Seting up trusts between two Samba-domains

Author:
Stefan Kania

Ort:
SambaXP 2019 Göttingen

4. Januar 2019

Content

1 Introduction

In this years tutorial we will cover the trust-management between two Active Directory-domains. We will talk about the basics how Microsoft handles trusts and the different kind of trusts you can set up. After that, you will set up two domains, each with it's own namespace. You will set up a DNS-proxy with bind9 to manage the nameresolution of the SRV-records between these two domains. After the set up of the trusts we will take a look at the user- and group-management between those domains.

2 Basics about trusts

Before we start setting up the trust we will start with some basics. Let's talk about some general definitions first:

Trusting Domain

In the *trusting domain* ≪A≫ you can access the users and groups from the *trusted domain* ≪B≫. The users and domain-groups from domain ≪B≫ can be used by the administrator of domain ≪A≫ to give permissions to those users and groups to access resources in domain ≪A≫. If the trust is a *one-way-trust*, the administrator from domain ≪A≫ must authenticate with his credentials at domain ≪B≫. This means, the administrator from domain ≪B≫ must set up an account for the administrator from domain ≪A≫. This account will be used by the administrator from domain ≪A≫ to authenticate at domain ≪B≫ to get access to the users and groups from domain ≪B≫.

Trusted Domain

Users and domain-groups from the *trusted domain* ≪B≫ can be used by the *trusting domain* ≪A≫. The users from domain ≪B≫ will not see any of the users from domain ≪A≫ if the trusts is a *one-way-trust*.Users from domain ≪A≫ can not access any resources from domain ≪B≫.

One-Way-Trust

The *one-way-trust* is only set up in one direction either from domain ≪A≫ to domain ≪B≫ or the other way around. If domain ≪A≫ trusts domain ≪B≫, then domain ≪B≫ will not trust domain ≪A≫.

Two-Way-Trust

The *two-way-trust* will be set up in both directions, so all users and domain-groups from either domain can have access to resources from the other domain. The *two-way-trust* is the default trust in an Active Directory.

Transitive trust

If you have more then two domains, or a Active directory-tree, or an Active Directory-forest then Kerberos is used for authentication, you can set up a *transitive trust*. The advantage of a *transitive trust* is that Kerberos is managing the authentication between the trusts. All the clients from domain ≪A≫ will always send there authentication-request to the own domaincontroller (which is always a kerberos-server), the domaincontroller will then manage the tickets. If now a user from domain ≪A≫ will access a share on a fileserver in domain ≪B≫, the user will get a ticket from his own domaincontroller for domain ≪B≫. The fileserver in domain ≪B≫ will check the ticket against it's own Kerberos-server in domain ≪B≫. So the clients on both sides don't know anything about the trust.

In an Active Directory-infrastructure you will always have a *transitive two-way-trust* so all domains will trust each other.

3 Different kind of trusts

After we talked about the general definitions we will now take a look at the different type of trusts. These are all the trusts Microsoft mentioned in it'd documentation, that do not mean that Samba is supporting all these trusts. After talking about the different type of trusts we will talk a look at the trusts Samba is supporting at the moment and which restrictions Samba still has.

3.1 Domain trust

If you have a single tree of domains with a toplevel domain which handles the main name-space (for example *example.net*). Then all your domains will use a subdomain from your toplevel domain. Let's take a look at the example in figure3.1. Underneath your toplevel domain you have two more domains *dom1.example.net* and *dom2.example.net* In this case all three domains will trust each other in both directions. This is called *domain trust*.

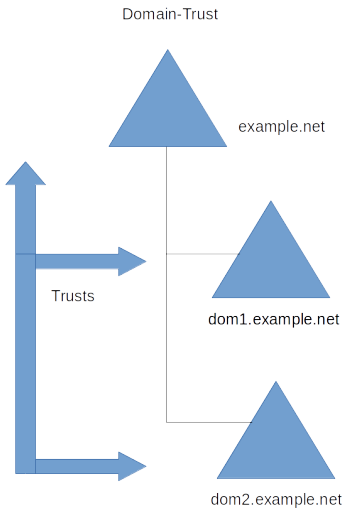

Abbildung 3.1: Domain trust

3.2 External trust

The *external trust* was first introduced with Windows-NT. The *external trust* is a trust between two domains, each with it's own namespace. In an external trust Kerberos is not used, so a single-sign-on is not possible. You can set up am *external trust* between two NT-style-domains or between am NT-style-domain and an Active directory-domain. In figure 3.2 you can see an example of an *external trust*:

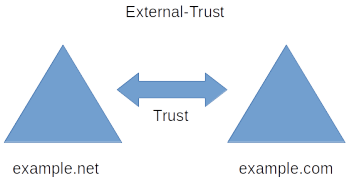

Abbildung 3.2: External trust

3.3 Forest trust

If you have two active dirctory-trees with different namespaces like *example.net* and *example.com* you can setup a *forest trust* between the two toplevel domains of your trees, then all the domain of both trees will trust each other. You do not have to setup a trust between every domain of the two trees, the trust will be managed over the toplevel domains. In figure 3.3 you can see a *forest trust*. As you can see the trust is managed over the toplevel domains all other trust will be established automatically. For a *forest trust* you need Kerberos to handle all the authentications between all the domains.

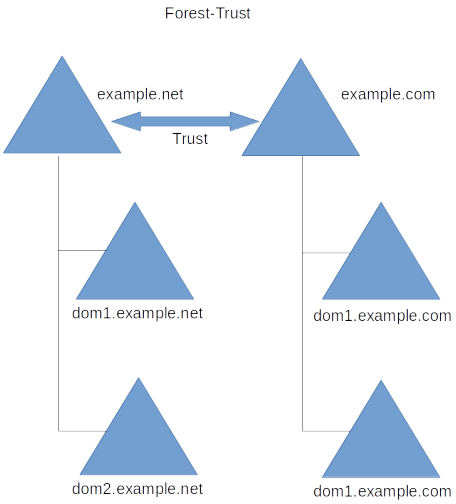

Abbildung 3.3: Forest trust

4 Samba and trusts

After we talked about the basics let's have look at the possibilities of Samba when it comes to trusts. As I mentioned before, not everything Microsoft explains is possible with Samba. So the most important question is: ≪What is already supported?≫

- *Forest trusts* between to forests as *two-way-trust* and *transitive trust*. This trust can be set up between two Samba-domains or a Samba-domain and a Windows-domain.
- *External trusts* between an active directory-domain and an NT-style-domain.
- Add users and groups of a *trusted domain* to groups of the *trusting domain*. But you must use the *SID* of the users and groups to add them to your group. It's not pssible to user the user- or group-name.
- Inside the *RSAT* you will see a *foreignSecurityPrincipal* for all added users and groups from a *trusted domain*. This is the way Microsoft shows that the user or group is part of a *trusting domain*.

A few things are still not possible if you want to use trusts together with Samba.

- Trusts between domains in the same tree with the same toplevel namespace.
- Filtering of SIDs to limit permissions.
- Both sides of the trust must give a full trust. That means, the administrator from domain ≪A≫ can manage all objects in domain ≪B≫ and vice versa.
- Selective authentication is not supported, at the moment. It's possible to create such a trust, but the *KDC* and *winbindd* will still ignore them.

5 The environment

Because the main interest in this tutorial is setting up the trust and managing the users and groups, I already prepared the domains for the tutorial. This year I used *Vagrant* to set up all the Linux-machines. Everyone of you will have four Linux-machines and one Windows-machine.

- Two Samba domaincontroller
  You will find two domaincontrollers, each with a different namespace. The domains are using the internal DNS-nameserver.
- One Samba-Linux-client
  This client will be a member in one of the domains to test the login and setting permissions.
- One Linux-system as DNS-proxy
  To set up a DNS-proxy for resolving the SRV-records between the two domains we will use *bind9*.
- One Windows-client
  This Windows-system will be a member in one of the domains to test the login.

We will use the Samba 4.9 packages from Louis van Belle. You can download the packages from his page *https://apt.van-belle.nl/*. The table 5.1 will give you an overview of all the parameters you need during the tutorial.

Parameter	domain 1	domain 2
DNS-suffix	s1.example.net	s2.exampl.net
Realm	S1.EXAMPLE.NET	S2.EXAMPLE.COM
NetBios-Name	S1	S2
IP-address	192.168.56.41	192.168.56.42
DNS-search	s1.example.net	s2.example.net

Tabelle 5.1: Information about the two domains

For all the Vagrant-machines there is a user *vagrant* with the password *vagrant*. The *root* also has the password *vagrant*.

6 Setting up a DNS-proxy

One of the most important things you have to do before you start setting up a trust is manage the nameresolution between the two domains. It's not enough to put the domaincontrollers in all /etc/hosts of all other domaincontrollers, because you must also be capable of resolving the SRV-records from all domains and all domaincontrollers. For this reason the easiest way to get nameresolution working is setting up a DNS-proxy between the two domains. The DNS-proxy will then forward the request between the to domains and to all external DNS-server to resolve any other hostname.

