
Part VI. Introduction to Samba

Table of Contents

9. introduction to samba	137
9.1. verify installed version	138
9.2. installing samba	139
9.3. documentation	140
9.4. starting and stopping samba	141
9.5. samba daemons	142
9.6. the SMB protocol	143
9.7. practice: introduction to samba	144
10. getting started with samba	145
10.1. /etc/samba/smb.conf	146
10.2. /usr/bin/testparm	147
10.3. /usr/bin/smbclient	148
10.4. /usr/bin/smbtree	150
10.5. server string	151
10.6. Samba Web Administration Tool (SWAT)	152
10.7. practice: getting started with samba	153
10.8. solution: getting started with samba	154
11. a read only file server	156
11.1. Setting up a directory to share	157
11.2. configure the share	157
11.3. restart the server	158
11.4. verify the share	158
11.5. a note on netcat	160
11.6. practice: read only file server	161
11.7. solution: read only file server	162
12. a writable file server	163
12.1. set up a directory to share	164
12.2. share section in smb.conf	164
12.3. configure the share	164
12.4. test connection with windows	164
12.5. test writing with windows	165
12.6. How is this possible ?	165
12.7. practice: writable file server	166
12.8. solution: writable file server	167
13. samba first user account	168
13.1. creating a samba user	169
13.2. ownership of files	169
13.3. /usr/bin/smbpasswd	169
13.4. /etc/samba/smbpasswd	169
13.5. passdb backend	170
13.6. forcing this user	170
13.7. practice: first samba user account	171
13.8. solution: first samba user account	172
14. samba authentication	173
14.1. creating the users on Linux	174
14.2. creating the users on samba	174
14.3. security = user	174
14.4. configuring the share	175
14.5. testing access with net use	175
14.6. testing access with smbclient	175
14.7. verify ownership	176
14.8. common problems	176
14.9. practice : samba authentication	178
14.10. solution: samba authentication	179
15. samba securing shares	180

15.1. security based on user name	181
15.2. security based on ip-address	181
15.3. security through obscurity	182
15.4. file system security	182
15.5. practice: securing shares	184
15.6. solution: securing shares	185
16. samba domain member	187
16.1. changes in smb.conf	188
16.2. joining an Active Directory domain	189
16.3. winbind	190
16.4. wbinfo	190
16.5. getent	191
16.6. file ownership	192
16.7. practice : samba domain member	193
17. samba domain controller	194
17.1. about Domain Controllers	195
17.2. About security modes	195
17.3. About password backends	196
17.4. [global] section in smb.conf	196
17.5. netlogon share	197
17.6. other [share] sections	197
17.7. Users and Groups	198
17.8. tdbsam	198
17.9. about computer accounts	199
17.10. local or roaming profiles	199
17.11. Groups in NTFS acls	200
17.12. logon scripts	201
17.13. practice: samba domain controller	202
18. a brief look at samba 4	203
18.1. Samba 4 alpha 6	205

Chapter 9. introduction to samba

This introduction to the Samba server simply explains how to install Samba 3 and briefly mentions the SMB protocol.

9.1. verify installed version

9.1.1. .rpm based distributions

To see the version of samba installed on Red Hat, Fedora or CentOS use **rpm -q samba**.

```
[root@RHEL52 ~]# rpm -q samba  
samba-3.0.28-1.el5_2.1
```

The screenshot above shows that RHEL5 has **Samba** version 3.0 installed. The last number in the Samba version counts the number of updates or patches.

Below the same command on a more recent version of CentOS with Samba version 3.5 installed.

```
[root@centos6 ~]# rpm -q samba  
samba-3.5.10-116.el6_2.i686
```

9.1.2. .deb based distributions

Use **dpkg -l** or **aptitude show** on Debian or Ubuntu. Both Debian 7.0 (Wheezy) and Ubuntu 12.04 (Precise) use version 3.6.3 of the Samba server.

```
root@debian7~# aptitude show samba | grep Version  
Version: 2:3.6.3-1
```

Ubuntu 12.04 is currently at Samba version 3.6.3.

```
root@ubul204:~# dpkg -l samba | tail -1  
ii samba 2:3.6.3-2ubuntu2.1 SMB/CIFS file, print, and login server for Unix
```

9.2. installing samba

9.2.1. .rpm based distributions

Samba is installed by default on Red Hat Enterprise Linux. If Samba is not yet installed, then you can use the graphical menu (Applications -- System Settings -- Add/Remove Applications) and select "Windows File Server" in the Server section. The non-graphical way is to use **rpm** or **yum**.

When you downloaded the .rpm file, you can install Samba like this.

```
[paul@RHEL52 ~]$ rpm -i samba-3.0.28-1.el5_2.1.rpm
```

When you have a subscription to RHN (Red Hat Network), then **yum** is an easy tool to use. This **yum** command works by default on Fedora and CentOS.

```
[root@centos6 ~]# yum install samba
```

9.2.2. .deb based distributions

Ubuntu and Debian users can use the **aptitude** program (or use a graphical tool like Synaptic).

```
root@debian7~# aptitude install samba
The following NEW packages will be installed:
  samba samba-common{a} samba-common-bin{a} tdb-tools{a}
0 packages upgraded, 4 newly installed, 0 to remove and 1 not upgraded.
Need to get 15.1 MB of archives. After unpacking 42.9 MB will be used.
Do you want to continue? [Y/n/?]
...
```

9.3. documentation

9.3.1. samba howto

Samba comes with excellent documentation in html and pdf format (and also as a free download from samba.org and it is for sale as a printed book).

The documentation is a separate package, so install it if you want it on the server itself.

```
[root@centos6 ~]# yum install samba-doc
...
[root@centos6 ~]# ls -l /usr/share/doc/samba-doc-3.5.10/
total 10916
drwxr-xr-x. 6 root root    4096 May  6 15:50 htmldocs
-rw-r--r--. 1 root root 4605496 Jun 14  2011 Samba3-ByExample.pdf
-rw-r--r--. 1 root root  608260 Jun 14  2011 Samba3-Developers-Guide.pdf
-rw-r--r--. 1 root root  5954602 Jun 14  2011 Samba3-HOWTO.pdf
```

This action is very similar on Ubuntu and Debian except that the pdf files are in a separate package named **samba-doc-pdf**.

```
root@ubul204:~# aptitude install samba-doc-pdf
The following NEW packages will be installed:
  samba-doc-pdf
...
```

9.3.2. samba by example

Besides the howto, there is also an excellent book called **Samba By Example** (again available as printed edition in shops, and as a free pdf and html).

9.4. starting and stopping samba

You can start the daemons by invoking **/etc/init.d/smb start** (some systems use **/etc/init.d/samba**) on any linux.

```
root@laika:~# /etc/init.d/samba stop
* Stopping Samba daemons [ OK ]
root@laika:~# /etc/init.d/samba start
* Starting Samba daemons [ OK ]
root@laika:~# /etc/init.d/samba restart
* Stopping Samba daemons [ OK ]
* Starting Samba daemons [ OK ]
root@laika:~# /etc/init.d/samba status
* SMBD is running [ OK ]
```

Red Hat derived systems are happy with **service smb start**.

```
[root@RHEL4b ~]# /etc/init.d/smb start
Starting SMB services: [ OK ]
Starting NMB services: [ OK ]
[root@RHEL4b ~]# service smb restart
Shutting down SMB services: [ OK ]
Shutting down NMB services: [ OK ]
Starting SMB services: [ OK ]
Starting NMB services: [ OK ]
[root@RHEL4b ~]#
```


9.5. samba daemons

Samba 3 consists of three daemons, they are named **nmbd**, **smbd** and **winbindd**.

9.5.1. nmbd

The **nmbd** daemon takes care of all the names and naming. It registers and resolves names, and handles browsing. According to the Samba documentation, it should be the first daemon to start.

```
[root@RHEL52 ~]# ps -C nmbd
  PID TTY          TIME CMD
 5681 ?            00:00:00 nmbd
```

9.5.2. smbd

The **smbd** daemon manages file transfers and authentication.

```
[root@RHEL52 ~]# ps -C smbd
  PID TTY          TIME CMD
 5678 ?            00:00:00 smbd
 5683 ?            00:00:00 smbd
```

9.5.3. winbindd

The **winbind daemon** (winbindd) is only started to handle Microsoft Windows domain membership.

Note that **winbindd** is started by the **/etc/init.d/winbind** script (two dd's for the daemon and only one d for the script).

```
[root@RHEL52 ~]# /etc/init.d/winbind start
Starting Winbind services:                [ OK ]
[root@RHEL52 ~]# ps -C winbindd
  PID TTY          TIME CMD
 5752 ?            00:00:00 winbindd
 5754 ?            00:00:00 winbindd
```

On Debian and Ubuntu, the winbindd daemon is installed via a separate package called **winbind**.

9.6. the SMB protocol

9.6.1. brief history

Development of this protocol was started by **IBM** in the early eighties. By the end of the eighties, most development was done by **Microsoft**. SMB is an application level protocol designed to run on top of NetBIOS/NetBEUI, but can also be run on top of tcp/ip.

In 1996 Microsoft was asked to document the protocol. They submitted CIFS (Common Internet File System) as an internet draft, but it never got final rfc status.

In 2004 the European Union decided Microsoft should document the protocol to enable other developers to write compatible software. December 20th 2007 Microsoft came to an agreement. The Samba team now has access to SMB/CIFS, Windows for Workgroups and Active Directory documentation.

9.6.2. broadcasting protocol

SMB uses the NetBIOS **service location protocol**, which is a broadcasting protocol. This means that NetBIOS names have to be unique on the network (even when you have different IP-addresses). Having duplicate names on an SMB network can seriously harm communications.

9.6.3. NetBIOS names

NetBIOS names are similar to **hostnames**, but are always uppercase and only 15 characters in length. Microsoft Windows computers and Samba servers will broadcast this name on the network.

9.6.4. network bandwidth

Having many broadcasting SMB/CIFS computers on your network can cause bandwidth issues. A solution can be the use of a **NetBIOS name server** (NBNS) like **WINS** (Windows Internet Naming Service).

