
Setting up a Samba Active Directory

Autor:
Stefan KANIA

Ort:
St. Michaelisdonn

May, 9th 2023

Inhalt

1	Introduction	3
2	Advantages of Samba 4	3
3	Disadvantages of Samba 4	3
4	Installing the software	4
4.1	Installing from the distribution repositories	4
4.2	Advantages of installing via the repositories	4
4.3	Third party packages	5
4.3.1	Disadvantages of third party packages	5
4.4	Commercial packages	5
4.4.1	Disadvantages of commercial packages	5
5	Which version to choose?	5
6	What functions can Samba provide	6
7	The first domain controller	7
7.1	Installation of the packages	7
7.2	packages for a file or print server	8
8	Setting up the first domain controller	8
8.1	Setup Bind9	9
8.2	Preparing for startup	10
8.3	After reboot	11
8.4	Setting up the time server	13
9	User management	13
10	A second domain controller	14
10.1	Preparing the second domain controller	14
10.2	Joining domain	15
10.3	Next step	15
10.4	After restart	15
11	Replication of share sysvol	17
11.1	Setting up replication	18

12 Integrating Linux clients	20
12.1 Preparing a Linux client	20
12.2 Admit the client to the domain	21
13 The fileserver	22
13.1 The administrative share	22
13.2 Base directory of users	24
14 Linux-client and smb-mount	24
14.1 Setting up libpam-mount	25
15 What else works with Samba?	26
16 conclusion	27
Stichwortverzeichnis	27

1 Introduction

This year we will talk about setting up an Active Directory domain with Samba 4 not only about the actual setup, but also about the advantages and also the disadvantages of a Samba domain, compared to a Windows domain.

You may be faced with the decision to upgrade an existing old domain, or you may want to set up a completely new environment. Then why use Samba 4 and not Microsoft Active Directory?

In this year's tutorial I will show you how easy it is to set up and manage an Active Directory with Samba 4 and what the differences are to a Microsoft Active Directory.

2 Advantages of Samba 4

One big advantage of Samba 4 as an Active Directory is probably the licensing cost. This is also the point that is mentioned most often. However, it is not always the actual costs that are decisive for the decision in favor of Samba 4, but often the difficulties to determine which licenses are needed at all. Often this point alone leads to the exclusion of Microsoft.

But you should not necessarily limit the view only to the licenses, many other points speak for the use of Samba 4 as Active Directory.

- You can, now and in the future, keep your entire infrastructure within your organization. Microsoft is moving more and more toward SaaS and IaaS.
- The hardware equipment of the servers has lower requirements. Especially when it comes to upgrading an existing environment, this can be a point that clearly shows the advantage of Samba 4.
- Together with other open source products, such as GlusterFS/Ceph and CTDB, highly available file servers can be implemented on standard hardware without additional licensing costs.

3 Disadvantages of Samba 4

Where there is a lot of light, there is also shadow, for this reason the disadvantages of using Samba 4 should also be addressed here.

- In addition to Active Directory knowledge, good Linux knowledge is always necessary for system administration. If you do not have access to good Linux knowledge in your company, you should always keep an eye on the costs for further training. It may make sense to hire appropriate experts or, at least for a transitional period, to bring in external specialists.
- It is not possible to integrate Microsoft Exchange. The reason is that for MS-Exchange the schema of the Active Directory must be extended and this extension is not supported by Samba 4. Also a migration of a Windows domain in which an Exchange is integrated is not possible. Not even if the Exchange server is removed from the domain beforehand. This is because remnants of the Exchange environment still remain in the Active Directory.
- Replication of the important Sysvol share is not supported automatically, but must be provided by additional services.
- Not all possibilities of administration of servers and Active Directory environment can be realized by Remote Server Administration Tools (RSAT). For some administrative activities it is necessary to use the command line.
- Not all features, such as a tree of multiple domains, are supported.

4 Installing the software

Samba can be installed from various sources, you have the following options. With all possibilities you should always make sure that you use the most current Samba versions possible. The Samba team always maintains only the last three versions. If you use older versions, you should make sure that possible security holes can be closed. With a release cycle of 6 months until the next version, this means that after 1.5 years you should update to a more current version at the latest.

Install from source

Since Samba is open source software, you always have the option to take the current source code and compile Samba yourself, this way you are always up to date in any case.

Advantages of compiling Samba yourself

You always have the latest software version. You can exclude individual parts when compiling and thus each installation is exactly adapted to the needs of the system. If you have the knowledge, you have of course the possibility to change the source code and adapt it to your needs.

Disadvantages of compiling Samba yourself

You need a development environment to compile Samba, which then either has to be available on every system, or is provided centrally. The compiled version must then be copied to the different systems. Changes to certain libraries (for example, through updates), can cause the Samba service to no longer function properly. Every self-compiled version should always be checked on a test system first. Additional knowledge is required for compiling and possibly adapting the source code. In case of possible security vulnerabilities, you are responsible to compile the required packages yourself.

4.1 Installing from the distribution repositories

All distributions come with Samba packages. But, since Samba, at least for now, is still dependent on the Heimdal Kerberos, the domain controller for Active Directory cannot be installed on every distribution. On all Redhat based distributions it is not possible to install a domain controller with the distribution packages. The same is true for Suse products. Whereby Suse supports Samba domain controllers, but relies on the (still) experimental MIT Kerberos.

4.2 Advantages of installing via the repositories

The packages can be installed directly via the respective package manager. Dependencies are automatically resolved. As far as you use versions of your distribution which are still supported with updates, security gaps are closed here fast. Due to the simple installation, Samba servers can be set up and managed even by administrators with little Linux knowledge. If you pay attention to uniform distributions, the Samba versions, and thus also the range of functions, are always identical.

Disadvantages of the installation over the repositories

You are always dependent on the maintainers of the distribution for updates. Most distributions do not provide the latest versions of Samba, so you may not be able to use features you need from current versions right away. However, depending on the features provided by a Samba server, it may make sense to use the latest version. More about this later.

4.3 Third party packages

For some distributions, repositories are provided by individual developers, which you can then include and install into your distribution. These packages are often very well maintained and therefore could be an alternative to compiling them yourself.

Advantages of third-party packages

The packages are mostly up to date and can be included in the corresponding distribution without much problems. The packages are mostly provided by dedicated Samba team members.

4.3.1 Disadvantages of third party packages

The packages are often maintained, compiled and deployed by only one person. If that person, for whatever reason, stops updating the packages or does so very slowly, security problems can arise. In the worst case, you will have to reinstall all servers to be able to run secure systems again.

4.4 Commercial packages

There are also commercial packages for Samba, for example the packages of the company Sernet, these packages are offered to you via a subscription. The price of the subscription depends on how many servers you want to use and for how long. The packages are provided here with all functions for a wide range of distributions. Also the function of the domain controller for Redhat based distributions is offered there. The costs for the subscriptions are not the price for the software, Samba is also in this case open source, but for the included support of updates and additional features, such as use of Redhat distributions as a domain controller.

Advantages of commercial packages

In addition to providing the packages, you can also book support to help you set up and run your Samba system. The packages are provided for different version of the distributions. All updates and security updates are provided in a timely manner and can be installed via the distribution's package manager, just like the initial installation. Even when upgrading to a newer Samba version, you can assume that the update process has been tested and is working.

4.4.1 Disadvantages of commercial packages

Subscriptions are not free of charge. But for the money you pay for the subscription you also get the regular updates, the latest versions and a fast response to security vulnerabilities.

5 Which version to choose?

After enumerating the options, the question now arises: which type of installation is best for me? This answer always depends on the function you want to use and the knowledge in your company.

Whenever you want to use domain controllers, it makes sense to always use a very recent version of Samba, if possible, because most changes are made in the domain controller area. Always pay attention to the release notes, in which the new features are described. If you want to use one of the new functions, this is only possible with an update.

Also when using *Cluster Trival Data BaseCTDB* as a cluster you should use the latest versions of Samba if possible, because also here there are very often changes with new versions.

