invis Server AD 11.x auf 12.x bzw. 13.x aktualisieren

Hinweis: *Die nachfolgenden Beschreibungen schließen die nie offiziell veröffentlichte invis-Server Version 10.5 mit ein.*

Bei diesem Upgrade gehen wir davon aus, dass Ihr Server unter openSUSE Leap 42.1 betrieben wird. Diese Version wird Stand Januar 2018 nicht mehr mit Updates versorgt und inzwischen wurden auch längst die Repositories entfernt.

Ende Januar 2018 läuft darüber hinaus auch openSUSE Leap 42.2 aus. Deren Repositories dürften aber noch eine Weile bestehen bleiben.

Auch wenn in den nachfolgenden Beschreibungen auf invis-Server Versionen 12.x und 13.x eingegangen wird, sollten Sie natürlich nicht auf bereits veraltete Versionen aktualisieren.

Letzte reguläre Updates installieren

Wir Ihre Ausgangsdistribution noch mit Updates versorgt, sollten Sie vor dem Distribution-Upgrade letztmalig reguläre Updates einspielen.

```
invis:~ # zypper ref
...
invis:~ # zypper up
```

In Einzelfällen ist es bei den von uns betreuten Servern dabei zu Problemen mit dem Bootmanager **grub** gekommen. Die installierten Updates machen auf jeden Fall einen Neustart des Servers erforderlich. Mit zerschossenem **grub** schlägt der Neustart natürlich fehl. Ist das Kind schon in den Brunnen gefallen, hilft die SuperGrubDisk weiter, mit der sich der Server leicht wieder starten lassen sollte.

Mit noch laufendem, oder mit der SuperGrubDisk gestartetem Server sollten Sie **grub** einfach mit YaST neu installieren lassen. Ggf. genügt es auch schon **grub** mit folgenden Kommandos wieder zum Leben zu erwecken:

```
invis:~ # grub2-install /dev/sda
invis:~ # grub2-install /dev/sdb
...
```

Samba Pakete aktualisieren

Dieser Schritt kann, je nach Ausgangssituation vor oder nach dem Distributions-Upgrade erfolgen. Wir halten für Leap 42.1 zumindest noch Samba 4.5.x Pakete vor.

Dem Umstieg von den Sernet-Samba Paketen auf unsere eigenen haben wir einen eigenen Abschnitt

gewidmet: Sernet Samba Pakete ersetzen

Distribution Upgrade Installieren

Jetzt müssen Sie Ihre Software Repositories für das Distribution-Upgrade vorbereiten. Aus Gewohnheit, sichere ich die Repo-Dateien zuvor:

```
invis:~ # cp -R /etc/zypp/repos.d /etc/zypp/repos.d.bak
```

Prüfen wir jetzt, ob ein CD/DVD Repository bei der Installation verwendet wurde. Ist das der Fall, kann es gelöscht werden:

```
invis:~ # grep "cd://" /etc/zypp/repos.d/*
/etc/zypp/repos.d/openSUSE-42.1-0.repo:baseurl=cd:///?devices=/dev/disk/by-id/ata-TSSTcorp_CDDVDW_SH-224BB_R8WS68BCB00TYX
invis:~ # rm /etc/zypp/repos.d/openSUSE-42.1-0.repo
```

Ersetzen wir jetzt in den Repository-Dateien die openSUSE Versionsnummer

```
invis:~ # sed -i 's/42\.1/42\.2/g' /etc/zypp/repos.d/*
```

Im nächsten Schritt muss noch eine Anpassung, an die Bennenung der Repositories vorgenommen werden:

```
invis:~ # sed -i 's/openSUSE_42\.2/openSUSE_Leap_42\.2/g' /etc/zypp/repos.d/*
```

Jetzt kann mit ***zypper ref*** getestet werden, ob alle Repositories funktionieren.

Sollte Ihre ursprüngliche Installation auf einem „unstable“ Repository basieren, wäre jetzt ein guter Zeitpunkt auf die Stable-Repositories zu wechseln. Überprüfen Sie es wie folgt:

```
invis:~ #  grep "_unstable" /etc/zypp/repos.d/*
/etc/zypp/repos.d/spins_invis_unstable.repo:[spins_invis_unstable]
```

Wie folgt, wechseln sie auf unsere Stable-Repositories:

```
invis:~ # sed -i 's/_unstable/_stable/g' /etc/zypp/repos.d/*
invis:~ # sed -i 's/\/unstable/\/stable/g' /etc/zypp/repos.d/*
```

Führen Sie jetzt sicherheitshalber noch ein Backup Ihrer Zarafa-Datenbank durch:

```
invis:~ # zdbdump
```

Damit sind alle Vorbereitungen abgeschlossen und wir kommen zum eigentlichen Distribution-Upgrade:

```
invis:~ # zypper ref
invis:~ # zypper dup
```

Hinweis: *Im Internet findet sich immer wieder der eigentlich vernünftige Hinweis, das Distribution-Upgrade mit der Option „no-allow-vendor-change“ zu starten. In unserem Fall, gerade beim Wechsel von „unstable“ zu „stable“ macht das aber keinen Sinn.*

Sind alle Updates installiert, folgt mit dem Reboot der spannende Moment. Bei unseren Tests hat dies nahezu problemlos funktioniert. Ein paar Dinge müssen aber nachgearbeitet werden.

Sehr wahrscheinlich können Sie auf diesem Wege einzelne openSUSE Leap Minor Releases überspringen, also beispielsweise direkt von 42.1 auf 42.3 aktualisieren. Ich persönlich habe das noch nicht gemacht, sondern ggf. zwei Distributions-Upgrades nacheinander durchgeführt.

In diesem Fall können Sie direkt nach dem ersten Upgrade das zweite vornehmen. Hier genügt eine einfache Anpassung der Repositories:

```
invis:~ # sed -i 's/42\.2/42\.3/g' /etc/zypp/repos.d/*
```

...und wieder das Upgrade:

```
invis:~ # zypper ref
invis:~ # zypper dup
```

Nacharbeit

Auf den ersten Blick sind bei unseren Upgrades folgende Probleme aufgefallen:

1. Das Theme des Bootmanagers Grub wurde auf openSUSEs Standard zurück gesetzt.
2. Der DHCP-Server startete nicht.
3. Der Webserver startete nicht.
4. Die CA Stammzertifikate des Systems werden nicht korrekt gepflegt

Punkt 1 ist reine Kosmetik, unschön, sind die drei folgenden Punkte.

Apache Webserver wieder in Betrieb nehmen

Hier ist die Lösung ist einfach. Ursache ist ein Upgrade der USV-Überwachungssoftware „apcupsd“. Beim Upgrade wurde eine mit Apache 2.4 inkompatible und insgesamt vollkommen unsinnige Apache Webserver-Konfiguration installiert. Um Sie unwirksam zu machen und künftigen Problemen vorzubeugen löschen Sie einfach alle aktiven Inhalte aus der Datei

```
/etc/apache2/conf.d/apcupsd
```

aber löschen Sie nicht die Datei im ganzen. Lassen Sie sie einfach mit einem Kommentar versehen an Ort und Stelle. Sie verhindern so, dass künftige Updates dieses Problem erneut verursachen.

