univention
be open.

Univention Corporate Server - Extended domain services documentation

Release 5.0

Sep 13, 2023

CONTENTS

CHAPTER

ONE

INTEGRATION OF UBUNTU CLIENTS INTO A UCS DOMAIN

Univention Corporate Server allows the integration of Ubuntu clients. Initially a standard Ubuntu installation needs to be performed. The following section describes the configuration changes, which need to be made to integrate the Ubuntu client into the UCS domain. After successful integration users can authenticate on the Ubuntu clients with their standard UCS domain password and username.

This configuration has been tested with Ubuntu 14.04 LTS, Ubuntu 16.04 LTS as well as Kubuntu 14.04 LTS.

Caution: In case a command fails or does not return the expected output, please make sure that all configuration options and files are entered and have been written as shown in this document. For example, some text editors do not preserve the indentation which is required for some configuration files.

1.1 Integration into the LDAP directory and the SSL certificate authority

After Ubuntu has been installed, some of it's configuration files need to be modified. To simplify the setup, the default configuration of the UCS Primary Directory Node should be copied to the Ubuntu system, for example:

```
# Become root
$ sudo bash <<"EOF"

# Set the IP address of the UCS Primary Directory Node, 192.0.2.3 in this example
export PRIMARY_DIRECTORY_NODE_IP=192.0.2.3

mkdir /etc/univention
ssh -n root@${PRIMARY_DIRECTORY_NODE_IP} 'ucr shell | grep -v ^hostname=' >/etc/
↪univention/ucr_primary_directory_node
echo "primary_directory_node_ip=${PRIMARY_DIRECTORY_NODE_IP}" >>/etc/univention/
↪ucr_primary_directory_node
chmod 660 /etc/univention/ucr_primary_directory_node
. /etc/univention/ucr_primary_directory_node

echo "${PRIMARY_DIRECTORY_NODE_IP} ${ldap_master}" >>/etc/hosts

EOF
```

By default UCS only authenticated users can search in the LDAP directory. As such, the Ubuntu client needs an account in the UCS domain to gain read access to the LDAP directory:

```
# Become root
$ sudo bash

$ . /etc/univention/ucr_primary_directory_node
```

(continues on next page)

(continued from previous page)

```
# Download the SSL certificate
$ mkdir -p /etc/univention/ssl/ucsCA/
$ wget -O /etc/univention/ssl/ucsCA/CAcert.pem \
    http://${ldap_master}/ucs-root-ca.crt

# Create an account and save the password
$ password="$(tr -dc A-Za-z0-9_ </dev/urandom | head -c20)"
$ ssh -n root@${ldap_master} udm computers/ubuntu create \
    --position "cn=computers,${ldap_base}" \
    --set name=$(hostname) --set password="${password}" \
    --set operatingSystem="$(lsb_release -is)" \
    --set operatingSystemVersion="$(lsb_release -rs)"
$ printf '%s' "$password" >/etc/ldap.secret
$ chmod 0400 /etc/ldap.secret

# Create ldap.conf
$ cat >/etc/ldap/ldap.conf <<__EOF__
TLS_CACERT /etc/univention/ssl/ucsCA/CAcert.pem
URI ldap://$ldap_master:7389
BASE $ldap_base
__EOF__
```

1.2 Configuration of the System Security Services Daemon (SSSD)

SSSD provides a set of daemons to manage access to remote directories and authentication mechanisms.

```
# Become root
$ sudo bash

$ . /etc/univention/ucr_primary_directory_node

# Install SSSD based configuration
$ DEBIAN_FRONTEND=noninteractive apt-get -y install sssd libnss-sss libpam-sss␣
↪libsss-sudo

# Create sssd.conf
$ cat >/etc/sssd/sssd.conf <<__EOF__
[sssd]
config_file_version = 2
reconnection_retries = 3
sbus_timeout = 30
services = nss, pam, sudo
domains = $kerberos_realm

[nss]
reconnection_retries = 3

[pam]
reconnection_retries = 3

[domain/$kerberos_realm]
auth_provider = krb5
krb5_kdcip = ${primary_directory_node_ip}
krb5_realm = ${kerberos_realm}
krb5_server = ${ldap_master}
krb5_kpasswd = ${ldap_master}
id_provider = ldap
ldap_uri = ldap://${ldap_master}:7389
```

