor' (1004) ...
Adding new user `professor' (1002) with group `professor' ...
Creating home directory `/home/professor/professor' ...
Copying files from `/etc/skel' ...
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
passwd: password updated successfully
Changing the user information for professor
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]
Adding new user `professor' to extra groups ...
Adding user `professor' to group `dialout' ...
Adding user `professor' to group `dip' ...
adduser: The group `fuse' does not exist.
Adding user `professor' to group `cdrom' ...
Adding user `professor' to group `audio' ...
Adding user `professor' to group `video' ...
Adding user `professor' to group `plugdev' ...
Adding user `professor' to group `users' ...
Adding user `professor' to group `floppy' ...

Adding user `professor' to group `netdev' ...
adduser: The group `scanner' does not exist.
Adding user `professor' to group `lp' ...
adduser: The group `lpadmin' does not exist.
Adding user `professor' to group `sudo' ...
adduser: The group `vboxsf' does not exist.
root@labsec:~#

```

Figura 5 – Comando *adduser professor*.

Após o cadastro dos usuários, devemos associá-los a seus devidos grupos, para isso foi utilizado o comando para modificação de usuários *usermod* com a opção *-aG* que significa “*append group* (acrescentar grupo)” seguido do grupo que desejamos acrescentar e o usuário em questão (NEGUS, 2012). A Figura 6 demonstra o uso deste comando para adicionar os usuários “aluno” ao grupo “alunos” e “professor” ao grupo “professores”.

```

root@labsec:~# usermod -aG professores professor
root@labsec:~# usermod -aG alunos aluno
root@labsec:~#

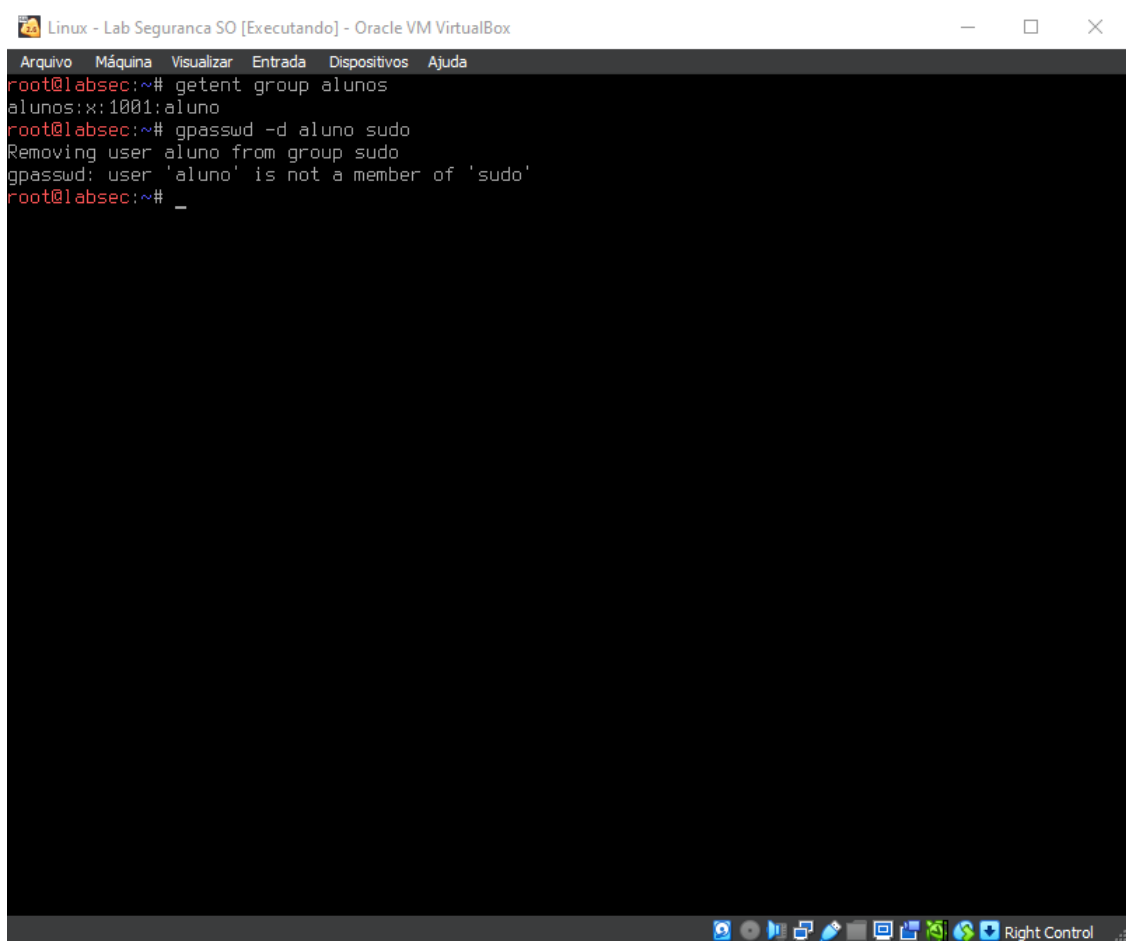
```

Figura 6 – Comando *usermod -aG*.

5.2.4 Remoção dos usuários do grupo *alunos* e do grupo *sudo*

É importante que apenas usuários devidamente autorizados tenham acesso a funções administrativas, as quais são proporcionadas para usuários pertencentes ao grupo “*sudo*”, portanto, por motivos de segurança, a atividade propõe que os usuários do grupo aluno sejam removidos deste grupo. Para isso, inicialmente identificamos os usuários per-

tencentos ao grupo alunos, utilizando o comando *getent group alunos*, que serve para consultar dados de um grupo específico, no caso, *alunos*. Após isso, removemos os usuários identificados no passo anterior com o comando *gpasswd*, utilizada para gerenciar usuários e senhas de grupos, com a opção “-d” para remover o usuário *aluno* do grupo *sudo*, resultando no seguinte comando *gpasswd -d aluno sudo*. Esta etapa é apresentada na Figura 7.



