NIS Server

Installation und Konfiguration eines NIS Servers,
der die Benutzerverwaltung für verschiedene Linux-Clients
zur Verfügung stellt

FSI 02

Beate Schaaf

Marc Oberhofer

INHALTSVERZEICHNIS

1 Einleitung

Im Modul „Mehrplatzbetriebssysteme“ erhielt jedes Team eine Aufgabenstellung zum Thema Linux und Mehrplatzbetriebssysteme zugelost. Die von uns gezogene Aufgabenstellung lautete wie folgt:

Installation und Konfiguration eines NIS Servers, der die Benutzerverwaltung für verschiedene Linux-Clients zur Verfügung stellt

Im Folgenden haben wir versucht, so anschaulich und nachvollziehbar wie möglich die dafür notwendigen Schritte bzw. Einstellungen darzustellen und zu erläutern.

2 NIS – was ist das?

Der Network Information Service (NIS) ist ursprünglich eine Entwicklung von SUN und als SUN Yellow Pagers oder YP bekannt. Da dieser Name durch die British Telecom geschützt ist und nicht ohne die entsprechenden Rechte genutzt werden darf, erfolgte die Umbenennung in Network Information Service – kurz NIS.

Der NIS-Server speichert Kopien von gemeinsamen Konfigurationsdateien (z. B. /etc/shadow, /etc/passwd, /etc/group) verschiedener vernetzter Computer in einer Datenbank. Die NIS Clients wiederum richten ihre Anfragen an diese Server, anstatt eigene Konfigurationsdateien zu benutzen.

NIS (Network Information System) ist also ein Protokoll und zugehörige Client-Server-Software, mit deren Hilfe beliebige Daten über ein (lokales) Netz verteilt werden können, so dass sie nur auf einem zentralen Server gepflegt werden müssen.

Typischerweise wird es für die Dateien /etc/passwd und /etc/group genommen. Eine Datei, die per NIS verteilt wird, wird NIS-MAP genannt.

2.1 Grundlegende NIS-Begriffe und Funktionserläuterung

Um NIS besser zu verstehen, müssen einige Fachausdrücke erläutert werden. Das Computersystem, auf dem die zentrale NIS-Datenbank gehalten wird, wird *NIS-Master-Server* genannt. Die anderen Computer im Netz, die auf diese Datenbank zugreifen, heißen *NIS-Clients*. Ein Computer kann gleichzeitig Client und Servers ein, er fragt sich dann selbst nach Einträgen in der Datenbank.

Darüber hinaus kann es weitere Computer geben, die Kopien der Datenbank bereithalten. Diese werden *NIS-Slave-Server* genannt. Ein Slave-Server kann die Rolle eines Masters übernehmen, wenn der Master abgeschaltet wird, abstürzt oder Netzkommunikationsprobleme hat. Slave-Server werden auch aus Geschwindigkeitsgründen eingesetzt: Wenn ein Master-Server zu langsam auf Anfragen antwortet, kann ein NIS-Client sich auch mit einem der Slave-Server verbinden, um bessere Antwortzeiten zu erhalten. NIS stellt automatisch sicher, dass allen Slave-Servern die aktuellen Daten zur Verfügung stehen. Die Daten werden dazu vom Master auf die Slave-Server übertragen, sobald Änderungen an der zentralen Datenbank vorgenommen werden. Hierdurch bleiben die Daten immer synchron. In kleinen Netzen sind Slave-Server nicht wirklich notwendig.

Ein lokales Computernetz kann logisch in verschiedene „NIS-Zonen“ eingeteilt werden, wobei jede Zone eine eigene Datenbank verwendet. Diese Zonen werden *NIS-Domänen* genannt. Jede Domäne benötigt ihren eigenen NIS-Master-Server, wobei es jedoch keine Einschränkungen gibt, welche Maschine Master für welche Domäne sein muss. Beispielsweise können Sie einen Rechner einrichten, der zwei NIS-Master-Server für zwei unterschiedliche Domänen beherbergt. Es muss nur sichergestellt sein, dass jeder Client weiß, zu welcher Domäne er gehört. Aus diesem Grunde benötigt jede Domäne einen *NIS-Domänenamen*, der eine eindeutige Zuordnung erlaubt. Ein Name wird immer benötigt, auch wenn nur eine einzige Domäne im Netz besteht.

Ein NIS-Client verwendet den NIS-Domänenamen, um einen geeigneten NIS-Maser-Server zu finden. Dies passiert üblicherweise beim Systemstart, wobei eine Verbindung zur Server-Datenbank aufgebaut wird. Dieser Prozess wird als „Binden an die NIS-Domäne“ bezeichnet.

Die NIS-Datenbank stellt verschiedene „Listen“ von Informationen zur Verfügung, beispielsweise die Liste der Benutzer, die im Netz arbeiten dürfen oder eine Liste aller Rechner, die Teil des Netzes sind. Die verschiedenen Auflistungen werden *NIS-Maps* genannt. Jede Liste hat einen Namen, der sie identifiziert. Es gibt einige vordefinierte Standardnamen, die in jeder NIS-Implementierung verwendet werden. Beispielsweise die passwd.byuid, die die Benutzer geordnet nach User-IDs enthält.

2.2 Verdeutlichendes Beispiel

User X will sich an einem beliebigen Rechner im Netzwerk anmelden. Der NIS Server ist nicht installiert. In diesem Fall müsste der User auf allen Rechner angelegt sein (lokal). Ist aber NIS installiert, ist es uns möglich, den User nachdem er auf dem NIS-Server angelegt wurde, jedem Rechner bekannt zu geben.

3 Systemvoraussetzungen Hard- und Software

Hardwaretechnische Voraussetzungen sind ein Standard-PC mit Netzwerk-Funktionalität.

Die von uns verwendete Linux-Distribution ist SUSE Linux 9.2, hierbei ist zu beachten, dass das NIS-Softwarepaket [Version ypserv (ypserv) 2.14] sowie ypbind [Version 1.17.3] im YaSt installiert wurde.

4 Installation eines NIS-Server

Die Rechner werden über Netzwerkkabel miteinander verbunden.
(direkte Verbindung über Cross-Over-Kabel)

Voraussetzung ist zudem, dass auf dem Server der ypserv-Prozess läuft, welcher die Anfragen der Clients annimmt und bearbeitet. Die Kommunikation läuft dabei über RPC ab. Der Begriff RPC ist unter 4.3 genauer erklärt.

4.1 Konfiguration der IP-Adresse

Nach dem Start des Linux-System wird mit Strg + Alt F2 auf die Shell-Console gewechselt. Anschließend muss mit dem Befehl ifconfig die IP-Adressen beider Rechner ausgelesen werden.

