Tipps und Tricks

In diesem Bereich des Wikis werden sporadisch Kurzanleitungen, Workarounds usw. zu verschiedenen Themenbereichen auftauchen.

VirtualBox

Da der invis Server auch in Form virtueller Maschinen nutzbar ist und wir auf Messen gelegentlich entsprechende Images verteilen, beginnt die Tipps und Tricks Ecke mit einem Verweis auf VirtualBox - unserem favorisierten Virtualisierungssystem. Da mit einigen Einträgen zum Thema VirtualBox zu rechnen sein wird, widme ich diesem Thema eine eigene Wiki-Seite.

Tipps und Tricks zu VirtualBox

Ubuntu Client mit sssd integrieren

Anbindung von Linux-Clients an einen invis-Server

1. Freigaben einbinden

Zur Anbindung eines Linux-Clients an den Server müssen zunächst die Server-Freigaben **home** und **shares** per NFSv4 ins lokale Verzeichnissystem eingehängt werden. Dazu sind in der Datei

```
/etc/fstab
```

folgende Einträge vorzunehmen:

```
invis.invis-net.loc/home /home nfs4 defaults 0 0
invis.invis-net.loc/shares /mnt/invis/shares nfs4 defaults 0 0
```

Dabei ist zu beachten, dass die Home-Verzeichnisse ggf. vorhandener Benutzer durch das Einhängen der Home-Freigabe des Server überdeckt werden. Werden weiterhin lokale Benutzer benötigt, so sollten deren Home-Verzeichnisse vorher nach

```
/local/home
```

verschoben werden.

Weiterhin muss das Zielverzeichnis zum Einhängen der Server-Freigabe „Shares“ zunächst manuell angelegt werden.

2. Benutzerverwaltung

Für die Anbindung eines Linux-Clients an einen invis-Server empfiehlt sich aktuell die Verwendung des SSS-Daemons. Dieser ist ggf. manuell nachzuinstallieren:

```
linux:~ # sudo apt-get install -y sssd
```

Danach ist unter dem Namen

```
/etc/sssd/sssd.conf
```

eine Konfigurationsdatei für den Daemon anzulegen:

```
[sssd]
config_file_version = 2
services = nss,pam
domains = default

[nss]
filter_groups = root
filter_user = root

[pam]

[domain/default]
ldap_uri = ldap://invis.invis-net.loc
ldap_search_base = dc=invis-net,dc=loc
ldap_schema = rfc2307
id_provider = ldap
ldap_user_uuid = entryuuid
ldap_group_uuid = entryuuid
ldap_id_use_start_tls = true
enumerate = true
cache_credentials = true
ldap_tls_cacertdir = /etc/ssl/certs
ldap_tls_cacert = /etc/ssl/certs/cacert.pem
chpass_provider = ldap
auth_provider = ldap
ldap_user_fullname = displayName
#cache_entry_timeout = 1
#refresh_expired_interval = 1
```

Damit **sssd** starten kann müssen die Zugriffsrechte angepasst werden:

```
linux:~ # sudo chmod 0600 /etc/sssd/sssd.conf
```

Danach wird das Stammzertifikat der Server-Zertifizierungsstelle benötigt. Die Datei liegt ebenfalls im oben genannten Verzeichnis bereit. Sie muss lokal nach

```
/etc/ssl/certs
```

kopiert werden:

```
linux:~ # sudo cp /mnt/invis/shares/service/VPN-Clients/cacert.pem
/etc/ssl/certs/
```

Jetzt kann **sssd** gestartet werden:

```
linux:~ # sudo service sssd start
```

Mit

```
linux:~ # getent passwd
```

kann überprüft werden, ob die Benutzerkonten aus dem LDAP-Verzeichnis des Servers zur Verfügung stehen.

Hat alles funktioniert, muss dafür gesorgt werden, dass **sssd** automatisch beim Systemstart startet:

```
linux:~ # sudo update-rc.d sssd defaults
```

Zarafa License-Daemon im Eigenbau

Offiziell wird openSUSE als Basis einer Zarafa-Version nicht unterstützt, was unter anderem dazu führt, dass weder Zarafa-Backup noch der License-Daemon dafür zur Verfügung stehen. Beide Tools enthalten Closed-Source-Komponenten und sind damit nicht Bestandteil der Open-Source-Pakete die wir in unserem Build-Service-Repository vorhalten.