```
Linux - Lab Seguranca SO [Executando] - Oracle VM VirtualBox
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda
root@labsec:~# getent group alunos
alunos:x:1001:aluno
root@labsec:~# gpasswd -d aluno sudo
Removing user aluno from group sudo
gpasswd: user 'aluno' is not a member of 'sudo'
root@labsec:~# _
```

Figura 7 – Comando *gpasswd*.

5.3 Configuração de permissões para arquivos

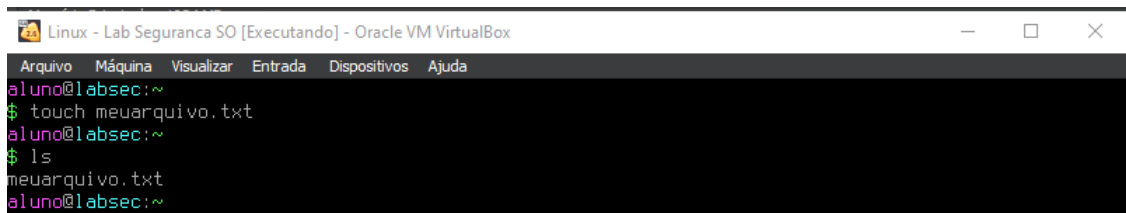
Permissões de arquivos servem para determinar o que cada usuário ou grupo pode, ou não, acessar no sistema, existem três permissões diferentes para arquivos no sistema operacional GNU/Linux:

- **Leitura:** Permissão para acessar o arquivo;
- **Escrita:** Permissão para alterar um arquivo;
- **Execução:** Permissão para executar um arquivo.

Essas três permissões são atribuídas a três níveis de autorização em relação ao arquivo: dono, grupo ao qual o dono pertence e outros, que se refere aos demais usuários. Nesta subseção será abordado a manipulação destas permissões.

5.3.1 Criação de arquivo e alteração de permissões

Para a manipulação de permissões, o arquivo “*meuarquivo.txt*” criado pelo usuário “*aluno*” será utilizado. O arquivo foi criado com a utilização do comando *touch*, conforme a Figura 8 apresenta.