9.7. practice: introduction to samba

0. !! Make sure you know your student number, anything **ANYTHING** you name must include your student number!
1. Verify that you can logon to a Linux/Unix computer. Write down the name and ip address of this computer.
2. Do the same for all the other (virtual) machines available to you.
3. Verify networking by pinging the computer, edit the appropriate hosts files so you can use names. Test the names by pinging them.
4. Make sure Samba is installed, write down the version of Samba.
5. Open the Official Samba-3 howto pdf file that is installed on your computer. How many A4 pages is this file ? Then look at the same pdf on samba.org, it is updated regularly.
6. Stop the Samba server.

Chapter 10. getting started with samba

10.1. /etc/samba/smb.conf

10.1.1. smbd -b

Samba configuration is done in the **smb.conf** file. The file can be edited manually, or you can use a web based interface like webmin or swat to manage it. The file is usually located in /etc/samba. You can find the exact location with **smbd -b**.

```
[root@RHEL4b ~]# smbd -b | grep CONFIGFILE
CONFIGFILE: /etc/samba/smb.conf
```

10.1.2. the default smb.conf

The default smb.conf file contains a lot of examples with explanations.

```
[paul@RHEL4b ~]$ ls -l /etc/samba/smb.conf
-rw-r--r-- 1 root root 10836 May 30 23:08 /etc/samba/smb.conf
```

Also on Ubuntu and Debian, smb.conf is packed with samples and explanations.

```
paul@laika:~$ ls -l /etc/samba/smb.conf
-rw-r--r-- 1 root root 10515 2007-05-24 00:21 /etc/samba/smb.conf
```

10.1.3. minimal smb.conf

Below is an example of a very minimalistic **smb.conf**. It allows samba to start, and to be visible to other computers (Microsoft shows computers in Network Neighborhood or My Network Places).

```
[paul@RHEL4b ~]$ cat /etc/samba/smb.conf
[global]
workgroup = WORKGROUP
[firstshare]
path = /srv/samba/public
```

10.1.4. net view

Below is a screenshot of the **net view** command on Microsoft Windows Server 2003 sp2. It shows how a Red Hat Enterprise Linux 5.3 and a Ubuntu 9.04 Samba server, both with a minimalistic smb.conf, are visible to Microsoft computers nearby.

```
C:\Documents and Settings\Administrator>net view
Server Name          Remark
-----
\\LAIKA               Samba 3.3.2
\\RHEL53              Samba 3.0.33-3.7.el5
\\W2003
The command completed successfully.
```

10.1.5. long lines in smb.conf

Some parameters in smb.conf can get a long list of values behind them. You can continue a line (for clarity) on the next by ending the line with a backslash.

```
valid users = Serena, Venus, Lindsay \
```

```
Kim, Justine, Sabine \  
Amelie, Marie, Suzanne
```

10.1.6. curious smb.conf

Curious but true: `smb.conf` accepts synonyms like **create mode** and **create mask**, and (sometimes) minor spelling errors like **browsable** and **browseable**. And on occasion you can even switch words, the **guest only** parameter is identical to **only guest**. And **writable = yes** is the same as **readonly = no**.

10.1.7. man smb.conf

You can access a lot of documentation when typing **man smb.conf**.

```
[root@RHEL4b samba]# apropos samba  
cupsaddsmb          (8)  - export printers to samba for windows clients  
lmhosts             (5)  - The Samba NetBIOS hosts file  
net                 (8)  - Tool for administration of Samba and remote CIFS servers  
pdbedit            (8)  - manage the SAM database (Database of Samba Users)  
samba               (7)  - A Windows SMB/CIFS fileserver for UNIX  
smb.conf [smb]     (5)  - The configuration file for the Samba suite  
smbpasswd           (5)  - The Samba encrypted password file  
smbstatus           (1)  - report on current Samba connections  
swat                (8)  - Samba Web Administration Tool  
tdbbackup           (8)  - tool for backing up and ... of samba .tdb files  
[root@RHEL4b samba]#
```

10.2. /usr/bin/testparm

10.2.1. syntax check smb.conf

To verify the syntax of the `smb.conf` file, you can use **testparm**.

```
[paul@RHEL4b ~]$ testparm  
Load smb config files from /etc/samba/smb.conf  
Processing section "[firstshare]"  
Loaded services file OK.  
Server role: ROLE_STANDALONE  
Press enter to see a dump of your service definitions
```

10.2.2. testparm -v

An interesting option is **testparm -v**, which will output all the global options with their default value.

```
[root@RHEL52 ~]# testparm -v | head  
Load smb config files from /etc/samba/smb.conf  
Processing section "[pub0]"  
Processing section "[global$]"  
Loaded services file OK.  
Server role: ROLE_STANDALONE  
Press enter to see a dump of your service definitions  
  
[global]  
dos charset = CP850  
unix charset = UTF-8  
display charset = LOCALE  
workgroup = WORKGROUP
```

```
realm =
netbios name = TEACHER0
netbios aliases =
netbios scope =
server string = Samba 3.0.28-1.el5_2.1
...
```

There were about 350 default values for smb.conf parameters in Samba 3.0.x. This number grew to almost 400 in Samba 3.5.x.

10.2.3. testparm -s

The samba daemons are constantly (once every 60 seconds) checking the smb.conf file, so it is good practice to keep this file small. But it is also good practice to document your samba configuration, and to explicitly set options that have the same default values. The **testparm -s** option allows you to do both. It will output the smallest possible samba configuration file, while retaining all your settings. The idea is to have your samba configuration in another file (like smb.conf.full) and let testparm parse this for you. The screenshot below shows you how. First the smb.conf.full file with the explicitly set option workgroup to WORKGROUP.

```
[root@RHEL4b samba]# cat smb.conf.full
[global]
workgroup = WORKGROUP

# This is a demo of a documented smb.conf
# These two lines are removed by testparm -s

server string = Public Test Server

[firstshare]
path = /srv/samba/public
```

Next, we execute testparm with the -s option, and redirect stdout to the real **smb.conf** file.

```
[root@RHEL4b samba]# testparm -s smb.conf.full > smb.conf
Load smb config files from smb.conf.full
Processing section "[firstshare]"
Loaded services file OK.
```

And below is the end result. The two comment lines and the default option are no longer there.

```
[root@RHEL4b samba]# cat smb.conf
# Global parameters
[global]
server string = Public Test Server

[firstshare]
path = /srv/samba/public
[root@RHEL4b samba]#
```

10.3. /usr/bin/smbclient

10.3.1. smbclient looking at Samba

With **smbclient** you can see browsing and share information from your smb server. It will display all your shares, your workgroup, and the name of the Master Browser. The -N switch

is added to avoid having to enter an empty password. The `-L` switch is followed by the name of the host to check.

```
[root@RHEL4b init.d]# smbclient -NL rhel4b
Anonymous login successful
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.0.10-1.4E.9]

Sharename      Type      Comment
-----
firstshare     Disk
IPC$           IPC       IPC Service (Public Test Server)
ADMIN$         IPC       IPC Service (Public Test Server)
Anonymous login successful
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.0.10-1.4E.9]

Server          Comment
-----
RHEL4B          Public Test Server
WINXP

Workgroup       Master
-----
WORKGROUP       WINXP
```

10.3.2. smbclient anonymous

The screenshot below uses **smbclient** to display information about a remote smb server (in this case a computer with Ubuntu 11.10).

```
root@ubull110:/etc/samba# testparm smbclient -NL 127.0.0.1
Anonymous login successful
Domain=[LINUXTR] OS=[Unix] Server=[Samba 3.5.11]

Sharename      Type      Comment
-----
share1         Disk
IPC$           IPC       IPC Service (Samba 3.5.11)
Anonymous login successful
Domain=[LINUXTR] OS=[Unix] Server=[Samba 3.5.11]

Server          Comment
-----

Workgroup       Master
-----
LINUXTR         DEBIAN6
WORKGROUP       UBU1110
```

10.3.3. smbclient with credentials

Windows versions after xp sp2 and 2003 sp1 do not accept guest access (the NT_STATUS_ACCESS_DENIED error). This example shows how to provide credentials with **smbclient**.

```
[paul@RHEL53 ~]$ smbclient -L w2003 -U administrator%stargate
Domain=[W2003] OS=[Windows Server 2003 3790 Service Pack 2] Server=...

Sharename      Type      Comment
-----
C$             Disk     Default share
```



```
IPC$          IPC          Remote IPC
ADMIN$        Disk         Remote Admin
...
```

10.4. /usr/bin/smbtree

Another useful tool to troubleshoot Samba or simply to browse the SMB network is **smbtree**. In its simplest form, smbtree will do an anonymous browsing on the local subnet, displaying all SMB computers and (if authorized) their shares.

Let's take a look at two screenshots of smbtree in action (with blank password). The first one is taken immediately after booting four different computers (one MS Windows 2000, one MS Windows xp, one MS Windows 2003 and one RHEL 4 with Samba 3.0.10).

```
[paul@RHEL4b ~]$ smbtree
Password:
WORKGROUP
PEGASUS
  \WINXP
  \RHEL4B                                Pegasus Domain Member Server
Error connecting to 127.0.0.1 (Connection refused)
cli_full_connection: failed to connect to RHEL4B<20> (127.0.0.1)
  \HM2003
[paul@RHEL4b ~]$
```

The information displayed in the previous screenshot looks incomplete. The browsing elections are still ongoing, the browse list is not yet distributed to all clients by the (to be elected) browser master. The next screenshot was taken about one minute later. And it shows even less.

```
[paul@RHEL4b ~]$ smbtree
Password:
WORKGROUP
  \W2000
[paul@RHEL4b ~]$
```

So we wait a while, and then run **smbtree** again, this time it looks a lot nicer.

```
[paul@RHEL4b ~]$ smbtree
Password:
WORKGROUP
  \W2000
PEGASUS
  \WINXP
  \RHEL4B                                Pegasus Domain Member Server
    \RHEL4B\ADMIN$                      IPC Service (Pegasus Domain Member Server)
    \RHEL4B\IPC$                        IPC Service (Pegasus Domain Member Server)
    \RHEL4B\domaindata                  Active Directory users only
  \HM2003
[paul@RHEL4b ~]$ smbtree --version
Version 3.0.10-1.4E.9
[paul@RHEL4b ~]$
```