If you use Samba as a simple file server, you can also use versions that are not quite as up-to-date. So it is quite sufficient here to use the packages from the distributions.

If you use Linux with a graphical user interface on your clients, you can use the packages that come with the distribution. This way you will always have a consistent version on your clients.

6 What functions can Samba provide

After describing the different types and their advantages and disadvantages in the previous sections, we will now look at what services Samba can provide on the network.

Domaincontroller

The *domaincontroller* is not a service in the strict sense, it is composed of several services. The following services taken together make up the function domaincontroller.

- DNS
In Active Directory, a DNS server is always needed to propagate the services, *Kerberos*, *LDAP* and the *global Catalog* to the clients. This is because the client always asks its DNS servers registered with it where it can find the corresponding services. Each domain controller provides its own DNS server. As DNS server either the internal DNS server of Samba can be used, or the *Bind9*. In larger environments the *Bind9* is preferable in any case, because only the *Bind9* supports DNS-round-robin, for example to perform load balancing via DNS at CTDB.
- Kerberos
Kerberos provides encryption of passwords and data transfer within the domain. The Heimdal Kerberos is used here. The MIT-Kerberos is, for the moment, *experimental*. Since a standard kerberos is used here, it can also be used for authentication for services outside the Active Directory domain, for example authentication on web servers.
- LDAP
In the LDAP all objects and their attributes are stored. These are users, groups, computers, DNS entries and other objects. Samba 4 uses an LDAP server specially developed for Samba for this purpose, which cannot be exchanged for another LDAP server, for example OpenLDAP.
- Global Catalog
In the *global Catalog*, information of all objects in the entire *forest* of an Active Directory environment is stored. This includes information from all domains of the Active Directory. Only this way it is possible that objects can be found in other domains of the Active Directory. Only the most important attributes of an object are stored here.
- time server
Every domain controller must also provide the function of the time server, so that Windows clients can synchronize the time when logging on. By default, Windows clients will only accept the time if the information is signed by the domain controller, so the time server must be configured accordingly.

File server

Samba as *fileservers* is probably the most used function of Samba. An active directory is not necessarily required for the file server function. On a *standalone fileservers* users can be created locally and then clients can access the shares set up there. In most cases, however, you will make a file server a member of a domain. It doesn't matter if it is a Samba 4 domain or a Microsoft domain, a Samba file server can provide its services anywhere.

Print server

You can also use Samba on your network as a *print server*. As a print server, Samba can provide printers for all client operating systems. For Windows clients, you can also deploy printer drivers (up to Type 3) on a Samba print server and automatically install them on Windows clients when they access it.

Samba as cluster

Samba in combination with *Cluster Trivial Database CTDB* you can set up a highly available and loadbalancing file server cluster in your network. Together with a cluster file system like *ClusterFS* or *Ceph*. CTDB has been an integral part of Samba 4 since Samba version 4.2, and can be installed in the same ways as Samba itself, as described above. No proprietary software is required to set up a CTDB cluster; all parts are open source products. Together with standard hardware, you can thus integrate a cost-effective cluster into your network. Also a CTDB cluster can run in a Samba 4 domain as well as in a Windows domain.

7 The first domain controller

In the first part, the first domain controller is to be set up. There are several steps necessary for this.

7.1 Installation of the packages

The installation of the Samba packages is the only big difference between the distributions. Since I can't go into all distributions here, the focus is on Debian, although the same commands can be used on Ubuntu systems. The reason is that only with Debian and Ubuntu all functions can be used with the packages of the distribution. Thus Debian and Ubuntu offer the best basis for the use of Samba, if you want to use the distribution packages. Of course you can also set up Samba, depending on the function, on different distributions in your network, but it makes sense to use all systems of a service on the same distribution if possible, this simplifies the administration of the systems.

When installing the packages under Debian, you will notice that only Samba version 4.13 is installed there. This version is no longer actively supported by the Samba team. By including the *bullseye-backports*, we will use Samba version 4.17 here. Packages from the backports are more up to date, but still stable enough that you can use them productively.

For the tutorial, you have already done the required packages by setting them up via Vagrant. The reason is that this ensures that we don't have to rely too much on wifi for the installation.

Packages for a domain controller

If you want to set up a domain controller, the first consideration is: should the internal DNS or the Bind9 be used. This is because additional packages are needed for the Bind9. If you want to use the internal DNS server install the packages from Listing 7.1.1:

```
apt install -t bullseye-backports samba libpam-heimdal heimdal-clients \
    ldb-tools winbind libpam-winbind smbclient libnss-winbind bind9-host
```

Listing 7.1.1: installing the packages internal DNS

If you want to use Bind9 as a DNS server, install the packages from Listing 7.1.2:


```
apt install -t bullseye-backports samba libpam-heimdal heimdal-clients \
    ldb-tools winbind libpam-winbind smbclient libnss-winbind bind9 \
    dnsutils bind9-host
```

Listing 7.1.2: installing packages Bind9 as DNS

When choosing the DNS server, the decision is not final, if you want to use the internal DNS first and switch to the Bind9 later, you can do it with the command `samba-upgradedns --dns-backen=BIND9_DLZ`. But then it will be necessary to configure the Bind9 before you restart the service.

7.2 Packages for a file or print server

For a file or print server, you will need the packages from the listing 7.2.1:

```
apt install -t bullseye-backports samba libpam-heimdal heimdal-clients \
    winbind libpam-winbind smbclient libnss-winbind bind9-host
```

Listing 7.2.1: packages for a file or print server

These are the packages you need for all services, additionally you need the `cups` package if you want to set up a print server. Since the actual printing is handled by *CUPS*. Samba only converts the printers that are set up in CUPS so that they can be used under Windows.

If your server is to be part of a cluster, install the `ctdb` package as well.

8 Setting up the first domain controller

Before actually setting up the first domain controller, it is important that you have made and/or checked the following, basic, settings on your system:

- Check that the `/etc/hosts` file has only the entries on *localhost* and the own IP address with the *fqdn* and the *hostname* entered. These values are read from the file when the domain controller is set up.
- Make sure that the system has a fixed IP address. A domain controller is always bound to one IP address. It is very difficult to change the IP address later.
- Check that the *fqdn* of the system is displayed to you with the command `hostname -f`.
- Delete the `/etc/samba/smb.conf` file. This file is from the installation of the packages (If you are using the sernet packages, this file is not present). When provisioning, the file is automatically generated.

One of the most important tools for managing an Active Directory domain is the `samba-tool` program. With the help of this program, you can perform almost all domain tasks. After the packages are installed, enter the command `samba-tool` without any other parameters and you will get an overview of the different tasks you can perform with *samba-tool*.

Now that you have considered the points, the next step is to set up *provisioning* of the domain controller. For *provisioning* you have two different options. On the one hand, you can simply call the command `samba-tool domain provision`, then all required information will be queried interactively. After the query, provisioning will follow.

The other option is to pass all required parameters to the command `samba-tool domain provision` on the command line. In this case, you have the possibility to pass additional parameters that you do not have in the interactive provisioning. You can display all possible parameters using the command `samba-tool domain provision -h`.

The simplest way of provisioning is the interactive approach, we will use that here as well. In Listing 8.1 you can see the output of provisioning:

```

root@addc-01:~# samba-tool domain provision
Realm [EXAMPLE.NET]:
Domain [EXAMPLE]:
Server Role (dc, member, standalone) [dc]:
DNS backend (SAMBA_INTERNAL, BIND9_FLATFILE, BIND9_DLZ, NONE) \
[SAMBA_INTERNAL]: BIND9_DLZ
Administrator password:
Retype password:
...
Server Role: active directory domain controller
Hostname: addc-01
NetBIOS Domain: EXAMPLE
DNS Domain: example.net
DOMAIN SID: S-1-5-21-2057776938-3237700937-2176600150

```

Listing 8.1: description

Except for the selection of the DNS server to use and the administrator's password, you can simply confirm all questions here with RETURN. Pay attention to the complexity rules for the password. The password must be at least seven characters long and contain at least three of the four categories (upper case letters, lower case letters, numbers, special characters). Furthermore, it is important to know that, unlike Microsoft, the password has an expiration date of 42 days.

