R|R|Z|N

Linux im Active Directory und als Terminal-Server

Robert Euhus

euhus@rrzn.uni-hannover.de

Ziel: Linux-Terminal-Server mit folgenden Eigenschaften:

- Nutzer aus dem bestehenden Active Directory (AD)
- Zugriff auf die Daten auf Shares im AD
- grafischer Linux-Desktop
- Einloggen von beliebigen Clients aus dem Netz

Einsatzgebiete

- Remote-Zugriff auf beschränkte Ressourcen, ohne gleich Vollzugriff via VPN zu gewähren
- Compute-Server unter Linux/Unix in einer Windows-(AD)-Umgebung nicht mit nur einem geteilten Account nutzen
- einzelne Linux-Desktops im AD-Umfeld (Verwaltung und Backup im AD)

Vorgehen

- Einbindung eines Linux-Rechners ins AD
- Bereitstellen der Terminal-Dienste

Eine ausführliche Dokumentation des hier beschriebenen Setups befindet sich auf der RRZN-Webseite mit Installationsanleitungen (ohne Support):
http://www.rrzn.uni-hannover.de/anleitungen.html

1 Linux im AD:

- Active Directory
- Linux
- Winbind

2 NX-Terminalserver:

- X-Window System
- NX-Protokoll
- FreeNX im Usermode

3 Schlussbemerkungen:

- Fazit
- Links/Literatur

Authentifizierung

Authentifizieren: Bestätigen der Identität eines Benutzers, Rechners oder Dienstes

- Kerberos ist seit Windows 2000 Standard für Authentifizierung im AD
- Beim Start eines Domänen-Mitgliedes (Client) baut die *Local Security Authority (LSA)* eine Verbindung zum Domaincontroller auf und authentifiziert sich als Maschine
- Benutzerauthentifizierungen laufen danach über diese Verbindung.

Das ältere NTLMv2 wird im Active Directory zwar noch unterstützt, jedoch hat Kerberos einige Vorteile:

- es ist ein offener, lang erprobter Standard
- alle Dienste des AD, wie z.B. das Einhängen der Shares, sind ohne erneute Authentifizierung nutzbar.

Kerberos: Begriffe

Realm separate Kerberos-Verwaltungseinheit, das ist hier die AD-Domäne; stets in Großbuchstaben.

Prinzipal Mitglied der Realm (Benutzer, Rechner oder Dienst), besteht aus:

- Bezeichner der Form: `<Name>/<Instanz>@<Realm>`
  z.B.

  `aduser@ADTEST.INTERN`

  `aduser/admin@ADTEST.INTERN`

  `host/relogin.adtest.intern@ADTEST.INTERN`

  `HTTP/www.adtest.intern@ADTEST.INTERN`
- kryptografischer Schlüssel zum Verschlüsseln von Tickets; wird als Hash aus dem Passwort generiert
- weitere Felder sind in der Spezifikation vorgesehen und werden vom AD benutzt, um Autorisierungsinformationen zu verteilen. Sie sind aber für unseren Anwendungsfall irrelevant.

Kerberos: Begriffe

KDC *(Key Distribution Center)* zentraler Authentifizierungsdienst von Kerberos, zuständig für die Vergabe von →Tickets

Ticket Identitätsnachweis für einen Prinzipal zum Zugriff auf einen anderen Prinzipal

TGT *(Ticket Granting Ticket)* spezielles Ticket für einen Prinzipal, mit dem automatisch weitere Tickets angefordert werden können. „Ausweis“

Kerberos: Funktionsweise und Eigenschaften

- der Benutzer authentifiziert sich nur einmal beim Key Distribution Center (KDC) und erhält von diesem ein Ticket Granting Ticket (TGT).
- die weitere Authentifizierung erfolgt automatisch ohne Interaktion des Anwenders
- will der Benutzer auf einen Dienst (oder Rechner) zugreifen, so wird automatisch für diesen Dienst ein Ticket beim KDC besorgt, mit dem der Benutzer gegenüber dem Dienst authentifiziert wird

$\Rightarrow$ ermöglicht Single Sign On (SSO) für den Nutzer

- auch die Authentifizierung von Diensten und Rechnern gegenüber Benutzern lässt sich erzwingen

