

Revisão do Tópico 209 – Compartilhamento de Arquivos

209.1 – Configurações do Servidor SAMBA

Processos e Configurações

O Samba funciona através de 2 processos principais:

- **smbd** – Responsável por possibilitar o compartilhamento de arquivos e impressoras através dos protocolos SMB ou CIFS
- **nmbd** – Processo que entende e pode responder requisições do protocolo NetBios, utilizado por sistemas Windows

Quanto às **portas** utilizadas temos:

- **smbd** – Portas TCP 139 e 445
- **nmbd** – Portas UDP 137 e 138

Há um terceiro processo utilizado quando o Samba está sendo inserido em um AD (Active Directory), que é o processo **winbind**. Esse processo TEM a função de fazer o Samba enxergar os usuários do AD.

Arquivos de Configuração e Logs

O principal arquivo de **configuração** é o **/etc/samba/smb.conf**.

No arquivo smb.conf, a seção [Global] conterá os parâmetros de configuração geral do Samba. Dentre as centenas de opções, podemos citar:

- **workgroup** – Define o workgroup (grupo de trabalho) ao qual o Samba fará parte
- **netbios name** – Nome do servidor Samba no protocolo NetBios
- **realm** – Usado para identificar o domínio ao qual o Samba será ingressado
- **server string** – Identificação do Servidor
- **wins support** – Usado para habilitar o suporte ao WINS (Windows Internet Name Service)
- **server role** – Usado para definir a forma de funcionamento do Samba: O mais comum e o que é coberto pelo LPIC-2 é o “standalone server”.
- **unix password sync** – Utilizado para permitir que ao se alterar a senha de um usuário no samba, pelo comando smbpasswd, a senha do mesmo usuário no Unix/Linux também seja alterada
- **log file** – Caminho e nome do arquivo de log
- **username map** – Mapeamento de Usuários (explicado a seguir)
- **security** – Modo de Segurança (explicado a seguir)

Os **arquivos de log** estão presentes no diretório **/var/log/samba**.

Compartilhamento de Diretórios

O compartilhamento de diretórios no Samba é feito através da definição de uma [tag] com o nome do compartilhamento e uma série de parâmetros. Os principais parâmetros são:

- `comment` = Descrição do Compartilhamento
- `path` = Caminho do diretório a ser compartilhado
- `guest ok` = Permitir ou não o acesso ao compartilhamento de usuários convidados, ou seja, usuários não cadastrados no samba
- `read only` = Define se o acesso será apenas para leitura
- `writable` = É o oposto do “read only”, define que o acesso pode ser feito com permissão de escrita
- `browseable` = Define se o compartilhamento será visível no servidor
- `valid users` = Define os usuários que poderão ter acesso ao compartilhamento

A tag **[Homes]** é utilizada para definir um compartilhamento padrão para todos os usuários do sistema.

Abaixo alguns exemplos de compartilhamento de diretórios.

[Homes]

```
comment = Home Directories
browseable = no
read only = no
```

[DiretorioX]

```
comment = Compartilhamento do Diretório X
path /opt/diretorioX
browseable = yes
writable = yes
guest ok = no
```

Um ponto importante é que sistemas Windows identificam qualquer compartilhamento cujo nome termine com o símbolo \$, como um compartilhamento oculto, os chamados “hidden shares”. Por exemplo:

[Secreto\$]

```
comment = Diretório Oculto
path = /tmp
guest ok = no
```

Compartilhamento de Impressoras

O Samba também pode ser utilizado para compartilhar com a rede impressoras instaladas no sistema. Para isso basta criar o seguinte compartilhamento:

[Printers]

```
comment = Impressoras
path = /var/spool/samba
browseable = yes
printable = yes
```

Também é importante criar o compartilhamento que disponibilizará os drives das impressoras instaladas no servidor, para que estes possam ser baixados e instalados pelas estações clientes (Windows):

[print\$]

```
comment = Drivers das Impressoras
path = /var/lib/samba/printers
browseable = yes
read only = yes
guest ok = no
```

Usuários no Samba

O Samba não utiliza os mesmos usuários configurados no Unix/Linux, contidos nos arquivos `passwd` e `shadow`. É preciso criar os usuários no Samba.