Sie sind aber Bestandteil der Free-Edition und können somit kostenlos genutzt werden, allerdings nur auf unterstützten Plattformen, wie etwa Ubuntu LTS.

Problematisch ist grundsätzlich, dass Zarafa seine eigene Software meist statisch gegen vorhandene System-Libraries linkt. Da sich diese von Distribution zu Distribution in Ihren Versionen unterscheiden, ist es nicht möglich SLES oder RedHat-Pakete einfach unter openSUSE zu nutzen.

Möglich ist aber den License-Daemon in einer Art Sandbox laufen zu lassen, in der er alle Libraries hat, die er benötigt. Die folgende Beschreibung erläutert den Aufbau der Sandbox auf einem Ubuntu-LTS System

VM einrichten

Zunächst muss dafür ein Ubuntu-LTS System installiert werden, hier empfiehlt sich die Arbeit mit Virtualbox. Entgegen der Angaben im Zarafa-Wiki spielt die Architektur der Sandbox sehr wohl eine Rolle. Da wir unsere invis-Server immer als 64Bit Systeme installieren, nutzen wir entsprechend auch ein 64Bit Ubuntu als Basis.

Zusätzlich zur einfachen Standard-Installation wird das Software-Paket „mklibs“ benötigt:

```
heinz@ubuntu:~$ sudo apt-get install mklibs
```

Zarafa-Pakete herunterladen und entpacken

Benötigt wird die zur installierten VM passende Zarafa Free-Edition, zu finden auf dem Zarafa-Download-Server: http://download.zarafa.com/community

```
heinz@ubuntu:~$ wget
http://download.zarafa.com/community/final/7.0/7.0.8-35178/zcp-7.0.8-35178-u
buntu-10.04-x86_64-free.tar.gz
....
heinz@ubuntu:~$ tar -xzvf zcp-7.0.8-35178-ubuntu-10.04-x86_64-free.tar.gz
```

Zarafa-Pakete installieren

In diesem Schritt müssen vorbereitend einige Zarafa-Pakete installiert werden. Benötigt werden:

- zarafa-licensed
- zarafa-common
- zarafa-client
- zarafa-server

```
heinz@ubuntu:~$ cd zcp-7.0.8-35178-ubuntu-10.04-x86_64/
heinz@ubuntu:~/zcp-7.0.8-35178-ubuntu-10.04-x86_64$ sudo dpkg -i zarafa-
licensed_7.0.8-35178_amd64.deb zarafa-common_7.0.8-35178_amd64.deb zarafa-
client_7.0.8-35178_amd64.deb zarafa-server_7.0.8-35178_amd64.deb
```

Es ist nicht notwendig vorherige Versionen zu deinstallieren, das erledigt ***dpkg*** automatisch.

Sandbox aufbauen

Vor dem Bau einer neuen Sandbox sollte die Ubuntu (oder was auch immer) Version aktualisiert werden:

```
heinz@ubuntu:~$ sudo apt-get update
heinz@ubuntu:~$ sudo apt-get upgrade
```

Jetzt muss eine entsprechende Verzeichnisstruktur erzeugt werden. Eingerichtet wird diese (weitgehend FHS-konform) unter „/opt“:

```
heinz@ubuntu:~$ sudo mkdir -p /opt/zarafa-licensed-7.0.8/lib
heinz@ubuntu:~$ sudo mkdir -p /opt/zarafa-licensed-7.0.8/bin
```

Anschließend werden die benötigten Zarafa-Komponenten in diese Verzeichnisse kopiert:

```
heinz@ubuntu:~$ sudo cp -p /usr/bin/zarafa-licensed /opt/zarafa-
licensed-7.0.8/bin/
```

```
heinz@ubuntu:~$ sudo cp -p /usr/bin/zarafa-report /opt/zarafa-
licensed-7.0.8/bin/
heinz@ubuntu:~$ sudo cp -p /usr/bin/zarafa-ssm /opt/zarafa-
licensed-7.0.8/bin/
heinz@ubuntu:~$ sudo cp -p /usr/lib/libzarafaclient.so /opt/zarafa-
licensed-7.0.8/lib
```

Abschließend müssen alle von „zarafa-licensed“ benötigten System-Libraries in der Sandbox installiert werden:

```
sudo mklibs-copy -d /opt/zarafa-licensed-7.0.8/lib/ /opt/zarafa-
licensed-7.0.8/bin/*
```

Ist dies abgeschlossen, kann die Verzeichnisstruktur in ein tar.gz-Archiv gepackt und auf den Zielserver verfrachtet werden. Empfehlenswert ist es natürlich sich für openSUSE ein init-Script zum Start des Dienstes zu erzeugen und auch die Konfigurationsdatei

```
/etc/zarafa/licensed.cfg
```

auf den Zielserver zu kopieren.