Wir haben uns dafür entschieden, im IP-Adressbereich 192.168.xxx.xxx zu arbeiten (in einem privaten Netzwerk).
Dementsprechend wurde der Server mit der Adresse 192.168.111.34 festgelegt (Befehl hierzu: ifconfig eth0 192.168.111.34).

Beim Client wurde die IP Adresse 192.168.111.35 eingerichtet.
(Befehl hierzu: ifconfig eth0 192.168.111.35).

ACHTUNG: Zur dauerhaften Konfiguration dieser IP-Adresse ist es notwendig, diese in der Datei /etc/sysconfig/network/ifcfg –eth-id-MAC ADRESSE zu konfigurieren (Befehl: joe sysconfig: Speichern der Datei mit Tastenkombination Strg + k+ x).

Bei der Einstellung ist zu beachten, falls die IP-Adresse geändert wird, auch die Broadcast, die Netmask und Netzwerk geändert wird. Wenn diese Konfiguration nicht durchgeführt wird, setzt sich der Wert der IP-Adresse immer wieder auf den ursprünglichen Wert in der sysconfig-Datei gespeichert ist, zurück!
(siehe Punkt 7 Aufgetretene Probleme und deren Lösungsansätze)

4.2 Setzen der Domäne

Mit einem Eintrag in das File /etc/defaultdomain wird die NIS-Domäne permanent bekannt gegeben. Dies funktioniert für temporäre Eingaben auch mit:

domainname + Domänename

in der Shell-Console, hat aber keine permanente Bekanntgabe zur Folge.

Diese Einstellung muss auf dem Server und auf dem Client erfolgen!

4.3 Überprüfen des Portmaps und Starten des Dienstes ypserv

Die Funktionalität der Yellow Pages basiert im wesentlich auf den Remote Procedure Calls (RPC's), dem Austausch von Anfragen zwischen Server und den Clients.

Der RCP Portmapper *portmap* ist ein Programm, das die RCP-Programm-Nummern in Portnummern übersetzt. Wenn ein RPC gestartet wird, wird portmap mitgeteilt, welchen Port es benutzen will und welche RPC-Nummer es ansprechen will.

Wenn ein Client eine RPC-Anfage an eine bestimmte Programm-Nummer richten will, wird er zuerst den portmap-Server kontaktieren, um die Nummer des Ports zu erfahren, auf dem dieses Programm läuft. Dann kann der Client die RPC-Pakete an den entsprechenden Port schicken.

Ob der Portmapper auf dem System aktiv ist, überprüfen wir mit dem Aufruf von *rpcinfo –p.*

Folgende Ausgaben sind möglich:

Rpcinfo: Kann den Portmapper nicht erreichen.....[hier muss der Portmapper gestartet werden mit dem Befehl /etc/init.d portmap start]. **Diese Einstellung muss auf dem Server und auf dem Client erfolgen!**

ODER

Programm	vers	proto	port	
100000	2	tcp	111	portmapper
100000	3	udp	111	portmapper

Hier läuft der Portmap, Anforderungen über TCP und UDP werden behandelt und unterstützt.

Zum Starten des Serverdienstes *ypserv* genügt die Eingabe: *ypserv start*

Nach erneuter Eingabe des Befehl rpcinfo –p erscheint nach erfolgreichem Start dieses Dienstes folgende Ausgabe

Programm	vers	proto	port	
100000	2	tcp	111	portmapper
100000	3	udp	111	portmapper
100004	2	udp	804	ypserv
100004	1	udp	804	ypserv
100004	2	tcp	807	ypserv
100004	1	tcp	807	ypserv

Dieser Dienst (ypserv) ist die Grundvoraussetzung, damit die NIS-Datenbank erstellt werden kann.

4.4 Anpassen der Datenbankerstellung

Im nächsten Schritt werden die zu verteilenden Informationen eingestellt.

Die Einstellung erfolgt unter dem File /var/yp/Makefile, die nur auf dem Server aktiv ist. Hier kann z. B. eingestellt werden, ob Änderungen automatisiert an weitere (Slave-)Server verteilt werden soll, ob die Passwd- und Shadow-Files vermischt werden sollen und - die wichtigste Einstellung - welche Maps kreiert werden sollen.

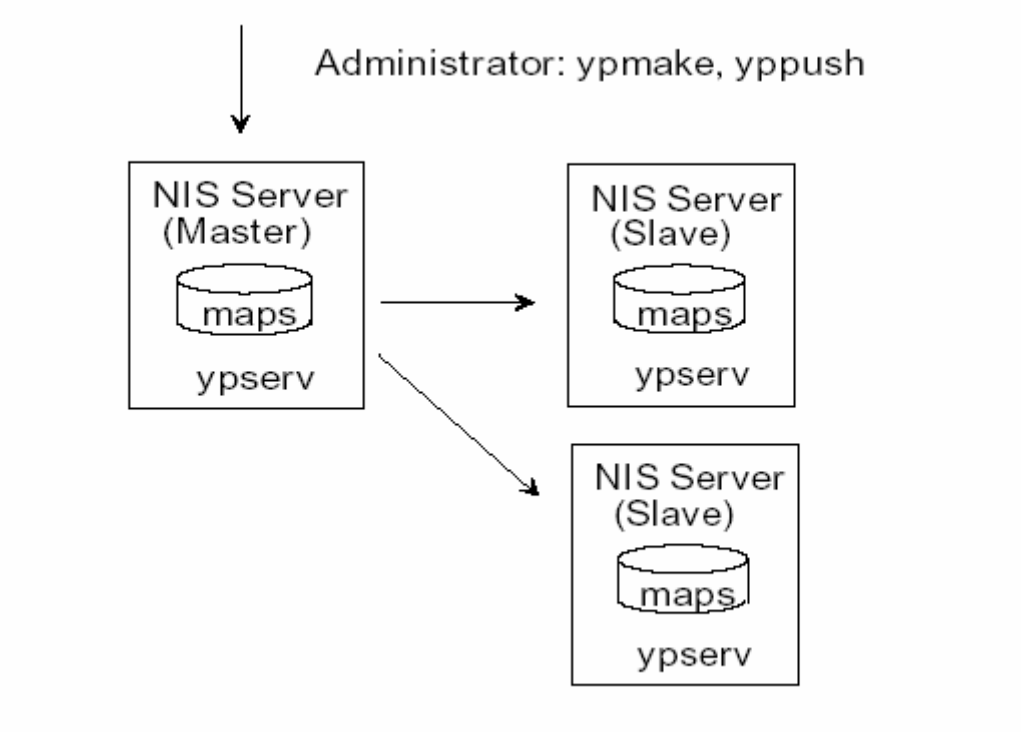
In die Konfigurationsdatei /var/yp/ypservers können alle YP-Server und Backup-Server eingetragen werden. Bei der Erstellung der Datenbank wird dieses File ausgewertet, was wir im nächsten Schritt näher ausführen.