HINT! ☺

You can change the administrator's password at any time, as root, with the command `samba-tool user setpassword administrator`.

8.1 Setup Bind9

Since we want to use the *Bind9* as DNS server here, but the DNS server is a main part of the domain controller, it must be set up before the actual service can be started. This requires changes to two files. First, modify the `/etc/bind/named.conf.options` file as shown in Listing 8.1.1:

```

forwarders {
    1.1.1.1;
};
tkey-gssapi-keytab "/var/lib/samba/bind-dns/dns.keytab";

dnssec-validation no;

```

Listing 8.1.1: customization of `named.conf.options` file

The *forwarders* entry specifies which DNS server to ask when the local bind9 cannot resolve a query. The *tkey-gssapi-keytab* entry points to the keytab file needed for the Bind9 to authenticate to the Kerberos service. Since the Bind9 must be able to read and write information, and the information is all stored in Active Directory, authentication is required. Since DNSSEC is not used, the option should be turned off here.

Now the changes to the file `named.conf.local` are missing. If Bind9 is used without Active Directory, the zones for which Bind9 is responsible are entered in this file. Since all zones are entered in the Active Directory, only the connection to the Active Directory is configured here. The entries you make here can be seen in Listing 8.1.2:

```
include "/var/lib/samba/bind-dns/named.conf";
```

Listing 8.1.2: adjustments to the `named.conf.local` file

Here, only the corresponding configuration file created during provisioning is referenced. In the file itself only one entry for the own Bind9 version is activated.

Now the Bind9 can be restarted and you can check if the Bind9 can also access the Active Directory. In Listing 8.1.3 you can see the corresponding steps and the result of the test:

```
root@addc-01:~# systemctl restart bind9
root@addc-01:~# tail -n 200 /var/log/syslog
...
samba_dlz: started for DN DC=example,DC=net
samba_dlz: starting configure
samba_dlz: configured writeable zone 'example.net
samba_dlz: configured writeable zone '_msdcs.example.net' ...
...
```

Listing 8.1.3: First start of Bind9

After the reboot, the end of the log file is displayed. Scroll up the display until you see the lines shown here. Now the Bind9 is configured and started.

8.2 Preparing for startup

Before you can start the domain controller service, it is necessary to perform additional steps.

Copy Kerberos client file

Copy the `/var/lib/samba/private/krb5.conf` file to the `/etc` directory. This ensures that the Kerberos server can be reached. There is already a `krb5.conf` in which many entries are redundant and not all needed entries are present.

Exclude NAT device

There is one more customization missing here in the tutorial environment. When working with Vagrant and Linux, a network card configured via NAT is always needed. Since Samba is always active on all network cards of the system and also all IP addresses that are active are always entered the DNS, we must ensure here that only the second network card is used.

The problem is that with all NAT entries of all hosts always the IP address 10.0.2.15 is used. If all systems now enter the IP address into the DNS, conflicts will occur. Therefore, enter the two lines from Listing 8.2.1 in the global area of `smb.conf`:

```
interfaces = 192.168.56.41
bind interfaces only = yes
```

Listing 8.2.1: select network card

This applies to all systems used during the tutorial.

Customize the resolver

Still missing is the customization of the *resolver*. In the future the domain controller should always query the own DNS server if a name has to be resolved. To do this, modify the `/etc/network/interfaces` file as shown in Listing 8.2.3:

```
allow-hotplug eth0
iface eth0 inet static
    address 10.0.2.15
    netmask 255.255.255.0
    gateway 10.0.2.2
    dns-nameservers 192.168.56.41
    dns-search example.net
```

Listing 8.2.2: description

Enable services

When you install Samba, all services are configured as *standalone server* by default. Accordingly, the required services are also started. Now, to start Samba as a domain controller as well, run the commands from Listing 8.2.2:

```
root@addc-01:~# systemctl disable --now smbd nmbd winbind
Synchronizing state of smbd.service with SysV service script with \
    /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable smbd
Synchronizing state of nmbd.service with SysV service script with \
    /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable nmbd
Synchronizing state of winbind.service with SysV service script with \
    /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable winbind.
Removed /etc/systemd/system/multi-user.target.wants/nmbd.service.
Removed /etc/systemd/system/multi-user.target.wants/winbind.service.
Removed /etc/systemd/system/multi-user.target.wants/smbd.service.

root@addc-01:~# systemctl unmask samba-ad-dc

root@addc-01:~# systemctl enable --now samba-ad-dc
Synchronizing state of samba-ad-dc.service with SysV service \
    script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable samba-ad-dc
Created symlink /etc/systemd/system/multi-user.target.wants/\
    samba-ad-dc.service -> /lib/systemd/system/samba-ad-dc.service.
```

Listing 8.2.3: activating services

With the first command `systemctl disable --now smbd nmbd winbind` all standalone services are disabled and also stopped immediately. This is because the domain controller brings its own instances of these services. The next command `systemctl unmask samba-ad-dc` is used to start the service in the first place. The third command `systemctl enable --now samba-ad-dc` enables the service and starts it immediately. Now, when the system is rebooted, the domain controller is also always started immediately.

NOTE! ⓘ

This example are the settings for systems in the Vagrant environment

Now, to ensure that all services will restart properly when the system is rebooted, reboot the system.

8.3 After reboot

Now that the domain controller has been restarted, here are a few tests to verify that the services are working.

Testing the DNS

The first thing to be tested is the DNS. Listing 8.3.1 shows the commands and the expected results:

```
root@addc-01:~# host addc-01
addc-01.example.net has address 192.168.56.41

root@addc-01:~# host -t srv _ldap._tcp.example.net
_ldap._tcp.example.net has SRV record 0 100 389 addc-01.example.net.

root@addc-01:~# host -t srv _kerberos._tcp.example.net
```

```
_kerberos._tcp.example.net has SRV record 0 100 88 addc-01.example.net.

root@addc-01:~# host -t srv _gc._tcp.example.net
_gc._tcp.example.net has SRV record 0 100 3268 addc-01.example.net.
```

Listing 8.3.1: Testing the DNS

The tests show that the own hostname and the services provided by the domain controller can be resolved.

Testing the Kerberos

To test the Kerberos server, a ticket should now be requested for the administrator, then the ticket is displayed. In Listing 8.3.2 you can see the commands and results:

```
root@addc-01:~# kinit administrator
administrator@EXAMPLE.NET's password:

root@addc-01:~# klist
Credentials cache: FILE:/tmp/krb5cc_0
    Principal: administrator@EXAMPLE.NET

    Issued Expires Principal
Jan 11 14:22:11 2023 Jan 12 00:22:11 2023 krbtgt/EXAMPLE.NET@EXAMPLE.NET
```

Listing 8.3.2: testing-kerberos

The first step is to request the ticket. The second step displays all of the user's tickets. Since no service has been accessed at this point, only the user's own *ticket-granting-ticket(tgt)* is displayed here.

Testing connection establishment

Two shares `sysvol` and `netlogon` are also always set up on each domain controller. These shares also exist on each Windows domain controller. The connection setup test is used to verify that the domain controller can provide these shares. Listing 8.3.3 shows the test:

```
root@addc-01:~# smbclient -L addc-01 -N

    Sharename Type Comment
    -----
    sysvol Disk
    netlogon Disk
    IPC$ IPC IPC Service (Samba 4.17.4-Debian)
SMB1 disabled -- no workgroup available

root@addc-01:~# klist
Credentials cache: FILE:/tmp/krb5cc_0
    Principal: administrator@EXAMPLE.NET

    Issued Expires Principal
Jan 11 14:22:11 2023 Jan 12 00:22:11 2023 krbtgt/EXAMPLE.NET@EXAMPLE.NET
Jan 11 14:43:52 2023 Jan 12 00:22:11 2023 cifs/addc-01@EXAMPLE.NET
```

Listing 8.3.3: test connection setup

The `-N` option ensures that you are no longer prompted for a password. Since you have already requested a Kerberos ticket before, Kerberos is immediately used for authentication here. From the result of the second command you can immediately see that you have received an appropriate service ticket.