(continues on next page)

(continued from previous page)

```
ldap_search_base = ${ldap_base}
ldap_tls_reqcert = never
ldap_tls_cacert = /etc/univention/ssl/ucsCA/CAcert.pem
cache_credentials = true
enumerate = true
ldap_default_bind_dn = cn=$(hostname),cn=computers,${ldap_base}
ldap_default_authtok_type = password
ldap_default_authtok = $(cat /etc/ldap.secret)
__EOF__
$ chmod 600 /etc/sssd/sssd.conf

# Install auth-client-config
$ DEBIAN_FRONTEND=noninteractive apt-get -y install auth-client-config

# Create an auth config profile for sssd
$ cat >/etc/auth-client-config/profile.d/sss <<__EOF__
[sss]
nss_passwd=   passwd:   compat sss
nss_group=    group:    compat sss
nss_shadow=   shadow:   compat
nss_netgroup= netgroup: nis

pam_auth=
        auth [success=3 default=ignore] pam_unix.so nullok_secure try_first_pass
        auth requisite pam_succeed_if.so uid >= 500 quiet
        auth [success=1 default=ignore] pam_sss.so use_first_pass
        auth requisite pam_deny.so
        auth required pam_permit.so

pam_account=
        account required pam_unix.so
        account sufficient pam_localuser.so
        account sufficient pam_succeed_if.so uid < 500 quiet
        account [default=bad success=ok user_unknown=ignore] pam_sss.so
        account required pam_permit.so

pam_password=
        password requisite pam_pwquality.so retry=3
        password sufficient pam_unix.so obscure sha512
        password sufficient pam_sss.so use_authtok
        password required pam_deny.so

pam_session=
        session required pam_mkhomedir.so skel=/etc/skel/ umask=0077
        session optional pam_keyinit.so revoke
        session required pam_limits.so
        session [success=1 default=ignore] pam_sss.so
        session required pam_unix.so
__EOF__
$ auth-client-config -a -p sss

# Restart sssd
$ service sssd restart
```

The commands **getent passwd** and **getent group** should now also display all users and groups of the UCS domain.

1.3 Configuring user logins

The home directory of a user should be created automatically during login:

```
# Become root
$ sudo bash

$ cat >/usr/share/pam-configs/ucs_mkhomedir <<__EOF__
Name: activate mkhomedir
Default: yes
Priority: 900
Session-Type: Additional
Session:
    required    pam_mkhomedir.so umask=0022 skel=/etc/skel
__EOF__

$ DEBIAN_FRONTEND=noninteractive pam-auth-update --force
```

During login users should also be added to some system groups:

```
# Become root
$ sudo bash

$ echo '*;*;*;Al0000-2400;audio,cdrom,dialout,floppy,plugdev,adm' \
   >>/etc/security/group.conf

$ cat >>/usr/share/pam-configs/local_groups <<__EOF__
Name: activate /etc/security/group.conf
Default: yes
Priority: 900
Auth-Type: Primary
Auth:
    required    pam_group.so use_first_pass
__EOF__

$ DEBIAN_FRONTEND=noninteractive pam-auth-update --force
```

By default the Ubuntu login manager only displays a list of local users during login. After adding the following lines an arbitrary user name can be used:

```
# Become root
$ sudo bash

# Add a field for a user name, disable user selection at the login screen
$ mkdir /etc/lightdm/lightdm.conf.d
$ cat >>/etc/lightdm/lightdm.conf.d/99-show-manual-userlogin.conf <<__EOF__
[SeatDefaults]
greeter-show-manual-login=true
greeter-hide-users=true
__EOF__
```

Kubuntu 14.04 uses `AccountService`, a D-Bus interface for user account management, which ignores the `/etc/lightdm.conf` file. Since there is no configuration file for `AccountService` the login theme needs to be changed to *classic* under *System Settings ▸ Login Screen (LightDM)*.

