Configuração do servidor NFS

Sumário

Índice de tabelas

Índice de Figuras

Capítulo 1

Configuração do servidor NFS

• Crias ACL'S e regras de acesso dos grupos do NFS;

• RPC (Remote Procedure Call);

•Trabalhar com o TCP/Wrappers utilizando Portmap.

1.1. Mãos a obra

O sistema de permissão para arquivos e diretórios em ambientes Unix/Linux utiliza o modelo UGO, onde é possível definir leitura, escrita e execução para o dono (u), grupo (g) e outros (g). Para melhorar e ampliar a segurança em ambientes de grande porte, com maior numero de maquinas, grupos e usuários, você pode utilizar extensões de permissionamento chamadas de ACLs.

Mas o que são ACLs?

As ACLs – (Access Control List) são listas de controle de acesso que estendem o nível de permissionamento de arquivos e diretórios padrão no mundo Unix/Linux.

Para utilizar instale o pacote acl.

```
# aptitude install acl
```

Após instalar o pacote precisamos configurar a partição para ter suporte as ACLs, isso porque as regras são baseadas em atributos estendidos no sistema de arquivos. Cada atributo estendido é um par atributo/valor associado a um arquivo ou diretório. Vamos a prática:

Abra o arquivo /etc/fstab e adicione a flag “acl” na coluna options

```
# vim /etc/fstab
```

```
/dev/sda5       /usr            ext3    defaults        0       2
/dev/sda6       /var            ext3    defaults,acl    0       2
/dev/sda8       /var/log        ext3    defaults        0       2
```

Veja em nosso exemplo que a flag foi adiciona ao ponto de montagem /var, que será compartilhada na rede via NFS.

Para aplicar as alterações remonte a partição com o comando mount

```
# mount – remount /var
```

Verifique se a partição esta montada com a opção acl

```
# mount | grep var
```

```
/dev/sda6 on /var type ext3 (rw,acl)
/dev/sda8 on /var/log type ext3 (rw)
```

Trabalhando com ACLs

Antes de começar a utilizar comandos com ACLs, crie o diretório que sera exportado via NFS, e copie alguns arquivos.

```
# mkdir /var/dados
```

```
# cp /etc/*.conf  /var/dados
```

Visualize o permissionamento estendido do diretório /var/dados com o comando getfacl.

```
# getfacl  /var/dados
```

```
getfacl: Removing leading '/' from absolute path names
# file: var/dados
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
```

Veja em nosso exemplo que o diretório possui o sistema de permissão padrão, onde o dono (root) pode ler, escrever e executar, grupo e outros só podem ler e executar resultando em forma octal o valor 755. Para adicionar permissões ACLs use o comando setfacl.

As principais opções de uso do setfacl são:

-m → Aplica permissão ACL;

-R → Aplica de forma recursiva a todos os arquivo e subdiretórios;

-b → Remove todas as permissões ACL;

-d → Aplica permissão padrão no diretório para criação de novos arquivos;

-k → Remove a permissão padrão.

Mas posso utilizar este tipo de solução junto com NFS?

Sim! E também com Samba. Agora nossa prática é exportar o diretório /var/dados através do arquivo /etc/exports. Vamos a prática:

Instale o pacote do NFS para Debian Lenny

 # aptitude install nfs-kernel-server

Abra o arquivo /etc/exports com o comando vim e adicione o seguinte conteúdo:

```
/var/dados      192.168.200.0/24(rw,no_subtree_check)
```

Em nosso exemplo o subdiretório dados que esta no diretório var, será compartilhado para a rede 192.168.200 com leitura e escrita.

Após gravar o arquivo releia os configuração do NFS

```
# exportfs -a
```

Configuração do cliente NFS

Nas maquinas clientes é necessário instalar o pacote nfs-common e portmap, e para visualizar que o servidor está compartilhando use o comando showmount -e + IP_do_servidor

```
# showmount -e 192.168.200.10
```

```
Export list for 192.168.200.10:
/var/dados 192.168.200.0/24
```

Com as informações que são o IP e o diretório exportado do servidor, será usado o comando mount para montar o compartilhamento.

```
# mount -t nfs  192.168.200.10:/var/dados /mnt
```

Verifique se o /mnt esta montado

```
# mount | grep mnt
```

```
192.168.200.10:/var/dados on /mnt type nfs (rw,addr=192.168.200.10)
```

Para você montar um compartilhamento via NFS ignorando as ACLs use

```
# mount -t nfs  192.168.200.10:/var/dados /mnt  -o noacl
```

Mas como as maquinas servidor e cliente se comunicam via NFS?