For the DNS-proxy we will use *bind9* on one of the Linux-machines. The packages you need are already installed and configured on the system with the IP-address *192.168.56.50*. To setup a DNS-proxy you have to edit the file /etc/bind/named.conf.option as you can see in listing 6.1:

```
forwarders {
       1.1.1.1;
       };
forward only;
dnssec-validation no;
dnssec-enable no;
allow-recursion { any; };
```

Listing 6.1: change in named.conf.options

After you set the options you have to configure the zone-forwarding in the file /etc/bind/named.conf.local as you can see in listing 6.2:

```
zone "s1.example.net" in {
type forward;
forwarders { 192.168.56.41; };
};

zone "s2.example.com" in {
type forward;
forwarders { 192.168.56.42; };
};
```

Listing 6.2: Changes in named.conf.local

As you can see inside the /etc/samba/smb.conf of both domincontollers the DNS-proxy is the forwarder for both domaincontrollers. Now you can test if you can resolve the SRV-records from both domains on both domaincontrollers.

In listing 6.3 you can see the result of all the commands:

```
vagrant@addc-01:~$ host -t srv _kerberos._tcp.s1.example.net
_kerberos._tcp.s1.example.net has SRV record 0 100 88 addc-01.s1.example.net.

vagrant@addc-01:~$ host -t srv _kerberos._tcp.s2.example.com
_kerberos._tcp.s2.example.com has SRV record 0 100 88 addc-02.s2.example.com.

vagrant@addc-02:~$ host -t srv _kerberos._tcp.s1.example.net
_kerberos._tcp.s1.example.net has SRV record 0 100 88 addc-01.s1.example.net.

vagrant@addc-02:~$ host -t srv _kerberos._tcp.s2.example.com
_kerberos._tcp.s2.example.com has SRV record 0 100 88 addc-02.s2.example.com.
```

Listing 6.3: Test the SRV-records

The next test you should do is trying to get a Kerberos-ticket from either domain. You can see an example in listing 6.4.

Important!

Remember writing the realm in capital letters

```
vagrant@addc-02:~$ kinit administrator@S2.EXAMPLE.COM
administrator@S2.EXAMPLE.COM's Password:

vagrant@addc-02:~$ klist
Credentials cache: FILE:/tmp/krb5cc_1000
       Principal: administrator@S2.EXAMPLE.COM

  Issued Expires Principal
Jan 2 18:19:28 2019 Jan 3 04:19:28 2019 krbtgt/S2.EXAMPLE.COM@S2.EXAMPLE.COM

vagrant@addc-02:~$ kinit administrator@S1.EXAMPLE.NET
administrator@S1.EXAMPLE.NET's Password:

vagrant@addc-02:~$ klist
Credentials cache: FILE:/tmp/krb5cc_1000
       Principal: administrator@S1.EXAMPLE.NET

  Issued Expires Principal
Jan 2 18:20:00 2019 Jan 3 04:20:00 2019 krbtgt/S1.EXAMPLE.NET@S1.EXAMPLE.NET
```

Listing 6.4: Testing the Kerberos

You can do the test from both domaincontrollers in both domains. After these tests you can be sure that the DNS-proxy is running.