Starten Sie jetzt den Webserver neu:

```
invis:~ # systemctl restart apache2.service
```

DHCP Server wieder in Betrieb nehmen

Hier ist das Problem ein klein bisschen ernster. Grundsätzlich wird das bestehende Problem beim Startversuch des ISC-DHCP Servers normalerweise sehr gut in der Datei

```
/var/log/rc.dhcpd.log
```

dokumentiert. Hier der entsprechende Auszug:

```
...
Error: Cannot start TLS session to 127.0.0.1:389: Can't contact LDAP server
Configuration file errors encountered -- exiting
...
```

In diesem Fall ist die Meldung leider zweideutig. Sie bedeutet entweder, dass der in Samba integrierte LDAP-Server nicht über eine TSL-gesicherte Verbindung oder schlicht gar nicht angesprochen werden kann.

Kontrollieren Sie im einfachsten Fall, ob Samba läuft. Im für dieses Beispiel zugrundeliegenden Upgrade, war Samba (obwohl es beim booten keine entsprechenden Hinweis gab) inaktiv.

```
invis:~ # systemctl status samba.service
* samba.service - Samba AD Daemon
   Loaded: loaded (/usr/lib/systemd/system/samba.service; enabled; vendor
preset: disabled)
   Active: inactive (dead) since Tue 2017-07-25 17:56:43 CEST; 18min ago
  Process: 1839 ExecStart=/usr/sbin/samba $SAMBAOPTIONS (code=exited,
status=0/SUCCESS)
 Main PID: 1839 (code=exited, status=0/SUCCESS)

Jul 25 17:56:40 invisad systemd[1]: Starting Samba AD Daemon...
Jul 25 17:56:42 invisad samba[1839]: [2017/07/25 17:56:42.680737,  0]
../source4/smbd/server.c:372(binary_smbd_main)
Jul 25 17:56:42 invisad samba[2546]: [2017/07/25 17:56:42.966732,  0]
../source4/smbd/server.c:115(sig_term)
Jul 25 17:56:42 invisad samba[2546]:   SIGTERM: killing children
Jul 25 17:56:42 invisad samba[2546]: [2017/07/25 17:56:42.966835,  0]
../source4/smbd/server.c:120(sig_term)
Jul 25 17:56:42 invisad samba[2546]:   Exiting pid 2546 on SIGTERM
```

Nachdem Samba gestartet wurde, ließ sich auch der DHCP-Server starten. Nach dem zweiten Reboot, lief alles ohne Probleme.

```
invis:~ # systemctl start samba.service
invis:~ # systemctl start dhcpd.service
```

CA Stammzertifikate erneuern

Woher dieses Problem stammt, konnten wir noch nicht klären. Auch nicht, wie es auf korrektem Weg wieder in Ordnung gebracht wird. Die nachfolgende Beschreibung ist lediglich ein Workaround.

Das Problem fällt nicht auf den ersten Blick auf. Im Verzeichnis

```
/etc/ssl
```

gibt es eine Verknüpfung namens „certs“ die auf das Verzeichnis

```
/var/lib/ca-certificates/pem
```

zeigt. Das ist soweit natürlich korrekt. Ein Blick in dieses Verzeichnis zeigt, dass darin auch einige Stammzertifikate zu finden sind. Üblicherweise liegen dort aber auch eine Reihe kryptisch benannter Dateisystemverknüpfungen die auf die einzelnen Zertifikatsdateien zeigen. Wenn Sie keine dieser Verknüpfungen sehen, haben Sie das merkwürdige Problem. Es wirkt sich dadurch aus, dass einige Zertifikatsüberprüfungen fehlschlagen. Das mag kein unmittelbares Problem sein, ich verspreche Ihnen aber, dass es Ihnen im unpassendsten Moment auf die Füße fällt.

Normalerweise lässt sich alles wieder mit folgendem Kommando in Ordnung bringen:

```
invis:~ # update-ca-certificates -f -v
```

Genau das klappt leider nicht. Um das Problem bis zur endgültigen Lösung zu umgehen legen Sie zunächst ein neues Verzeichnis an:

```
invis:~ # mkdir /var/lib/ca-certificates/pem2
```

Löschen Sie jetzt den vorhandenen Link auf das ursprüngliche Verzeichnis und legen Sie statt dessen einen neuen an, der auf das neue leere Verzeichnis zeigt:

```
invis:~ # rm /etc/ssl/certs
invis:~ # ln -s /var/lib/ca-certificates/pem2/ /etc/ssl/certs
```

Führen Sie jetzt den Befehl zur Pflege der Stammzertifikate aus:

```
invis:~ # update-ca-certificates -f -v
```

Dabei wird zwar gewarnt, dass das Ziel des Links nicht korrekt ist, hinterher funktioniert aber erst einmal alles. Eine endgültige Klärung des Phänomens steht noch aus.

Grub Theme

Stellen wir der Form halber auch unser Grub-Theme wieder her. Kann ja nicht schaden auch so etwas mal gemacht zu haben.

Das invis-Theme liegt in Form eines tar.gz-Archivs im Dokumentationsverzeichnis des invisAD-setup Paketes. Von dort können Sie es direkt an Ort und Stelle entpacken:

```
invis:~ # tar -xzf /usr/share/doc/packages/invisAD-
setup/examples/grub/invis8.tar.gz -C /boot/grub2/themes/
```

Während des Upgrades wurde eine Kopie der alten Grub-Konfigurationsdatei erstellt. Stellen Sie diese wieder her. Wenn Sie möchten können Sie ja die neue von openSUSE generierte Datei ebenfalls sichern.

```
invis:~ # cp /etc/default/grub /etc/default/grub.suse
invis:~ # cp /etc/default/grub.old /etc/default/grub
```

Ändern Sie jetzt in Datei

```
/etc/default/grub
```

die Versionsnummer Ihres invis-Servers auf die Nummer ab, auf die aktualisieren möchten:

```
...
GRUB_DISTRIBUTOR="invisAD Server 13.1"
...
```

Jetzt muss die Änderung noch in eine aktive Grub2-Konfiguration umgesetzt werden:

```
invis:~ # grub2-mkconfig -o /boot/grub2/grub.cfg
```

Damit sollten alle „Problemchen“ des Upgrades beseitigt sein.

Jetzt geht es darum von Zarafa auf Kopano zu migrieren und Ihren invis-Server von Version 11.x auf 12.x anzuheben.

Von Zarafa zu Kopano

Dieser Schritt wartet mit etwas mehr Arbeit auf… Beginnen wir damit, die notwendigen Repositories einzurichten. Wir gehen nachfolgend davon aus, dass Sie für Zarafa einen bestehenden Subskriptionsvertrag haben und wir somit die für Vertragskunden zur Verfügung stehenden Kopano-Repositories umsteigen können.