I added the version number of **smbtree** in the previous screenshot, to show you the difference when using the latest version of smbtree (below a screenshot taken from Ubuntu Feisty Fawn). The latest version shows a more complete overview of machines and shares.

```
paul@laika:~$ smbtree --version
Version 3.0.24
```

```
paul@laika:~$ smbtree
Password:
WORKGROUP
  \\W2000
    \\W2000\firstshare
    \\W2000\C$           Default share
    \\W2000\ADMIN$      Remote Admin
    \\W2000\IPC$        Remote IPC
PEGASUS
  \\WINXP
cli_rpc_pipe_open: cli_nt_create failed on pipe \srvsvc to machine WINXP.
Error was NT_STATUS_ACCESS_DENIED
  \\RHEL4B               Pegasus Domain Member Server
    \\RHEL4B\ADMIN$      IPC Service (Pegasus Domain Member Server)
    \\RHEL4B\IPC$        IPC Service (Pegasus Domain Member Server)
    \\RHEL4B\domaindata  Active Directory users only
  \\HM2003
cli_rpc_pipe_open: cli_nt_create failed on pipe \srvsvc to machine HM2003.
Error was NT_STATUS_ACCESS_DENIED
paul@laika:~$
```

The previous screenshot also provides useful errors on why we cannot see shared info on computers winxp and w2003. Let us try the old **smbtree** version on our RHEL server, but this time with Administrator credentials (which are the same on all computers).

```
[paul@RHEL4b ~]$ smbtree -UAdministrator%Stargatel
WORKGROUP
  \\W2000
PEGASUS
  \\WINXP
    \\WINXP\C$           Default share
    \\WINXP\ADMIN$      Remote Admin
    \\WINXP\share55
    \\WINXP\IPC$        Remote IPC
  \\RHEL4B               Pegasus Domain Member Server
    \\RHEL4B\ADMIN$      IPC Service (Pegasus Domain Member Server)
    \\RHEL4B\IPC$        IPC Service (Pegasus Domain Member Server)
    \\RHEL4B\domaindata  Active Directory users only
  \\HM2003
    \\HM2003\NETLOGON    Logon server share
    \\HM2003\SYSVOL      Logon server share
    \\HM2003\WSUSTemp    A network share used by Local Publishing ...
    \\HM2003\ADMIN$      Remote Admin
    \\HM2003\tools
    \\HM2003\IPC$        Remote IPC
    \\HM2003\WsusContent A network share to be used by Local ...
    \\HM2003\C$          Default share
[paul@RHEL4b ~]$
```

As you can see, this gives a very nice overview of all SMB computers and their shares.

10.5. server string

The comment seen by the **net view** and the **smbclient** commands is the default value for the **server string** option. Simply adding this value to the global section in **smb.conf** and restarting samba will change the option.

```
[root@RHEL53 samba]# testparm -s 2>/dev/null | grep server
server string = Red Hat Server in Paris
```

After a short while, the changed option is visible on the Microsoft computers.

```
C:\Documents and Settings\Administrator>net view
Server Name                Remark
-----
\\LAIKA                    Ubuntu 9.04 server in Antwerp
\\RHEL53                   Red Hat Server in Paris
\\W2003
```

10.6. Samba Web Administration Tool (SWAT)

Samba comes with a web based tool to manage your samba configuration file. **SWAT** is accessible with a web browser on port 901 of the host system. To enable the tool, first find out whether your system is using the **inetd** or the **xinetd** superdaemon.

```
[root@RHEL4b samba]# ps fax | grep inet
15026 pts/0    S+      0:00      \_ grep inet
2771 ?          Ss      0:00 xinetd -stayalive -pidfile /var/run/xinetd.pid
[root@RHEL4b samba]#
```

Then edit the **inetd.conf** or change the **disable = yes** line in **/etc/xinetd.d/swat** to **disable = no**.

```
[root@RHEL4b samba]# cat /etc/xinetd.d/swat
# default: off
# description: SWAT is the Samba Web Admin Tool. Use swat \
#              to configure your Samba server. To use SWAT, \
#              connect to port 901 with your favorite web browser.
service swat
{
    port                = 901
    socket_type         = stream
    wait                = no
    only_from           = 127.0.0.1
    user                = root
    server               = /usr/sbin/swat
    log_on_failure      += USERID
    disable              = no
}
[root@RHEL4b samba]# /etc/init.d/xinetd restart
Stopping xinetd:          [ OK ]
Starting xinetd:          [ OK ]
[root@RHEL4b samba]#
```

Change the **only from** value to enable swat from remote computers. This examples shows how to provide swat access to all computers in a /24 subnet.

```
[root@RHEL53 xinetd.d]# grep only /etc/xinetd.d/swat
only_from = 192.168.1.0/24
```

Be careful when using SWAT, it erases all your manually edited comments in **smb.conf**.

10.7. practice: getting started with samba

1. Take a backup copy of the original smb.conf, name it smb.conf.orig
2. Enable SWAT and take a look at it.
3. Stop the Samba server.
4. Create a minimalistic smb.conf.minimal and test it with testparm.
5. Use testparm -s to create /etc/samba/smb.conf from your smb.conf.minimal .
6. Start Samba with your minimal smb.conf.
7. Verify with smbclient that your Samba server works.
8. Verify that another (Microsoft) computer can see your Samba server.
9. Browse the network with net view, smbtree and with Windows Explorer.
10. Change the "Server String" parameter in smb.conf. How long does it take before you see the change (net view, smbclient, My Network Places,...) ?
11. Will restarting Samba after a change to smb.conf speed up the change ?
12. Which computer is the master browser master in your workgroup ? What is the master browser ?
13. If time permits (or if you are waiting for other students to finish this practice), then install a sniffer (wireshark) and watch the browser elections.

10.8. solution: getting started with samba

1. Take a backup copy of the original smb.conf, name it smb.conf.orig

```
cd /etc/samba ; cp smb.conf smb.conf.orig
```

2. Enable SWAT and take a look at it.

```
on Debian/Ubuntu: vi /etc/inetd.conf (remove # before swat)
```

```
on RHEL/Fedora: vi /etc/xinetd.d/swat (set disable to no)
```

3. Stop the Samba server.

```
/etc/init.d/smb stop (Red Hat)
```

```
/etc/init.d/samba stop (Debian)
```

4. Create a minimalistic smb.conf.minimal and test it with testparm.

```
cd /etc/samba ; mkdir my_smb_confs ; cd my_smb_confs
```

```
vi smb.conf.minimal
```

```
testparm smb.conf.minimal
```

5. Use testparm -s to create /etc/samba/smb.conf from your smb.conf.minimal .

```
testparm -s smb.conf.minimal > ../smb.conf
```

6. Start Samba with your minimal smb.conf.

```
/etc/init.d/smb restart (Red Hat)
```

```
/etc/init.d/samba restart (Debian)
```

7. Verify with smbclient that your Samba server works.

```
smbclient -NL 127.0.0.1
```

8. Verify that another computer can see your Samba server.

```
smbclient -NL 'ip-address' (on a Linux)
```

9. Browse the network with net view, smbtree and with Windows Explorer.

```
on Linux: smbtree
```

```
on Windows: net view (and WindowsKey + e)
```

10. Change the "Server String" parameter in smb.conf. How long does it take before you see the change (net view, smbclient, My Network Places,...) ?

```
vi /etc/samba/smb.conf
```

```
(should take only seconds when restarting samba)
```

11. Will restarting Samba after a change to smb.conf speed up the change ?

```
yes
```

12. Which computer is the master browser master in your workgroup ? What is the master browser ?

The computer that won the elections.

This machine will make the list of computers in the network

13. If time permits (or if you are waiting for other students to finish this practice), then install a sniffer (wireshark) and watch the browser elections.

On ubuntu: `sudo aptitude install wireshark`

then: `sudo wireshark`, select interface

Chapter 11. a read only file server

11.1. Setting up a directory to share

Let's start with setting up a very simple read only file server with Samba. Everyone (even anonymous guests) will receive read access.

The first step is to create a directory and put some test files in it.

```
[root@RHEL52 ~]# mkdir -p /srv/samba/readonly
[root@RHEL52 ~]# cd /srv/samba/readonly/
[root@RHEL52 readonly]# echo "It is cold today." > winter.txt
[root@RHEL52 readonly]# echo "It is hot today." > summer.txt
[root@RHEL52 readonly]# ls -l
total 8
-rw-r--r-- 1 root root 17 Jan 21 05:49 summer.txt
-rw-r--r-- 1 root root 18 Jan 21 05:49 winter.txt
[root@RHEL52 readonly]#
```

11.2. configure the share

11.2.1. smb.conf [global] section

In this example the samba server is a member of WORKGROUP (the default workgroup). We also set a descriptive server string, this string is visible to users browsing the network with net view, windows explorer or smbclient.

```
[root@RHEL52 samba]# head -5 smb.conf
[global]
workgroup = WORKGROUP
server string = Public Anonymous File Server
netbios name = TEACHER0
security = share
```

You might have noticed the line with **security = share**. This line sets the default security mode for our samba server. Setting the security mode to **share** will allow clients (smbclient, any windows, another Samba server, ...) to provide a password for each share. This is one way of using the SMB/CIFS protocol. The other way (called **user mode**) will allow the client to provide a username/password combination, before the server knows which share the client wants to access.