7 Setting up the trust

Now that you did all the preparations and all the test passes you can start setting up the trust with `samba-tool`. You can set up the trust from any of the domaincontroller in any of domain. In my example in listing 7.1 I will do it from the domaincontroller *addc-01.s1.example.net*:

```
root@addc-01:/home/vagrant# samba-tool domain trust create s2 \
                          --type=forest --direction=both \
                          --create-location=both \
```

```
                          -U administrator@S2.EXAMPLE.COM
LocalDomain Netbios[S1] DNS[s1.example.net] SID[S-1-5-21-3126357314-\
                          3577825812-2501707040]
RemoteDC Netbios[ADDC-02] DNS[addc-02.s2.example.com] ServerType[PDC,\
                          GC,LDAP,DS,KDC,TIMESERV,CLOSEST,WRITABLE,\
                          GOOD_TIMESERV,FULL_SECRET_DOMAIN_6]
Password for [administrator@S2.EXAMPLE.COM]:
RemoteDomain Netbios[S2] DNS[s2.example.com] SID[S-1-5-21-2406301074\
                          -2875553281-1464783146]
Creating remote TDO.
Remote TDO created.
Setting supported encryption types on remote TDO.
Creating local TDO.
Local TDO created
Setting supported encryption types on local TDO.
Setup local forest trust information...
Namespaces[2] TDO[s2.example.com]:
TLN: Status[Enabled] DNS[*.s2.example.com]
DOM: Status[Enabled] DNS[s2.example.com] Netbios[S2] \
                       SID[S-1-5-21-2406301074-2875553281-1464783146]
Setup remote forest trust information...
Namespaces[2] TDO[s1.example.net]:
TLN: Status[Enabled] DNS[*.s1.example.net]
DOM: Status[Enabled] DNS[s1.example.net] Netbios[S1] \
                       SID[S-1-5-21-3126357314-3577825812-2501707040]
Validating outgoing trust...
OK: LocalValidation: DC[\\addc-02.s2.example.com] CONNECTION[WERR_OK] \
                       TRUST[WERR_OK] VERIFY_STATUS_RETURNED
Validating incoming trust...
OK: RemoteValidation: DC[\\addc-01.s1.example.net] CONNECTION[WERR_OK] \
                       TRUST[WERR_OK] VERIFY_STATUS_RETURNED
Success.
```

Listing 7.1: Setting up the trust

Remember to use capital letter for the realm of the administrator from domain *s2*. With command in listing 7.1 you created a *forest-trust*. Remember a *forest-trust* is always bidirectional and transitive.

Testing the trust

After you set up the trust you should test if everything went right. So the first test will be to show the trust between our two domains. In listing 7.2 you will see the command and the result:

```
root@addc-01:/home/vagrant# samba-tool domain trust show s2
LocalDomain Netbios[S1] DNS[s1.example.net] SID[S-1-5-21-3126357314-\
              3577825812-2501707040]
TrustedDomain:

NetbiosName: S2
DnsName: s2.example.com
SID: S-1-5-21-2406301074-2875553281-1464783146
Type: 0x2 (UPLEVEL)
Direction: 0x3 (BOTH)
Attributes: 0x8 (FOREST_TRANSITIVE)
PosixOffset: 0x00000000 (0)
```

```
kerb_EncTypes: 0x18 (AES128_CTS_HMAC_SHA1_96,AES256_CTS_HMAC_SHA1_96)
Namespaces[2] TDO[s2.example.com]:
TLN: Status[Enabled] DNS[*.s2.example.com]
DOM: Status[Enabled] DNS[s2.example.com] Netbios[S2] \
              SID[S-1-5-21-2406301074-2875553281-1464783146]
```

Listing 7.2: Testing the trust

Repeat the test from the other domaincotroller the result should be the same. Because you can have more then one trust the next test in listing 7.3 will show you, how you can list all trusts from or to your domain:

```
root@addc-01:/home/vagrant# samba-tool domain trust list
Type[Forest] Transitive[Yes] Direction[BOTH] Name[s2.example.com]
```

Listing 7.3: List all trusts

In different domains you can have different results. The result depends on the trust you have established with this domain.

If you, after setting up the trust, having problems with assigning users from a trusting domain to your domain then you should test if the trust is still valid. In listing 7.4 you can see the test to check if the trust is still valid:

```
root@addc-01:/home/vagrant# samba-tool domain trust validate s2 \
                          -Uadministrator@S2.EXAMPLE.COM
LocalDomain Netbios[S1] DNS[s1.example.net] SID[S-1-5-21-3126357314-\
                          3577825812-2501707040]
LocalTDO Netbios[S2] DNS[s2.example.com] SID[S-1-5-21-2406301074-\
                          2875553281-1464783146]
OK: LocalValidation: DC[\\addc-02.s2.example.com] CONNECTION[WERR_OK] \
                          TRUST[WERR_OK] VERIFY_STATUS_RETURNED
OK: LocalRediscover: DC[\\addc-02.s2.example.com] CONNECTION[WERR_OK]
RemoteDC Netbios[ADDC-02] DNS[addc-02.s2.example.com] ServerType[PDC,GC,\
                          LDAP,DS,KDC,TIMESERV,CLOSEST,WRITABLE,\
                          GOOD_TIMESERV,FULL_SECRET_DOMAIN_6]
Password for [administrator@S2.EXAMPLE.COM]:
OK: RemoteValidation: DC[\\addc-01.s1.example.net] CONNECTION[WERR_OK] \
                          TRUST[WERR_OK] VERIFY_STATUS_RETURNED
OK: RemoteRediscover: DC[\\addc-01.s1.example.net] CONNECTION[WERR_OK]
```

Listing 7.4: Validation of the trust

Now you can be sure that the trust is working. You can do this test from any domaincontroller in both domains.