Testing the LDAP server

The last service to be tested now is the LDAP. Since the package `ldb-utils` was installed in the tutorial environment, the corresponding commands for accessing the LDAP server are also available. When accessing the LDAP server, the `ldap` protocol is used, along with `kerberos`. The connection is encrypted by using Kerberos. Due to security issues, access via `ldaps` was disabled, some time ago. Listing 8.3.4 shows the access to the LDAP and then the Kerberos tickets are listed again:

```
root@addc-01:~# ldbsearch -H ldap://addc-01 cn=administrator -k yes
# record 1
dn: CN=Administrator,CN=Users,DC=example,DC=net
...

root@addc-01:~# klist
Credentials cache: FILE:/tmp/krb5cc_0
Principal: administrator@EXAMPLE.NET

Issued Expires Principal
Jan 11 14:22:11 2023 Jan 12 00:22:11 2023 krbtgt/EXAMPLE.NET@EXAMPLE.NET
Jan 11 14:43:52 2023 Jan 12 00:22:11 2023 cifs/addc-01@EXAMPLE.NET
Jan 11 14:53:17 2023 Jan 12 00:22:11 2023 ldap/addc-01@EXAMPLE.NET
```

Listing 8.3.4: testing the LDAP

Now you have tested all the services that the domain controller provides.

8.4 Setting up the time server

The last thing left is to set up the time server so that the Windows clients can synchronize the time when they log in. To do this, first install the `ntp` package. Then copy the `/data/ntp.conf` file to the `/etc` directory. The important line in the file is `ntpsigndsocket /var/lib/samba/ntp_signd/` because the socket that signs the time packets to the client is located in the directory specified here. Only the permissions on the directory are not correct, the group `ntp` must have the permissions `r-x` on the directory. Change the owning group with the command `chgrp ntp /var/lib/samba/ntp_signd/`. Then restart the `ntp`.

This completes the full setup of the first domain controller.

9 User management

At this point I would like to very briefly discuss user and group management. You can manage users and groups from the command line of the domain controller as well as from the *Remote Server Administration Tools* (RSAT) provided by Microsoft. All actions you perform for users and groups on the command line can be tracked in RSAT and vice versa. To use RSAT, you need a Windows client that is a member of the domain. Adding a Windows client to the domain is no different from adding it to a Windows domain.

You manage the users with the command `samba-tool user`. If you just type the command, you will see all possible options.

For group management use the command `samba-tool group`. Again, if you enter only the command, you will get a list of options.

In figure 9.1 you can see the overview of users and groups.

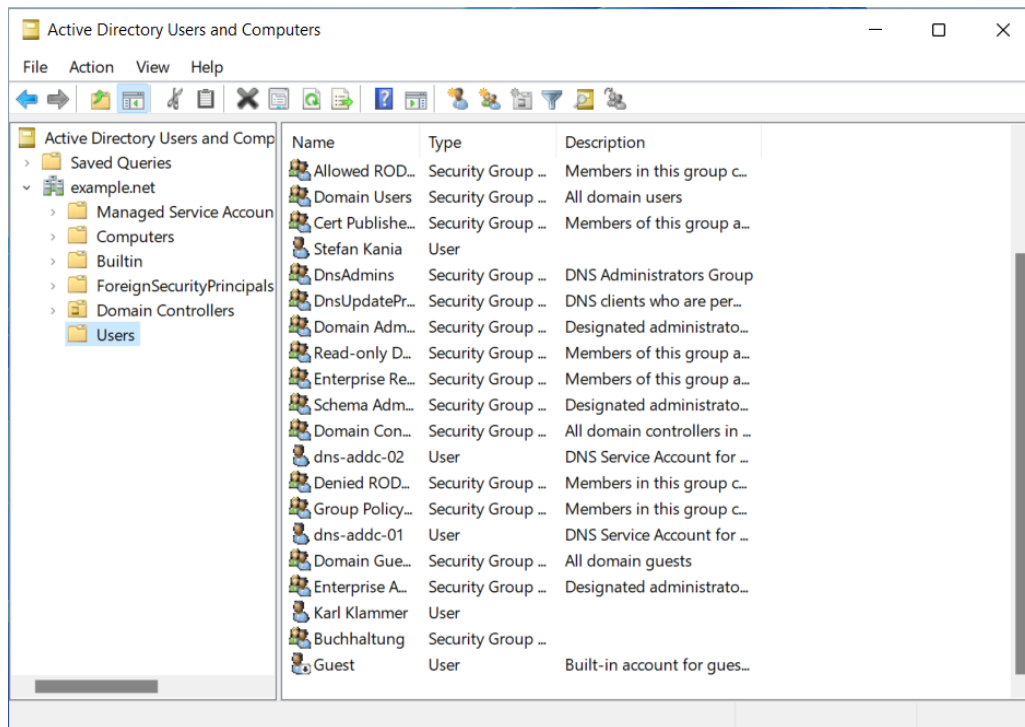


Abbildung 9.1: users and groups

10 A second domain controller

Up to this point, you are running only one domain controller in your domain. If the one domain controller fails, no one can log on to the domain. If the domain controller also fails to restart, you may lose all user information and experience significant downtime. To avoid this *single point of failure*, it is advisable to always operate at least two domain controllers in one domain.

The machine for the second domain controller is already installed. Each additional domain controller can always provide all services, so the failure of one domain controller does not lead to the failure of the entire domain.

10.1 Preparing the second domain controller

As with the first domain controller, the first step is to install the packages. This step has already been done here in the tutorial. In addition to the basic settings you already made on the first domain controller before provisioning, it is important that you set the *resolver* of the second system so that the IP address of the first domain controller is entered as the *nameserver*, otherwise the domain cannot be found and the new domain controller cannot join the domain. Only when you can resolve the hostname of the first domain controller and the *srv-records* you may proceed with the next steps. Listing 10.1.1 shows again the corresponding commands:

```
root@addc-02:/# host addc-01
addc-01.example.net has address 192.168.56.41

root@addc-02:/# host -t srv _ldap._tcp.example.net
_ldap._tcp.example.net has SRV record 0 100 389 addc-01.example.net.

root@addc-02:/# host -t srv _kerberos._tcp.example.net
_kerberos._tcp.example.net has SRV record 0 100 88 addc-01.example.net.

root@addc-02:~# host -t srv _gc._tcp.example.net
```

_gc._tcp.example.net has SRV record 0 100 3268 addc-01.example.net.

Listing 10.1.1: testing name resolution

10.2 Joining domain

After you have set all the defaults, you can now add the domain controller to the domain. To do this, the `samba-tool` program is used again. This time, however, the operation is performed directly through the command targets, since certain parameters are required that cannot be passed when using the program interactively. Listing 10.2.1 shows the command and an abbreviated output of the result:

```
root@addc-02:~# samba-tool domain join --dns-backend=BIND9_DLZ example.net DC \
--realm=example.net -U administrator
Finding a writeable DC for domain 'example.net'.
Found DC addc-01.example.net
...
Joined domain EXAMPLE (SID S-1-5-21-2057776938-3237700937-2176600150) as a DC
```

Listing 10.2.1: add second domain controller

During the process, all Active Directory data is replicated from the first domain controller to the second, and replication is set up between the two domain controllers so that all changes, no matter which domain controller you make the changes on, are always replicated to the other domain controller.

10.3 Next step

To ensure that the domain controller is then fully integrated into the domain, the following steps are now required.