Para isso é utilizado o comando **smbpasswd**. As principais opções são:

- `-a` : adiciona um novo usuário
 - `-x` : deleta um usuário
 - `-d` : desabilita um usuário
 - `-e` : habilita um usuário
 - `<sem opções>` : Altera a senha de um usuário existente
-

Montando Remotamente os Compartilhamentos do Samba

O modo mais usado para montar no Linux os compartilhamentos criados no Samba é através do protocolo CIFS (Common Internet File System).

Com ele instalado, através do pacote `cifs-utils`, basta montar o compartilhamento através do comando **mount** ou pelo **/etc/fstab**.

Exemplo de uso do `mount`:

```
# mount -t cifs -o username=usuario-samba,password=senha //IP/Compartilhamento /mnt/diretorio
```

Outra opção é pela configuração do `fstab`. Alguns exemplos

```
//IP/Compartilhamento /mnt/diretorio cifs username=usuario-samba,password=senha 0 0
//IP/Compartilhamento /mnt/diretorio cifs credentials=arquivo 0 0
```

* Utilizando a opção `credentials`, o “arquivo” conterá as informações de usuário e senha.

Mapeamento de Usuários

Há no Samba uma opção chamada “**username map**” que precisa ser utilizada principalmente quando o nome do usuário no Windows é diferente do nome do usuário no Samba. Também é útil para fazer com que o Samba entenda vários usuários como um só.

A configuração é feita da seguinte maneira no arquivo `smb.conf`:

username map = `/etc/samba/nome-do-arquivo`

O arquivo indicado conterá uma linha para cada referência:
<usuário do samba> = <usuário informado no acesso>

Níveis de Segurança

Através do parâmetro “**security**” no smb.conf, é possível definir o nível ou tipo de segurança utilizado pela Samba.

São dois tipos macro: “user level” e “share level”, conforme descrito abaixo:

- **Share Level**
 - **share** = Cada compartilhamento possui uma senha para acesso
 - **User Level**
 - **user** (padrão) – O acesso aos compartilhamento é feito através da autenticação de usuários, com o uso de senhas individuais
 - **server** – A autenticação é feita por um servidor remoto (Não recomendado por falhas de segurança)
 - **domain** – O Samba irá validar o usuário e senha em um Controlador de Domínio Windows
 - **ads** – O Samba irá ingressar como membro de um AD (Active Directory)
-

Samba como Membro do AD

Primeiramente, para que o Samba possa ingressar em um AD, é preciso que o serviço **winbind** esteja disponível. É através deste serviço que o Samba conseguirá enxergar os usuários cadastrados no AD.

Referente às configurações, os seguintes parâmetros devem estar devidamente definidos:

- **workgroup** = Grupo de trabalho utilizado. Exemplo: DOMINIOEXEMPLO
- **realm** = Domínio utilizado no Windows. Exemplo: dominioexemplo.com.br
- **security** = ads
- **netbios name** = Nome do Samba no NetBios. Exemplo: servidor-samba

Outros pontos importantes:

- Configura o servidor em que o Samba está instalado para utilizar o servidor DNS do Windows
- Configurar o arquivo /etc/nsswitch.conf, incluindo a referência ao winbind ao final dos registros passwd e shadow.

Por fim, para fazer o Samba ingressar no AD é utilizado o comando “net”:

```
# net ads join -U Administrator
```

Para testar, basta utilizar o comando wbinfo:

```
# wbinfo -u : Para listar os usuários
```

```
# wbinfo --ping-dc : Para realizar um teste de acesso ao DC (Domain Controller)
```

Principais Comandos utilizados no Samba

Comando smbpasswd

Função: Criar e gerenciar os usuários do Samba.