With these settings the login for domain members should be possible after a restart of LightDM or a reboot.

1.4 Kerberos integration

Every UCS domain provides a Kerberos domain. Since Kerberos relies on DNS, the Ubuntu client should use a UCS Directory Node (Primary Directory Node, Backup Directory Node or Replica Directory Node) as its DNS server. The following steps provide an example configuration for Kerberos:

```
# Become root
$ sudo bash

$ . /etc/univention/ucr_primary_directory_node

# Install required packages
$ DEBIAN_FRONTEND=noninteractive apt-get install -y heimdal-clients ntpdate

# Default krb5.conf
$ cat >/etc/krb5.conf <<__EOF__
[libdefaults]
    default_realm = $kerberos_realm
    kdc_timesync = 1
    ccache_type = 4
    forwardable = true
    proxiable = true
    default_tkt_enctypes = arcfour-hmac-md5 des-cbc-md5 des3-hmac-sha1 des-cbc-crc␣
↪des-cbc-md4 des3-cbc-sha1 aes128-cts-hmac-sha1-96 aes256-cts-hmac-sha1-96
    permitted_enctypes = des3-hmac-sha1 des-cbc-crc des-cbc-md4 des-cbc-md5 des3-
↪cbc-sha1 arcfour-hmac-md5 aes128-cts-hmac-sha1-96 aes256-cts-hmac-sha1-96
    allow_weak_crypto=true

[realms]
$kerberos_realm = {
   kdc = $primary_directory_node_ip $ldap_master
   admin_server = $primary_directory_node_ip $ldap_master
   kpasswd_server = $primary_directory_node_ip $ldap_master
}
__EOF__

# Synchronize the time with the UCS system
$ ntpdate -bu $ldap_master

# Test Kerberos: kinit will ask you for a ticket and the SSH login to the Primary␣
↪Directory Node should work with ticket authentication:
$ kinit Administrator
$ ssh -n Administrator@$ldap_master ls /etc/univention

# Destroy the kerberos ticket
$ kdestroy
```

1.5 Limitations of the Ubuntu domain integration

It is currently not possible to change the user password at the LightDM login manager. Instead, the password can be changed via the **kpasswd** command after login or via the UMC module *Change password*.

1.6 Additional references

- https://help.ubuntu.com/community/LDAPClientAuthentication
- https://help.ubuntu.com/community/SingleSignOn
- https://help.ubuntu.com/community/PamCcredsHowto
- http://labs.opinsys.com/blog/2010/03/26/user-management-with-sssd-on-shared-laptops/

CHAPTER

TWO

INTEGRATION OF LINUX/UNIX SYSTEMS INTO A UCS DOMAIN

These are general instructions for the integration of Unix/Linux-based non-UCS systems - referred to in the following simply as Unix systems - in the trust context of the UCS domain.

The integration of Ubuntu clients is documented with example step-by-step instructions in *Integration of Ubuntu clients into a UCS domain* (page 1).

The integration of macOS clients is documented with example step-by-step instructions[1] in the UCS manual. macOS systems use a deviating domain integration based on Samba 4.

Not all integration steps need to be performed. In this way, for example, a Unix system can merely be integrated in the IP management and access the NTP server without integrating the system in the UCS user management (e.g., if it is a database server on which no user login is performed anyway).

2.1 Managing the systems in the Univention Management Console

A *Computer: Linux* object can be created in the UMC computer management. This allows the integration of the Unix system in the DNS/DHCP and network administration of the Univention Management Console

If the Nagios support is enabled under *[Options]*, remote Nagios checks can also be applied against the system.

