Building a SAMBA Server using arch linux

René Rasmussen

18. september 2007

1 Disclaimer

Although I have made the best effort to make sure this guide works, I cannot guarantee that it will not break your system. Now you are warned!
If you run into a problem I will of course try and help you resolve it.
If you find any errors in this guide I would appreciate it if you let me know about it. The easiest way to contact me is through the contact page on my website. The address is at the bottom of the page. Alternatively you can contact me through the arch linux forum[1]. I'm registered under the nickname **madeye**.

2 Introduction

Making a file server isn't that hard. The hard part comes when you need to make and keep it secure against malicious attacks. I will show you how I try to do this. But first we must make a plan for what we want to do.

- What do you want to share. => what is the layout of your data and partitions.
- Who need to have access. => What users, computers and IP addresses.
- What access is needed. => Read only or Read/Write access.

The easiest way for me to show you how to install a samba server, is by guiding you through an example installation. I'm assuming you have performed an initial install of arch linux and that you have it up and running. As we are installing a server you should use a static IP address. Throughout this setup I will use the ip-address **192.168.100.3**.

If you need help in installing arch, I suggest you take a look at the official install guide[3] or the beginners guide[4]. Both can be found on the arch linux homepage. You should also be so familiar with the commandline that you can perform installation and configurations on a linux system.

In addition to installing the samba server we will also install a ssh server (making it easy to manage our server from our workstation)and an iptables based firewall to help secure it.

3 The Plan

- Make a new user
- Install the OpenSSH server and configure it.
- Install iptables and make initial configuration.
- Install the samba server.
- Make a shared folder with read only access.
- Make a shared folder with read/write access.
- Make a shared folder based on the username.
- Configure the samba server
- Change configuration of iptables if necesarry.

[1] http://bbs.archlinux.org
[2] http://www.gnu.org/copyleft/fdl.html
[3] http://wiki.archlinux.org/index.php/Official_Arch_Linux_Install_Guide
[4] http://wiki.archlinux.org/index.php/Beginners_Guide

4 Make a day to day user

For the day to day business it's not a good idea to run as the root user, so we'll make a user who will have the ability to su to root. This user is also necesary as we will not allow root to login through the ssh server. To give a user the ability to su to root, he needs to be a member of the **wheel** group.
At the command prompt issue the following command :

[**root@server**]$ adduser

You will be asked some questions about the new user. Here is a record of what I used.

```
Login name for new user [ ] : john
User ID ('UID') [ defaults to next available ] : <ENTER>
Initial group [ users ] : <ENTER>
Additional groups (comma seperated) [ ] : wheel,optical,floppy
Home directory [ /home/john ] <ENTER>
Shell [ /bin/bash ] <ENTER>
Expiry date (YYYY-MM-DD) [ ] : <ENTER>

New account will be created as follows :
----------------
Login name....... : john
UID.............. : [ Next available ]
Initial group.... : users
Additional groups : wheel,optical,floppy
Home directory... : /home/john
Shell............ : /bin/bash
Expiry date...... : [ never ]

This is it... if you want to bail out, hit Control-C. Otherwise ,
press ENTER to go ahead and make the account.


Creating new account...


Changing the user information for john
Enter new value, or press ENTER for the default
  Full name [ ] : <ENTER>
  Room number [ ] : <ENTER>
  Work phone [ ] : <ENTER>
  Home phone [ ] : <ENTER>
  Other [ ] : <ENTER>
Enter new UNIX password : <your password here>
Retype new UNIX password : <password again>
passwd: password updated succesfully


Account setup complete.
```

That should take care of setting up our user.

5 OpenSSH server

5.1 Install

To install OpenSSH use the following command :

```
[root@server]$ pacman -S openssh
```

5.2 Configure

There isn't much that needs to be configured to make the ssh server work.

1. Edit the file */etc/hosts.allow*, and add the hosts you want to give access to. You can see in the following table what choices you have.

