OpenVPN server

Návod navazuje na Linuxtero

Článek obsahuje chyby. Spodní skript má fungovat dobře

Literatura

- http://www.root.cz/clanky/jak-na-openssl/
- http://www.root.cz/clanky/openvpn-vpn-jednoduse/
- http://openvpn.net/index.php/documentation/howto.html#examples
- http://www.czela.net/wiki/index.php/OpenVPN
- http://openvpn.net/index.php/open-source/documentation/howto.html#examples

Krok za krokem

- Instalace programů (server - klienti):

```
apt-get install openvpn openssl
```

- Pro generování certifikátů pro uživatele je potřeba vytvořit certifikační autoritu. Certifikační autorita (v tomto případě server) zajišťuje důvěryhodnost připojení uživatelů k serveru. V první řadě vytvoříme adresář pro certifikační autoritu a potřebné podadresáře:

```
cd /etc/ssl/
mkdir demoCA demoCA/certs demoCA/crl demoCA/newcerts demoCA/private
```

- Nyní je třeba ještě vytvořit prázdný soubor index.txt a soubor serial s obsahem „01“:

```
touch /etc/ssl/demoCA/index.txt
echo 01 > /etc/ssl/demoCA/serial
```

- Dále vygenerujeme certifikát certifikační autority a podepíšeme jej sám sebou:

```
cd /etc/ssl/demoCA
openssl req -new -x509 -nodes -out cacert.pem -keyout cakey.pem -days 3650
```

- Následně umístíme vygenerovaný klíč a certifikát do správných adresářů. Certifikát přesuneme do /etc/ssl/demoCA a klíč do /etc/ssl/demoCA/private. Klíč by měl být právy ochráněn před neoprávněnou manipulací, změníme práva na 400 (chmod 400 cakey.pem):

```
mv cacert.pem certs/ && mv cakey.pem private/
chmod 400 private/cakey.pem
```

- Správnost cest je třeba ještě jednou zkontrolovat v konfiguračním souboru programu openssl nacházející se v „/etc/ssl/openssl.cnf“ v sekci „[CA_default]“. Defaultní nastavení souboru je následující:

```
[ CA_default ]

dir             = ./demoCA            # Kořenový adresář CA
certs           = $dir/certs            # Adresář obsahující vydané
certifikáty
crl_dir         = $dir/crl              # Adresář obsahující CRL
database        = $dir/index.txt        # Index databáze

new_certs_dir   = $dir/newcerts         # Adresář pro nové certifikáty

certificate     = $dir/certs/cacert.pem       # Certifikát CA
serial          = $dir/serial           # Soubor se sérií certifikátů
(počítá)

crl             = $dir/crl/crl.pem          # Aktuální CRL
private_key     = $dir/private/cakey.pem# Soukromý klíč CA
RANDFILE        = $dir/private/.rand    # Soubor pro generování náhodných
čísel

policy          = policy_anything       # Politiku certifikátů zvolíme
volnou
```

Pokud z nějakého důvodu nefungují relativní cesty pomocí „$dir“, je třeba je zadat staticky. (Chyba vzniká v bodu 8)

```
[ CA_default ]

dir     = /etc/ssl/demoCA        # Where everything is kept
certs       = /etc/ssl/demoCA/certs        # Where the issued certs are kept
crl_dir     = $dir/crl        # Where the issued crl are kept
database    = /etc/ssl/demoCA/index.txt    # database index file.
#unique_subject = no            # Set to 'no' to allow creation of
                    # several ctificates with same subject.
new_certs_dir   = /etc/ssl/demoCA/newcerts        # default place for new
certs.

certificate = /etc/ssl/demoCA/certs/cacert.pem     # The CA certificate
#certificate    = $dir/cacert.pem     # The CA certificate
serial      = /etc/ssl/demoCA/serial         # The current serial number
crlnumber   = $dir/crlnumber    # the current crl number
                    # must be commented out to leave a V1 CRL
crl     = $dir/crl.pem         # The current CRL
private_key = /etc/ssl/demoCA/private/cakey.pem# The private key
RANDFILE    = $dir/private/.rand    # private random number file

x509_extensions = usr_cert        # The extentions to add to the cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt    = ca_default        # Subject Name options
cert_opt    = ca_default        # Certificate field options
```

```
# Extension copying option: use with caution.
# copy_extensions = copy

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crlnumber must also be commented out to leave a V1 CRL.
# crl_extensions    = crl_ext

default_days    = 365           # how long to certify for
default_crl_days= 30            # how long before next CRL
default_md  = sha1            # which md to use.
preserve    = no            # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy      = policy_anything
```