Auf dem Zielserver muss noch das Verzeichnis

```
/etc/zarafa/license
```

angelegt werden. Hierin werden erworbene Lizenzschlüssel geschrieben. Ohne Lizenzschlüssel können mit laufendem License-Daemon aber immerhin 3 Outlook-Clients angebunden werden, was somit der Free-Edition entspricht.

Hinweis: *Fertige Pakete (zld4invis) für den License-Daemon auf invis-Servern stehen unter* ***http://invis.invis-server.org/index.php?page=invis-7-2*** *zum Download bereit. Sie enthalten bereits alle notwendigen Komponenten inkl. init-Script.*

Linux-Clients und NFS-Fileserver

Die Gruppen-basierte Zusammenarbeit auf einem Linux-Fileserver gestaltet sich schwierig, wenn diese per NFS auf den Fileserver zugreifen. Zwar lassen sich mit gesetztem SGID-Bit auf den Freigabe-Verzeichnissen Gruppen-Besitzrechte auf alle Objekte im Ordner vererben, nicht aber die Zugriffsrechte. Letztere sind von der „umask“ abhängig.

Auf openSUSE-Systemen ist die vorgegebene umask „022“, was bedeutet, das neu angelegte Dateien und Verzeichnisse für die besitzende Gruppe kein Schreibrecht gewähren:

```
Verzeichnis
         | u | g | o |
----------------------
Default  | 7 | 7 | 7 |
umask    | 0 | 2 | 2 |
----------------------
```

```
Ergebnis | 7 | 5 | 5 | = rwx,r-x,r-x

Datei
Default  | 6 | 6 | 6 |
umask    | 0 | 2 | 2 |
----------------------
Ergebnis | 6 | 4 | 4 | = rw-,r--,r--
```

Um pauschal auch für die besitzende Gruppe Schreibrecht zu gewähren muss die umask auf den Wert „002“ geändert werden:

```
Verzeichnis
         | u | g | o |
----------------------
Default  | 7 | 7 | 7 |
umask    | 0 | 0 | 2 |
----------------------
Ergebnis | 7 | 7 | 5 | = rwx,rwx,r-x

Datei
Default  | 6 | 6 | 6 |
umask    | 0 | 0 | 2 |
----------------------
Ergebnis | 6 | 6 | 4 | = rw-,rw-,r--
```

Die umask ist prinzipiell Benutzer-bezogen. Sie kann an mehreren Stellen im System geändert werden. Wichtig dabei ist, dass die Änderung auf dem Fileserver-Client wirksam ist. Eine Änderung auf dem Server selbst, etwa in /etc/profile o.ä. bleibt wirkungslos.

Da openSUSE mit dem „pam_umask“ Modul arbeitet, kann die umask auch in den Einstellungen der einzelnen Benutzerkonten vorgenommen werden. Eingetragen wird eine persönliche umask in das „Gecos-Feld“. Hier ein Auszug aus einer entsprechend angepassten passwd-Datei:

```
...
stefan:x:10000:100:Stefan Schäfer,umask=002:/local/home/stefan:/bin/bash
...
```

Da invis Server eine LDAP-basierte zentrale Benutzerverwaltung anbieten, **muss** die gezeigte Anpassung selbstverständlich im LDAP-Verzeichniseintrag der einzelnen Benutzer vorgenommen werden.

Melden Sie sich dazu über den Link „Verzeichnisdienst“ auf der Administrationsseite des Portals am LDAP-Verzeichnis an und ändern Sie das Feld „gecos“ betroffenen Benutzereinträge wie folgt ab:

DN: uid=**username**,ou=Users,ou=Benutzerverwaltung,dc=**invis-net**,dc=loc

Von: „System User“ zu „System User,umask=002“

Ab invis Version 6.7-R3 entspricht dies der Vorgabe, wenn Benutzer über das invis-Portal angelegt werden. Die Vorgabe kann in der Datei „/srv/www/htdocs/portal/config.php“ an die eigenen Wünsche angepasst werden.