Kerberos: Voraussetzungen

- Zeitstempel sind sicherheitsrelevanter Bestandteil des Kerberos-Protokolls
  $\Rightarrow$ synchrone Zeit auf den beteiligten Rechner ist Voraussetzung
  $\longrightarrow$ gemeinsamen Zeitserver nutzen (i.Allg. Domaincontroller)
- Informationen für die Kerberos-Nutzung werden über DNS Service Recource Records (SRV-RRs) verteilt
  $\longrightarrow$ DNS-Server des AD nutzen (i.Allg. auch der Domaincontroller)

nsswitch – Name Service Switch

Frühere Unix-artige Systeme verwalteten Benutzerinformationen und andere Konfigurationsdaten ausschließlich in einigen festgelegten Dateien, wie z.B.:

`/etc/passwd`	unter anderem: Benutzername, uid, primäre gid, Home-Verzeichnis, Shell
`/etc/group`	Gruppenname, gid, zusätzliche Nutzer in der Gruppe
`/etc/hosts`	Auflösung IP $\leftrightarrow$ Rechnername

Bei heutigen Linux-Systemen erlaubt der *Name Service Switch (nsswitch)* die Bereitstellung dieser Daten aus verschiedenen Quellen, inklusive der zuvor genannten Dateien. Die Konfiguration erfolgt in `/etc/nsswitch.conf`.

PAM – Pluggable Authentication Modules

- eine Softwarebibliothek, die eine API für Authentifizierungsdienste zur Verfügung stellt:
  - account — Account-Verifikation (Darf der Nutzer diesen Dienst nutzen?)
  - auth — Authentifizierung (i.Allg. Passwortabfrage)
  - password — Passwortänderung
  - session — Sitzungsverwaltung (Öffnen und Schließen vorbereiten)
- in den Konfigurationsdateien unter `/etc/pam.d` werden einzelnen Diensten (ssh, login, imap,...) die zu nutzenden Module zugeordnet.

Vorsicht bei Änderungen an der PAM-Konfiguration!

Beim Bearbeiten der PAM-Konfiguration immer größte Vorsicht walten lassen: stets Backups anlegen und an mindestens einer Stelle als root eingeloggt bleiben, bis sichergestellt (getestet) ist, dass man sich auch wieder als root einloggen kann!

Winbind – Die Brücke zwischen den Welten

Der Winbind-Dienst ist Teil des Samba-Projektes; er übernimmt immer mehr die Rolle der LSA[1] unter Windows:

- hält eine wiederverwendbare Verbindung zum DC offen und bietet seine Dienste lokal an; darüber läuft auch die Authentifizierung der Nutzer. Hierfür benötigt der Rechner ein Maschinenkonto im AD.
- kann Anmeldeinformationen cachen, so dass eine Anmeldung auch möglich ist, wenn der Domain Controller nicht erreichbar ist.

Folgende Module, die den Winbindd nutzen, werden mitgeliefert:

- ein PAM-Modul ermöglicht die Authentifizierung gegen Kerberos im AD
- ein nsswitch-Modul übernimmt u.a. die Namensauflösung von Benutzern und das ID-Mapping.

Wichtig: der Linuxrechner sollte den DNS-Server und den Zeitserver des Active Directories nutzen (i.Allg. beides der DC).

[1] LSA = Local Security Authority

ID-Mapping

Problem

Zuordnung Windows-Nutzer zu Unix-Nutzer schwierig, da Nutzer i.Allg. nicht einzeln auf Unix-System einrichtbar (und analog Gruppen).
Müssen aber wegen Dateiberechtigungen zur Verfügung stehen.

Lösung

- Nutzer- & Gruppen-Namen dynamisch per `nsswitch` von DC (analog wie User-Lookup in einem LDAP)
- Windows-Namen und -SID werden lokal nicht verwendeten `uid` und `gid` zugeordnet (idmap).

