Univention Corporate Server - Extended domain services documentation

Release 5.2

Feb 05, 2025

CONTENTS

1 Integration of Linux/Unix systems into a UCS domain **1**

1.1 Managing the systems in the Univention Management Console 1

1.2 Configuration of the name resolution 1

1.3 Configuration of the time server 1

1.4 Access to user and group information of the UCS domain 2

1.5 Integrating into Kerberos 2

1.6 Accessing a UCS print server 3

2 Advanced SSL certificate handling **5**

2.1 Managing additional certificates with `univention-certificate` 5

2.2 Signing of certificate signing requests by the UCS certificate authority 6

Index **7**

CHAPTER

ONE

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.

For the integration of Ubuntu clients, Univention offers a dedicated tool. For more information, see Ubuntu domain joins[1].

The integration of macOS clients is documented with example step-by-step instructions[2] 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).

1.1 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.

1.2 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
```

1.3 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/5.2/en/domain-ldap/domain-join.html#ubuntu-domain-join

[2] https://docs.software-univention.de/manual/5.2/en/domain-ldap/domain-join.html#macos-domain-join

1.4 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
```

1.5 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 1).

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`[3]).
- `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)

[3] https://docs.software-univention.de/manual/5.2/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).

1.6 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

TWO

ADVANCED SSL CERTIFICATE HANDLING

2.1 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.

2.1.1 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.

2.1.2 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
```

2.1.3 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.

2.1.4 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.

2.1.5 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`.

2.2 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
```

INDEX

E

environment variable
- kerberos/realm, 2
- ldap/base, 2
- ssl/default/days, 6

K

kerberos/realm, 2

L

ldap/base, 2

S

ssl/default/days, 6