Command	Description
sshd: ALL	Everyone can connect
sshd: 192.168.100.5	Only this IP address can connect
sshd: 192.168.100.0/255.255.255.0	Only this IP address network can connect

   For starters I will allow access to the IP address network. So we'll add the line

```
sshd: 192.168.100.0/255.255.255.0
```

   to */etc/hosts.allow*

2. Next we'll tighten the security a little on the ssh daemon. To do this we'll make some changes to the file */etc/ssh/sshd_ config*, so fire up your favorite editor and make the following changes.

```
# only accept ssh 2 connections
Protocol 2

# only accept connections on specified IP address
ListenAddress 192.168.100.3

# Automatically close connection if no login has occured within 120 seconds
LoginGraceTime 120

# Deny root to login remotely
PermitRootLogin no

# Check if client is still connected.
# An encrypted message will be send to the client after 120 seconds of inactivity,
# requesting a return message.
ClientAliveInterval 120

# If no reply is received from the client 5 times, the connection will be terminated.
ClientAliveCountMax 5
```

3. Finally we need to make sure the daemon starts when we boot the computer, so add **sshd** to the daemons array in the file */etc/rc.conf*

5.3 Start the server

Now start the server with :

```
[root@server]$ /etc/rc.d/sshd start
```

5.4 Our first SSH remote login

Okay, it's now time to tryout the remote login facility. Fire up a terminal on your workstation and type the following command :

[**john@workstation**]$ ssh 192.168.100.3 −l john

You will be greeted with a message like this one.

```
The authenticity of host '192.168.100.3 (192.168.100.3)' can't be established.
RSA key fingerprint is d5:aa:39:1a:a5:29:72:26:b5:0e:23:5a:e2:88:14:a9.
Are you sure you want to continue connecting (yes/no)?
```

After you answer **yes** to the question, the following will be displayed.(Enter the password when requested).

```
Warning: Permanently added '192.168.100.3' (RSA) to the list of known hosts.
john@192.168.100.3's password:
Last login: Sun Feb 18 16:19:41 2007 from 192.168.100.31
```

Congratulations!! You have succesfully connected to your server.
The SSH session can be closed with **exit**.
The rest of the setup can now be done through the ssh connection if you wish.

6 iptables

6.1 Install

To install iptables use the following command : (remember to su to root!)

[**root@server**]$ pacman −S iptables

6.2 Configure

First of we disable IP forwarding as we don't need it in our server.
To do this edit the file */etc/conf.d/iptables* and change the file to read like this :

```
# Configuration for iptables rules

IPTABLES=/usr/sbin/iptables
IP6TABLES=/usr/sbin/ip6tables

IPTABLES_CONF=/etc/iptables/iptables.rules
IP6TABLES_CONF=/etc/iptables/ip6tables.rules
IPTABLES_FORWARD=0 # disable IP forwarding?
```

Next I have a script for setting up the firewall for securing our server. It's heavily based on a description I found on Linux.org[5]
I used the command **scp** to copy it to the server using ssh. Here is the exact command :

[**john@workstation**]$ scp /home/john/ip.sh john@192.168.100.3:/home/john

You'll need to give it the password when it asks for it. The file should now be on the server.

Here is the contents of the script.

```
#!/bin/sh

IPTABLES=/usr/sbin/iptables
LAN=192.168.100.0/24
```

[5] http://www.linux.org/lessons/advanced/x313.html

```
OWNIP=192.168.100.3

# start by flushing the rules
$IPTABLES -F

# Delete all chains
$IPTABLES -X

# allow packets coming from the machine
$IPTABLES -A INPUT -i lo -j ACCEPT
$IPTABLES -A OUTPUT -o lo -j ACCEPT

# allow outgoing traffic
$IPTABLES -A OUTPUT -o eth0 -j ACCEPT

# allow established and related connections
$IPTABLES -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

# block spoofing
$IPTABLES -A INPUT -s 127.0.0.0/8 -i ! lo -j DROP
$IPTABLES -A INPUT -s $OWNIP -j DROP

# stop bad packets
$IPTABLES -A INPUT -m state --state INVALID -j DROP

# NMAP FIN/URG/PSH
$IPTABLES -A INPUT -i eth0 -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP
# stop Xmas Tree type scanning
$IPTABLES -A INPUT -i eth0 -p tcp --tcp-flags ALL ALL -j DROP
$IPTABLES -A INPUT -i eth0 -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP
# stop null scanning
$IPTABLES -A INPUT -i eth0 -p tcp --tcp-flags ALL NONE -j DROP
# SYN/RST
$IPTABLES -A INPUT -i eth0 -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
# SYN/FIN
$IPTABLES -A INPUT -i eth0 -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP
# stop sync flood
$IPTABLES -N SYNFLOOD
$IPTABLES -A SYNFLOOD -p tcp --syn -m limit --limit 1/s -j RETURN
$IPTABLES -A SYNFLOOD -p tcp -j REJECT --reject-with tcp-reset
$IPTABLES -A INPUT -p tcp -m state --state NEW -j SYNFLOOD
# stop ping flood attack
$IPTABLES -N PING
$IPTABLES -A PING -p icmp --icmp-type echo-request -m limit --limit 1/second -j RETURN
$IPTABLES -A PING -p icmp -j REJECT
$IPTABLES -I INPUT -p icmp --icmp-type echo-request -m state --state NEW -j PING

##################################
## What we allow
##################################

# tcp ports

## restricted tcp things ##

# ssh
$IPTABLES -A INPUT -p tcp -m tcp -s $LAN --dport 22 -j ACCEPT
# samba (netbios)
$IPTABLES -A INPUT -p tcp -m tcp -s $LAN --dport 137:139 -j ACCEPT
# samba (not using netbios)
$IPTABLES -A INPUT -p tcp -m tcp -s $LAN --dport 445 -j ACCEPT
```

```
# udp ports

## restricted udp things ##

# Samba (Netbios)
$IPTABLES -A INPUT -p udp -m udp -s $LAN --dport 137:139 -j ACCEPT
$IPTABLES -A INPUT -p udp -m udp --sport 137:138 -j ACCEPT

# finally - drop the rest

$IPTABLES -A INPUT -p tcp --syn -j DROP
```