ID-Mapping: idmap_rid

- Mapping Windows-Name zu SID durch Abfrage DC einfach,
- Mapping SID zu uid/gid (`idmap`) schwierig:

  SID besteht aus 96-Bit Domän-ID und 32-BIT Relativ-ID (RID)

  uid/gid traditionell 16 Bits, heute unter Linux 31 Bits

⇒ Winbind Idmap Backend `rid`:

  Verwendung der RID + fester Offset ergibt lokal verwendete uid/gid

- auch andere Möglichkeiten konfigurierbar:
  - Wahl aus reservierter Range und dauerhafte Fixierung (tdb,ldap)
    (tdb: unterschiedliche uid/gid auf verschiedenen Rechnern, Datenbank)
    (ldap: zusätzlicher Verwaltungsaufwand)
  - Aus dem AD auslesen, wenn Services-For-Unix auf DC installiert sind
    (eher nicht empfehlenswert)
  - kein Mapping, sondern direkt (Windows-Name=Unix-Name)
    (dafür müssen alle AD-Benutzer lokal existieren)

Konfiguration in /etc/samba/smb.conf

```
[global]
realm = ADTEST.INTERN
workgroup = ADTEST
security = ADS
ntlm auth = No
winbind rpc only = Yes

template shell = /bin/bash
allow trusted domains = No
idmap backend = rid:ADTEST.INTERN=100000-199999
idmap uid = 100000 - 199999
idmap gid = 100000 - 199999
winbind enum users = Yes
winbind enum groups = Yes
winbind use default domain = Yes

winbind refresh tickets = Yes
```

Daten des Active Directory

für das nsswitch-Modul

Achtung: andere Syntax in Versionen 3.0-3.2 z.B. Debian Lenny

für das PAM-Modul

Aufnahme des Linux-Rechners ins AD

Anlegen eines Maschinenkontos für den Linuxrechner im AD (Kerberos host-Prinzipal), so dass dieser Domänenmitglied wird:

```
net ads join -U <Domänenadministrator>
```

Keine Änderungen am DC nötig!

- Idmap Backend `rid` errechnet `uids` und `gids` aus den SIDs im AD
- andere benötigte Informationen stellt Winbind über das NSSwitch-Modul standardisiert für alle AD-Nutzer zur Verfügung (Shell, Home-Verzeichnis: `/home/ADTEST/<adusername>`)
- anders als beim Einsatz von LDAP oder SfU[2] müssen keine neuen Dienste auf dem Domaincontroller installiert und gepflegt werden

[2] Microsofts *Services for Unix*, veraltet, mittlerweile nicht mehr empfohlen

Einrichtung von NSSwitch

Es muss lediglich in der NSSwitch-Konfigurationsdatei `/etc/nsswitch.conf` jeweils für `passwd` und `group` das Modul `winbind` hinzugefügt werden.

```
...
passwd: winbind compat
group:  winbind compat
shadow: compat
...
```

- Bei dieser Reihenfolge der Module (`winbind compat`) werden Nutzer stets zuerst im AD gesucht.

⇒ warten auf Timeout, wenn Winbind nicht verfügbar ist und man sich als lokaler Nutzer einloggen will.

→ in der Testphase lieber die Reihenfolge tauschen.

Einrichtung von PAM (bei Debian)

Es gibt deutliche Unterschiede zwischen den Distributionen.
Hier am Beispiel von Debian:

`/etc/pam.d/common-account` (Account-Verifikation)

```
account sufficient pam_winbind.so
account required pam_unix.so
```

`/etc/pam.d/common-auth` (Authentifizierung)

```
auth sufficient pam_winbind.so krb5_auth krb5_ccache_type=FILE
auth required pam_unix.so nullok_secure use_first_pass
```

`krb5_auth`	Kerberos benutzen
`krb5_ccache...`	TGT holen und speichern für spätere Nutzung
`use_first_pass`	das beim vorhergehenden Modul eingegebene Passwort verwenden (für Nutzer, die nicht im AD sind)

Auch hier wird zuerst Winbind befragt, so dass es zu Timeouts bei Nichtverfügbarkeit kommen kann.

Einrichtung von PAM (bei Debian)

`/etc/pam.d/common-password` (Passwortänderung; nicht getestet)

```
password sufficient pam_winbind.so
password required pam_unix.so nullok obscure md5
```

`/etc/pam.d/common-session` (Sitzungsverwaltung)

```
session required pam_mkhomedir.so skel=/etc/skel/ umask=0077
session required pam_unix.so
```

Das Modul `pam_mkhomedir.so` bewirkt, dass ein noch nicht vorhandenes Home-Verzeichnis bei der Anmeldung automatisch angelegt wird.