Vorbereitung

Für diese Repositories benötigen Sie einen Account für das Kopano-Portal. Der Account an sich ist kostenfrei. Registrieren Sie sich unter **https://portal.kopano.com/user/register**

Sie erhalten nach kurzer Zeit eine Bestätigungsmail mitsamt dem Zugangspasswort für das Portal. Nachdem Sie sich angemeldet haben können Sie im Portal über den Menüpunkt „Subscriptions“ Ihre Lizenznummer nutzen um den Zugang zu den Kopano-Repositories freizuschalten.

Ihre Lizenznummer finden Sie in der Datei:

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

Im Portal können Sie, wenn Sie möchten auch ein eigenes Passwort setzen. Benutzername und Passwort des Portals entsprechen den Zugangsdaten für die Repositories. Es werden in der Basis zwei Repositories benötigt.

Kopano-Core

```
[Kopano-openSUSE-42.2_limited]
name=Kopano-openSUSE-42.2_limited
enabled=1
autorefresh=1
baseurl=https://download.kopano.io/limited/core:/final/openSUSE_Leap_42.2
path=/
type=rpm-md
keeppackages=0
```

Kopano WebApp

```
[Kopano-Webapp]
name=Kopano-Webapp
enabled=1
autorefresh=1
baseurl=https://download.kopano.io/limited/webapp:/final/openSUSE_Leap_42.2/
path=/
type=rpm-md
keeppackages=0
```

Legen Sie beide Repositories als Dateien in

```
/etc/zypp/repos.d
```

an. Die Dateien sollten einen wiedererkennbaren Namen tragen und müssen auf „.repo“ enden. Führen Sie jetzt

```
invis:~ # zypper ref
```

aus. Zypper wird Sie dabei für jedes der beiden Repositories nach den Zugangsdaten fragen.

Paket Upgrade

Hinweis: *Die nachfolgende Beschreibung fußt auf der Original-Kopano Migrationsanleitung:* ***https://documentation.kopano.io/kopano_migration_manual/zcp_migration.html#zcp-to-kopano-migration-script***

Zunächst sollte die bestehende Zarafa Konfiguration gesichert werden:

```
invis:~ # cp -R /etc/zarafa /etc/zarafa.bak
```

Danach folgt mit der Deinstallation der Zarafa-Pakete der „Point of no return“. Sie können sich wie folgt zunächst eine Liste der Pakete ausgeben lassen:

```
invis:~ # rpm -qa | egrep '(mapi|gsoap|libical|vmime|zarafa)' | awk '{ print $1 }'
```

Es kann nicht schaden sich die Liste zunächst anzeigen zu lassen. Geübten Augen fällt dabei eventuell das eine oder andere Paket auf, welches nicht deinstalliert werden sollte. Bei unseren Upgrades war das jedoch nicht der Fall.

Auf die gleiche Weise lassen sich die gefundenen Pakete auch deinstallieren:

```
invis:~ # rpm -qa | egrep '(mapi|gsoap|libical|vmime|zarafa)' | awk '{ print $1 }' | xargs zypper rm
```

Deinstallieren Sie zusätzlich noch das Paket „z-push2“.

```
invis:~ # zypper rm z-push2
```

Jetzt können die Kopano-Pakete installiert werden. Hier eine bereinigte Liste für den invis-Server:

```
kopano-server-packages
kopano-bash-completion
z-push-kopano
z-push-autodiscover
z-push-kopano-gabsync
z-push-ipc-memcached
z-push-ipc-sharedmemory
z-push-config-apache
z-push-config-apache-autodiscover
kopano-webapp
kopano-webapp-lang
kopano-webapp-plugin-contactfax
kopano-webapp-plugin-desktopnotifications
kopano-webapp-plugin-filepreviewer
kopano-webapp-plugin-folderwidgets
kopano-webapp-plugin-gmaps
kopano-webapp-plugin-intranet
kopano-webapp-plugin-pimfolder
kopano-webapp-plugin-quickitems
kopano-webapp-plugin-spell
kopano-webapp-plugin-spell-de-de
kopano-webapp-plugin-titlecounter
kopano-webapp-plugin-webappmanual
```

Übernehmen Sie diese Liste in eine Textdatei aus Ihrem Server und installieren Sie die Pakete dann wie folgt:

```
invis:~ # zypper in `cat paketliste`
```

Konfiguration anpassen

Komplizierter wird es jetzt. Jetzt muss die Kopano Konfiguration an unsere Umgebung angepasst werden. Dies kann ausgehend von der noch jungfräulichen Kopano-Installation ausgehen oder von der gesicherten Zarafa-Konfiguration. Empfohlen, weil sauberer wird der erste Weg. Leider ist er auf der etwas aufwendigere.

Etwas einfacher können wir es uns machen, in dem wir unsere Vorlage-Konfigurationen aus Github nutzen. Dazu muss auf Ihrem Server, wenn nicht bereits geschehen „git“ installiert werden.

```
invis:~ # zypper in git
```

Danach können Sie unser Github Repository auf Ihren Server klonen:

```
invis:~ # git clone https://github.com/invisserver/invisAD-setup.git
```

Wechseln Sie in das neu entstandene Verzeichnis „invisAD-setup“ und checken sie die letzte Stable-Version aus. Derzeit ist dies Version 12.2.

```
invis:~/invisAD-setup # git checkout Stable_12.2
```

Sichern Sie jetzt die derzeitige Kopano-Konfiguration und kopieren Sie jetzt die neuen Konfigurationen ins Kopano-Verzeichnis:

invisAD Versionen 12.x

```
invis:~/invisAD-setup # cp -R /etc/kopano /etc/kopano.bak
invis:~/invisAD-setup # cp usr/share/doc/packages/invisAD-
setup/examples/kopano/*.cfg /etc/kopano/
invis:~/invisAD-setup # cp usr/share/doc/packages/invisAD-
setup/examples/kopano/config.php /etc/kopano/webapp/
invis:~/invisAD-setup # cp usr/share/doc/packages/invisAD-
setup/examples/kopano/kopano /etc/sysconfig/
```

invisAD Versionen 13.x

```
invis:~/invisAD-setup # cp -R /etc/kopano /etc/kopano.bak
invis:~/invisAD-setup # cp usr/share/sine/templates/groupware/kopano/*.cfg
/etc/kopano/
invis:~/invisAD-setup # cp
usr/share/sine/templates/groupware/kopano/config.php /etc/kopano/webapp/
invis:~/invisAD-setup # cp usr/share/sine/templates/groupware/kopano/kopano
/etc/sysconfig/
```

Jetzt müssen Sie ein paar Anpassungen vornehmen. Beginnen wir mit der LDAP-Konfiguration.