Um die Maps automatisiert erzeugen zu lassen, benutzt man den Befehl *make*. Den Einstellungen im File Makefile entsprechend, werden automatisch mit dem Zusatz zu *make* -C /var/yp die Verzeichnisse erstellt. Der Parameter –C und Angabe des Verzeichnisses bewirkt die Erstellung der Datenbank in dem genannten Verzeichnis.

Achtung: Bei jeder Änderung der Benutzerverwaltung muss ein sogenanntes update der Datenbank erfolgen. Dies bedeutet, der Befehl *make* muss auf dem Server in dem Verzeichnis /var/yp, welches die Datenbank enthält, neu ausgeführt werden! Dies hat zur Folge, dass die Datenbank neu generiert wird.

Der Befehle *yppush* bewirkt, dass eine Kopie der aktualisierten Datenbank vom Master zum Slave transferiert wird. In anderen Distributionen heißt dieser Befehl auch *ypmake*.

Zur Verdeutlichung anbei eine Skizze.



HINWEIS: Wir setzen keinen Slave-Server ein, aber in größeren Netzen kann er folgende Vorteile bringen:

- Ø Er dient als Redundanz beim Ausfall des Master-Servers.
- Ø Er dient der Entlastung des Master-Servers.
- Ø In segmentierten Netzen ermöglicht er das Ansprechen eines Servers mittels Broadcast.

Sollten weitere Slave-Server manuell in die Map-Generierung miteinbezogen werden, muss der Befehl *ypinit* mit dem Parameter*–m* unter /usr/lip/ gestartet werden. Der Parameter *–m* muss nur dann angewendet werden, wenn der lokale PC auch der Master-Server ist.

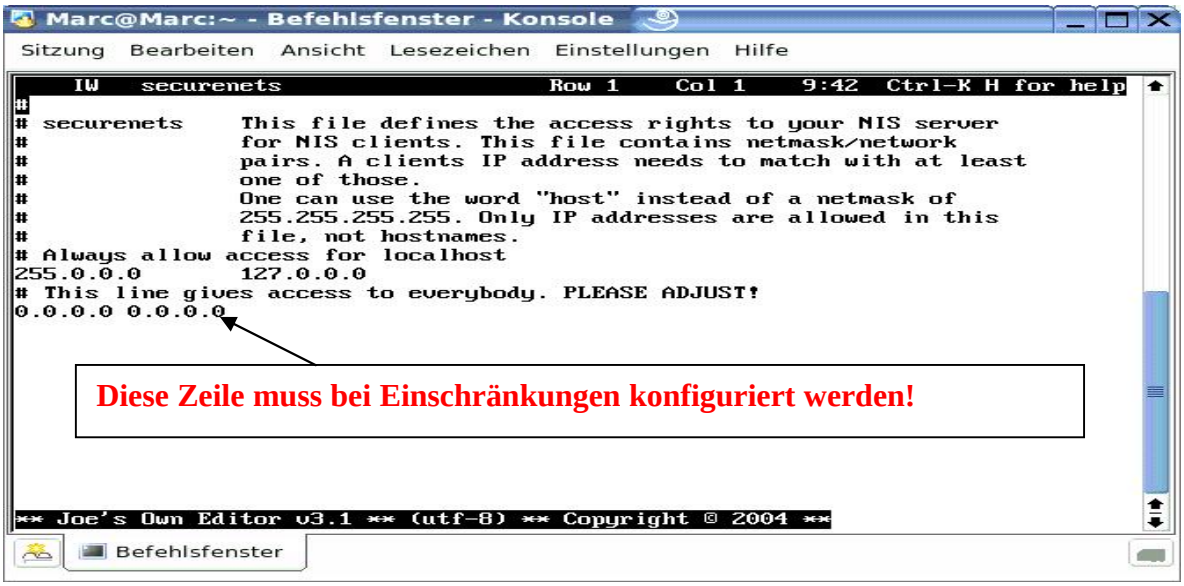
Eine Erfolgskontrolle über die Erstellung der Datenbank kann man durchführen, indem man überprüft, ob unter dem Verzeichnis /var/yp ein Verzeichnis mit dem Domainname erstellt wurde. Der Inhalt kann sich wie folgt darstellen (je nach Einstellung der Makefile-Datei)

NIS-MAPS (Auszug)

Information	**Lokale Dateien**	**NIS-Map (/var/yp/*DOMAINNAME)***
Nutzerkennung	/etc/passwd	passwd.byname Passwd.byuid
Gruppenkennzeichnungen	/etc/group	group.byname group.bygid
Rechnernamen	/etc/hosts	hosts.byname hosts.byaddr
Netzwerkname	/etc/networks	networks.byname networks.byaddr
Netzwerkdienste	/etc/services	services.byname services.byaddr
RPCs	/etc/rpc	rpc.by.name rpc.by.number

4.5 Einschränkung des Map-Zugriffs

Damit nicht jeder in die freigegebenen Maps des Master-Servers einblicken kann, besteht die Möglichkeit, nur Anfragen von vorher definierten PCs zuzulassen. Diese PCs werden in dem File /var/yp/securenets definiert. Hier möchten wir gleich auf den Punkt 8 (Sicherheit) verweisen.



ACHTUNG: Die Grundeinstellung beinhaltet keine Einschränkungen, Zugriff ist jederzeit möglich!

5 Konfiguration eines NIS-Clients

Der Client-Dienst basiert auf dem *ypbind* Daemon. Dieser Dienst ermöglicht dem Client herauszufinden, auf welchen Server er zugreifen muss. Er sendet also unsere Client-Anforderung an den NIS-Server.

5.1 Ypbind-Daemon

Zum Anpassen dieses Dienstes wechseln wir in das Verzeichnis /etc/yp.conf. In dieser Datei müssen folgende Einträge enthalten sein:

Domain *(DOMAINNAME)* server *(HOSTNAME)*
Der Client sucht nach Hostname für die Domaine

Domain *(DOMAINNAME)* broadcast
Der Client fragt per Rundspruch im lokalen Netzwerk nach der Domäne

Oder, wie es auch bei uns funktionierte:

ypserver (*IP-Adresse)*
Der Client spricht direkt per IP-Adresse die Domäne an

HINWEIS: Sollte die Konfigurationsdatei yp.conf nicht existieren oder nicht korrekt konfiguriert sein, so wird die Anfrage an das Netzwerk immer per Broadcast gestellt.