Besser nicht einen Windows-Share direkt als Homeverzeichnis nutzen, da:

- darauf keine Unix-Domain-Sockets angelegt werden können, werden aber von einigen Anwendungen benötigt (z.B. KDE)
- Konfigurationsdateien liegen unter Linux im Homeverzeichnis (beginnen mit einem Punkt, also unter Linux „versteckt“)
  → würde unter Windows im Profil-Verzeichnis liegen

Einhängen der AD-Shares

Zum Einhängen der Shares aus dem AD werden folgende Pakete benötigt:

- **smbfs** Mount-Programm `mount.cifs` und Hilfsprogramm `cifs.upcall` für Kerberos-Tickets
- **keyutils** Werkzeugsatz für die kernelinterne Schlüsselverwaltung, die von Dateisystemen und anderen genutzt werden kann

In `/etc/request-key.conf` Folgendes anfügen:

```
create cifs.spnego * * /usr/sbin/cifs.upcall %k %d
create dns_resolver * * /usr/sbin/cifs.upcall %k
```

`mount.cifs` holt sich nun automatisch mit dem TGT des Nutzers ein Kerberos-Ticket für den Dateiserver.

Damit kann ein Share `\\dc1.adtest.intern\homes` vom Benutzer ohne erneute Passworteingabe wie folgt eingehängt werden:

```
mount.cifs //dc1.adtest.intern/homes/
                              ~/adhomes/ -o sec=krb5i,guest,directio
```

(`directio` deaktiviert Caching, sonst Probleme bei paralleler Nutzung)

Optional: SSH-Login mit Kerberos-Ticket

- Linux-Rechner im AD als sicher für die Weiterleitung von Kerberos-Tickets markieren (unklar, ob dies wirklich nötig ist)
- GSSAPI[3] für SSH aktivieren:
  Auf dem Server in `/etc/ssh/sshd_config`:

```
GSSAPIAuthentication yes
GSSAPIKeyExchange yes
GSSAPICleanupCredentials yes
```

  Auf dem Client z.B. in `/etc/ssh/ssh_config`

```
GSSAPIAuthentication yes
GSSAPIDelegateCredentials yes
```

Aber: im Moment noch nicht wirklich nützlich, da der NX-Client (s.u.) das noch nicht kann. Gepatchtes Putty für Textmodus geht aber.

[3] Generic Security Services Application Program Interface

Optional: Einrichtung von Kerberos

nicht notwendig, aber hilfreich zur Fehlersuche

In `/etc/krb5.conf` eintragen:

```
[libdefaults]
default_realm = ADTEST.INTERN
```

TGT holen, anzeigen und wieder löschen:

```
user@adtest:~$ kinit aduser@ADTEST.INTERN
Password for aduser@ADTEST.INTERN:
user@adtest:~$ klist -5
Ticket cache: FILE:/tmp/krb5cc_101125
Default principal: aduser@ADTEST.INTERN

Valid starting    Expires           Service principal
08/18/09 15:30:20 08/19/09 01:30:51 krbtgt/ADTEST.INTERN@ADTEST.INTERN
      renew until 08/19/09 15:30:20
user@adtest:~$ kdestroy
user@adtest:~$ klist -5
klist: No credentials cache found (ticket cache FILE:/tmp/krb5cc_1000)
```

Probleme

- Winbind-Dienst stürzt vereinzelt ab (insbesondere bei Nutzung der AD-Shares? genauer Grund noch unklar)
  Lösung: automatischer Neustart,
  in `/usr/share/samba/panic-action` unten anfügen:

  ```
  echo "/etc/init.d/winbind restart" | at now + 1 minute
  ```

  Jedoch so derzeit noch Fehlermeldungen, vermutlich wegen Umgebungsvariablen; funktioniert aber
- Winbind (und Samba) in Debian Lenny sind sehr alt, daher eher mehr Probleme mit Windows 2008 Server zu erwarten
  Lösung: Winbind aus Lenny-Backports benutzen:
  `http://www.backports.org`

mögliche Alternative: Likewise Open

- vor einiger Zeit als OpenSource freigegebenes alternatives Paket zur Aufnahme von Linux-Rechnern in eine AD-Domäne
- scheint irgendwie auf Samba zu basieren
- im Ubuntu-Repository, aber Registrierung für Downloads direkt von der Homepage
- CIFS-Server nicht frei