ldap.cfg

```
...
ldap_bind_user = ldap.admin@ihredomain.tld
```

```
...
ldap_bind_passwd = ihrpwd
...
ldap_search_base = DC=ihredomain,DC=tld
...
```

Übernehmen Sie die Werte aus der gleichnamigen Datei Ihrer gesicherten Zarafa Konfiguration. Die beiden Dateien unterscheiden sich noch an weiteren Stellen. Ändern Sie aber bitte nur die drei oben aufgeführten Werte.

server.cfg

```
...
system_email_address   = administrator@ihredomain.tld
...
mysql_user             = zarafa
...
mysql_password         = ihrdbpwd
...
mysql_database         = zarafa
...
server_ssl_key_file    = /etc/invis/private/zarafa.pem
...
cache_cell_size                              = Wert aus alter
Konfiguration
....
```

Auch hier gilt, dass alle anderen Werte, die sich in Bezug auf Ihre alte Konfiguration geändert haben so bleiben, wie Sie in unserer Kopano-Vorlage stehen. Bei den Datenbank-Zugangsdaten mag es sonderbar aussehen, dass hier jetzt „zarafa“ steht, aber es ist ja auch Ihre alte Datenbank.

Sie müssen noch den Pfad zu den Attachements und die Besitzrechte daran ändern:

```
invis:~ # mv /srv/zarafa/ /srv/kopano/
invis:~ # chown -R kopano.kopano /srv/kopano
```

Sie können jetzt den Kopano-Server Daemon starten und testen, ob er läuft:

```
invis:~ # systemctl start kopano-server.service
invis:~ # kopano-admin -l
```

Wenn der letzte Befehl Ihnen eine korrekte Liste Ihrer Kopano-Benutzer zeigt, sieht es schon mal ziemlich gut aus.

Damit sind schon alle Konfigurationsanpassungen abgeschlossen und Sie können alle Kopano-Daemons starten, prüfen ob Sie laufen und für den automatischen Start vorsehen:

```
invis:~ # systemctl start kopano-daemon.service
invis:~ # systemctl status kopano-daemon.service
```

```
invis:~ # systemctl enable kopano-daemon.service
```

Die Daemons sind:

- kopano-server
- kopano-search
- kopano-gateway
- kopano-spooler
- kopano-ical
- kopano-precense
- kopano-dagent

Kopano WebApp ins invis-Portal integrieren

Für diesen Zweck müssen Sie sich zunächst eine LDIF-Datei anlegen und diese dann ins ActiveDirectory importieren:

```
dn: cn=KopanoApp,cn=invis-Portal,cn=Informationen,cn=invis-server,dc=invis-
net,dc=loc
objectClass: top
objectClass: iPortEntry
iPortEntryName: KopanoApp
cn: KopanoApp
iPortEntrySSL: FALSE
iPortEntryURL: [servername]/webapp
iPortEntryDescription: Die moderne Webapp der Groupware "Kopano" bietet
Zugriff auf E-Mails, Terminkalender, Augabenverwaltung und Notizen in
frischem "Look & Feel".
iPortEntryActive: FALSE
iPortEntryPosition: Lokal
iPortEntryButton: Groupware
iPortEntryPriv: user
```

Passen Sie die Datei an Ihre Umgebung an. Dies betrifft eigentlich nur den DN. Hier ist die Domäne anzupassen. Importieren Sie die Datei jetzt wie folgt:

```
invis:~ # invisad:~ # ldbadd -v -H /var/lib/samba/private/sam.ldb
webapp.ldif
Added cn=KopanoApp,cn=invis-Portal,cn=Informationen,cn=invis-server,dc=bae-
net,dc=loc
Added 1 records successfully
```

Jetzt muss noch die Schaltfläche für die alte Zarafa Webapp deaktiviert und die neue Kopano-Schaltfläche aktiviert werden:

```
invis:~ # swpestat ZarafaApp FALSE
modifying entry "cn=ZarafaApp,cn=invis-Portal,cn=Informationen,cn=invis-
server,DC=bae-net,DC=loc"

Der Eintrag "ZarafaApp" ist jetzt auf "FALSE" gesetzt
```

```
invis:~ # swpestat KopanoApp TRUE
modifying entry "cn=KopanoApp,cn=invis-Portal,cn=Informationen,cn=invis-
server,DC=bae-net,DC=loc"

Der Eintrag "KopanoApp" ist jetzt auf "TRUE" gesetzt
```

Testen Sie es jetzt über Ihr invis-Portal aus.

ownCloud Upgrade

Nach dem Server Upgrade ist ownCloud noch auf Stand von Version 8.1 aus den openSUSE Repositories. Sie können dies bis auf Version 9.1 aktualisieren. Gehen Sie dabei nach der **Anleitung** hier im Wiki vor.

Lassen Sie sich nicht davon stören, dass die Repositories der Versionen 8.2 und 9.0 nicht für openSUSE Leap 42.2 zur Verfügung stehen. Die Versionen für Leap 42.1 funktionieren genauso.

VirtualBox Upgrade

Hinweis: *Mit Veröffentlichung von invis-Server Version 13 sind wir auf die VirtualBox-Pakete aus der openSUSE Leap Distribution umgestiegen. Ursache dafür waren Probleme mit den Original Paketen von Oracle unter openSUSE Leap 42.2 und neuer. Die in openSUSE Leap enthaltenen Pakete sind allerdings weniger aktuell als die Oracle eigenen Pakete, trotzdem empfehlen wir auf diese Pakete umzusteigen. Wer dennoch mit den Oracle Paketen experimentieren möchte folgt der nachfolgenden Beschreibung.*

Aktualisierung der Oracle VirtulaBox Pakete

VirtualBox steht inzwischen in Version 5.2 zur Verfügung, installiert ist auf invis-Servern 11.x Version 5.0.

Vorbereitend sollten Sie überprüfen, ob sich auf Ihrem invis-Server noch alte Kernelquellen Verzeichnisse befinden. Wenn ja, entfernen Sie sie:

```
invis:~ # ls -l /usr/src/
insgesamt 24
lrwxrwxrwx  1 root root   18 25. Jul 17:31 linux -> linux-4.4.74-18.20
drwxr-xr-x 23 root root 4096 25. Jul 11:47 linux-4.1.39-56
drwxr-xr-x  3 root root 4096 25. Jul 11:47 linux-4.1.39-56-obj
drwxr-xr-x 24 root root 4096 25. Jul 17:35 linux-4.4.74-18.20
drwxr-xr-x  3 root root 4096 25. Jul 17:31 linux-4.4.74-18.20-obj
...
```

Im Beispiel hier sind noch Verzeichnisse des Kernels 4.1 vorhanden.

```
invis:~ # rm -rf /usr/src/linux-4.1.39-56*
```

Da nach dem bisherigen Upgrade VirtualBox in Ermangelung an frisch kompilierten Kernel-Modulen ohnehin nicht aktiv ist, sollte das Upgrade einfach sein. Entfernen Sie einfach VirtualBox 5.0:

```
invis:~ # zypper rm VirtualBox-5.0
```

und installieren Sie unmittelbar danach die aktuelle Version:

```
invis:~ # zypper in VirtualBox-5.1
```

Alles Weitere wie das Kompilieren neuer Kernel-Module läuft beinahe automatisch. Das Webfrontend **phpVirtualBox** wurde bereits im Zuge des Distribution-Upgrades auf die notwendige Version angehoben.