Nun wird der Dienst *ypbind* (im Verzeichnis /usr/sbin) durch den Aufruf *ypbind start* gestartet.

Zur Überprüfung, ob der ypbind-Dienst funktioniert, rufen wir den Portmapper auf (Befehl: *rpcinfo –p),* siehe Punkt 4.3*:*

Programm	vers	proto	port	
100000	2	tcp	111	portmapper
100000	3	udp	111	portmapper
100007	2	udp	1001	ypbind
100007	2	tcp	1021	ypbind

Die Funktionalität des NIS kann nun getestet werden, in dem auf dem Client versucht wird, sich mit einem Konto anzumelden, welches lokal auf dem Server liegt.

Die Anmeldung sollte dann auf der CLI oder auch Shell (Command Line Interface) - Oberfläche bei einem funktionierenden System möglich sein.

Bei Änderungen, welche die Dienste betreffen, müssen die Dienste beendet und wieder neu gestartet werden, damit die Änderung greift!

6 Funktionskontrolle

In der folgenden Funktionskontrolle sind alle auszuführenden Schritte detailliert aufgeführt, die einzugebenden Befehle sind kursiv formatiert.

6.1 Server-Installation

- System booten, in der KDE-Anmeldemaske mit *Strg + Alt + F2* zur CLI wechseln (siehe Shell im Stichwortverzeichnis)
- Einloggen als Root
- Ifconfig zur Ausgabe der IP-Adresse, IP Adresse konfigurieren (*ifconfig eth0 192.168.111.34*), Adresse notieren um später die Kommunikation zwischen Server und Client zu überprüfen
- Wechsel ins Verzeichnis /etc
- Aufruf des File Defaultdomain mit joe (*joe Defaultdomain*)
- Setzen des Domainname „fire“, sichern des File Defaultdomain mit *Strg + k + x*
- Eingabe des Befehls *rpcinfo –p* zum Test des Portmappers, Portmapper Dienst muss gestartet sein, siehe Punkt 4.3)
- Wechsel ins Verzeichnis /var/yp
- *Ypserv start* eingeben, um diesen Dienst zu starten
- Eingabe des Befehls *make –C* zum Erstellen bzw. Update der Datenbank
- Kontrolle ob die NIS-Datenbank erstellt wurde (Verzeichnis mit dem Namen der NIS Domäne wurde erstellt)
- Zusätzlich kann der Ypbind Dienst gestartet werden, damit kann das *yptest* ausgeführt werden, welches die Funktion des NIS Servers überprüft. Eingabe *Ypbind start*

6.2 Client-Installation

- System booten, in der KDE- Anmeldemaske mit *Strg + Alt + F2* zur CLI wechseln (siehe Shell im Stichwortverzeichnis)
- Einloggen als root
- Ifconfig zur Ausgabe der IP-Adresse, IP Adresse konfigurieren (*ifconfig eth0 192.168.111.35*)
- Ping auf den Server anhand der notierten IP-Adresse des Servers *(ping 192.168.111.34),* um die Verbindung auf Funktionalität zu prüfen
- Ping mit *Strg + c* abbrechen
- Setzen des Domainname „fire“, sichern des File Defaultdomain mit *Strg + k + x* zum sichern.
- Wechseln ins Verzeichnis /etc. In Konfigurationsfile Yp.conf den NIS – Server incl. IP Adresse eintragen
- Start ypbind (mit *ypbind star*t)
- *Start ypwhich* zur Ausgabe der IP-Adresse des Servers, wenn der NIS-Server erreichbar ist
- Tool *Yptest* einsetzten um Funktion zu überprüfen

Allen Usern, die in der NIS Datenbank eingetragen sind, können sich jetzt an dem Client anmelden (CLI)

7 Aufgetretene Probleme und deren Lösungsansätze

Bei der Konfiguration unseres NIS-Servers sind wir teilweise an für uns schwierige Punkte gestoßen, da sich in der Linux-Version 9.2 im Gegensatz zu Vorgänger-Versionen einiges getan hat.

Nachfolgend haben wir daher die Punkte aufgelistet, die sich bei uns als Fehlerquellen erwiesen haben und mit einem kurzen Lösungsansatz ergänzt.

7.1 IP-Adressen und Domainname

Die IP-Adresse sollte möglichst in der Konfigurationsdatei unter /etc/sysconfig/network/ifcfg Mac-Adresse eingetragen werden.

Ebenso sollte der Domainname in dem File /etc/defaultdomain eingetragen werden. Damit wird verhindert, dass temporär eingetragene oder geänderte IP-Adressen und Domänenamen nach einem Neustart wieder auf die Einstellung der o. g. Konfigurationsdateien zurückgesetzt werden.

Des Weiteren sollte darauf geachtet werden, dass der Servername in der Konfigurationsdatei ypservers als IP-Adresse eingetragen ist. Dies bewirkt, dass auf jeden Fall der Server ansprechbar ist, unabhängig von einer aktivierten Namensauflösung (DNS).

Fehlermeldung: yp_bind: Domaen not bound
Möglicherweise wurde die Domäne nicht in allen Konfigurationsdateien bekannt gegeben, in unserem Fall /etc/yp.conf.

Zur möglichen Fehleranalyse sollte der Befehl *yptest* ausgeführt werden.

7.2 Konfigurationsdateien der Dienste ypbind und ypservers

Alle wichtigen Einstellungen erfolgen unter den Konfigurationsdateien /etc/sysconfig/ypbind bzw. /etc/sysconfig/ypservers. In Vorgänger-Linux-Versionen und in anderen Distributionen werden teilweise andere Konfigurationsdateien benutzt, darauf wird aber in dieser Beschreibung nicht eingegangen.

7.3 YP-Tools

Die YP-Tools können sehr hilfreich bei der Fehlersuche sein, im einzelnen sind sie im Anhang aufgelistet.

Wenn das Anmelden des Clients am Server nicht möglich war, erhält der Administrator eine Benachrichtigung, die lautet: „You have a mail in /var/mail/root“. Darin können nützliche Informationen über die Fehlerquellen hinterlegt sein.

7.4 Firewall

In unserer Konfiguration haben wir die Firewall komplett deaktiviert. Dies sollte aber nur dann erfolgen, wenn es sich um ein privates Netzwerk handelt, das keine Verbindung ins Internet, Intranet oder ähnliches hat.