2.2 Configuration of the name resolution

The Unix system should use a name server from the UCS domain: All UCS Directory Nodes (i.e., Primary Directory Node, Backup Directory Node and Replica Directory Node) operate a DNS server. One or more of these UCS system should be entered in the `/etc/resolv.conf`, e.g.:

```
domain example.com
nameserver 192.0.2.08
nameserver 192.0.2.9
```

2.3 Configuration of the time server

All UCS Directory Nodes (i.e., Primary Directory Node, Backup Directory Node and Replica Directory Node) operate a NTP server.

The configuration differs depending on the NTP software used, but is set under `/etc/ntp.conf` on most Linux systems, e.g.:

```
server primary.example.com
server backup.example.com
```

[1] https://docs.software-univention.de/manual/5.0/en/domain-ldap/domain-join.html#macos-domain-join

2.4 Access to user and group information of the UCS domain

The *Name Service Switch* (NSS) is an interface for configuring the data sources for users, groups and computers. NSS is present on all Linux versions and most Unix systems.

If the Unix system used provides support for an NSS module for LDAP access - as is the case in most Linux distributions - user and group information can be read out of the UCS LDAP directory.

The configuration files of the NSS LDAP module differ depending on the Linux/Unix version.

As a general rule, the following settings must be set there:

- The DN of the LDAP base of the UCS domain (saved in the Univention Configuration Registry Variable `ldap/base` on UCS servers) needs to be configured on the system.
- The LDAP server, ports and authentication credentials to be used. The fully qualified domain names of one or more UCS Directory Nodes should be entered here. By default UCS LDAP servers only allow authenticated LDAP access.
- In the standard setting, only TLS-secured access is possible on UCS-LDAP servers. The accessing Unix system must therefore use the root certificate of the UCS-CA. The certificate can be found on the Primary Directory Node in the file `/etc/univention/ssl/ucsCA/CAcert.pem` and can be copied into any directory, e.g., `/etc/ucs-ssl/`. The UCS root certificate must then be configured in the LDAP configuration files. If the Unix system uses OpenLDAP as the LDAP implementation, it is usually the file `/etc/openldap/ldap.conf` or `/etc/ldap/ldap.conf`. The line for OpenLDAP is as follows:

```
TLS_CACERT /etc/ucs-ssl/CAcert.pem
```

If the NSS LDAP service has been set up correctly, the following two commands should output all users and groups:

```
getent passwd
getent group
```

2.5 Integrating into Kerberos

UCS employs the Kerberos implementation Heimdal. For this reason, Heimdal should also be used to access the Kerberos realm on the Unix system. Only the Heimdal client libraries need to be installed on the Unix system.

Kerberos requires correct time synchronization, see *Configuration of the name resolution* (page 7).

The configuration is performed in the `/etc/krb5.conf` file on most systems. Here is an example configuration:

- `KERBEROSREALM` must be replaced by the name of the UCS Kerberos realm (saved in the Univention Configuration Registry Variable `kerberos/realm`[2]).
- `PRIMARYIP` must be replaced by the IP address of the Primary Directory Node.
- `PRIMARYFQDN` must be replaced by the fully qualified domain name of the Primary Directory Node.

```
[libdefaults]
    default_realm = KERBEROSREALM
    default_tkt_enctypes = arcfour-hmac-md5 des-cbc-md5 des3-hmac-sha1 \
       des-cbc-crc des-cbc-md4 des3-cbc-sha1 aes128-cts-hmac-sha1-96   \
       aes256-cts-hmac-sha1-96
    permitted_enctypes = des3-hmac-sha1 des-cbc-crc des-cbc-md4 \
       des-cbc-md5 des3-cbc-sha1 arcfour-hmac-md5               \
       aes128-cts-hmac-sha1-96 aes256-cts-hmac-sha1-96
    allow_weak_crypto=true
    kdc_timesync = 1
    ccache_type = 4
```

(continues on next page)

[2] https://docs.software-univention.de/manual/5.0/en/appendix/variables.html#envvar-kerberos-realm

(continued from previous page)

```
    forwardable = true
    proxiable = true

[realms]
KERBEROSREALM = {
   kdc = PRIMARYIP PRIMARYFQDN
   admin_server = PRIMARYIP PRIMARYFQDN
   kpasswd_server = PRIMARYIP PRIMARYFQDN
}
```

The Heimdal PAM module then needs to be installed. In general, the installation of the module should adapt the PAM configuration automatically.

