**04\_08 NOTA DE AULA (SEG DE ARQ-PASTAS)**

Windows Server -> Server Manager

\*\*\*Gerenciamento de Usuários e Grupos\*\*\*

Dentro do Server Manager -> Tools -> Computer Management -> Local Users and Groups -> Users

Para criar um novo usuário, botão direito -> New User

Apenas administradores ou usuários com permissão concedida pelo adm podem criar novos usuários

Dentro de Groups, é possível gerenciar os grupos que cada usuário será inserido

\_\_\_\_\_

\*\*\*Gerenciamento de Pastas e Permissões\*\*\*

C:\ Criar a pasta SOTurmaB e dentro dela 3 pastas: RH, Financeiro e Contabilidade

Abrir propriedade da pasta RH -> Aba Security. É possível gerenciar as permissões de grupos ou usuários sobre a pasta.

Qualquer pasta criada, por padrão, pode ser acessada por qualquer usuário da rede.

Para remover a permissão de um usuário, é preciso desabilitar a herança de permissões na aba Security -> Advanced -> Disable Inheritance.

Existem duas opções para desabilitar: Converter para permissão explícita, podendo definir individualmente o que cada usuário pode fazer nas pastas (allow) ou Desabilitar a herança de todos os usuários. Ambas opções permitem remover usuários e grupos de acessar a pasta, ou apenas definir ou "Allows" como read, modify e write, por exemplo.

**11\_08 NOTA DE AULA (SEG DE ARQ-PASTAS2)**

Windows Server -> Server Manager

\*\*\*Gerenciamento de Usuários e Grupos\*\*\*

Dentro do Server Manager -> Tools -> Computer Management -> Local Users and Groups -> Users

Criar 2 usuários: Rosa e Pedro. Eles trabalham no setor de rh, então é possível criar um grupo para eles.

Dentro de Local Users and Groups -> Groups, botão direito e New Group (rh).

Add -> Advanced -> Find Now para adicionar os 2 no grupo.

A partir dai, criar 3 grupos da seguinte forma:

rosa.silva e pedro.carlos -> RH

sonia.almeida e marcia.ribeiro -> Marketing

jaceline.santos e elis.regina -> Financeiro

\_\_\_\_\_\_\_\_\_

Criar a pasta Aula1108 dentro de C:\

Dentro de Aula1108 criar as pastas: rh, marketing, financeiro, softwares e comum. Vamos distribuir as permissões para acesso das pastas.

Apenas o grupo do rh pode acessar a pasta rh. O mesmo vale para a pasta e grupo de marketing e financeiro.

Na pasta softwares, todos os usuários podem ter acesso. Nesse caso, não é preciso acrescentar nenhum grupo, basta deixar o grupo Users com acesso.

Na pasta comum, todos usuários terão acesso (Grupo Users), porém apenas ler/gravar.

Na pasta financeiro terá uma exclusividade. O grupo do financeiro pode apenas ler. A usuária jaceline.santos poderá modificar a pasta financeiro. No caso, acontece uma pegadinha, pois Jaceline pode modificar e ela é do grupo financeiro que só pode ler. Então, o que prevalece é a maior permissão, ela terá permissão de modificar.

**18\_08 NOTA DE AULA (SEG DE ARQ-PASTAS LINUX)**

adduser -> cria um usuário

adduser rh1 -> cria o usuário rh1

Após isso, perguntará a senha 2x e alguns outros dados como nome, telefone, etc

Caso não queira preencher, basta pressionar enter e dps confirmar

/etc/passwd -> armazena todas as contas de usuário.

As contas são criadas na pasta /etc/passwd

nano /etc/passwd -> abre o arquivo /etc/passwd no editor nano

Dessa forma, temos a lista de usuários. No final da lista estão os usuários que criei.