Darüber hinaus besteht die Möglichkeit, einen Port der Firewall zu öffnen, damit der Client mit dem Server anhand der yp-Dienste kommunizieren kann.

7.5 *Einschränkungen von NIS / KDE*

Bei der Verwendung von NIS gibt es Einschränkungen. Bei der Anmeldung auf der KDE –Oberfläche werden diverse Fehlermeldungen ausgegeben und der Anmelde-Vorgang wird abgebrochen. Die Ursache liegt darin, das verschiedene Files und vor allem das fehlende Homeverzeichnis des anzumeldenden Users nicht vorhanden ist.

Abhilfe schafft hier der Einsatz eines NFS–Servers, welcher die Verzeichnisse der User zur Verfügung stellt und so die Anmeldung an der KDE ermöglicht.

8 Sicherheit

Im Zuge unserer Aufgabenstellung haben wir uns nicht mit dem Thema Sicherheit befasst. Da wir aber gerne eine Übersicht über alle Belange von NIS geben wollten, haben wir im gesamten Punkt 8 (inkl. Unterpunkte) einen Auszug aus dem Artikel „NIS für Linux“ von Thorsten Kukuk (erschienen im Linux-Magazin 09/1998) eingefügt. Der genaue Link ist in der Quellenangabe zu finden.

Aktuell wird das Verfahren LDAP und Kerberos eingesetzt (s. Stichwortverzeichnis), die in Punkto Sicherheit dem NIS Verfahren überlegen sind!

„.....Ein oft vernachlässigtes Problem mit NIS ist dessen Unsicherheit. Jeder User kann sich mit *ypcat passwd.byname* die User-Daten und verschlüsselten Passwörter ansehen.

Dies hat den gleichen Effekt, wie ein *cat /etc/passwd* auf Systemen ohne NIS. Auf diesen Rechnern wurde das Problem mit der Einführung von Shadow-Passwörtern behoben. Mit NIS ist dies nicht ganz so einfach. Jeder im Internet, der den Domainnamen errät und weiß, auf welchem Rechner *ypserv* läuft, kann sich die Map anzeigen lassen.

Dieses Problem existiert nicht, wenn die *passwd*-Datei nur auf jedem Rechner lokal vorliegt. Dafür gibt es eine zweigleisige Lösung: Zum einen wird ein spezieller Portmapper installiert, der nur Daten an freigegebene Rechner schickt. Außerdem verhindert er, das spezielle Anfragen an den *ypserv*-Prozeß weitergereicht werden.

Diese Anfragen kommen im normalen Betrieb nicht vor, und werden gerne von Hackern benutzt. Dazu gehört zum Beispiel die Broadcast-Suche mit *rpcinfo -b 100004 2* . Eigentlich sollten sich daraufhin alle Rechner melden, auf denen ein *ypserv* Prozeß läuft. Da *ypbind* aber eine andere RPC-Funktion benutzt, die einen Domainnamen mitliefert und nur eine Antwort bekommt, wenn der Server die Domain unterstützt, ist diese Eigenschaft nicht notwendig. Damit bekommen Unbefugte nur heraus, wo alles ein NIS Server läuft.

ypserv selber besitzt auch eine Methode, um Hackern das Leben schwerer zu machen. Alle *ypserv*-Implementierungen unterstützen das *securenets*-Feature. In dieser Datei steht, welche Subnetze und Rechner als sicher gelten und Anfragen stellen dürfen.

Ein typisches */var/yp/securenets* File sieht z.B. folgendermaßen aus:

/var/yp/securenets

```
#
# securenets    This file defines the access rights to your NIS server
#               for NIS clients. This file contains netmask/network
#               pairs. A clients IP address needs to match with at least
#               one of those.
#
#               One can use the word "host" instead of a netmask of
#               255.255.255.255. Only IP addresses are allowed in this
#               file, not hostnames.
#
# Always allow access for localhost
255.0.0.0       127.0.0.0

# Unsere beiden Subnetze:
255.255.254.0   10.234.223.0
255.255.254.0   10.234.214.0
# Die Server duerfen auch mit dem Backbone Interface:
host            10.234.128.228
host            10.234.128.229
```

Mein Linux *ypserv*-Paket unterstützt außerdem noch die *tcp_wrapper* Library. Und es können spezielle Regeln in einer Konfigurationsdatei angegeben werden, die festlegen, in welchem Fall die Daten aus den Maps wie verändert werden müssen, bevor sie an den Client geschickt werden dürfen.

Dabei darf nie vergessen werden, das NIS die über das Netz verschickten Daten nicht verschlüsselt, und auch keine Form der User-Authentifizierung vorgenommen wird. Dadurch gibt es jede Menge Angriffspunkte.“

8.1 Shadow Passwörter und NIS

„Die Solaris- und meine Linux-Variante von *ypserv* unterstützen Shadow-Support. Da *ypserv* selber keine Ahnung hat, was in welcher Map steht, kann jeder *ypserv* damit "Nachgerüstet" werden. Es müssen nur die entsprechenden Regeln im */var/yp/Makefile* hinzugefügt werden.

Allgemein muss gesagt werden, dass Shadow-Passwörter über NIS nicht sehr viel Sinn, dafür aber umso mehr Probleme machen. Vor allem sind sie nicht sicherer, es kann höchstens Hackern das Leben schwerer gemacht werden. Dazu kommt, dass dies von der C-Library unterstützt werden muss.

Unter Linux sieht es so aus, das die Linux *libc* 5 dieses nicht unterstützt, und somit Shadow-Passwörter über NIS nicht funktionieren. In den News lese ich dazu immer häufiger, dass es daran liegt, das *ypcat shadow* fehlschlägt.

pcat gibt die Fehlermeldung aus, dass die Map *shadow* nicht existiert. Diese Fehlermeldung ist natürlich richtig, hat aber nichts mit dem Problem zu tun.

Es gibt keine Map *shadow*, sie heißt *shadow.byname*. Und da es sich um keine Standard-NIS-Map handelt, gibt es kein Alias *shadow* für *shadow.byname*, so wie es ein Alias *passwd* für *passwd.byname* gibt. Das Problem ist ein anderes: Der *shadow*-Support in der *libc5* hat keine Ahnung von NIS und benutzt es nicht. Eine *libc5* mit NYS-Support eincompiliert unterstützt Shadow Passwörter über NIS. Ich rate aber davon ab, diese *libc* zu benutzen. Der eingebaute *ypbind*-Code ist zwar theoretisch sehr schön, macht aber auch viele Probleme.

