Rauchs Wiki

Inhalt

Artikel

Quellennachweise

Artikellizenzen

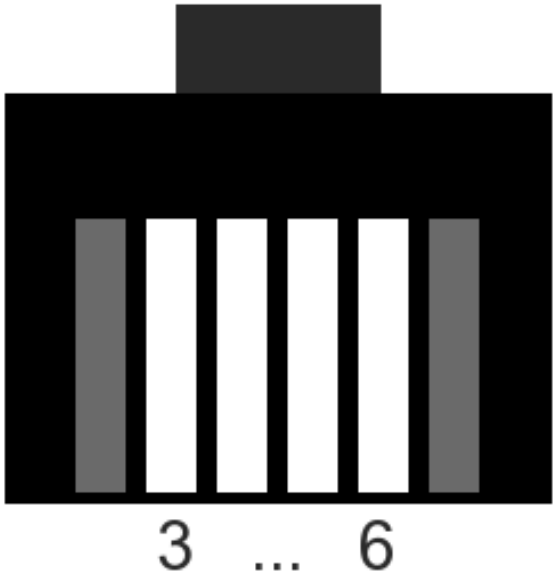
Ascend DSL Pipes

Ascend DSL Pipes

Da es Ascend leider nicht mehr gibt und Dokumentation etwas rar ist, hier die Belegung des Kabels zur Verbindung von 2 Ascend DSL-Pipes mit RJ11-WAN-Port

Achtung: Es existieren auch DSL-Pipes mit RJ45-WAN-Port! Ich weiss leider nicht, wie diese beschalten sind! Angeblich sollen dort die Adern 3 und 4 belegt werden!

Für die DSL-Pipes mit RJ-11 gilt folgende Belegung ausgehend von einem 4-poligen Stecker:



Hier müssen die Adern 4 und 5 belegt werden, und zwar an beiden Steckern _ohne_ Kreuzung! Ich habe dazu einfach 2 Adern eines CAT5-Twisted Pair-Kabels verwendet und das funktioniert problemlos mit voller Geschwindigkeit der DSL-Pipes (2,3 MBit)

Diese 2 Ascend DSL Pipes sind jetzt im Produktiveinsatz und binden ein etwas entferntes Gebäude (ca. 100m) an ein bestehendes Netzwerk über 2 Adern eines normalen 12-poligen Erdkabels an. Auch über diese Verbindung haben wir die volle Geschwindigkeit der DSL Pipes von 2,3 MBit gemessen.

Rauch 21:58, 14. Jul 2006 (CEST)

Faxen mit OpenOffice

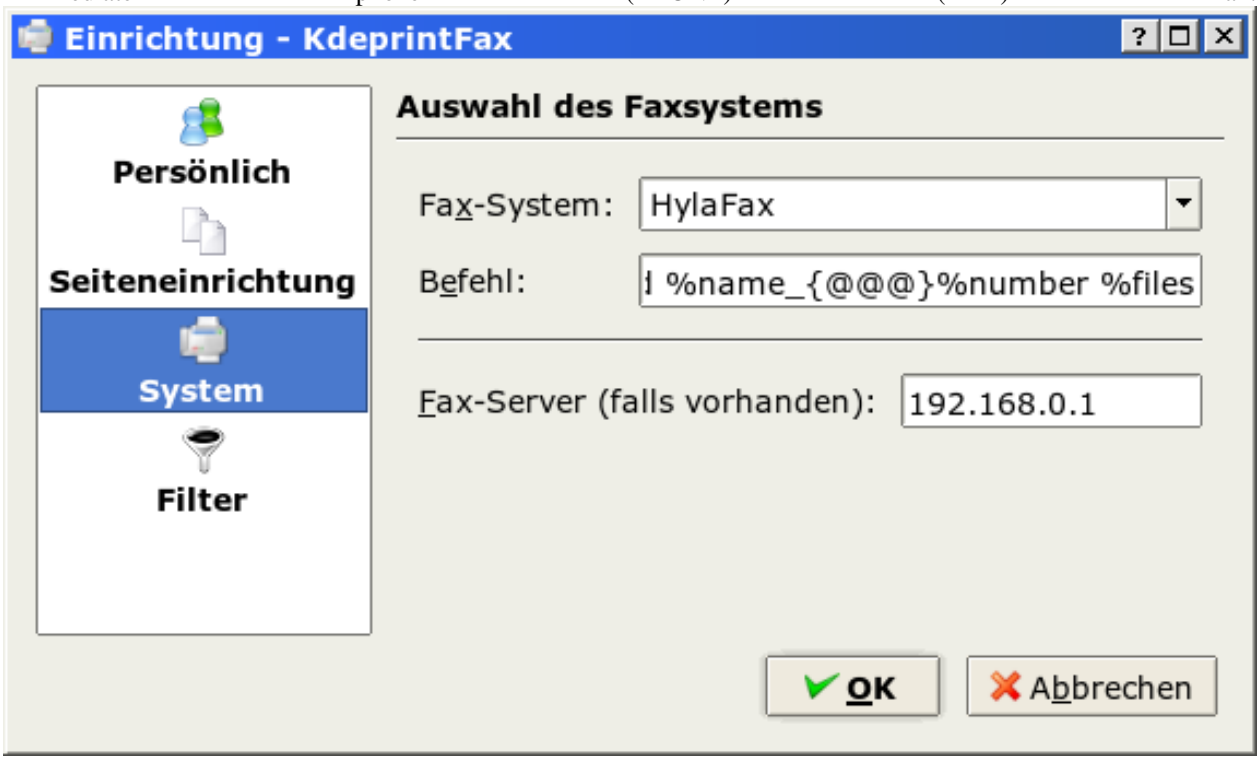
Die Einrichtung eines Faxservers mit Hylafax ist kein Problem dank der Anleitung unter [[1]].

Das einzige, was etwas Suchen und Nachdenken erforderte, war die Einrichtung des SuSE 9.3 Clients mit OpenOffice.

Nach Installation von kdegraphics3-fax sowie hylafax-client (nicht das Paket sendfax installieren, da mit diesem der Versand aus kdeprintfax fehlschlägt!) muss das Programm /usr/lib/ooo-2.0/program/spadmin aufgerufen werden und ein neuer Faxdrucker eingerichtet werden.

Einfach "Neuer Drucker" anwählen und in der Eingabemaske als Auswahl die Option "eine Faxlösung anbinden" auswählen. Als Treiber soll der Standardtreiber verwendet werden.

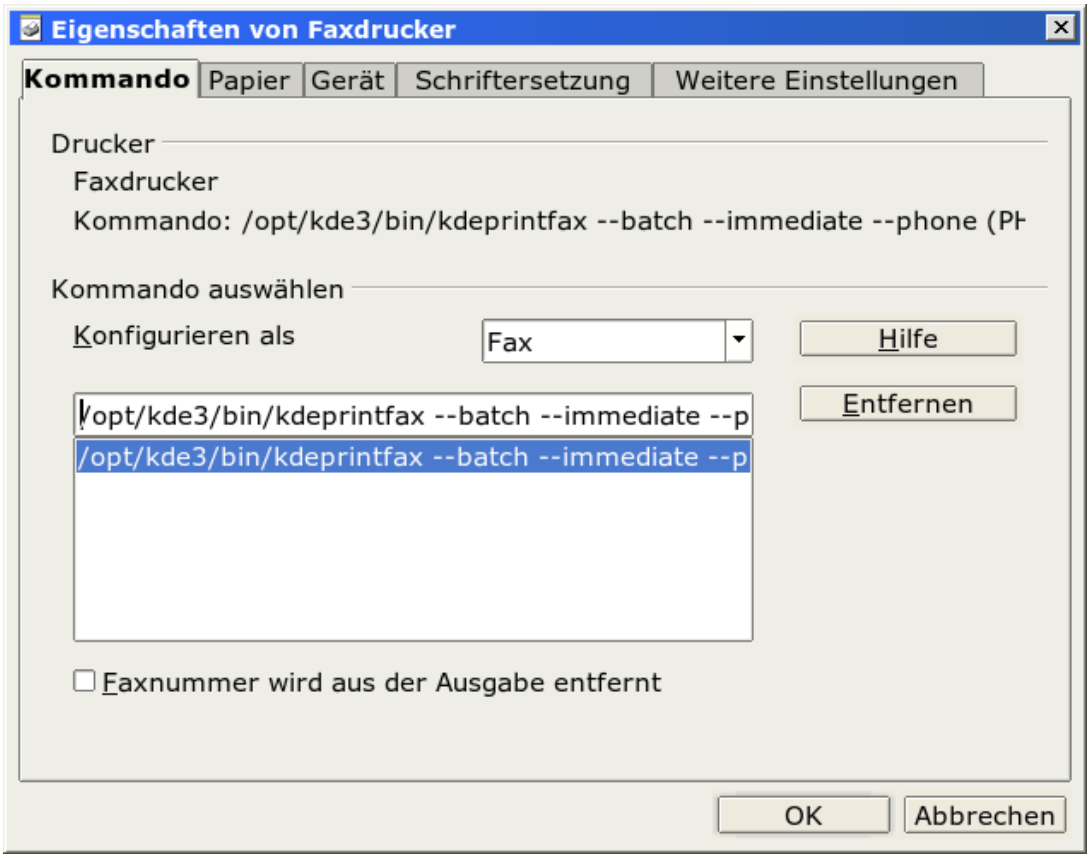
Als Kommandozeile zum Drucken geben Sie unter SuSE 9.3 den Pfad "/opt/kde3/bin/kdeprintfax --batch --immediate --phone (PHONE) (TMP)" an.