Then Kerberos authentication during login should work via PAM and password changes should be possible via **kpasswd**.

To allow SSH logins via Kerberos, the options `GSSAPIAuthentication` and `GSSAPIKeyExchange` should be set to `yes` in the configuration file of the SSH daemon (typically `/etc/ssh/sshd_config`).

2.6 Accessing a UCS print server

UCS uses the *Common Unix Printing System* (CUPS) to implement print services. The Unix system can use the UCS print servers by installing the CUPS client programs. In addition the CUPS server needs to be configured for the clients, typically in the configuration file `/etc/cups/client.conf`, e.g.:

```
ServerName printserver.example.com
```

CHAPTER

THREE

ADVANCED SSL CERTIFICATE HANDLING

3.1 Managing additional certificates with `univention-certificate`

Every UCS domain has its own SSL certificate authority. The SSL certificates are created automatically for all UCS systems during the installation (Primary Directory Node) or during the domain join (all other system roles).

The command **`univention-certificate`** can be used to manage these certificates, e.g., if it proves necessary to create a certificate for the integration of an external system. The command is executed as `root` on the Primary Directory Node.

3.1.1 Storage of the certificates

The certificates are stored in the directory `/etc/univention/ssl/` on the Primary Directory Node and synchronized on all Backup Directory Node systems. A subdirectory with the name of the certificate is kept in the directory `/etc/univention/ssl/` for every certificate, which contains the following files:

`req.pem`
: This file contains the original request with which the certificate was created.

`openssl.cnf`
: This file contains the OpenSSL configuration at the time the certificate was created.

`cert.pem`
: The file represents the actual certificate.

`private.key`
: The file contains the private key for the certificate.

3.1.2 Displaying the certificates

The following command is used to display a list of all the available, valid certificates:

```
$ univention-certificate list
```

An individual SSL certificate can be displayed with the following command:

```
$ univention-certificate dump -name fullyqualifiedhostname
```

3.1.3 Checking the validity of a certificate

This command checks whether a certificate is valid or invalid:

```
$ univention-certificate check -name fullyqualifiedhostname
```

A certificate may be invalid because it has either been revoked or has expired.

3.1.4 Revoking a certificate

The following command is used to revoke a certificate:

```
$ univention-certificate revoke -name fullyqualifiedhostname
```

It is then no longer valid, but remains stored in the file system. Certificates of UMC computer objects do not need to be revoked manually.

3.1.5 Creating a certificate

The following command can be used to create a new certificate:

```
$ univention-certificate new -name fullyqualifiedhostname
```

The fully qualified domain name of the computer should be given as the name. By default the certificate is valid for five years. The standard value can be changed by setting the Univention Configuration Registry Variable `ssl/default/days`.

3.2 Signing of certificate signing requests by the UCS certificate authority

A certificate signing request (CSR) is a request submitted to a certificate authority (CA) to create a digital signature. A CSR typically occurs in the form of a file. This section describes how a CSR is signed by the UCS CA.

CERTIFICATE
: is the file name of the certificate to be created.

REQUEST
: is the file with the CSR in either PEM or DER format. A file in PEM format is a text file containing a base64 encoded block enclosed between `BEGIN CERTIFICATE` and `END CERTIFICATE`. A request in binary DER format must be first converted to the PEM format with the following command:

```
$ openssl req \
  -inform  der -in  request.der \
  -outform pem -out req.pem
```

The following command then processes the CSR and creates the certificate:

```
$ openssl ca -batch -config /etc/univention/ssl/openssl.cnf \
  -in REQUEST -out CERTIFICATE \
  -passin file:/etc/univention/ssl/password
```

CHAPTER

FOUR

CONNECTING AN EXTERNAL OPENLDAP SERVER VIA SYNCREPL

This chapter describes the read-only integration of an external OpenLDAP server via a **syncrepl** proxy. This allows the external system to access the LDAP data of the UCS domain without being a member of the domain itself. This guide principally applies to any Unix system with OpenLDAP. The guide has been tested with Debian 7 Wheezy. Syncrepl is part of OpenLDAP starting with version 2.2.