```
Linux - Lab Seguranca SO [Executando] - Oracle VM VirtualBox
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda
aluno@labsec:~
$ touch meuarquivo.txt
aluno@labsec:~
$ ls
meuarquivo.txt
aluno@labsec:~
```

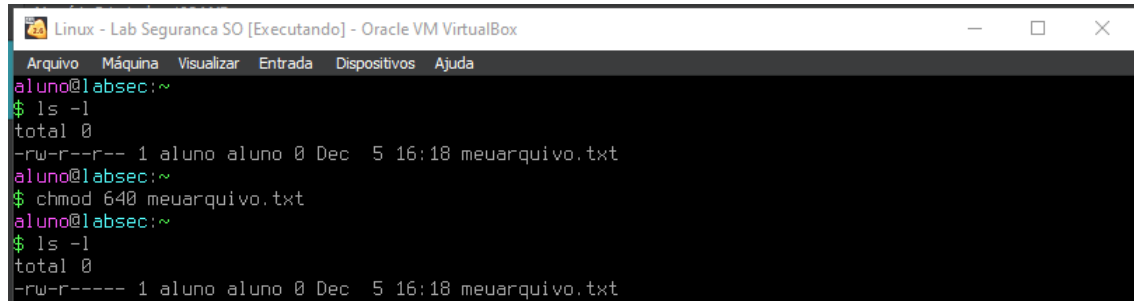
Figura 8 – Comando *touch*.

Para manipular permissões é utilizado o comando *chmod*, que deve ser acompanhado de dois dados: as permissões e o arquivo ou diretório em questão (NEGUS, 2012). As permissões podem ser escritas de duas formas:

- **Numérica:** as permissões dos três níveis são especificadas em três valores que vão de 0 a 7 e representam a soma das permissões desejadas, cada permissão possui um valor, execução vale 1, escrita vale 2, e leitura vale 4. Esta forma é apresentada na figura 9.
- **Com opções:** outra forma é especificar as permissões de cada nível que se deseja alterar de atribuindo os símbolos desejados, os níveis são *u* para o dono do arquivo, *g* para o grupo do arquivo e *o* para outros. Os símbolos equivalentes às permissões são *r* para leitura, *w* para escrita e *x* para execução. Esta forma é apresentada na figura 10.

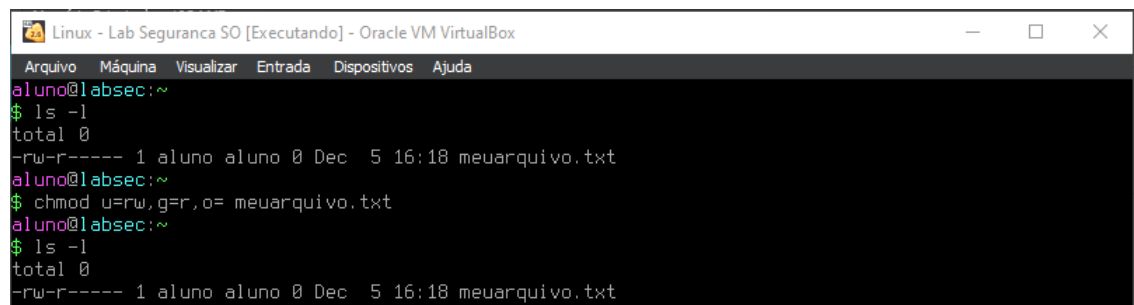
Para verificar as permissões dos arquivos foi utilizado o comando *ls* de listagem com a opção “-l”, que apresenta informações detalhadas dos arquivos e diretórios, incluindo suas permissões (NEGUS, 2012). A saída do comando apresenta as permissões da seguinte forma: do segundo ao décimo caractere da linha são agrupados em 3 trios que representam as permissões para os níveis de dono, grupo e outros, respectivamente, dentro do trio de caracteres são representadas as permissões de leitura, escrita e execução, respectivamente, caso haja um traço no lugar da permissão, esta permissão não foi concedida a este nível. Este comando pode ser observado na Figura 9 e na Figura 10.

A atividade propõe que as permissões do arquivo criado sejam: leitura e escrita para o dono, leitura para o grupo e nenhuma permissão para outros. Esta configuração foi realizada pelas duas formas de escritas que o comando *chmod* permite, a forma numérica pode ser observada na Figura 9 e a por opções na Figura 10.



```
Linux - Lab Seguranca SO [Executando] - Oracle VM VirtualBox
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda
aluno@labsec:~
$ ls -l
total 0
-rw-r--r-- 1 aluno aluno 0 Dec  5 16:18 meuarquivo.txt
aluno@labsec:~
$ chmod 640 meuarquivo.txt
aluno@labsec:~
$ ls -l
total 0
-rw-r----- 1 aluno aluno 0 Dec  5 16:18 meuarquivo.txt
```

Figura 9 – Comando *chmod* no formato numérico.