Der Parameter --batch steht hierbei dafür, das das KDEPrintFax-Fenster sofort wieder geschlossen werden soll, --immediate für den sofortigen Faxversand, (PHONE) ist die Telefonnummer die in OpenOffice angegeben wird, und (TMP) das nach PostScript umgewandelete Dokument von OpenOffice.

Bevor das erste Mal ein Fax verschickt wird, sollten Sie erst einmal KDEPrintFax aufrufen und ebenfalls konfigurieren. Unter "Einstellungen ->KdePrintfax einrichten" füllen Sie mindestens die Felder Name unter Persönlich, sowie unter System das Feld Faxserver aus, sofern Ihr Hylafax-Server auf einem anderen PC läuft. Alle anderen Einstellungen kÃ¶nnen so übernommen werden.

So, jetzt können Sie direkt aus OpenOffice faxen!



Quellennachweise

[1] http://www.hylafax.org/howto/index.html

Setup Suse 10.2 Server

HowTo: Setup an openSUSE 10.2 Server

- Do an minimal install (no graphical system).

I have deactivated the SUSEFirewall-script at install, as I use another system for this, but that's only personal favour ;) I also deactivated IPv6 and skipped the registration- and update-source-step, as we will use smart here.

- configure your network

call 'yast lan', choose traditional setup and set your IP, DNS, hostmask and whatever you need.

I will use the address 192.168.0.2/24 in this example with 192.168.0.1 as gateway, as this howto was written using a virtual machine. hostname is set to server, domain to example.org

- install smart

```
server:~ # mkdir downloads && cd downloads
server:~/downloads # wget ftp://ftp.gwdg.de/pub/linux/misc/suser-guru/smart/10.2/i586/smart-latest.rpm
server:~/downloads # yast -i rpm-python
server:~/downloads # rpm -ivh smart-latest.rpm
```

Now this was the last time, we used YaST :)

- grab the content of the following box and save to ~/channels.conf

```
[suse-updates]
type = rpm-md
name = openSUSE 10.2 Updates
baseurl = ftp://ftp.gwdg.de/pub/linux/suse/ftp.suse.com/suse/update/10.2/

[suse]
type = yast2
name = openSUSE 10.2 OSS
baseurl = ftp://download.opensuse.org/distribution/10.2/repo/oss/

[suse-non-oss]
type = yast2
name = openSUSE 10.2 Non-OSS
baseurl = ftp://download.opensuse.org/distribution/10.2/repo/non-oss/
```

and execute

```
server:~ # smart channel --add ~/channels.conf
```

Answer 'y' to this three entries to add these channels to smart.

Let's go on:

```
server:~ # smart update && smart upgrade
```

If there was an kernel update, reboot now. To the time of writing, there was none, so I'll go on here.

Now proceed:

- DNS with bind
- Network time protocol daemon

- DHCP configuration with DDNS
- Apache, MySQL & PHP
- Mailserver with Postfix, MySQL & Dovecot
- Configure Mailman
- Security
- FTP Server
- rsync daemon
- More to come soon...

Setup Suse 10.2 Server/Security

Security

We did not do anything for security yet, so here we go:

Firewall

First step in securing our server is setting up some firewall rules using iptables. I personally do not like the SuSEFirewall-Script, so I write my own rules in a shellscript, though feel free to use SuSEFirewall. if you feel more comfortable with it.

Below you can find an example for a simple iptables-script, though I can really recommend the Linux iptables Pocket reference [1] by O'Reilly.

This script is well-commented, so it should be quite self-explanatory.

```
#!/bin/bash
# your path to iptables
IPTABLES=/usr/sbin/iptables
# internal network interfaces and subnets
INTIF=lo
# your lan network interface
LAN=eth0
# your lan subnet
LOCALNET=192.168.0.0/24
# ANY-network (for everything coming from outwards), should be 0.0.0.0/0
ANYNET=0.0.0.0/0
# your WAN-interface (if this server is connected directly to the internet)
WAN=dsl0
# set default policies
$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT ACCEPT
$IPTABLES -P FORWARD DROP
# delete all previous rules
$IPTABLES -F
################################################
#       rules for incoming connections         #
################################################
# accept answers and established connections
```

```
$IPTABLES -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
################################################
#       loopback device settings               #
################################################
# open loopback device
$IPTABLES -A INPUT -i $INTIF -j ACCEPT
################################################
#       SSH                                    #
################################################
# allow SSH from internal network
$IPTABLES -A INPUT -s $LOCALNET -p tcp --dport 22 -j ACCEPT
# allow ssh from everywhere (only do this, if you really need to access your server from everywhere!)
$IPTABLES -A INPUT -s $ANYNET -p tcp --dport 22 -j ACCEPT
################################################
#       HTTP                                   #
################################################
# allow HTTP from everywhere
$IPTABLES -A INPUT -p tcp -s $ANYNET --dport 80 -j ACCEPT
# allow HTTPS from everywhere
$IPTABLES -A INPUT -p tcp -s $ANYNET --dport 443 -j ACCEPT
################################################
#       Mail                                   #
################################################
# allow SMTP from everywhere
$IPTABLES -A INPUT -p tcp -s $ANYNET --dport 25 -j ACCEPT
# allow POP3 from everywhere
$IPTABLES -A INPUT -p tcp -s $ANYNET --dport 110 -j ACCEPT
# allow IMAP from everywhere
$IPTABLES -A INPUT -p tcp -s $ANYNET --dport 143 -j ACCEPT
################################################
#       ICMP                                   #
################################################
# allow ping (you may want to change this to DROP)
$IPTABLES -A INPUT -p icmp -j ACCEPT
$IPTABLES -A OUTPUT -d $ANYNET -p icmp -j ACCEPT
################################################
#       Allow everything from internal net     #
################################################
# you may want to set this, if your server has 2 interfaces, one internal, one external
# $IPTABLES -A INPUT -s $LOCALNET -d $LOCALNET -p all -j ACCEPT
################################################
#       Drops and Logs                         #
################################################
# Log everything not allowed until now and drop it
$IPTABLES -A INPUT -p tcp -s $ANYNET -d $ANYNET -j LOG
$IPTABLES -A INPUT -p udp -s $ANYNET -d $ANYNET -j LOG
$IPTABLES -A INPUT -p tcp -s $ANYNET -d $ANYNET -j DROP
```

```
$IPTABLES -A INPUT -p udp -s $ANYNET -d $ANYNET -j DROP
################################################
#       Rules for outgoing traffic             #
################################################
# accept answers and established connections
$IPTABLES -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
$IPTABLES -A OUTPUT -p all -j ACCEPT
```

I saved this file as `/etc/init.d/iptables` and call it from `/etc/init.d/boot.local`.

SSH

Per default, SSH is quite open (meaning root login allowed, no limits on login tries, login with normal user password).

We will close all of these holes and also go one step further, but let's begin with editing `/etc/ssh/sshd_config`:

```
# only allow SSHv2
Protocol 2
# specify the keys for SSHv2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
# disallow root to login via SSH
PermitRootLogin no
# disconnect after 3 tries
MaxAuthTries 3
# do strict checking on users homedir permissions
StrictModes yes
# enable Authentication via RSA keys
RSAAuthentication yes
PubkeyAuthentication yes
# specify the file with the public key for our users
AuthorizedKeysFile      .ssh/authorized_keys
# disable various "bad" options
RhostsRSAAuthentication no
HostbasedAuthentication no
IgnoreUserKnownHosts yes
IgnoreRhosts yes
PasswordAuthentication no
PermitEmptyPasswords no
X11Forwarding no
# disable PAM authentification (as we only use keys now)
UsePAM no
# only allow login, if user is in group `ssh`
AllowGroups ssh
```

Generate a key for your ssh login (while logged in as normal user, **not** root, as we don't allow root to login via SSH):

```
user@server:~> ssh-keygen -t rsa -b 4096
Enter file in which to save the key(/home/user/.ssh/id_rsa):<ENTER>
Created directory '/home/user/.ssh'
```

```
Enter passphrase (empty for no passphrase):<your passphrase for the key>
Enter same passphrase again:<<your passphrase for the key>
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
<some hexadezimal string> user@server
user@server:~> cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Now copy the file id_dsa to your local machine to ~/.ssh/.id_rsa.server and `rcsshd reload` the SSH daemon on the server. Do not forget to `chmod 0700` it, otherwise SSH will deny to use it!

If you choose id_rsa as filename instead, you can skip the following step.

Edit ~/.ssh/config on your local machine:

```
Host = example.org
HostName = example.org
IdentityFile = ~/.ssh/id_rsa.server
```

So, everyone's back? You should be able to login to your server now from your local machine:

```
user@localmachine:~> ssh example.org
The authenticity of host 'example.org (192.168.0.2)' can't be established.
RSA key fingerprint is <some hexadezimal string>.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'example.org' (RSA) to the list of known hosts.
Enter passphrase for key '/home/user/.ssh/id_rsa.server':
Last login: Sun Feb 11 14:45:22 2007 from 192.168.0.4
Have a lot of fun...
user@server:~>
```

If you try to login now without a key, you will get something like the following:

```
user@localmachine:~> ssh example.org -l root
Permission denied (publickey,keyboard-interactive).
user@localmachine:~>ssh example.org -l user
Permission denied (publickey,keyboard-interactive).
```

So we are fine here now!

Next Step: Preventing script kiddies from spamming our logfiles (and getting some more information)

Fail2Ban

Download and install Fail2Ban from my repository:

```
server: # wget ftp://rauchs-home.de/suse/10.2/RPMS/noarch/fail2ban-0.7.7-0.rauch.3.SuSE1020.noarch.rpm
server: # rpm -ivh fail2ban-0.7.7-0.rauch.3.SuSE1020.noarch.rpm
```

Edit `/etc/fail2ban/jail.conf` to activate reactions to SSH:

```
# I enhanced bantime to one hour! You may want to set this to a few minutes for testing and increase later
bantime = 3600
[ssh-iptables]
enabled = true
```

```
filter  = sshd
action  = iptables[name=SSH, port=ssh, protocl=tcp]
          mail-whois[name=SSH, dest=your.email@example.org]
logpath = /var/log/messages
maxretry = 3
```

With this config, you will receive a mail each time a IP gets banned.

Start Fail2Ban and install the init script now:

```
server: # rcfail2ban start
server: # insserv fail2ban
```

knockd

Install knockd on your server and knock (the client) on your home machine

```
server: # smart install knockd
home: # smart install knock
```

Edit `/etc/knockd.conf`:

```
[options]
  UseSyslog
[opencloseSSH]
  sequence      = 2222:udp,3333:udp,4444:udp
  seq_timeout   = 15
  tcpflags      = syn
  start_command = /usr/sbin/iptables -I INPUT 1 -s %IP% -p tcp --dport ssh -j ACCEPT
  cmd_timeout   = 30
  stop_command  = /usr/sbin/iptables -D INPUT -s %IP% -p tcp --dport ssh -j ACCEPT
```

Import lines to change are `sequence` and `tcpflags`. Change `sequence` to a combination of your own choice and set `tcpflags` to `syn`.

The reason for only using `syn` is, that we use DROP on our Firewall rules, so there will be no ack from our server on those ports.

You might want to change `/etc/sysconfig/knockd` too, if you want to choose another device than eth0 to have knockd listening on:

```
KNOCKD_OPTIONS="-d -i ethx"
```

Start knockd and try knocking from your home machine while watching `/var/log/messages` on the server:

```
server: # rcknockd start
server: # tail -f /var/log/messages
<DATE> server knockd: starting up, listening on eth0
<DATE> server knocks: <your client ip> opencloseSSH: Stage 1
<DATE> server knockd: <your client ip> opencloseSSH: Stage 2
<DATE> server knockd: <your client ip> opencloseSSH: Stage 3
<DATE> server knockd: <your client ip> opencloseSSH: OPEN SESAME
<DATE> server knockd: opencloseSSH: running command: /usr/sbin/iptables -I INPUT 1 -s <your client ip> -p tcp --dport ssh -j ACCEPT
<DATE> server knockd: opencloseSSH: running command: /usr/sbin/iptables -D INPUT -s <your client ip> -p tcp --dport ssh -j ACCEPT
```

Just fine, knockd works!

Yet we have to adapt our firewall script to disallow ssh per default:

```
# disallow ssh from everywhere (we now use knockd for opening the port)
$IPTABLES -A INPUT -s $ANYNET -p tcp --dport 22 -j DROP
```

You may still want to open the SSH-port for one static ip (if you have one), so write this line **before** the line above (and replace 192.168.0.1 with your real IP ;)):

```
# allow ssh login from my static ip without knockd
$IPTABLES -A INPUT -s 192.168.0.1 -p tcp --dport 22 -j ACCEPT
```

That's it, nobody knows that you are running ssh, but you can still connect after knocking the correct sequence :)

LogWatch

Wanna get a summary on what's going on with your server? Try LogWatch:

```
server: # smart channel --add http://software.opensuse.org/download/server:/monitoring/openSUSE_10.2/server:monitoring.repo
server: # smart update && smart install logwatch
```

Edit `/etc/logwatch/conf/logwatch.conf`:

```
MailTo = your.email@example.org
```

We want LogWatch to run automatically just after midnight, so we call `crontab -e` as root and add this line:

```
15 0 * * * /usr/sbin/logwatch
```

Thus we will receive one mail at 00:15 each day about what happened the day before. If you want to know more about configuring Logwatch, take a look at the manual [2].

RKHunter

Install:

```
server: # smart install rkhunter
server: # rkhunter --update
```

As rkhunter does not fully support openSUSE 10.2 (yet?), you will only receive mails containing "Please inspect this machine, because it can be infected".

To fix this, follow the steps below:

- Modify `/etc/rkhunter.conf` and add a # before MAIL-ON-WARNING.
- Modify `/etc/cron.daily/suse.de-rkhunter` as below:

```
#!/bin/sh
/usr/bin/rkhunter --disable-md5-check --cronjob | mail -s 'rkhunter Daily run' your.email@example.org
```

That's all!

For more information visit Rootkit.nl [3].

SecCheck

SecCheck is a nice tool, to see, what has changed on your machine from one day to the other.

```
server: # smart install seccheck
```

Edit `/etc/sysconfig/seccheck`:

```
SECCHK_USER="your.mail@example.org"
```

Finished :)

Back to Suse 10.2 Server | proceed to FTP Server

Quellennachweise

[1] http://www.oreilly.de/catalog/lnxiptablespr/index.html
[2] http://www2.logwatch.org:81/tabs/docs/
[3] http://www.rootkit.nl/

Setup Suse 10.2 Server/DNS

Setting up bind

```
server:~ # smart install bind
```

This should install 2 packages, bind and bind-chrootenv.

Now edit /etc/named.conf and change the following settings:

```
  options {
    listen-on port 53 { 192.168.0.2; 127.0.0.1; }
    listen-on-v6 { none; }
    allow-query { any; }
  };
  include "/etc/named.d/logging.conf";
  include "/etc/named.d/example.org.conf";
```

Now a bit of security for our DNS:

```
server:~ # dnssec-keygen > /etc/rndc.conf
```

Copy the outcommented part of `/etc/rndc.conf` into `/etc/named.conf`

Now create `/etc/named.d/example.org.conf` with the following content:

```
zone "example.org" {
  type master;
  file "personal/example.org.db";
  allow-transfer { none; }:
};
zone "0.168.192.in-addr.arpa" IN {
  type master;
  file "reverse/192.168.0.0";
  allow-query { any; };
  allow-transfer { none; };
```

```
};
```

and `/etc/named.d/logging.conf`:

```
logging {

  channel default_file { file "/var/log/named/default.log" versions 3 size 5m; severity dynamic; print-time yes; };
  channel general_file { file "/var/log/named/general.log" versions 3 size 5m; severity dynamic; print-time yes; };
  channel database_file { file "/var/log/named/database.log" versions 3 size 5m; severity dynamic; print-time yes; };
  channel security_file { file "/var/log/named/security.log" versions 3 size 5m; severity dynamic; print-time yes; };
  channel config_file { file "/var/log/named/config.log" versions 3 size 5m; severity dynamic; print-time yes; };
  channel resolver_file { file "/var/log/named/resolver.log" versions 3 size 5m; severity dynamic; print-time yes; };
  channel xfer-in_file { file "/var/log/named/xfer-in.log" versions 3 size 5m; severity dynamic; print-time yes; };
  channel xfer-out_file { file "/var/log/named/xfer-out.log" versions 3 size 5m; severity dynamic; print-time yes; };
  channel notify_file { file "/var/log/named/notify.log" versions 3 size 5m; severity dynamic; print-time yes; };
  channel client_file { file "/var/log/named/client.log" versions 3 size 5m; severity dynamic; print-time yes; };
  channel unmatched_file { file "/var/log/named/unmatched.log" versions 3 size 5m; severity dynamic; print-time yes; };
  channel queries_file { file "/var/log/named/queries.log" versions 3 size 5m; severity dynamic; print-time yes; };
  channel network_file { file "/var/log/named/network.log" versions 3 size 5m; severity dynamic; print-time yes; };
  channel update_file { file "/var/log/named/update.log" versions 3 size 5m; severity dynamic; print-time yes; };
  channel dispatch_file { file "/var/log/named/dispatch.log" versions 3 size 5m; severity dynamic; print-time yes; };
  channel dnssec_file { file "/var/log/named/dnssec.log" versions 3 size 5m; severity dynamic; print-time yes; };
  channel lame-servers_file { file "/var/log/named/lame-servers.log" versions 3 size 5m; severity dynamic; print-time yes; };

  category default { default_file; };
  category general { general_file; };
  category database { database_file; };
  category security { security_file; };
  category config { config_file; };
  category resolver { resolver_file; };
  category xfer-in { xfer-in_file; };
  category xfer-out { xfer-out_file; };
  category notify { notify_file; };
  category client { client_file; };
  category unmatched { unmatched_file; };
  category queries { queries_file; };
  category network { network_file; };
  category update { update_file; };
  category dispatch { dispatch_file; };
  category dnssec { dnssec_file; };
  category lame-servers { lame-servers_file; };