`http://www.likewise.com/products/likewise_open/`

Das X-Window System

- ein Software-System und Netzwerk-Protokoll
- stellt Standardbaukasten zum Bau einer grafischen Benutzeroberfläche zur Verfügung.
  Unter anderem:
  - Darstellung von Grafiken (Bitmaps)
  - Zeichnen und Bewegen von Fenstern
  - Tastatur- und Mauseingaben
- Aussehen und Verhalten wird bestimmt von Programm und Windowmanager
- Netzwerkprotokoll ist unverschlüsselt
  $\Rightarrow$immer über ssh tunneln (`ssh -X` bzw. `ssh -Y`)

R|R|Z|N

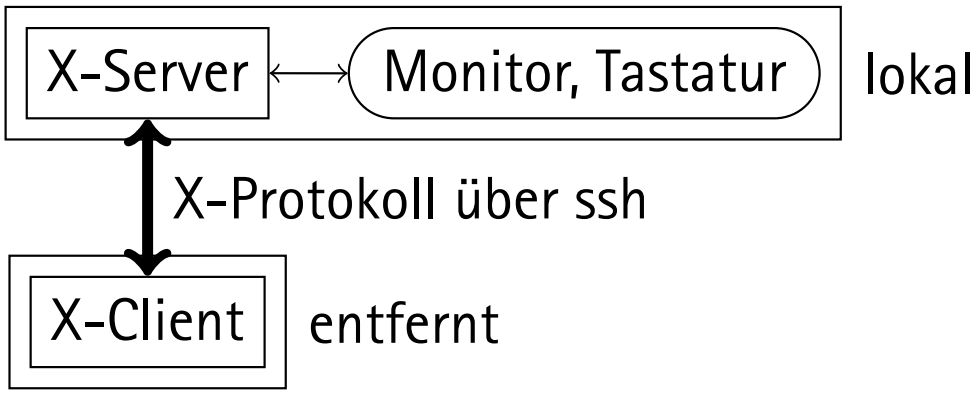
Client-Server-Modell

X-Server

- läuft auf dem lokalen Rechner
- zuständig für Grafikausgabe
- verarbeitet Maus- und Tastatureingaben

X-Clients

- Windowmanager und Anwendungsprogramme
  Bsp: KDE, Firefox, xterm, Openoffice.org
- können auch auf einem entfernten Rechner laufen
- Ein- und Ausgabe über den X-Server



NX-Terminalserver: X-Window System

Problem:

- mit Toolkits (GTK, QT, ...) geschriebene Programme benötigen meist viele Roundtrips (X-Server↔X-Client); oft auch werden Grafiken (Knöpfe) mehrfach gezeichnet