11.2.2. smb.conf [share] section

The share is called pubread and the path is set to our newly created directory. Everyone is allowed access (**guest ok = yes**) and security is set to read only.

```
[pubread]
path = /srv/samba/readonly
comment = files to read
read only = yes
guest ok = yes
```

Here is a very similar configuration on Ubuntu 11.10.


```
root@ubull10:~# cat /etc/samba/smb.conf
[global]
workgroup = LINUXTR
netbios name = UBU1110
security = share
[roshare1]
path = /srv/samba/readonly
read only = yes
guest ok = yes
```

It doesn't really matter which Linux distribution you use. Below the same config on Debian 6, as good as identical.

```
root@debian6:~# cat /etc/samba/smb.conf
[global]
workgroup = LINUXTR
netbios name = DEBIAN6
security = share
[roshare1]
path = /srv/samba/readonly
read only = yes
guest ok = yes
```

11.3. restart the server

After testing with **testparm**, restart the samba server (so you don't have to wait).

```
[root@RHEL4b readonly]# service smb restart
Shutting down SMB services:                [ OK ]
Shutting down NMB services:                [ OK ]
Starting SMB services:                     [ OK ]
Starting NMB services:                     [ OK ]
```

11.4. verify the share

11.4.1. verify with smbclient

You can now verify the existence of the share with **smbclient**. Our **pubread** is listed as the fourth share.

```
[root@RHEL52 samba]# smbclient -NL 127.0.0.1
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.0.33-3.7.e15]

  Sharename      Type            Comment
  -----
  IPC$           IPC            IPC Service (Public Anonymous File Server)
  global$       Disk
  pub0          Disk
  pubread       Disk          files to read
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.0.33-3.7.e15]

  Server          Comment
  -----
  TEACHER0       Samba 3.0.33-3.7.e15
  W2003EE

  Workgroup       Master
  -----
  WORKGROUP       W2003EE
```

11.4.2. verify on windows

The final test is to go to a Microsoft windows computer and read a file on the Samba server. First we use the **net use** command to mount the pubread share on the driveletter k.

```
C:\>net use K: \\teacher0\pubread
The command completed successfully.
```

Then we test looking at the contents of the share, and reading the files.

```
C:\>dir k:
Volume in drive K is pubread
Volume Serial Number is 0C82-11F2

Directory of K:\

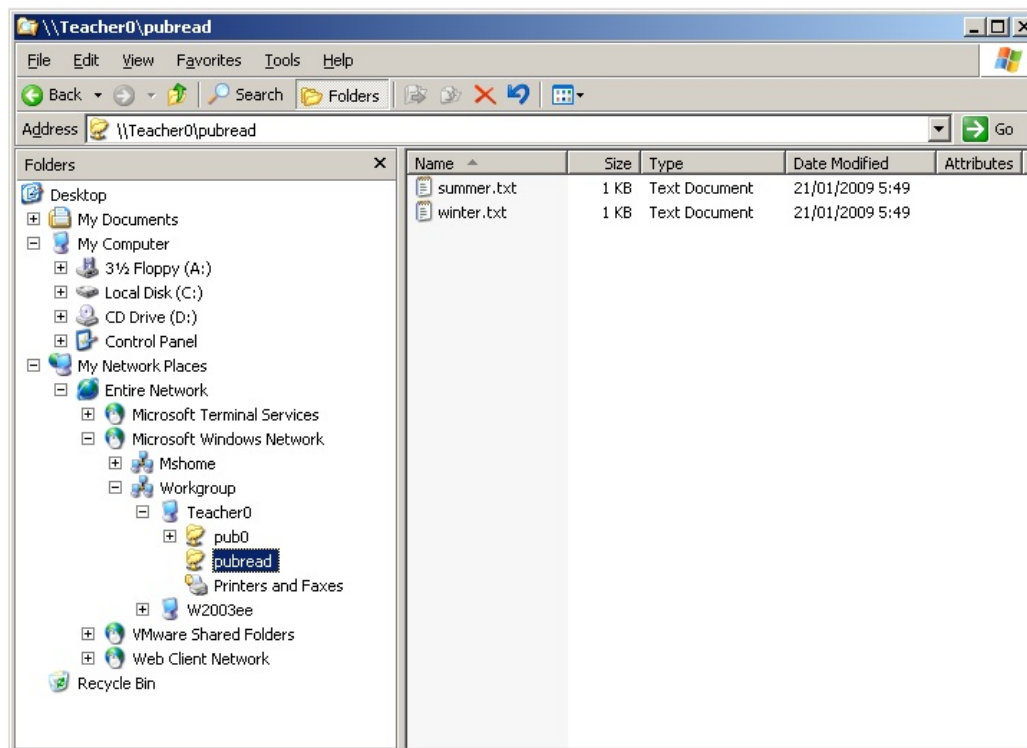
21/01/2009  05:49    <DIR>          .
21/01/2009  05:49    <DIR>          ..
21/01/2009  05:49                   17 summer.txt
21/01/2009  05:49                   18 winter.txt
               2 File(s)                35 bytes
               2 Dir(s)  13.496.242.176 bytes free
```

Just to be on the safe side, let us try writing.

```
K:\>echo very cold > winter.txt
Access is denied.
```

```
K:\>
```

Or you can use windows explorer...



11.5. a note on netcat

The Windows command line screenshot is made in a Linux console, using **netcat** as a pipe to a Windows command shell.

The way this works, is by enabling netcat to listen on the windows computer to a certain port, executing cmd.exe when a connection is received. Netcat is similar to cat, in the way that cat does nothing, only netcat does nothing over the network.

To enable this connection, type the following on the windows computer (after downloading netcat for windows).

```
nc -l -p 23 -t -e cmd.exe
```

And then connect to this machine with netcat from any Linux computer. You end up with a cmd.exe prompt inside your Linux shell.

```
paul@laika:~$ nc 192.168.1.38 23
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\>net use k: /delete
net use k: /delete
k: was deleted successfully.
```

11.6. practice: read only file server

1. Create a directory in a good location (FHS) to share files for everyone to read.
2. Make sure the directory is owned properly and is world accessible.
3. Put a textfile in this directory.
4. Share the directory with Samba.
5. Verify from your own and from another computer (smbclient, net use, ...) that the share is accessible for reading.
6. Make a backup copy of your smb.conf, name it smb.conf.ReadOnlyFileServer.

11.7. solution: read only file server

1. Create a directory in a good location (FHS) to share files for everyone to read.

choose one of these...

```
mkdir -p /srv/samba/readonly
```

```
mkdir -p /home/samba/readonly
```

```
/home/paul/readonly is wrong!!
```

```
/etc/samba/readonly is wrong!!
```

```
/readonly is wrong!!
```

2. Make sure the directory is owned properly and is world accessible.

```
chown root:root /srv/samba/readonly
```

```
chmod 755 /srv/samba/readonly
```

3. Put a textfile in this directory.

```
echo Hello World > hello.txt
```

4. Share the directory with Samba.

You smb.conf.readonly could look like this:

```
[global]
workgroup = WORKGROUP
server string = Read Only File Server
netbios name = STUDENTx
security = share
```

```
[readonlyX]
path = /srv/samba/readonly
comment = read only file share
read only = yes
guest ok = yes
```

test with testparm before going in production!

5. Verify from your own and from another computer (smbclient, net use, ...) that the share is accessible for reading.

On Linux: `smbclient -NL 127.0.0.1`

On Windows Explorer: browse to My Network Places

On Windows cmd.exe: `net use L: //studentx/readonly`

6. Make a backup copy of your smb.conf, name it smb.conf.ReadOnlyFileServer.

```
cp smb.conf smb.conf.ReadOnlyFileServer
```

Chapter 12. a writable file server

12.1. set up a directory to share

In this second example, we will create a share where everyone can create files and write to files. Again, we start by creating a directory

```
[root@RHEL52 samba]# mkdir -p /srv/samba/writable
[root@RHEL52 samba]# chmod 777 /srv/samba/writable/
```

12.2. share section in smb.conf

There are two parameters to make a share writable. We can use **read only** or **writable**. This example shows how to use **writable** to give write access to a share.

```
writable = yes
```

And this is an example of using the **read only** parameter to give write access to a share.

```
read only = no
```

12.3. configure the share

Then we simply add a share to our file server by editing **smb.conf**. Below the check with **testparm**. (We could have changed the description of the server...)

```
[root@RHEL52 samba]# testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[pubwrite]"
Processing section "[pubread]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions

[global]
  netbios name = TEACHER0
  server string = Public Anonymous File Server
  security = SHARE

[pubwrite]
  comment = files to write
  path = /srv/samba/writable
  read only = No
  guest ok = Yes

[pubread]
  comment = files to read
  path = /srv/samba/readonly
  guest ok = Yes
```

12.4. test connection with windows

We can now test the connection on a windows 2003 computer. We use the **net use** for this.

```
C:\>net use L: \\teacher0\pubwrite
net use L: \\teacher0\pubwrite
The command completed successfully.
```

12.5. test writing with windows

We mounted the **pubwrite** share on the L: drive in windows. Below we test that we can write to this share.

```
L:\>echo hoi > hoi.txt

L:\>dir
Volume in drive L is pubwrite
Volume Serial Number is 0C82-272A

Directory of L:\

21/01/2009  06:11    <DIR>          .
21/01/2009  06:11    <DIR>          ..
21/01/2009  06:16                6 hoi.txt
               1 File(s)                6 bytes
               2 Dir(s)  13.496.238.080 bytes free
```

12.6. How is this possible ?

Linux (or any Unix) always needs a user account to gain access to a system. The windows computer did not provide the samba server with a user account or a password. Instead, the Linux owner of the files created through this writable share is the Linux guest account (usually named nobody).

```
[root@RHEL52 samba]# ls -l /srv/samba/writable/
total 4
-rwxr--r-- 1 nobody nobody 6 Jan 21 06:16 hoi.txt
```

So this is not the cleanest solution. We will need to improve this.

12.7. practice: writable file server

1. Create a directory and share it with Samba.
2. Make sure everyone can read and write files, test writing with smbclient and from a Microsoft computer.
3. Verify the ownership of files created by (various) users.

12.8. solution: writable file server

1. Create a directory and share it with Samba.

```
mkdir /srv/samba/writable
```

```
chmod 777 /srv/samba/writable
```

the share section in smb.conf can look like this:

```
[pubwrite]
path = /srv/samba/writable
comment = files to write
read only = no
guest ok = yes
```

2. Make sure everyone can read and write files, test writing with smbclient and from a Microsoft computer.

to test writing with smbclient:

```
echo one > count.txt
echo two >> count.txt
echo three >> count.txt
smbclient //localhost/pubwrite
Password:
smb: \> put count.txt
```

3. Verify the ownership of files created by (various) users.

```
ls -l /srv/samba/writable
```

Chapter 13. samba first user account

13.1. creating a samba user

We will create a user for our samba file server and make this user the owner of the directory and all of its files. This anonymous user gets a clear description, but does not get a login shell.

```
[root@RHEL52 samba]# useradd -s /bin/false sambanobody
[root@RHEL52 samba]# usermod -c "Anonymous Samba Access" sambanobody
[root@RHEL52 samba]# passwd sambanobody
Changing password for user sambanobody.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

13.2. ownership of files

We can use this user as owner of files and directories, instead of using the root account. This approach is clear and more secure.

