Open Source Teil 2 – Linux für Fortgeschrittene und Administratoren

Lukas Feiler

lukas.feiler@lukasfeiler.com

NFS-Server

Allgemeines

Bei NFS-Servern handelt es sich um die serverseitige Implementierung des *Network File System* (NFS). NFS wurde ursprünglich von Sun Microsystems (http://www.sun.com) entwickelt. Die erste öffentliche Release erfolgte im Jahre 1985 und wurde als NFS Version 2 bezeichnet[1].

Heute kommt in nahezu allen Umgebungen NFS Version 3 zum Einsatz. Erste Implementierungen von NFS Version 4 sind jedoch bereits erhältlich. Version 4 ist in RFC 3010 spezifiziert und wird wesentlich zur Verbesserung der Sicherheit von NFS beitragen.

NFS setzt (auch in Version 3) grundsätzlich auf dem User Datagram Protocol (UDP) auf. Dieses kann – im Unterschied zu TCP – nicht gewährleisten, dass alle versendeten Datenpakete auch tatsächlich den Empfänger erreichen. Da dies für die Funktionsfähigkeit von NFS jedoch erforderlich ist, wurde in NFS eine entsprechende Funktionalität implementiert. Da aber eben diese Funktionalität bereits wesentlich effizienter von TCP zur Verfügung gestellt wird, verwenden manche Implementierungen von NFS bereits TCP (so auch NFS Version 4).

NFS ist grundsätzlich ein zustandsloses Protokoll, da zur Funktionsfähigkeit eines NFS-Servers nicht erforderlich ist, dass dieser Daten über die, mit ihm verbundenen Clients speichert. Bei jedem Request teilt der Client dem NFS-Server alle Informationen mit, die zur Abarbeitung des Requests erforderlich sind. Ein Request enthält daher grundsätzlich die ID des Users, ein sog. File Handle und die Information welche Operation mit dem File Handle ausgeführt werden soll.

Ob der, den Request sendende Benutzer Zugriff erhält ist durch die vom Client übertragene User-ID determiniert. Hat der Benutzer am Client-System beispielsweise die User-ID 501 so erhält er nur dann Zugriff, wenn es am NFS-Server die User-ID 501 gibt und diese Zugriff auf das bezeichnete File hat. Hieraus ergeben sich zwei sicherheitstechnische Probleme: erstens kann es sein, dass die User-ID des Benutzers am NFS-Server nicht diesem Benutzer zugewiesen wurde sondern für andere Zwecke verwendet wird. Zweitens bietet NFS grundsätzlich keine Möglichkeit die, im Request enthaltene User-ID zu überprüfen. Es entsteht daher eine sog. Trust-Relationship zwischen dem NFS-Server und dem NFS-Client. Denn wenn es das Betriebssystem des NFS-Clients ermöglicht Requests mit gefälschten User-IDs zu versenden, so kann hierdurch Zugriff auf den NFS-Server erlangt werden. Der NFS-Server ist daher stets nur so sicher wie die NFS-Clients.

[1] NFS Version 1 wurde hingegen niemals veröffentlicht.

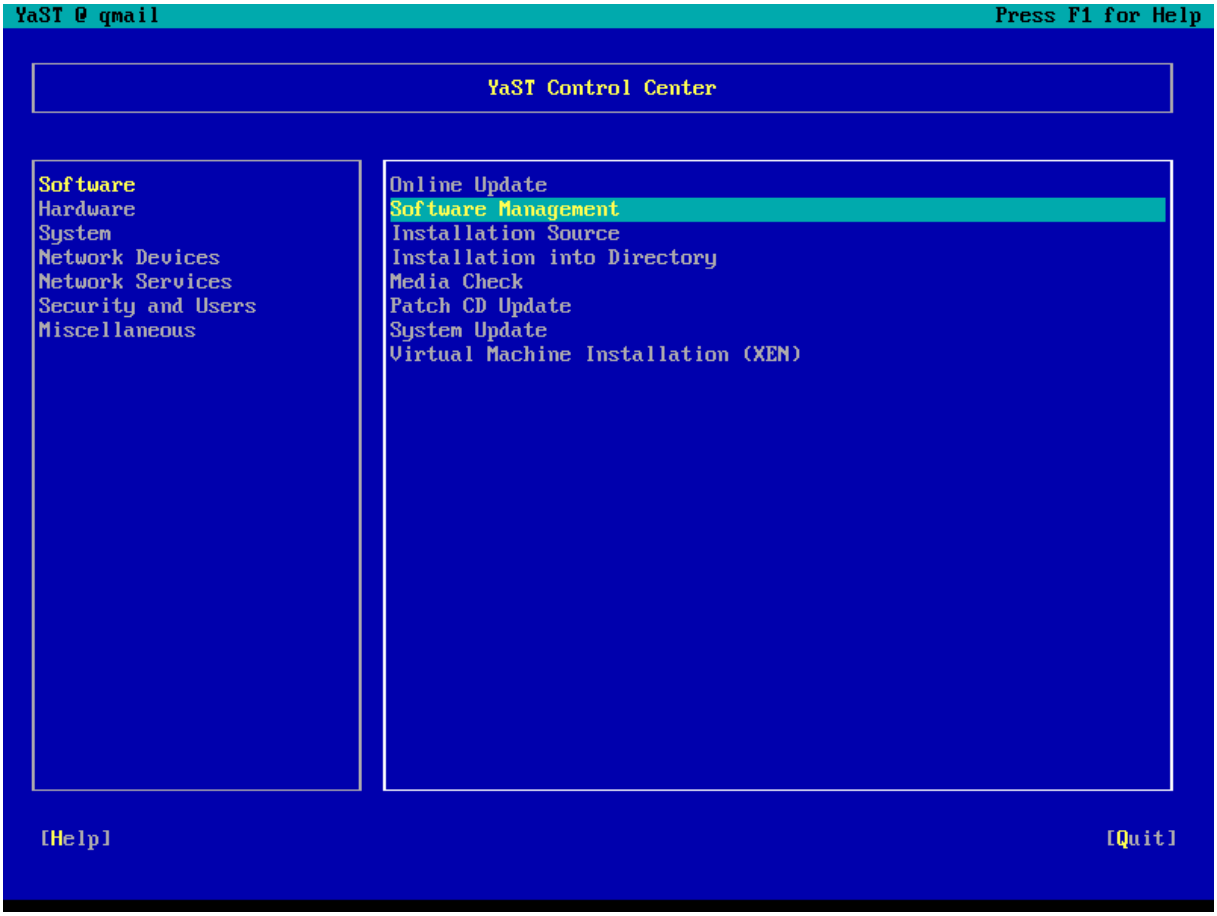
Wie bereits ausgeführt ist in einem Request neben der User-ID stets eine NFS File Handle enthalten. Ein File Handle besteht va. aus einer Inode-Number, die es ermöglicht eine Datei eindeutig zu identifizieren. Manche Angriffsarten bestehen daher – neben der Verwendung einer gültigen User-ID – im Erraten gültiger Inode-Numbers.

Installation eines NFS-Servers unter SUSE Linux 10

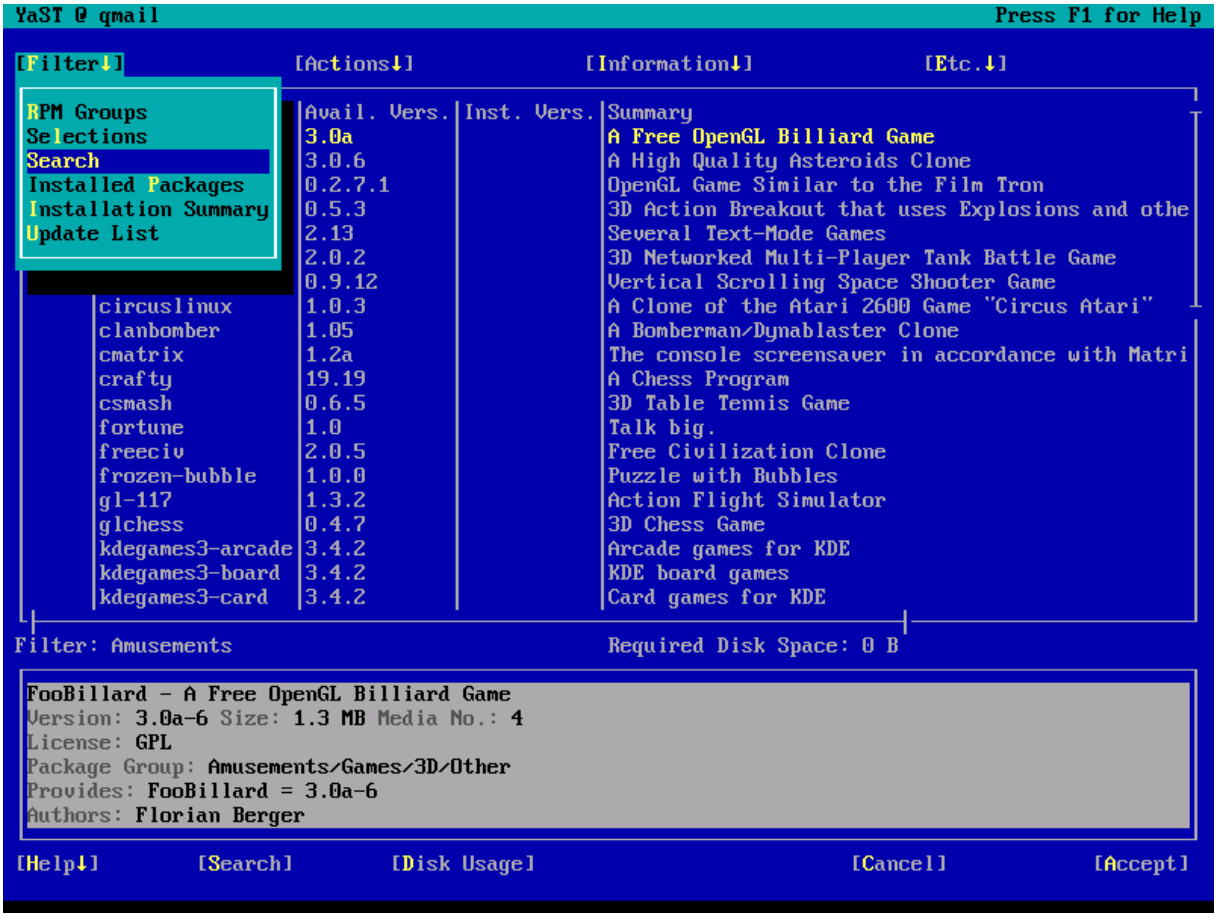
Da unter Linux ein Großteil der Implementierung eines NFS-Servers bereits im Kernel erfolgt ist, sind nur wenige Pakete zu Installieren. Diese sind:

- nfs-utils
- nfsidmap
- yast2-nfs-client
- yast2-nfs-server

Am leichtesten geht die Installation dieser Pakete mit YAST von der Hand. Als *root* ist daher zunächst der Befehl *yast* auf der Command-Line einzugeben. Unter der, im linken Frame gelisteten Kategorie „Software“ findet sich das YAST-Modul „Software Management“:

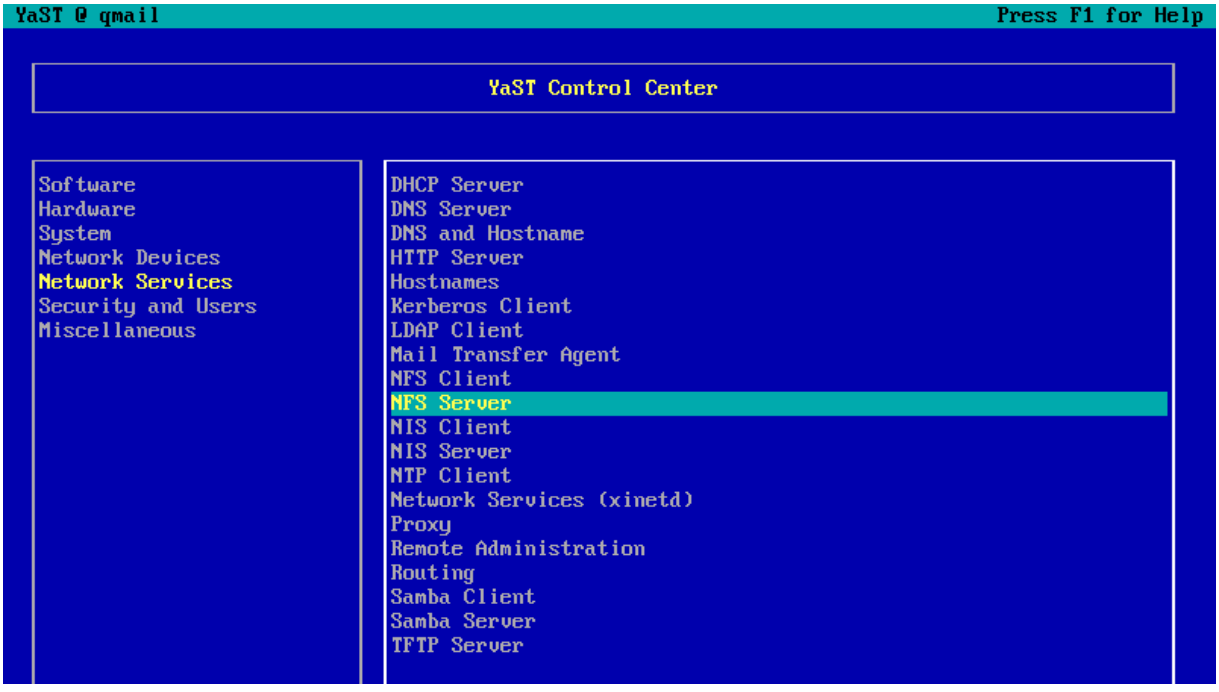


Über den Menüpunkt „Filter“ lässt sich mit „Search“ nach allen Paketen suchen, die die Zeichenkette „nfs“ im Namen enthalten:

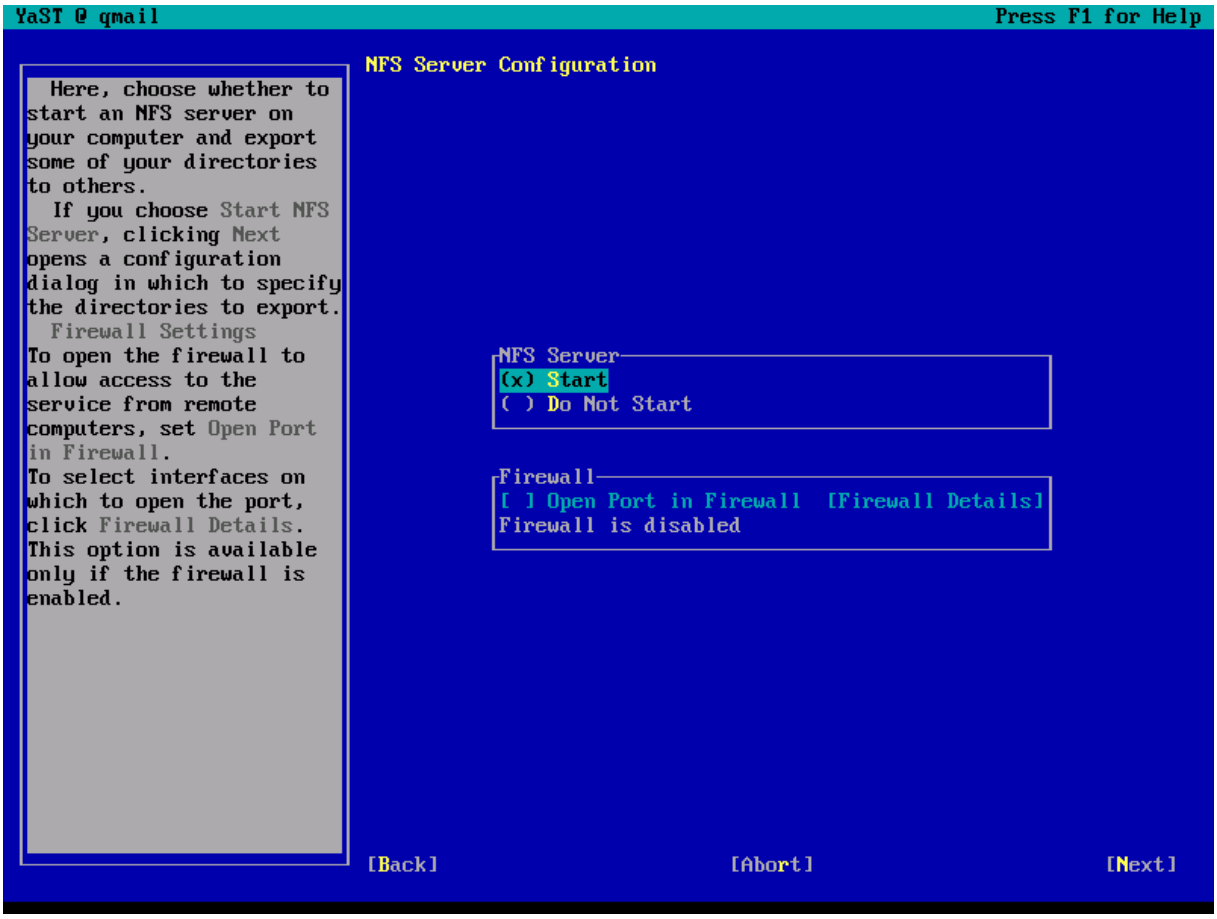


Konfiguration & Verwaltung eines NFS-Servers unter SUSE Linux 10

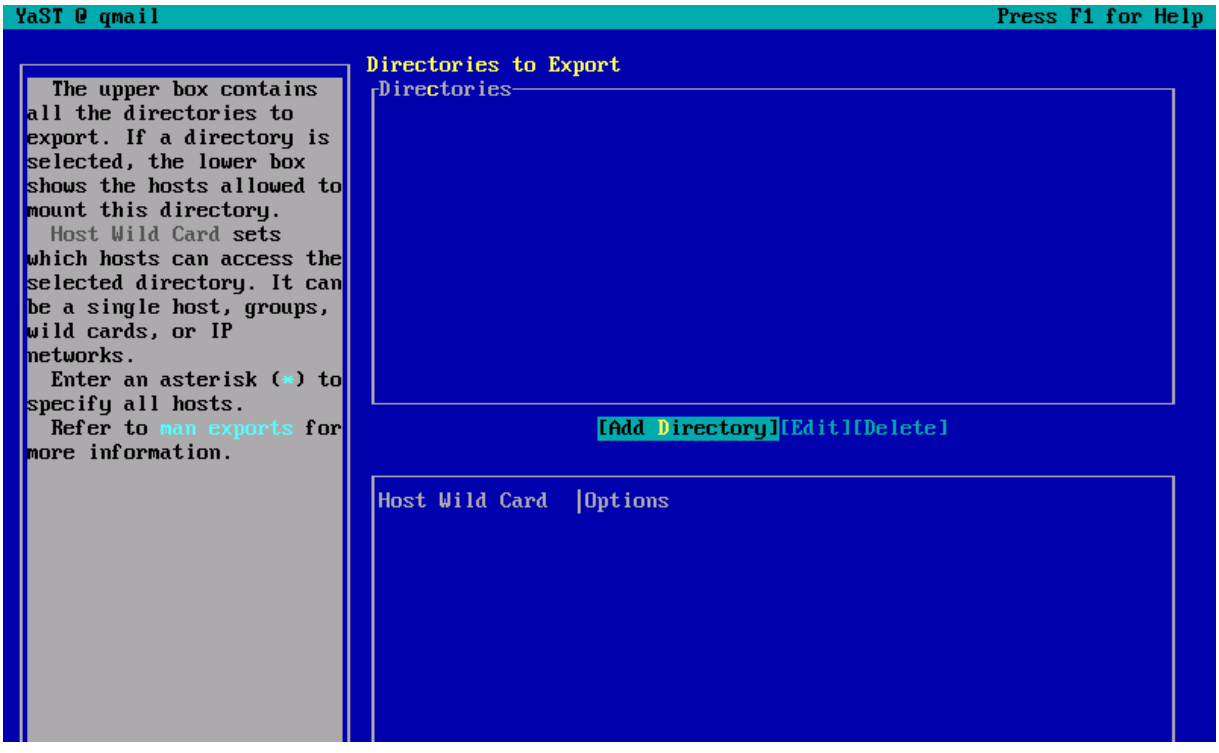
Nach erfolgter Installation der Pakte und einem Neustart von YAST, sollten sich nun unter der Kategorie „Network Services“ die Module „NFS Client“ und „NFS Server“ befinden:



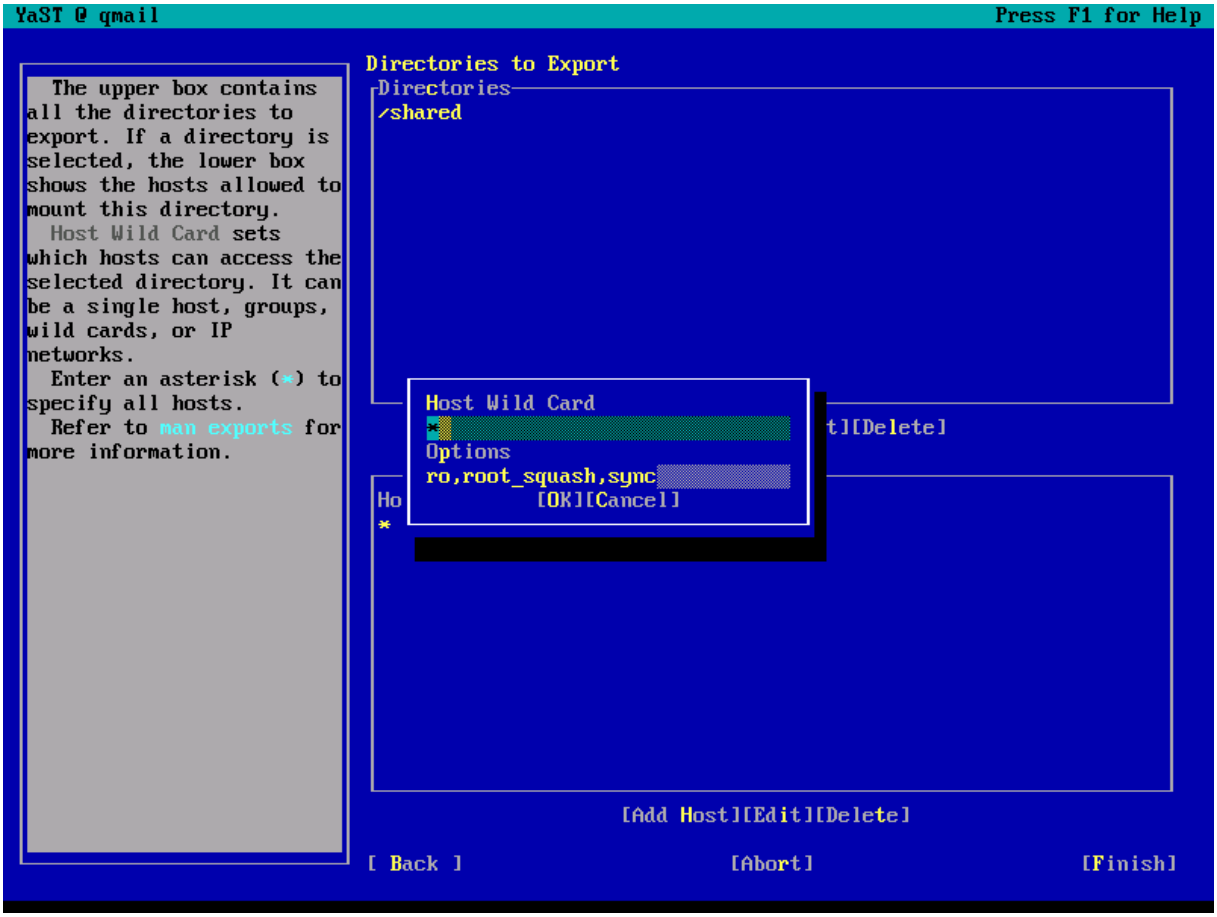
Zunächst empfiehlt es sich den NFS-Server automatisch beim Start des Betriebssystems zu starten:



Nun können einzelne Verzeichnisse per NFS anderen Computern zur Verfügung gestellt werden. Man bezeichnet dies als „exportieren“ eines Verzeichnisses bzw. das exportierte Verzeichnis schlicht als „Export“. Über „Add Directory“ kann ein neues Verzeichnis exportiert werden:



Wird beispielsweise das (zuvor angelegte) Verzeichnis „/shared“ exportiert, präsentiert YAST folgende Eingabemaske:



Es sind hier für das exportierte Verzeichnis „/shared“ zwei Eingaben zu tätigen:

- **Host Wild Card**

  Hiermit kann angegeben werden welche Computersysteme Zugriff auf dieses Verzeichnis haben sollen. Der Default „*“ (d.h. Zugriff für alle Computersysteme) sollte in Produktivumgebungen niemals verwendet werden! Es sollten explizit jene Systeme aufgelistet werden, die Zugriff auf das Verzeichnis benötigen – es gilt das Prinzip Least Privilege.

  Eine Eingabe könnte daher lauten:

  - *„192.168.1.66:192.168.1.67“*: ermöglicht nur diesen beiden IP-Adressen den Zugriff
  - *„192.168.1.0/255.255.255.0“*: ermöglicht dem Netzwerk 192.168.1.0 (d.h. allen IP-Adressen, die mit 192.168.1. beginnen) den Zugriff.
  - *„*.ny.example.com“*: ermöglicht den Zugriff für alle IP-Adressen, die sich in einen Domainnamen auflösen lassen, der mit .ny.example.com endet.

- **Options**

  Die Options eines Exports sind sowohl für Funktionalität als auch für die Sicherheit des NFS-Servers von größter Wichtigkeit. Die Default-Optionen „ro,root_squash,sync“ bieten ein meist angemessenen Grad an Sicherheit:

  - *ro:* Diese Option bewirkt, dass das Verzeichnis nur read-only (d.h. ohne die Möglichkeit schreibend darauf zuzugreifen) exportiert wird. Selbst wenn ein Benutzer Schreibrechte für eine konkrete Datei haben sollte, kann er nur lesend zugreifen.

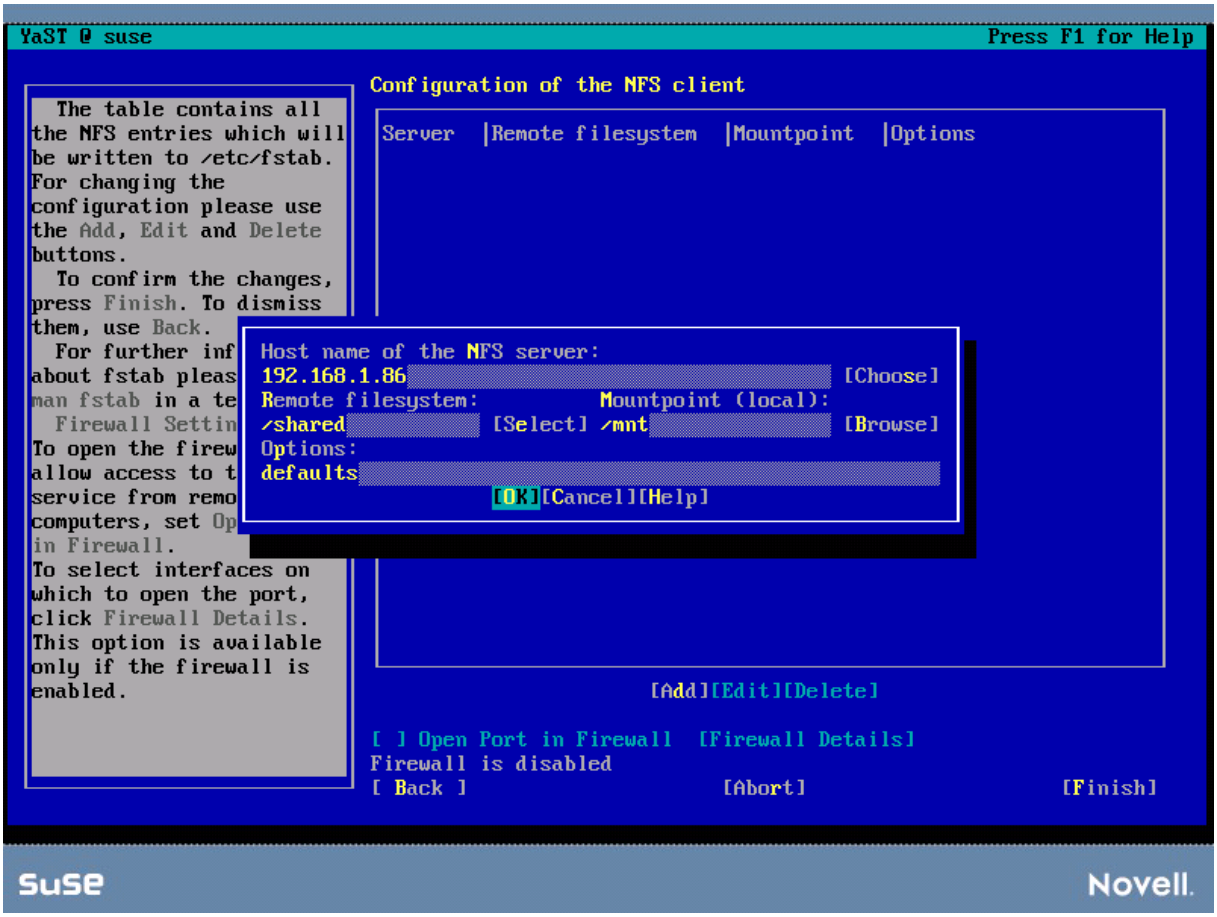
- *root_squash: Mit dieser Option werden die sonst umfassenden Rechte eines root-User eines NFS-Clients auf die Berechtigungen des Users „nobody“ herabgestuft. Diese Option sollte stets gesetzt sein!*
- *sync: Dies bedeutet, dass die Bestätigung von IO-Operationen (z.B. dem Schreiben einer Datei) erst an den Client gesendet wird, wenn die Operation erfolgreich abgeschlossen wurde.*

Weitere wichtige Optionen sind:

- *rw*: Das Gegenteil der Option ro. rw exportiert das Verzeichnis mit der grundsätzlichen Möglichkeit des read/write (d.h. lesenden und schreibenden) Zugriffs. Ob ein Benutzer tatsächlich lesenden bzw. schreibenden Zugriff erhält ist jedoch noch von den Berechtigungen auf Dateiebene abhängig. Ein derartiger Export von Verzeichnissen sollte wirklich nur in äußerst Vertrauenswürdigen Umgebungen erfolgen.
- *all_squash*: Alle Benutzer eine NFS-Client werden auf die Berechtigungen des Users „nobody“ herabgestuft.

Konfiguration eines NFS-Clients unter SUSE Linux 10

Unter der Kategorie „Network Services“ findet sich in YAST das Modul „NFS Client“. Durch Auswahl der Option „Add“ ist es möglich die, von einem NFS-Server exportierten Verzeichnisse in das eigene Dateisystem einzubinden; zu „mounten“:



Hierbei sind anzugeben:

- *Host name of the NFS server*: die IP-Adresse, der Host-Name oder der Domainname des NFS-Servers. Z.B 192.168.1.86
- *Remote filesystem*: Der Name des Verzeichnisses, wie es am NFS-Server exportiert wurde. Z.B /shared
- *Mountpoint (local)*: Unter welchem (bereits existierenden Verzeichnis) das exportierte Verzeichnis „eingehängt“ (bzw. „gemountet“) werden soll. Z.B. /mnt
- *Options*: Ermöglicht die Angabe von mount-Optionen. „defaults“ sollte nicht verwendet werden. Wichtige Optionen sind:
    - *rw*: Bewirkt, dass versucht wird das exportierte Verzeichnis im read/write-Modus zu mounten. Wurde das Verzeichnis vom NFS-Server jedoch nur mit ro (read-only) exportiert, besteht ungeachtet dieser Option nur die Möglichkeit eines lesenden Zugriffs.
    - *ro*: mountet das Verzeichnis read-only (ungeachtet ob der NFS-Server es mit rw oder ro exportiert hat)
    - *nosuid*: Bewirkt, dass SUID- und SGID-Bits für ausführbare Dateien unwirksam werden
    - *nodev*: Bewirkt, dass Device-Files nicht als solche behandelt werden.
    - *noexec*: Unabhängig von den Berechtigungen für einzelne Dateien, können im gemounteten Verzeichnis gespeicherte Programme nicht ausgeführt werden

  Um ein angemessenes Maß an Sicherheit zu erreichen, sollte jedenfalls „ro,nosuid,nodev,noexec“ verwendet werden.

NTP-Daemon

Allgemeines

Das Network Time Protocol (NTP) ermöglicht die Synchronisation der Systemuhren mehrer Computersysteme. Dies ist aus mehreren Gründen erforderlich:

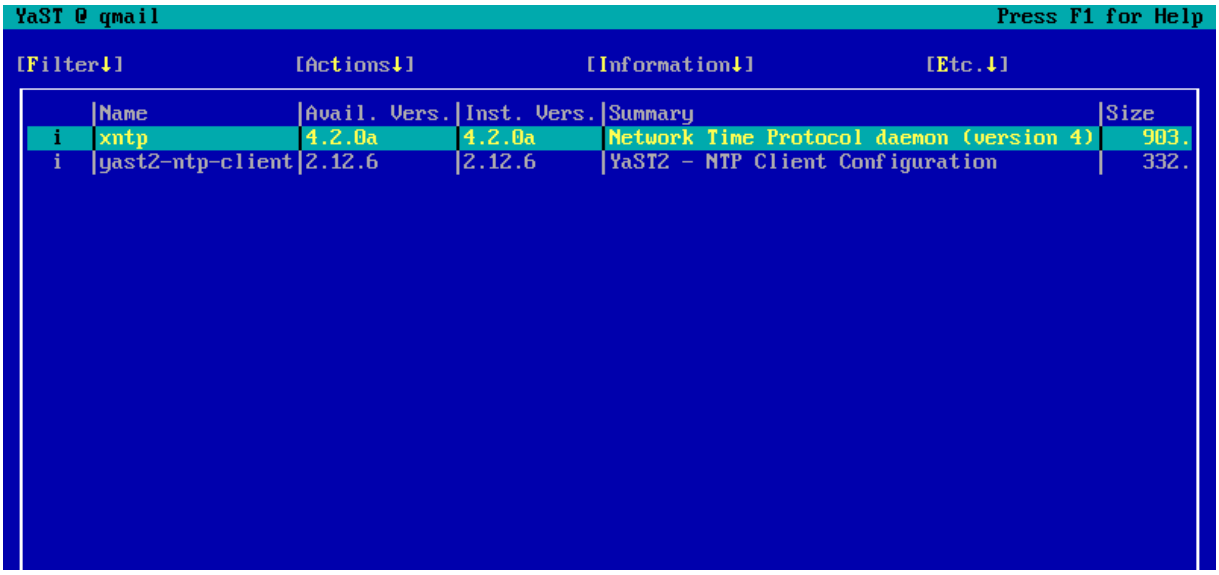
- Um in Log-Files aufgezeichnete Ereignisse in Relation zu einander setzen zu können, ist ihre zeitliche Abfolge meist von größter Wichtigkeit.
- Benutzer könnten durch, in der Zukunft liegende „last modified"-Werte von Dateien verwirrt werden.
- Selbiges gilt für E-Mails und die Information wann diese Empfangen wurden.
- Programme wie SSH, die Verschlüsselungsverfahren implementieren sind von einer korrekten Systemzeit abhängig.

International konnte man sich glücklicher Weise auf einen Zeit-Standard, genannt „International Atomic Time" (TAI) einigen. Dieser wird durch Atomuhren gemessen. Durch eine Zeitangabe in UTC (Universal Coordinated Time) – früher GMT (Greenwich Mean Time) bezeichnet – wird die Differenz zu TAI angegeben (z.B: 12:23 +0100).

Um die Systemzeit auf einem Server korrekt zu halten, ist eine Synchronisation mit einem NTP-Server erforderlich. Je nach Entfernung eines NTP-Servers von einer Atomuhr sind unterschiedliche Bezeichnungen üblich. Ein Stratum 1 NTP-Server erhält seine Zeit unmittelbar von einer Atomuhr (meist per Funksignal). Ein Stratum 2 NTP-Server erhält seine Zeit wiederum von einem Stratum 2 NTP-Server usw.

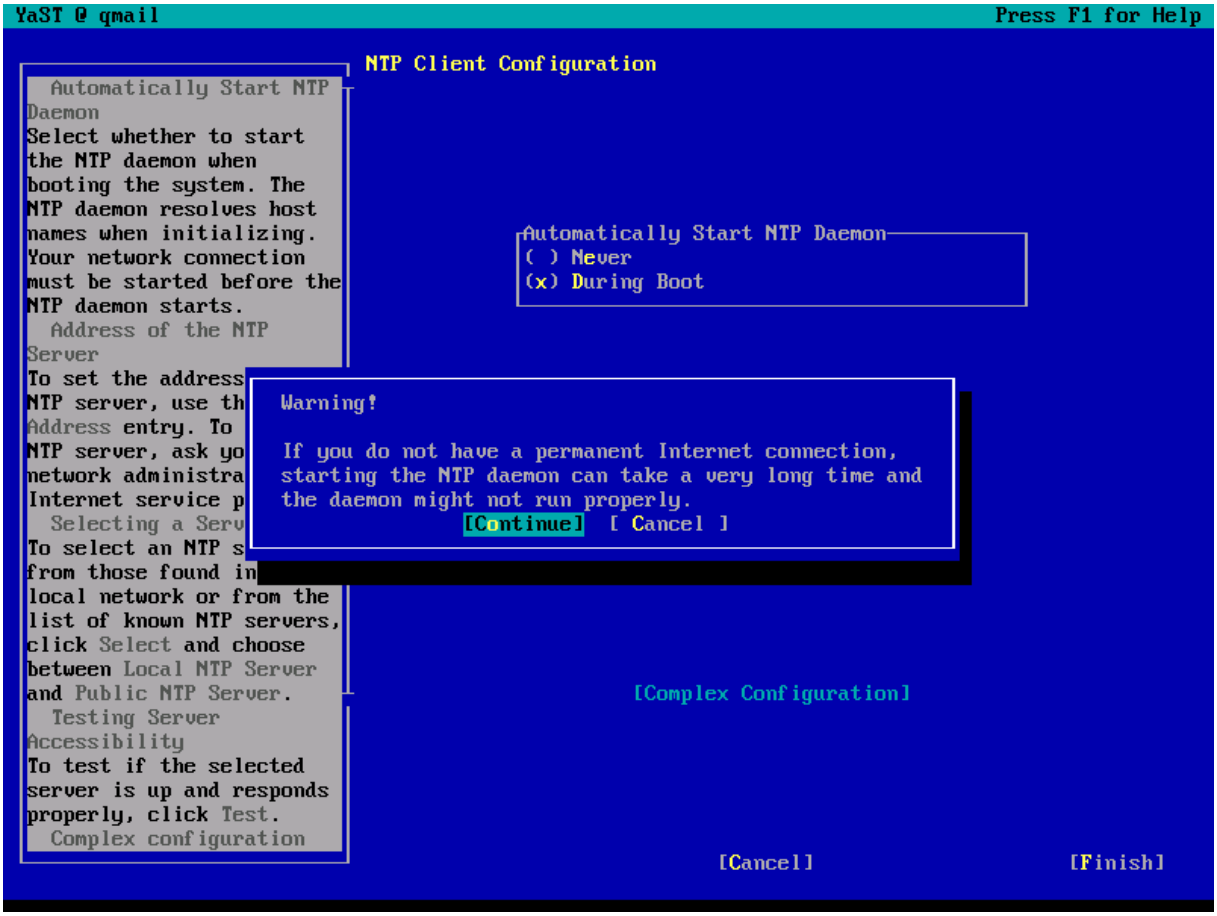
Installation unter SUSE Linux 10

Es ist das Paket xntp zu installieren. In YAST ist daher wieder in der Kategorie „Software" das Modul „Sofware Mangement" auszuwählen. Es ist wieder eine Suche über „Filter" → „Search" vorzunehmen, diesmal nach „ntp".