Uso e Principais Opções:

- -a : adiciona um novo usuário
- -x : deleta um usuário
- -d : desabilita um usuário
- -e : habilita um usuário
- <sem opções> : Altera a senha de um usuário existente

Comando smbclient

Função: Permite o acesso a compartilhamentos feitos através dos protocolos SMB/CIFS, incluindo no Samba.

Uso e Principais Opções:

```
# smbclient //IP/Compartilhamento -U usuario
# smbclient -L //IP -U usuario : Lista os compartilhamentos disponíveis
```

Comando testparm

Função: Verifica a sintaxe das configurações do arquivo smb.conf, exibindo também os parâmetros definidos.

Uso e Principais Opções:

```
# testparm : Verifica a sintaxe e exibe os parâmetros definidos no arquivo
# testparm -v : Verifica a sintaxe e exibe todos os parâmetros do Samba, incluindo os parâmetros default.
```

Comando smbstatus

Função: Exibe detalhes das atuais conexões no Samba

Uso e Principais Opções:

```
# smbstatus : Exibe os processos e compartilhamentos
# smbstatus -S : Exibe apenas os compartilhamentos em uso
```

Comando nmblookup

Função: Faz a resolução de nomes NetBios em IP e vice versa

Uso e Principais Opções:

```
# nmblookup -S servidor-samba : Verifica o status do host pelo nome NetBios
```

nmblookup -A 192.168.1.210 : Verifica o status do host pelo IP

Comando samba-tool

Função: Ferramenta de administração do Samba. Muito utilizado quando o Samba é usado como Controlador de Domínios (PDC).

Uso e Principais Opções:

samba-tool processes : Lista os processos
samba-tool dbcheck : Verifica a base de dados do AD local

Comando smbcontrol

Função: Envia sinais e comandos aos processos do samba: smbd, nmbd, winbind

Uso e Principais Opções:

smbcontrol all reload-config
smbcontrol winbind shutdown

Comando net

Função: Ferramenta de administração similar ao comando net do Windows.

Uso e Principais Opções:

net -S localhost -U usuario share
net -S servidor-samba time

209.2 – Configurações do Servidor NFS

O NFS (Network File System) permite que diretórios compartilhados remotamente sejam montados como locais.

Nesse modelo, o **servidor** NFS é quem está disponibilizando um diretório para ser compartilhado e o **cliente** realiza a montagem desse compartilhamento remotamente.

A versão atual do NFS é a 4.1, mas o software de cliente/servidor suporta também todas as versões anteriores. A versão 3.0 ainda é muito utilizada.

Os Processos do NFS

Um servidor NFS funciona através de vários processos, entre eles temos 3 principais:

- **rcpbind** (portmap) : Realiza a função de mapeamento de portas (portmapper), ou seja, permite ao cliente identificar as portas em que está sendo disponibilizada o serviço. Essas portas são por padrão dinâmicas, então o serviço é essencial.
- **nfsd** : Processo que atende as requisições do cliente.
- **rpc.mountd** : Executa as solicitações encaminhadas pelo nfsd, montando efetivamente os compartilhamentos entre cliente-servidor.

Criando os Compartilhamentos

No servidor, os compartilhamentos devem ser criados no arquivo **/etc/exports** (mais comum), ou em arquivos dentro do diretório **/etc/exports.d/**.

Nesse arquivo, cada linha define um compartilhamento, no seguinte formato:
<Diretorio-Compartilhado> <Clientes-Permitidos>(Opções)

Dentre as opções informadas, as mais comuns e importantes são:

- **rw** – Permite Leitura e Escrita.
- **ro** – Permite Apenas Leitura (read-only). Opção padrão.
- **sync** – Operações síncronas. Opção padrão.
- **async** – Operações assíncronas, utilizando um cache interno.
- **no_subtree_check** – O subtree_check é feito quando um subdiretório é compartilhado mas a partição em que o diretório está não. Nesses casos o NFS faz uma checagem para verificar se o subdiretório continua na mesma partição. O no_subtree_check desabilita essa verificação. É a opção padrão.
- **root_squash** – Acessos como root no cliente são feitos com usuário anônimo (nobody) no servidor. Opção padrão.
- **no_root_squash** – Acessos como root no cliente são feitos como root no servidor.
- **all_squash** – Acessos de usuários comuns no cliente são feitos com usuário anônimo (nobody) no servidor.
- **no_all_squash** – Acessos de usuários comuns no cliente são feitos com o mesmo usuário (quando existir) no servidor. Opção padrão.