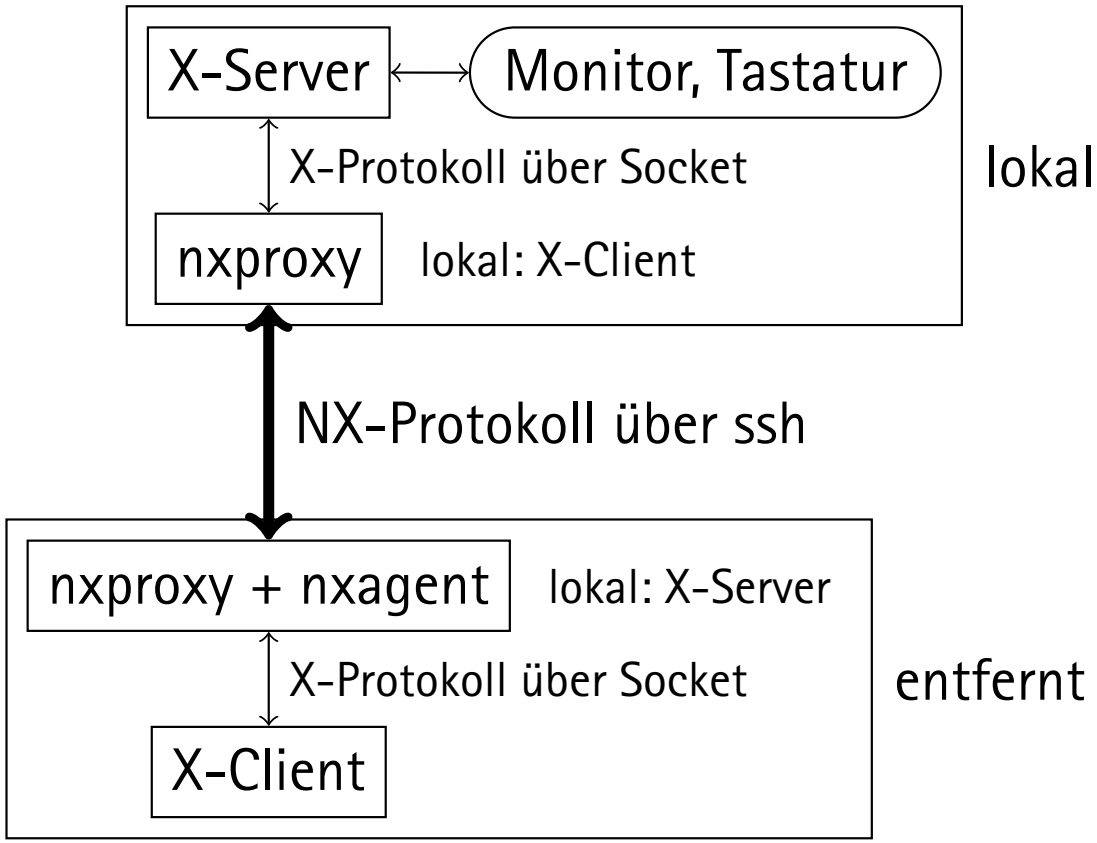
⇒ schlecht nutzbar bei hoher Latenz und/oder geringer Bandbreite der Verbindung (Modem, ISDN, DSL-Uplink): zähe Bedienung, langsamer Aufbau der Oberfläche

Lösung: NX-Protokoll[4] – eine Erweiterung des X-Protokolls

- drastische Reduzierung der Roundtrips
- intelligentes Zwischenspeichern (Caching)
- Datenkompression

[4] Entwickelt von NoMachine: www.nomachine.com, Kernbibliotheken und Protokoll GPL

NX-Terminalserver: NX-Protokoll



Ein NX-„Server“ bietet zusätzliche Funktionen:

- Suspend & Resume
- Übertragung von Audioausgabe,
- Nutzung lokaler Drucker und Freigaben
- Bilder werden komprimiert (runtergerechnet)
- z.T. dynamische Anpassung an vorhandene Bandbreite
- z.T. zentrale Verwaltung von Sitzungen

NX-Server

NX-Server von NoMachine:

- kostenloser NX-Server max. 2 Nutzer; 10 Nutzer ab $750, mehr ab $1200
- kostenloser NX-Client für Windows, Linux, Mac OS X, Solaris
- Funktionsweise:
  - Einloggen auf dem Server stets als Nutzer `nx` mit passwortlosem ssh-Key
  - erst später mit su oder ssh Wechsel in den gewählten Nutzeraccount

⇒ Ermöglicht Zusatzfunktionen (zentrale Verwaltung, geteilte Sitzungen), jedoch aus sicherheitstechnischer Sicht problematisch.

⟶ Der mitgelieferte ssh-Key sollte durch selbsterzeugten ersetzt werden!

FreeNX

- Bash-Wrapper-Skripte für GPL-lizensierte Bibliotheken von NoMachine
- als „Drop In"-Ersatz für den NX-Server von NoMachine entwickelt
- funktioniert auch im Usermode:
  - Nutzer loggt sich (fast) ganz normal mit ssh in seinen Account ein
  - der „NX-Server" wird vom Nutzer selbst gestartet.

Eigenschaften:

- kein zentraler Nutzer mit passwortlosem SSH-Key
- persönliche SSH-Keys können verwendet werden
- theoretisch auch Login mittels Kerberos-Ticket, jedoch praktisch bisher nicht realisierbar, da zu alter ssh-Client in NoMachines NX-Client

→ zur Nutzung des NX-Client von NoMachine muss dort zusätzlich ein Wrapper installiert werden.

Im Folgenden wird die Einrichtung von Server und Client zur Nutzung des Usermode beschrieben.