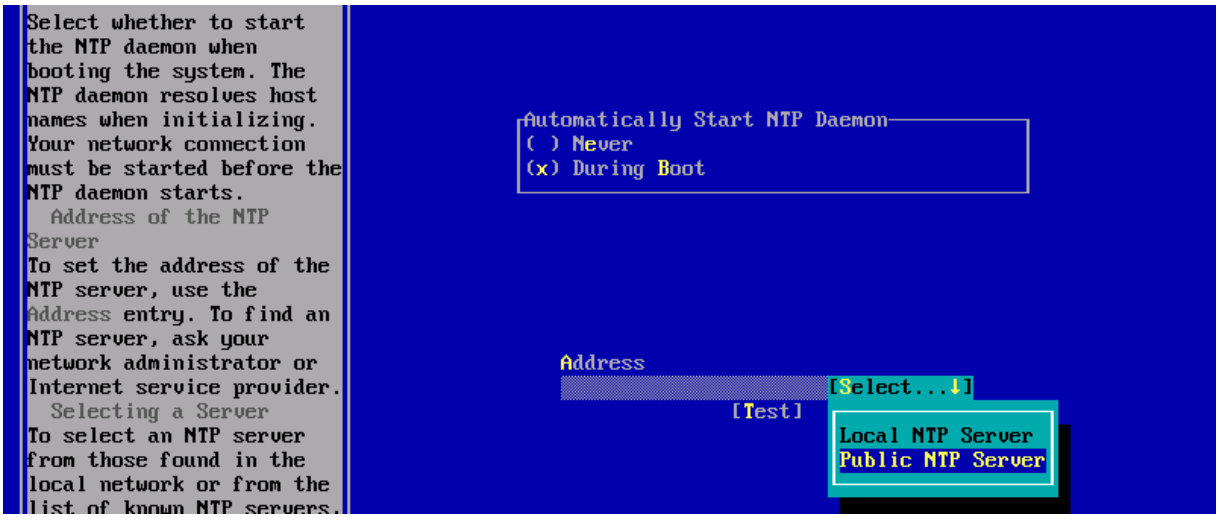


Konfiguration

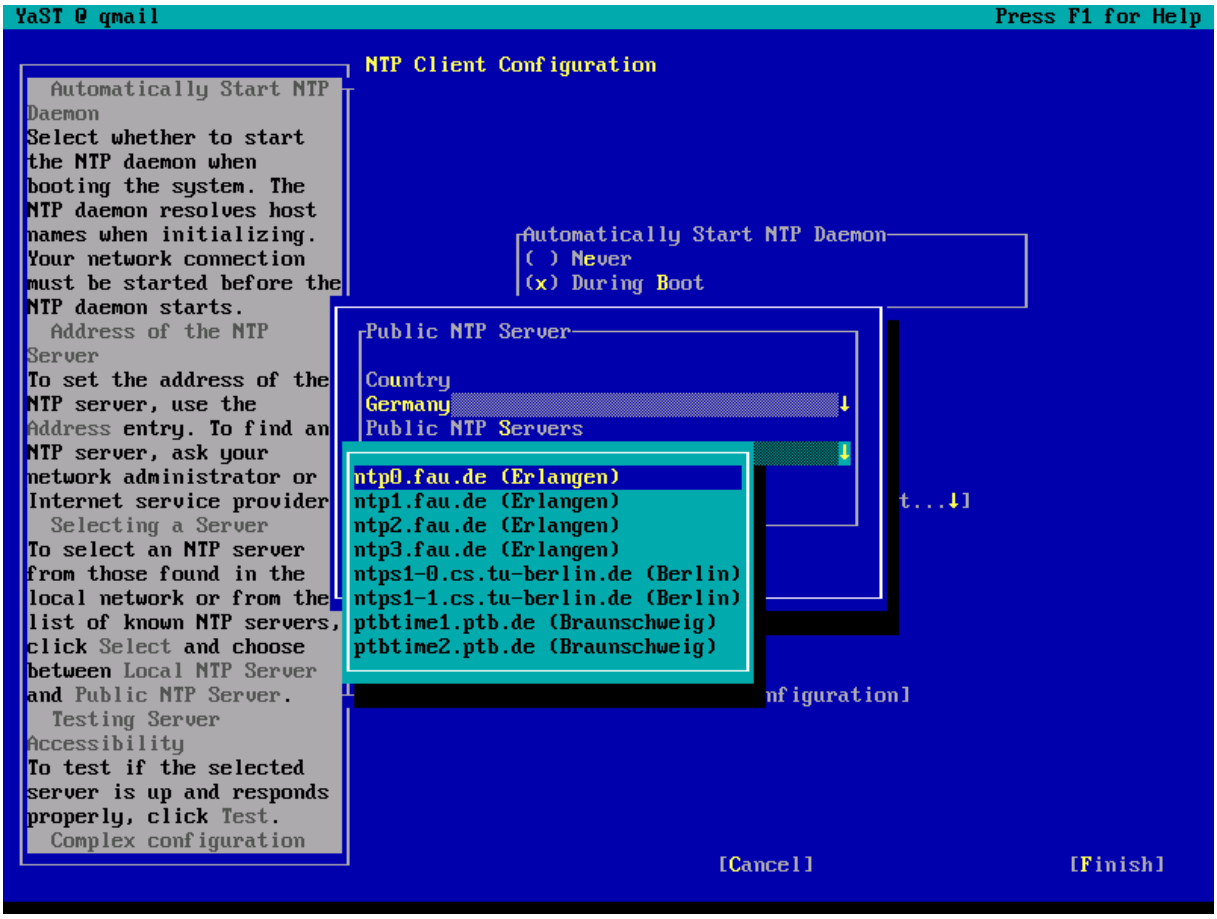
Nach erfolgter Installation von xntp kann YAST dazu verwendet werden den NTP-Client zu konfigurieren. Hierzu ist in der Kategorie „Network Services“ das Modul „NTP Client“ auszuwählen. Zunächst sollte der NTP-Daemon automatisch starten („During Boot“). Da ein Server über eine ständige Internetverbindung verfügt kann die angezeigt Warnmeldung ignoriert werden:



Nun ist der NTP-Server zu bestimmen, mit dem die Zeitsynchronisation erfolgen soll. Da anzunehmender Weise kein lokaler NTP-Server zur Verfügung steht ist ein öffentlicher NTP-Server auszuwählen („Public NTP-Server“):



Nun ist ein geographisch möglichst nahe gelegener NTP-Server zu wählen:

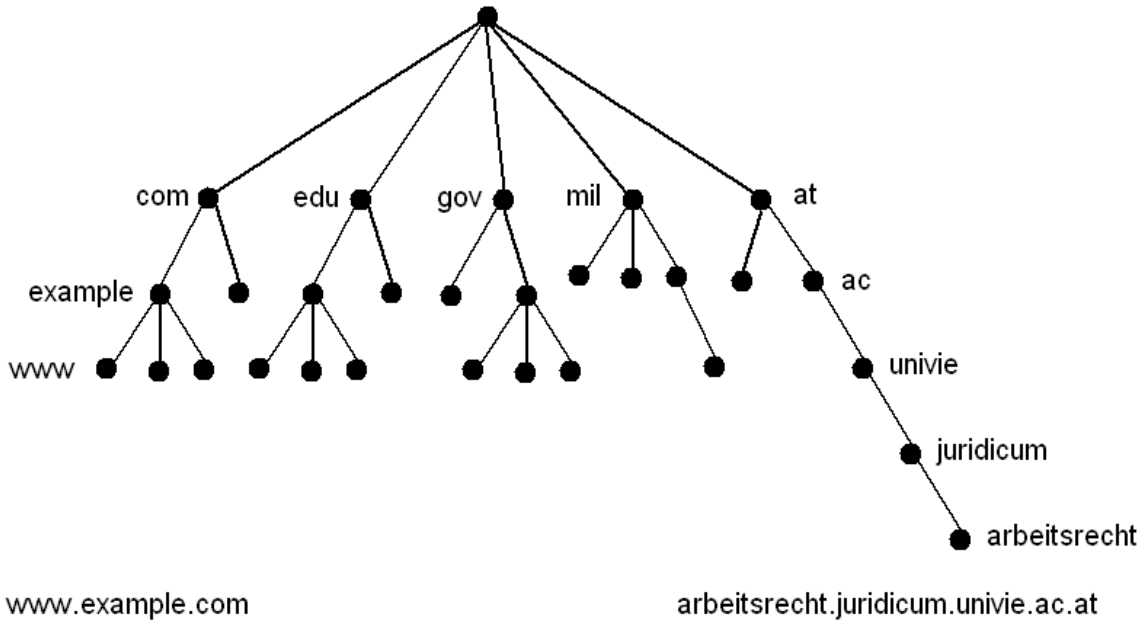


Durch Wahl der Option „Test“ sollte der gewählte NTP-Server sofort getestet werden. Hiermit ist die Konfiguration des NTP-Clients grundsätzlich bereits beendet. Es ist nur noch erforderlich mittels „Finish“ die Konfiguration zu speichern. Die komplexen Konfigurationsoptionen (unter „Complex Configuration) sollten im Zweifel nicht verändert werden.

DNS-Server

Allgemeines

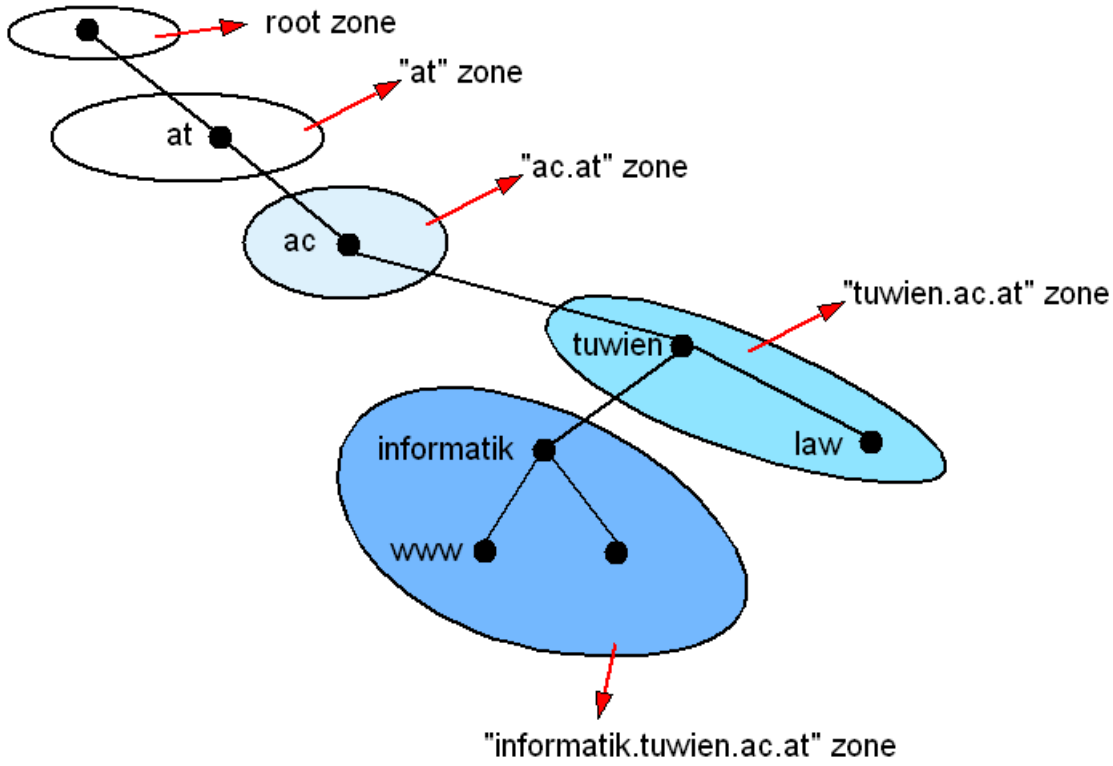
Das Domain Name System (DNS) ist eine weltweite, dezentrale Datenbank in der Informationen über IP-Adressen und Domainnamen gespeichert werden. Es dient vor allem dazu Domainnamen in IP-Adressen aufzulösen. Das Domain Name System kann zutreffend als „verkehrter Baum“ dargestellt werden:



Die in der ersten Ebene des Baumes enthaltene Elemente (hier exemplarisch: com, edu, gov, mil, at) bezeichnet man als Top Level Domains (TLDs) oder First Level Domains. Elemente der zweiten Ebene (z.B. example.com od. ac.at) werden als Second Level Domain bezeichnet.
Der Baumstruktur des DNS entsprechend sollte ein Domainnamen daher tatsächlich von rechts nach links gelesen werden! So arbeitet man sich beim Lesen des Domainnamen im DNS-Baum von oben nach unter herab („top-down Prinzip“ anstatt „bottom-up“).

Der dezentrale Aspekt von DNS manifestiert sich in der Möglichkeit die Autorität für sog. Zones delegieren zu können. Die Zone höchster ebene ist die sog. Root-Zone. In dieser ist die Information gespeichert welche DNS-Server die Autorität für die „Top-Level-Zones“ besitzen (z.B. welche DNS-Server die Autorität für die Zone „at“ besitzt). Jeder DNS-Server, der die Autorität für eine Zone hat ist in der Lage – ähnlich einem guten Manager – bestimmte Teile zu delegieren für andere jedoch selbst die Autorität zu behalten. In folgendem (fiktiven) Beispiel wurde einem bestimmten DNS-Server die Autorität für die Zone tuwien.ac.at zugewiesen. Dieser delegiert nur die Autorität für informatik.tuwien.ac.at an einen anderen DNS-Server. law.tuwien.ac.at bleibt demgegenüber jedoch in der Zone tuwien.ac.at. Wollen die Informatiker der TU nun einen neuen Domainnamen – z.B. ftp.informatik.tuwien.ac.at anlegen, so können sie dies autonom tun, da sie die Autorität für ihre Zone

besitzen. Die Juristen der TU können ohne Zustimmung von zentraler Stelle jedoch keine neue Sub-Domain anlegen, da sich nicht die Autorität für ihre Zone besitzen:



Um eine Redundanz zu erreichen gibt es für eine Zone meist einen primären und einen sekundären DNS-Server (politisch unkorrekt auch als Master und Slave bezeichnet). Der sekundäre DNS-Server ist grundsätzlich so konfiguriert, dass er die Daten des primären DNS-Servers spiegelt.

Praktischer Hinweis: Die manuelle Auflösung eines Domainnamens in eine IP-Adresse ist mit dem Command-Line-Tool nslookup möglich. Es ist sowohl unter Windows als auch unter Linux verfügbar. So gibt folgender Befehl die IP-Adresse von www.example.com aus:

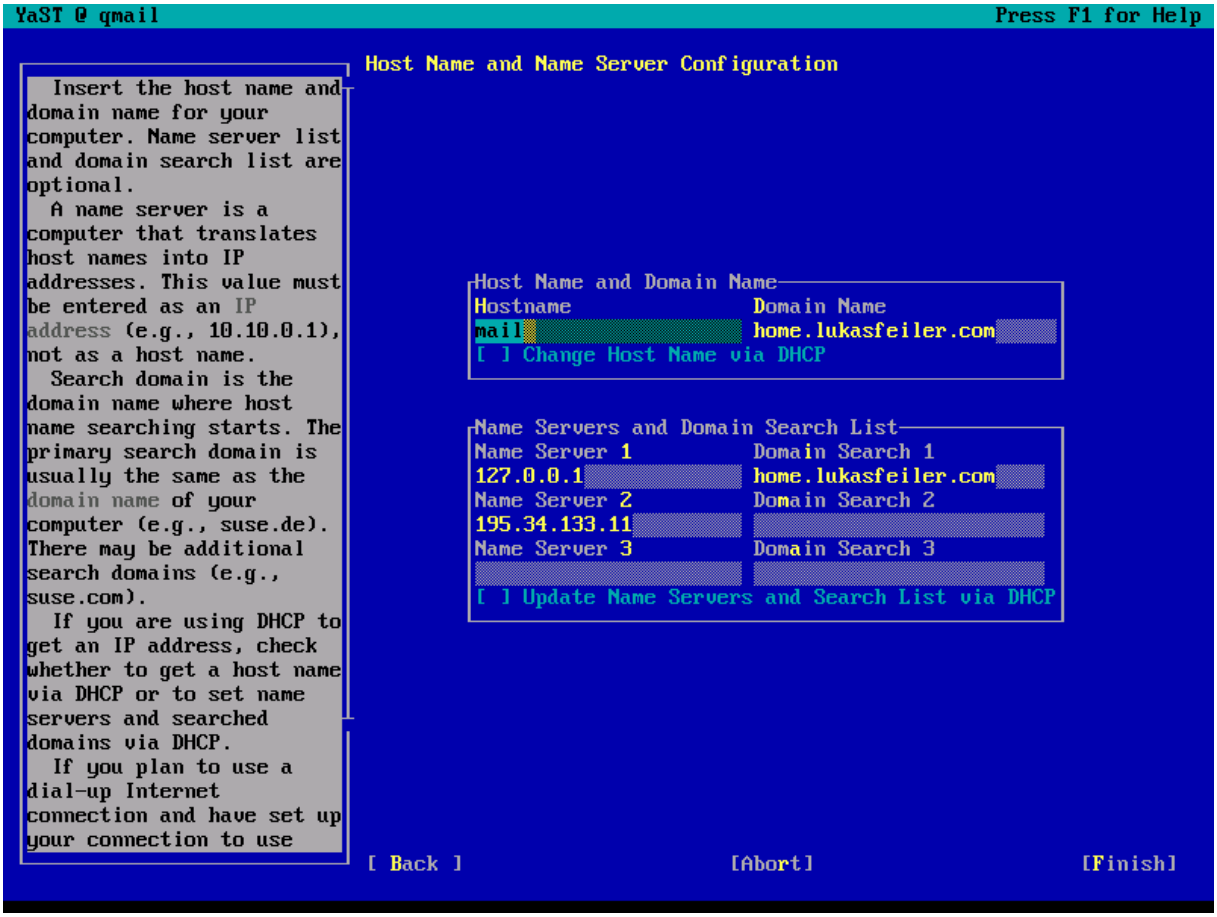
```
nslookup www.example.com
```

Installation unter SUSE Linux 10

Der weltweit am häufigsten verwendet DNS-Server ist BIND (Berkeley Internet Name Domain). Auch auf Linux ist er der DNS-Server der Wahl. Es sind die Pakete bind und bind-chrootenv zu installieren. In YAST ist daher wieder in der Kategorie „Software“ das Modul „Software Management“ auszuwählen. Es ist abermals eine Suche über „Filter“ → „Search“ vorzunehmen, diesmal nach „bind“.

Konfiguration von BIND

Bevor die Konfiguration des DNS-Servers selbst in die Hand genommen wird, sollte für das System ein korrekter Host-Name gesetzt werden. Hierzu ist in YAST aus der Kategorie „Network Services“ das Modul „DNS and Hostname“ auszuwählen woraufhin folgende Eingabemaske angezeigt wird:

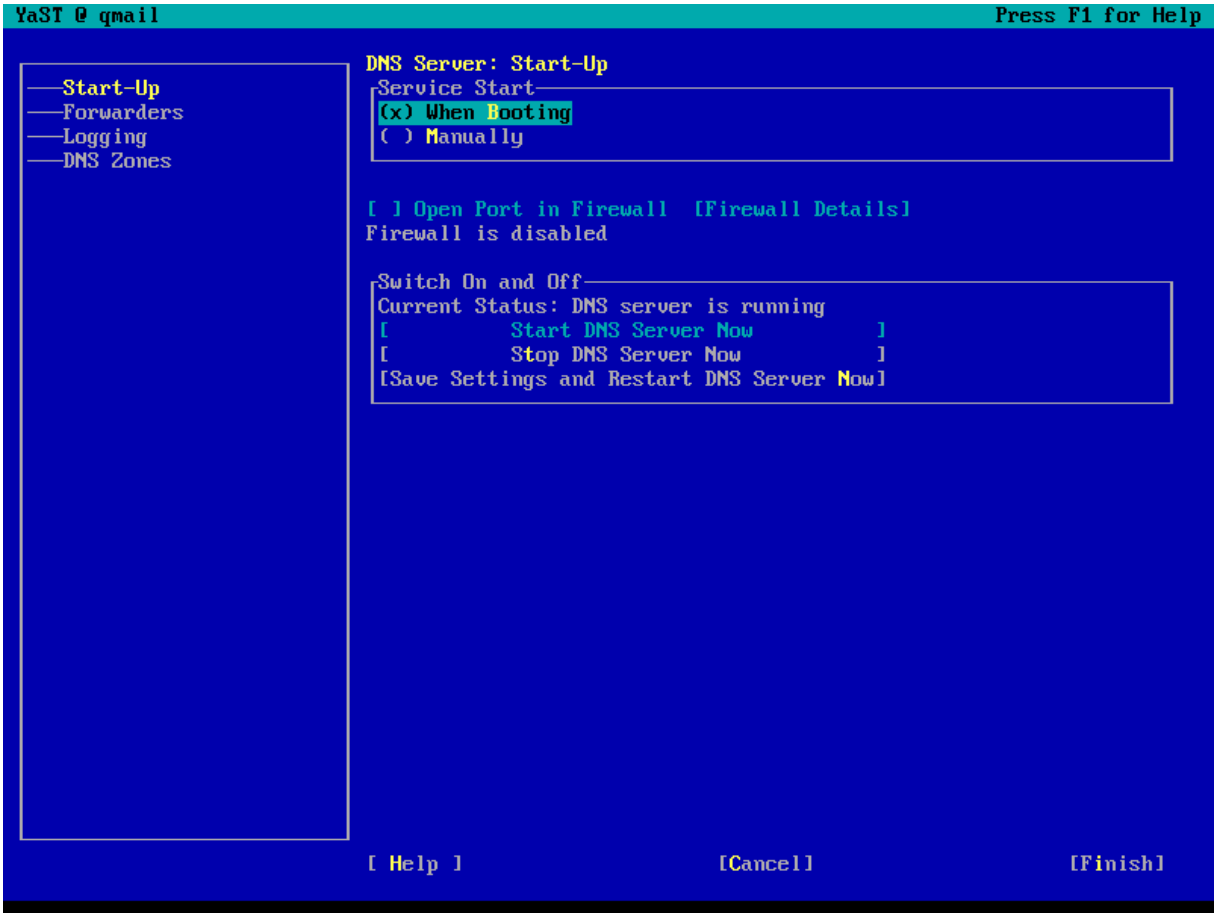


Es sind nun einzutragen:

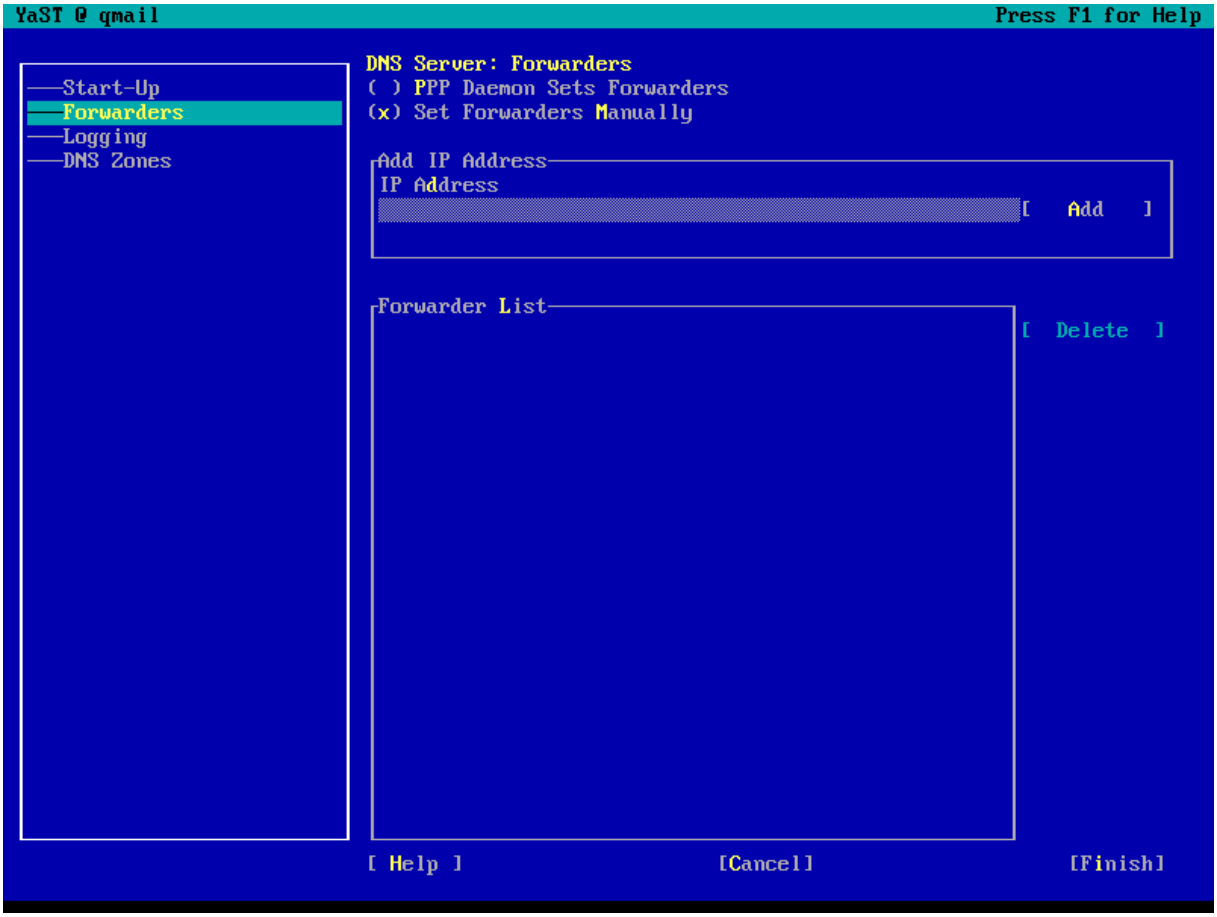
- *Hostname*: Der Name dieses Computers. Soll dieser Computer beispielsweise später als Mail-Server verwendet werden, so liegt es nahe den Host-Name „mail“ zu vergeben.
- *Domain Name*: Der Name der DNS-Domain in der sich der Computer befindet. In Kombination mit dem Hostname ergibt sich daraus der sog. Full-Qualified-Domain-Name (FQDN) des Computers. In obiger Abbildung lautet er mail.home.lukasfeiler.com.
- *Name Server 1-3*: Hier sind die DNS-Server einzutragen, die zwecks Auflösung eines Domain-Names in eine IP-Adresse befragt werden sollen. Name Server 2 wird nur dann befragt, wenn Name Server 1 nicht erreichbar ist. Da im Folgenden auf dem lokalen Computersystem Einen DNS-Server konfiguriert werden soll, ist jedenfalls der Primäre DNS-Server (Name Server 1) auf die IP-Adresse des lokalen Systems zu setzen. Diese ist stets 127.0.0.1 (das sog. „loopback interface“, eine Art virtuelle Netzwerkkarte).
- *Domain Search 1-3*: Wird währen des Betriebs des Systems versucht sich zu einem anderen System ohne Angabe eines FQDN zu verbinden (z.B. wird nur „mail“ angegeben) so wird versucht durch Anhängung der, unter Domain Search 1-3 gelisteten Domains das andere Computersystem zu finden. Meist ist es angebracht hier dasselbe wie unter *Domain Name* anzugeben.

Durch die Auswahl von „Finish“ werden die eingegeben Konfigurationsdaten gespeichert.

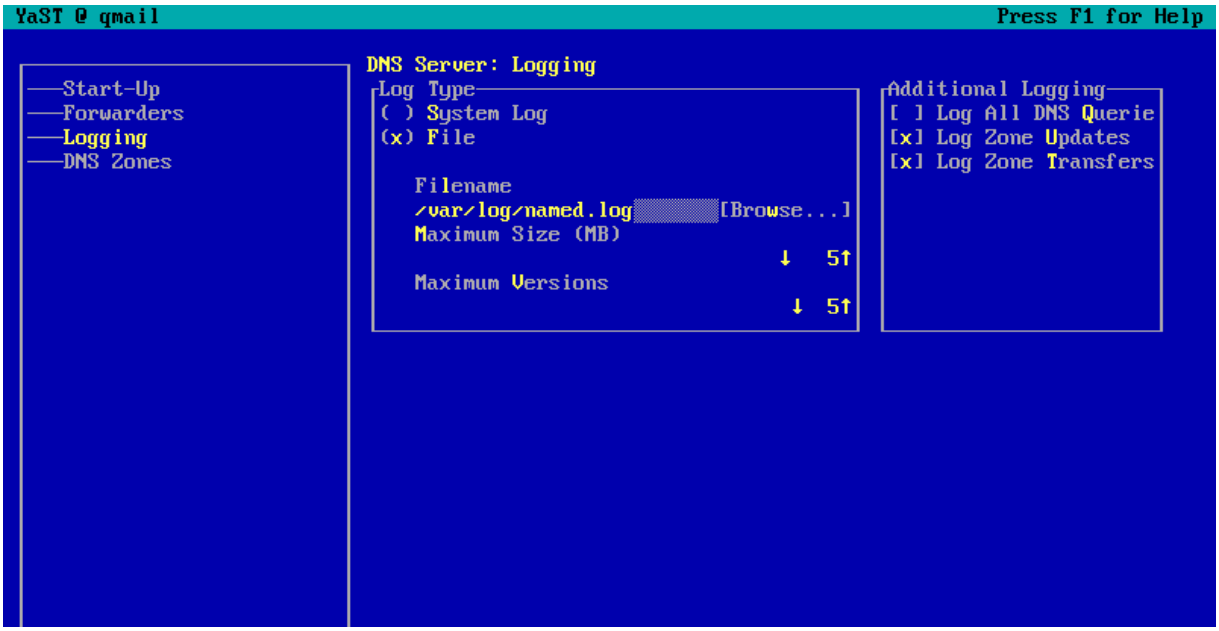
Im nächsten Schritt kann die Konfiguration des DNS-Servers selbst erfolgen. Zu diesem Zweck ist in YAST aus der Kategorie „Network Services“ das Modul „DNS Server“ auszuwählen. Das sich nun darstellende Interface ist in zwei Teile geteilt. Im linken Frame finden sich Kategorien von Konfigurationsoption. Am Anfang ist „Start-Up“ ausgewählt und daher im rechten Frame angezeigt. Es empfiehlt sich den DNS-Server automatisch starten zu lassen wenn das Betriebssystem startet (Start Service: „When Booting“):



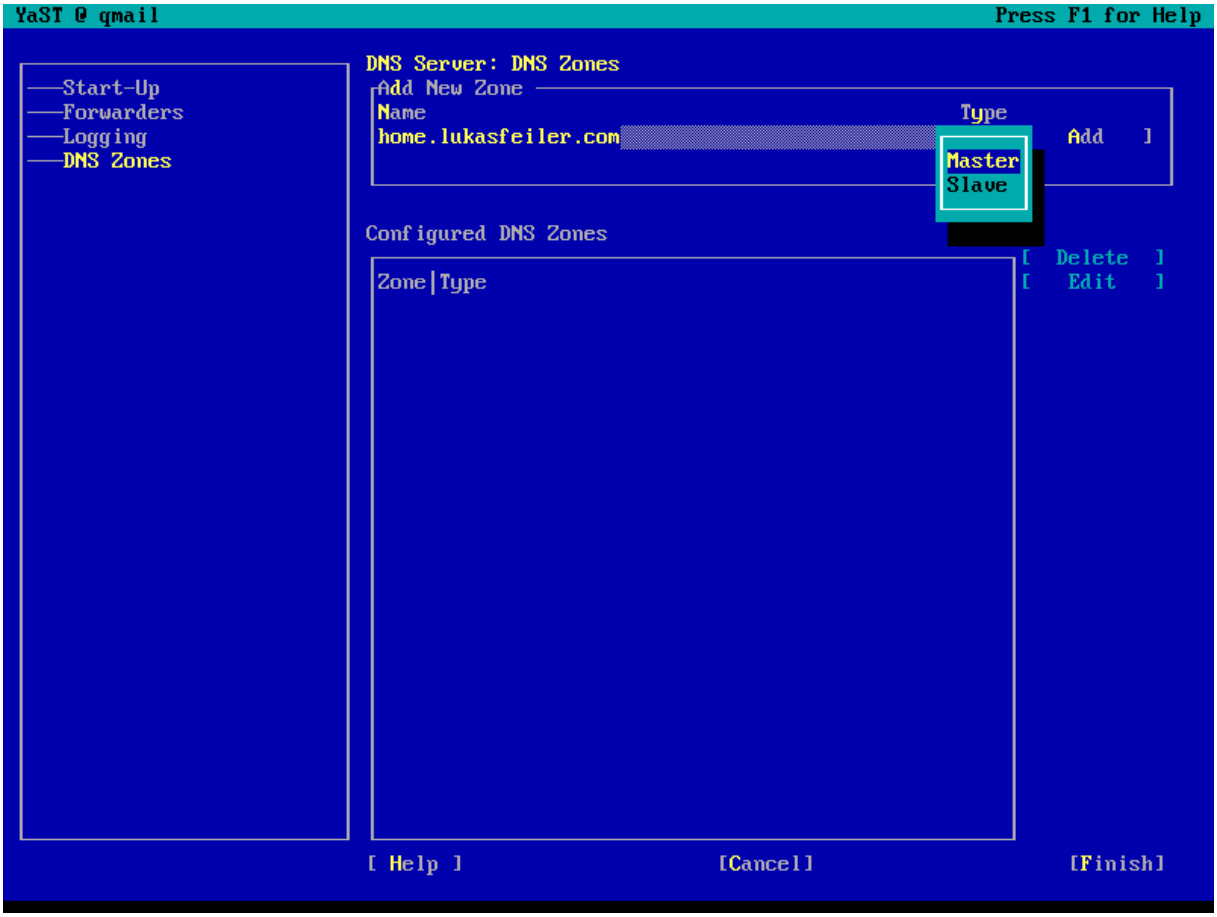
Die zweite Kategorie des linken Frames ist „Forwarders“. Werden für einen DNS-Server Forwarders eingetragen, so leitet er alle Anfragen nach Domains bzw. Zones für die er selbst nicht die Autorität besitzt an die eingetragen Forwarders weiter. Sind keine Forwarders eingetragen, so versucht der DNS-Server selbst durch die Befragung beliebiger anderer DNS-Server die Anfrage zu beantworten. Forwarders zu verwenden kann die Sicherheit in geringem Maße erhöhen, da nur noch mit einem bekannten System (dem Forwarder) kommuniziert wird. Die Performance leidet jedoch in aller Regel darunter. Folgende Abbildung zeigt die Möglichkeit Forwarders zu setzen:



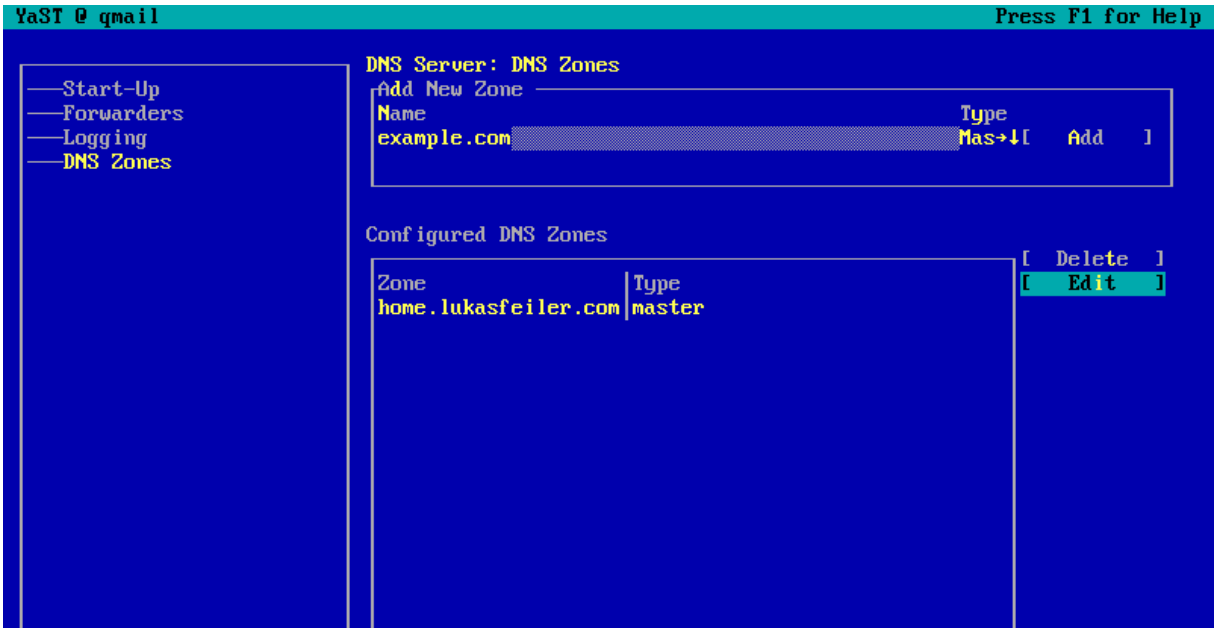
„Logging“, die dritte Kategorie im linken Frame ermöglicht eine Konfiguration des Logging von BIND. Am Anfang ist ein Logging mittels Syslog vollkommen ausreichend. Bei größeren DNS-Servern empfiehlt es sich jedoch den „Log Type“ auf File zu setzen. Als Filename ist /var/log/named.log üblich. Als maximale Log File Größe („Maximum Size“) ist meist 5 MB und als Anzahl der aufzuhebenden Log Files („Maximum Versions“) 5 angebracht. Zone Updates und Transfers sollt im Zweifel auch geloggt werden:



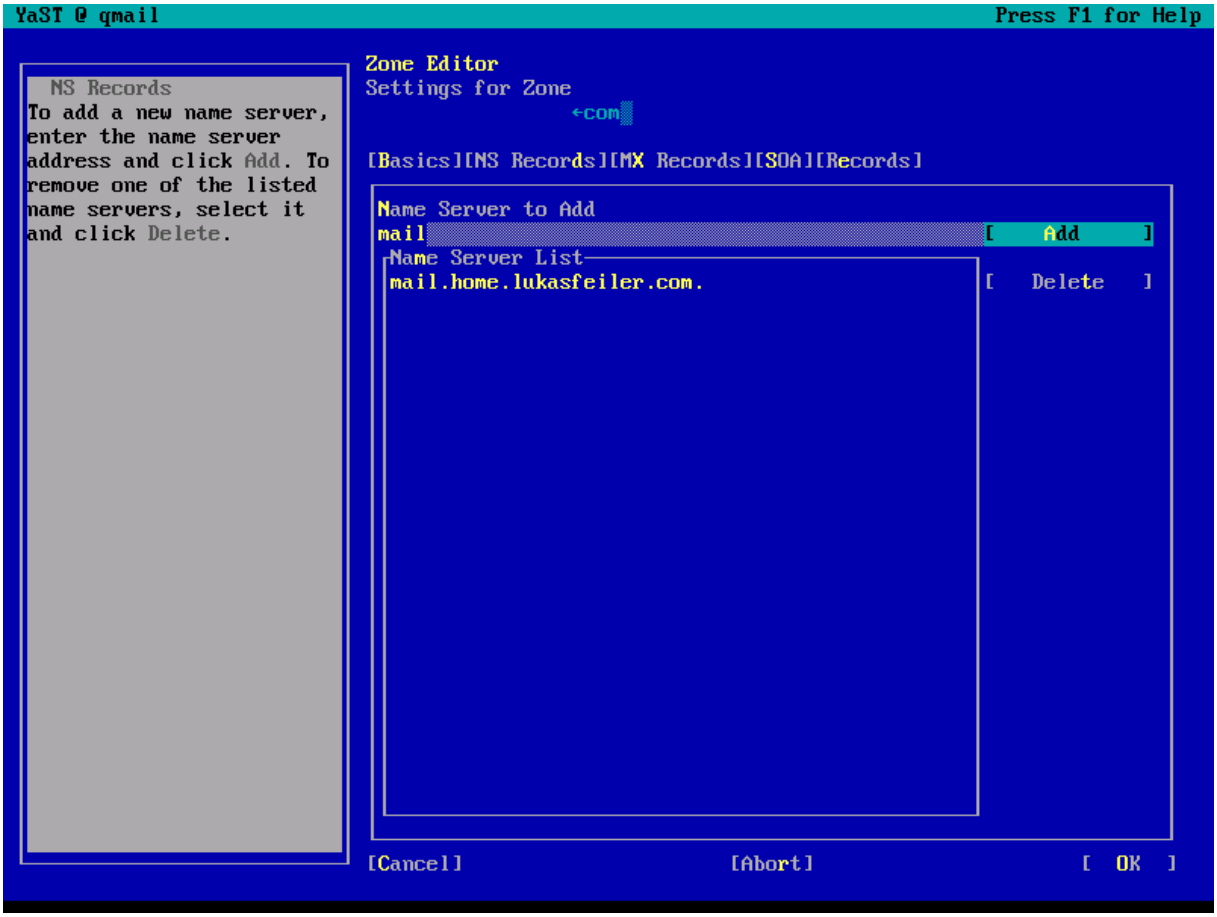
Unter dem vierten Menüpunkt des linken Frames, „DNS Zones“ können die Zones, für die der DNS-Server die Autorität besitzt konfiguriert werden. Unter „Name“ ist der Name der neuen Zone einzutragen (diesfalls die fiktive Zone home.lukasfeiler.com). Als „Type“ ist im Zweifel immer „Master“ auszuwählen und dann auf „Add“ zu gehen:



Danach scheint die Zone in der Liste „Configured DNS Zones auf“. Um neue Hosts in der Zone anzulegen, ist sie aus der Liste auszuwählen und auf „Edit“ zu gehen:

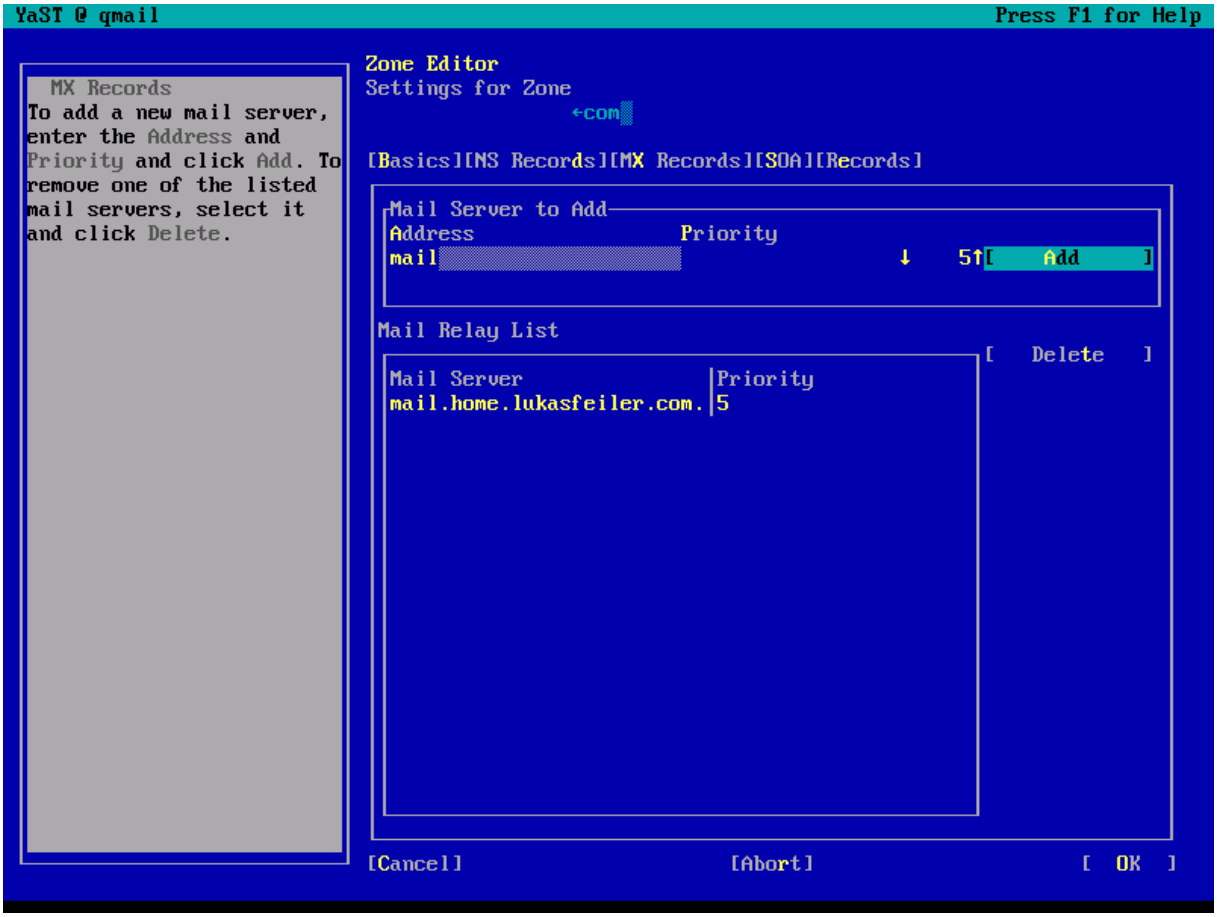


Vorab müssen NS- und sollten MX-Records erstellte werden. NS-Records geben an welcher DNS-Server (bzw. Name Server [NS]) für diese Zone die Autorität besitzt. Sie können unter dem zweiten der Menüpunkte, „NS Records“ angegeben werden. Da in unserem Beispiel der lokale Rechner die Autorität für die Zone home.lukasfeiler.com hat, ist der Name des lokalen Rechners einzugeben: „mail“. Durch die Auswahl von „Add“ wird der FQDN mail.home.lukasfeiler.com.[2] als DNS-Server hinzugefügt.