```
[root@RHEL52 samba]# chown -R sambanobody:sambanobody /srv/samba/
[root@RHEL52 samba]# ls -al /srv/samba/writable/
total 12
drwxrwxrwx 2 sambanobody sambanobody 4096 Jan 21 06:11 .
drwxr-xr-x 6 sambanobody sambanobody 4096 Jan 21 06:11 ..
-rwxr--r-- 1 sambanobody sambanobody    6 Jan 21 06:16 hoi.txt
```

13.3. /usr/bin/smbpasswd

The sambanobody user account that we created in the previous examples is not yet used by samba. It just owns the files and directories that we created for our shares. The goal of this section is to force ownership of files created through the samba share to belong to our sambanobody user. Remember, our server is still accessible to everyone, nobody needs to know this user account or password. We just want a clean Linux server.

To accomplish this, we first have to tell Samba about this user. We can do this by adding the account to **smbpasswd**.

```
[root@RHEL52 samba]# smbpasswd -a sambanobody
New SMB password:
Retype new SMB password:
Added user sambanobody.
```

13.4. /etc/samba/smbpasswd

To find out where Samba keeps this information (for now), use **smbd -b**. The **PRIVATE_DIR** variable will show you where the smbpasswd database is located.

```
[root@RHEL52 samba]# smbd -b | grep PRIVATE
PRIVATE_DIR: /etc/samba
[root@RHEL52 samba]# ls -l smbpasswd
-rw----- 1 root root 110 Jan 21 06:19 smbpasswd
```

You can use a simple cat to see the contents of the **smbpasswd** database. The sambanobody user does have a password (it is secret).

```
[root@RHEL52 samba]# cat smbpasswd
```

```
sambanobody:503:AE9 ... 9DB309C528E540978:[U ]:LCT-4976B05B:
```

13.5. passdb backend

Note that recent versions of Samba have **tdbsam** as default for the **passdb backend** paramater.

```
root@ubull110:~# testparm -v 2>/dev/null| grep 'passdb backend'

passdb backend = tdbsam
```

13.6. forcing this user

Now that Samba knows about this user, we can adjust our writable share to force the ownership of files created through it. For this we use the **force user** and **force group** options. Now we can be sure that all files in the Samba writable share are owned by the same sambanobody user.

Below is the renewed definition of our share in smb.conf.

```
[pubwrite]
path = /srv/samba/writable
comment = files to write
force user = sambanobody
force group = sambanobody
read only = no
guest ok = yes
```

When you reconnect to the share and write a file, then this sambanobody user will own the newly created file (and nobody needs to know the password).

13.7. practice: first samba user account

1. Create a user account for use with samba.
2. Add this user to samba's user database.
3. Create a writable shared directory and use the "force user" and "force group" directives to force ownership of files.
4. Test the working of force user with smbclient, net use and Windows Explorer.

13.8. solution: first samba user account

1. Create a user account for use with samba.

```
useradd -s /bin/false smbguest
```

```
usermod -c 'samba guest'
```

```
passwd smbguest
```

2. Add this user to samba's user database.

```
smbpasswd -a smbguest
```

3. Create a writable shared directory and use the "force user" and "force group" directives to force ownership of files.

```
[userwrite]
path = /srv/samba/userwrite
comment = everyone writes files owned by smbguest
read only = no
guest ok = yes
force user = smbguest
force group = smbguest
```

4. Test the working of force user with smbclient, net use and Windows Explorer.

```
ls -l /srv/samba/userwrite (and verify ownership)
```

Chapter 14. samba authentication

14.1. creating the users on Linux

The goal of this example is to set up a file share accessible to a number of different users. The users will need to authenticate with their password before access to this share is granted. We will first create three randomly named users, each with their own password. First we add these users to Linux.

```
[root@RHEL52 ~]# useradd -c "Serena Williams" serena
[root@RHEL52 ~]# useradd -c "Justine Henin" justine
[root@RHEL52 ~]# useradd -c "Martina Hingis" martina
[root@RHEL52 ~]# passwd serena
Changing password for user serena.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@RHEL52 ~]# passwd justine
Changing password for user justine.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@RHEL52 ~]# passwd martina
Changing password for user martina.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

14.2. creating the users on samba

Then we add them to the **smbpasswd** file, with the same password.

```
[root@RHEL52 ~]# smbpasswd -a serena
New SMB password:
Retype new SMB password:
Added user serena.
[root@RHEL52 ~]# smbpasswd -a justine
New SMB password:
Retype new SMB password:
Added user justine.
[root@RHEL52 ~]# smbpasswd -a martina
New SMB password:
Retype new SMB password:
Added user martina.
```

14.3. security = user

Remember that we set samba's security mode to share with the **security = share** directive in the [global] section ? Since we now require users to always provide a userid and password for access to our samba server, we will need to change this. Setting **security = user** will require the client to provide samba with a valid userid and password before giving access to a share.

Our [global] section now looks like this.

```
[global]
workgroup = WORKGROUP
netbios name = TEACHER0
server string = Samba File Server
security = user
```

14.4. configuring the share

We add the following [share] section to our smb.conf (and we do not forget to create the directory /srv/samba/authwrite).

```
[authwrite]
path = /srv/samba/authwrite
comment = authenticated users only
read only = no
guest ok = no
```

14.5. testing access with net use

After restarting samba, we test with different users from within Microsoft computers. The screenshots use the **net use** First serena from Windows XP.

```
C:\>net use m: \\teacher0\authwrite stargate /user:serena
The command completed successfully.

C:\>m:

M:\>echo greetings from Serena > serena.txt
```

The next screenshot is martina on a Windows 2000 computer, she succeeds in writing her files, but fails to overwrite the file from serena.

```
C:\>net use k: \\teacher0\authwrite stargate /user:martina
The command completed successfully.

C:\>k:

K:\>echo greetings from martina > Martina.txt

K:\>echo test overwrite > serena.txt
Access is denied.
```

14.6. testing access with smbclient

You can also test connecting with authentication with **smbclient**. First we test with a wrong password.

```
[root@RHEL52 samba]# smbclient //teacher0/authwrite -U martina wrongpass
session setup failed: NT_STATUS_LOGON_FAILURE
```

Then we test with the correct password, and verify that we can access a file on the share.

```
[root@RHEL52 samba]# smbclient //teacher0/authwrite -U martina stargate
Domain=[TEACHER0] OS=[Unix] Server=[Samba 3.0.33-3.7.el5]
smb: \> more serena.txt
getting file \serena.txt of size 14 as /tmp/smbmore.QQfmSN (6.8 kb/s)
one
two
three
smb: \> q
```

14.7. verify ownership

We now have a simple standalone samba file server with authenticated access. And the files in the shares belong to their proper owners.

```
[root@RHEL52 samba]# ls -l /srv/samba/authwrite/
total 8
-rwxr--r-- 1 martina martina  0 Jan 21 20:06 martina.txt
-rwxr--r-- 1 serena  serena  14 Jan 21 20:06 serena.txt
-rwxr--r-- 1 serena  serena   6 Jan 21 20:09 ser.txt
```

14.8. common problems

14.8.1. NT_STATUS_BAD_NETWORK_NAME

You can get `NT_STATUS_BAD_NETWORK_NAME` when you forget to create the target directory.

```
[root@RHEL52 samba]# rm -rf /srv/samba/authwrite/
[root@RHEL52 samba]# smbclient //teacher0/authwrite -U martina stargate
Domain=[TEACHER0] OS=[Unix] Server=[Samba 3.0.33-3.7.el5]
tree connect failed: NT_STATUS_BAD_NETWORK_NAME
```

14.8.2. NT_STATUS_LOGON_FAILURE

You can get `NT_STATUS_LOGON_FAILURE` when you type the wrong password or when you type an unexisting username.

```
[root@RHEL52 samba]# smbclient //teacher0/authwrite -U martina STARGATE
session setup failed: NT_STATUS_LOGON_FAILURE
```

14.8.3. usernames are (not) case sensitive

Remember that usernames on Linux are case sensitive.

```
[root@RHEL52 samba]# su - MARTINA
su: user MARTINA does not exist
```

```
[root@RHEL52 samba]# su - martina
[martina@RHEL52 ~]$
```

But usernames on Microsoft computers are not case sensitive.

```
[root@RHEL52 samba]# smbclient //teacher0/authwrite -U martina stargate
Domain=[TEACHER0] OS=[Unix] Server=[Samba 3.0.33-3.7.el5]
smb: \> q
[root@RHEL52 samba]# smbclient //teacher0/authwrite -U MARTINA stargate
Domain=[TEACHER0] OS=[Unix] Server=[Samba 3.0.33-3.7.el5]
smb: \> q
```

14.9. practice : samba authentication

0. Make sure you have properly named backups of your smb.conf of the previous practices.
1. Create three users (on the Linux and on the samba), remember their passwords!
2. Set up a shared directory that is only accessible to authenticated users.
3. Use smbclient and a windows computer to access your share, use more than one user account (windows requires a logoff/logon for this).
4. Verify that files created by these users belong to them.
5. Try to change or delete a file from another user.

14.10. solution: samba authentication

1. Create three users (on the Linux and on the samba), remember their passwords!

```
useradd -c 'SMB user1' userx
```

```
passwd userx
```

2. Set up a shared directory that is only accessible to authenticated users.

The shared section in `smb.conf` could look like this:

```
[authwrite]
path = /srv/samba/authwrite
comment = authenticated users only
read only = no
guest ok = no
```

3. Use `smbclient` and a windows computer to access your share, use more than one user account (windows requires a logoff/logon for this).

```
on Linux: smbclient //studentX/authwrite -U user1 password
```

```
on windows net use p: \\studentX\authwrite password /user:user2
```

4. Verify that files created by these users belong to them.

```
ls -l /srv/samba/authwrite
```

5. Try to change or delete a file from another user.

you should not be able to change or overwrite files from others.