- Pokud je již všechno správně nastaveno, lze začít generovat certifikáty. Vygenerujeme ho tedy pro server.

! Při vyplňování parametrů je nutné zadat stejné parametry uvedené v sekci [policy_match] v konfiguračním souboru „/etc/ssl/openssl.cnf“. ! match = má se rovnat (pro countryName vyplnuji vzdy „CZ“) ! optional = nemusí se vyplnovat ! supplied = asi se musi vyplnovat, napriklad IP nebo jmeno (kazdy certifikat uniq?)

```
mkdir server && cd server
openssl req -new -nodes -out request.pem -keyout key.pem -days 1095
```

- Z předchozího příkazu dostaneme dva soubory (žádost „request.pem“ a klíč „key.pem“). Následně potvrdíme certifikační autoritou žádost o vydání certifikátu:

```
openssl ca -in request.pem -out cert.pem
```

! Zde může nastat problém popsaný v bodě 6!

```
# Chyba
openssl ca -in request.pem -out cert.pem
Using configuration from /usr/lib/ssl/openssl.cnf
Error opening CA certificate ./demoCA/cacert.pem
17226:error:02001002:system library:fopen:No such file or
directory:bss_file.c:352:fopen('./demoCA/cacert.pem','r')
17226:error:20074002:BIO routines:FILE_CTRL:system lib:bss_file.c:354:
unable to load certificate
```

```
# Dobře
openssl ca -in request.pem -out cert.pem
Using configuration from /usr/lib/ssl/openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
```

```
 ...
 ... výpis detailů ceritifikátu
 ...
Certificate is to be certified until Nov 18 11:27:39 2010 GMT (365 days)
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

- Zkopírujeme soubory na patřičná místa na serveru

```
mv cert.pem /etc/openvpn/cert.pem
mv key.pem /etc/openvpn/key.pem
```

- Nyní vytvoříme certifikát pro uživatele, a potvrdíme certifikační autoritou:

```
mkdir client && cd client
openssl req -new -nodes -out request.pem -keyout key.pem -days 1095
openssl ca -in request.pem -out cert.pem
```

- Po úspěšném provedení předchozího příkazu dostáváme certifikát potvrzený certifikační autoritou. Tímto způsobem vytvoříme certifikát pro server a klienty. Bezpečnou cestou přeneseme klíč , certifikát a certifikát certifikační autority do počítačů. Dále je třeba vytvořit soubor s Diffie-Hellmann algoritmem (více o DH na: http://www.algoritmy.net/article/84/Diffie-Hellman)

```
openssl dhparam -out /etc/ssl/demoCA/dh1024.pem 1024
```

- V serveru vytvoříme konfigurační soubor pro připojení virtuální sítě. Konfigurační soubor vytvoříme v „/etc/openvpn/vpn_server.conf“. Soubor bude obsahovat následující kód:

```
mode server
tls-server
dev tap0
port 1194

ifconfig 10.0.1.1 255.255.255.0
ifconfig-pool 10.0.1.100 10.0.1.200 255.255.255.0
duplicate-cn
proto udp

ca /etc/ssl/demoCA/certs/cacert/cacert.pem
cert /etc/openvpn/cert.pem
key /etc/openvpn/key.pem
dh /etc/ssl/demoCA/dh1024.pem

log-append /var/log/openvpn
status /tmp/vpn.status 10

user root
group root
comp-lzo
```

```
verb 3

keepalive 1 220
```

- V počítači klienta vytvoříme konfigurační soubor pro připojení virtuální sítě. Editujeme soubor „/etc/openvpn/vpn_client.conf“ a doplníme správné údaje pro připojení:

```
remote 1.2.3.4 ### IP adresa serveru
tls-client
dev tap
pull

mute 10
ca /etc/openvpn/cacert.pem
cert /etc/openvpn/cert.pem
key /etc/openvpn/key.pem

comp-lzo
verb 3
```

- Nyní máme konfiguraci hotovou a lze ji vyzkoušet. Nejdříve spustíme vpn server „/etc/init.d/openvpn start“. A dále použijeme nastavení pro otevření spojení:

```
cd /etc/openvpn
openvpn --config ./vpn_client.conf
```

Dostaneme vypis podobny tomuto:

```
OpenVPN 2.0.9 i486-pc-linux-gnu [SSL] [LZO] [EPOLL] built on May 21 2007
WARNING: No server certificate verification method has been enabled.  See
http://openvpn.net/howto.html#mitm for more info.
WARNING: file 'vpn.key' is group or others accessible
LZO compression initialized
Control Channel MTU parms [ L:1574 D:138 EF:38 EB:0 ET:0 EL:0 ]
Data Channel MTU parms [ L:1574 D:1450 EF:42 EB:135 ET:32 EL:0 AF:3/1 ]
Local Options hash (VER=V4): 'd79ca330'
Expected Remote Options hash (VER=V4): 'f7df56b8'
UDPv4 link local (bound): [undef]:1194
UDPv4 link remote: 1.2.3.4:1194
TLS: Initial packet from 1.2.3.4:1194, sid=3810d946 e78ea6ad
VERIFY OK: depth=1,
/C=CS/ST=Czech/L=Plzen/O=server/OU=server/CN=duckd/emailAddress=duckd@email.
cz
VERIFY OK: depth=0,
/C=CS/ST=Czech/L=Remote/O=client/OU=duckd/CN=duckd/emailAddress=duckd@email.
cz
Data Channel Encrypt: Cipher 'BF-CBC' initialized with 128 bit key
Data Channel Encrypt: Using 160 bit message hash 'SHA1' for HMAC
authentication
Data Channel Decrypt: Cipher 'BF-CBC' initialized with 128 bit key
Data Channel Decrypt: Using 160 bit message hash 'SHA1' for HMAC
```

```
authentication
Control Channel: TLSv1, cipher TLSv1/SSLv3 DHE-RSA-AES256-SHA, 1024 bit RSA
[duckd] Peer Connection Initiated with 1.2.3.4:1194
SENT CONTROL [duckd]: 'PUSH_REQUEST' (status=1)
PUSH: Received control message: 'PUSH_REPLY,ping 10,ping-restart
220,ifconfig 10.0.1.100 255.255.255.0'
OPTIONS IMPORT: timers and/or timeouts modified
OPTIONS IMPORT: --ifconfig/up options modified
TUN/TAP device tap0 opened
ifconfig tap0 10.0.1.100 netmask 255.255.255.0 mtu 1500 broadcast 10.0.1.255
Initialization Sequence Completed
```

- Na serveru a clientu se lze přesvědčit že jsou vytvořeny zařízení. Příkaz: ifconfig, zařízení tap0. A měl by fungovat ping.

Skript

- Soubory na serveru:
  - vpn_server.conf ### Upravíme parametry připojení v konfiguračním souboru pro server
  - bash server_install.sh ### Pokud máme tyto dva soubory ve stejném adresáři, instalujeme openvpn na server. Výsledkem má být běžící openvpn server v terminálu.
  - server_generuj_certifikat.sh ### Generuj přístup jednomu klientovi a zabal do balíku /etc/ssl/demoCA/client_X.tar.
- Soubory na klientovi:
  - vpn_client.conf ### Upravíme klientovo konfigurák\
  - client_X.tar ### balík získáme ze serveru (bezpečnou cestou)\
  - bash client_install.sh ### pokud máme tyto tři soubory ve stejném adresáři, instalujeme openvpn na klienta\

Server

- vpn_server.conf