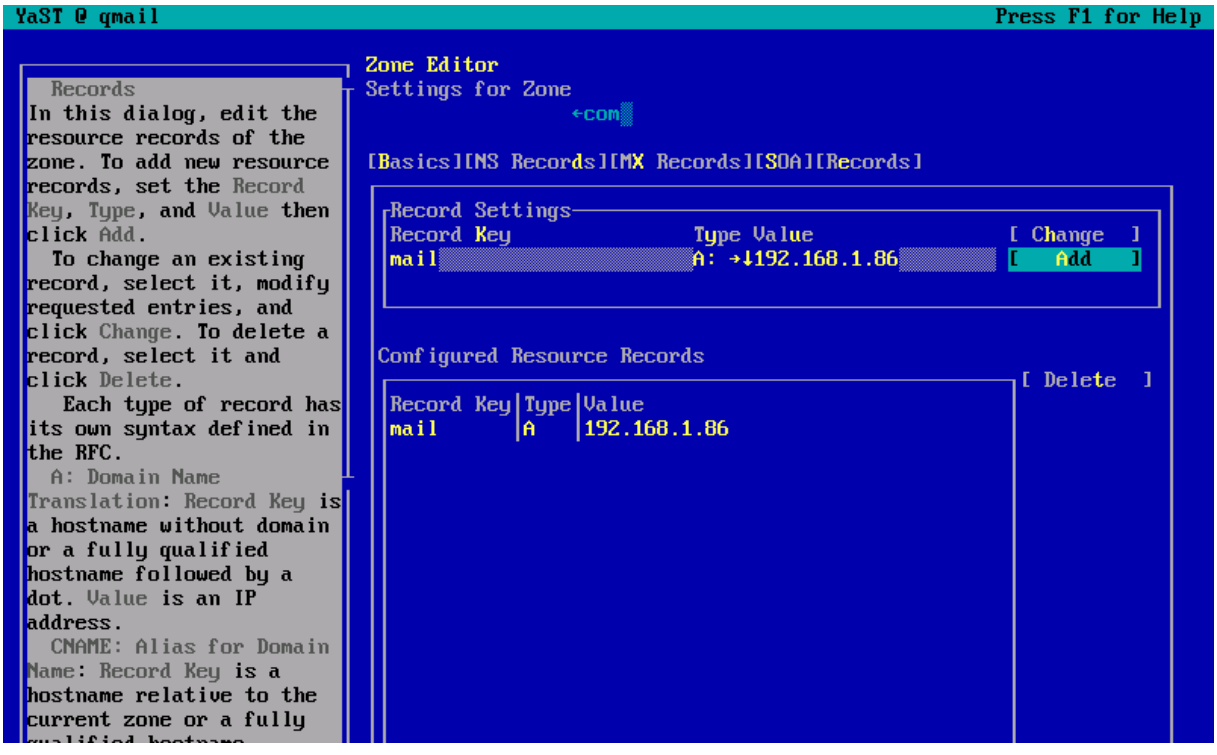


Unter dem dritten Menüpunkt, „MX Records“ können die MX-Records der Zone angegeben werden. Diese bestimmen an welches System E-Mails geschickt werden sollen die an <user>@<zone_name> geschickt werden – in unserem Beispiel also an welchen SMTP-Server E-Mails geleitet werden sollen die an <user>@home.lukasfeiler.com adressiert sind. Um mail.home.lukasfeiler.com auch alle E-Mails der Zone home.lukasfeiler.com empfangen zu lassen ist „mail“ als „Address“ (unter „Mail Server to Add“) einzutragen. Die „Priority“ ist nur relevant wenn mehrere Mail-Server für eine Zone existieren. Diesfalls werden alle Mails an den SMTP-Server geleitet, der einen niedrigeren Priority-Wert aufweist (funktional gesehen eigentlich ein Distance-Wert).

[2] In BIND enden FQDN stets mit einem Punkt, der die Root-Zone andeutet. Andernfalls würde es sich um einen zur aktuellen Zone relativen Namen handeln.



Nun ist der fünfte der Menüpunkte, „Records“ auszuwählen. Um den Host mail der Zone home.lukasfeiler.com hinzuzufügen, ist „mail“ als „Record Key“ einzutragen. Der Record „Type“ sollte „A“ lauten, was eine Übersetzung eines Domainnamens in eine IP-Adresse ermöglicht. Als „Value“ ist die IP-Adresse des Computers mail.home.lukasfeiler.com einzutragen, diesfalls 192.168.1.86. Durch Auswahl von „Add“ wird der Host der Zone hinzugefügt:



Durch die Auswahl von „OK“ wird die Konfiguration der Zone home.lukasfeiler.com beendet. Durch ein weiteres „Finish“ wird die erstellte DNS-Server Konfiguration gespeichert.

Nun kann durch die Ausführung des Befehls

```
nslookup mail.home.lukasfeiler.com
```

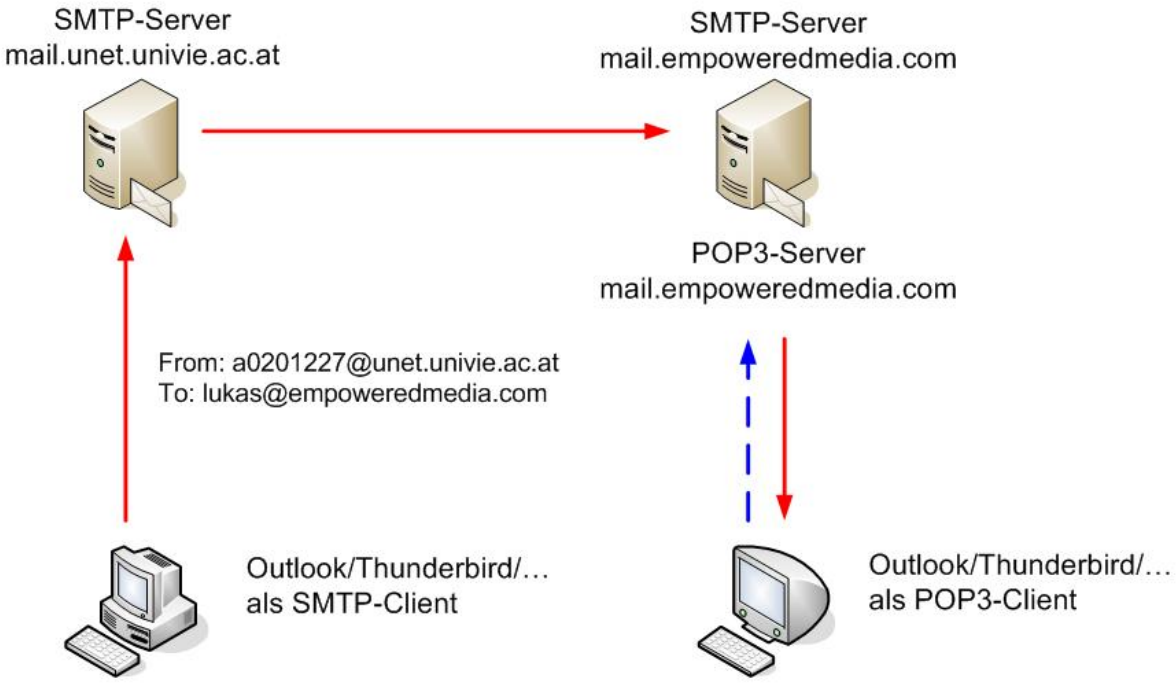
Die Funktionalität des DNS-Servers überprüft werden.

Mail-Server – Postfix

Allgemeines

Mail-Server zählen nach wie vor in nahezu jeder Organisation zu den Anwendungen höchster Wichtigkeit.

Zum Versenden eines E-Mails kommt hierbei das Protokoll SMTP (Simple Mail Transfer Protocol) zum Einsatz. Der Empfang von E-Mails erfolgt demgegenüber meist mittels POP3 (Post Office Protocol Version 3), bzw. in selteneren Fällen mittels IMAP (Internet Message Access Protocol). Nun folgend eine Darstellung typischen Weges eines E-Mails:

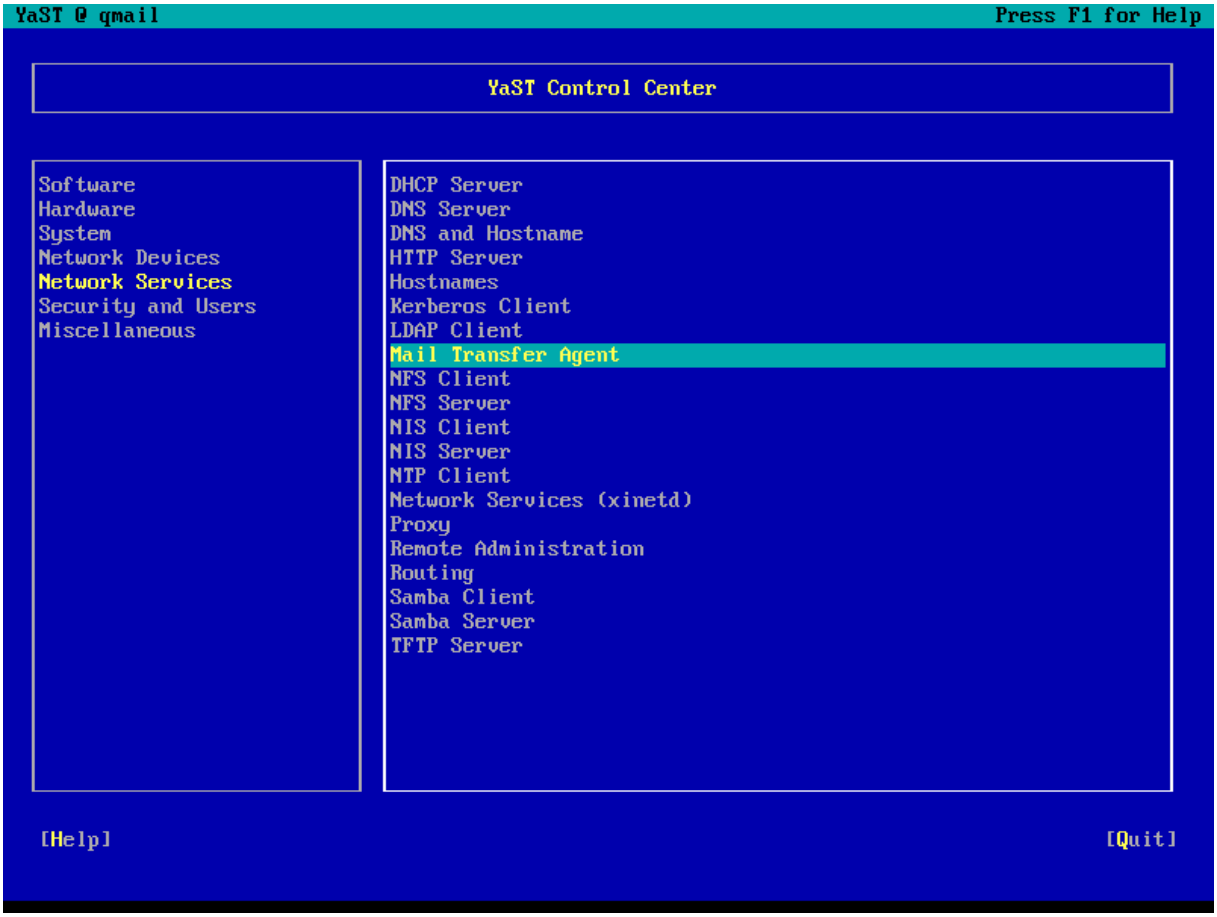


Installation

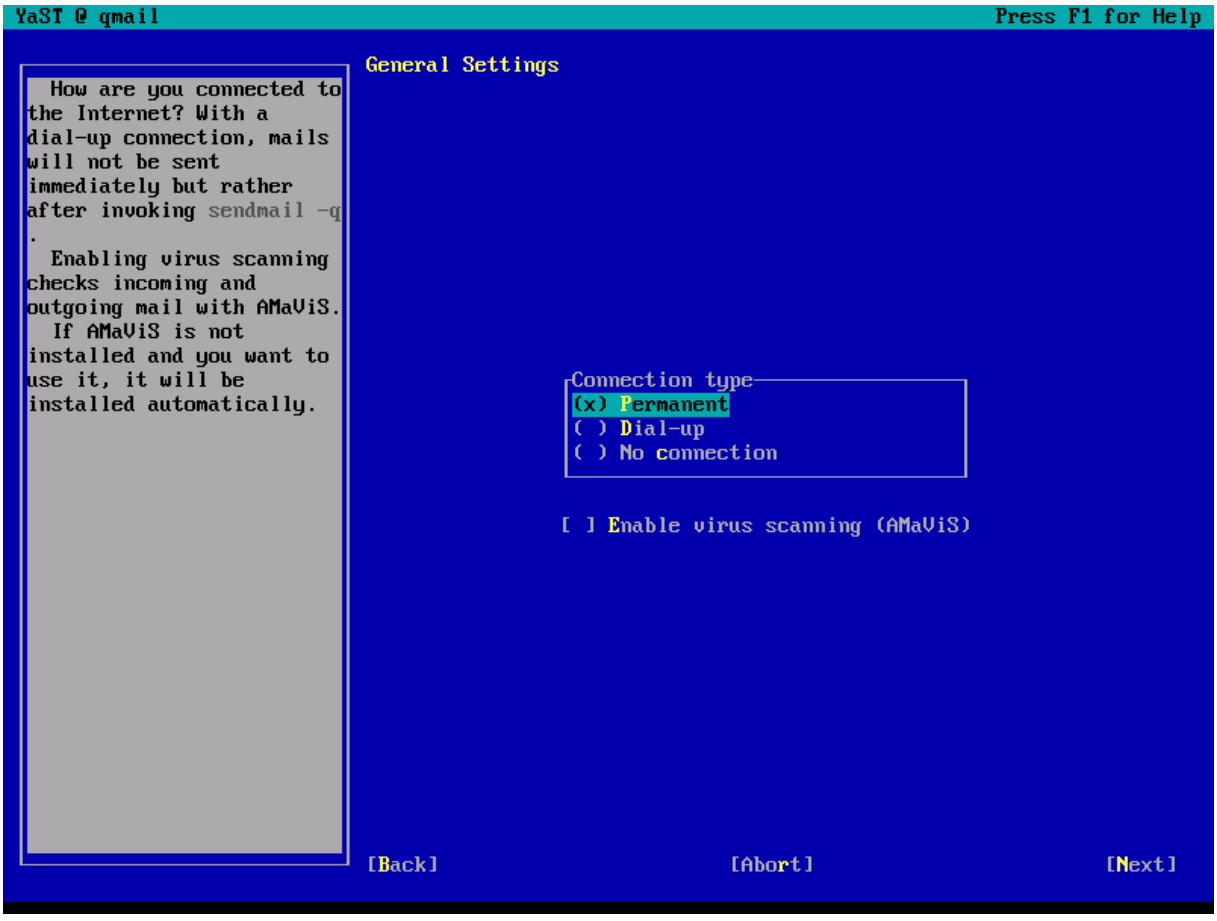
Auf einer Standard-Installation von SUSE 10 Postfix bereits vorhanden. Um einen voll funktionsfähigen Mailserver zu erhalten muss jedoch auch das Paket qpopper installiert werden. Hierzu ist in YAST aus der Kategorie „Sofware“ wieder das Modul „Software Management“ auszuwählen. Dann kann eine Suche nach qpopper über „Filter“→“Search“ erfolgen.

Konfiguration

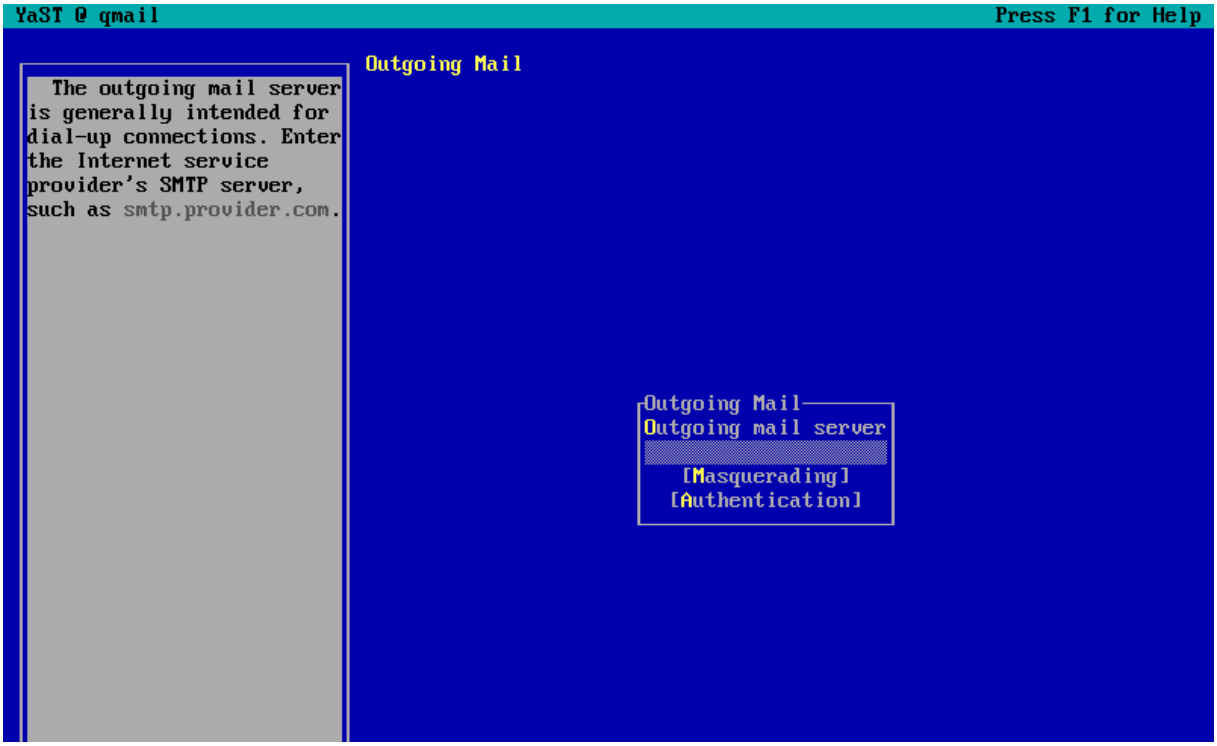
Nach erfolgreicher Installation von qpopper kann die Konfiguration von Postfix in Angriff genommen werden. Hierzu ist in YAST aus der Kategorie „Network Services“ das Modul „Mail Transfer Agent“ auszuwählen:



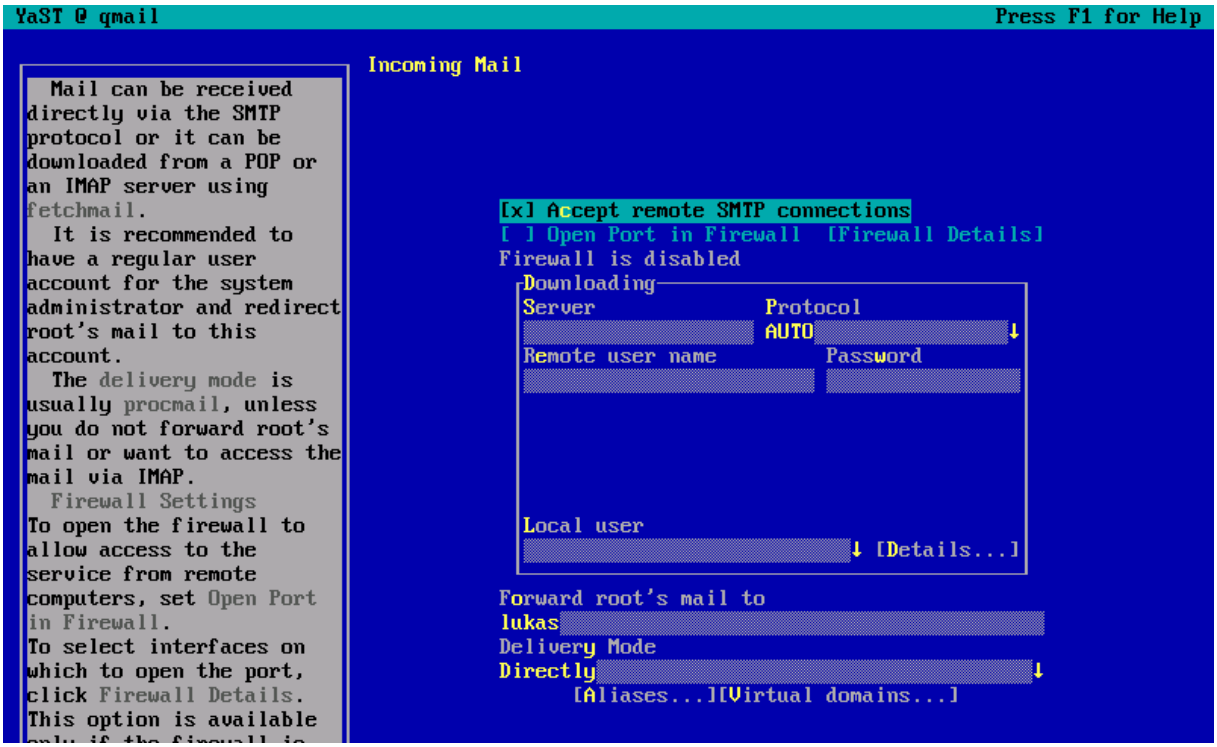
Im nun folgenden Konfigurationsdialog ist „Connection Type“ auf „Permanent“ zu setzen:



Die nun angezeigten Einstellungen für ausgehende E-Mails können beibehalten werden:



Nach der Auswahl von „Next“ präsentiert sich ein Konfigurationsdialog für eingehende Nachrichten („Incoming Mail“). Zunächst ist hier „Accept remote SMTP connections“ zu aktivieren. Die „Downloading“-Funktionalität soll hier deaktiviert bleiben. „Forward root's mail to“ sollte auf den regulären Benutzernamen des Administrators gesetzt werden. Der „Delivery Mode“ sollte hierbei auf „Directly“ belassen werden:



Der FQDN des Mailservers ist im hier dargestellten Konfigurationsbeispiel mail.home.lukasfeiler.com. Soll der Mail Server jedoch nicht nur an *@mail.home.lukasfeiler.com sondern auch an *@home.lukasfeiler.com adressierte E-Mails empfangen können, ist ausnahmsweise an eine Konfigurationsdatei selbst hand anzulegen. In /etc/postfix/main.cf wäre zu diesem Zweck die Zeile

```
mydestination = $myhostname, localhost.$mydomain
```

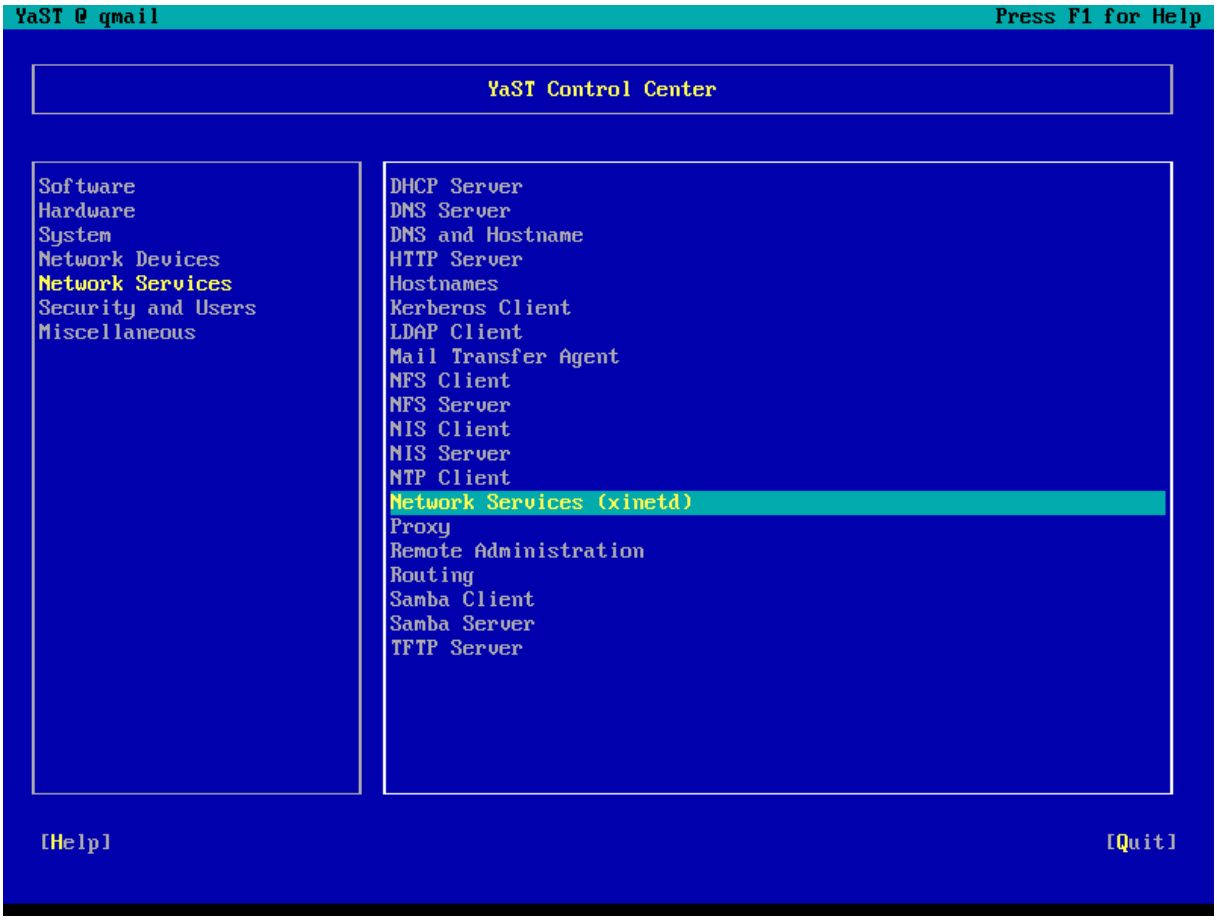
in

```
mydestination = $myhostname, localhost.$mydomain, $mydomain
```

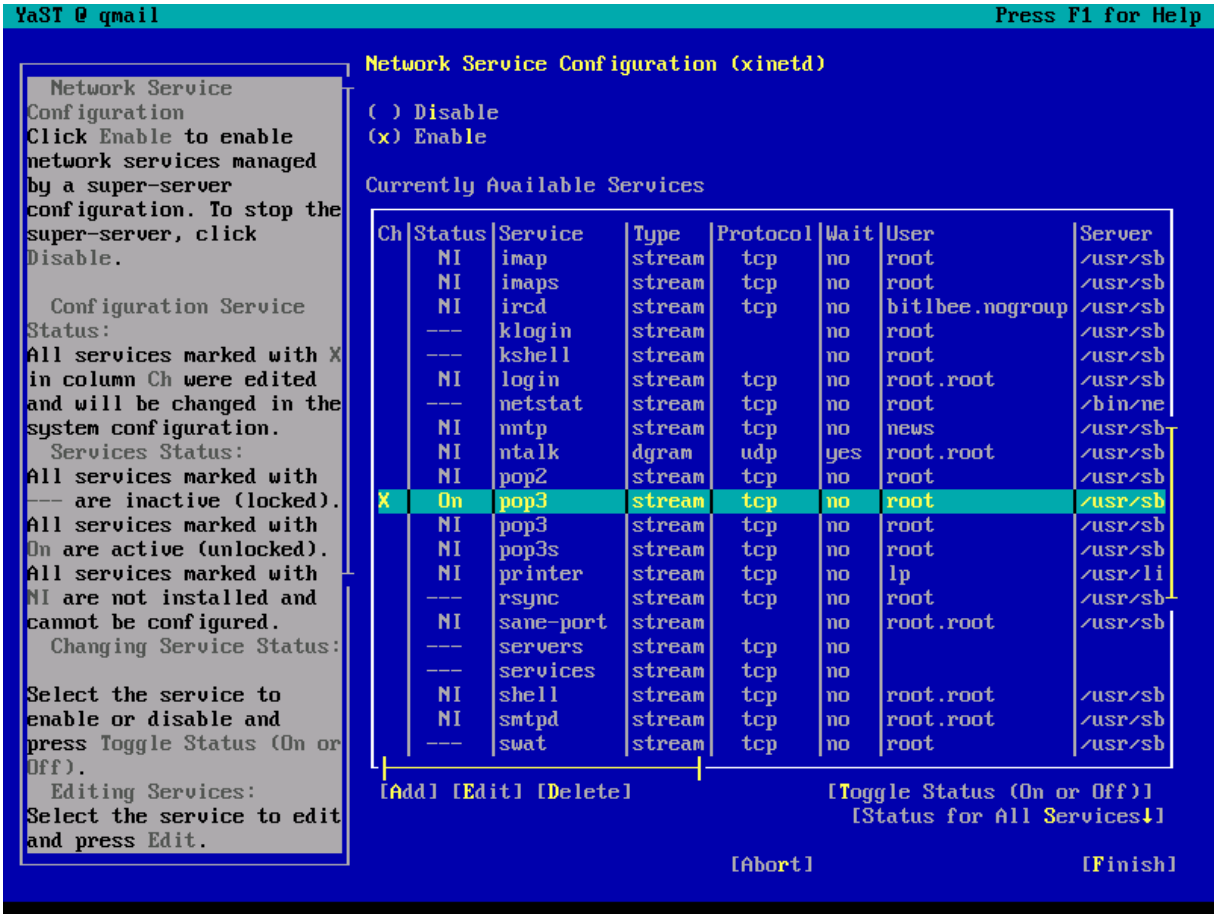
zu verändern.

Der POP3-Server qpopper

Um den POP3-Server qpopper freizuschalten muss erst eine Konfiguration des Dienstes xinitd vorgenommen werden. Zu diesem Zweck ist in YAST das Modul „Network Services (xinitd)“ aus der Kategorie „Network Services“ zu starten:



Am nun erscheinenden Screen ist die Option „Enable“ zu aktivieren. Weiters ist der Dienst pop3 (Server: /usr/sbin/popper) auszuwählen und durch „Toggle Status (On or Off)“ zu aktivieren:



Ch	Status	Service	Type	Protocol	Wait	User	Server
	NI	imap	stream	tcp	no	root	/usr/sb
	NI	imaps	stream	tcp	no	root	/usr/sb
	NI	ircd	stream	tcp	no	bitlbee.nogroup	/usr/sb
	---	klogin	stream		no	root	/usr/sb
	---	kshell	stream		no	root	/usr/sb
	NI	login	stream	tcp	no	root.root	/usr/sb
	---	netstat	stream	tcp	no	root	/bin/ne
	NI	nntp	stream	tcp	no	news	/usr/sb
	NI	ntalk	dgram	udp	yes	root.root	/usr/sb
	NI	pop2	stream	tcp	no	root	/usr/sb
X	On	pop3	stream	tcp	no	root	/usr/sb
	NI	pop3	stream	tcp	no	root	/usr/sb
	NI	pop3s	stream	tcp	no	root	/usr/sb
	NI	printer	stream	tcp	no	lp	/usr/li
	---	rsync	stream	tcp	no	root	/usr/sb
	NI	sane-port	stream		no	root.root	/usr/sb
	---	servers	stream	tcp	no		
	---	services	stream	tcp	no		
	NI	shell	stream	tcp	no	root.root	/usr/sb
	NI	smtpd	stream	tcp	no	root.root	/usr/sb
	---	swat	stream	tcp	no	root	/usr/sb

Durch die Auswahl von „Finish“ ist die Konfiguration von Postfix und qpopper als POP3-Server damit abgeschlossen.

Verwaltung

Zu den Verwaltungsaufgaben eines Mail-Serveradministrators gehört insbesondere das Anlegen neuer Mail-Accounts. In der hier vorgestellten Konfiguration erfolgt dies durch das Anlegen eines neuen System-Accounts. Dies kann über das YAST-Modul „User Management“ aus der Kategorie „Security and Users“ bewerkstelligt werden.

Spam-Abwehr mit Linux

Allgemeines

Das Versenden von Spam ist in den letzten Jahren zu einem profitablen Geschäft geworden. Aus diesem Grund gehen Spammer immer geschickter vor, um bestehende Spam-Filter zu umgehen. So besteht ein ständiger Wettlauf zwischen Spam-Versendern und Herstellern von Anti-Spam-Lösungen. Eine Open-Source Lösung der Apache Group ist SpamAssassin. Es handelt sich hierbei um eine der besten am Markt erhältlichen Anti-Spam-Lösungen, da sie in der Lage ist mehrere Verfahren wie DNS Blacklists (DNSBL)[3], SPF[4] und Inhaltsanalyse zu kombinieren und daraus einen sog. Spam-Score zu errechnen, der die Wahrscheinlichkeit ausdrückt, mit der es sich bei dem betreffenden Mail tatsächlich um Spam handelt. Dies ermöglicht ein differenziertes Vorgehen um sowohl die false positives (bloß vermeintlichen Spams) als auch die false negatives (vermeintlich erwünschte E-Mails) minimal zu halten.

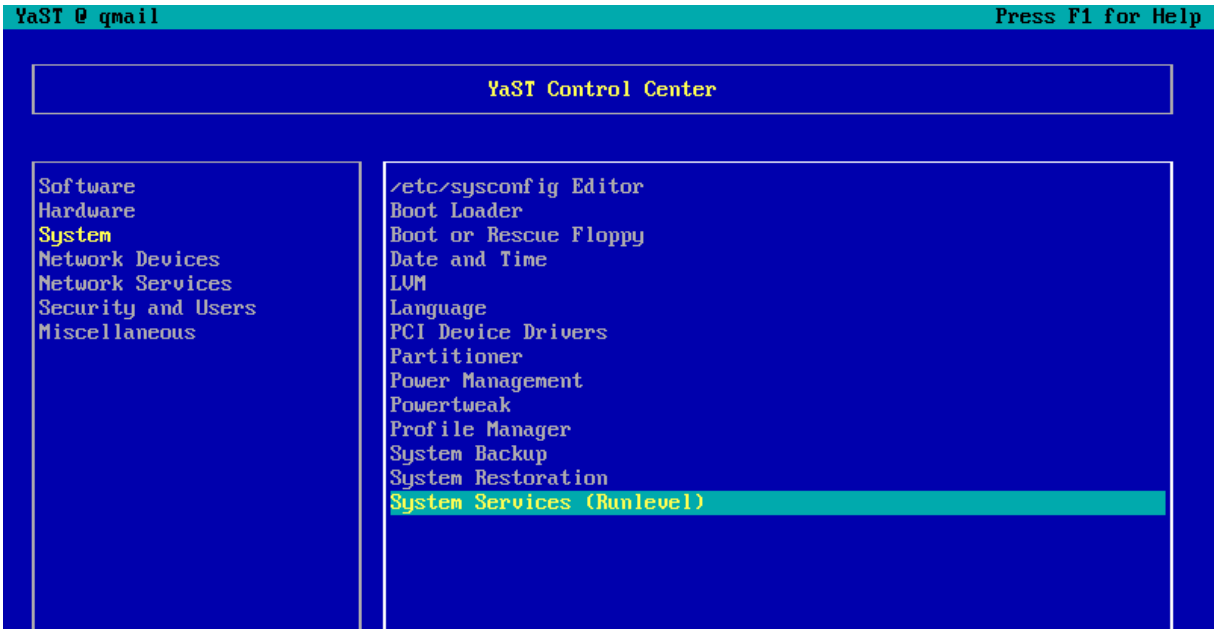
Die Integration von SpamAssassin in Postfix kann unter SUSE 10 mit Hilfe von Amavis bewerkstelligt werden. Amavis agiert hierbei gleichsam als Gateway zwischen Postfix und SpamAssassin.

Installation von Amavis und SpamAssassin

Amavis erfordert die Installation des Pakets „amavis-new“. Für SpamAssassin sind die Pakete „spamassassin“ und „perl-spamassassin“ zu installieren.

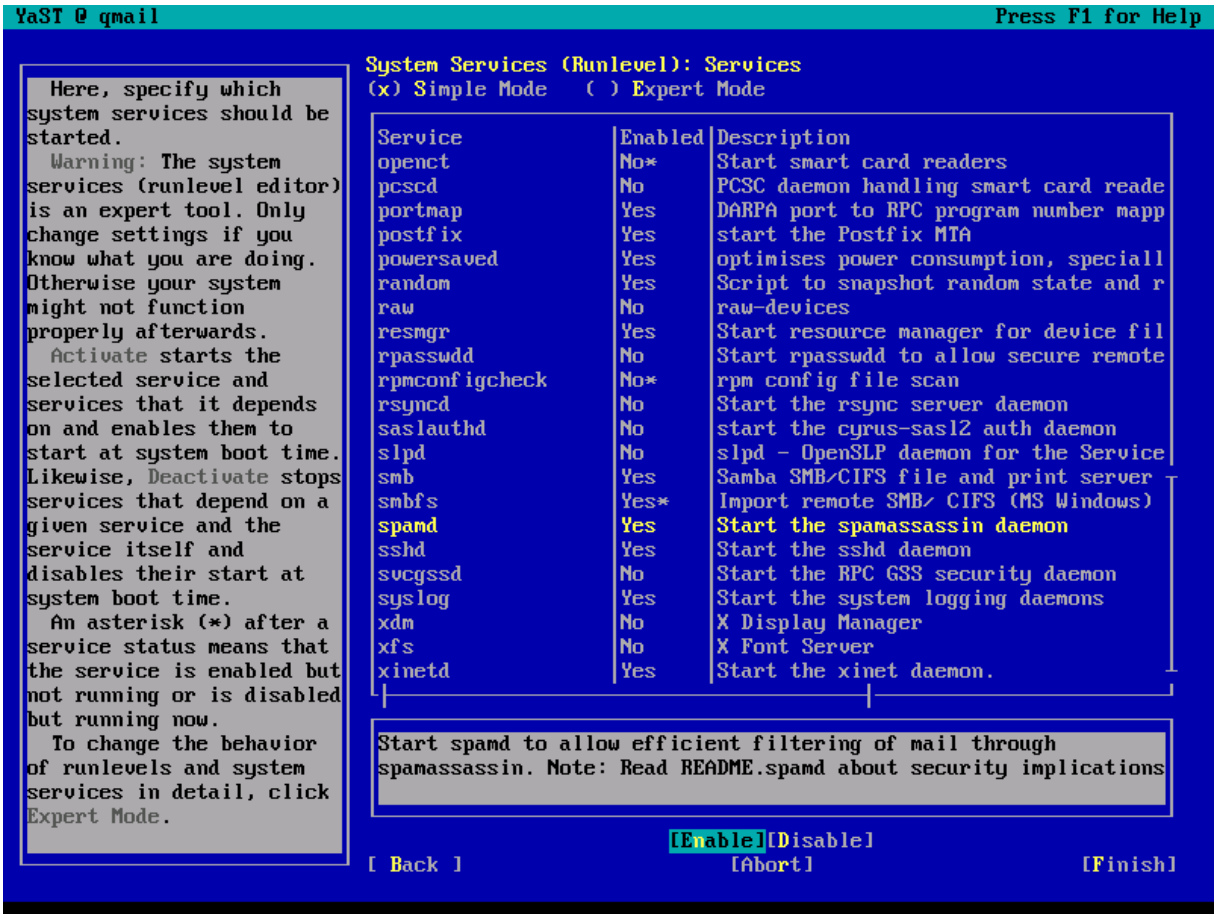
Konfiguration

Zunächst ist der zu SpamAssassin gehörige Dienst spamd zu starten. Dies kann mit Hilfe des YAST-Moduls „System Services (Runlevel)“ aus der Kategorie „System“ durchgeführt werden:

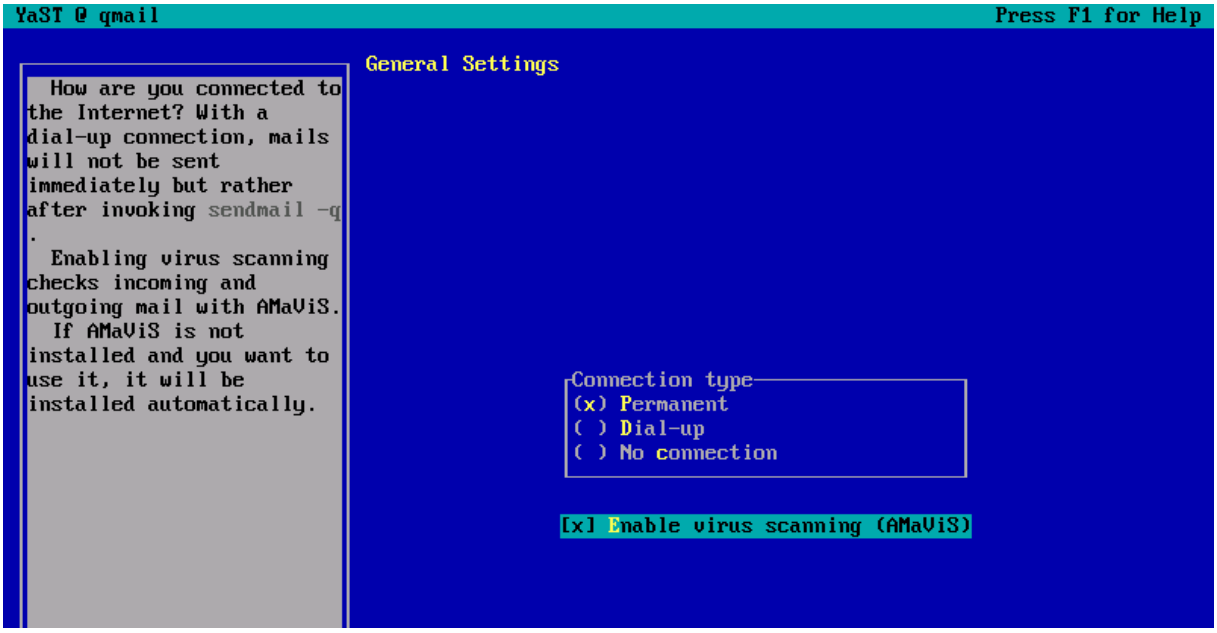


[3] Vgl. http://en.wikipedia.org/wiki/DNSBL
[4] Vlg. http://www.openspf.org

Nun ist aus der Liste das Service „spamd“ auszuwählen und die Option „Enable“ zu aktivieren:



Nun hat die Einbindung von Amavis in Postfix zu erfolgen. Zu diesem Zweck ist aus der Kategorie „Network Services“ das Modul „Mail Transfer Agent“ zu starten. Auf dem nun dargestellten Screen ist die Option „Enable Virus Scanning (AMaViS)“ zu aktivieren:



Durch zweimaliges „Next“ und abschließendes „Finish“ ist die Konfiguration von SpamAssassin und AMaViS abgeschlossen

Virenschutz

Allgemeines

Um der täglichen Flut von E-Mail-Viren Herr zu werden ist ein Virenschutz unumgänglich.