Chapter 15. samba securing shares

15.1. security based on user name

15.1.1. valid users

To restrict users per share, you can use the **valid users** parameter. In the example below, only the users listed as valid will be able to access the tennis share.

```
[tennis]
path = /srv/samba/tennis
comment = authenticated and valid users only
read only = No
guest ok = No
valid users = serena, kim, venus, justine
```

15.1.2. invalid users

If you are paranoia, you can also use **invalid users** to explicitly deny the listed users access. When a user is in both lists, the user has no access!

```
[tennis]
path = /srv/samba/tennis
read only = No
guest ok = No
valid users = kim, serena, venus, justine
invalid users = venus
```

15.1.3. read list

On a writable share, you can set a list of read only users with the **read list** parameter.

```
[football]
path = /srv/samba/football
read only = No
guest ok = No
read list = martina, roberto
```

15.1.4. write list

Even on a read only share, you can set a list of users that can write. Use the **write list** parameter.

```
[football]
path = /srv/samba/golf
read only = Yes
guest ok = No
write list = eddy, jan
```

15.2. security based on ip-address

15.2.1. hosts allow

The **hosts allow** or **allow hosts** parameter is one of the key advantages of Samba. It allows access control of shares on the ip-address level. To allow only specific hosts to access a share, list the hosts, separated by comma's.


```
allow hosts = 192.168.1.5, 192.168.1.40
```

Allowing entire subnets is done by ending the range with a dot.

```
allow hosts = 192.168.1.
```

Subnet masks can be added in the classical way.

```
allow hosts = 10.0.0.0/255.0.0.0
```

You can also allow an entire subnet with exceptions.

```
hosts allow = 10. except 10.0.0.12
```

15.2.2. hosts deny

The **hosts deny** or **deny hosts** parameter is the logical counterpart of the previous. The syntax is the same as for hosts allow.

```
hosts deny = 192.168.1.55, 192.168.1.56
```

15.3. security through obscurity

15.3.1. hide unreadable

Setting **hide unreadable** to yes will prevent users from seeing files that cannot be read by them.

```
hide unreadable = yes
```

15.3.2. browsable

Setting the **browseable = no** directive will hide shares from My Network Places. But it will not prevent someone from accessing the share (when the name of the share is known).

Note that **browsable** and **browseable** are both correct syntax.

```
[pubread]
path = /srv/samba/readonly
comment = files to read
read only = yes
guest ok = yes
browseable = no
```

15.4. file system security

15.4.1. create mask

You can use **create mask** and **directory mask** to set the maximum allowed permissions for newly created files and directories. The mask you set is an AND mask (it takes permissions away).

```
[tennis]
path = /srv/samba/tennis
read only = No
```

```
guest ok = No
create mask = 640
directory mask = 750
```

15.4.2. force create mode

Similar to **create mask**, but different. Where the mask from above was a logical AND, the mode you set here is a logical OR (so it adds permissions). You can use the **force create mode** and **force directory mode** to set the minimal required permissions for newly created files and directories.

```
[tennis]
path = /srv/samba/tennis
read only = No
guest ok = No
force create mode = 444
force directory mode = 550
```

15.4.3. security mask

The **security mask** and **directory security mask** work in the same way as **create mask** and **directory mask**, but apply only when a windows user is changing permissions using the windows security dialog box.

15.4.4. force security mode

The **force security mode** and **force directory security mode** work in the same way as **force create mode** and **force directory mode**, but apply only when a windows user is changing permissions using the windows security dialog box.

15.4.5. inherit permissions

With **inherit permissions = yes** you can force newly created files and directories to inherit permissions from their parent directory, overriding the create mask and directory mask settings.

```
[authwrite]
path = /srv/samba/authwrite
comment = authenticated users only
read only = no
guest ok = no
create mask = 600
directory mask = 555
inherit permissions = yes
```

15.5. practice: securing shares

1. Create a writable share called sales, and a readonly share called budget. Test that it works.
2. Limit access to the sales share to ann, sandra and veronique.
3. Make sure that roberto cannot access the sales share.
4. Even though the sales share is writable, ann should only have read access.
5. Even though the budget share is read only, sandra should also have write access.
6. Limit one shared directory to the 192.168.1.0/24 subnet, and another share to the two computers with ip-addresses 192.168.1.33 and 172.17.18.19.
7. Make sure the computer with ip 192.168.1.203 cannot access the budget share.
8. Make sure (on the budget share) that users can see only files and directories to which they have access.
9. Make sure the sales share is not visible when browsing the network.
10. All files created in the sales share should have 640 permissions or less.
11. All directories created in the budget share should have 750 permissions or more.
12. Permissions for files on the sales share should never be set more than 664.
13. Permissions for files on the budget share should never be set less than 500.
14. If time permits (or if you are waiting for other students to finish this practice), then combine the "read only" and "writable" statements to check which one has priority.
15. If time permits then combine "read list", "write list", "hosts allow" and "hosts deny". Which of these has priority ?

15.6. solution: securing shares

1. Create a writable share called sales, and a readonly share called budget. Test that it works.

see previous solutions on how to do this...

2. Limit access to the sales share to ann, sandra and veronique.

```
valid users = ann, sandra, veronique
```

3. Make sure that roberto cannot access the sales share.

```
invalid users = roberto
```

4. Even though the sales share is writable, ann should only have read access.

```
read list = ann
```

5. Even though the budget share is read only, sandra should also have write access.

```
write list = sandra
```

6. Limit one shared directory to the 192.168.1.0/24 subnet, and another share to the two computers with ip-addresses 192.168.1.33 and 172.17.18.19.

```
hosts allow = 192.168.1.
```

```
hosts allow = 192.168.1.33, 172.17.18.19
```

7. Make sure the computer with ip 192.168.1.203 cannot access the budget share.

```
hosts deny = 192.168.1.203
```

8. Make sure (on the budget share) that users can see only files and directories to which they have access.

```
hide unreadable = yes
```

9. Make sure the sales share is not visible when browsing the network.

```
browsable = no
```

10. All files created in the sales share should have 640 permissions or less.

```
create mask = 640
```

11. All directories created in the budget share should have 750 permissions or more.

```
force directory mode = 750
```

12. Permissions for files on the sales share should never be set more than 664.

```
security mask = 750
```

13. Permissions for files on the budget share should never be set less than 500.

```
force security directory mask = 500
```

14. If time permits (or if you are waiting for other students to finish this practice), then combine the "read only" and "writable" statements to check which one has priority.

15. If time permits then combine "read list", "write list", "hosts allow" and "hosts deny". Which of these has priority ?

Chapter 16. samba domain member

16.1. changes in smb.conf

16.1.1. workgroup

The **workgroup** option in the global section should match the netbios name of the Active Directory domain.

```
workgroup = STARGATE
```

16.1.2. security mode

Authentication will not be handled by samba now, but by the Active Directory domain controllers, so we set the **security** option to domain.

```
security = Domain
```

16.1.3. Linux uid's

Linux requires a user account for every user accessing its file system, we need to provide Samba with a range of uid's and gid's that it can use to create these user accounts. The range is determined with the **idmap uid** and the **idmap gid** parameters. The first Active Directory user to connect will receive Linux uid 20000.

```
idmap uid = 20000-22000  
idmap gid = 20000-22000
```

16.1.4. winbind use default domain

The **winbind use default domain** parameter makes sure winbind also operates on users without a domain component in their name.

```
winbind use default domain = yes
```

16.1.5. [global] section in smb.conf

Below is our new global section in **smb.conf**.

```
[global]  
workgroup = STARGATE  
security = Domain  
server string = Stargate Domain Member Server  
idmap uid = 20000-22000  
idmap gid = 20000-22000  
winbind use default domain = yes
```

16.1.6. realm in /etc/krb5.conf

To connect to a Windows 2003 sp2 (or later) you will need to adjust the kerberos realm in **/etc/krb5.conf** and set both lookup statements to true.

```
[libdefaults]
default_realm = STARGATE.LOCAL
dns_lookup_realm = true
dns_lookup_kdc = true
```

16.1.7. [share] section in smb.conf

Nothing special is required for the share section in smb.conf. Remember that we do not manually create users in smbpasswd or on the Linux (/etc/passwd). Only Active Directory users are allowed access.

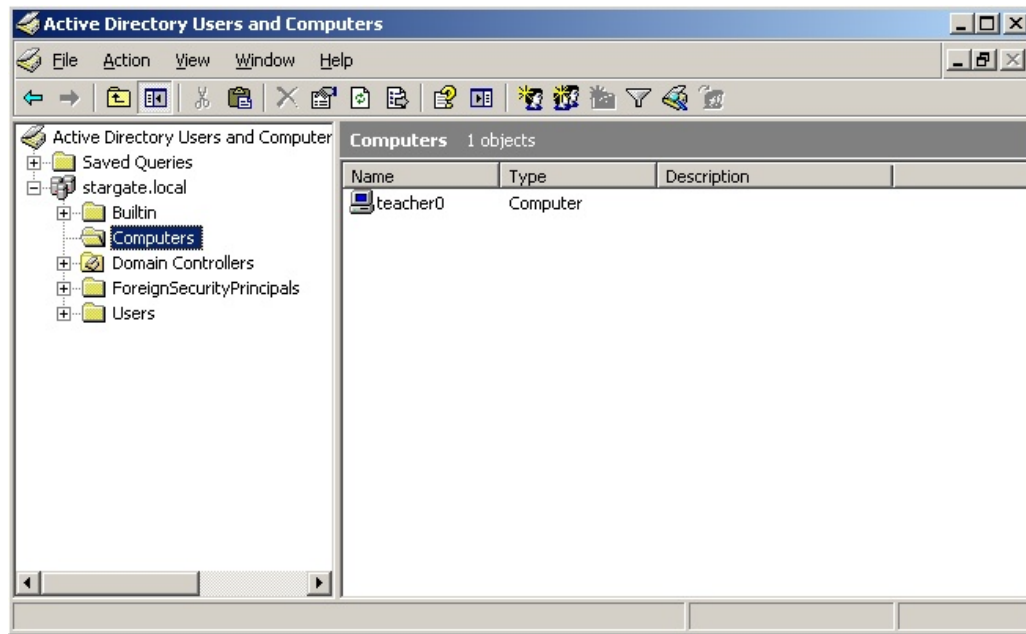
```
[domaaindata]
path = /srv/samba/domaaindata
comment = Active Directory users only
read only = No
```

16.2. joining an Active Directory domain

While the Samba server is stopped, you can use **net rpc join** to join the Active Directory domain.

```
[root@RHEL52 samba]# service smb stop
Shutting down SMB services:          [ OK ]
Shutting down NMB services:         [ OK ]
[root@RHEL52 samba]# net rpc join -U Administrator
Password:
Joined domain STARGATE.
```

We can verify in the aduc (Active Directory Users and Computers) that a computer account is created for this samba server.