};
```

PS: Thanks to the Gentoo Wiki [1] for this :)

But logging will not work yet, we have to create the files and folders:

```
server:~ # mkdir /var/lib/named/var/log/named/
server:~ # touch /var/lib/named/var/log/named/client.log
```

```
server:~ # touch /var/lib/named/var/log/named/config.log
server:~ # touch /var/lib/named/var/log/named/database.log
server:~ # touch /var/lib/named/var/log/named/default.log
server:~ # touch /var/lib/named/var/log/named/dispatch.log
server:~ # touch /var/lib/named/var/log/named/dnssec.log
server:~ # touch /var/lib/named/var/log/named/general.log
server:~ # touch /var/lib/named/var/log/named/lame-servers.log
server:~ # touch /var/lib/named/var/log/named/network.log
server:~ # touch /var/lib/named/var/log/named/notify.log
server:~ # touch /var/lib/named/var/log/named/queries.log
server:~ # touch /var/lib/named/var/log/named/resolver.log
server:~ # touch /var/lib/named/var/log/named/security.log
server:~ # touch /var/lib/named/var/log/named/unmatched.log
server:~ # touch /var/lib/named/var/log/named/update.log
server:~ # touch /var/lib/named/var/log/named/xfer-in.log
server:~ # touch /var/lib/named/var/log/named/xfer-out.log
server:~ # chown -R named:named /var/lib/named/var/log/named/
```

Next start bind with `rcnamed start`. If any errors come up, you most probably forgot an `;` somewhere.

But... nobody knows, what adresses we have yet?

Now edit `/var/lib/named/personal/example.org.db`

```
$TTL    86400

@       IN      SOA     ns.example.org. your.email.example.org. (
                        2007020901      ; Serial
                        10800   ; Refresh
                        3600    ; Retry
                        604800  ; Expire
                        86400 ) ; Minimum
example.org.          IN NS   ns.example.org.
ns.example.org.               IN A    192.168.0.2
example.org.          IN A    192.168.0.2
mail.example.org.             IN A    192.168.0.2
example.org.          IN MX  10 mail.example.org.
```

and `/var/lib/named/reverse/192.168.0.0`:

```
$TTL    86400

@       IN      SOA     example.org. (
                        2007020901      ; Serial
                        10800   ; Refresh
                        3600    ; Retry
                        604800  ; Expire
                        86400 ) ; Minimum

                 IN NS   example.org.
2               IN PTR  example.org.
```

```
2               IN PTR  server.example.org.
```

If you change and restart the name server, do not forget to increase the serial by 1!

Now we will test, if it actually works:

```
server:~ # rcnamed start
server:~ # nslookup example.org
Server:     192.168.0.2
Address:    192.168.0.2#53

Name:   example.org
Address: 192.168.0.2

server:~ # host example.org
example.org has address 192.168.0.2
example.org mail is handled by 10 mail.example.org
```

Back to Setup Suse 10.2 Server Next Step (NTP)

Quellennachweise

[1] http://gentoo-wiki.com/HOWTO_Setup_a_DNS_Server_with_BIND#Logging_conf

Setup Suse 10.2 Server/NTP

NTP Server

Now we will set up our local time server with editing `/etc/ntp.conf` and add the line below:

```
  server de.pool.ntp.org
```

You may want to choose a pool near your location from ISC [1]

Now start and install the NTP service:

```
server:~ # rcntp start
Try to get initial date and time via NTP from  de.pool.ntp.org     done
Starting network time protocol daemon                              done
server:~ # insserv ntp
```

Now NTP runs just fine and we are one step further, but still a long way to go...

Back to Setup Suse 10.2 Server | proceed to DHCP

Quellennachweise

[1] http://ntp.isc.org/bin/view/Servers/NTPPoolServers

Setup Suse 10.2 Server/DHCP

DHCP

First we need to install the DHCP-server:

```
server:~ # smart install dhcp-server
```

Now we edit `/etc/dhcpd.conf`:

```
authoritative;
ddns-update-style interim;

option subnet-mask 255.255.255.0;
option broadcast-address 192.168.0.255;
option routers 192.168.0.1;
option domain-name-servers 192.168.0.2;

option domain-name "example.org";

subnet 192.168.0.0 netmask 255.255.255.0 {
  range 192.168.0.151 192.168.0.200
  default-lease-time 259200;
  max-lease-time 518400;
}
```

You remember the DNS-configuration?

We will need to add another key info into dhcpd.conf and modify the DNS config too.

```
  server:~ # cd /etc &&  dnssec-keygen -a HMAC-MD5 -b 128 -n USER DHCP_UPDATER
```

This command will result in two files being created beginning with Kdhcp_updater. The file we need is the Kdhcp_updater*.private. View the content of this file and write down the key. Now add the following paragraph to your `/etc/dhcpd.conf` (don't forget to actually replace the key with yours):

```
  key DHCP_UPDATER {
    algorithm HMAC-MD5.SIG-ALG.REG.INT;
    secret H94JQHKbevJZwzd4OOTu5G==;
  };
  zone example.org. {
    primary 192.168.0.2;
    key DHCP_UPDATER;
  }
  zone 0.168.192.in-addr.arpa. {
    primary 192.168.0.2;
    key DHCP_UPDATER;
  }
```

Now edit `/etc/named.d/example.org` (which you've created before) and add the bold lines:

```
key DHCP_UPDATER {
  algorithm HMAC-MD5.SIG-ALG.REG.INT;
  secret H94JQHKbevJZwzd4OOTu5G==;
};
zone "example.org" {
  type master;
  file "personal/example.org.db";
  allow-transfer { none; }:
  allow-update { key DHCP_UPDATER; };
};
zone "0.168.192.in-addr.arpa" IN {
  type master;
  file "reverse/192.168.0.0";
  allow-query { any; };
  allow-transfer { none; };
  allow-update { key DHCP_UPDATER; };
};
```

As last step, edit `/etc/sysconfig/dhcpd`:

```
DHCPD_INTERFACE="eth0"
```

After you are done with this, reload named and (re)start dhcpd:

```
server:~ # rcnamed reload
server:~ # rcdhcpd start
```

Back to Setup Suse 10.2 Server | Proceed to Apache, MySQL & PHP

Setup Suse 10.2 Server/LAMP

Apache, MySQL & PHP

MySQL

```
server:~ # smart install mysql
```

You may want to edit `/etc/my.cnf` to have MySQL only listen on localhost for security enhancement:

```
[mysqld]
bind-address   = 127.0.0.1
```

After having installed mysql, start it, set a new root password and make it start at boot:

```
server:~ # rcmysql start
server:~ # mysqladmin -u root password 'new password'
server:~ # mysqladmin -u root -h server.example.org password 'new password'
server:~ # insserv mysql
```

Attention: logrotate will fail with MySQL after having changed the password! To fix this, create the file '/root/.my.cnf` with the following content:

```
[mysqladmin]
password = <your password>
user = root
```

Apache

```
server:~ # smart install apache2
server:~ # rcapache start
server:~ # insserv apache
```

Now that was easy ;)

For PHP, we will use another repository with newer packages:

```
server:~ # smart channel --add http://software.opensuse.org/download/server:/php/openSUSE_10.2/server:php.repo

server:~ # smart update

server:~ # smart install apache2-mod_php5 php5-gd php5-bz2 php5-gettext php5-iconv php5-imap php5-mbstring php5-mcrypt php5-mysql php5-zlib

server:~ # rcapache restart
```

Place a file with the following content in `/srv/www/htdocs/`:

```
<?php
  phpinfo();
?>
```

and call this file from your webbrowser (e.g. http://192.168.0.2/info.php).The phpinfo-page should come up and show you the configuration.

Now we will change some settings for PHP in `/etc/php5/apache2/php.ini`:

```
safe_mode = On
expose_php = Off
```

```
error_reporting = E_ALL
display_errors = Off
log_errors = On
; for XHTML-compliance
arg_seperator.output = "&"
register_globals = Off
magic_quotes_gpc = Off
include_path = ".:/usr/share/php5:/usr/share/php5/PEAR"
mysql.allow_persistent = On
```

and reload apache:

```
server:~ # rcapache2 reload
```

That's it! You now have an running Apache with MySQL and PHP.

Virtual Hosts and SSL

Now we will create some virtual hosts with Apache.

Edit `/etc/apache/listen.conf`:

```
Listen 80
<IfDefine SSL>
  <IfDefine !NOSSL>
    <IfModule mod_ssl.c>
      Listen 443
      NameVirtualHost *:443
    </IfModule>
  </IfDefine>
</IfDefine>
NameVirtualHost *:80
```

We create our self-signed certificate:

```
server:~ # cd /etc/apache2/
server:/etc/apache2 # openssl genrsa -out server.key -des3 1024
server:/etc/apache2 # openssl req -new -key server.key -out server.csr
```

Be sure to set "Common Name" to the name of your server (example.org)!

You can now use the generated server.csr for requesting a signed certificate from e.g. CACert [1] or selfsign your certificate:

```
server:/etc/apache2 # openssl req -new -x509 -days 1460 -key server.key -out server.crt
```

Same thing for "Common Name" as above...

Let's remove the passphrase from the server key, otherwise we will have to type it on every apache (re)start and move the certificate and key to the correct place:

```
server:/etc/apache2 # openssl rsa -in server.key -out server_new.key
server:/etc/apache2 # mv server_new.key ssl.key/server.key
server:/etc/apache2 # mv server.crt ssl.crt/
```

Now edit `/etc/sysconfig/apache2`:

```
APACHE_SERVER_FLAGS="SSL"
```

Let's go on with the server config and create a file in `/etc/apache2/vhosts.d/` (I called it 00-ssl.conf):

```
<IfDefine SSL>
  <IfDefine !NOSSL>
    <VirtualHost *:443>
      DocumentRoot "/srv/www/htdocs"
      ErrorLog /var/log/apache2/ssl_error_log
      TransferLog /var/log/apache2/ssl_access_log
      ServerName example.org:443
      ServerAlias server.example.org
      ServerAdmin your.email@example.org
      DirectoryIndex index.html index.shtml index.php
      AddType text/shtml .shtml
      AddOutputFilter INCLUDES .shtml
      Include /etc/apache2/conf.d/*.conf
      ScriptAlias /cgi-bin/ "/srv/www/cgi-bin/"
      <Directory "/srv/www/cgi-bin">
        AllowOverride None
        Options +ExecCGI -Includes
        Order allow,deny
        Allow from all
        SSLOptions +StdEnvVars
      </Directory>
      <Directory "/srv/www/htdocs">
        Options FollowSymLinks Includes
        AllowOverride None
        Order allow,deny
        Allow from all
      </Directory>
      <IfModule mod_userdir.c>
        UserDir public_html
        Include /etc/apache2/mod_userdir.conf
      </IfModule>
      SSLEngine on
      SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
      SSLCertificateFile /etc/apache2/ssl.crt/server.crt
      SSLCertificateKeyFile /etc/apache2/ssl.key/server.key
      <Files ~ "\.(cgi|shtml|phtml|php3?)$">
        SSLOptions +StdEnvVars
      </Files>
      SetEnvIf User-Agent ".*MSIE.*" \
               nokeepalive ssl-unclean-shutdown \
               downgrade-1.0 force-response-1.0
    </VirtualHost>
  </IfDefine>
</IfDefine>
```

and another one (called 01-default.conf):

```
<VirtualHost *:80>
  DocumentRoot "/srv/www/htdocs"
  ErrorLog /var/log/apache2/error_log
  TransferLog /var/log/apache2/access_log
  ServerName example.org:80
  ServerAlias server.example.org
  ServerAdmin your.email@example.org
  DirectoryIndex index.html index.shtml index.php
  AddType text/shtml .shtml
  AddOutputFilter INCLUDES .shtml
  Include /etc/apache2/conf.d/*.conf
  ScriptAlias /cgi-bin/ "/srv/www/cgi-bin/"
  <Directory "/srv/www/cgi-bin">
    AllowOverride None
    Options +ExecCGI -Includes
    Order allow,deny
    Allow from all
  </Directory>
  <Directory "/srv/www/htdocs">
    Options FollowSymLinks Includes
    AllowOverride None
    Order allow,deny
    Allow from all
  </Directory>
  <IfModule mod_userdir.c>
    UserDir public_html
    Include /etc/apache2/mod_userdir.conf
  </IfModule>
</VirtualHost>
```

After an apache restart (reload is not sufficient!), we have two Virtual Hosts, one with SSL, one without showing to the same location!

Back to Setup Suse 10.2 Server | proceed to Mailserver

Quellennachweise

[1] http://www.cacert.org

Setup Suse 10.2 Server/Mailserver

Mailserver

PostfixAdmin

Download and unpack PostfixAdmin [1]:

```
server:~downloads # wget http://high5.net/page7_files/postfixadmin-2.1.0.tgz
server:~downloads # tar xzf postfixadmin-2.1.0.tgz
server:~downloads # cd postfixadmin-2.1.0
```

Edit `DATABASE_MYSQL.TXT` and change the usernames and passwords:

```
#
# Postfix / MySQL
#
USE mysql;
INSERT INTO user (Host,User,Password) VALUES
('localhost','server_postfix',password('your new password'));
INSERT INTO db (Host,Db,User,Select_priv) VALUES
('localhost','server_postfix','server_postfix','Y');
# Postfix Admin user & password
INSERT INTO user (Host,User,Password) VALUES
('localhost','server_postfixad',password('another new password'));
INSERT INTO db (Host,
Db,User,Select_priv,Insert_priv,Update_priv,Delete_priv) VALUES
('localhost','server_postfix','server_postfixad','Y','Y','Y','Y');
FLUSH PRIVILEGES;
GRANT USAGE on server_postfix.* TO server_postfix@localhost;
GRANT SELECT, INSERT,DELETE,UPDATE ON server_postfix.* TO
server_postfix@localhost;
GRANT USAGE on server_postfix.* TO server_postfixad@localhost;
GRANT SELECT, INSERT,DELETE,UPDATE ON server_postfix.* TO
server_postfixad@localhost;
CREATE DATABASE server_postfix;
USE server_postfix;
```

As you can see, I changed postfix and postfixadmin to server_postfix and server_postfixad. It's just sort of personal organisation on how to do things, so feel free to leave postfix and postfixadmin ;)

Now we put this file into MySQL:

```
server:~downloads/postfixadmin-2.1.0 # mysql -u root -p < DATABASE_MYSQL.TXT
```

OK, we have that database now, next we copy all the files of postfixadmin into our webroot...

```
server: # cp -a ~/downloads/postfixadmin-2.1.0 /srv/www/htdocs/postfixadmin && cd /srv/www/htdocs/postfixadmin/
server: # mv config.inc.php.sample config.inc.php
```

...and edit the config:

```
$CONF['database_type'] = 'mysql';
$CONF['database_host'] = 'localhost';
$CONF['database_user'] = 'server_postfixad';
$CONF['database_password'] = 'another new password';
$CONF['database_name'] = 'server_postfix';
$CONF['database_prefix'] = '';

$CONF['admin_email'] = 'postmaster@example.org';

$CONF['default_aliases'] = array (
      'abuse' => 'abuse@example.org',
      'hostmaster' => 'hostmaster@example.org',
      'postmaster' => 'postmaster@example.org',
      'webmaster' => 'webmaster@example.org'
);

$CONF['domain_path'] = 'YES';
$CONF['domain_in_mailbox'] = 'YES';
$CONF['footer_text'] = 'Return to example.org';
$CONF['footer_link'] = 'http://example.org';
```

Open your webbrowser and point him to http://example.org/postfixadmin/ and complete the setup.
Do not forget to **delete** DATABASE_MYSQL.TXT and setup.php afterwards!

As you can see, there is **no** authorization required, so we will add one:

```
server: # cd /srv/www/htdocs/postfixadmin/admin/
server: # echo  "AuthType Basic" > .htaccess
server: #  echo "AuthName \"Password Required\"" >> .htaccess
server: #  echo "AuthUserFile /srv/www/htdocs/postfixadmin/admin/.htpasswd" >> .htaccess
server: #  echo "Require User root" >> .htaccess
server: # htpasswd2 -c .htpasswd root
```

Add the following line to `/etc/apache2/vhosts.d/01-default.conf`:

```
  Redirect permanent /postfixadmin https://example.org/postfixadmin/
```

This will make us always use SSL when accessing PostfixAdmin.

Add the following to `/etc/apache2/vhosts.d/00-ssl.conf` to allow the .htaccess to work:

```
  <Directory "/srv/www/htdocs/postfixadmin/">
    AllowOverride AuthConfig
  </Directory>
```

and reload apache.

Let's go on to the real postfix configuration.

Postfix

Because I have disabled IPv6, i had to change `/etc/postfix/main.cf`:

```
inet_protocols = ipv4
```

Install postfix-mysql (postfix itself should be already there):

```
server: # smart install postfix-mysql
```

Open `/etc/postfix/main.cf and add/replace the following settings:

```
myhostname = server.example.org
virtual_alias_maps = mysql:/etc/postfix/mysql_virtual_alias_maps.cf
virtual_gid_maps = static:51
virtual_mailbox_base = /home/postfix
virtual_mailbox_domains = mysql:/etc/postfix/mysql_virtual_domains_maps.cf
virtual_mailbox_maps = mysql:/etc/postfix/mysql_virtual_mailbox_maps.cf
virtual_minimum_uid = 51
virtual_transport = virtual
virtual_uid_maps = static:51
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions =
      permit_mynetworks,
      reject_unauth_destination,
      permit_sasl_authenticated
smtpd_sasl_auth_enable = yes
smtpd_sasl_local_domain = $myhostname
smtpd_sasl_security_options = noanonymous
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
inet_interfaces = $myhostname, localhost
mynetworks_style = subnet
mynetworks = 192.168.0.0/24, 127.0.0.0/8
```

Create the following files in `/etc/postfix/`:

- mysql_virtual_alias_maps.cf

```
user = server_postfix
password = your new password
hosts = localhost
dbname = server_postfix
table = alias
select_field = goto
where_field = address
```

- mysql_virtual_domains_maps.cf

```
user = server_postfix
password = your new password
hosts = localhost
dbname = server_postfix
table = domain
```

```
select_field = domain
where_field = domain
additional_conditions = and backupmx = '0' and active = '1'
```

- mysql_virtual_mailbox_maps.cf

```
user = server_postfix
password = your new password
hosts = localhost
dbname = server_postfix
table = mailbox
select_field = maildir
where_field = username
```

Create the directory structure for virtual mailboxes:

```
server: # mkdir -p /home/postfix/example.org/test
server: # chmod -R 700 /home/postfix/
server: # chown -R postfix:postfix /home/postfix/
```

Edit `/etc/aliases` and set the alias for root:

```
root: your.mail@example.org
```

After this call the following commands:

```
server: # postalias /etc/aliases
server: # rcpostfix reload
```

Without this, you will not receive any mails sent to root from services!

That should be all for postfix.

Dovecot

For installing Dovecot, we will add another repository to smart:

```
server: # smart channel --add http://software.opensuse.org/download/server:/mail/openSUSE_10.2/server:mail.repo
server: # smart update
server: # smart install dovecot
```

Edit `/etc/dovecot/dovecot-sql.conf`:

```
driver = mysql
connect = host=localhost dbname=server_postfix user=server_postfix password=your new password
default_pass_scheme = MD5-CRYPT
password_query = SELECT password FROM mailbox WHERE username = '%u'
user_query = SELECT maildir AS home, 51 AS uid, 51 AS gid FROM mailbox WHERE username = '%u'
```

Edit `/etc/dovecot/dovecot.conf`:

```
mail_location = maildir:/home/postfix/%d/%n
first_valid_uid = 51
userdb sql {
  args = /etc/dovecot/dovecot-sql.conf
}
```

Create a test-account with PostfixAdmin and try to login!

PS: One disadvantage of this config is, that real accounts do not work as email login! Only the virtual logins work and you may have two different passwords for a real user and a mail user with the same user name.

You also may want to comment out the following sections in dovecot.conf with this config:

```
  passdb pam{
  }
```

If someday I get this to work, I'll update this page...

Webmail

AS webmailer we use Roundcube [2]. For webmail, we will create a subdomain called webmail.example.org. Create the file `/etc/apache2/vhosts.d/02-webmail.example.org.conf`:

```
<VirtualHost *:80>
  ServerName webmail.example.org
  ServerAdmin your.email@example.org
  DocumentRoot /srv/www/vhosts/webmail.example.org/
  CustomLog /var/log/webmail.example.org_access_log combined
  ErrorLog /var/log/webmail.example.org_error_log
  DirectoryIndex index.php
  Include /etc/apache2/conf.d/*.conf
  <IfModule mod_php5.c>
    php_admin_flag engine on
    php_admin_value open_basedir "/srv/www/vhosts/webmail.example.org:/tmp"
  </IfModule>
  <Directory "/srv/www/vhosts/webmail.example.org">
    Order allow,deny
    Allow from all
  </Directory>
</VirtualHost>
```

But webmail.example.org does not resolve yet, so we have to add a line to `/var/lib/named/personal/example.org.db` (See DNS for more information):

```
webmail.example.org    IN A    192.168.0.2
```

Do not forget to update the Serial and restart named!

Now download Roundcube, untar to `/srv/www/vhosts/webmail.example.org` and move the contents of the extracted dir one directory down:

```
server: # wget http://kent.dl.sourceforge.net/sourceforge/roundcubemail/roundcubemail-0.1beta2.2.tar.gz
server: # tar xzf roundcubemail-0.1beta2.2.tar.gz -C /srv/www/vhosts/webmail.example.org/
server: # mv /srv/www/vhosts/webmail.example.org/roundcubemail-0.1beta2/* /srv/www/vhosts/webmail.example.org/
server: # rm -rf /srv/www/vhosts/webmail.example.org/roundcubemail-0.1beta2/
```

Create an mysql-user and a database for roundcube:

```
server: # mysql -u root -p
mysql> CREATE DATABASE server_roundcube;
mysql> CREATE USER 'server_roundcube'@'localhost' IDENTIFIED BY 'password';
```

```
mysql> GRANT USAGE,INSERT,SELECT,UPDATE,DELETE ON server_roundcube.* TO server_roundcube@localhost;
```

Open SQL/mysql5.initial.sql and add this line at the top:

```
USE server_roundcube;
```

and load this file into MySQL:

```
server: # mysql -u root -p < SQL/mysql5.initial.sql
```

Configure Roundcube now:

```
server: # cp config/db.inc.php.dist config/db.inc.php
server: # cp config/main.inc.php.dist config/main.inc.php
```

and edit those two files. `config/db.inc.php`:

```
$rcmail_config['db_dsnv'] = 'mysql://server_roundcube:password@localhost/server_roundcube';
```

`config/main.inc.php`:

```
$rcmail_config['default_host'] = 'localhost';
$rcmail_config['smtp_server'] = 'localhost';
$rcmail_config['smtp_user'] = '';
$rcmail_config['smtp_pass'] = '';
```

Make the logs and temp-directories writeable for the webserver:

```
server: # chown -R wwwrun logs
server: # chown -R wwwrun temp
```

That's it! You should now be able to login with your previously created user!

Spam- and Virusfilter

Install a few packages:

```
server: # smart install amavisd-new clamav spamassassin
```

Edit `/etc/postfix/master.cf` and add the following lines:

```
smtp-amavis unix -   -      n   -     2    smtp
  -o smtp_data_done_timeout=1200
  -o smtp_send_xforward_command=yes
  -o disable_dns_lookups=yes
  -o max_use=20
127.0.0.1:10025  inet n -     n    -    -   smtpd
  -o content_filter=
  -o smtpd_restriction_classes=
  -o smtpd_delay_reject=no
  -o smtpd_client_restrictions=permit_mynetworks,reject
  -o smtpd_helo_restrictions=
  -o smtpd_sender_restrictions=
  -o smtpd_recipient_restrictions=permit_mynetworks,reject
  -o smtp_data_restrictions=reject_unauth_pipelining
  -o mynetworks=127.0.0.0/8
```

```
  -o smtpd_error_sleep_time=0
  -o smtpd_soft_error_limit=1001
  -o smtp_hard_error_limit=1000
  -o smtpd_client_connection_count_limit=0
  -o smtpd_client_connection_rate_limit=0
  -o smtpd_milters=
  -o local_header_rewrite_clients=
  -o local_recipient_maps=
  -o relay_recipient_maps=
  -o receive_override_options=no_header_body_checks,no_unknown_recipient_checks
```

Edit `/etc/postfix/main.cf`:

```
content_filter=smtp-amavis:[127.0.0.1]:10024
```

Start/reload Amavis/Postfix and let clamav fetch new virus definitions:

```
server: # rcamavis start && insserv amavis
server: # rcpostfix reload
server: # freshclam
server: # rcclamd start && insserv clamd
server: # rcfreshclam start && insserv freshclam
```

That should have been all :) Back to Suse 10.2 Server | proceed to Mailman

Quellennachweise

[1] http://high5.net/page9.html
[2] http://www.roundcube.net/

Setup Suse 10.2 Server/Mailman

Mailman

Install Mailman via smart:

```
server: # smart install mailman
```

and check your Postfix configuration:

```
recipient_delimiter = +
unknown_local_recipient_reject_code = 550
virtual_alias_maps = mysql:/etc/postfix/mysql_virtual_alias_maps.cf hash:/var/lib/mailman/data/virtual-mailman
virtual_mailbox_maps = mysql:/etc/postfix/mysql_virtual_mailbox_maps.cf hash:/var/lib/mailman/data/virtual-mailman
alias_maps = hash:/etc/aliases hash:/var/lib/mailman/data/aliases
```

edit '/var/lib/mailman/Mailman/mm_cfg.py`:

```
MTA = 'Postfix'
POSTFIX_STYLE_VIRTUAL_DOMAINS=['example.org']
DEFAULT_EMAIL_HOST = 'example.org'
DEFAULT_URL_HOST = 'example.org'
POSTFIX_ALIAS_CMD = '/usr/sbin/postalias'
POSTFIX_MAP_CMD = '/usr/sbin/postmap'
DELIVERY_MODULE = 'SMTPDirect'
SMTPHOST = '127.0.0.1'
SMTPPORT = '25'
add_virtualhost(DEFAULT_URL_HOST,DEFAULT_EMAIL_HOST)
```

call genaliases and while we are at it and set a password:

```
server: # cd /usr/lib/mailman/ && bin/genaliases
server: # bin/mmsitepass
```

Restart Apache and Postfix now and create the initial mailing list:

```
server: # rcapache2 restart
server: # rcpostfix restart
server: # /usr/lib/mailman/bin/newlist
```

Mailing List-name has to be `mailman`, set your email adress, choose a password and give the previously chosen password at the end.

Now you can create the mailinglists you want to either via the newlist command or via web interface on http://example.org/mailman/create.

Back to Suse 10.2 Server | proceed to Security

Setup Suse 10.2 Server/FTP

VSFTPD

As we will use FTP only for anonymous usage, we will not set up any mechanisms for virtual users, authentication and so on.

```
server: # smart install vsftpd
```

Modify `/etc/vsftpd.conf`:

```
write_enable=NO
nopriv_user=ftp
ftpd_banner="Welcome to example.org FTP service."
hide_ids=YES
local_enable=NO
anonymous_enable=YES
anon_world_readable_only=YES
anon_upload_enable=NO
syslog_enable=YES
xferlog_enable=YES
vsftp_log_file=/var/log/vsftp.log
xferlog_std_format=YES
xferlog_file=/var/log/xferlog
```

The default path for anonymous ftp is `/srv/ftp/`, if you want to change this, you will have to set the home directory of the user `ftp` to whatever you want to.

Now let's start vsftpd and make it start at boot time:

```
server: # rcvsftpd start
server: # insserv vsftpd
```

That's it, you now have your own anonymous FTP.

VSFTPD and Fail2Ban

Add the following lines to your `/etc/fail2ban/jail.conf`, if you want to allow user login for ftp:

```
[vsftpd-iptables]

enabled  = true
filter   = vsftpd
action   = iptables[name=VSFTPD, port=ftp, protocol=tcp]
         mail-whois[name=VSFTPD, dest=your.mail@example.org]
logpath  = /var/log/messages
maxretry = 3
bantime  = 60
```

Back to Suse 10.2 Server | proceed to rsyncd

Setup Suse 10.2 Server/rsyncd

rsyncd

In the previous step, we set up anonymous ftp, now let's setup anonymous rsync for the same directory, so that people can mirror our ftp in an easy way!

rsyncd is normally already installed, if this is not the case, install it now:

```
server: # smart install rsync
```

Edit `/etc/rsyncd.conf`:

```
gid = users
read only = true
use chroot = true
transfer logging = true
log format = %h %o %f %l %b
log file = /var/log/rsyncd.log
hosts allow = trusted.hosts
# disable slp refresh
slp refresh = 0
# now our entry for ftp follows:
[ftp]
  comment = anon rsyncd export for our anon ftp
  path = /srv/ftp
  read only = yes
  uid = nobody
  gid = nobody
  hosts allow = 0.0.0.0/0
  transfer logging = yes
```

And, as always, start and install rsyncd:

```
server: # rcrsyncd start
server: # insserv rsyncd
```

Now everybody can sync your ftp tree via:

```
client: > rsync -a rsync://example.org/ftp/ <his local directory>
```

though I recommend this command:

```
client: > rsync -av --partial --progress --delete-after rsync://example.org/ftp/ <his local directory>
```

Back to Suse 10.2 Server

Setup Suse 10.2 Server/Cleanup

Cleanup

In my opinion, even a text only installation of suse 10.2 contains too much software, people normally do not need on a web-server, so I compiled a list of packages to uninstall for myself:

Amount of packages before cleaning up: 412 (including smart and rpm-python)

Packages to remove:

```
rpm -e a2ps acpid alsa apparmor-docs apparmor-parser apparmor-profiles
apparmor-utils autofs autoyast2 bootsplash bootsplash-theme-SuSE cairo
cpufrequtils cups cups-client cups-drivers cups-libs cyrus-sasl-saslauthd
dbus-1-mono dosfstools foomatic-filters ghostscript-library glitz gtk2
gutenprint hfsutils hplip hplip-hpijs jfsutils joe ksh libgimpprint
libzypp-zmd-backend manufacturer-PPDs Mesa mono-core mono-data mono-web
ntfsprogs pango ppp pptp preload providers python-qt qscintilla qt3 rug scpm
siga smpppd sqlite-zmd suspend tcsh usbutils vlan wireless-tools wol wvdial
wvstreams xfsprogs xorg-x11-libfontenc xorg-x11-libICE xorg-x11-libs
xorg-x11-libSM xorg-x11-libX11 xorg-x11-libXau xorg-x11-libXdmcp
xorg-x11-libXext xorg-x11-libXfixes xorg-x11-libxkbfile xorg-x11-libXmu
xorg-x11-libXp xorg-x11-libXpm xorg-x11-libXprintUtil xorg-x11-libXrender
xorg-x11-libXt xorg-x11-libXv yast2-apparmor yast2-backup yast2-bluetooth
yast2-bootfloppy yast2-irda yast2-iscsi-client yast2-nfs-client
yast2-nis-client yast2-power-management yast2-printer yast2-profile-manager
yast2-samba-client yast2-samba-server yast2-scanner yast2-sound yast2-support
yast2-tv ypbind yp-tools zmd
```

This list was made on a clean install of Suse 10.2 with all upgrades on 2007-08-19. Do not uninstall these packages, if you don't know, what you are doing!

Amount of packages after cleaning up: 314 This are 98 packages less!

Strato Smart Channellist

This is a channel list for Suse 9.3 vServer at Strato

```
[server:php]
type = rpm-md
name = PHP and extensions (SUSE_Linux_9.3)
baseurl = http://software.opensuse.org/download/server:/php/SUSE_Linux_9.3/

[suse-9.3]
type = yast2
name = Suse 9.3
baseurl = ftp://81.169.163.136/pub/linux/suse/i386/9.3

[rpm-sys]
type = rpm-sys
name = RPM System

[suse-updates]
type = yast2
name = suse-updates
baseurl = ftp://ftp.serverkompetenz.de/pub/mirror/ftp.suse.com/pub/suse/i386/9.3/

[server:messaging]
type = rpm-md
name = Server Messaging (SUSE_Linux_9.3)
baseurl = http://software.opensuse.org/download/server:/messaging/SUSE_Linux_9.3/
```

Gateway

Gateway

Use the following iptables commands for a gateway:

```
# load kernelmodule for Masquerading
/sbin/modprobe ipt_MASQUERADE
# activate forwarding
echo "1" > /proc/sys/net/ipv4/ip_forward
# allow forwarding traffic from internal network and masquerade
$IPTABLES -A FORWARD -s 192.168.0.0/24 -j ACCEPT
$IPTABLES -A POSTROUTING -t nat -s 192.168.0.0/24 -j MASQUERADE
```

Reset MS-SQL Identity

```
DBCC CHECKIDENT(tablename, RESEED, 0)
```

JS Suche in sortierter Dropdownliste

Jeder kennt bestimmt dieses Beispiel an JS-Code:

```
function selectbyValue(ID,dropdownlistname)
{
    for( var i=0; i < document.getElementById(dropdownlistname).length;i++)
    {
        if(document.getElementById(dropdownlistname).options[i].value == ID)
        {
            document.getElementById(dropdownlistname).options[i].selected = true;
            return;
        }
    }
}
```

Dieses obige Beispiel funktioniert bei jeder Dropdownbox, wird aber bei sehr vielen Einträgen seeeehr langsam.

In meinem Fall hatte ich die Möglichkeit, die Werte der Dropdownbox numerisch aufsteigend sortieren zu lassen, womit dann unterer Code funktioniert:

```
function selectByValue(ID,dropdownlistname,offsetstart,offsetend)
{
    if(ID.length > 3)
    {

        check = Math.floor(offsetstart + (offsetend-offsetstart)/2);
        if(document.getElementById(dropdownlistname).options[check].value == ID)
        {
```

```
                document.getElementById(dropdownlistname).options[check].selected = true;
            } else {
                if(document.getElementById(dropdownlistname).options[check].value > ID)
                {
                    if(offsetstart < check){
                        searchKostenstelleLog(ID,field_offset,offsetstart,check);
                    }
                } else {
                    if(offsetend > check){
                        searchKostenstelleLog(ID,field_offset,check,offsetend);
                    }
                }
            }
        }
    }
```

Dieser Code ist bei großen Mengen um einiges schneller als jedes Element einzeln zu durchlaufen.

ASP.Net/Server Variables

Server Variables in ASP can be printed out via the following code:

```
For each objItem in Request.ServerVariables
    Response.Write(objItem & "=" & Request.ServerVariables(objItem) & "<br>")
Next
```

The PHP equivalent would be

```
<?php
print_r($_SERVER);
?>
```

Migrating Strato Courier to Dovecot

Migrating Strato Courier to Dovecot

For installing Dovecot and Postfix see the Suse 10.2 tutorial for Dovecot [1]. For now, these lines are written using a clean install of Suse 10.2, informations will be added as soon as I have completed my migration.

Situation:

- Strato Image of Suse 9.3 with Plesk
- Installed and usable Courier/qmail (from plesk)
- multiple domains (I have 2 domains running here)
- installed smart [2]

Our goal:

- installed and usable dovecot with postfix and mysql with having migrated all mail boxes.

Installing needed software

This tutorial tries to cover the steps needed to convert a backup from a Strato Courier Install (from Plesk) to a fresh dovecot installation (including the installation and configuration of dovecot itself) on Suse 10.2.

The new server is a plain text install of Suse 10.2 (and cleaned up a bit) with installed smart.

smart repositories:

```
[server:mail]
type = rpm-md
name = Email services (openSUSE_10.2)
baseurl = ftp://192.168.9.1/opensuse/repositories/server:/mail/openSUSE_10.2/

[suse]
type = yast2
name = openSUSE 10.2 OSS
baseurl = ftp://download.opensuse.org/opensuse/distribution/10.2/repo/oss/

[suse-updates]
type = rpm-md
name = openSUSE 10.2 Updates
baseurl = ftp://ftp.gwdg.de/pub/linux/suse//ftp.suse.com/suse/update/10.2/
```

Installing mysql and dovecot:

```
smart install dovecot mysql cyrus-sasl-digestmd5 cyrus-sasl-saslauthd postfix-mysql
```

download postfixadmin

```
wget http://kent.dl.sourceforge.net/sourceforge/postfixadmin/postfixadmin-2.1.0.tgz
```

I assume you already configured MySQL correctly and we'll proceed with creating the database for postfix:

Take a look at DATABASE_MYSQL.TXT from postfixadmin, change the lines with passwords to your needs and load MySQL with the file:

```
mysql -u root -p < DATABASE_MYSQL.TXT
```

Configuring postfix

/etc/postfix/main.cf:

```
virtual_alias_maps = hash:/etc/postfix/virtual, mysql:/etc/postfix/mysql_virtual_alias_maps.cf
virtual_gid_maps = static:51
virtual_mailbox_base = /home/postfix
virtual_mailbox_domains = mysql:/etc/postfix/mysql_virtual_domains_maps.cf
virtual_mailbox_maps = mysql:/etc/postfix/mysql_virtual_mailbox_maps.cf
virtual_minimum_uid = 51
virtual_transport = virtual
virtual_uid_maps = static:51
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions =
      permit_mynetworks,
      reject_unauth_destination,
      permit_sasl_authenticated
smtpd_sasl_auth_enable = yes
smtpd_sasl_local_domain = $myhostname
smtpd_sasl_security_options = noanonymous
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
mynetworks_style = host
```

Create the following files in `/etc/postfix/`:

- mysql_virtual_alias_maps.cf

```
user = postfix
password = your new password
hosts = localhost
dbname = postfix
table = alias
select_field = goto
where_field = address
```

- mysql_virtual_domains_maps.cf

```
user = postfix
password = your new password
hosts = localhost
dbname = postfix
table = domain
select_field = domain
where_field = domain
additional_conditions = and backupmx = '0' and active = '1'
```

- mysql_virtual_mailbox_maps.cf

```
user = postfix
password = your new password
hosts = localhost
dbname = postfix
```

```
table = mailbox
select_field = maildir
where_field = username
```

Create the directory structure for virtual mailboxes:

```
server: # mkdir -p /home/postfix/example.org/test
server: # chmod -R 700 /home/postfix/
server: # chown -R postfix:postfix /home/postfix/
```

Edit `/etc/aliases` and set the alias for root:

```
root: your.mail@example.org
```

After this call the following commands:

```
server: # postalias /etc/aliases
server: # rcpostfix reload
```

Without this, you will not receive any mails sent to root from services!

Dovecot

Edit `/etc/dovecot/dovecot-sql.conf`:

```
driver = mysql
connect = host = localhost dbname=server_postfix user=server_postfix password=your new password
default_pass_scheme = MD5-CRYPT
password_query = SELECT password FROM mailbox WHERE username = '%u'
user_query = SELECT maildir AS home, 51 AS uid, 51 AS gid FROM mailbox WHERE username = '%u'
```

Edit `/etc/dovecot/dovecot.conf`:

```
protocols = imap imaps pop3 pop3s
listen = *
disable_plaintext_auth = no
info_log_path = /var/log/dovecot.log
mail_location = maildir:/home/postfix/%d/%u
first_valid_uid = 51
auth default {
  mechanisms = plain
  userdb sql {
    args = /etc/dovecot/dovecot-sql.conf
  }
  userdb passwd {
  }
  passdb sql {
    args = /etc/dovecot/dovecot-sql.conf
  }
}
```

After this, start dovecot.

Creating your first mailbox

fire up mysql and state the following commands:

```
USE postfix;
INSERT INTO domain VALUES
('your.domain','',0,0,0,NULL,0,NOW(),NOW(),1);
INSERT INTO mailbox VALUES
('yourname@your.domain','$1$ztrewq$t49C5MNMFHz1RAYcFtN05.','Your
Name','your.domain/yourname@your.domain',0,'your.domain',NOW(),NOW(),1);
```

The password used here is 'qwertz123' crypted with md5-crypt, your username to be used is 'yourname@your.domain'

Quit the MySQL-Client and try to connect to dovecot imap using any imap client.

Quellennachweise

[1] http://wiki.rauchs-home.de/index.php/Setup_Suse_10.2_Server/Mailserver
[2] http://wiki.rauchs-home.de/index.php/Setup_Suse_10.2_Server

Leafnode for T-Online

/etc/leafnode/config

```
expire = 30
server = news.t-online.de
server = support-forums.novell.com
server = news.gmane.org
```

Quelle(n) und Bearbeiter des/der Artikel(s)

Ascend DSL Pipes *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1501 *Bearbeiter*: Rauch

Faxen mit OpenOffice *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1506 *Bearbeiter*: Rauch

Setup Suse 10.2 Server *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1616 *Bearbeiter*: Rauch

Setup Suse 10.2 Server/Security *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1602 *Bearbeiter*: Rauch

Setup Suse 10.2 Server/DNS *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1529 *Bearbeiter*: Rauch

Setup Suse 10.2 Server/NTP *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1534 *Bearbeiter*: Rauch

Setup Suse 10.2 Server/DHCP *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1533 *Bearbeiter*: Rauch

Setup Suse 10.2 Server/LAMP *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1615 *Bearbeiter*: Rauch

Setup Suse 10.2 Server/Mailserver *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1631 *Bearbeiter*: Rauch

Setup Suse 10.2 Server/Mailman *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1583 *Bearbeiter*: Rauch

Setup Suse 10.2 Server/FTP *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1596 *Bearbeiter*: Rauch

Setup Suse 10.2 Server/rsyncd *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1591 *Bearbeiter*: Rauch

Setup Suse 10.2 Server/Cleanup *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1624 *Bearbeiter*: Rauch

Strato Smart Channellist *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1611 *Bearbeiter*: Rauch

Gateway *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1610 *Bearbeiter*: Rauch

Reset MS-SQL Identity *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1622 *Bearbeiter*: Rauch

JS Suche in sortierter Dropdownliste *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1639 *Bearbeiter*: Rauch

ASP.Net/Server Variables *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1513 *Bearbeiter*: Rauch

Migrating Strato Courier to Dovecot *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1634 *Bearbeiter*: Rauch

Leafnode for T-Online *Quelle*: http://wiki.rauchs-home.de/index.php?oldid=1614 *Bearbeiter*: Rauch

Quelle(n), Lizenz(en) und Autor(en) des Bildes

Bild:Rj_11.png *Quelle*: http://wiki.rauchs-home.de/index.php?title=Datei:Rj_11.png *Lizenz*: GNU Free Documentation License *Bearbeiter*: Benutzer:Jhartmann

Bild:Kdeprintfax.png *Quelle*: http://wiki.rauchs-home.de/index.php?title=Datei:Kdeprintfax.png *Lizenz*: unbekannt *Bearbeiter*: Rauch

Bild:Spadminfax.png *Quelle*: http://wiki.rauchs-home.de/index.php?title=Datei:Spadminfax.png *Lizenz*: unbekannt *Bearbeiter*: Rauch

Lizenz