Criando Zona Segura

Para criar uma "zona segura" com arquivos somente leitura no Ubuntu, você tem algumas abordagens eficazes:

Método 1: Usando chmod recursivo

```
# Torna todos os arquivos somente leitura (remove permissões de escrita e execução)
chmod -R a-wx /caminho/para/pasta

# Ou define permissões específicas: leitura para todos, sem escrita/execução
chmod -R 444 /caminho/para/pasta
```

Método 2: Usando ACLs (Access Control Lists) - mais robusto

```
# Instala ACL se necessário
sudo apt install acl

# Define ACL padrão para a pasta (herança automática)
setfacl -R -m d:u::r--,d:g::r--,d:o::r-- /caminho/para/pasta
setfacl -R -m u::r--,g::r--,o::r-- /caminho/para/pasta

# Remove todas as permissões de execução e escrita
setfacl -R -m d:u::-wx,d:g::-wx,d:o::-wx /caminho/para/pasta
```

Método 3: Montagem somente leitura (mais seguro)

```
# Cria um ponto de montagem bind somente leitura
sudo mount --bind /pasta/original /pasta/segura
sudo mount -o remount,ro /pasta/segura
```

Método 4: Usando chattr (atributos imutáveis)

```
# Torna arquivos imutáveis (nem root pode modificar facilmente)
sudo chattr +i /caminho/para/pasta/*
sudo chattr +i /caminho/para/pasta

# Para reverter: chattr -i
```

Script para automação

```
#!/bin/bash
PASTA="/caminho/para/zona/segura"

# Remove permissões de escrita e execução
find "$PASTA" -type f -exec chmod 444 {} \;
find "$PASTA" -type d -exec chmod 555 {} \;

# Define ACL padrão para novos arquivos
setfacl -R -m d:u::r--,d:g::r--,d:o::r-- "$PASTA"
```

Recomendação: Combine o método 1 (chmod) com ACLs para garantir que novos arquivos herdem automaticamente as permissões restritivas. O método 3 (montagem readonly)

oferece a maior segurança, pois até mesmo o root tem dificuldade para modificar os arquivos.

Limitando a escrita a um usuário/app

Para limitar a escrita em uma zona segura apenas a uma aplicação específica usando ACLs, aqui estão as abordagens mais eficazes:

ACL por Usuário Dedicado (Mais Seguro)

1. Criar usuário dedicado para a aplicação

```
# Cria usuário sem shell e home
sudo useradd -r -s /bin/false -d /nonexistent app_writer

# Ou com shell se necessário para debugging
sudo useradd -m -s /bin/bash app_writer
```

2. Configurar ACLs restritivas

```
PASTA="/zona/segura"

# Remove todas as permissões padrão
sudo chmod 000 "$PASTA"

# Define ACLs específicas
sudo setfacl -m u:app_writer:rwx "$PASTA"        # App pode tudo
sudo setfacl -m u:root:rx "$PASTA"               # Root pode ler/navegar
sudo setfacl -m g::--- "$PASTA"                  # Grupo sem permissão
sudo setfacl -m o::--- "$PASTA"                  # Outros sem permissão

# ACLs padrão para herança em subpastas/arquivos
sudo setfacl -m d:u:app_writer:rwx "$PASTA"
sudo setfacl -m d:u:root:rx "$PASTA"
sudo setfacl -m d:g::--- "$PASTA"
sudo setfacl -m d:o::--- "$PASTA"

# Para arquivos existentes
find "$PASTA" -type f -exec sudo setfacl -m u:app_writer:rw {} \;
find "$PASTA" -type f -exec sudo setfacl -m u:root:r {} \;
find "$PASTA" -type f -exec sudo setfacl -m g::--- {} \;
find "$PASTA" -type f -exec sudo setfacl -m o::--- {} \;
```

3. Executar aplicação como usuário dedicado

```
# Método 1: sudo
sudo -u app_writer /caminho/para/aplicacao

# Método 2: systemd service
sudo tee /etc/systemd/system/minha-app.service << EOF
[Unit]
Description=Minha Aplicação
After=network.target

[Service]
Type=simple
User=app_writer
```

```
Group=app_writer
ExecStart=/caminho/para/aplicacao
Restart=always

[Install]
WantedBy=multi-user.target
EOF

sudo systemctl enable minha-app.service
sudo systemctl start minha-app.service
```