Aktualisieren Sie abschließend noch den VirtualBox Extension-Pack, so Sie ihn installiert haben. Bedenken Sie hierbei, dass Sie ihn kostenfrei nur für private oder Evaluationszwecke nutzen dürfen. Die gewerbliche Nutzung ist lizenzpflichtig.

```
invis:~ # wget
http://download.virtualbox.org/virtualbox/5.1.24/Oracle_VM_VirtualBox_Extens
ion_Pack-5.1.24-117012.vbox-extpack
```

Hinweis: *Den korrekten Link finden Sie unter https://www.virtualbox.org/wiki/Downloads*

Installieren Sie den Extension-Pack wie folgt:

```
invis:~ # VBoxManage extpack install
Oracle_VM_VirtualBox_Extension_Pack-5.1.24-117012.vbox-extpack
```

Damit ist VirtualBox auf Stand.

Umstieg auf die openSUSE Leap Pakete von VirtualBox

Anders als beim Oracle Paket ist VirtualBox unter openSUSE Leap auf mehrere Pakete verteilt. Zunächst sollte das vorhandene Oracle Paket entfernt werden, fahren Sie dazu zunächst alle VMs herunter.

Deinstallieren Sie jetzt das vorhandene VirtualBox-Paket:

```
invis:~ # zypper rm VirtualBox-5.0
```

Entfernen Sie jetzt das vorhandene VirtualBox-Repository, suchen Sie zunächst nach dessen Nummer mit:

```
invis:~ # zypper repos
invis:~ # zypper rr nummer
```

Installieren Sie jetzt alle notwendigen Pakete:

```
virtualbox
virtualbox-vnc
virtualbox-websrv
phpvirtualbox
invis-vboxinit
libvncserver0
```

Übernehmen Sie die aufgeführten Pakete in eine Textdatei und installieren Sie alle Pakete dann wie folgt:

```
invis:~ # zypper in `cat /pfad/zur/datei`
```

Bevor Sie vorhandene virtuelle Maschinen wieder starten können müssen Sie das nötige Kernel-Modul laden:

```
invis:~ # systemctl start vboxdrv.service
```

Aktualisieren Sie abschließend noch den VirtualBox Extension-Pack, so Sie ihn installiert haben. Bedenken Sie hierbei, dass Sie ihn kostenfrei nur für private oder Evaluationszwecke nutzen dürfen. Die gewerbliche Nutzung ist lizenzpflichtig.

invis:~ # wget http://download.virtualbox.org/virtualbox/5.1.30/Oracle_VM_VirtualBox_Extension_Pack-5.1.30-118389.vbox-extpack

Hinweis: Den korrekten Link finden Sie unter https://www.virtualbox.org/wiki/Downloads

Installieren Sie den Extension-Pack wie folgt:

invis:~ # VBoxManage extpack install Oracle_VM_VirtualBox_Extension_Pack-5.1.30-118389.vbox-extpack

Damit ist VirtualBox auf Stand.

invisAD-setup Upgrade

Nach den bisherigen Upgrade-Prozessen läuft bei Ihnen eigentlich ein, wenn auch rundweg aktualisierter, invis-Server 11.x. Um auch Dinge, wie beispielsweise das invis-Portal zu aktualisieren und schlussendlich einen invis-Server 12.x respektive 13.x zu erhalten muss auch das invisAD-setup Paket aktualisiert werden. Weiterhin ist im Anschluss daran auch noch ein Stück Handarbeit zu erledigen.

Upgrade auf 12.x

veraltet

Paket Upgrade

Die Installation des neuen invis-Server Pakets führt zu einem manuell aufzulösenden Konflikt:

```
invis:~ # zypper in invisAD-setup-12
Repository-Daten werden geladen...
Installierte Pakete werden gelesen...
Paketabhängigkeiten werden aufgelöst...

Problem: invisAD-setup-12-12.2-13.1.noarch steht in Konflikt mit invisAD-
setup-11, das von invisAD-setup-11-11.0-16.1.noarch zur Verfügung gestellt
wurde
 Lösung 1: Deinstallation von invisAD-setup-11-11.0-16.1.noarch
 Lösung 2: invisAD-setup-12-12.2-13.1.noarch nicht installieren

Wählen Sie aus den obigen Lösungen mittels Nummer oder brechen Sie (a)b
[1/2/a] (a):
```

Wählen Sie dabei Lösung 1.

Konfigurationsanpassungen

Im Zuge der Neuinstallation des Paketes, werden einige zur invis-Server Installation gehörenden Dateien als „.rpmsave“ Dateien gesichert. Diese gilt es wiederherzustellen bzw. die neuen Dateien anzupassen.

Dies betrifft zunächst die invis-Server eigenen Konfigurationsdateien unter:

```
/etc/invis
```

Übernehmen Sie aus der Datei „invis-pws.conf.rpmsave“ die Passwörter des MySQL-root Kontos und des LDAP-Admins in die neue Datei „invis-pws.conf“. So Sie einen Benutzer zum Kopano- respektive Zarafa-Admin erkoren haben, sollten Sie dessen Passwort ebenfalls an die entsprechende Stelle in diese Datei eintragen.

Verfahren Sie für die Datei „invis.conf“ in gleicher Weise. Gehen Sie dabei sorgsam vor. Kontrollieren Sie jeden einzelnen Wert und übernehmen Sie die ursprünglichen Werte aus „invis.conf.rpmsafe“.

Für die Option „usedGroupware“ müssen Sie natürlich, wenn Sie früher Zarafa eingesetzt haben, jetzt „kopano“ eintragen.

In der neuen Datei sind ein paar neue Konfigurationsparameter hinzugekommen. Gehen wir Sie durch:

```
...
# Kontakt aus den Zertifikaten ist lokaler Email-Absender
mailSender:certmail@invis-net.loc
mailSenderName:certowner
...
```

Diese Option legt den Absender automatisch versendeter Mails fest. Ihr invis-Server versucht beispielsweise bei Problemen mit Festplatten oder RAID automatisch den zuständigen Administrator zu kontaktieren. Tragen Sie hier bitte den vollständigen Namen (mailSenderName) und dessen korrekte Email-Adresse (mailSender) des gewünschten Absenders ein.

Ebenfalls neu ist der folgende Eintrag:

```
...
# Welche ERP Loesung wird eingesetzt
# moeglich: kivitendo, wawision, none
usedERP:usederp
...
```

Was hier einzutragen ist, müssen Sie natürlich besser wissen, als wir.