- Copy the two configuration files `named.conf.options` and `named.conf.local` to the second domain controller. Here in the tutorial, you can find the two files in the `/data` directory of the second domain controller.
- Copy the `/var/lib/samba/private/krb5.conf` file to the `/etc` directory.
- Make sure that after a system reboot, the own IP address is entered as resolver in the file `/etc/resolv.conf`.
- Edit the file `/etc/samba/smb.conf`, as already on the first domain controller, that only the IP address 192.168.56.42 is used by Samba.
- As on the first domain controller, you need to disable the standalone services and enable the `samba-ad-dc` service.

Only when using the Vagrant environment

When adding to the domain, all network cards are entered in DNS for the new domain controller, including the NAT interface with IP 10.0.2.15. Since all systems use the same IP address for the NAT interface, be sure to remove the IP address from DNS using the command from Listing 10.3.1:

```
root@addc-01:~# samba-tool dns delete addc-01 example.net addc-02 A 10.0.2.15 -N
Record deleted successfully
```

Listing 10.3.1: Entfernen des NAT-Interfaces

Then restart the server.

10.4 After restart

After restarting, you can run the tests you already ran on the first domain controller on the second domain controller.

You will notice a big difference when querying the srv records, because both domain controllers are now displayed there. This is also the correct display, because the services are offered by both domain controllers and thus also propagated via DNS. If you repeat one of the commands for resolving the srv record several times, you will notice that the entries are always displayed in a different order. This is caused by Bind9 which changes the order of the entries again and again via DNS-round-robin. So the requests of the clients are always forwarded to other domain controllers and both domain controllers are used. This also speaks for the use of Bind9 as a name server.

It is useful to check occasionally if the replication between the domain controllers works. Listing 10.4.1 shows the command and the output:

```
root@addc-01:~# samba-tool drs showrepl
default-first-site-name\ADDC-01
DSA Options: 0x000001
DSA object GUID: 7410dc40-16c2-400c-b3a7-7582f7e83e74
DSA invocationId: 4b9a305e-fca3-49eb-af5b-09a6b2caae40

==== INBOUND NEIGHBORS ====

DC=ForestDnsZones,DC=example,DC=net
  Default-First-Site-Name\ADDC-02 via RPC
    DSA object GUID: b19f98d3-ed2c-4a17-ba76-bd81546f1034
    Last attempt @ Wed Jan 11 18:12:28 2023 UTC was successful
    0 consecutive failure(s).
    Last success @ Wed Jan 11 18:12:28 2023 UTC

CN=Configuration,DC=example,DC=net
  Default-First-Site-Name\ADDC-02 via RPC
    DSA object GUID: b19f98d3-ed2c-4a17-ba76-bd81546f1034
    Last attempt @ Wed Jan 11 18:12:28 2023 UTC was successful
    0 consecutive failure(s).
    Last success @ Wed Jan 11 18:12:28 2023 UTC

CN=Schema,CN=Configuration,DC=example,DC=net
  Default-First-Site-Name\ADDC-02 via RPC
    DSA object GUID: b19f98d3-ed2c-4a17-ba76-bd81546f1034
    Last attempt @ Wed Jan 11 18:12:28 2023 UTC was successful
    0 consecutive failure(s).
    Last success @ Wed Jan 11 18:12:28 2023 UTC

DC=example,DC=net
  Default-First-Site-Name\ADDC-02 via RPC
    DSA object GUID: b19f98d3-ed2c-4a17-ba76-bd81546f1034
    Last attempt @ Wed Jan 11 18:12:28 2023 UTC was successful
    0 consecutive failure(s).
    Last success @ Wed Jan 11 18:12:28 2023 UTC

DC=DomainDnsZones,DC=example,DC=net
  Default-First-Site-Name\ADDC-02 via RPC
    DSA object GUID: b19f98d3-ed2c-4a17-ba76-bd81546f1034
    Last attempt @ Wed Jan 11 18:12:28 2023 UTC was successful
    0 consecutive failure(s).
    Last success @ Wed Jan 11 18:12:28 2023 UTC

==== OUTBOUND NEIGHBORS ====

DC=ForestDnsZones,DC=example,DC=net
  Default-First-Site-Name\ADDC-02 via RPC
    DSA object GUID: b19f98d3-ed2c-4a17-ba76-bd81546f1034
```

```

        Last attempt @ NTTIME(0) was successful
        0 consecutive failure(s).
        Last success @ NTTIME(0)

CN=Configuration,DC=example,DC=net
    Default-First-Site-Name\ADDC-02 via RPC
        DSA object GUID: b19f98d3-ed2c-4a17-ba76-bd81546f1034
        Last attempt @ NTTIME(0) was successful
        0 consecutive failure(s).
        Last success @ NTTIME(0)

CN=Schema,CN=Configuration,DC=example,DC=net
    Default-First-Site-Name\ADDC-02 via RPC
        DSA object GUID: b19f98d3-ed2c-4a17-ba76-bd81546f1034
        Last attempt @ NTTIME(0) was successful
        0 consecutive failure(s).
        Last success @ NTTIME(0)

DC=example,DC=net
    Default-First-Site-Name\ADDC-02 via RPC
        DSA object GUID: b19f98d3-ed2c-4a17-ba76-bd81546f1034
        Last attempt @ NTTIME(0) was successful
        0 consecutive failure(s).
        Last success @ NTTIME(0)

DC=DomainDnsZones,DC=example,DC=net
    Default-First-Site-Name\ADDC-02 via RPC
        DSA object GUID: b19f98d3-ed2c-4a17-ba76-bd81546f1034
        Last attempt @ NTTIME(0) was successful
        0 consecutive failure(s).
        Last success @ NTTIME(0)

==== KCC CONNECTION OBJECTS ====

Connection --
    Connection name: 40b844fa-13c7-4530-9c95-44ba24fba175
    Enabled : TRUE
    Server DNS name : addc-02.example.net
    Server DN name : CN=NTDS Settings,CN=ADDC-02,CN=Servers,\
CN=Default-First-Site-Name,CN=Sites,CN=Configuration,\
DC=example,DC=net
    TransportType: RPC
    options: 0x000001
Warning: No NC replicated for Connection!
Listing 10.4.1: checking replication

```

The query was performed on the second domain controller. Here we distinguish between *INBOUND NEIGHBORS* and *OUTBOUND NEIGHBORS* replication. So what data comes in and what is sent out. Now here are only two domain controllers in use, if you run multiple domain controllers, the list is correspondingly long. Mostly you only want to see the possible errors and not what worked. For this reason there is the option *--summary* which only shows a short result if everything works. If there are errors at any point of the replication, only the errors will be shown. Listing 10.4.2 shows the summarized display:

```

root@addc-01:~# samba-tool drs showrepl --summary
[ALL GOOD]

```

Listing 10.4.2: Summary display of replication test

As long as only this one line is displayed, everything is fine.

All that is left is to set up the time server. Proceed exactly as you did with the first domain controller. At this point, all information from the LDAP is now replicated to both domain controllers. No

matter on which of the domain controllers make changes, the other domain controller will apply the changes.

All that is missing now is replication of the share `sysvol`. The replication of the share follows in the next section.

11 Replication of share `sysvol`

In addition to the database containing Active Directory objects, which is present on all domain controllers, the data of the share `sysvol` is also required. This share contains the information about the *Group Policy (GPO)* and any logon scripts that may still exist. This information must also be provided on all domain controllers. In a Microsoft Active Directory, share replication is implemented using a separate protocol; unfortunately, this is not possible in this way under Samba. Therefore, it is necessary to set up share replication manually.

When replicating the share `sysvol` in a Samba 4 domain, one of the domain controllers will always be the writing instance and all other domain controllers will replicate the data only from this domain controller. Only which of the domain controllers should take on this role? In principle, any domain controller can take over the role. But, there is one domain controller that is predestined to take over this role.