Installation von AntiVir

Um AntiVir zu installieren, muss lediglich ein Paket namens „antivir“ installiert werden. Amavis erkennt nach erfolgter Installation das nun zur Verfügung stehende Anti-Viren-Programm gleichsam automatisch.

Eine Demonstration des beliebten Open-Source Viren-Scanners Clam AV muss leider unterbleiben, da sich Installation (d.h. Kompilation) und Konfiguration unter SUSE von Hand erfolgen müsste und hier zu vermeidende Komplexitäten aufweist.

IT-Sicherheit unter Linux

Allgemeines

Sicherheit ist ein, zunehmend an Wichtigkeit gewinnender Bereich der Informationstechnologie. Es wird geschätzt, dass weltweit bereits ca. die Hälfte des IT-Budgets für IT-Sicherheit aufgewendet wird. Linux ist als Betriebssystem bestens dafür ausgerüstet das stetig wachsende Sicherheitsbedürfnis von Managern und IT-Spezialisten zu befriedigen. Anerkannter Weise sind drei Aspekte der IT-Sicherheit zu unterscheiden:

- *Vertraulichkeit von Daten (Confidentiality)*
- *Integrität von Daten (Integrity)*
- *Verfügbarkeit von Daten (Availability)*

Aus den englischen Bezeichnungen ergibt sich die leicht zu merkende „Eselsbrücke“ CIA.
Im Folgenden soll auf spezielle Bereiche der IT-Sicherheit im Zusammenhang mit Linux eingegangen werden.

Authentifizierung: Passwörter & Public Key Authentication

Passwörter stellen eine der häufigsten Sicherheitsvorkehrungen dar. Aus diesem Grund stellen sie ein beliebtes Angriffsziel dar. Mittels sog. Brute Force Attacks versuchen Angreifer sich Zugang zu einem System zu verschaffen indem sich alle nur möglich Passwörter nacheinander probieren. Bei schlechten Passwörtern kann dies bereits zum Erfolg führen. Eine etwas versiertere Methode ist der sog. Dictionary Attack. Hierbei verwendet der Angreifer ein sog. Password Dictionary, das häufig verwendete Passwörter (z.B. Charaktere aus erfolgreichen Filmen etc) enthält. Um diesen und anderen Gefahren für eine Passwort-Authentifizierung aus dem Weg zu gehen, sollte wann immer möglich eine Public Key Authentication zu Einsatz kommen. Insbesondere bei Remote-Root-Logins via SSH ist eine solche leicht einzurichten und zu administrieren.

Ein User hat lediglich seinen Public Key (zu finden unter ~/.ssh/id_dsa.pub auf dem Rechner des Users) der Datei /root/.ssh/authorized_keys auf dem Server anzufügen.

Zugriffsrechte auf Dateiebene

Da Linux (wie alle anderen UNIX- und UNIX-ähnlichen Betriebssysteme) ein File-Based-Operating System ist, können nahezu alle für das System relevanten Zugriffsrechte auf Dateiebene festgelegt werden.
Der Befehl

```
ls –l
```

zeigt die, im aktuellen Verzeichnis befindlichen Dateien und Ordner und die entsprechenden Zugriffsrechte an:

```
drwx------  2 lukas users 4096 Feb 18 14:54 Documents
drwx------  6 lukas users 4096 Feb 20 06:17 Maildir
drwxr-xr-x  2 lukas users 4096 Feb 18 14:54 bin
```

```
drwxr-xr-x  2 lukas users 4096 Feb 18 14:54 public_html
```

Die erste Spalte enthält abgesehen vom ersten Zeichen (“d” steht für Directory) dreimal die Buchstaben rwx bzw. einen Bindestrich an Stelle eines oder mehrerer Zeichen. Das erste rwx-Set bezieht sich auf den in Spalte 3 angegebenen Owner der Datei; das zweite rwx-Set auf die in Spalte 4 angegebene Group und das dritte rwx-Set auf Others (d.h. alle anderen Benutzer die weder der Owner noch in der angegebenen Gruppe sind).

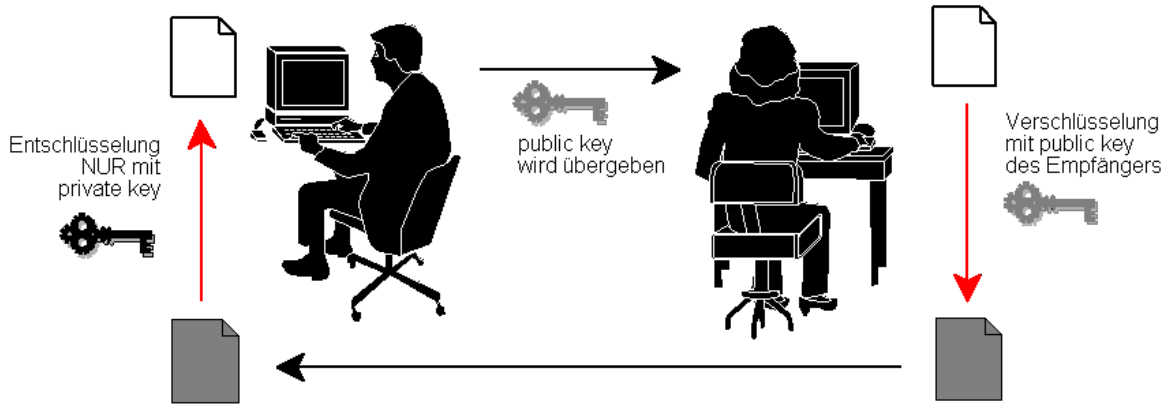
Besondere Aufmerksamkeit verdienen sog. SUID bzw. SGID-Scripts. Es handelt sich hierbei um Scripts bzw. Anwendungen, die grundsätzlich von jedem Benutzer mit root-Rechten ausgeführt werden können. Dies ist manchmal zur Erreichung einer, dem User root vorbehaltenen Funktionalität erforderlich. Weist ein SUID bzw. SGID-Script jedoch eine Sicherheitslücke auf, kann dies zur Kompromittierung des gesamten Systems führen. SUID steht für Set User ID und bewirkt, dass das Programm mit den Rechten des Users läuft, der Owner der Datei ist. SGID bewirkt demgegenüber, dass die Datei mit den Rechten der Group ausgeführt wird, die der Datei zugewiesen ist. SUID/SGID-Scripts lassen sich daran erkennen, dass ls –l für sie anstatt eines „x“ für Execute ein „s“ ausgibt. Ein Beispiel ist /bin/su. So liefert der Befehl

```
ls –l /bin/su
-rwsr-xr-x  1 root root 27648 Sep 10 05:56 /bin/su
```

Dies lässt erkennen, dass es sich um ein SUID-Programm handelt (an Stelle des „x“ des ersten rwx-Sets wird ein „s“ angezeigt).

Verschlüsselung mit GnuPG

GnuPG (GNU Privacy Guard) ist eine Open-Source-Implementierung von OpenPGP (RFC 2440). Es kommen hierbei sog. asymmetrische Verschlüsselungsverfahren zu Einsatz. Diese unterscheiden sich von symmetrischen Verfahren dadurch, dass der für Ver- und Entschlüsselung verwendete Key nicht derselbe ist – und daher in diesem Sinne eine Asymmetrie zwischen den beiden Seiten der Ver- und Entschlüsselung besteht. Der Vorteil von asymmetrischen Verfahren besteht darin, dass Sender und Empfänger der geheimen Nachricht nicht zuvor ein gemeinsames Passwort vereinbaren müssen. Denn dies ist bei unbekannten Kommunikationspartnern nicht möglich bzw. stellt sich allgemein wieder die Frage wie das Passwort auf sichere Weise (d.h. verschlüsselt) ausgetauscht werden könnte. Asymmetrische Verfahren lösen diese Probleme:



Wie in der Abbildung dargestellt übergibt der Empfänger der Nachricht seinen sog. Public Key dem Absender der Nachricht (bzw. veröffentlicht seinen Public Key im Internet). Da zu jedem Public Key auf komplementäre Weise ein Private Key passt, können mit dem einen Key verschlüsselte Daten nur mit dem bestimmten anderen Key entschlüsselt werden. So kann im oben dargestellten Beispiel nur der Empfänger der mit seinem Public Key verschlüsselten Nachricht diese entschlüsseln, da nur er den komplementären Private Key Besitzt.

Im Folgenden soll der Umgang mit GnuPG beschrieben werden:
Ein neues Schlüsselpaar kann mit dem Befehl

```
gpg --gen-key
```

generiert werden. Um den Public Key eines anderen Benutzers seinem Schlüsselbund hinzuzufügen ist der Befehl

```
gpg --import <public_key_file>
```

erforderlich. Befindet sich auf Ihrem Schlüsselbund ein Public Key mit der E-Mail-Adresse lukas.feiler@lukasfeiler.com so kann eine mit diesem verschlüsselte Nachricht wie folgt erzeugt werden:

```
cat plaintext.txt | gpg -er lukas.feiler@lukasfeiler.com > cyphertext
```

Um sich die verschlüsselte Nachricht am Terminal ausgeben zu lassen empfiehlt es sich die Option „a“ hinzuzufügen:

```
echo “This is my secret message” | gpg -ear lukas.feiler@lukasfeiler.com
```

Die derart erstellten verschlüsselten Nachrichten, können nun nur von jemandem entschlüsselt werden, der in Besitz des komplementären Private Key ist.

Datensicherung und Datenrettung

Allgemeines

Datenverluste können unterschiedliche Ursachen zurückzuführen sein. Das Spektrum reicht von Hardware-bedingten Festplattenfehlern zur vorsätzlicher Datenlöschung durch Hacker oder einer versehentlichen Datenlöschung durch Mitarbeiter. Ungeachtet der unterschiedlichen Bedrohungsszenarien ist die Anzuwendende Sicherheitsvorkehrung meist dieselbe: Backups.

Vor der Auswahl konkreter Backup-Tools sollte eine genaue Backup-Policy erstellt werden, die festhält was, wann, wie oft, weshalb und auf welches Medium gesichert werden soll. Darüber hinaus sollte jede Organisation die Möglichkeit des Off-Site-Storage von Backup-Medien in Erwägung ziehen. Denn die beste Backup-Lösung ist einem Einbrecher, der sowohl die Serverfestplatte als auch die Backup-Medien stiehlt meist hilflos ausgeliefert.

Das Backup-Tool tar

Das wohl beliebteste Open-Source-Werkzeug zum erstellen von Backups ist tar. Ähnlich wie ZIP oder RAR ermöglicht tar die Erstellung von (komprimierten) Dateiarchiven.
So erzeugt der Befehl

```
tar zcvf /root/home_backup_200602241600.tar.gz /home
```

ein, mittels gzip komprimiertes Archiv aller Dateien und Verzeichnisse unter /home. Zieht man die Komprimierung bzip2 gzip vor so ist die Option „z“ in „j“ zu wandeln:

```
tar jcvf /root/home_backup_200602241600.tar.gz /home
```

Ein zuvor erstelltes Backup kann nun durch den Befehl

```
tar zxvf /root/home_backup_200602241600.tar.gz
```

in das aktuelle Verzeichnis extrahiert werden. Wurde eine bzip2-Komprimierung gewählt so hat auch die Extraktion mit der Option „j“ zu erfolgen:

```
tar jxvf /root/home_backup_200602241600.tar.gz
```

Der Inhalt eines tar-Archivs (auch Table of Contents, TOC genannt) kann mit dem Befehl

```
tar ztf /root/home_backup_200602241600.tar.gz
```

bzw.

```
tar jtf /root/home_backup_200602241600.tar.gz
```

ausgegeben werden.

Datenrettungs-Tools

Zur Wiederherstellung von Dateien aus beschädigten Dateisystemen empfiehlt sich insbesondere das Werkzeug foremost (http://foremost.sourceforge.net).

Linux als Firewall

Allgemeines

Als Firewall bezeichnet man eine Sicherheitsvorkehrung, die in der Lage ist ein oder mehrerer Systeme vor bestimmten unautorisierten Zugriffen aus dem Netzwerk zu schützen.
Zum Zwecke eines besseren Verständnisses der Funktionsweise einer Firewall soll eine kurze Einführung in die vier Schichten der TCP/IP-Protokollfamilie erfolgen.

Die Netzwerkschicht: auf dieser Schicht finden sich unmittelbar mit der konkret verwendeten Hardware in Zusammenhang stehende Protokolle bzw. die Hardware selbst. Meist kommt Twisted-Pair Ethernet zum Einsatz.

Die Internetschicht: auf dieser Schicht befindet sich das Protokoll IP (Internet Protocol). Dieses ermöglicht die Herstellung einer logischen Verbindung zwischen zwei Computersystemen. Auf der Ebene von IP spielen insbesondere die Absender- und die Zieladresse (Source und Destination-IP) ein große Rolle. Auf Firewalls kann leicht eine Filterung anhand einer der beiden vorgenommen werden.

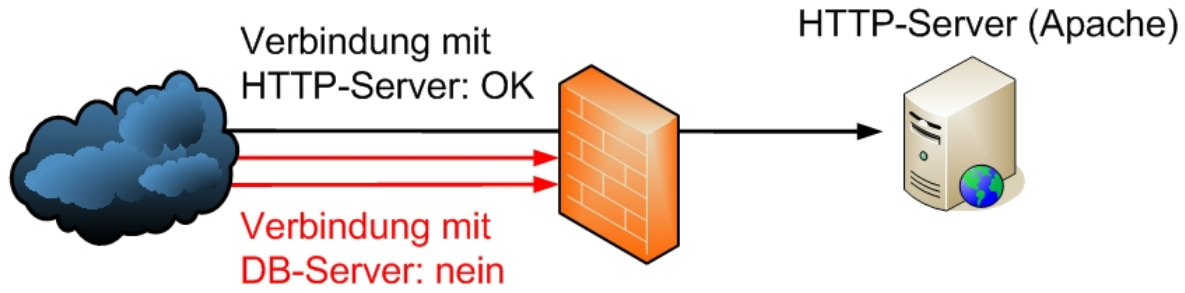
Die Transportschicht: auf dieser Schicht kommen die Protokolle TCP (Transmission Control Protocol) und UDP (User Datagram Protocol) zum Einsatz. Beide Protokolle besitzen konträre Eigenschaften.
TCP ist gekennzeichnet durch:

- Verbindungsorientiertheit: um Daten über TCP austauschen zu können, ist vorab explizit durch das Hin-und-her-Senden dreier Pakete (sog. Three-Way-Handshake) eine Verbindung aufzubauen.
- Zuverlässigkeit: TCP stellt sicher, dass alle versendeten Daten auch Tatsächlich ankommen. Um dies zu ermöglichen muss grundsätzlich der Empfang jedes Paketes bestätigt werden, was sich negativ auf die Performance auswirkt.
- Es bietet darauf aufbauenden Protokollen einen ununterbrochenen Datenstrom: Protokolle der Applikationsschicht (z.B. HTTP) müssen sich nicht einzelnen Paketen beschäftigen – dies besorgt TCP.

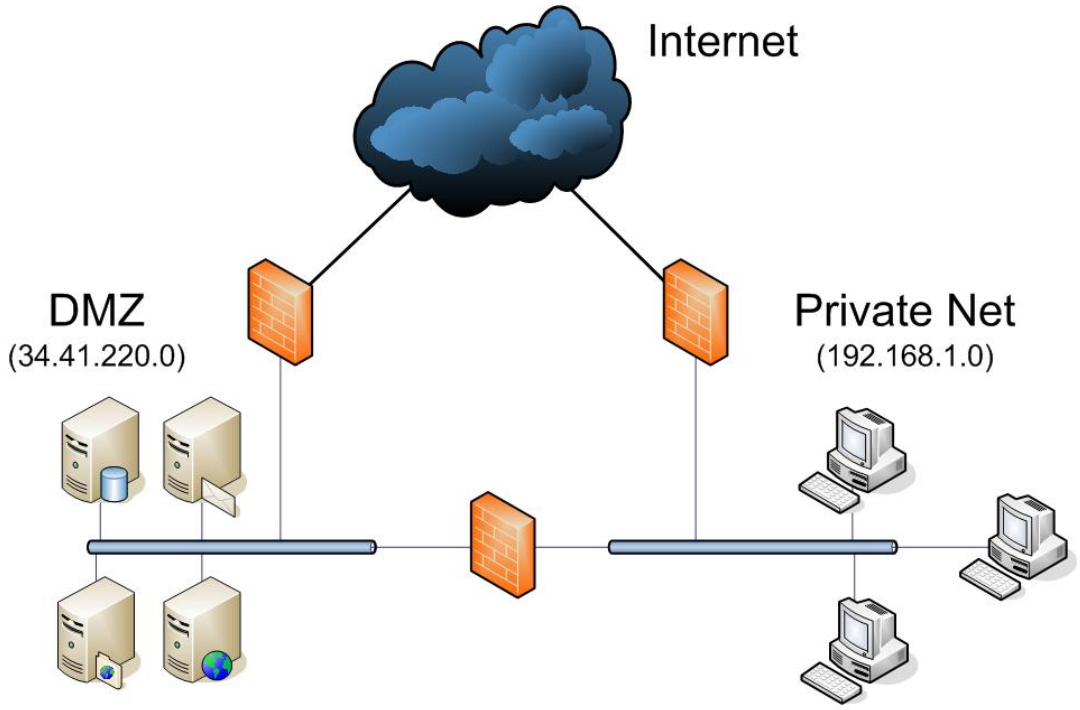
Demgegenüber weist UDP all diese Eigenschaften nicht auf. Es ist daher zum einen performanter zum anderen jedoch weniger „sicher“.
Für beide Protokolle gilt jedoch, dass sie in jedem Datenpaket einen Quell- und einen Ziel-Port (Source und Destination-Port) enthalten. Dies ermöglicht es dem empfangenden Betriebssystem die eingehenden Pakete bestimmten Diensten zuzuordnen. So sind beispielsweise Pakete an den Port 80 per Konvention an einen HTTP-Server (z.B. Apache) gerichtet. Anhand der Port-Informationen kann auf der Firewall eine vortreffliche Filterung erfolgen.
Meist wird auch das Protokoll ICMP (Internet Control Message Protocol) zur Transportschicht gezählt. Im Gegensatz zu TCP und UDP können jedoch keine anderen Protokolle auf Applikationsebene auf ICMP aufbauen. ICMP dient ausschließlich dem Versand von Fehlernachrichten bzw. im Fall von Echo-Request und Echo-Reply-Paketen (sog. „ping“) zur Überprüfung der Netzwerkfunktionalität.

Die Applikationsschicht: Auf dieser Schicht befinden sich alle Applikationsprotokolle wie HTTP, HTTPS, FTP, SMTP, POP3, IMAP, DNS, SSH, telnet, NFS oder NTP.