Außerdem enthält der Shadow Parser-Code einige Fehler, womit nicht alle korrekten Shadow-Einträge als solche erkannt werden. Anders sieht das mit der neuen GNU C-Library 2 aus. Diese hat einen vollständigen Support für Shadow-Passwörter über NIS und bereitet in dieser Hinsicht keine Probleme.

Man muss nur bedenken, dass alte, mit *libc5* kompilierte Programme nichts davon haben. Es sollte also nur benutzt werden, wenn alle Programme mit der *glibc* kompiliert wurden, oder die *libc5*-Programme nicht auf die Passwörter zugreifen müssen.

Jeder, der Shadow Passwörter über NIS benutzt, gibt damit die Sicherheitsvorteile vom Shadow-System auf. Daher sollte es für normale User nur benutzt werden, wenn die anderen Features von Shadow benötigt werden, wie *login expire* oder das Erzwingen eines neuen Passworts.

Ansonsten sollten die Passwörter der NIS-User von der Shadow-Datei in die *passwd*-Datei kopiert werden, um auf die Shadow-Map zu verzichten. Das hat außerdem den Vorteil, dass die C-Library der Clients keinen Shadow-Support benötigt.

Es gibt natürlich auch Methoden, um es schwieriger zu machen, an die verschlüsselten Passwörter zu kommen. Eine von allen Implementierungen unterstützte Methode ist die Abfrage der Port-Nummer, von der die Anfrage kommt.

Die Anfrage wird nur beantwortet, wenn die Port-Nummer kleiner als 1024 und somit von einem reservierten Port kommt. Das hat den Hintergrund, das nur *root* Ports kleiner 1024 benutzen darf, aber kein normaler User.

Dieses kann mit dem "-s" Switch von *makedbm* erzwungen werden. Das Linux *ypserv*->Programm hat noch eine */etc/ypserv.conf* Konfigurationsdatei, in der genauere Regeln eingetragen werden. Diese sollte immer den Vorzug erhalten, da sie bedeutend schneller sind. Ausserdem kann damit auch bestimmt werden, das z.B. nur *root* auf einem Client die Passwörter sieht, aber kein normaler User. Diese sehen nur ein "x". Das ist genauso sicher wie Shadow-Passwörter über NIS, ist aber bedeutend einfacher und unproblematischer. Wenn diese Funktion benutzt wird, darf natürlich im Netz kein *ypserv* laufen, der dieses Feature nicht besitzt.

Zusätzlich ist zu beachten, welche Rechner auf den *ypserv* zugreifen dürfen. DOS- oder Windows-Rechner sollten niemals auf NIS zugreifen dürfen. Unter diesen Betriebssystemen gibt es keine *root*-Rechte, somit kann jeder User Ports kleiner 1024 benutzen.

Außerdem werden die Daten nicht verschlüsselt übertragen, somit kann jeder am Netz horchen und auf die Daten warten. Es ist also sehr wichtig, sich gut zu überlegen, was in der */var/yp/securenets*-Datei steht.“

8.2 Alternativen zu NIS

„NIS ist ein Dienst, der unter jedem Unix mehr oder weniger gut implementiert ist, und daher vor allem in heterogenen Netzwerken kaum Probleme bereitet. Er ist deshalb sehr beliebt und wird fast überall eingesetzt. Er ist aber auch schon sehr alt und unsicher und vermisst einige grundlegende Sicherheitsmechanismen, wie gerade am Beispiel mit Shadow-Passwörtern gut zu sehen ist.

Alternativen gibt es kaum. Sun hat mit Solaris 2.x NIS+ eingeführt, was Secure RPC benutzt. Es ist aber langsam und schwer zu administrieren. Und es wird bisher nur von Solaris unterstützt. Das nächste Betriebssystem, welches NIS+ unterstützt, wird, welch Wunder, Linux sein. Ich habe zwar schon von anderen Unix-Herstellern gehört, das diese NIS+ unterstützen wollen, von lauffähigen Implementierungen habe ich aber noch nichts gesehen oder gehört.

Ich habe vor über einem Jahr damit angefangen, den NIS+ Client Support für Linux zu implementieren. Für *libc5* gibt es Patches, die GNU C-Library 2.1 wird NIS+ defaultmäßig unterstützen. Ansonsten wird versucht, *Hesoid* und *LDAP* als Ersatz zu benutzen, der Support durch die Betriebssysteme ist aber noch schlechter.

Eine Alternative könnte die Secure Shell sein. Das Prinzip dabei ist, dass ein Rechner aufgesetzt wird, der die Master */etc/passwd* und */etc/shadow* Datei enthält. Regelmäßig werden diese Dateien mit *scp* verschlüsselt über das Netz auf alle anderen Rechner verteilt. Der Nachteil ist, dass jeder User sich auf dem Master einloggen muss um dort das Passwort zu ändern. Und es dauert, bis die Änderungen verteilt werden. Änderungen auf lokalen Rechnern gehen dabei verloren.“

9 Anhang

NIS-Befehle

Domainname: zeigt oder ändert (abhängig von Parametern) den NIS-Domänen-Name

ypcat: gibt eine Datenbankliste aus

Beispiel:

ypcat passwd
Ausgabe aller User in dem Server-Map inkl. Ihrer ID- und Home-Verzeichnisse

ypmatch: zeigt die Werte eines oder mehrerer Schlüssel

Beispiel:

ypmatch –x
Ausgabe aller benutzten Nicknames (Namen der Maps)

ypwhich gibt den Namen des NIS-Servers zurück, der die Datenbank bereitstellt

Beispiel:

ypwhich
Ausgabe fire

ypserv Dämon auf dem YP Datenbankserver. Hier wird die eigentliche Arbeit geleistet

ypbind Dämon, stellt die Verbindung mit dem YP Datenbankserver her, indem er sich an ihn bindet

yppasswdd Dämon, welcher eine Änderung von Passwd-Daten von Clients aus zulässt (wurde in unserer Funktionskontrolle nicht berücksichtigt). Dieser Dienst muss zum Einsatz auf dem Server gestartet werden.

9.1 Stichwortverzeichnis

IP-Adresse

Das **Internet Protocol** (**IP**) (auch *Internetprotokoll*) ist ein in Computernetzen weit verbreitetes Netzwerkprotokoll. Es ist eine (bzw. *die*) Implementierung der *Internet*-Schicht des TCP/IP-Modells bzw. der *Vermittlungs*-Schicht des OSI-Modells.