Speicherüberlauf Cyrus Index-DB

Cyrus IMAP speichert die den Mailboxen zugehörigen Index-Datenbanken unter /var/lib/imap im Berkeley-DB (sleepycat) Format ab. Die für dieses Format übliche Konfigurationsdatei „DB_CONFIG“ fehlt im Verzeichnis /var/lib/imap/db. Das bedeutet, dass die Datenbank mit Standardwerten betrieben wird. Es kann vorkommen, das der per Default vorgegebene für die Datenbanken zur Verfügung stehende maximale Speicherplatz überschritten wird. Die Folge ist, dass Cyrus die weitere Annahme in die von ihm verwalteten Postfächer verweigert.

Dieser Fehler äußert sich in /var/log/messages mit folgenden Zeilen:

```
Feb 21 11:01:47 invis5bio lmtpunix[1802]: DBERROR db4: Logging region out of
memory; you may need to increase its size
Feb 21 11:01:47 invis5bio lmtpunix[1802]: DBERROR: opening
/var/lib/imap/deliver.db: Cannot allocate memory
Feb 21 11:01:47 invis5bio lmtpunix[1802]: DBERROR: opening
/var/lib/imap/deliver.db: cyrusdb error
Feb 21 11:01:47 invis5bio lmtpunix[1802]: FATAL: lmtpd: unable to init
duplicate delivery database
Feb 21 11:01:47 invis5bio master[17315]: service lmtpunix pid 1802 in READY
state: terminated abnormally
```

Behoben werden kann dieser Fehler durch erzeugen einer Konfigurationsdatei für die Berkeley-DB unter dem Namen „DB_CONFIG“ in /var/lib/imap/db:

```
set_cachesize 0 2097152 1

# Data Directory
#set_data_dir db

# Transaction Log settings
set_lg_regionmax 2097152
set_lg_bsize 2097152
set_lg_max 4194304
set_tx_max 200
set_tas_spins 1
#set_lg_dir logs
```

Daran anschließend ist Cyrus neuzustarten:

```
linux:~ #/etc/init.d/cyrus restart
```

Group-e -- Passwwort des Benutzers "config" zurück setzen

Das Passwort des Benutzers „config“ wird unabhängig von der generellen Benutzerverwaltung in Group-e immer in der MySQL-Datenbank verwaltet. Um es zurückzusetzen muss eine SQL-Anweisung über das MySQL-Komandozeilen Frontend abgesetzt werden:

```
linux:~ # mysql -u root -p

...

mysql> use groupe;
Database changed
mysql> UPDATE `ModulCfg` SET `CfgVal`=MD5( 'config' ) WHERE
`ModulCfg`.`FKModul`='global' AND `ModulCfg`.`FKCfgKey`='cfg/pwd' AND
`ModulCfg`.`FKObjID`=0;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> quit
Bye
linux:~ #
```

Danach kann man sich wieder als Benutzer „config“ mit dem Passwort „config“ anmelden.

OpenVPN

Damit openVPN genutzt werden kann, wird eine CRL (Certificate Revocation List) benötigt. Auf älteren Installationen wird diese Datei nicht automatisch angelegt (Seit invis Version 9.2) ist dies der Fall), auch findet sich in den easy-RSA Tools kein Script um eine solche CRL zu erzeugen. Sie muss also manuell mittels ***openssl*** erzeugt werden.

Zunächst muss die auf dem System vorhandene OpenSSL-Version ermittelt werden. Für diesen Zweck gibt es ein vorgefertigtes Script:

```
invis:/etc/openvpn/invis-server.loc # ./whichopensslcnf
/openssl-1.0.0.cnf
**********************************************************
  No /openssl-1.0.0.cnf file could be found
  Further invocations will fail
**********************************************************
```

Auch wenn die ausgegebene Meldung auf einen Fehler hindeutet, so liegt nahe, dass hier openSSL in Version 1.0.0 installiert ist und auf die in der Ausgabe genannte Konfigurationsdatei zurückgegriffen werden muss. Das Erstellen der CRL sieht dann wie folgt aus:

```
invis:/etc/openvpn/invis-server.loc # openssl ca -config ./openssl-1.0.0.cnf
-gencrl -keyfile ./keys/ca.key -cert ./keys/ca.crt -out ./keys/crl.pem
```

Danach ist im Unterverzeichnis „keys“ die Datei „crl.pem“ zu finden, auf die in der OpenVPN Konfiguration bezug genommen wird.

Nein Danke Zensursula

Achtung, der folgende Text ist nicht ganz neutral. Er enthält persönliche Meinugsäußerungen. Ich bitte dies zu entschuldigen.