Os serviços que utilizam o portmap como NIS e NFS não utilizam uma porta em especifico, estes serviços enviam uma chamada RPC – Remote Procedure Call para a máquina servidora causando a execução de uma determina subrotina. Vamos acompanhar um passo a passo de como uma maquina cliente se comunica com uma maquina servidora.

1 – Tanto na maquina cliente e servidora está rodando o portmap, que utiliza uma porta fixa (111);

2 – A maquina servidora esta exportando um diretório via NFS;

3 – A maquina cliente está pronta para montar o diretório remoto da maquina servidora;

4 – A maquina cliente faz a requisição de NFS a um servidor na hora da montagem, enviando um RPC tipo NFS;

5 – O servidor que utiliza o portmapper monitora o serviço NFS, e ao receber uma conexão referente ao NFS, direciona o cliente para as portas certas (normalmente a porta 2049/UDP);

6 – O cliente recebe via RPC a resposta e completa a montagem. o cliente para as portas certas (normalmente a porta 2049/UDP).

TCP/Wrappers utilizando Portmap

Os TCP Wrappers são utilizados para aplicar regras de acesso aos servidores utilizados em seu sistema, podendo permitir ou negar as conexões a eles. Isso é possível através de um biblioteca chamada libwrap.

Mas como posso descobrir quais serviços utilizam esta biblioteca?

Através do comando ldd e o cominho do binário responsável pelo serviço. Exemplos:

SSH

```
# ldd $(which sshd) | grep libwrap
```

Todos os serviços que o super daemon inetd controla

```
# ldd $(which inetd) | grep libwrap
```

Portmap para liberar ou bloquear acessos ao servidor NFS

```
# ldd $(which portmap) | grep libwrap
```

Para controla esses acesso são usados dois arquivos, o /etc/hosts.allow para liberar acessos e o arquivo /etc/hosts.deny para configuraração de acessos negados para determinados Ips.

Exemplo prático:

Para bloquear o acesso ao servidor NFS e liberar apenas para a nossa rede:

```
# vim /etc/hosts.deny
```

```
portmap: ALL EXCEPT 192.168.200.0/24
```

Capítulo 2

Gerenciando

2.1. Objetivos

•Trobleshooting: Criar um compartilhamento seguro, utilizando ACL's.

2.1. Troubleshooting

Como posso melhorar a segurança nos compartilhamentos NFS?

Com o uso de ACL podemos personalizar o acesso a grupos e usuários aos diretórios que serão exportados via NFS. Vamos a prática:

Cenário:

Será exportado um diretório que ira pertencer ao grupo vendas, onde o dono é usuário root. Existem dois usuários, tux e linus que fazem parte deste grupo e assim poderão ler, escrever e executar resultando em uma permissão de forma octal 770.

Mas onde que as ACLs entram?

O desafio neste é cenário é dar permissão de leitura a um outro usuário, que não faz do grupo vendas e sim do grupo logística. O usuário aluno para o sistema de permissão padrão é considerado outros, assim perdendo total acesso ao diretório com permissão 770.

A solução neste caso é estender as permissão apenas para o usuário aluno, aumentando a segurança do diretório exportado.

Prática:

```
# setfacl -R -m u:aluno:r  /var/dados
```

Em nosso exemplo o comando setfacl adicionou permissão ACL de forma recursiva ao diretório /var/dados. Veja o resultado com o comando getfacl.

```
# getfacl   /var/dados
```

```
getfacl: Removing leading '/' from absolute path names
# file: var/dados
# owner: root
# group: vendas
user::rwx
user:aluno:r--
group::rwx
mask::rwx
other::---
```

Uma outra forma de verificar se o diretório possui permissão ACL, é usar o comando ls com as flags -ld. Na coluna de permissão será apresentado um sinal de "+"

```
# ls -ld   /var/dados
```

```
drwxrwx---+ 2 root vendas 4096 Nov 11 23:39 /var/dados
```