```
Linux - Lab Seguranca SO [Executando] - Oracle VM VirtualBox
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda
aluno@labsec:~
$ ls -l
total 0
-rw-r----- 1 aluno aluno 0 Dec  5 16:18 meuarquivo.txt
aluno@labsec:~
$ chmod u=rw,g=r,o= meuarquivo.txt
aluno@labsec:~
$ ls -l
total 0
-rw-r----- 1 aluno aluno 0 Dec  5 16:18 meuarquivo.txt
```

Figura 10 – Comando *chmod* com opções.

5.3.2 Alteração de dono e grupo do arquivo

Em muitas situações a posse de um arquivo pode ser alterada, e para manter a segurança do sistema, deve-se alterar o dono e, caso necessário, o grupo do arquivo. O comando *chown* pode ser usado para tal operação, que deve receber como dados o nome do novo usuário ou grupo e o nome do arquivo ou diretório, a especificação pelo dono pode ser feita simplesmente utilizando o nome do usuário, já no caso do grupo o “:” precisa ser utilizado antes do nome grupo (NEGUS, 2012). O uso deste comando pode é apresentado na Figura 11.

Para a verificação do dono e grupo do arquivo foi utilizado o comando *ls -l*. A atividade propõe que o arquivo tenha sua posse transferida para o usuário “professor” e o grupo “professores”. A execução desta etapa é apresentada na Figura 11.

```

root@labsec:~# ls -l /home/aluno/aluno/
total 0
-rw-r----- 1 aluno aluno 0 Dec  5 16:18 meuarquivo.txt
root@labsec:~# chown professor /home/aluno/aluno/meuarquivo.txt
root@labsec:~# ls -l /home/aluno/aluno/
total 0
-rw-r----- 1 professor aluno 0 Dec  5 16:18 meuarquivo.txt
root@labsec:~# chown :professores /home/aluno/aluno/meuarquivo.txt
root@labsec:~# ls -l /home/aluno/aluno/
total 0
-rw-r----- 1 professor professores 0 Dec  5 16:18 meuarquivo.txt
root@labsec:~# _

```

Figura 11 – Comando *chown*.

5.4 Gerenciamento de autenticação

A autenticação de usuários é a principal forma de acesso ao sistema, devendo ocorrer de forma devidamente autorizada, portanto há a necessidade de gerenciar a autenticação do sistema. Nesta subseção será abordado a observação de autenticações recentes, configuração da obrigatoriedade de autenticação e arquivos importantes referente à autenticação.

5.4.1 Observação de autenticações

Para observar os usuários que se autenticaram no sistema recentemente podemos utilizar o comando *lastlog*, que apresenta a lista de usuários do sistema e em caso de autenticação apresenta a data e a porta na qual aconteceu sua última autenticação, conforme pode ser visto na Figura 12 (NEGUS, 2012).

```

root@labsec:~# lastlog
Username      Port    From      Latest
root          tty1              Sat Dec  7 08:08:34 -0300 2024
daemon                **Never logged in**
bin                  **Never logged in**
sys                  **Never logged in**
sync                 **Never logged in**
games                **Never logged in**
man                  **Never logged in**
lp                   **Never logged in**
mail                 **Never logged in**
news                 **Never logged in**
uucp                 **Never logged in**
proxy                **Never logged in**
www-data             **Never logged in**
backup               **Never logged in**
list                 **Never logged in**
irc                  **Never logged in**
gnats                **Never logged in**
nobody              **Never logged in**
lapt                 **Never logged in**
student            tty1    Thu Jun 23 09:18:36 -0300 2022
professor          tty1    Sat Dec  7 08:08:27 -0300 2024
aluno              tty1    Sat Dec  7 08:05:47 -0300 2024
root@labsec:~# _

```

Figura 12 – Comando *lastlog*.

5.4.2 Obrigatoriedade de autenticação

É possível em Sistemas Operacionais permitir que usuário não seja obrigado a autenticar-se para realizar o *login*, porém isso não é recomendado por motivos de segurança.

Em sistemas GNU/Linux essa configuração pode ser realizada da seguinte forma: no arquivo */etc/passwd* encontre o usuário desejado, o segundo item da linha do usuário é um “x” que indica que a senha do usuário está criptografada e armazenada no arquivo */etc/passwd*, remova este “x” e o usuário não irá possuir uma senha vinculada, o que, na prática, retira a obrigatoriedade de autenticação (SILVA, 2020).

A Figura 13 mostra a remoção da obrigatoriedade de senha para o usuário *root*. A Figura 14 mostra o usuário *root* realizando *login* sem a necessidade de uma senha.

```
root::0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534:/:nonexistent:/usr/sbin/nologin
student:x:1000:1000:Aluno SO,,,:/home/student:/bin/bash
professor:x:1001:1003:,,,:/home/professor:/bin/bash
aluno:x:1002:1004:,,,:/home/aluno:/bin/bash
```

Figura 13 – Alteração no arquivo */etc/passwd*.