Por exemplo:

```
/var/log      192.168.1.111(ro,sync,no_subtree_check)
/opt/nfs      192.168.1.0/24(rw,async,no_subtree_check)
```

O comando exportfs

Após configurar os compartilhamentos, deve ser utilizado o comando “**exportfs**” para disponibilizá-los aos clientes. As principais opções do exportfs são:

- -a : Habilita todos os compartilhamentos definidos no /etc/exports
- -r : Reexporta os compartilhamentos, após mudanças nas configurações por exemplo.
- -ua : Desabilita todos os compartilhamentos
- -v : Exibe os compartilhamentos disponíveis com detalhes
- <sem opções> : Exibe os compartilhamentos disponíveis

O exportfs também pode ser utilizado para exportar um diretório temporariamente, utilizando-se o seguinte formato:

```
# exportfs <IP-Cliente>:<compartilhamento> -o <lista-de-opções>
```

Por exemplo:

```
# exportfs 192.168.1.220:/home/ricardo -o ro,async
```

Montando um Compartilhamento Remotamente

Os compartilhamentos são montados diretamente com o uso do comando **mount**, mais precisamente o **mount.nfs**, ou através do **/etc/fstab**.

A sintaxe do comando mount é:

```
# mount -t nfs <IP-Servidor>:<Compartilhamento> <Ponto-de-Montagem>
```

Por exemplo:

```
# mount -t nfs 192.168.1.210:/var/log /mnt/log-remoto
```

A definição no arquivo /etc/fstab segue o exemplo abaixo:

```
192.168.1.210:/var/log /mnt/log-remoto nfs defaults 0 0
```

O comando showmount

Função: O showmount é usado para listar os compartilhamentos disponíveis, tanto no servidor quanto a partir do cliente.

Uso e Principais Opções:

No servidor, basta utilizar:

```
# showmount -e
```

No cliente, o uso é:

```
# showmount -e IP-Servidor
```

O comando nfsstat

Função: Exibir estatísticas do NFS, tanto no cliente como no servidor.

Uso e Principais Opções:

As principais opções são:

- -m : Exibe estatísticas dos compartilhamento atualmente montados
 - -c : Estatísticas do cliente NFS
 - -s : Estatísticas do servidor NFS
 - -3 : Estatísticas dos compartilhamentos que usam a versão 3 do NFS
 - -4 : Estatísticas dos compartilhamentos que usam a versão 4 do NFS
 - -l : Exibe as estatísticas em um formato de lista
-

O comando rpcinfo

Função: Exibir informações do protocolo RCP (Remote Procedure Call), utilizado pelo NFS.

Uso e Principais Opções:

A principal opção é o “**rpcinfo -p**”, que exibe a lista de processos e portas utilizados no momento pelo NFS.

O Pseudo FileSystem

A partir da versão 4, ao invés do NFS exportar cada compartilhamento individualmente, todos os compartilhamentos são exportados como se fossem um filesystem único, o que é chamado de pseudo-filesystem.

Isso é feito internamente pelo NFS e é transparente para o cliente.

TCP Wrapper

Assim como outros serviços do Linux, o NFS também utiliza as bibliotecas do TCP Wrapper, dessa forma é possível utilizar os arquivos /etc/hosts.allow e /etc/hosts.deny para adicionar uma camada extra de segurança no uso do serviço.

Por exemplo, para liberar apenas o acesso da rede 192.168.1.0/24 ao servidor, pode ser feita a seguinte configuração:

No /etc/hosts.allow:
rpcbind: 192.168.1.0/24

No /etc/hosts.deny:
rpcbind: ALL

* Em alguns sistemas, ao invés de rpcbind o processo pode ter o nome portmap.