The fsmo roles

According to Microsoft, every domain controller in a domain has equal rights and you can change all data on all domain controllers. However, this is not the whole truth. There are seven additional roles the *flexible single master operation roles (fsmo)*. These roles set certain functions on a domain controller of the domain. Among others, there is the *PdcEmulationMasterRole* role. If the domain controller is not accessible with this role, certain changes, such as password changes, can no longer be made. Also, the domain controller with this role is always the default selection when you launch the GPO editing tool. Since the information of the GPOs is changed there by default, it makes sense to set this domain controller as the master of the replication as well. How to determine the domain controller that holds this, and also all other roles, you can see in Listing 11.1:

```
root@addc-02:~# samba-tool fsmo show
SchemaMasterRole owner: CN=NTDS Settings,CN=ADDC-01,CN=Servers,\
    CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=example,DC=net
InfrastructureMasterRole owner: CN=NTDS Settings,CN=ADDC-01,CN=Servers,\
    CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=example,DC=net
RidAllocationMasterRole owner: CN=NTDS Settings,CN=ADDC-01,CN=Servers,\
    CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=example,DC=net
PdcEmulationMasterRole owner: CN=NTDS Settings,CN=ADDC-01,CN=Servers,\
    CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=example,DC=net
DomainNamingMasterRole owner: CN=NTDS Settings,CN=ADDC-01,CN=Servers,\
    CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=example,DC=net
DomainDnsZonesMasterRole owner: CN=NTDS Settings,CN=ADDC-01,CN=Servers,\
    CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=example,DC=net
ForestDnsZonesMasterRole owner: CN=NTDS Settings,CN=ADDC-01,CN=Servers,\
    CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=example,DC=net
```

Listing 11.1: list all fsmo

It doesn't matter on which of your domain controllers you start the query.

In our case, all roles are on `addc-01`. This is also the first domain controller we set up. All roles can be transferred to another domain controller later if it becomes necessary. One reason would be if you want to permanently remove the domain controller from the domain.

11.1 Setting up replication

To replicate the `sysvol` share, the `rsync` program is used. Together with the `systemd`, a `rsync` server is set up on the first domain controller. On all other domain controllers, `rsync` is used as a client to replicate the information from the server.

Setting up the rsync server

First, install the `rsync` package. For the `rsync` server, you need a configuration file `/etc/rsyncd.conf`. You can see the contents of the file in Listing 11.1.1:

```
[sysvol]
path = /var/lib/samba/sysvol/
comment = Samba sysvol
uid = root
gid = root
read only = yes
auth users = sysvol-repl
secrets file = /etc/samba/rsync.secret
```

Listing 11.1.1: The `rsyncd.conf` file

A copy of the file can be found on the domain controller `addc-01` in the `/data` directory. Copy the file to the `/etc` directory. In the `/etc/samba` directory, create the `rsync.secret` file with the contents from Listing 11.1.2:

```
sysvol-repl:secret
```

Listing 11.1.2: contents of file `rsync.secret`

Next, make sure that the file belongs to the user and group `root` and that the file has the permissions `600`. If the permissions are not correct, the service will not start. Then restart the service with the command `systemctl restart rsync`. Check the status with `systemctl status rsync`.

Setup rsync client

On all other domain controllers, `rsync` is required to replicate the data from the `rsync` server. Therefore, install `rsync` on the other domain controller as well.

After installation, create a file `/etc/samba/rsync.pass`. The file must contain only the password of the user defined on the `rsync` server. Make sure that the permissions are the same as on the server.

Now you can run the first test with the command from Listing 11.1.3. The parameter `--dry-run` ensures that only the right thing is replicated. The parameter `--delete-after` must not be forgotten, because this parameter ensures that files which have been deleted on the `rsync` server since the last replication are also deleted on the client.

```
root@addc-02:~# rsync -XAavz --delete-after --dry-run \
    --password-file=/etc/samba/rsync.pass \
    rsync://sysvol-repl@addc-01/sysvol \
    /var/lib/samba/sysvol/
receiving file list ... done
./
example.net/
example.net/policies/
example.net/Polices/{31B2F340-016D-11D2-945F-00C04FB984F9}/
example.net/Polices/{31B2F340-016D-11D2-945F-00C04FB984F9}/GPT.INI
example.net/Polices/{31B2F340-016D-11D2-945F-00C04FB984F9}/MACHINE/
example.net/Polices/{31B2F340-016D-11D2-945F-00C04FB984F9}/USER/
example.net/Polices/{6AC1786C-016F-11D2-945F-00C04FB984F9}/
example.net/Polices/{6AC1786C-016F-11D2-945F-00C04FB984F9}/GPT.INI
```

```
example.net/Policies/{6AC1786C-016F-11D2-945F-00C04FB984F9}/MACHINE/
example.net/Policies/{6AC1786C-016F-11D2-945F-00C04FB984F9}/USER/
example.net/scripts/
```

```
sent 59 bytes received 1,128 bytes 791.33 bytes/sec
total size is 40 speedup is 0.03 (DRY RUN)
```

Listing 11.1.3: First test of replication

Only when you see this result, you can remove the `--dry-run` parameter from the command and perform the replication. A look at the `/var/lib/samba/sysvol/example.net/` directory now shows that the data has been replicated.

After the first replication was successful, now make sure that the replication is executed regularly. To do this, you can write the command directly into the *crontab* of the *root* user, or you can first create a shell script which is then executed via the *cron*. In the `/data` directory, on the domain controller *addc-02*, you will find a script called `sysvol-repl.bash` that you can use for this. How often you run the script depends on how many changes you make to GPOs and login scripts.

So now you have a domain with two domain controllers replicating all the data you need. The next section is about integrating Linux clients and file servers.

12 Integrating Linux clients

The term *Linux client* here is always in reference to the Active Directory domain. Both a later Linux file server and a Linux workstation are, from the domain's point of view, always just clients that want to use central authentication via Active Directory. The setup of the actual file server follows only in the next section.

On a Linux client you install the identical Samba packages as already on the domain controllers, you do not need the packages for `bind9`, just like the package `ldb-utils` here. For the tutorial all necessary packages are already installed.

Since the following steps are identical on all client systems, it doesn't matter on which of the systems you start. I will use the Vagrant host *fs01* in the examples here, since it will be configured as a file server afterwards.

12.1 Preparing a Linux client

on the client, now perform the following steps:

- Again, the first thing you can do is delete the `/etc/samba/smb.conf` file, as it is unusable for use as an Active Directory client.
- Check the file `/etc/hosts` there must be, besides the *localhost* line, only one line with the own IP address, the fqdn and the short name.
- Make sure that at least one domain controller, better two, is entered as *resolver*.
- Check that the command `hostname -f` displays the fqdn of the client

Now the new `smb.conf` is missing which is now filled with the contents from Listing 12.1.1:

```
[global]
    workgroup = EXAMPLE
    realm = EXAMPLE.NET
    security = ADS
    winbind refresh tickets = yes
    winbind use default domain = yes
```

```

template shell = /bin/bash
idmap config * : range = 10000 - 19999
idmap config EXAMPLE : backend = rid
idmap config EXAMPLE : range = 10000000 - 19999999
interfaces = 192.168.56.51
bind interfaces only = yes

```

Listing 12.1.1: the `smb.conf` file for the client

A prepared file can be found in the `/data` directory, both on the *client-01* and on the *fs-01* file server.