8 Managing users and groups

Now we are at the point were we can assign users and groups from a trusting domain to a group of the trusted domain. Remember, here we have a two-way-trust so yo can assign users and groups in both directions. Before we can assign users and groups you must create some users and groups in both domains. Just create a few users and groups with `samba-tool`.

Listing users and groups

Now that the trust is working, you can list all users and groups from both domains with `wbinfo -<u|g> --domain=<domain>` as you can see in listing 8.1 you will not see the users and groups from the trusting domain, you see only the users and groups from your domain:

```
root@addc-02:/home/vagrant# wbinfo -u --domain=s2
S2\administrator
S2\guest
S2\krbtgt
S2\user1
S2\user2
S2\user3
S2\user4
root@addc-02:/home/vagrant# wbinfo -u --domain=s1

root@addc-02:/home/vagrant# wbinfo -g --domain=s2
S2\cert publishers
S2\ras and ias servers
S2\allowed rodc password replication group
S2\denied rodc password replication group
S2\dnsadmins
S2\enterprise read-only domain controllers
S2\domain admins
S2\domain users
S2\domain guests
S2\domain computers
S2\domain controllers
S2\schema admins
S2\enterprise admins
S2\group policy creator owners
S2\read-only domain controllers
S2\dnsupdateproxy
S2\dom2-g1
S2\dom2-g2

root@addc-02:/home/vagrant# wbinfo -g --domain=s1
```

Listing 8.1: List users and groups with wbinfo

The reason is, maybe you will not see all users and groups from the trusting domain, because of some security settings (especially if the trusting domain is a Windows-domain), so the command was disabled for domaincontroller in the actual Samba-version. To see all users you can do an LDAP-query with `samba-tool` as you can see in listing 8.2:

```
root@addc-02:/home/vagrant# samba-tool user list -H ldap://addc-01 \
                          -U administrator@S1.EXAMPLE.NET
Password for [administrator@S1.EXAMPLE.NET]:
Guest
hhirsch
ktom
Administrator
krbtgt
adent
ptau

root@addc-02:/home/vagrant# samba-tool user list -H ldap://addc-02 \
                          -U administrator@S2.EXAMPLE.COM
Password for [administrator@S2.EXAMPLE.COM]:
```

```
user1
user2
Guest
user4
Administrator
user3
krbtgt
```

Listing 8.2: List users via LDAP

Here you can see that it's still possible to see all users and groups from both domains.

Using wbinfo

After you are able to list all users from both domains you need to find the *SID* from a user or group add the *SID* as a member to your group. The first step with `wbinfo` is to get some more information from your domains as you can see in listing 8.3:

```
root@addc-02:/home/vagrant# wbinfo --all-domains
BUILTIN
S2
S1

root@addc-02:/home/vagrant# wbinfo --own-domain
S2

root@addc-02:/home/vagrant# wbinfo --trusted-domains
BUILTIN
S2
S1

root@addc-02:/home/vagrant# wbinfo --online-status
BUILTIN : active connection
S2 : active connection
S1 : active connection
```

Listing 8.3: Using wbinfo

Now lets see how you can get the *SID* from the users and groups. In listing 8.4 you see all necessary command:

```
root@addc-02:/home/vagrant# wbinfo -n s1\\ktom
S-1-5-21-3126357314-3577825812-2501707040-1104 SID_USER (1)

root@addc-02:/home/vagrant# wbinfo -n s2\\user1
S-1-5-21-2406301074-2875553281-1464783146-1104 SID_USER (1)

root@addc-02:/home/vagrant# wbinfo -n s1\\dom1-g1
S-1-5-21-3126357314-3577825812-2501707040-1108 SID_DOM_GROUP (2)

root@addc-02:/home/vagrant# wbinfo -n s2\\dom2-g1
S-1-5-21-2406301074-2875553281-1464783146-1108 SID_DOM_GROUP (2)

root@addc-02:/home/vagrant# wbinfo -i s1\\hhirsch
S1\hhirsch:*:3000018:3000019::/home/S1/hhirsch:/bin/false

root@addc-02:/home/vagrant# wbinfo -i s2\\user2
```

```
S2\user2:*:3000020:100::/home/S2/user2:/bin/false
```

Listing 8.4: Show the SID of users and groups

As you can see, you will get all information of all users from both domains.