```
root@labsec:~# exit
logout
Welcome to antiX. Powered by Debian.
labsec login: root
Last login: Sat Dec 7 00:22:03 -03 2024 on tty1
root@labsec:~#
```

Figura 14 – Usuário *root* acessando o sistema sem senha.

5.4.3 Arquivo */etc/shadow*

O arquivo */etc/shadow* armazena dados importantes relacionados às senhas dos usuários do sistema, os quais podem ser alterados conforme a necessidade para razões de segurança, uma representação textual do arquivo pode ser observada na Figura 15 (SILVA, 2020).


```

root@labsec:~# cat /etc/shadow
root:$y$j9T$07vX6D08Is4n0idZgEwdS.$Jjq2y4S6NW9MDDpoi7nYlypq/ypF7NUavZnTGEEg2:19166:0:99999:7:::
daemon*:18930:0:99999:7:::
bin*:18930:0:99999:7:::
sys*:18930:0:99999:7:::
sync*:18930:0:99999:7:::
games*:18930:0:99999:7:::
man*:18930:0:99999:7:::
lp*:18930:0:99999:7:::
mail*:18930:0:99999:7:::
news*:18930:0:99999:7:::
uucp*:18930:0:99999:7:::
proxy*:18930:0:99999:7:::
www-data*:18930:0:99999:7:::
backup*:18930:0:99999:7:::
list*:18930:0:99999:7:::
irc*:18930:0:99999:7:::
gnats*:18930:0:99999:7:::
nobody*:18930:0:99999:7:::
Lapt*:18930:0:99999:7:::
student:$y$j9T$IJpQD7RSJUSKwboqXWEy0$VGtsnI08MusA.j0Q4Y8rQtRCrQUI2i2oFHGEwnA$svuc.:19166:0:99999:7:::
professor:$y$j9T$B0bx0C2nYY0wv.jq6QmMPx0$J3EddeZcRfRohp0.K.j.qo0IJGn/dEUDCscCxxYmYt.j18:20064:0:99999:7:::
aluno:$y$j9T$VnnxvhRh/GkK.5/V1aq7W.$2/Kp8bQC99QEJt7IrD1ckuvyha9Cg6eE4sBnmvX5mt/:20064:0:99999:7:::
root@labsec:~#

```

Figura 15 – Arquivo `/etc/shadow`.

Para apresentar os campos de dados presentes no arquivo, utilizaremos como exemplo o usuário *student*, os dados presentes no arquivo são:

- **Nome do usuário:** identifica o usuário em questão pelo seu nome de usuário no sistema;
- **Senha:** a senha é armazenada no segundo campo de forma criptografada, visando manter a senha segura mesmo que um invasor obtenha acesso ao arquivo `/etc/shadow`;
- **Última modificação:** data da última modificação realizada na senha deste usuário, a data é dada em um formato específico, sendo a quantidade de dias decorridos após 01/01/1970, data selecionada para servir de referência para sistemas UNIX;
- **Tempo mínimo para alteração:** quantidade mínima de dias que o usuário deve esperar para poder alterar sua senha;
- **Tempo máximo para alteração:** representa a quantidade máxima de dias que o usuário pode manter a mesma senha, antes de ser obrigado a alterá-la;
- **Tempo para aviso prévio:** o sistema operacional pode alertar o usuário sobre a obrigatoriedade da troca de senha, o sexto campo representa com quantos dias antes da data limite o sistema emitirá este aviso;
- **Tempo de inatividade para desativação:** campo não adicionado ao usuário *student*, representaria a quantidade de dias que o usuário pode permanecer inativo sem ter sua conta bloqueada;
- **Data de expiração da conta:** não adicionado ao usuário *student*, representaria a data de expiração da conta do usuário, a data é dada com base no *Unix Epoch*.

5.4.4 Arquivo */etc/passwd*

O arquivo */etc/passwd* armazena dados importantes relacionados aos usuários do sistema, estes dados auxiliam no gerenciamento do sistema operacional e podem ser alterados conforme a necessidade, uma representação textual do arquivo pode ser observada na Figura 16 (SILVA, 2020).