The external OpenLDAP server is described as `extldap.univention.test` below and synchronizes with the Primary Directory Node, which uses the LDAP base `dc=univention,dc=test`.

The following steps must be run on the OpenLDAP system and the UCS system as the `root` user.

4.1 Creating a computer account

For `extldap.univention.test`, a *Linux* computer object must be created in the Univention Management Console computer management and a DNS forward and reverse zone assigned to the computer.

4.2 Activation of syncrepl on the Primary Directory Node

Now a syncrepl proxy needs to be set up on the Primary Directory Node. The required configuration files are downloaded from https://updates.software-univention.de/download/syncrepl/ucs5-syncrepl-proxy-setup.tar.bz2 as a TAR archive.

The downloaded archive must firstly be extracted on the Primary Directory Node:

```
$ tar -xvf ucs4-syncrepl-proxy-setup.tar.bz2
```

The subdirectory `UCS_Primary_Directory_Node` contains two Univention Configuration Registry sub-file templates for the LDAP server configuration file (`/etc/ldap/slapd.conf`). Sub-files are a mechanism in Univention Configuration Registry which can be used to generate a configuration file from several individual templates. More detailed information can be found in the UCS manual. The two sub-files are now copied into the template directory:

```
$ mv UCS_Primary_Directory_Node/8*.conf /etc/univention/templates/files/etc/ldap/
↪slapd.conf.d/
$ mv UCS_Primary_Directory_Node/syncrepl-proxy.conf /etc/univention/templates/
↪files/etc/ldap/
```

The info file now needs to be copied. It registers the sub-file templates and the Univention Configuration Registry variables used:

```
$ mv UCS_Primary_Directory_Node/syncrepl-proxy.info /etc/univention/templates/info/
```

Then the `slapd.conf` is regenerated from the template:

```
$ ucr commit /etc/ldap/slapd.conf
$ ucr commit /etc/ldap/syncrepl-proxy.conf
```

4.3 Initial transfer of the LDAP data

Now an initial copy of the UCS data is created and transferred to the external system. In addition, an initial configuration file for the OpenLDAP service is copied onto the external system (`slapd.conf`).

```
$ slapcat -f /etc/ldap/slapd.conf > data.ldif
$ cat remote_system/template-slapd.conf | ucr filter > remote_system/slapd.conf
$ scp remote_system/slapd.conf data.ldif extldap.univention.test:
$ rm data.ldif
```

The LDAP schema data and the SSL certificates from the UCS Primary Directory Node are now passed to the external LDAP server:

```
$ rsync -aR /usr/share/univention-ldap/schema extldap.univention.test:/
$ rsync -aR /var/lib/univention-ldap/local-schema extldap.univention.test:/
$ rsync -aR /etc/univention/ssl/extldap.univention.test extldap.univention.test:/
$ rsync -aR /etc/univention/ssl/ucsCA/CAcert.pem extldap.univention.test:/
```

4.4 Configuration of the LDAP service on the third-party system

The configuration of the external LDAP server is now adapted. It must be noted that only a minimal `slapd.conf` is installed here, which should be expanded with local adaptations as necessary:

```
$ systemctl stop slapd
$ cp /etc/ldap/slapd.conf /root/backup-slapd.conf
$ cp /root/slapd.conf /etc/ldap
```

A number of settings now need to be adapted in the provided `/etc/ldap/slapd.conf` template:

- `extldap.univention.test` must be replaced with the fully qualified domain name of the external LDAP server
- `dc=univention,dc=test` must be replaced with the LDAP base actually used
- `REMOTE_UPDATE_PASSWORD` must be replaced with the password used to access the LDAP database

4.5 Importing the initial LDAP copy

The initial copy of the UCS directory data is now imported and the LDAP server restarted. The file permissions of the `/var/lib/ldap/` directory and the `/etc/ldap/slapd.conf` file differ depending on the Linux/Unix version:

```
$ mkdir /root/ldap_backup_dir
$ mv /var/lib/ldap/*.* /root/ldap_backup_dir
$ slapadd -f /etc/ldap/slapd.conf -l /root/data.ldif
$ chown openldap.openldap /var/lib/ldap/*.*
$ chgrp openldap /etc/ldap/slapd.conf
$ chgrp -R openldap /etc/univention/ssl
$ systemctl start slapd
```

The configuration of the external LDAP server is now complete. The following command (performed on the Primary Directory Node) can be used to check whether the external LDAP server can be reached via the `LDAPS` protocol:

```
$ ldapsearch -x -H ldaps://extldap.univention.test -b cn=Subschema -s base
```

Whenever schema files are added on the UCS Primary Directory Node, the following steps have to be repeated. First an updated `slapd.conf` needs to be generated for the remote LDAP server which includes all UCS schema files. Then all required files need to be copied to the remote LDAP server:

```
$ cat remote_system/template-slapd.conf | ucr filter > remote_system/slapd.conf
$ scp remote_system/slapd.conf extldap.univention.test:
$ rsync -aR /usr/share/univention-ldap/schema extldap.univention.test:/
$ rsync -aR /var/lib/univention-ldap/local-schema extldap.univention.test:/
```

And after that the following steps need to be repeated on the external LDAP server:

```
$ systemctl stop slapd
$ cp /etc/ldap/slapd.conf /root/backup-slapd.conf
$ cp /root/slapd.conf /etc/ldap
$ chgrp openldap /etc/ldap/slapd.conf
$ systemctl start slapd
```

If the external system is a Debian system, the `SLAPD_SERVICES` variable may need to be adapted in `/etc/default/slapd`. In addition, the `SLAPD_CONF` variable can be used to specify the `/etc/ldap/slapd.conf` file as the configuration file for the `slapd`, if this is not the standard for the OpenLDAP version used.

4.6 Activation of the syncrepl proxy

If the LDAP connection works, the configuration of the syncrepl proxy can be activated on the Primary Directory Node. This is done by saving the `REMOTE_UPDATE_PASSWORD` password configured above in the `/etc/replica-001.secret` file and entering the address of the external LDAP server in the form of a LDAP-URI in the Univention Configuration Registry Variable `ldap/replica/target/uri`:

```
$ echo -n 'REMOTE_UPDATE_PASSWORD' >/etc/replica-001.secret
$ chmod 600 /etc/replica-001.secret
$ ucr set ldap/replica/target/uri=ldaps://extldap.univention.test/
$ ucr commit /etc/ldap/syncrepl-proxy.conf
$ systemctl restart slapd
```

If several systems are connected, the corresponding LDAP-URIs can be entered in the variable separated with commas and additional replica password files created. The number in the name of the password files is incremented by one for each additional system.

The replication originates from the Primary Directory Node and is performed via `LDAPS` to the host name of the external LDAP server system. This requires working name resolution (typically via DNS). The host name must be specified as a fully qualified domain name to allow checking of the SSL certificate.

To allow convenient LDAP search via **`ldapsearch -x expression`** on the external LDAP server the file `/etc/ldap/ldap.conf` may be configured like this:

```
TLS_CACERT /etc/univention/ssl/ucsCA/CAcert.pem
HOST FQDN
BASE LDAPBASE
```

4.7 Testing the replication

The replication via **syncrepl** can be tested by changing the description of an existing user for example. When an LDAP search is performed on the external server, the changed description should then be displayed.

INDEX

E

K

L

S