Ebenfalls anzupassen ist die Konfigurationsdatei des invis-Portals:

```
/etc/invis/portal/config.php
```

Hier ist die originale Datei aus der ursprünglichen Installation allerdings erhalten geblieben. Sie müssen lediglich ein paar Einträge ergänzen.

Ändern Sie zunächst für die Variable **$GROUPWARE** den Wert von „zarafa“ auf „kopano“

Entfernen Sie die folgende Zeile:

```
...
$USER_PW_MIN_STRENGTH = '80'; // 0 = Check disabled, 100 = Max
...
```

und fügen sie statt dessen folgende Zeile ein:

```
...
$USER_PW_COMPLEX = 'on';
...
```

Mit dieser Option legen Sie fest, ob Sie komplexe Passwörter im Sinne der ActiveDirectory Richtlinien fordern oder nicht. Diesen Wert können Sie manuell anpassen, er wird aber auch vom Script ***pwsettings*** gesetzt.

Ändern Sie jetzt noch im Array „$SERVER_SERVICES“ alle Dienste die „zarafa“ im Namen enthalten in „kopano-“ um.

Abschließende Kosmetik

Abschließend geht es lediglich darum ein paar Versionsnummern anzupassen. Schließlich sollen Sie ja auch sehen, dass Sie jetzt einen invis-Server 12.x verwenden.

Ändern Sie zunächst in den Dateien

```
/etc/issue
```

und

```
/etc/issue.net
```

jeweils die Versionen des invis-Servers und von openSUSE auf die aktuelle Version. Welche invis-Server Version Sie genau installiert haben verrät Ihnen ***rpm***:

```
invis:~ # rpm -qa invisAD-setup-12
invisAD-setup-12-12.2-13.1.noarch
```

Im Beispiel ist es Version 12.2.

Die korrekte invis-Server-Versionsnummer können Sie natürlich auch noch in die Bootmanager-Konfiguration übernehmen. Dazu editieren Sie die Datei

```
/etc/default/grub
```

in der Zeile „GRUB_DISTRIBUTOR“ und führen Sie anschließend erneut folgendes Kommando aus:

```
invis:~ # grub2-mkconfig -o /boot/grub2/grub.cfg
```

Damit ist das Upgrade abgeschlossen.

Upgrade auf 13.x

Die Installation des neuen invis-Server Pakets führt zu einem manuell aufzulösenden Konflikt:

```
invis:~ # zypper in invisAD-setup-13
Repository-Daten werden geladen...
Installierte Pakete werden gelesen...
Paketabhängigkeiten werden aufgelöst...

Problem: invisAD-setup-13-13.0-12.2.noarch steht in Konflikt mit invisAD-
setup-11, das von invisAD-setup-11-11.0-16.2.noarch zur Verfügung gestellt
wurde
 Lösung 1: Deinstallation von invisAD-setup-11-11.0-16.2.noarch
 Lösung 2: invisAD-setup-13-13.0-12.2.noarch nicht installieren

Wählen Sie aus den obigen Lösungen mittels Nummer oder brechen Sie (a)b
[1/2/a] (a):
```

Wählen Sie dabei Lösung 1.

Konfigurationsanpassungen

Im Zuge der Neuinstallation des Paketes, werden einige zur invis-Server Installation gehörenden Dateien als „.rpmsave“ Dateien gesichert. Diese gilt es wiederherzustellen bzw. die neuen Dateien anzupassen.

Dies betrifft zunächst die invis-Server eigenen Konfigurationsdateien unter:

```
/etc/invis
```

Übernehmen Sie aus der Datei „invis-pws.conf.rpmsave“ die Passwörter des MySQL-root Kontos und des LDAP-Admins in die neue Datei „invis-pws.conf“. So Sie einen Benutzer zum Kopano- respektive Zarafa-Admin erkoren haben, sollten Sie dessen Passwort ebenfalls an die entsprechende Stelle in diese Datei eintragen.

Verfahren Sie für die Datei „invis.conf“ in gleicher Weise. Gehen Sie dabei sorgsam vor. Kontrollieren Sie jeden einzelnen Wert und übernehmen Sie die ursprünglichen Werte aus „invis.conf.rpmsafe“.

Für die Option „usedGroupware“ müssen Sie natürlich, wenn Sie früher Zarafa eingesetzt haben, jetzt „kopano“ eintragen.

In der neuen Datei sind ein paar neue Konfigurationsparameter hinzugekommen. Gehen wir Sie durch:

```
...
$IP_NETBASE_ADDRESS = '192.168.221.0';
...
```

Es muss die Basisadresse Ihres IP-Netzwerkes eingetragen werden.

```
...
# Kontakt aus den Zertifikaten ist lokaler Email-Absender
mailSender:certmail@invis-net.loc
mailSenderName:certowner
...
```

Diese Option legt den Absender automatisch versendeter Mails fest. Ihr invis-Server versucht beispielsweise bei Problemen mit Festplatten oder RAID automatisch den zuständigen Administrator zu kontaktieren. Tragen Sie hier bitte den vollständigen Namen (mailSenderName) und dessen korrekte Email-Adresse (mailSender) des gewünschten Absenders ein.

Ebenfalls neu ist der folgende Eintrag:

```
...
# Welche ERP Loesung wird eingesetzt
# moeglich: kivitendo, wawision, none
usedERP:usederp
...
```

Was hier einzutragen ist, müssen Sie natürlich besser wissen, als wir.

Ebenfalls anzupassen ist die Konfigurationsdatei des invis-Portals:

```
/etc/invis/portal/config.php
```

Hier ist die originale Datei aus der ursprünglichen Installation allerdings erhalten geblieben. Sie müssen lediglich ein paar Einträge ergänzen.

Ändern Sie zunächst für die Variable **$GROUPWARE** den Wert von „zarafa“ auf „kopano“

Entfernen Sie die folgende Zeile:

```
...
$USER_PW_MIN_STRENGTH = '80'; // 0 = Check disabled, 100 = Max
...
```

und fügen sie statt dessen folgende Zeile ein:

```
...
$USER_PW_COMPLEX = 'on';
...
```

Mit dieser Option legen Sie fest, ob Sie komplexe Passwörter im Sinne der ActiveDirectory Richtlinien fordern oder nicht. Diesen Wert können Sie manuell anpassen, er wird aber auch vom Script ***pwsettings*** gesetzt.

Ändern Sie jetzt noch im Array „$SERVER_SERVICES“ alle Dienste die „zarafa“ im Namen enthalten in „kopano-“ um.

Anpassen der Apache-Konfiguration

Mit der kommenden Veröffentlichung von Version 13.1 wird das Setup des Apache Webservers umfassend überarbeitet. (Version 13.0 unterscheidet sich auch schon deutlich von Ihren Vorgängerversionen, hat aber die darein gesteckten Erwartungen nicht erfüllt und wurde für 13.1 und folgende noch einmal vollständig überarbeitet.)