Script de Monitoramento e Validação

```bash
#!/bin/bash
# Script para verificar e corrigir permissões

PASTA="/zona/segura"
USER_PERMITIDO="app_writer"

# Função para configurar ACLs
configurar_acls() {
    echo "Configurando ACLs para $PASTA..."

    # Pasta principal
    sudo setfacl -m u:$USER_PERMITIDO:rwx "$PASTA"
    sudo setfacl -m u:root:rx "$PASTA"
    sudo setfacl -m g::--- "$PASTA"
    sudo setfacl -m o::--- "$PASTA"

    # ACLs padrão
    sudo setfacl -m d:u:$USER_PERMITIDO:rwx "$PASTA"
    sudo setfacl -m d:u:root:rx "$PASTA"
    sudo setfacl -m d:g::--- "$PASTA"
    sudo setfacl -m d:o::--- "$PASTA"

    # Aplicar recursivamente
    find "$PASTA" -type d -exec sudo setfacl -m u:$USER_PERMITIDO:rwx {} \;
    find "$PASTA" -type f -exec sudo setfacl -m u:$USER_PERMITIDO:rw {} \;
    find "$PASTA" -exec sudo setfacl -m g::--- {} \;
    find "$PASTA" -exec sudo setfacl -m o::--- {} \;
}

# Monitoramento com inotify
monitorar() {
    inotifywait -m -r -e create,modify,delete --format '%w%f %e %u' "$PASTA" |
    while read arquivo evento usuario; do
        if [ "$usuario" != "$USER_PERMITIDO" ] && [ "$usuario" != "root" ]; then
            echo "ALERTA: Usuário $usuario tentou $evento em $arquivo"
            # Opcional: reverter mudanças não autorizadas
        fi
    done
}
```

```
configurar_acls
# monitorar &  # Descomente para monitoramento contínuo
```

Usando Namespaces (Isolamento Adicional)

```
#!/bin/bash
# Script para executar app em namespace isolado

PASTA="/zona/segura"
USER_APP="app_writer"

# Cria namespace e monta pasta com permissões restritas
sudo unshare -m -U --map-root-user bash -c "
    # Remonta pasta como readonly para todos exceto usuário específico
    mount --bind $PASTA $PASTA

    # Executa aplicação como usuário correto
    sudo -u $USER_APP /caminho/para/aplicacao
"
```

Verificação e Teste

```
# Verificar ACLs configuradas
getfacl /zona/segura

# Testar acesso negado
sudo -u outro_usuario touch /zona/segura/teste.txt  # Deve falhar

# Testar acesso permitido
sudo -u app_writer touch /zona/segura/teste.txt     # Deve funcionar

# Monitorar tentativas de acesso
sudo tail -f /var/log/auth.log | grep "zona/segura"
```

Hardening Adicional

```
# 1. Audit logs
sudo auditctl -w /zona/segura -p wa -k zona_segura_acesso

# 2. Monitoramento contínuo
sudo tee /etc/systemd/system/monitor-zona.service << EOF
[Unit]
Description=Monitor Zona Segura
After=network.target

[Service]
Type=simple
ExecStart=/usr/local/bin/monitor-zona.sh
Restart=always

[Install]
WantedBy=multi-user.target
EOF

# 3. Backup de ACLs
```

```
getfacl -R /zona/segura > /backup/acls-zona-segura.txt
# Restaurar: setfacl --restore=/backup/acls-zona-segura.txt
```

Várias alternativas

1. Permissões Básicas (chmod/umask)

```
# Chmod recursivo
chmod -R 444 /pasta        # Somente leitura para todos
chmod -R a-wx /pasta       # Remove escrita e execução
chmod -R u-wx,g-wx,o-wx /pasta  # Sintaxe alternativa

# Umask para novos arquivos
umask 333  # Novos arquivos sem permissão de escrita/execução
```

2. Access Control Lists (ACLs)

```
# ACLs padrão (herança)
setfacl -R -m d:u::r--,d:g::r--,d:o::r-- /pasta
setfacl -R -m u::r--,g::r--,o::r-- /pasta

# ACLs para usuários específicos
setfacl -m u:usuario:r-- /pasta
```

3. Atributos de Arquivo (chattr)

```
# Imutável (nem root modifica facilmente)
chattr +i /pasta/*
chattr +i /pasta

# Somente adição (append-only)
chattr +a /pasta/*

# Sem dump
chattr +d /pasta/*

# Compressão
chattr +c /pasta/*
```

4. Montagem Somente Leitura

```
# Bind mount readonly
mount --bind /origem /destino
mount -o remount,ro /destino

# Loop mount readonly
mount -o loop,ro arquivo.img /ponto

# Tmpfs readonly
mount -t tmpfs -o ro tmpfs /ponto
```

7. SELinux/AppArmor

```
# SELinux contexts
semanage fcontext -a -t readonly_t "/pasta(/.*)?"
restorecon -R /pasta
```

```
# AppArmor profile
echo "/pasta/** r," >> /etc/apparmor.d/programa
```

8. Capabilities

```
# Remove capabilities de escrita
setcap cap_dac_override-ep /bin/programa
```

10. Quotas de Disco

```
# Quota zero para escrita
edquota -u usuario
# Define blocks e inodes como 0 para escrita
```

16. Hardlinks e Symlinks

```
# Hardlinks para arquivos protegidos
ln /arquivo_original /pasta_segura/arquivo

# Symlinks para pasta readonly
ln -s /pasta_readonly /pasta_acesso
```

17. Systemd

```
# Unit com ReadOnlyPaths
[Service]
ReadOnlyPaths=/pasta
ProtectSystem=strict
```

23. Monitoramento e Alertas

```
# inotify para detectar tentativas de escrita
inotifywait -m -r -e modify,create,delete /pasta

# auditd para logs
auditctl -w /pasta -p wa -k pasta_protegida
```