8.1 Testing the authentication

With `wbinfo` you can test the authentication process of the different users from both domains.

You will see two types of authentication. The first test will be the *plaintext password authentication*. This type of authentication is always taking place when a user logs in local to the system. *plaintext* do's not mean that the password will be send without encryption it's just the name for the login process. The second type is the *challange/response password authentication*. This type of authentication is using NTLM or Kerberos. In listing 8.1.1 you will see the result of both authentication processes:

```
root@addc-02:/home/vagrant# wbinfo -a s1\\hhirsch
Enter s1\hhirsch's password:
plaintext password authentication succeeded
Enter s1\hhirsch's password:
challenge/response password authentication succeeded

root@addc-02:/home/vagrant# wbinfo -a s2\\user1
Enter s2\user1's password:
plaintext password authentication succeeded
Enter s2\user1's password:
challenge/response password authentication succeeded
```

Listing 8.1.1: Testing the authentication

You can also test which domaincontrollers are responsible for the authentication. Listing 8.1.2 shows the result of the test:

```
root@addc-02:/home/vagrant# wbinfo --ping-dc
checking the NETLOGON for domain[S2] dc connection to "addc-02.s2.example.com" \
            succeeded

root@addc-02:/home/vagrant# wbinfo --ping-dc --domain=s1
checking the NETLOGON for domain[s1] dc connection to "addc-01.s1.example.net" \
            succeeded
```

Listing 8.1.2: Testing the domaincontrollers

Assigning user and groups

Now you have reached the point were you can assign users and groups from the *trusted domain* to any of the groups of the *trusting domain*. As I mentioned before you can't assign the users and groups via the name, you must use the SID. Listing 8.1.3 is showing all steps:

```
root@addc-02:/home/vagrant# wbinfo -n s1\\ktom
S-1-5-21-3126357314-3577825812-2501707040-1104 SID_USER (1)

root@addc-02:/home/vagrant# samba-tool group addmembers dom2-g1 \
          S-1-5-21-3126357314-3577825812-2501707040-1104
```

```
Added members to group dom2-g1

root@addc-02:/home/vagrant# wbinfo -n s1\\dom1-g1
S-1-5-21-3126357314-3577825812-2501707040-1108 SID_DOM_GROUP (2)

root@addc-02:/home/vagrant# samba-tool group addmembers dom2-g1 \
          S-1-5-21-3126357314-3577825812-2501707040-1108
Added members to group dom2-g1

root@addc-02:/home/vagrant# samba-tool group listmembers dom2-g1
S-1-5-21-3126357314-3577825812-2501707040-1108
user2
S-1-5-21-3126357314-3577825812-2501707040-1104
user1
```

Listing 8.1.3: Assigning members to a group

The groupmembership is valid on all members in the trusted domain, so if you have a fileserver as a member you can assign the group and giving permissions to the group.

8.2 Looking at the trust with Windows

After you did all the tests directly on the domaincontroller you can take a look how Windows is showing the trust in the *RSAT*. In figure 8.1 you will see the trust in `Active Directory domains and trusts`:

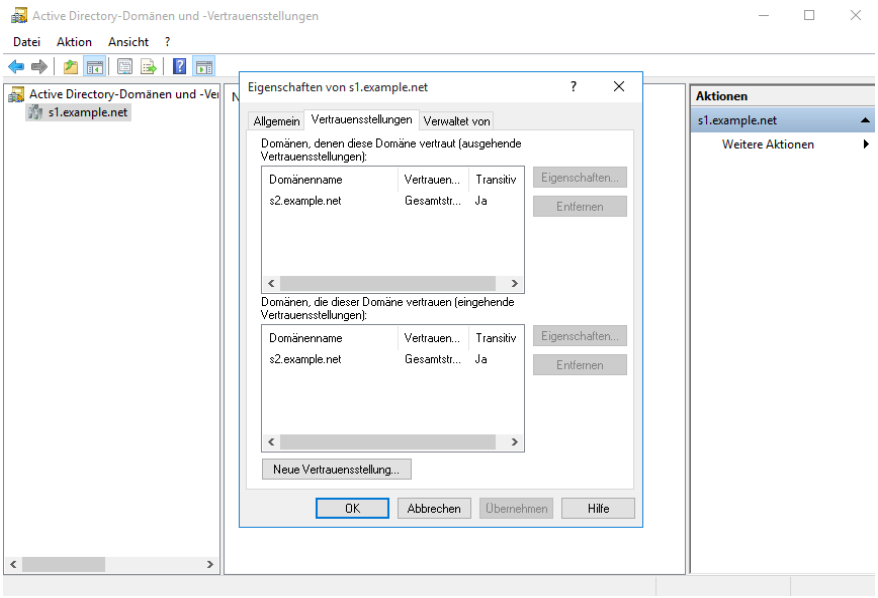

Abbildung 8.1: Trust management

The second figure 8.2 is showing how you see the groupmembership in the *ADUC* of the *RSAT*.