Auch wenn 13.1 (Stand Jan. 2018) noch nicht als „stable“ Release veröffentlicht wurde, wird nachfolgend der Umbau des Apache-Webserver Setups darauf beschrieben. Zunächst ein Blick auf die Unterschiede:

Bisher wurden folgende Apache vHosts eingerichtet:

- **intern** und **intern-ssl**: Zugriff auf das invis-Portal und darüber auf alle installierten Webapplikationen aus dem lokalen Netzwerk heraus.
- **extern**: Zugriff auf das invis-Portal und darüber auf alle installierten Webapplikationen aus dem Internet. Der Zugriff erfolgte ausschließlich verschlüsselt und über einen zufällig beim Setup zufällig bestimmten verschobenen Port. Der vHost ist so konfiguriert, dass er ein Deep-Linking auf installierte Apps am invis-Portal vorbei nach Möglichkeit verhindert.
- **owncloud**: Zugriff auf ownCloud sowohl aus dem lokalen Netz als auch dem Internet. er Zugriff

erfolgte ausschließlich verschlüsselt und über einen zufällig beim Setup zufällig bestimmten verschobenen Port. Damit in ownCloud erzeugte Freigabe Links konsistent sind, muss der Zugriff auf ownCloud sowohl von intern als auch aus dem Internet über die gleiche URL erfolgen.
- **z-push**: Zugriff auf ActiveSync via Internet.
- **dehydrated**: Zugriff auf das Let's Encrypt Challenge-Verzeichnis über Port 80.

Leider hat sich dieses Setup als zunehmend unpraktikabel erwiesen. Einerseits funktioniert der Deep-Linking-Schutz nicht fehlerfrei, andererseits bereitet der Zugriff auf ownCloud über einen verschobenen Port Dritten dann Probleme, wenn Sie hinter einem Proxy sitzen. Darüber hinaus war es mit diesem Setup nur schwer möglich ein konsistentes Setup der Kopano-Deskapp zu realisieren, welches sowohl aus dem lokalen Netz als auch dem Internet funktioniert. Durch immer wieder durchgeführte Anpassungen an diesem Setup wurde es auch zunehmend unübersichtlich. Die vollständige Überarbeitung war dringend notwendig. Dabei ging der Ansatz von invis-Server 13.0 bei weitem nicht ausreichend, er wies ähnliche Schwachpunkte auf wie das alte Setup.

Mit 13.1 werden folgende vHosts eingeführt:

- **intern** und **intern-ssl**: Diese vHost wurden kaum verändert.
- **extern**: Auch dies entspricht funktional dem alten Setup.
- **extern-combined**: Hierüber kann direkt auf z-push (ActiveSync), ownCloud und die Kopano-Webapp verschlüsselt über den Standard-Port 443. Dies ermöglicht den ownCloud-Zugriff für Dritte auch wenn Sie hinter einem Proxy sitzen und extern wie intern funktionierende KopanoDeskApp Setups.
- **dehydrated**: Zugriff auf das Let's Encrypt Challenge-Verzeichnis über Port 80.

Das neue Setup wurde vor allem durch die konsequente Verwendung von Apache Server-Flags flexibler gestaltet. Individuelle Applikations-Konfigurationen wurden aus den vHost-Konfigurationen in globale Konfigurationen verschoben und die Sicherheit des Gesamt-Setups wurde erhöht.

Für den Umstieg auf das neue Setup muss zunächst das Apache-Konfigurationsverzeichnis

```
/etc/apache2/conf.d
```

bereinigt werden:

1. Löschen Sie zunächst ggf. vorhandene Zarafa-Konfigurationsdateien.
2. Je nach dem, welche Webapplikationen sie installiert haben finden Sie dort die Dateien „dokuwiki.conf“, „owncloud.conf“, „wawision.conf“, „kivitendo.conf“ und „kimai.conf“. Diese müssen durch ihre überarbeiteten Pendants ersetzt werden. Sie finden die neuen Dateien unter:

   ```
   /usr/share/sine/templates
   ```

   Für jede der genannten Applikationen finden Sie im zuvor genannten Verzeichnis ein Unterverzeichnis, welches die jeweils neue Datei enthält. Ausnahme sind hier „Wawision“ und „Kivitendo“, die im Unterverzeichnis „erp“ zu finden sind.

3. Entfernen Sie aus der Datei

   ```
   /etc/apache2/listen.conf
   ```

den „IfDefine ownCloud“ Container vollständig (alle 3 Zeilen).

4. Sichern Sie jetzt das Verzeichnis

   ```
   /etc/apache2/vhosts.d
   ```

   und notieren Sie sich aus der Datei

   ```
   /etc/apache2/vhosts.d/invis-sslvh.conf
   ```

   den im Tag des „virtualHost“ Containers aufgeführten Port und den Namen des vHosts hinter „ServerName“. Entfernen Sie dann alle Dateien außer den beiden Templates.

5. Kopieren Sie jetzt alle mit „vh-“ beginnend benannten Dateien aus

   ```
   /usr/share/sine/templates/webserver
   ```

   nach:

   ```
   /etc/apache2/vhosts.d
   ```

6. Ersetzen Sie jetzt in der Datei

   ```
   /etc/apache2/vhosts.d/vh-extern.conf
   ```

   den Platzhalter „httpsport“ durch den zuvor notierten Port und in der ganzen Datei den Platzhalter „your.ddns-domain.net“ durch den zuvor notierten Server-Namen.

7. Ersetzen Sie jetzt in der Datei

   ```
   /etc/apache2/vhosts.d/vh-combined-ext.conf
   ```

   in der ganzen Datei den Platzhalter „your.ddns-domain.net“ durch den zuvor notierten Server-Namen.

8. Jetzt müssen Sie für alle auf Ihrem Server installierten Web-Applikationen die zugehörigen Apache-Server-Flags setzen. Die Flags sind genauso benannt, wie die zugehörigen Applikationen, allerdings müssen Sie vollständig groß geschrieben werden. Beispiel:

   ```
   invis:~ # a2enflag DOKUWIKI
   ```

   Setzen Sie zusätzlich das Flag „OWNCERTS“. Sind alle Flags gesetzt muss der Webserver neu gestartet werden.

   ```
   invis:~ # systemctl restart apache2.service
   ```

Damit ist der Umstieg auf das neue Webserver-Setup abgeschlossen. Sie können abschließend noch in Ihrem Router die Portweiterleitung für den deaktivierten ownCloud-vHost entfernen und in der SuSEfirewall2 des invis-Servers den entsprechenden Port schließen.

Verwendung von Let's Encrypt Zertifikaten vorbereiten

Eine der wichtigsten Neuerungen des invis-Servers ab Version 12 ist die Verwendung offizieller Server-Zertifikate von Let's Encrypt für externe Server-Zugriffe. Um von dieser Neuerung Gebrauch zu machen sind Apache-Konfigurationsdateien anzupassen und das Let's Encrypt System zu aktivieren.