16.3. winbind

16.3.1. adding winbind to nsswitch.conf

The **winbind daemon** is talking with the Active Directory domain.

We need to update the **/etc/nsswitch.conf** file now, so user group and host names can be resolved against the winbind daemon.

```
[root@RHEL52 samba]# vi /etc/nsswitch.conf
[root@RHEL52 samba]# grep winbind /etc/nsswitch.conf
passwd:      files winbind
group:       files winbind
hosts:       files dns winbind
```

16.3.2. starting samba and winbindd

Time to start Samba followed by **winbindd**.

```
[root@RHEL4b samba]# service smb start
Starting SMB services: [ OK ]
Starting NMB services: [ OK ]
[root@RHEL4b samba]# service winbind start
Starting winbindd services: [ OK ]
[root@RHEL4b samba]#
```

16.4. wbinfo

16.4.1. verify the trust

You can use **wbinfo -t** to verify the trust between your samba server and Active Directory.

```
[root@RHEL52 ~]# wbinfo -t
checking the trust secret via RPC calls succeeded
```

16.4.2. list all users

We can obtain a list of all user with the **wbinfo -u** command. The domain is not shown when the **winbind use default domain** parameter is set.

```
[root@RHEL52 ~]# wbinfo -u
TEACHER0\serena
TEACHER0\justine
TEACHER0\martina
STARGATE\administrator
STARGATE\guest
STARGATE\support_388945a0
STARGATE\pol
STARGATE\krbtgt
STARGATE\arthur
STARGATE\harry
```

16.4.3. list all groups

We can obtain a list of all domain groups with the **wbinfo -g** command. The domain is not shown when the **winbind use default domain** parameter is set.

```
[root@RHEL52 ~]# wbinfo -g
BUILTIN\administrators
BUILTIN\users
BATMAN\domain computers
BATMAN\domain controllers
BATMAN\schema admins
BATMAN\enterprise admins
BATMAN\domain admins
BATMAN\domain users
BATMAN\domain guests
BATMAN\group policy creator owners
BATMAN\dnsupdateproxy
```

16.4.4. query a user

We can use **wbinfo -a** to verify authentication of a user against Active Directory. Assuming a user account **harry** with password **stargate** is just created on the Active Directory, we get the following screenshot.

```
[root@RHEL52 ~]# wbinfo -a harry%stargate
plaintext password authentication succeeded
challenge/response password authentication succeeded
```

16.5. getent

We can use **getent** to verify that winbindd is working and actually adding the Active directory users to **/etc/passwd**.

```
[root@RHEL52 ~]# getent passwd harry
harry:*:20000:20008:harry potter:/home/BATMAN/harry:/bin/false
[root@RHEL52 ~]# getent passwd arthur
arthur:*:20001:20008:arthur dent:/home/BATMAN/arthur:/bin/false
[root@RHEL52 ~]# getent passwd bilbo
bilbo:*:20002:20008:bilbo baggins:/home/BATMAN/bilbo:/bin/false
```

If the user already exists locally, then the local user account is shown. This is because winbind is configured in **/etc/nsswitch.conf** after **files**.

```
[root@RHEL52 ~]# getent passwd paul
paul:x:500:500:Paul Cobbaut:/home/paul:/bin/bash
```

All the Active Directory users can now easily connect to the Samba share. Files created by them, belong to them.

16.6. file ownership

```
[root@RHEL4b samba]# ll /srv/samba/domaindata/
total 0
-rwxr--r-- 1 justine 20000 0 Jun 22 19:54 create_by_justine_on_winxp.txt
-rwxr--r-- 1 venus   20000 0 Jun 22 19:55 create_by_venus.txt
-rwxr--r-- 1 maria   20000 0 Jun 22 19:57 Maria.txt
```

16.7. practice : samba domain member

1. Verify that you have a working Active Directory (AD) domain.
2. Add the domain name and domain controller to `/etc/hosts`. Set the AD-DNS in `/etc/resolv.conf`.
3. Setup Samba as a member server in the domain.
4. Verify the creation of a computer account in AD for your Samba server.
5. Verify the automatic creation of AD users in `/etc/passwd` with `wbinfo` and `getent`.
6. Connect to Samba shares with AD users, and verify ownership of their files.

Chapter 17. samba domain controller

17.1. about Domain Controllers

17.1.1. Windows NT4

Windows NT4 works with single master replication domain controllers. There is exactly one PDC (Primary Domain Controller) in the domain, and zero or more BDC's (Backup Domain Controllers). Samba 3 has all features found in Windows NT4 PDC and BDC, and more. This includes file and print serving, domain control with single logon, logon scripts, home directories and roaming profiles.

17.1.2. Windows 200x

With Windows 2000 came Active Directory. AD includes multimaster replication and group policies. Samba 3 can only be a member server in Active Directory, it cannot manage group policies. Samba 4 can do this (in beta).

17.1.3. Samba 3

Samba 3 can act as a domain controller in its own domain. In a Windows NT4 domain, with one Windows NT4 PDC and zero or more BDC's, Samba 3 can only be a member server. The same is valid for Samba 3 in an Active Directory Domain. In short, a Samba 3 domain controller can not share domain control with Windows domain controllers.

17.1.4. Samba 4

Samba 4 can be a domain controller in an Active Directory domain, including managing group policies. As of this writing, Samba 4 is not released for production!

17.2. About security modes

17.2.1. security = share

The 'Windows for Workgroups' way of working, a client requests connection to a share and provides a password for that connection. Anyone who knows a password for a share can access that share. This security model was common in Windows 3.11, Windows 95, Windows 98 and Windows ME.

17.2.2. security = user

The client will send a userid + password before the server knows which share the client wants to access. This mode should be used whenever the samba server is in control of the user database. Both for standalone and samba domain controllers.

17.2.3. security = domain

This mode will allow samba to verify user credentials using NTLM in Windows NT4 and in all Active Directory domains. This is similar to Windows NT4 BDC's joining a native Windows 2000/3 Active Directory domain.

17.2.4. security = ads

This mode will make samba use Kerberos to connect to the Active Directory domain.

17.2.5. security = server

This mode is obsolete, it can be used to forward authentication to another server.

17.3. About password backends

The previous chapters all used the **smbpasswd** user database. For domain control we opt for the **tdbsam** password backend. Another option would be to use LDAP. Larger domains will benefit from using LDAP instead of the not so scalable **tdbsam**. When you need more than one Domain Controller, then the Samba team advises to not use **tdbsam**.

17.4. [global] section in smb.conf

Now is a good time to start adding comments in your **smb.conf**. First we will take a look at the naming of our domain and server in the **[global]** section, and at the domain controlling parameters.

17.4.1. security

The security must be set to **user** (which is the default). This mode will make samba control the user accounts, so it will allow samba to act as a domain controller.

```
security = user
```

17.4.2. os level

A samba server is the most stable computer in the network, so it should win all browser elections (**os level** above 32) to become the **browser master**

```
os level = 33
```

17.4.3. passdb backend

The **passdb backend** parameter will determine whether samba uses **smbpasswd**, **tdbsam** or **ldap**.

```
passdb backend = tdbsam
```

17.4.4. preferred master

Setting the **preferred master** parameter to **yes** will make the **nmbd** daemon force an election on startup.

```
preferred master = yes
```

17.4.5. domain logons

Setting the **domain logons** parameter will make this samba server a domain controller.

```
domain logons = yes
```

17.4.6. domain master

Setting the **domain master** parameter can cause samba to claim the **domain master browser** role for its workgroup. Don't use this parameter in a workgroup with an active NT4 PDC.

```
domain master = yes
```

17.4.7. [global] section

The screenshot below shows a sample [global] section for a samba domain controller.

```
[global]
# names
workgroup = SPORTS
netbios name = DCSPORTS
server string = Sports Domain Controller
# domain control parameters
security = user
os level = 33
preferred master = Yes
domain master = Yes
domain logons = Yes
```

17.5. netlogon share

Part of the microsoft definition for a domain controller is that it should have a **netlogon share**. This is the relevant part of smb.conf to create this netlogon share on Samba.

```
[netlogon]
comment = Network Logon Service
path = /srv/samba/netlogon
admin users = root
guest ok = Yes
browseable = No
```

17.6. other [share] sections

We create some sections for file shares, to test the samba server. Users can all access the general sports file share, but only group members can access their own sports share.

```
[sports]
comment = Information about all sports
path = /srv/samba/sports
valid users = @ntsports
read only = No

[tennis]
comment = Information about tennis
path = /srv/samba/tennis
valid users = @nttennis
read only = No
```



```
[football]
comment = Information about football
path = /srv/samba/football
valid users = @ntfootball
read only = No
```

17.7. Users and Groups

To be able to use users and groups in the samba domain controller, we can first set up some groups on the Linux computer.

```
[root@RHEL52 samba]# groupadd ntadmins
[root@RHEL52 samba]# groupadd ntsports
[root@RHEL52 samba]# groupadd ntfootball
[root@RHEL52 samba]# groupadd nttennis
```

This enables us to add group membership info to some new users for our samba domain. Don't forget to give them a password.

```
[root@RHEL52 samba]# useradd -m -G ntadmins Administrator
[root@RHEL52 samba]# useradd -m -G ntsports,nttennis venus
[root@RHEL52 samba]# useradd -m -G ntsports,nttennis kim
[root@RHEL52 samba]# useradd -m -G ntsports,nttennis jelena
[root@RHEL52 samba]# useradd -m -G ntsports,ntfootball figo
[root@RHEL52 samba]# useradd -m -G ntsports,ntfootball ronaldo
[root@RHEL52 samba]# useradd -m -G ntsports,ntfootball pfaff
```

It is always safe to verify creation of users, groups and passwords in `/etc/passwd`, `/etc/shadow` and `/etc/group`.

```
[root@RHEL52 samba]# tail -11 /etc/group
ntadmins:x:507:Administrator
ntsports:x:508:venus,kim,jelena,figo,ronaldo,pfaff
ntfootball:x:509:figo,ronaldo,pfaff
nttennis:x:510:venus,kim,jelena
Administrator:x:511:
venus:x:512:
kim:x:513:
jelena:x:514:
figo:x:515:
ronaldo:x:516:
pfaff:x:517:
```