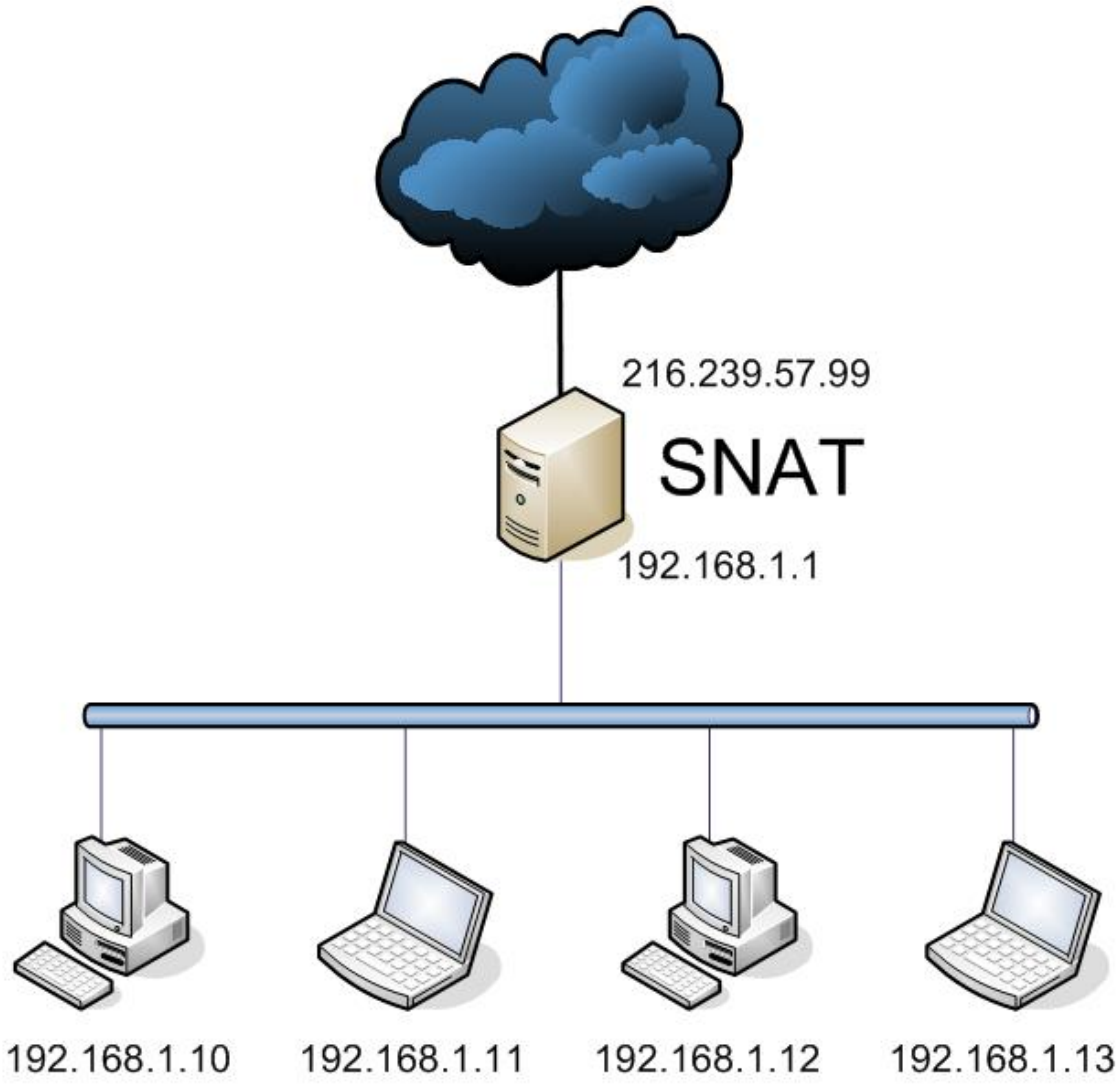
Nach dieser kurzen Einführung in die Schichten der TCP/IP-Protokollfamilie sollen verschiedene Einsatzszenarien von Firewalls besprochen werden. In der folgenden Abbildung wurde eine Firewall vor einem HTTP-Server eingerichtet um zu verhindern, dass anderer Datenverkehr als HTTP zum Server gelangt. Läuft beispielsweise auf dem HTTP-Server auch eine Datenbank (z.B. MySQL) so soll die Firewall Verbindungsversuche zu dieser blockieren. Dies ist leicht durch die Inspektion der Ziel-Ports eingehender TCP-Pakete möglich. In Worten ausgedrückt muss die Firewallkonfiguration daher etwa so lauten: „Lasse nur TCP-Pakete hindurch, die an den Port 80 (HTTP) adressiert sind“. Pakete, die an andere Ports (z.B 3306 für MySQL) gerichtet sind werden hingegen von der Firewall blockiert.



Häufig ist es jedoch erforderlich Server nicht nur vor Angriffen von außen (d.h. aus dem Internet) sondern auch vor Angriffen von Innen durch die eigenen Mitarbeiter zu schützen. Zu diesem Zweck kommt meist eine sog. DMZ (Demilitarized Zone). Hierbei schützt eine Firewall vor Angriffen aus dem Internet und eine andere vor Angriffen von innen:

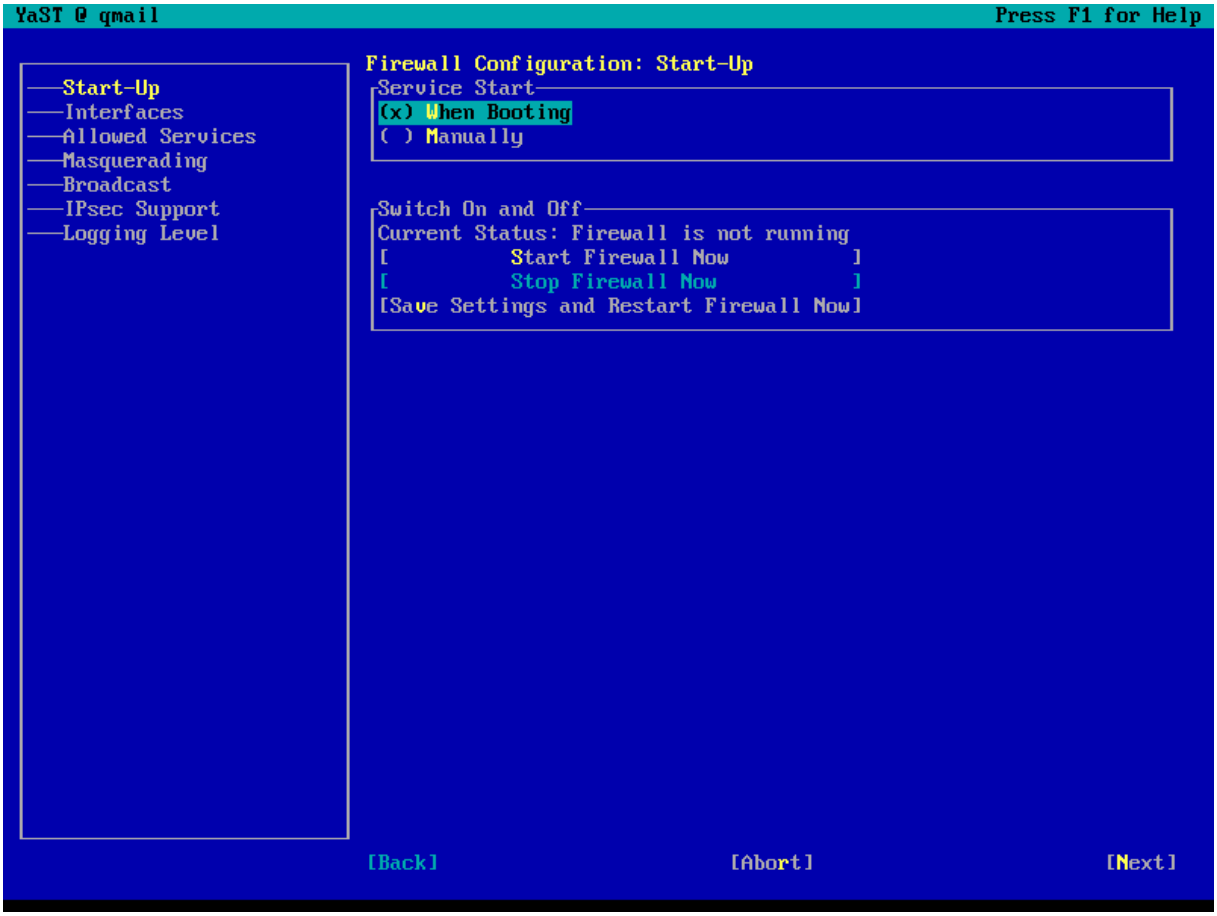


Ein weiterer, in Zusammenhang mit Firewalls höchst wichtiger Aspekt ist (S)NAT (Source Network Address Translation). In folgender Abbildung ist das private Netzwerk 192.168.1.0 dargestellt. Der auch als Firewall agierende Router ist mit dem privaten Netzwerk durch die IP-Adresse 192.168.1.1 und mit dem Internet durch die öffentliche IP-Adresse 216.239.57.99 verbunden. SNAT ist die Lösung für folgendes Problem: private IP-Adressen werden im Internet nicht geroutet (d.h. nicht weitergeleitet). Die Funktionsweise von SNAT kann folgender Weise beschrieben werden:
Soll ein Datenpaket aus dem internen Netzwerk an einen Server im Internet gesandt werden, so wird es zunächst an die SNAT-fähige Firewall, die bei den Clients als Default-Gateway eingetragen ist, übermittelt. Die Firewall verändert hierauf die Absenderadresse des Datenpakets (von beispielsweise 192.168.1.10 zu der öffentlichen IP-Adresse der Firewall, diesfalls 216.239.57.99). Das so veränderte Datenpaket wird dann an den gewünschten Server im Internet übermittelt. Der Server schickt dann seine Antwort-Pakete an die Quell-Adresse der Pakete (diesfalls 216.239.57.99). Kommen die Pakete an der Firewall an, erkennt sie diese als zu einer bestimmten Verbindung gehörig und ändert nun die Ziel-Adresse von 216.239.57.99 in 192.168.1.10 wodurch sie an den ursprünglichen Absender geleitet werden können.

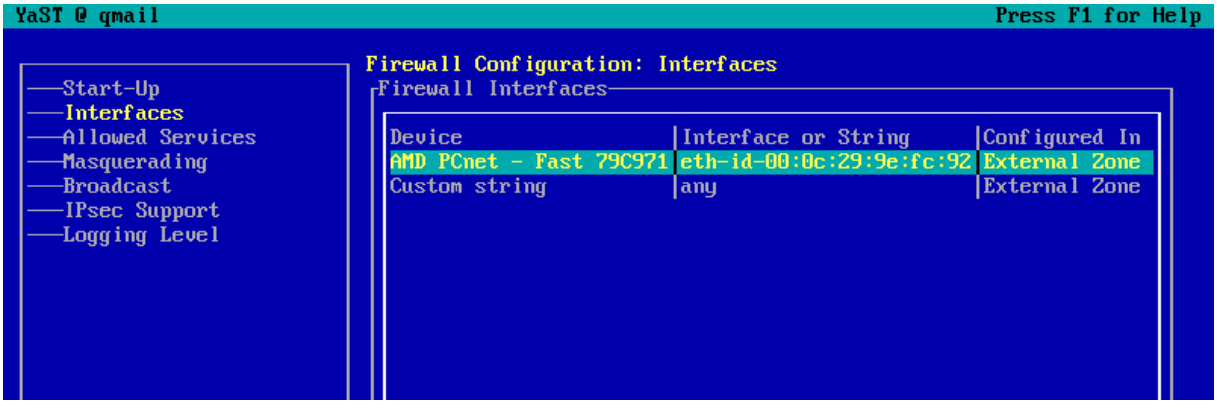


Einrichten einer Host-based Firewall unter SUSE Linux 10

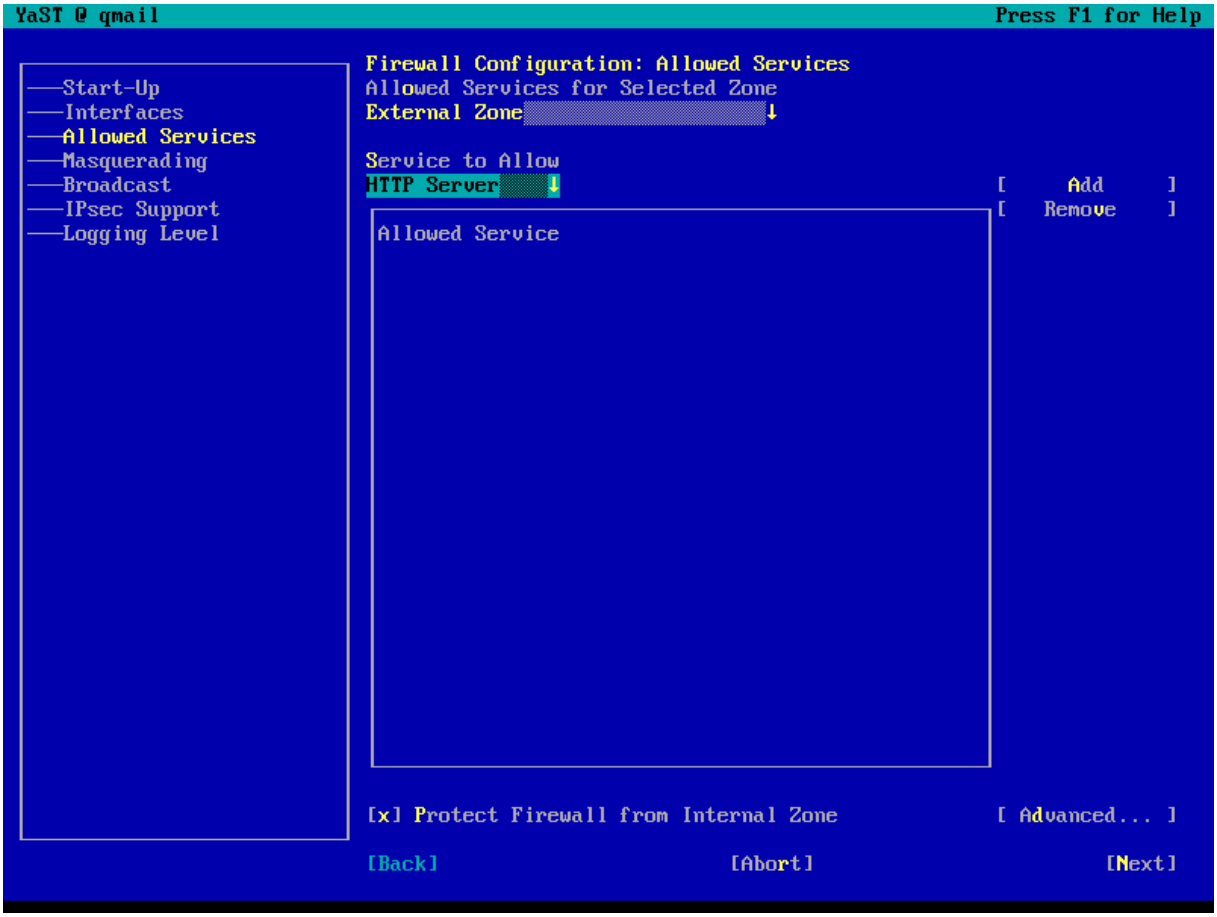
Eine sog. Host-based Firewall läuft auf dem zu schützenden Rechner selbst. Eine solche sollte neben einer Network-based Firewall auf Servern stets zum Einsatz kommen. Unter SUSE Linux kann die Konfiguration der Firewall mit YAST durchgeführt werden. Hierzu ist aus der Kategorie „Security and Users“ das Modul „Firewall“ zu starten. Dieses teilt das Interface in ein linkes und ein rechtes Frame. Je nach Auswahl eines Menüpunktes aus dem linken Frame werden im rechten Frame die entsprechenden Konfigurationsoptionen angezeigt. Zunächst ist unter „Start-Up“ die Option „Start Service When Booting“ zu empfehlen um die Firewall automatisch starten zu lassen:



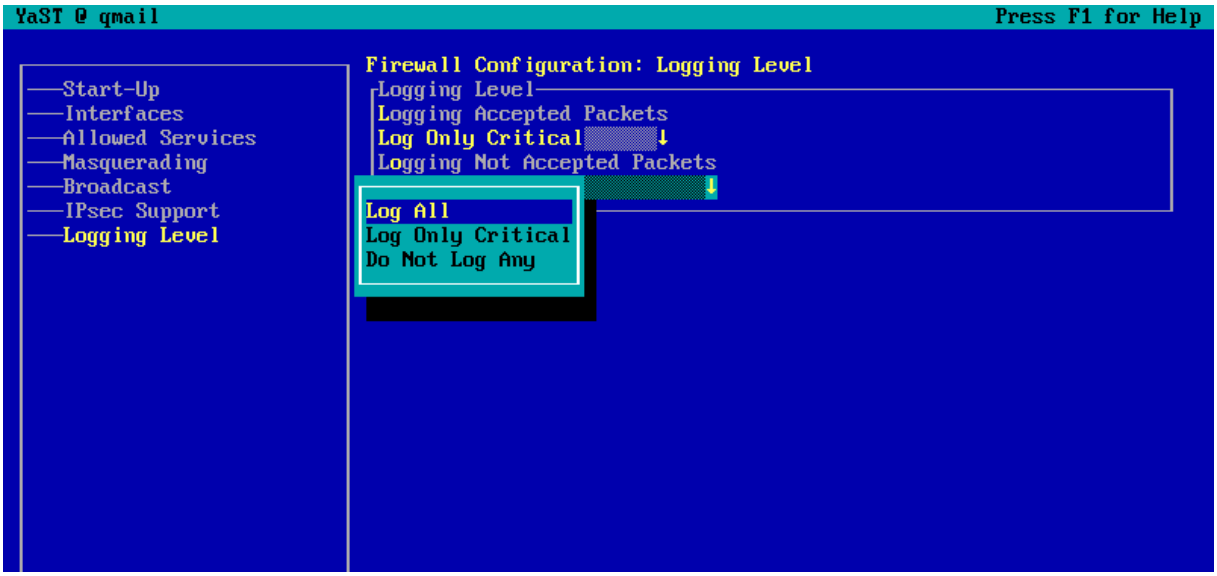
Unter dem Menüpunkt „Interfaces“ findet sich eine Liste der, am Server installierten Network Interfaces bzw. Netzwerkkarten. Ein durchschnittlicher Server ist nur mit einer Netzwerkkarte ausgestattet. Diese sollte in der Spalte „Configured in“ den Wert „External Zone“ anzeigen:



Nun können unter „Allowed Services“ die zuzulassenden Dienste definiert werden. Für "Allowed Services for Selected Zone" sollte „External Zone“ ausgewählt sein. Nun kann unter „Service to Allow“ der gewünschte Dienst (z.B. HTTP) ausgewählt und mittels „Add“ der Liste der zugelassenen Dienste hinzugefügt werden. Dieser Vorgang ist mit jedem Dienst zu wiederholen, der von außen zugänglich sein soll. Abschließend ist darauf zu achten, dass „Protect Firewall from internal Zone“ aktiviert ist. Folgende Abbildung zeigt das Hinzufügen eines Dienstes:



Unter dem Menüpunkt „Logging Level“ sollte „Logging Not Accepted Packets“ jedenfalls auf „Log All“ gestellt werden um die Funktionsfähigkeit der Firewall jederzeit leicht überprüfen zu können:



Der Default aller andern Einstellungen kann beibehalten werden. Nun ist im Menüpunkt „Start-Up“ die Option „Save Settings and Restart Firewall Now“ auszuwählen. Danach kann mit „Next“ und am darauf folgenden Screen, der eine Zusammenfassung präsentiert mit „Accept“ die Konfiguration der Firewall abgeschlossen werden.

Die Funktionalität der Firewall sollte am besten manuell verifiziert werden. Zu diesem Zweck sollte auf der Firewall bzw. dem Server der Befehl

```
tail –f /var/log/firewall
```

abgesetzt werden. Die Option –f bewirkt, dass tail nicht bloß die letzten zehn Zeilen der Datei ausgibt, sondern sich gleichsam an das Ende der Datei „dranhängt“ und alle an das Ende der Datei hinzugefügten Zeilen ausgibt.

Auf einem entfernten Rechner kann nun versucht werden sich zu gesperrten Ports zu verbinden. Am ist dies auf allen Betriebssystemen unter Verwendung des Tools telnet möglich. Ist beispielsweise der Port 21 (FTP) nicht freigegeben so kann mit folgendem Befehl versucht werden, sich dennoch zu Port 21 zu verbinden:

```
telnet <SERVER_IP> 21
```

Der am Server zuvor abgesetzte tail –f Befehl sollte nun neue Log-Einträge anzeigen, die festhalten, dass ein eingehendes Datenpaket an den Port 21 blockiert wurde. Ein derartiger Log-Eintrag sieht etwa so aus:

```
Feb 22 18:25:11 mail kernel: SFW2-INext-DROP-DEFLT IN=eth0 OUT=
MAC=00:0c:29:9e:fc:92:00:c0:9f:1f:59:99:08:00 SRC=192.168.1.66
DST=192.168.1.86 LEN=48 TOS=0x00 PREC=0x00 TTL=128 ID=22793 DF
PROTO=TCP SPT=3159 DPT=21 WINDOW=65535 RES=0x00 SYN URGP=0 OPT
(020405B401010402)
```

Hieraus ist ersichtlich, dass das blockierte Datenpaket von 192.168.1.66 (SRC - Source) an 192.168.1.86 (DST - Destination) über das Protokoll TCP (PROTO) vom Port 3159 (SPT – Source Port) an den Port 21 (DPT – Destination Port) geschickt wurde. Dass es sich um den Versuch eines Verbindungsaufbaus handelt ist daraus erkenntlich, dass nur das SYN-Bit ohne ACK (f. Acknowledgement) gesetzt war.

Empfohlene Literatur/Quellen

Welsh/Dalheimer/Kaufman, Running Linux, 3rd Edition, O'Reilly & Associates, 1999

Peikari/Chuvakin, Security Warrior, O'Reilly & Associates, 2004

Barrett/Silverman, SSH – The Secure Shell, O'Reilly & Associates, 2001

Preston, Unix Backup & Recovery, O'Reilly & Associates, 1999

Frisch, Unix System Administration, 2. Aufl, O'Reilly & Associates, 2000

Garfinkel/Spafford, Practical Unix and Internet Security, 3rd Edition, O'Reilly & Associates, 2003

Dent, Postfix: The Definitive Guide, O'Reilly, 2003

Stanfield/Smith, Linux System Administration, Sybex, 2001

Kirch/Dawson, Linux Network Administrator's Guide, 2ed Edition, O'Reilly & Associates, 2000

Zwicky/Cooper/Chapman, Building Internet Firewalls, 2ed Edition, O'Reilly & Associates, 2000

Albitz/Liu, DNS and BIND, 4th Edition, O'Reilly & Associates, 2001

Northcutt/Zeltser/Winters/Fredrick/Ritchey, Inside Network Perimeter Security: The Definitive Guide to Firewalls, VPNs, Routers, and Intrusion Detection Systems, New Riders Publishing, 2002

Schwartz, SpamAssassin, O'Reilly & Associates, 2004