R|R|Z|N

Einrichtung des Servers

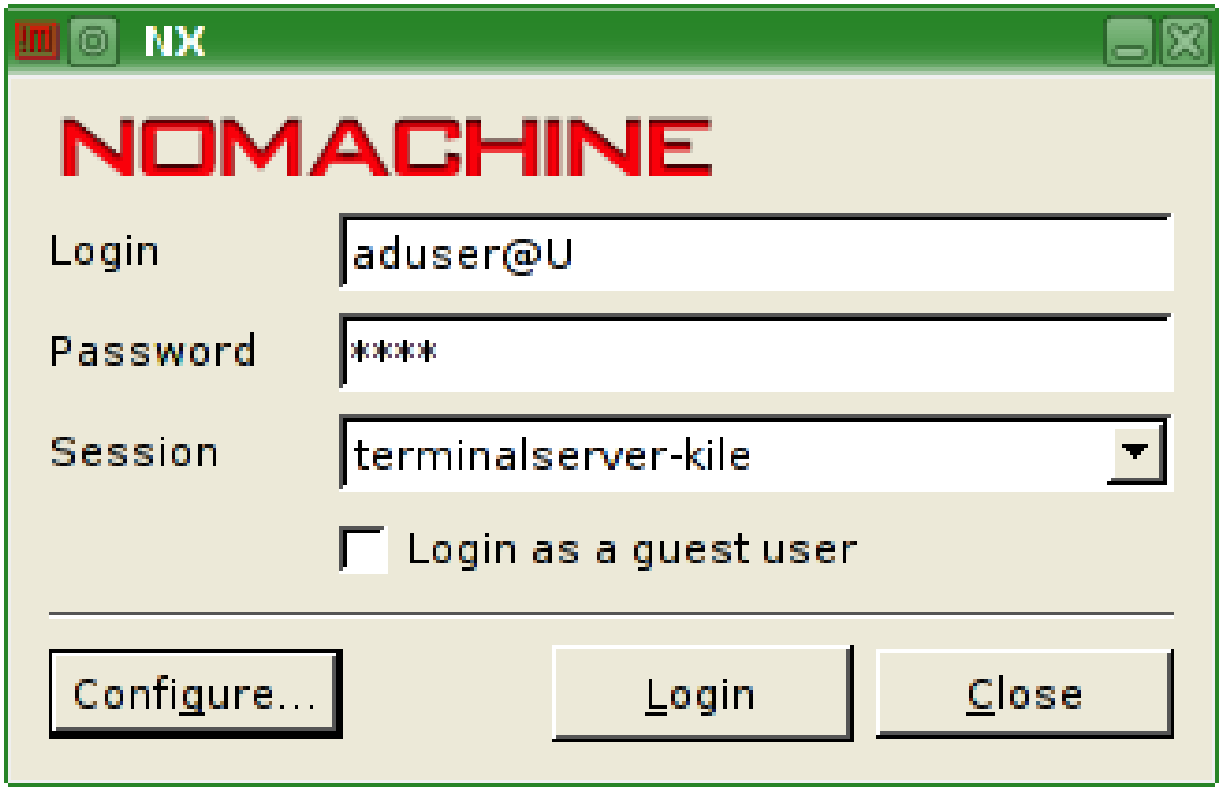
- für Usermode vorbereitetes FreeNX entpacken als `/usr/local/NX4U`:
  `http://download.berlios.de/freenx/NX4U.tar.gz`
- `/usr/local/NX4/etc/node.conf` nur Usermode-Authentifizierung:

```
ENABLE_USERMODE_AUTHENTICATION="1"
ENABLE_SSH_AUTHENTICATION="0"
ENABLE_SU_AUTHENTICATION="0"
ENABLE_USER_DB="0"
ENABLE_FORCE_ENCRYPTION="1"
```

- Auf unseren Server sollte XFCE als grafische Umgebung, da „leichter" als Gnome und KDE; dies ist aber im NX-Client von Nomachine nicht vorgesehen, deshalb Symlink anlegen:
  `ln -s /usr/local/startxfce4 /usr/local/bin/startkde`

R|R|Z|N

NX-Client von NoMachine



- mit FreeNX-Patch als Usermode-Client nutzbar
- auch gezielt nur einzelne Programme auswählbar (ohne komplette Umgebung)