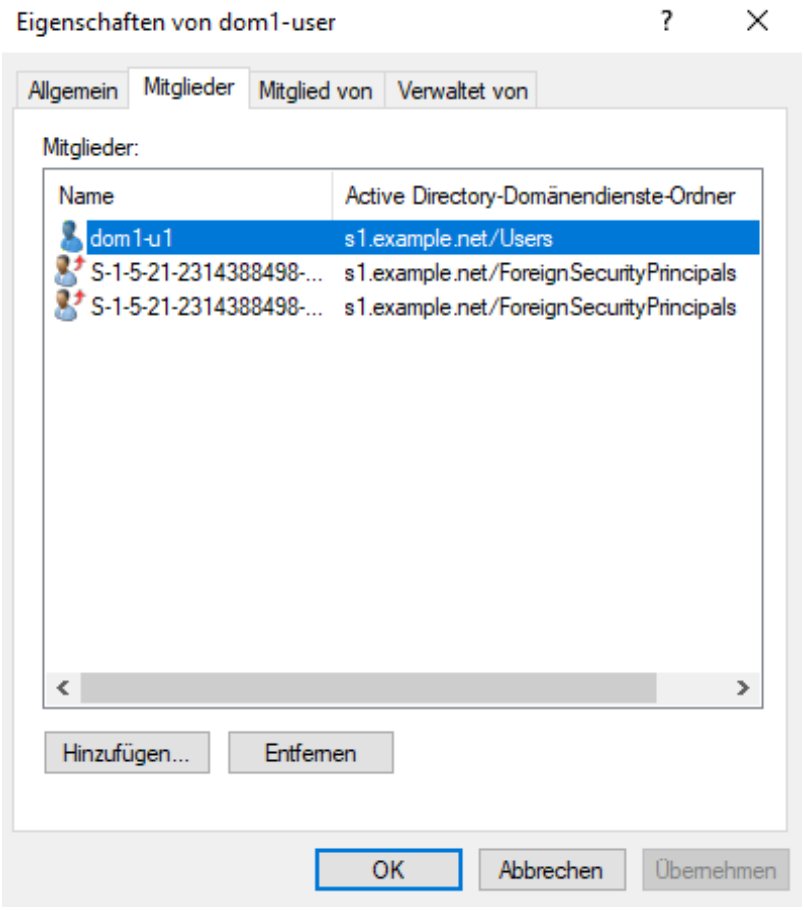

Abbildung 8.2: Showing the groupmembership

9 Using the trusts on a Linux-clients

After setting up the trust now we will take a look at a Linux-client and how you have to configure the client to use the users and groups from both domains. If you want to use users from both domains you must configure *winbind* on all your clients to be able to resolve all users and groups from both domains.

Joining the Linux-Client

Before you can join the client you must configure the client via the smb.conf. You must set up an ID-mapping for both domains in your smb.conf as shown in listing 9.1:

```
[global]
      workgroup = s1
      realm = S1.EXAMPLE.NET
      security = ADS
      winbind refresh tickets = Yes
      template shell = /bin/bash
      idmap config * : range = 10000 - 19999
      idmap config S1 : backend = rid
      idmap config S1 : range = 1000000 - 1999999
      idmap config S2 : backend = rid
```

```
    idmap config S2 : range = 10000000 - 19999999
```

Listing 9.1: smb.conf for both domains

Now you can join the Linux-client to the domain. On your Linux-machine the smb.conf is already prepared to join the machine to domain *s1*. So you can join the machine as in listing 9.2:

```
root@linux-client:/home/vagrant# net ads join -U administrator
Enter administrator's password:
Using short domain name -- S1
Joined 'LINUX-CLIENT' to dns domain 's1.example.net'

root@linux-client:/home/vagrant# net ads testjoin
Join is OK
```