IP bildet die erste vom Übertragungsmedium unabhängige Schicht der Internet-Protokoll-Familie. Das bedeutet, dass mittels IP-Adresse und Subnetzmaske (*subnet mask*) Computer innerhalb eines Netzwerkes in logische Einheiten, so genannte Subnetze, gruppiert werden können. Auf dieser Basis ist es möglich, Computer in größeren Netzwerken zu adressieren und Verbindungen zu ihnen aufzubauen, da logische Adressierung die Grundlage für Routing (Wegewahl und Weiterleitung von Netzwerk-Paketen) ist. IP stellt die Grundlage des Internets dar.

http://de.wikipedia.org/wiki/Ip

RPC

Remote Procedure Call, oder kurz **RPC**, ist ein Netzwerkprotokoll auf der fünften, teilweise auch sechsten Schicht des ISO/OSI-Modells. Mit Hilfe von RPC können über ein Netzwerk Funktionsaufrufe auf entfernten Rechnern durchgeführt werden.

RPC wurde ursprünglich durch Sun Microsystems für NFS entwickelt. Der genaue Aufbau von RPC wird in den RFCs 1057 und 1831 beschrieben. Die Idee hinter RPC beruht auf dem Client-Server-Modell, es sollte die gemeinsame Nutzung von Programmfunktionen über Rechnergrenzen ermöglichen. Ein RPC-Aufruf läuft fast immer synchron ab, das heißt, dass der aufrufende Client mit der Ausführung des weiteren Programmcodes wartet bis er eine Antwort der Prozedur vom Server erhalten hat.

Es lassen sich weiterhin drei inkompatible Versionen von RPC unterscheiden, das ONC RPC, das vielfach auch als Sun RPC bezeichnet wird, ist die am weitesten verbreitete RPC Variante. ONC RPC steht hierbei für Open Network Computing Remote Procedure Call, für diese RPC Variante findet sich unter anderem auch eine Implementierung in Linux.

Die wichtigste Komponente auf der Serverseite ist der Portmapper-Daemon, der auf den UDP- und TCP-Port 111 lauscht. Der Portmapper übernimmt die Koordination der durch den Client gewünschten Funktionsaufrufe. Jedes Programm, das auf dem Server RPC-Dienste zur Verfügung stellen will, muss daher dem Portmapper bekannt sein.

Neben NFS (Network File System) basiert unter anderem noch NIS (Network Information Service) in weiten Teilen auf RPC-Aufrufen.

http://de.wikipedia.org/wiki/Remote_Procedure_Call

Shell-Console

Ein **Kommandozeileninterpreter**, auch *Kommandointerpreter* oder (ungenau) *Shell* genannt, ist ein Computerprogramm, welches eine Zeile Text in der Kommandozeile einliest, diesen Text als Kommando interpretiert und ausführt, z.B. durch Starten weiterer Programme.

http://de.wikipedia.org/wiki/Unix-Shell

TCP

Das *Transmission Control Protocol* (**TCP**) ist eine Vereinbarung (Protokoll) darüber, auf welche Art und Weise Daten zwischen Computern ausgetauscht werden sollen. Alle am Datenaustausch beteiligten Computer kennen diese Vereinbarungen und befolgen sie. Es ist damit ein zuverlässiges, verbindungsorientiertes Transportprotokoll in Computernetzwerken. Es ist Teil der TCP/IP-Protokollfamilie.!

http://de.wikipedia.org/wiki/Transmission_Control_Protocol

UDP

Das *User Datagram Protocol* (**UDP**) ist ein minimales, verbindungsloses Netzprotokoll. Es gehört zur Transportschicht der TCP/IP-Protokollfamilie und ist im Gegensatz zu TCP nicht auf Zuverlässigkeit ausgelegt.

UDP erfüllt im Wesentlichen den Zweck, die durch die IP-Schicht hergestellte Endsystemverbindung um eine Anwendungsschnittstelle (Ports) zu erweitern. Die Qualität der darunter liegenden Dienste, insbesondere die Zuverlässigkeit der Übertragung, erhöht UDP hingegen nicht.

http://de.wikipedia.org/wiki/UDP

YaSt

Yet another Setup Tool (englisch für „Noch ein weiteres Installationswerkzeug“) ist ein betriebssystemweites Installations- und Konfigurationswerkzeug, welches in der SuSE-Linux-Distribution zum Einsatz kommt. Darüber hinaus ist es Bestandteil von United Linux.

http://de.wikipedia.org/wiki/YaST

NIS

Network Information Service (auch Network Information System) oder kurz NIS ist ein Verzeichnisdienst zur Verteilung von Konfigurationsdaten wie Benutzernamen oder Rechnernamen in einem Computernetzwerk. NIS entstand bei Sun Microsystems als "Yellow Pages" (YP) Client-Server Protokoll.

NIS/YP kann als zentrales Verzeichnis für Benutzeraccounts, Rechnernamen und andere brauchbare Daten in einem Computernetzwerk verwendet werden. Zum Beispiel wird in einer üblichen UNIX-Umgebung die Liste der Benutzer (für die Authentifizierung) in der Datei /etc/passwd untergebracht. Durch das Benutzen von NIS wird eine weitere "globale" Benutzerliste hinzugefügt die zur Authentizifizierung der Benutzer auf jedem Rechner verwendet werden kann.

Sun hat die Lizenz zu dieser Technik an praktisch alle Unix-Hersteller vergeben.

Der Name Yellow Pages (deutsch: Gelbe Seiten) ist in Großbritannien eine registrierte Marke der Britisch Telekomunications für ihr Branchentelefonbuch. Sun änderte daher den Namen des Systems zu NIS, obwohl alle Kommandos und Funktionen immer noch mit "yp" anfangen.

In modernen Rechnernetzen ersetzen Systeme wie LDAP und Kerberos zunehmend NIS, da sie gegenüber NIS als weitgehend moderner und sicherer angesehen werden.

http://de.wikipedia.org/wiki/Network_Information_Service

LDAP

Das **Lightweight Directory Access Protocol** (**LDAP**) ist in der Computertechnik ein Netzwerkprotokoll, das die Abfrage und die Modifikation von Informationen eines Verzeichnisdienstes (eine im Netzwerk verteilte hierarchische Datenbank) erlaubt. Die aktuelle Version ist in RFC 2251 spezifiziert.

http://de.wikipedia.org/wiki/LDAP

NFS

Der **Network File Service** - abgekürzt **NFS** (früher: Network File System) - ist ein von Sun Microsystems entwickeltes Protokoll, das den Zugriff auf Dateien über ein Netzwerk ermöglicht. Dabei werden die Dateien nicht (wie z.B. bei FTP) übertragen, sondern die Benutzer können auf Dateien, die sich auf einem entfernten Rechner befinden, so zugreifen, als wenn sie auf ihrer lokalen Festplatte abgespeichert wären.

http://de.wikipedia.org/wiki/Network_File_System

FTP

Das **File Transfer Protocol** (engl. für "Dateiübertragungsverfahren", kurz **FTP**), ist ein in RFC 959 spezifiziertes Netzwerkprotokoll zur Dateiübertragung über TCP/IP-Netzwerke. FTP ist in der Anwendungsschicht des TCP/IP Protokollstapels angesiedelt.

