invisAD 10.4 -> invisAD 11.0

Mit Version 11.0 des invis-Servers ergeben sich ein paar strukturelle Unterschiede. Allem voran geht dabei die Umstellung der „Public Key Infrastructure“, also die Organisation von Schlüsseln und Zertifikaten für die verschiedenen Verschlüsselungsaufgaben.

Bisher wurden zwei sogenannte Zertifizierungsstellen (CA) erstellt, davon diente eine zur Verwaltung der Schlüssel und Zertifikate für die verschiedenen Serverdienste (LDAP, Mail, Web) und eine zweite ausschließlich für openVPN. Zur Pflege der Schlüssel- und Zertifikate wurde das Toolkit „easy-rsa“ in Version 2.0 verwendet, der Rest direkt mit ***openssl*** und eigenen Scripten.

Das Toolkit **„easy-rsa“** liegt inzwischen in Version 3.0 vor und wurde im Zuge der Weiterentwicklung deutlich verbessert. Nach ersten Tests damit war klar, dass easy-rsa 3.0 ideal zur Verwaltung aller Schlüssel- und Zertifikaten eines invis-Servers ist. Kurzum ab Version 11.0 verfügen invis-Server nur noch über eine mit **easy-rsa** erstellte Zertifizierungsstelle.

Aus dem bisherigen Script ***serverkeys*** zur Generierung von Serverzertifikaten wurde ***inviscerts***. Die Handhabung des Scripts wird im Abschnitt „invis Administration“ erläutert.

Beim Upgrade von 10.4 auf 11.0 müssen die Schritte zum Aufbau einer PKI die ***sine*** während des Setups erledigt manuell vorgenommen werden.

Aufbau einer Public Key Infrastruktur mit easy-rsa

Unter anderem zur Nutzung im invis-Server wurde ein easy-rsa 3.x RPM im Repository „spins:invis:common“ gebaut. Die nachfolgende Anleitung beschreibt, wie mit easy-rsa eine PKI manuell aufgebaut wird. Die spätere Verwaltung von Zertifikaten übernimmt auf invis-Servern das Script ***inviscerts***.

Ab Version 3.0 stellt easy-rsa nur noch ein einziges Script ***easyrsa*** für alle Aufgaben zur Verfügung. Sämtliche Konfigurationsdateien liegen in:

```
/etc/easy-rsa
```

Vor dem Aufbau einer PKI muss im genannten Verzeichnis die Datei „vars“ an die eigenen Bedürfnisse angepasst werden. Es ist im Unterschied zu älteren easy-rsa Versionen nicht mehr notwendig die Variablen in dieser Datei manuell mittels ***source*** in die aktuelle Shell Umgebung zu laden. Dies erledigt ***easyrsa*** selbsttätig.

Vorbereitung

Angepasst werden müssen zumindest folgende Zeilen:

Name der PKI:
(Zeile 65)

```
...
set_var EASYRSA_PKI             "$EASYRSA/fsp-net.loc"
```

```
...
```

PKI individualisieren:
(Ab Zeile 84)

```
...
set_var EASYRSA_REQ_COUNTRY     "DE"
set_var EASYRSA_REQ_PROVINCE    "Hessen"
set_var EASYRSA_REQ_CITY        "Schotten"
set_var EASYRSA_REQ_ORG         "FSP Computer und Netzwerke"
set_var EASYRSA_REQ_EMAIL       "stefan@invis-server.org"
set_var EASYRSA_REQ_OU          "invis-Server.org"
...
```

Die vorgegebene Länge der DH-Parameter ist auf 2048 Bit eingestellt, dies reicht nach gegenwärtigem Stand der Technik aus. Eine Erhöhung auf 4096 Bits verlängert den Bau der DH-Datei immens (Die Rede ist hier von Stunden).

Voreingestellt sind Gültigkeitsdauern für CA und damit signierte Zertifikate von 10 Jahren. Kann man lassen, die Lebensdauer der Zertifikate im Vergleich zur CA zu verkürzen ist auch OK.

Jetzt kann die PKI vorbereitet werden:

```
linux:~ # easyrsa init-pki
```

Damit wird unter „/etc/easy-rsa“ eine Verzeichnisstruktur für die neue PKI angelegt und die Vorbereitungen sind abgeschlossen.

PKI erzeugen

Benötigt werden folgende Komponenten:

1. Eine CA zum signieren von Server und Client Zertifikaten
2. Eine Diffie-Hellman Parameter Datei für sicheren Schlüsselaustausch
3. Eine CRL Datei um Zertifikate zurückzuziehen

CA erstellen:

Auch hier genügt ein einziger Befehl:

```
linux:~ # easyrsa build-ca
```

Beim Bau der CA wird ein Passwort erfragt, dieses Passwort wird später beim signieren von Zertifikaten benötigt. Dieses Passwort darf nicht in falsche Hände gelangen.

Diffie-Hellman Parameter erstellen:

```
linux:~ # openssl gendh -out $path/$domain/dh.pem -2 2048
```

```
linux:~ # openssl gendh -out $path/$domain/dh_512.pem -2 512
```

Die Diffie-Hellman Parameter-Dateien werden bewusst direkt mit ***openssl*** erstellt, da das Script ***easyrsa*** immer die in der Konfiguration vorgegebene Schlüssellänge verwendet. Diese sollte auf 4096Bit eingestellt sein. Der Bau einer 4096 Bit großen DH-Parameterdatei dauert allerdings je nach System mehrere Stunden. Außerdem wird für die überarbeitete Postfix-Konfiguration noch eine 512Bit lange Datei benötigt.

CRL erzeugen

```
linux:~ # easyrsa gen-crl
```

Hierfür wird wieder das CA-Passwort benötigt.

Aufbau der invis-Server PKI Verzeichnisstruktur

Es sind drei neue Verzeichnisse anzulegen:

```
linux:~ # mkdir /etc/invis/certs
linux:~ # mkdir /etc/invis/private
linux:~ # mkdir /etc/openvpn/keys
```

Jetzt muss das Stammzertifikat der Zertifizierungsstelle „verteilt“ werden:

```
linux:~ # cp /etc/easy-rsa/yourdomain.tld/ca.crt /etc/invis/certs/
linux:~ # cp /etc/easy-rsa/yourdomain.tld/ca.crt /etc/openvpn/keys/
```

Ebenfalls zu kopieren sind die CRL, sowie die Diffie-Hellman-Parameter Dateien

```
linux:~ # cp /etc/easy-rsa/yourdomain.tld/dh.pem /etc/openvpn/keys/
linux:~ # cp /etc/easy-rsa/yourdomain.tld/crl.pem /etc/openvpn/keys/
linux:~ # cp /etc/easy-rsa/yourdomain.tld/dh*.pem /etc/postfix/
```

Jetzt können wie im Abschnitt „invis Administration“ beschrieben Schlüssel und Zertifikate für den Server erstellt werden.

Um die Umstellung zu komplettieren, müssen die Pfade zu Schlüsseln- und Zertifikaten in den Konfigurationsdateien des Apache-Webservers, von Postfix und ggf. Dovecot, von Samba und openVPN an die neuen Verzeichnisse und Dateinamen anzupassen.

Eine detaillierte Anleitung dafür folgt.