17.8. tdbsam

Next we must make these users known to samba with the `smbpasswd` tool. When you add the first user to **tdbsam**, the file `/etc/samba/passdb.tdb` will be created.

```
[root@RHEL52 samba]# smbpasswd -a root
New SMB password:
```

```
Retype new SMB password:
tdbsam_open: Converting version 0 database to version 3.
Added user root.
```

Adding all the other users generates less output, because tdbsam is already created.

```
[root@RHEL4b samba]# smbpasswd -a root
New SMB password:
Retype new SMB password:
Added user root.
```

17.9. about computer accounts

Every NT computer (Windows NT, 2000, XP, Vista) can become a member of a domain. Joining the domain (by right-clicking on My Computer) means that a computer account will be created in the domain. This computer account also has a password (but you cannot know it) to prevent other computers with the same name from accidentally becoming member of the domain. The computer account created by Samba is visible in the `/etc/passwd` file on Linux. Computer accounts appear as a normal user account, but end their name with a dollar sign. Below a screenshot of the windows 2003 computer account, created by Samba 3.

```
[root@RHEL52 samba]# tail -5 /etc/passwd
jelena:x:510:514::/home/jelena:/bin/bash
figo:x:511:515::/home/figo:/bin/bash
ronaldo:x:512:516::/home/ronaldo:/bin/bash
pfaff:x:513:517::/home/pfaff:/bin/bash
w2003ee$:x:514:518::/home/nobody:/bin/false
```

To be able to create the account, you will need to provide credentials of an account with the permission to create accounts (by default only root can do this on Linux). And we will have to tell Samba how to do this, by adding an **add machine script** to the global section of `smb.conf`.

```
add machine script = /usr/sbin/useradd -s /bin/false -d /home/nobody %u
```

You can now join a Microsoft computer to the sports domain (with the root user). After reboot of the Microsoft computer, you will be able to logon with Administrator (password Stargate1), but you will get an error about your roaming profile. We will fix this in the next section.

When joining the samba domain, you have to enter the credentials of a Linux account that can create users (usually only root can do this). If the Microsoft computer complains with **The parameter is incorrect**, then you possibly forgot to add the **add machine script**.

17.10. local or roaming profiles

For your information, if you want to force local profiles instead of roaming profiles, then simply add the following two lines to the global section in `smb.conf`.

```
logon home =  
logon path =
```

Microsoft computers store a lot of User Metadata and application data in a user profile. Making this profile available on the network will enable users to keep their Desktop and Application settings across computers. User profiles on the network are called **roaming profiles** or **roving profiles**. The Samba domain controller can manage these profiles. First we need to add the relevant section in smb.conf.

```
[Profiles]  
comment = User Profiles  
path = /srv/samba/profiles  
readonly = No  
profile acls = Yes
```

Besides the share section, we also need to set the location of the profiles share (this can be another Samba server) in the global section.

```
logon path = \\%L\Profiles\%U
```

The **%L** variable is the name of this Samba server, the **%U** variable translates to the username. After adding a user to smbpasswd and letting the user log on and off, the profile of the user will look like this.

```
[root@RHEL4b samba]# ll /srv/samba/profiles/Venus/  
total 568  
drwxr-xr-x  4 Venus Venus   4096 Jul  5 10:03 Application Data  
drwxr-xr-x  2 Venus Venus   4096 Jul  5 10:03 Cookies  
drwxr-xr-x  3 Venus Venus   4096 Jul  5 10:03 Desktop  
drwxr-xr-x  3 Venus Venus   4096 Jul  5 10:03 Favorites  
drwxr-xr-x  4 Venus Venus   4096 Jul  5 10:03 My Documents  
drwxr-xr-x  2 Venus Venus   4096 Jul  5 10:03 NetHood  
-rwxr--r--  1 Venus Venus 524288 Jul  5  2007 NTUSER.DAT  
-rwxr--r--  1 Venus Venus  1024 Jul  5  2007 NTUSER.DAT.LOG  
-rw-r--r--  1 Venus Venus   268 Jul  5 10:03 ntuser.ini  
drwxr-xr-x  2 Venus Venus   4096 Jul  5 10:03 PrintHood  
drwxr-xr-x  2 Venus Venus   4096 Jul  5 10:03 Recent  
drwxr-xr-x  2 Venus Venus   4096 Jul  5 10:03 SendTo  
drwxr-xr-x  3 Venus Venus   4096 Jul  5 10:03 Start Menu  
drwxr-xr-x  2 Venus Venus   4096 Jul  5 10:03 Templates
```

17.11. Groups in NTFS acls

We have users on Unix, we have groups on Unix that contain those users.

```
[root@RHEL4b samba]# grep nt /etc/group  
...  
ntadmins:x:506:Administrator  
ntsports:x:507:Venus,Serena,Kim,Figo,Pfaff  
nttennis:x:508:Venus,Serena,Kim  
ntfootball:x:509:Figo,Pfaff
```

```
[root@RHEL4b samba]#
```

We already added Venus to the **tdbsam** with **smbpasswd**.

```
smbpasswd -a Venus
```

Does this mean that Venus can access the tennis and the sports shares ? Yes, all access works fine on the Samba server. But the **nttennis** group is not available on the windows machines. To make the groups available on windows (like in the ntfs security tab of files and folders), we have to map unix groups to windows groups. To do this, we use the **net groupmap** command.

```
[root@RHEL4b samba]# net groupmap add ntgroup="tennis" unixgroup=nttennis type=d
No rid or sid specified, choosing algorithmic mapping
Successfully added group tennis to the mapping db
[root@RHEL4b samba]# net groupmap add ntgroup="football" unixgroup=ntfootball type=d
No rid or sid specified, choosing algorithmic mapping
Successfully added group football to the mapping db
[root@RHEL4b samba]# net groupmap add ntgroup="sports" unixgroup=ntsports type=d
No rid or sid specified, choosing algorithmic mapping
Successfully added group sports to the mapping db
[root@RHEL4b samba]#
```

Now you can use the Samba groups on all NTFS volumes on members of the domain.

17.12. logon scripts

Before testing a logon script, make sure it has the proper carriage returns that DOS files have.

```
[root@RHEL4b netlogon]# cat start.bat
net use Z: \\DCSPORTS0\SPORTS
[root@RHEL4b netlogon]# unix2dos start.bat
unix2dos: converting file start.bat to DOS format ...
[root@RHEL4b netlogon]#
```

Then copy the scripts to the netlogon share, and add the following parameter to **smb.conf**.

```
logon script = start.bat
```

17.13. practice: samba domain controller

1. Setup Samba as a domain controller.
2. Create the shares salesdata, salespresentations and meetings. Salesdata must be accessible to all sales people and to all managers. SalesPresentations is only for all sales people. Meetings is only accessible to all managers. Use groups to accomplish this.
3. Join a Microsoft computer to your domain. Verify the creation of a computer account in /etc/passwd.
4. Setup and verify the proper working of roaming profiles.
5. Find information about home directories for users, set them up and verify that users receive their home directory mapped under the H:-drive in MS Windows Explorer.
6. Use a couple of samba domain groups with members to set acls on ntfs. Verify that it works!
7. Knowing that the %m variable contains the computername, create a separate log file for every computer(account).
8. Knowing that %s contains the client operating system, include a smb.%s.conf file that contains a share. (The share will only be visible to clients with that OS).
9. If time permits (or if you are waiting for other students to finish this practice), then combine "valid users" and "invalid users" with groups and usernames with "hosts allow" and "hosts deny" and make a table of which get priority over which.

Chapter 18. a brief look at samba 4

18.1. Samba 4 alpha 6

A quick view on Samba 4 alpha 6 (January 2009). You can also follow this guide <http://wiki.samba.org/index.php/Samba4/HOWTO>

Remove old Samba from Red Hat

```
yum remove samba
```

set a fix ip address (Red Hat has an easy GUI)

download and untar

```
samba.org, click 'download info', choose mirror, dl samba4 latest alpha
```

once untarred, enter the directory and read the howto4.txt

```
cd samba-4.0.0alpha6/
```

```
more howto4.txt
```

first we have to configure, compile and install samba4

```
cd source4/
```

```
./configure
```

```
make
```

```
make install
```

Then we can use the provision script to setup our realm. I used booi.schot as domain name (instead of example.com).

```
./setup/provision --realm=BOOI.SCHOT --domain=BOOI --adminpass=stargate \
--server-role='domain controller'
```

i added a simple share for testing

```
vi /usr/local/samba/etc/smb.conf
```

then i started samba

```
cd /usr/local/samba/sbin/
```

```
./samba
```

I tested with smbclient, it works

```
smbclient //localhost/test -Uadministrator%stargate
```

I checked that bind (and bind-chroot) were installed (yes), so copied the srv records

```
cp booi.schot.zone /var/named/chroot/etc/
```

then appended to named.conf

```
cat named.conf >> /var/named/chroot/etc/named.conf
```


I followed these steps in the howto4.txt

```
vi /etc/init.d/named [added two export lines right after start()]
chmod a+r /usr/local/samba/private/dns.keytab
cp krb5.conf /etc/
vi /var/named/chroot/etc/named.conf
--> remove a lot, but keep allow-update { any; };
```

restart bind (named!), then tested dns with dig, this works (stripped screenshot!)

```
[root@RHEL52 private]# dig _ldap._tcp.dc._msdcs.booi.schot SRV @localhost

; (1 server found)
;; global options: printcmd
;; Got answer:
;; -HEADER- opcode: QUERY, status: NXDOMAIN, id: 58186
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;_ldap._tcp.dc._msdcs.booi.schot. IN SRV

;; AUTHORITY SECTION:
.      10800 IN SOA A.ROOT-SERVERS.NET....

;; Query time: 54 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue Jan 27 20:57:05 2009
;; MSG SIZE rcvd: 124

[root@RHEL52 private]#
```

made sure /etc/resolv.conf points to himself

```
[root@RHEL52 private]# cat /etc/resolv.conf
search booi.schot
nameserver 127.0.0.1
```

start windows 2003 server, enter the samba4 as DNS!

ping the domain, if it doesn't work, then add your redhats hostname and your realm to windows/system32/drivers/etc/hosts

join the windows computer to the domain

reboot the windows

log on with administrator stargate

start run dsa.msc to manage samba4

create an OU, a user and a GPO, test that it works