O usuário (na lista) aparece no seguinte formato:

rh1:x:1001:1001:,,,:/home/rh1:/bin/bash

rh2:x:1002:1002:,,,:/home/rh1:/bin/bash

- O rh1 é o login do usuário

- O "x" é uma indicação de que a senha está armazenada no /etc/shadow

- O primeiro 1001 é a uid do usuário.

- O segundo 1001 é o gid do grupo do usuário (lembrando que quando se cria um novo usuário, automaticamente é criado um grupo no seu nome)

- As " , , , " são as informações adicionais como nome, telefone, etc que não foram preenchidas

- /home/rh1 é uma pasta para utilização pessoal do usuário

- /bin/bash é um interpretador de comando do linux. Vai interpretar tudo que eu digitar no terminal

/etc/shadow -> armazena as senhas criptografadas do usuário

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ctrl + alt + f2 -> abre um novo terminal

ctrl + alt + f3/f4/f5/f6/f7/f8/f9 -> vai abrindo um novo terminal 3/4/5/6/7/8/9

No novo terminal, ele pede o login/senha. Foi logado no rh1 que criamos.

Dando pwd no novo terminal teremos a resposta /home/rh1, como foi dito acima.

passwd -> troca a senha do usuário

exit ou logout -> efetua o logoff na conta do usuário

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ctrl + alt + f1 -> para voltar para o terminal 1 do root

Criar os usuários: rh1(já criado), rh2(já criado), compras1, compras2, fin1, fin2

addgroup -> cria um grupo

addgroup rh -> cria um grupo de nome rh

/etc/group -> armazena os grupos de usuários

Cada grupo tem seu gid, da mesma forma que os usuários tem seus ids.

gpasswd -a rh1 rh -> adiciona o usuário rh1 no grupo rh

Agora usando nano /etc/group, o grupo rh será listado da seguinte forma:

rh:x:1007:rh1 (mostrando que rh1 está no grupo)

Outra forma de adicionar usuários é dentro do arquivo de grupo, adicionar manualmente os usuários na linha dos grupos que eu quiser.

No caso, da seguinte forma:

rh:x:1007:rh1, rh2

groups rh1 -> exibe os grupos que o usuário rh1 pertence

groups rh2 -> exibe os grupos que o usuário rh2 pertence

gpasswd -d rh1 rh -> apaga o usuário rh1 do grupo rh

Assim como pode-se adicionar manualmente pelo arquivo de grupo, também é possível deletar.

De forma resumida, para adicionar ou deletar usuários para um grupo, pode-se executar via linha de comando ou editando o arquivo de grupo manualmente.

id rh1 -> exibe o id do usuário (uid) rh1 e de todos os grupos (gid) que ele pertence.

userdel -> apaga uma conta de usuário

userdel rh1 -> apaga a conta de usuário rh1

groupdel -> apaga o grupo

groupdel marketing -> apaga o grupo de usuário marketing

Outra forma de apagar usuário ou grupo é pelo nano, assim como se adiciona ou remove usuários de grupos.

**01\_09 NOTA DE AULA (SEG DE ARQ-PASTAS LINUX3)**

adduser xrh1

adduser xrh2

nano /etc/passwd

adduser ycompras1

adduser ycompras2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

addgroup grh

addgroup gcompras

nano /etc/group

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Adc xrh1 e xrh2 no grupo grh:

gpasswd -a xrh1 grh -> adc o usuário xrh1 no grupo grh

grh: xrh1, xrh2 -> outra forma é adc manualmente no arquivo nano (xrh2 para o grupo grh)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Adc ycompras1 e 2 no grupo gcompras:

gpasswd -a ycompras1 gcompras

gpasswd -a ycompras2 gcompras

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

groups ycompras1 -> mostra os grupos do ycompras1. Ele pertence ycompras1 e gcompras.