Es wird benutzt, um Dateien vom Server zum Client (Download), vom Client zum Server (Upload) oder clientgesteuert zwischen zwei Servern zu übertragen. Neben dem File Transfer Protocol (FTP) gibt es auch noch das IBM Transfer Protocol, welches die Verbindung von PC zu Mainframe Umgebungen ermöglicht.

http://de.wikipedia.org/wiki/File_Transfer_Protocol

RFC

Die **Requests for Comments** (kurz *RFC*; zu deutsch etwa *Bitten um Kommentare*) sind eine Reihe von technischen und organisatorischen Dokumenten des RFC-Editor zum Internet (ursprünglich ARPANET), die am 7. April 1969 begonnen wurden. Bei der ersten Veröffentlichung noch im ursprünglichen Wortsinne zur Diskussion gestellt, behalten RFC auch dann ihren Namen, wenn sie sich durch allgemeine Akzeptanz und Gebrauch zum Standard entwickelt haben.

http://de.wikipedia.org/wiki/Request_for_Comments

Kerberos

Kerberos ist ein verteilter Authentifizierungsdienst (Netzwerkprotokoll) zur Authentifizierung, der für offene und unsichere Computernetze (z. B. das Internet) von Steve Miller und Clifford Neuman entwickelt wurde. Die zurzeit aktuelle Version ist Kerberos5. Die Version 5 des Protokolls ist, im Gegensatz zu Version 4, in ASN.1 definiert. Kerberos5 wird in RFC 4120 definiert.

Kerberos bietet sichere und einheitliche Authentisierung in einem ungesicherten TCP/IP-Netzwerk aus sicheren Hostrechnern. Die Authentisierung übernimmt eine *vertrauenswürdige dritte Partei*. Diese dritte Partei ist ein besonders geschützter Kerberos 5-Netzwerkdienst. Kerberos unterstützt Single Sign On, d. h. ein Benutzer muss sich nur noch einmal anmelden, dann kann er alle Netzwerkdienste nutzen, ohne ein weiteres Mal ein Passwort eingeben zu müssen. Kerberos übernimmt die weitere Authentifizierung.

Der Name leitet sich vom Höllenhund Kerberos aus der griechischen Mythologie ab, der den Eingang zur Unterwelt bewacht.

http://de.wikipedia.org/wiki/Kerberos_%28Informatik%29

NIS-Domäne

Ein lokales Computernetz kann logisch in verschiedene „NIS-Zonen“ eingeteilt werden, wobei jede Zone eine eigene Datenbank verwendet. Diese Zonen werden *NIS-Domänen* genannt. Jede Domäne benötigt ihren eigenen NIS-Master-Server, wobei es jedoch keine Einschränkungen gibt, welche Maschine Master für welche Domäne sein muss. Beispielsweise können Sie einen Rechner einrichten, der zwei NIS-Master-Server für zwei unterschiedliche Domänen beherbergt. Es muss nur sichergestellt sein, dass jeder Client weiß, zu welcher Domäne er gehört. Aus diesem Grunde benötigt jede Domäne einen *NIS-Domänenamen*, der eine eindeutige Zuordnung erlaubt. Ein Name wird immer benötigt, auch wenn nur eine einzige Domäne im Netz besteht.

Ein NIS-Client verwendet den NIS-Domänenamen, um einen geeigneten NIS-Maser-Server zu finden. Dies passiert üblicherweise beim Systemstart, wobei eine Verbindung zur Server-Datenbank aufgebaut wird. Dieser Prozess wird als „Binden an die NIS-Domäne“ bezeichnet.

http://www.bresink.de/osx/nisOLD-de.html

KDE

KDE (**K D**esktop **E**nvironment, ursprünglich *K*ool *D*esktop *E*nvironment, heute hat das *K* keine bestimmte Bedeutung mehr) ist ein frei verfügbarer Desktop, das heißt eine grafische Benutzeroberfläche mit vielen Zusatzprogrammen für den täglichen Gebrauch. Er ist vorrangig für Computer gemacht, auf denen ein Unix-Betriebssystem läuft, wie z. B. GNU/Linux, BSD oder Solaris. Über Cygwin kann KDE auch unter Windows betrieben werden.

KDE wird unter anderem in folgenden Distributionen als Standardarbeitsfläche eingesetzt: Mandriva (Mandrake), Kubuntu und Knoppix. Neben GNOME in SUSE LINUX, Debian, Fedora Core, Gentoo Linux und Slackware.

http://de.wikipedia.org/wiki/KDE

10 Quellen- Nachweis

Anbei finden Sie alle Quellen, auf die wir uns berufen. Die Datumsangabe in Klammer zeigt den letzten Zugriff auf die entsprechende Internet-Seite.

http://www.bresink.de/osx/nisOLD-de.html (14.09.05)

http://www.linux-magazin.de/Artikel/ausgabe/1998/09/NIS/nis.html (07.08.05)

http://www.linuxfibel.de/printversion/nis_srv.htm (26.06.05)

http://www.unet.univie.ac.at/aix/cmds/aixcmds6/ypbind.htm (14.09.05)

http://www.sun3zoo.de/de/yp.html#01 (21.09.05)

http://de.wikipedia.org/wiki/Ip (09.10.2005)

http://de.wikipedia.org/wiki/Remote_Procedure_Call (05.10.2005)

http://de.wikipedia.org/wiki/File_Transfer_Protocol (05.10.2005)

http://de.wikipedia.org/wiki/Kerberos_%28Informatik%29 (05.10.2005)

http://de.wikipedia.org/wiki/Network_File_System (05.10.2005)

http://de.wikipedia.org/wiki/LDAP (05.10.2005)

http://de.wikipedia.org/wiki/Network_Information_Service (05.10.2005)

http://de.wikipedia.org/wiki/YaST (05.10.2005)

http://de.wikipedia.org/wiki/UDP (05.10.2005)

http://de.wikipedia.org/wiki/Transmission_Control_Protocol (05.10.2005)

http://de.wikipedia.org/wiki/Unix-Shell (05.10.2005)

http://de.wikipedia.org/wiki/KDE (09.10.2005)

http://de.wikipedia.org/wiki/Request_for_Comments (11.10.2005)