```
root@labsec:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
student:x:1000:1000:Aluno SO,,,:/home/student:/bin/bash
professor:x:1001:1003:,,,:/home/professor:/bin/bash
aluno:x:1002:1004:,,,:/home/aluno:/bin/bash
root@labsec:~#
```

Figura 16 – Arquivo */etc/passwd*.

Para apresentar os campos de dados presentes no arquivo, utilizaremos como exemplo o usuário *student*, os dados presentes no arquivo são:

- **Nome de usuário:** identifica o usuário em questão pelo seu nome de usuário no sistema;
- **Marcador de autenticação:** quando marcado com “x” significa que a senha do usuário está armazenada no arquivo */etc/shadow*, caso não haja esta marcação, não será necessário a autenticação para este usuário, a Subseção 5.4.2 apresenta o uso deste campo;
- **UID:** armazena o número de identificação do usuário;
- **GID:** armazena o número de identificação do grupo ao qual o usuário pertence;
- **Informações adicionais:** armazena informações adicionais do usuário, no caso do usuário *student* está armazenado o nome completo do usuário, “Aluno SO”;
- **Diretório pessoal:** armazena o caminho para o diretório pessoal do usuário, onde ficam seus arquivos no sistema;
- **shell padrão:** caminho para o executável do *shell* padrão do usuário.

5.5 Gerenciamento de registros

Um mecanismo presente em Sistemas Operacionais é a capacidade de realizar registros de atividades realizadas no sistema, chamamos estes registros de registros em *log*. Diversas ações podem ser armazenadas em *logs*, como, por exemplo, cada chamada de sistema realizada, cada usuário autenticado, ou outros comportamentos que podem se tornar suspeitos e serem submetidos a uma futura análise. Estes *logs* são extremamente relevantes, pois auxiliam na detecção de atividades suspeitas que possam indicar uma tentativa de invasão, como falhas de autenticação ou de autorização (SILBERSCHATZ, 2015).

5.5.1 Arquivos do diretório */var/log/*

O diretório */var/log/* armazena arquivos de *logs* para prover um melhor controle sobre a segurança do sistema operacional, este diretório armazena diferentes categorias, separadas em diferentes arquivos, como os arquivos (NEGUS, 2012):

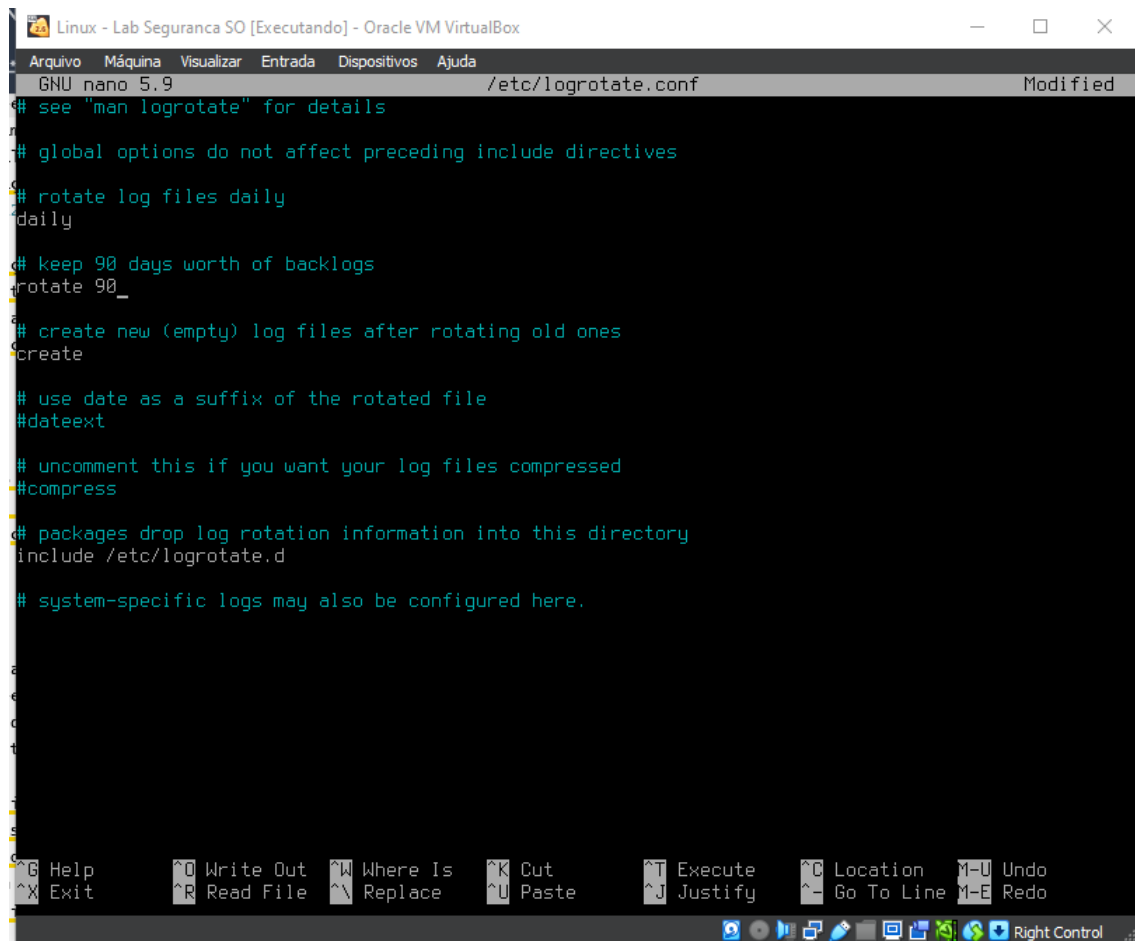
- ***syslog***: este arquivo armazena *logs* gerais do sistema, que registra diversos eventos relevantes de categorias que não possuem um arquivo específico para registro;
- ***kern.log***: este arquivo armazena mensagens originadas no *kernel* do sistema operacional, estas mensagens envolvem eventos como problemas relacionados ao hardware ou até mesmo de mau funcionamento do núcleo;
- ***auth.log***: este arquivo armazena *logs* relacionados à autenticação de usuários, como tentativas de *login* ou mudança de privilégios por meio do comando *sudo*;
- ***daemon.log***: este arquivo armazena mensagens provenientes de *daemons*, que são processos do sistema sendo executados em segundo plano, o que se torna relevante devido à execução discreta destes processos, que não interagem com o usuário.

5.5.2 Configuração do arquivo */etc/logrotate.conf*

Sistemas Operacionais GNU/Linux possuem uma ferramenta chamada *logrotate*, que automatiza algumas atividades referentes aos *logs*, como a rotação e gerenciamento de histórico de *logs*. Esta ferramenta pode ser configurada alterando o arquivo */etc/logrotate.conf* conforme a necessidade, a atividade propõe a duas configurações importantes, a rotação diária de *logs*, e o armazenamento de três meses de histórico. Na prática, isso implica na criação de um novo arquivo de *log* todo dia, e o armazenamento dos últimos 90 arquivos (3 meses) (SILVA, 2020).

Para aplicar a configuração proposta, o arquivo */etc/logrotate.conf* foi alterado, utilizando o editor de texto *nano*, e a rotação foi alterada para “*daily*” (diária) e o arma-

zenamento foi alterado para “*rotate 90*”, para armazenar 90 rotações. O arquivo alterado pode ser observado na Figura 17.



The screenshot shows a terminal window titled "Linux - Lab Seguranca SO [Executando] - Oracle VM VirtualBox". The window displays the GNU nano 5.9 editor editing the file /etc/logrotate.conf. The file content is as follows:

```
# see "man logrotate" for details
# global options do not affect preceding include directives
# rotate log files daily
daily
# keep 90 days worth of backlogs
rotate 90
# create new (empty) log files after rotating old ones
create
# use date as a suffix of the rotated file
#dateext
# uncomment this if you want your log files compressed
#compress
# packages drop log rotation information into this directory
include /etc/logrotate.d
# system-specific logs may also be configured here.
```

The bottom of the window shows a status bar with various keyboard shortcuts: ^G Help, ^X Exit, ^O Write Out, ^R Read File, ^W Where Is, ^N Replace, ^K Cut, ^U Paste, ^T Execute, ^J Justify, ^C Location, ^_ Go To Line, M-U Undo, M-E Redo, and a "Right Control" button.

Figura 17 – Arquivo */etc/logrotate.conf*.

5.5.3 Identificação de serviços ativos

Outra atividade importante no âmbito na segurança em Sistemas Operacionais é a observação dos serviços ativos no sistema, pois permite a identificação de serviços suspeitos, que poderiam acarretar falhas de segurança como roubo de serviço ou recusa de serviço, citadas na Seção 3.

A verificação dos serviços de um sistema operacional Linux pode ser realizada com o auxílio do comando *service*, acompanhado da opção “*-status-all*”, e para filtrar os serviços ativos podemos redirecionar a saída do comando anterior para o comando *grep*, responsável por filtrar as linhas que contenham um determinado texto, como todo serviço ativo possui o marcador “[+]” em sua linha, vamos utilizar este termo para filtrar, o resultado desta operação pode ser visualizado na Figura 18.