R|R|Z|N

Einrichtung des NX-Client von NoMachine

- kostenlos von NoMachine herunterladbar[5], einfach zu benutzen
- enthält einen angepassten openssh-Client (`nxssh.exe` resp. `nxssh` im Unterverzeichnis `bin`),
- dieser wird umbenannt in `mxssh` durch einen Wrapper ersetzt (`http://download.berlios.de/freenx/`)
- bei der Einrichtung des Accounts im NX-Client `@` vor den Servernamen und `@U` hinten an den Nutzernamen setzen:

  Host: `@terminalserver.adtest.intern`

  Login: `aduser@U`

[5] `http://www.nomachine.com/download.php`

Probleme

- FreeNX wird scheinbar nicht mehr weiterentwickelt und ist als Sammlung von (großen) Shellskripten wohl auch de facto kaum wartbar.
- Patchen des NX-Clients auf dem Client-Rechner ist nicht optimal.
- von anderem Linux aus evtl. Probleme mit der Tastaturbelegung
  Lösung: auf dem Server aktuelle keymap setzen:

```
setxkbmap de -keycodes 'evdev+aliases(qwertz)' \
             -symbols 'pc+de+inet(evdev)'
```

mögliche Alternativen

Neatx NX-Server aus internem Google-Projekt, Usermode soll angeblich funktionieren, noch recht jung und unfertig, wird langsam entwickelt
http://code.google.com/p/neatx/

TaciX NX-Server + Client vom Paketbetreuer von FreeNX in Ubuntu, basiert auf D-Bus, jung und unfertig, kaum Weiterentwicklung derzeit
https://edge.launchpad.net/tacix

X2go sieht interessant aus, anderer Ansatz aus Client und Server, alter Windows-Client (08/2009) nicht nutzbar, aber Neuer wohl deutlich besser (ungetestet), aktive Weiterentwicklung
http://www.x2go.org

OpenNX OSS-Ersatz für NX-Client von Nomachine, Usermode funktioniert scheinbar nicht, scheinbar nur sporadische Weiterentwicklung
http://sourceforge.net/projects/opennx/

X2 TS kommerzieller TerminalServer, nicht weiter untersucht
http://www.2x.com/terminalserver/

Fazit

- Setup nicht ganz ausgereift, aber durchaus funktionsfähig
- Ermöglich Remote-Zugriff auf Daten und Programme, ohne sicherheits-technisch deutlich riskanteren Vollzugriff auf internes Netz via VPN
- kein zusätzlicher Aufwand für AD-Administration

Links/Literatur

- Kerberos-Tutorial-Reihe der iX (2007) aus dem Heise-Verlag:
  - I Einführung in Kerberos (iX 3/2007)
  - II „Kerberisierung“ von Netzwerkdiensten (iX 4/2007)
  - III Netzweites Single Sign-On (iX 5/2007)
- AD-Integration-Tutorial-Reihe der iX (2008):
  - I Migration der Authentifizierung (iX 10/2008)
  - II AD-Benutzerinformationen für unixoide Systeme (iX 11/2008)
  - III Linux-Dienste für SSO an AD anbinden und Erweiterung auf Active Directory Forest (iX 12/2008)
- aus der Samba-Tutorial-Reihe der iX (2008):
  - II Samba als AD-Mitglied (iX 4/2008)
- Samba Dokumentation umfassen: Manpage smb.conf und Samba-Howto
  `http://de.samba.org/samba/docs/man/Samba-HOWTO-Collection/`
- RRZN-Anleitung zur Einbindung von Linux ins AD:
  `http://www.rrzn.uni-hannover.de/anl-linclient-ads.html`

Links/Literatur

- Blog-Eintrag von Felipe Alfaro Solana zum FreeNX Usermode
  http://www.felipe-alfaro.org/blog/2009/01/18/freenx-usermode-authentication-and-mac-os-x/
- FreeNX 7.3.0 Freigabe-Mitteilung: http://mail.kde.org/pipermail/freenx-knx/2008-August/007324.html
- RRZN-Anleitungen zu FreeNX und NX-Clients im Usermode:
  http://www.rrzn.uni-hannover.de/anl-nx-linserv.html
  http://www.rrzn.uni-hannover.de/anl-nx-client.html