Listing 9.2: Joining the Linux-client

After restarting the *smbd*, *nmbd* and *winbind* you can test if you can get the users from both domains with `wbinfo`. In listing 9.3 you see a few tests with `wbinfo` on the client:

```
root@linux-client:/home/vagrant# wbinfo -m
BUILTIN
LINUX-CLIENT
S1
S2

root@linux-client:/home/vagrant# wbinfo --online-status
BUILTIN : online
LINUX-CLIENT : online
S1 : online
S2 : online

root@linux-client:/home/vagrant# net rpc trustdom list -Uadministrator
Enter administrator's password:
Trusted domains list:

S2 S-1-5-21-2406301074-2875553281-1464783146

Trusting domains list:

S2 S-1-5-21-2406301074-2875553281-1464783146

root@linux-client:/home/vagrant# wbinfo -n s1\\ktom
S-1-5-21-3126357314-3577825812-2501707040-1104 SID_USER (1)

root@linux-client:/home/vagrant# wbinfo -n s2\\user1
S-1-5-21-2406301074-2875553281-1464783146-1104 SID_USER (1)
```

Listing 9.3: Tests with wbinfo

If you want to see all users from either domain, on a Linux-client you can still use the command `wbinfo -<u|g> --domain=<domainname>`.

Using users and groups

After you have joined the machine and you are able to see all the users and groups, you must edit the file /etc/nsswitch.conf as it is shown in listing 9.4:

```
passwd: compat winbind
group: compat winbind
```

Listing 9.4: Canging nsswitch.conf

After changing the setting in nsswich.conf you can test the users and groups with `getent` as you can see it in listing 9.5:

```
root@linux-client:/home/vagrant# getent group s1\\dom1-g1
S1\dom1-g1:x:1001108:

root@linux-client:/home/vagrant# getent group s2\\dom2-g1
S2\dom2-g1:x:10001108:

root@linux-client:/home/vagrant# getent passwd s1\\ktom
S1\ktom:*:1001104:1000513:ktom:/home/S1/ktom:/bin/bash

root@linux-client:/home/vagrant# getent passwd s2\\user1
S2\user1:*:10001104:10000513:user1:/home/S2/user1:/bin/bash
```

Listing 9.5: Testing users and groups with getent

If you can see your users and groups from both domains, you can start setting permissions inside the filesystem. I created some home-directories for users from the different domains to test the login. Listing 9.6 is chowing a few examples:

```
root@linux-client:/# mkdir /data-dom1

root@linux-client:/# mkdir /data-dom2

root@linux-client:/# chgrp s1\\dom1-g1 /data-dom1

root@linux-client:/# chgrp s2\\dom2-g1 /data-dom2

root@linux-client:/# chown s1\\ktom /data-dom1

root@linux-client:/# chown s2\\user2 /data-dom2

root@linux-client:/# ls -ld /data-dom1
drwxr-xr-x 2 S1\ktom S1\dom1-g1 4096 Jan 4 16:32 /data-dom1

root@linux-client:/# ls -ld /data-dom2
drwxr-xr-x 2 S2\user2 S2\dom2-g1 4096 Jan 4 16:32 /data-dom2


root@linux-client:/# mkdir /home/S1

root@linux-client:/# mkdir /home/S2

root@linux-client:/# chmod 755 /home/S*

root@linux-client:/# mkdir /home/S1/ktom

root@linux-client:/# chown s1\\ktom /home/S1/ktom/

root@linux-client:/# mkdir /home/S2/user1

root@linux-client:/# chown s2\\user1 /home/S2/user1
```

Listing 9.6: Setting permissions

The last test will be to see if you can login to the Linux-client with users from both domains. In listing 9.7 you will see the test:

```
vagrant@linux-client:~$ ssh s1\\ktom@192.168.56.51
s1\ktom@192.168.56.51's password:
...
Last login: Fri Jan 4 16:34:52 2019 from 192.168.56.51
S1\ktom@linux-client:~$

vagrant@linux-client:~$ ssh s2\\user1@192.168.56.51
s2\user1@192.168.56.51's password:
...
Last login: Fri Jan 4 16:42:09 2019 from 192.168.56.51
S2\user1@linux-client:~$
```

Listing 9.7: Testing the login

Now you can start creating some shares and use your system as fileserver for both domains.

10 Conclusion

With Samba 4.9 it is possible to not only setup a trust between active directory-domains, but also adding users and groups from a *trusting domain* to a *trusted domain*. So now you can use the trust in a productive environment, but remember there a still a few thing not working, especially that it's not possible to take permissions from the administrator. Both administrators of a trust can manage both domains.

Index