Execute the script with the command

```
[root@server]$ sh /home/john/ip.sh
```

Afterwards you must save the rules using this command

```
[root@server]$ /etc/rc.d/iptables save
```

Finally we must make sure that iptables is started on boot. So add **iptables** to the daemons array in */etc/rc.conf*

7 Samba server

7.1 Install

To install Samba use the following command :

```
[root@server]$ pacman -S samba
```

7.2 Configure

There are actually quite a few things we need to do before the samba server is ready for use. Don't worry, we'll take it one step at a time.

1. I never use the [homes] share with a samba server. The reason is that I like to have all the shared data in the same location (for easier backup). For that purpose I will make a directory under */var* where we'll place our shares.

   ```
   [root@server]$ cd /var
   [root@server]$ mkdir samba
   ```

2. Next we need to make the group for the sambausers and add the users to it.

   ```
   [root@server]$ groupadd sambausers
   [root@server]$ usermod -aG sambausers john
   ```

3. Then it's time to make some directories

   ```
   [root@server]$ cd /var/samba
   [root@server]$ mkdir iso
   [root@server]$ mkdir data
   [root@server]$ mkdir users
   [root@server]$ mkdir users/john
   ```

4. Change group, user and permissions on the directories

```
[root@server]$ cd /var
[root@server]$ chown -R john.users samba
[root@server]$ chmod -R 775 samba
[root@server]$ chmod -R 700 samba/users/*
```

5. cd to the directory containing the samba config file and make a copy of the default config file.

```
[root@server]$ cd /etc/samba
[root@server]$ cp smb.conf.default smb.conf
```

Now edit the file and change the lines if necesarry

```
# Global section

# Set the workgroup
workgroup = WORKGROUP

# Name displayed to clients, not to be confused with the netbios name
server string = Samba Server

# Set security
security = user

# grant access only to the hosts on the LAN, except the router
hosts allow = 192.168.100. EXCEPT 192.168.100.254

# I don't use the printers on my samba server, so I turn this off
load printers = no

# Backend to store user information in. The old backend smbpasswd will
# eventually be removed. So all new installations are encouraged to use
# the tdbsam backend
passdb backend = tdbsam

# Define what interfaces to listen on.
interfaces = 192.168.100.3/24

# Share Definitions
[iso]
  comment = Linux ISO images
  path = /var/samba/iso
  browseable = yes
  writable = no
  valid users = @sambausers

[data]
  comment = Writeable data folder for sharing documents
  path = /var/samba/data
  browseable = yes
  writable = yes
  valid users = @sambausers

[home]
  comment = %U Private Folder
  path = /var/samba/users/%U
  browseable = no
  writable = yes
  valid users = %U
```

Comment out the rest of the shares. Also all references to the printersharing. As I told earlier the printer shares aren't used in my setup. If you need them, leave them intact.

