

AD/Kerberosanbindung KDE-NEON/Ubuntuclients inkl. Loginvertrauen

Getestet mit UCS 4.4

Der Join selbst

Bevor man beginnt muss man eine bestehende LDAP-Anbindung deaktivieren und alle Pakete mit PURGE deinstallieren und den [Univention Domain-Join Client](#) installieren.

```
apt remove nscd nslcd ldap-auth-client ldap-auth-config --purge && apt
autormove --purge
```

Bei /etc/nsswitch.conf die LDAP Einträge entfernen.

ACHTUNG: Manuelle Einträge in den Pamconfigs werden beim Domainjoin überschrieben. Deshalb Einträge danach setzen.

```
add-apt-repository ppa:univention-dev/ppa
DEBIAN_FRONTEND=noninteractive apt-get install univention-domain-join
```

Der nächste Part hier scheint mittlerweile behoben zu sein. Die Resolvconf kann bleiben wie sie ist. Ich lass das hier mal zur Info noch so stehen. DNSMASQ darf keiner laufen.

Der Join mit Ubuntu/KDE-NEON funktioniert nicht out of the Box. Die Resolvconf muss mit einer anderen Funktion ihrer selbst verlinkt werden. Natürlich kann man beim Join auch die MasterIP angeben. Dann funktioniert der Join. Nur geht dann meist der Login nicht, und ein SSO am Ubuntu ist nicht möglich. Auch darf kein DNSmasq laufen.

```
rm /etc/resolv.conf
ln -s /run/systemd/resolve/resolv.conf /etc/.
```

Hier den DNSmasq Eintrag auskommentieren:

```
nano /etc/NetworkManager/NetworkManager.conf
```

Networkmanager neu starten. Ob man richtig am DNS hängt kann man ganz einfach mit folgenden Commando testen:

```
host -la $(dnsdomainname)
```

oder eben direkt:

```
host -la meinedomain.wtf
```

Weiters muss noch in der Datei /etc/lsb-release die Zeile DISTRIB_ID=neon zu

DISTRIB_ID=Ubuntu umgeschrieben werden. Danach geht auch der Join normal.

Loginvertrauen

Bindet man einen Linuxclient per LDAP an kann man alle Benutzer mit einem Realm bestimmen. Somit können sich auch nur diese User auf der Maschine einloggen. Bedient man sich aber der Kerberosanbindung, können sich alle User in der Domäne auf Clients einloggen. Dies ist sehr unschön. Für Ubuntu/Linuxclients die also die volle Domänenanbindung mit Kerberos und SSO genießen, bedient man sich ganz einfach PAM.

Zuerst erstellt man sich ein File:

```
nano /etc/security/access-login.conf
```

Mit folgendem Inhalt:

[access-login.conf](#)

```
# Warning: This file is auto-generated and might be overwritten by
#          univention-config-registry.
#          Please edit the following file(s) instead:
# Warnung: Diese Datei wurde automatisch generiert und kann durch
#          univention-config-registry ueberschrieben werden.
#          Bitte bearbeiten Sie an Stelle dessen die folgende(n)
Datei(en):
#
#          /etc/univention/templates/files/etc/security/access-login.conf
#

+:Administrators,root,Domain Admins,firmen-mitarbeiter-
user,backuppc:ALL
- :ALL:ALL
```

Das heißt nur Mitglieder von den genannten Gruppen Administrators, root, Domain Admins, firmen-mitarbeiter-user dürfen sich einloggen. Die nächste Zeile besagt „alles andere verweigern“. Dies gilt für alle Services, also auch IMAP, SSH, usw. Man kann auch einzelne Services verwenden. Da diese Config aber nur Clientmaschinen betrifft, sollte das so genügen. Diese Config wird dann mit Foreman ausgerollt. Gut. Also nächstes wird das File noch in der Datei /etc/pam.d/common-account definiert. Hierfür fügen wir diese eine Zeile ein.

[common-account](#)

```
#
# /etc/pam.d/common-account - authorization settings common to all
services
#
# This file is included from other service-specific PAM config files,
```

```
# and should contain a list of the authorization modules that define
# the central access policy for use on the system. The default is to
# only deny service to users whose accounts are expired in /etc/shadow.
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by
default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.
#
# here are the per-package modules (the "Primary" block)
account [success=1 new_authtok_reqd=done default=ignore]
pam_unix.so
# here's the fallback if no module succeeds
account requisite pam_deny.so
# prime the stack with a positive return value if there isn't one
already;
# this avoids us returning an error just because nothing sets a success
code
# since the modules above will each just jump around
account required pam_permit.so
# and here are more per-package modules (the "Additional" block)
account sufficient pam_localuser.so

account required pam_access.so accessfile=/etc/security/access-
login.conf listsep=, maxent=0x400001

account [default=bad success=ok user_unknown=ignore] pam_sss.so
# end of pam-auth-update config
```

Ab dem Zeitpunkt ist die Config gültig.

Sicherheit HOME

Möchte man das ein Home mit sämtlichen Daten von „/etc/skel“ beim Ersten Login angelegt wird muss man Pam konfigurieren.

```
nano /etc/pam.d/common-session
...
session required pam_mkhome.so skel=/etc/skel umask=0007
...
```

Hier ist drauf zu achten dass, das ganze and der richtigen Stelle steht. Beispiel Datei mit UCS Anbindung:

[common-session](#)

```
#
# /etc/pam.d/common-session - session-related modules common to all
# services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define tasks to be
# performed
# at the start and end of sessions of *any* kind (both interactive and
# non-interactive).
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by
# default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
session [default=1]                pam_permit.so
# here's the fallback if no module succeeds
session requisite                   pam_deny.so
# prime the stack with a positive return value if there isn't one
# already;
# this avoids us returning an error just because nothing sets a success
# code
# since the modules above will each just jump around
session required                   pam_permit.so
# The pam_umask module will set the umask according to the system
# default in
# /etc/login.defs and user settings, solving the problem of different
# umask settings with different shells, display managers, remote
# sessions etc.
# See "man pam_umask".
session optional                   pam_umask.so
# and here are more per-package modules (the "Additional" block)
session required                   pam_mkhome.so umask=0007 skel=/etc/skel
session required                   pam_unix.so
session optional                   pam_sss.so
session optional                   pam_mount.so
session optional                   pam_systemd.so
# end of pam-auth-update config
```

Somit wird das Home wie folgt mit dem Benutzerprofil von /etc/skel erstellt:

```
ls -l /home
insgesamt 36
drwx----- 18 backuppc backuppc          4096 Sep 20  2018 backuppc/
drwxrwx---  45 peter      peter-maingroup 4096 Mai 20 10:41 peter/
drwxrwx---  22 hans       hans-maingroup  4096 Jun 14 10:19 hans/
```

drwxrwx--- 24 susi susi-maingroup 4096 Jun 14 12:54 susi/

From:

<https://deepdoc.at/dokuwiki/> - DEEPDOC.AT - enjoy your brain

Permanent link:

https://deepdoc.at/dokuwiki/doku.php?id=prebuilt_systems:ucs:ad_kerberosanbindung_ubuntuclients_inkl_loginvertrauen&rev=1569088556

Last update: 2025/11/29 22:06