Ab Oktober 2009 soll das Web-Sperren-Gesetz gegen die Verbreitung von Kinderpornographie in Kraft treten. Da dieses Gesetzt, trotz des sicherlich aufrichtigen Ansatzes, am Thema vorbei geht und allenfalls eine Zensurinfrastruktur im Internet etabliert, zeigen wir hier eine Möglichkeit auf, es zu umgehen. Möglicherweise hilft das ja dabei die Wirkungslosigkeit dieses Gesetzes zu verdeutlichen.

Wer einen invis-Server im eigenen Netz betreibt und sich (zumindest vorerst) sicher sein möchte, dass seine Nameserverabfragen nicht von „manipulierten“ Nameservern beantwortet werden, kann dies durch Ändern der „forwarders“ in der bind-Konfiguration umgehen.

Ändern Sie einfach in der Datei „/etc/named.conf“ die Einträge hinter „forwarders“:

```
        # The forwarders record contains a list of servers to which queries
        # should be forwarded.  Enable this line and modify the IP address
to
        # your provider's name server.  Up to three servers may be listed.
        # Die folgende Zeile ist um die IP-Adresse des fuer Sie zustaendigen
DNS
        # zu erweitern.

        forwarders { 194.25.2.129; 192.168.178.1; };

        # Enable the next entry to prefer usage of the name server declared
in
        # the forwarders section.

        forward first;
```

Tragen Sie statt des T-Online Nameservers und etwa dem einer Fritzbox (wie hier gezeigt) bis zu drei Nameserver aus der unter http://www.ungefiltert-surfen.de zu findenden Liste ein und starten Sie ***bind*** mit:

```
Kommandozeile: /etc/init.d/named restart
```

neu.

Deeplinks verhindern

Wenn der Webserver eines invis Servers auch via HTTPs aus dem Internet erreichbar ist, können die einzelnen Applikationen derzeit noch durch direkte Eingabe der Zieladresse im Browser ohne Umweg über das invis Portal erreicht werden.

Dieses Verhalten ist aus Sicherheitsgründen eher bedenklich und in der Regel nicht erwünscht. Um dies zu verhindern muss die Apache-Konfigurationsdatei /etc/apache2/vhosts.d/i7ssl.conf erweitert

werden.

Tragen Sie dort für jede Applikation die nicht direkt angesprochen werden soll folgenden Eintrag ein:

```
    # Deeplinks verhindern
    <Directory /srv/www/htdocs/phpMyAdmin>
        SetEnvIfNoCase Referer "^http://invis.invis-net.loc" dontblock
        SetEnvIfNoCase Referer "^https://your.dyndns-domain.net" dontblock
        Order Deny,Allow
        Deny from all
        Allow from env=dontblock
    </Directory>
```

Das Beispiel zeigt den Deeplink-Schutz für das Verzeichnis von phpMyAdmin. Sie müssen lediglich die Domain-Namen an Ihre Gegebenheiten anpassen und den Apache neustarten. Danach sind die geschützten Applikationen nur noch über das invis-Portal erreichbar.

Die gezeigten Einträge sind ab invis Release 6.6-R4 generell für alle Applikationen Standard.

Faxgate-Client unter Linux nutzen

Um den Faxgate-Client unter Linux nutzen zu können sind ein paar Kleinigkeiten zu beachen:

1. Die Firewall des Linux-Clients muss Port 50000/TCP geöffnet haben.
2. Die „.jar“ Datei ist mittels der Befehlszeile „java -jar FaxgateClient.jar“ zu starten. Das lässt sich ja unter KDE oder Gnome als fertiges Icon auf den Desktop legen.
3. In der hosts-Datei von Client und Server sollten trotz funktionierendem DNS in Einträgen mit der IP „127.0.0.2“ stattdessen die realen IP-Adressen verwendet werden.
4. Da unter Linux nicht unter fremder Benutzerkennung gedruckt werden kann, muss für jeden Fax-Benutzer Serverseitig eine Fax-Konfiguration via YaST erzeugt werden. Wird dort die MSN für eingehende Faxe nicht eingetragen, kann dieser Eintrag nur für den Faxversand genutzt werden. Der Faxempfang kann so weiterhin über die Benutzerkennung „fax“ erfolgen. D.h. alle Faxe landen im speziellen Faxpostfach und sind für alle Nutzer lesbar.