Um Let's Encrypt zu nutzen müssen Sie folgende Apache-Konfigurationsdateien anpassen:

invis-Server 12.x

- z-push_vh.conf
- invis-sslvh.conf
- owncloud_vh.conf

invis-Server 13.x

- vh-extern.conf
- vh-combined-ext.conf

Alle Dateien finden Sie im Verzeichnis:

```
/etc/apache2/vhosts.d
```

In allen genannten Dateien finden Sie relativ weit am Anfang die folgenden beiden Zeilen:

Hinweis: *wenn Sie auf Version 13 aktualisieren haben Sie diesen Punkt bereits im vorigen Schritt erledigt*

```
    SSLCertificateFile /etc/apache2/ssl.crt/invis-server.crt
    SSLCertificateKeyFile /etc/apache2/ssl.key/invis-server.key
```

Ersetzen Sie diese bitte durch folgende Zeilen:

```
    <IfDefine LETSENCRYPT>
        #   You can use per vhost certificates if SNI is supported.
        SSLCertificateFile /etc/dehydrated/certs/your.ddns-
domain.net/cert.pem
        SSLCertificateKeyFile /etc/dehydrated/certs/your.ddns-
domain.net/privkey.pem
        SSLCertificateChainFile /etc/dehydrated/certs/your.ddns-
domain.net/chain.pem
    </IfDefine>

    <IfDefine OWNCERTS>
        SSLCertificateFile /etc/apache2/ssl.crt/invis-server.crt
        SSLCertificateKeyFile /etc/apache2/ssl.key/invis-server.key
    </IfDefine>
```

Ersetzen Sie dann im oberen „IfDefine“ Container in allen drei Dateien „your.ddns-domain.net“ mit

dem Hostnamen, über den Sie Ihren invis-Server via Internet erreichen.

Sie können Ihren Apache-Webserver jetzt zunächst wieder mit den eigenen Zertifikaten in Betrieb nehmen. Dazu muss noch ein sogenanntes Server-Flag gesetzt und Apache neu gestartet werden:

Hinweis: *wenn Sie auf Version 13 aktualisieren haben Sie diesen Punkt bereits im vorigen Schritt erledigt*

```
invis:~ # a2enflag OWNCERTS
invis:~ # systemctl restart apache2.service
```

Jetzt muss ein neuer Apache vHost eingerichtet werden. Dieser vHost wird zur Aktualisierung der Zertifikate benötigt. Die hierfür benötigte Konfigurationsdatei hat Ihr invis-Server bereits im Gepäck. Kopieren Sie sie einfach ins oben genannte vHosts-Verzeichnis:

invis-Server 12.x

```
invis:~ # cp /usr/share/doc/packages/invisAD-
setup/examples/dehydrated/dehydrated.conf /etc/apache2/vhosts.d/
```

invis-Server 13.x

```
invis:~ # cp /usr/share/sine/templates/webserver/vh-dehydrated.conf
/etc/apache2/vhosts.d/
```

In dieser Datei müssen Sie jetzt lediglich eine kleine Anpassung vornehmen. Ermitteln Sie die IP-Adresse Ihrer externen Netzwerkschnittstelle:

```
invis:~ # ifconfig extern
extern    Link encap:Ethernet  Hardware Adresse 00:15:90:31:DD:23
          inet Adresse:192.168.178.43  Bcast:192.168.178.255
Maske:255.255.255.0
...
```

Tragen Sie diese Adresse wie nachfolgend gezeigt in obige Datei ein:

```
...
<Virtualhost 192.168.178.43:80>
    ServerName your.ddns-domain.net
    DocumentRoot /srv/www/htdocs/dehydrated
...
```

Ersetzen Sie auch hier „your.ddns-domain.net“ durch den Namen über den Sie Ihren invis-Server via Internet ansprechen. Legen Sie jetzt das DocumentRoot-Verzeichnis für den neuen vHost an:

```
invis:~ # mkdir -p "/srv/www/htdocs/dehydrated/.well-known/acme-challenge"
invis:~ # chown dehydrated "/srv/www/htdocs/dehydrated/.well-known/acme-
challenge"
```

Jetzt muss der Let's Encrypt-Client ***dehydrated*** noch konfiguriert werden. Auch hier hat Ihr invis-Server bereits eine vorbereitete Konfigurationsdatei im Gepäck, die an die richtige Stelle kopieren

können:

invis-Server 12.x

```
invis:~ # cp /usr/share/doc/packages/invisAD-
setup/examples/dehydrated/config /etc/dehydrated/
```

invis-Server 13.x

```
invis:~ # cp /usr/share/sine/templates/webserver/dehydrated/config
/etc/dehydrated/
```

In dieser Datei ist lediglich eine kleine Anpassung vorzunehmen. Tragen Sie in der Zeile „CONTACT_EMAIL“ relativ weit unten in der Datei eine gültige Email-Adresse die beim Erstellen der Zertifikate genutzt werden kann und befreien Sie die Zeile vom führenden Kommentarzeichen:

```
...
CONTACT_EMAIL=info@echtedomain.de
...
```

Zum Abschluss der Konfiguration müssen Sie noch in der Datei

```
/etc/dehydrated/domains.txt
```

den Namen Ihres Servers eintragen, der in Ihr Server-Zertifikat eingetragen wird. Dies ist natürlich wieder der Name unter dem Ihr invis-Server via Internet erreichbar ist. Alle Beispiel-Namen, die Sie in der Datei vorfinden sind zu entfernen.

Weiterhin tragen Sie in die Datei

```
/etc/dehydrated/hooks.sh
```

in den Funktion „deploy_cert()“ folgende Zeile ein:

```
deploy_cert() {
...

    # Reload Apache Webserver
    systemctl reload apache2.service
}
```

Für die Inbetriebnahme, bzw. den Wechsel des eigenen Zertifikats auf ein verifiziertes von Let's Encrypt müssen Sie auf Ihrem Router noch eine zusätzliche Portweiterleitung einrichten. Dort muss Port 80 auf Ihren invis-Server weitergeleitet werden. Selbstverständlich muss Port 80 dann auch in der Firewall geöffnet werden.

Ändern Sie dazu in der Datei

```
/etc/sysconfig/SuSEfirewall2
```

Zeile 291 (ca.). Tragen Sie einfach die Zahl 80 durch Leerzeichen getrennt zusätzlich ein:

```
FW_SERVICES_EXT_TCP="80 443 53xxx 50xxx 59xxx 1194"
```

Starten Sie jetzt die Firewall neu:

```
invis:~ # rcSuSEfirewall2 restart
```

Zur Inbetriebnahme müssen Sie jetzt nur noch unser Script ***actdehydrated*** ausführen:

```
invis:~ # actdehydrated
```

Testen Sie es aus indem Sie einfach eine „https“ Verbindung zu auf den im Internet gültigen Namen Ihres invis-Servers öffnen.