6. Now use the command **pdbedit** (This only works if you have **passdb backend = tdbsam** in your *smb.conf*). This is the preferred way for future installations. But if you want to use the old **smbpasswd** way, it is included in the Tips & Tricks section.

```
[root@server]$ pbedit -a john
```

```
New SMB password:
Retype new SMB password:
```

After entering the password an overview of the user is displayed

```
Unix username: john
NT username:
Account Flags: [U ]
User SID: S-1-5-21-1571225510-2317324014-3328139642-1003
Primary Group SID: S-1-5-21-1571225510-2317324014-3328139642-513
Full Name: ,,,
Home Directory: \serverjohn
HomeDir Drive:
Logon Script:
Profile Path: \serverjohnprofile
Domain: SERVER
Account desc:
Workstations:
Munged dial:
Logon time: 0
Logoff time: Tue, 19 Jan 2038 04:14:07 CET
Kickoff time: Tue, 19 Jan 2038 04:14:07 CET
Password last set: Sun, 11 Mar 2007 15:33:35 CET
Password can change: Sun, 11 Mar 2007 15:33:35 CET
Password must change: Tue, 19 Jan 2038 04:14:07 CET
Last bad password : 0
Bad password count : 0
Logon hours : FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

7.3 Start the server

Now it's time to fire up the samba server. You can do that either by issuing the command

```
[root@server]$ /etc/rc.d/samba start
```

or by rebooting after adding **samba** to the daemons array in */etc/rc.conf* (You should add the daemon anyhow, else samba wont start when you reboot your machine!)

7.4 Access shares

There are two different ways of mapping the shares to a local directory. **smbmount** and **mount.cifs** the last which can only be used with the **mount** command.

Of course it is also possible to use a graphical tool like **smb4k**, but that is actually just a frontend for the shown commands.

First let us try **smbmount** to access the shares. To do this as a normal user, you'll have to change the permissions on the files */usr/bin/smbmnt* and */usr/bin/smbumount*. For details please see the chapter on tips and tricks.

```
[john@workstation]$ cd ~
[john@workstation]$ mkdir smb
[john@workstation]$ smbmount //server/home ~/smb -o workgroup=WORKGROUP username=john
```

It asks for your password. And when given, the share will be connected.
Try the command

```
[john@workstation]$ mount
```

and verify that you have a reference to the share on the server. Here you can also see that it's mounted read/write (rw).
Unmount the share again with the command

```
[john@workstation]$ smbumount ~/smb
```

Next let us use **cifs** to access the shares.

```
[root@workstation]$ mount -t cifs //192.168.100.3/home ~/smb -o username=john
```

As you undoubtly already have noticed, it's only possible to connect the share using the root user. Unless you change the permissions on the files /sbin/mount.cifs and /sbin/umount.cifs of course. For details please see the chapter on tips and tricks. Then it's possible to use the following command to map the share to the local directory as a normal user.

```
[john@workstation]$ mount.cifs //192.168.100.3/home ~/smb -o username=john
```

To umount the share use the command

```
[john@workstation]$ umount.cifs ~/smb
```

Another way to access the shares on the server is to use the program smbclient. This works like a ftp client.

```
[john@workstation]$ smbclient //server/home
```

Just enter your password when asked for it, and you will be presented with a prompt like in a ftp client. To quit the client just type **quit**.

8 Tips and tricks

I have made this chapter to address some issues and tips & tricks which may be needed to make the setup work.

smbmount and smbumount

To make it possibe for a normal user to connect to a share we need to change the permissions on the files */usr/bin/smbmnt* and */usr/bin/smbumount* on the workstation. As they need to be executed with root permissions, the SUID bit needs to be set. To do this enter the command

[**root@workstation**]$ chmod u+s /usr/bin/smbmnt /usr/bin/smbumount

If you need to copy files larger than 2GB to a samba share then you have to include the option **lfs** when mounting the share with **smbmount**.

[**john@workstation**]$ smbmount //server/home ~/smb -o workgroup=WORKGROUP,username=john,lfs

mount.cifs and umount.cifs

To make it possibe for a normal user to connect to a share we need to change the permissions on the files */sbin/mount.cifs* and */sbin/umount.cifs* on the workstation. As they need to be executed with root permissions, the SUID bit needs to be set. To do this enter the command

[**root@workstation**]$ chmod u+s /sbin/mount.cifs /sbin/umount.cifs

smbpasswd

Here is how you can use **smbpasswd** (May be deprecated in the future) to add the users to the samba user list. Enter a password when asked for it.

[**root@server**]$ smbpasswd -a john

```
New SMB password:
Retype new SMB password:
```