These settings are identical on all Linux clients in your domain. The parameters have the following meanings:

- *workgroup = example*
Here the NetBIOS name of the domain is specified. Also as a member of AD, the parameter is called *workgroup*.
- *realm = EXAMPLE.NET*
The *realm* is the information for the Kerberos-domain. For this realm the Samba server will look for a KDC.
- *security = ADS*
This specifies that your server is a member of an AD domain.
- *winbind refresh tickets = yes*
This parameter will automatically refresh Kerberos tickets when the user is logged in and the ticket expires.
- *winbind use default domain = yes* If you display the users with `wbinfo -u`, the users will always be displayed with the domain name in front. If you have only one domain, without any trusts to other domains, you can use this parameter to make sure that the user name is listed without the domain name. This has the advantage that you can also work with only the user name when assigning permissions. Of course, this also applies to the groups from your domain.
- *template shell = /bin/bash*
You must not forget this parameter at all. Without it, a user can log in from AD, but he will be logged out immediately, because the user is not assigned a shell in AD, but it is needed for a successful login.
- *idmap config * : range = 10000 - 19999*
In addition to the groups and users that you create as an administrator, there are also the Built-in-Groups. These groups have their own shortened SID. You must also configure ID mapping for these groups. The configuration of the Built-in-Groups via the star is *idmap config * : range = 1000000 - 1999999*. The parameter *idmap config * : backend = tdb* you would actually have to configure as well, but this parameter is set automatically by Samba4. You can test it with the command *testparm*.
- *idmap config EXAMPLE : backend = rid*
The IDs of the users must be generated from the SIDs of the AD users. There are several ways to do this. The default setting for the *winbind* is to use *.tdb* files. This generates random UIDs and assigns them to users and stores them in the *.tdb* file. The disadvantage of this method is that this way every user on every Linux system gets a different UID. By switching to the backend *idmap_rid*, the RID of the user from the AD domain is always chosen. Since this is unique, the ID of the users and groups on the Linux system is also unique. This means that the user always has the same UID on all Linux systems in the entire domain. However, there is one problem that cannot be solved in this way, and that is that the UIDs on the DCs in your domain are always managed in AD and are therefore always different from those on the other systems in the domain. The easiest way around this problem is to not use the domain controllers as file servers and to always store all files on other Samba servers in the domain.
- *idmap config EXAMPLE : = 1000000 - 1999999*
Here you specify the range in which the UIDs of the users should be located.

12.2 Admit the client to the domain

Now the time has come, the client can be added to the domain. To join the domain, you need a user account in the domain and the user must be a member of the *domain admins* group. Listing 12.2.1 shows the command, the associated output, and the test to see if the join was successful:

```
root@fs-01:~# net ads join -U administrator
Password for [EXAMPLE\administrator]:
Using short domain name -- EXAMPLE
Joined 'FS-01' to dns domain 'example.net'.

root@fs-01:~# net ads testjoin
Join is OK

root@fs-01:~# host fs-01
fs-01.example.net has address 192.168.56.51
```

Listing 12.2.1: record client

With this, you have added the first client to the domain. to have all services start correctly, the easiest way is to restart the client. It is not necessary to adjust the services here, as on the domain controllers.

Now test whether all users and groups from the domain are found. Use the command `wbinfo -u` and `wbinfo -g`. All users and groups will be displayed now.

However, if you try to give a user or a group on the client rights in the file system, this still fails. The local user management does not yet know the users and groups of the domain, for this you still need the adjustment from Listing 12.2.2 in the file `/etc/nsswitch.conf`:

```
root@fs-01:~# host fs-01
fs-01.example.net has address 192.168.56.51
```

Listing 12.2.2: adjustments to nsswitch.conf

NOTE! ☺

In some distributions, the customization is already done automatically by adding it to the domain .

Now you can use the users and groups to grant permissions, also the command `getent passwd <username>`, will now show you the user in passwd format.

This means that the client is now a member of the domain.

13 The file server

What makes a Linux client a fileserver? Just the provision of shares. After adding the Linux client *fs01* to the domain, you can now set up the shares. The shares should be usable later on a Windows client and in such a way that the user does not notice any difference to a Windows file server. The behavior when creating and editing files and directories shall be identical. The management of file system rights should be exactly the same as on a Windows file server.

To achieve exactly that, it is important to perform as many tasks as possible directly under Windows, above all, the assignment of permissions.

13.1 The administrative share

The first share you set up should always be an administrative share. In this share then, under Windows, the directories for the users are set up and provided with rights. Then the associated shares are created, which are then used by the users. This ensures that all permissions are exactly the same as on a Windows server.

Setting up the administrative share

For the share, create the `/admin` directory and set the owning group to *domain admins* and the permissions to *775*.

All shares are entered in `smb.conf`. The activation of Windows-compliant permissions are also set here. Listing 13.1.1 shows the modified `smb.conf`:

```
[global]
    workgroup = EXAMPLE
    realm = EXAMPLE.NET
    security = ADS
    winbind refresh tickets = yes
    winbind use default domain = yes
    template shell = /bin/bash
    idmap config * : range = 10000 - 19999
    idmap config EXAMPLE : backend = rid
    idmap config EXAMPLE : range = 10000000 - 19999999
    interfaces = 192.168.56.51
    bind interfaces only = yes
    vfs objects = acl_xattr
    inherit acls = yes
    acl group control = yes
    inherit owner = windows and unix

[admin-share]
    path=/admin
    read only = no
    browsable = no
    administrative share = yes
```

Listing 13.1.1: The administrative share

Four new parameters have been added to the *global* section:

- *vfs objects = acl_xattr*
The option ensures that Windows-compliant permissions can be set and that the behavior is similar to that of a Windows server. This also includes that the *administrator* cannot access a folder to which he has no permission.
- *inherit acls = yes*
With this option all ACLs are always inherited to all subfolders.
- *acl group control = yes*
Under Linux only the owning user can change permissions, under Windows however also a group can if it is owner of an entry. With this parameter Samba makes sure that this is also possible in a share.
- *inherit owner = windows and linux*
With this parameter you make sure that both the SID of the windows user and the UID of the linux user are entered as owner and that this information is also transferred to new entries.

NOTE! ☺

You can also write the four parameters into each share separately, then you can set up the handling of permissions in the shares differently. If you put the parameters in the global section of `smb.conf`, the values apply to all shares.

These are the only steps you take directly on the Samba server, all other directories you want to share afterwards you create under Windows, in the admin share, and assign the permissions there as well. Only this way ensures that the permissions correspond to those of a Windows server.

Each additional share now always points to a directory within the administrative share that was created under Windows and assigned permissions. For this reason, you can always keep the entries in `smb.conf` identical. Listing 13.1.2 shows the extract from `smb.conf` for an additional share:


```
[files]
    path=/admin/files
    read only = no
    browsable = no
```

Listing 13.1.2: additional-share

Users can now connect to the share. You can also provide shares through GPOs so that when a user logs in, the shares are automatically included. However, managing and setting up GPOs is no longer part of this tutorial.

13.2 Base directory of users

I would like to create one additional share here and that is the share for the *base directory* of users. The reason is that in the next section I will show you how to mount the shares most easily on a Linux client.

The directory to which the share should point should be named `/home/EXAMPLE`. The reason is that in the event that you later establish trusts with other domains, you can then provide a base directory for those users of the other domain as well, without possibly running into naming conflicts. In Listing 13.2.1 you can see all the commands required for this, including setting the permissions.

```
root@fs-01:~# mkdir /home/EXAMPLE
root@fs-01:~# chgrp "domain users" /home/EXAMPLE/
root@fs-01:~# chmod 775 /home/EXAMPLE/
```

Listing 13.2.1: create-base-directory

NOTE! ☺

Here in the tutorial you create the directories for the users by hand. In practice, you would manage this process via a GPO

All that's missing is the share. You can see the entry for the `smb.conf` in Listing 13.2.2:

```
[users]
    path=/home/EXAMPLE
    read only = no
    browsable = no
```

Listing 13.2.2: share for base directory

Now create a directory for one of your users in the directory `/home/EXAMPLE` and give the user all rights to his directory. You can see all the steps in Listing 13.2.3:

```
root@fs-01:~# getent passwd skania
skania:*:10001109:10000513::/home/EXAMPLE/skania:/bin/bash
```

```
root@fs-01:~# mkdir /home/EXAMPLE/skania

root@fs-01:~# chown skania /home/EXAMPLE/skania/

root@fs-01:~# chmod 770 /home/EXAMPLE/skania/
```

Listing 13.2.3: creating the directory for a user

You are now well prepared for the last section of the tutorial.

14 Linux-client and smb-mount

Since you are now already setting permissions on Windows and all permissions are the same as Windows permissions, it makes sense that users who may need to access the same dataset with Linux clients as Windows users should uniformly use the SMB protocol. Because if you mount the shares of your Samba server via *CIFS*, the rights will always be identical when changing or creating entries in the share. If you would now additionally set up *NFS* at this point and access the same dataset, the permissions would no longer be consistent.