Lembrando que sempre que um usuário é criado, automaticamente um grupo com seu nome é criado.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

adduser sergio

addgroup lara

userdel ou deluser sergio

groupdel ou delgroup lara

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

cd /

mkdir aula0109

cd aula0109

touch arq1 proga estoque html

ls -l -< mostra as permissões dos arquivos ou diretórios

Ls -l é apresentado da seguinte forma:

-rw-r--r-- 1 root root 0 Set 1 19:31 arq1

-rw-r--r-- 1 root root 0 Set 1 19:31 estoque

-rw-r--r-- 1 root root 0 Set 1 19:31 html

-rw-r--r-- 1 root root 0 Set 1 19:31 proga

root root significa que o usuário root tem acesso ao arquivo e o grupo root tem acesso também, respectivamente

chgrp grh arq1 -> define que o grupo grh terá acesso no arquivo arq1

chgrp altera o grupo que terá acesso ao arquivo ou diretório

chgrp gcompras estoque -> define que o grupo gcompras terá acesso no arquivo estoque

ls -l -> para ver as permissões:

-rw-r--r-- 1 root grh 0 Set 1 19:31 arq1

-rw-r--r-- 1 root gcompras 0 Set 1 19:31 estoque

-rw-r--r-- 1 root root 0 Set 1 19:31 html

-rw-r--r-- 1 root root 0 Set 1 19:31 proga

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

chown -> altera o dono do arquivo ou diretório

chown xrh1 arq1 -> define que o usuário xrh1 será o dono do arquivo arq1

chown ycompras1 estoque -> define que o usuário ycompras1 será o dono do arquivo estoque

chown xrh2.grh proga -> define que o usuário xrh2 será o dono e o grupo grh terá acesso ao arquivo proga

chown xrh1.gcompras html -> defique que o usuário xrh1 será o dono e o grupo gcompras terá acesso ao arquivo html

ls -l -> para ver as permissões:

-rw-r--r-- 1 xrh1 grh 0 Set 1 19:31 arq1

-rw-r--r-- 1 ycompras1 gcompras 0 Set 1 19:31 estoque

-rw-r--r-- 1 xrh1 gcompras 0 Set 1 19:31 html

-rw-r--r-- 1 xrh2 grh 0 Set 1 19:31 proga

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ls -l arq1 -> permissões do arquivo arq1

-rw-r--r-- 1 xrh1 grh 0 Set 1 19:31 arq1

chmod -> altera permissões em cima de um arquivo (read, write, execute)

rw- -> u (u representa o dono do arq)

r-- -> g (g representa o grupo com acesso ao arq)

r-- -> o (o representa os outros)

chmod u+x, g+w, o+x arq1 -> Acrescenta a permissão para executar (x) para o dono de arq1

Acrescenta a permissão para gravar (w) para os grupos de arq1

Acrescenta a permissão para executar (x) para os outros

ou

chmod u=rwx, g=rw, o=rx arq1 -> Define ler (r), gravar (w) e executar (x) para o dono de arq1

Define ler (r) e gravar (w) para os grupos de e arq1

Define ler (r) e executar (x) para os outros

ls -l arq1

-rwxrw-r-x 1 xrh1 grh 0 Set 1 19:31 arq1

Lembrando que: r = Ler, w = Gravar e x = Executar

chmod u=rx, g+x, o+wx proga -> Define ler (r) e executar (x) para o dono de proga

Acrescenta executar (x) para os grupos de proga

Acrescenta gravar (w) e executar (x) para os outros

CADA UM DOS CÓDIGOS CHMOD DEVEM SER FEITO EM LINHA DIFERENTE.

COLOQUEI SEPARADO POR VÍRGULA PARA FICAR MAIS PRÁTICO DE DIGITAR

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Modo Octal de mudança de permissão:

1 - Executar

2 - Gravar

4 - Ler

A soma dos números correspondentes as permissões acima definem as permissões do dono, grupo e outros (respectivamente) sobre determinado arquivo

chmod 567 estoque

5 -> Ler (4) e Executar (1) para o dono do arquivo estoque

6 -> Ler (4) e Gravar (2) para os grupos do arquivo estoque

7 -> Ler (4), Gravar (2) e Executar (1) para os outros