```
root@labsec:~# service --status-all | grep '[ + \]'
[ + ] cron
[ ? ] cryptdisks
[ ? ] cryptdisks-early
[ ? ] hwclock.sh
[ ? ] kmod
[ ? ] mount-configfs
[ ? ] networking
[ + ] rsyslog
[ + ] udev
[ ? ] umountnfs-alternative.sh
root@labsec:~# _
```

Figura 18 – Comando *service*.

Devido à forma que o *grep* interpreta caracteres especiais, os serviços com status indefinidos, demarcados com “[?]” também são retornados, mas podem ser ignorados.

5.6 Outros mecanismos de segurança

O problema da segurança em Sistemas Operacionais é amplamente discutido e a comunidade busca constantemente soluções nesta área, algumas soluções importantes que visam serão apresentadas a seguir.

5.6.1 SELinux

O SELinux é um mecanismo de controle de acesso com multipolíticas, que constitui uma infraestrutura de segurança para o núcleo Linux, permitindo a aplicação de políticas diversas aos recursos do sistema operacional. Uma desvantagem desta infraestrutura é sua alta complexidade, que torna sua compreensão e configuração difícil, devido a este fator, outros sistemas que visam o mesmo objetivo vem sendo desenvolvidos buscando uma complexidade menor para se tornarem mais simples de se aplicar ao núcleo Linux ([MAZIERO, 2019](#)).

5.6.2 *Pluggable Authentication Modules*

Outro recurso disponível no Linux é o PAM (*Pluggable Authentication Modules*), um mecanismo baseado em uma biblioteca compartilhada que pode ser utilizado por qualquer componente do sistema operacional que precise autenticar usuários para realizar alguma atividade. O funcionamento do PAM permite o carregamento de módulos de autenticação conforme especificado no arquivo de configurações. Essa modularização possibilita que novos módulos sejam adicionados à configuração e, em seguida, fiquem imediatamente disponíveis para qualquer componente do sistema ([SILBERSCHATZ, 2015](#)).

6 Discussão dos Resultados

Os resultados obtidos foram satisfatórios, alcançando o objetivo de compreender e aplicar conceitos e recursos de segurança em um sistema GNU/Linux com sucesso. O processo não proporcionou grandes dificuldades, apenas uma curva de aprendizado um pouco acima do esperado, que foi superada com o auxílio da bibliografia. Foi possível realizar com êxito configurações importantes relacionadas a segurança, como a implementação de requisitos para as senhas, atribuição de permissões a arquivos e remoção de usuários do grupo *sudo*. Além disso, foi possível identificar e conhecer arquivos importantes para a segurança do sistema operacional.

7 Conclusões

O objeto do trabalho de compreender e aplicar conceitos e recursos de segurança em Sistemas Operacionais foi atingido com sucesso e sua importância para a segurança da informação ficou clara ao longo do processo. Durante os procedimentos foram realizadas configuração e verificação de arquivos importantes para a segurança em sistemas operacionais Linux. O aprendizado obtido durante o trabalho é de fundamental importância para a formação de um cientista da computação, visto que Sistemas Operacionais são amplamente utilizados e podem se tornar alvos de invasores, comprometendo a segurança da informação. Além de todos os procedimentos realizados, ainda há diversos recursos de segurança não explorados durante a atividade, que podem prover meios eficazes de proteger dados e recursos.

8 Referências

- MAZIERO, C. A. *Sistemas Operacionais: Conceitos e Mecanismos*. [S.l.]: UFPR, 2019. Citado 2 vezes nas páginas 4 e 21.
- MRAZ, T. *Manpage, pwquality.conf*. 2020. <https://manpages.debian.org/testing/libpwquality-common/pwquality.conf.5.en.html>. [Acessado 13-12-2024]. Citado na página 6.
- NEGUS, C. *Linux bible*. [S.l.]: John Wiley & Sons, 2012. v. 772. Citado 6 vezes nas páginas 6, 11, 13, 14, 15 e 19.
- SILBERSCHATZ, A. *Fundamentos de Sistemas Operacionais*. [S.l.]: LTC, 2015. Citado 4 vezes nas páginas 4, 6, 19 e 21.
- SILVA, G. M. da. *Guia Foca Linux, Segurança*. 2020. <https://www.guiafoca.org/guiaonline/seguranca/>. [Acessado 13-12-2024]. Citado 5 vezes nas páginas 9, 10, 16, 18 e 19.