But how to connect to the Samba server? The easiest way is if the Linux client is also a member of the Active Directory, then using the shares is quite easy.

One problem has to be considered especially when using SMB shares: Authentication must be performed to access a share. Not a good solution is that in the *fstab* user name and password are entered. It is better to use authentication that is already there anyway, namely Kerberos authentication. This is because whenever a user logs on to a Linux machine that is a member of the domain, the user also receives a Kerberos ticket with which to authenticate. This ticket can then be used by *libpam-mount* that the user authenticates to the file server and the share is also mounted immediately.

14.1 Setting up libpam-mount

First, make sure that the Vagrant VM *client01* becomes a member of the domain. A suitable *smb.conf* can be found in the */data* directory, copy the file to the */etc/samba* directory and add the machine to the domain, just as you did with the file server earlier. Remember the settings for the resolver.

The package *libpam-mount* is already installed on the client. Configure *libpam-mount* using the file */etc/security/pam_mount.conf.xml*. The part at the bottom is important. Create the needed entries before the line *</pam_mount>* at this point you now want the *files* and *users* share to be mounted automatically when a user logs in. To do this, you need the entries from Listing 14.1.1:

```
<volume
    fstype="cifs"
    server="fs-01.example.net"
    path="users/%(DOMAIN_USER)"
    mountpoint="/home/EXAMPLE/%(DOMAIN_USER)"
    sgrp="domain users"
    options="nodev,nosuid,sec=krb5,user=%(USER),cruid=%(USERUID),\
            workgroup=EXAMPLE,vers=3.1.1" />

<volume
    fstype="cifs"
    server="fs-01.example.net"
    path="files"
    mountpoint="/files"
    sgrp="domain users"
    options="nodev,nosuid,sec=krb5,user=%(USER),cruid=%(USERUID),\
            workgroup=EXAMPLE,vers=3.1.1" />
```

Listing 14.1.1: entries in the *pam_mount.conf.xml* file

The parameters have the following meanings:

- *user="%(DOMAIN_USER)"*
By using the parameter *user* you restrict the availability of the share to specific users -- in this case to all users of the domain.
- *"fstype="cifs"*
This sets the file system type -- in this case *cifs*.

- *server="fs-01.example.net"*
In this parameter, you specify the server where you have set up the share.
- *path="users%(DOMAIN_USER)"*
This is the share and the path within the share to mount. In the first example, the home directory of the corresponding user is always mounted. In the second example, only the name of the share is specified, since the entire share is to be mounted here.
- *mountpoint="/home/EXAMPLE/%(DOMAIN_USER)"*
Here you specify the mountpoint on the local system where the share should be mounted.
- *sgrp="domain users"* The parameter specifies which group the user must be a member of for this mount option to be processed. Here we have specified the group *domain users*, so the mountpoint will be created for every user existing in the Active Directory, but not for local users existing only on the client. This way you ensure that there are no error messages when logging in local users.
- *option="sec=krb5,cuid=%(USERID),workgroup=EXAMPLE,vers=3.1.1"*
The *option* parameter allows you to specify various mount parameters. Use the *sec=krb5* option to specify the security level during authentication (by default, *ntlmv1*). Since Samba 4 provides Kerberos, Kerberos should also be used for authentication here. Therefore, the parameter *sec=krb5* is set at this point. In addition, the parameter *cuid=%(USERID)* is required. This parameter ensures that the correct user realm is passed during Kerberos authentication. Without this parameter, Kerberos authentication will fail. Use the *workgroup=EXAMPLE* option to set the domain where the user is located.

IMPORTANT! ⚠

Do not forget the parameter *vers=3.1.1*. This parameter determines which SMB version is used to access the share. The default value here is SMBv1, this version is insecure and is also no longer supported by Samba, in the base installation

The complete file can be found in the `/data` directory on the Vagrant host *client01*.

You do not need to create or delete the *mountpoints*, this is done automatically by the line `<mkmountpoint enable="1" remove="true" >`

TIPP! ☺

If mounting does not work, you can change the line in the file `pam_mount.conf.xml` in the line *debug enable="0"* and then use `journalctl -f` to see in the log where the error occurs. It makes sense to set the debug level to 3

After logging in as domain user, you can find the messages from Listing 14.1.2 if you enter the command `mount`:

```
skania@client-01:~$ mount
...
//fs-01.example.net/users/skania on /home/EXAMPLE/skania type cifs\
(rw,nosuid,nodev,relatime,vers=3.1.1,sec=krb5,cuid=10001109,\
cache=strict,username=skania,domain=EXAMPLE,uid=10001109,forceuid\
gid=10000513,forcegid,addr=192.168.56.51,file_mode=0755,dir_mode=0755\
,soft,nounix,serverino,mapposix,rsize=4194304,wsz=4194304,\
bsize=1048576,echo_interval=60,actimeo=1,user=skania)

//fs-01.example.net/files on /files type cifs\
(rw,nosuid,nodev,relatime,vers=3.1.1,sec=krb5,cuid=10001109,\
cache=strict,username=skania,domain=EXAMPLE,uid=10001109,forceuid\
,gid=10000513,forcegid,addr=192.168.56.51,file_mode=0755,dir_mode=0755\
,soft,nounix,serverino,mapposix,rsize=4194304,wsz=4194304,\
bsize=1048576,echo_interval=60,actimeo=1,user=skania)
```

Listing 14.1.2: output of command "mount"

So now you can use the shares of the Samba server on any Linux client, with or without GUI.

15 What else works with Samba?

Everything cannot be covered in one day, besides the topics mentioned here, the following topics would be important for running a domain:

- administration of GPOs.
- Possibilities of migration, on the one hand of existing Windows domains and on the other hand the migration of old Samba 3 domains.
- Setting up trust relationships with other domains.
- Disaster recovery to be able to restore all objects as quickly as possible after a possible total failure of a domain.
- Use of Kerberos for authentication of services.
- Implementation of DHCP server with DDNS.
- Inclusion of network paper baskets and virus scanners.
- Using clusters with CTDB.
- setting up file system quotas.
- Server stored profiles along with folder redirection.
- A possible schema extension.

16 Conclusion

Of course, it is not possible to cover the entire spectrum of possibilities for a Samba domain in one day, but you have been given an overview of the possibilities here and can see that setting up a domain, file server, and clients can be done without much difficulty. The tutorial also shows that it does not matter to users whether there is a Samba or Windows domain in the background.

Index

- acl group control, 23
- acl_xattr, 23

- base directory, 24
- Bind9, 6, 7, 9
- bullseye-backports, 7

- Ceph, 7
- CIFS, 24
- cron, 20
- crontab, 20
- CTDB, 5, 7
- CUPS, 8

- DNS, 6
- DNS servers, 6
- domaincontroller, 6

- fileserver, 6, 22
- forest, 6
- forwarders, 9
- fqdn, 8
- fsmo, 18

- global Catalog, 6
- GlusterFS, 7
- GPO, 18
- group management, 13
- Group Policy, 18

- Heimdal-Kerberos, 6

- inherit owner, 23

- Kerberos, 6
- krb5.conf, 10

- LDAP, 6
- ldap, 13
- ldaps, 13
- libpam-mount, 25
- Linux client, 20

- MIT-Kerberos, 6
- mountpoint, 26

- named.conf.local, 9
- named.conf.options, 9
- nameserver, 14
- NETBIOS name, 21
- ntlm, 26
- ntp, 13
- ntp.conf, 13

- pam_mount.conf.xml, 25
- PdcEmulationMasterRole, 18
- print server, 7
- provisioning, 8

- realm, 21
- Remote Server Administration Tools, 13
- resolver, 11, 20
- RSAT, 13
- rsync, 18
- rsync server, 18
- rsyncd.conf, 18

- samba-tool, 8, 15
- srv-record, 14
- systemd, 18
- sysvol, 17

- tgt, 12
- tkey-gssapi-keytab, 9

- user management, 13