```
mode server
port 1194
proto tcp-server
tls-server
dev tap0

ifconfig 10.0.1.1 255.255.255.0
ifconfig-pool 10.0.1.100 10.0.1.200 255.255.255.0
duplicate-cn

ca /etc/openvpn/cert/cacert.pem
cert /etc/openvpn/cert/cert.pem
key /etc/openvpn/cert/key.pem
dh /etc/openvpn/cert/dh1024.pem
```

```
log-append /var/log/openvpn
status /tmp/vpn.status 10

user root
group root
comp-lzo
verb 3

keepalive 1 220
```

- server_install.sh

```
wd=`pwd`

echo -e '\E[31m 1) Instalace programů na server \033[0m'
apt-get install -y openvpn openssl

echo -e '\E[31m 2) Vytvoříme adresář pro certifikační autoritu a potřebné
podadresáře \033[0m'
mkdir -p /etc/ssl/demoCA/{certs,private,crl,newcerts}

echo -e '\E[31m 3) Vytvoříme prázdný soubor index.txt a soubor serial s
obsahem 01 \033[0m'
touch /etc/ssl/demoCA/index.txt
echo 01 > /etc/ssl/demoCA/serial

echo -e '\E[31m 4) Vygenerujeme certifikát certifikační autority a
podepíšeme jej sám sebou \033[0m'
cd /etc/ssl/demoCA
openssl req -new -x509 -nodes -out cacert.pem -keyout cakey.pem -days 3650

echo -e '\E[31m 5) Umístíme vygenerovaný klíč a certifikát do správných
adresářů \033[0m'
chmod 400 cakey.pem
mv cakey.pem private/

echo -e '\E[31m 6) Vytvoříme soubor s Diffie-Hellmann algoritmem \033[0m'
openssl dhparam -out dh1024.pem 1024

echo -e '\E[31m 7) Vytvoříme žádost o certifikát a klíč pro openvpn server
\033[0m'
cd /etc/ssl
openssl req -new -nodes -out request.pem -keyout key.pem -days 1095

echo -e '\E[31m 8) Žádost o certifikát potvrdíme a vydáme certifikát
\033[0m'
openssl ca -in request.pem -out cert.pem
rm request.pem
mv cert.pem demoCA/certs/
mv key.pem demoCA/private/
```

```
echo -e '\E[31m 9) Zkopírujeme potřebné soubory do adresáře s openvpn
\033[0m'
mkdir /etc/openvpn/cert/
cp /etc/ssl/demoCA/{cacert.pem,certs/cert.pem,private/key.pem,dh1024.pem}
/etc/openvpn/cert/

echo -e '\E[31m 10) Přesunem konfigurak \033[0m'
cd $wd
cp vpn_server.conf /etc/openvpn/vpn_server.conf

echo -e '\E[31m 11) Startujem :) \033[0m'
/etc/init.d/openvpn start
```

- server_generuj_certifikat.sh

```

cd /etc/ssl/demoCA/

function generuj_certifikat
{
  openssl req -new -nodes -out request.pem -keyout key.pem -days 1095
  openssl ca -in request.pem -out cert.pem
  rm request.pem
  tar -cvf client_$number.tar key.pem cert.pem cacert.pem
  rm key.pem cert.pem
}

for (( number=1 ; $number-1000 ; number=$number+1 ))
do
  if [ -f client_$number.tar ]
    then
      echo the file exists
    else
      generuj_certifikat
      exit
  fi
done
```

- Pokud chcete odebrat openvpn ze serveru včetně veškerých vytvořených soborů a adresářů

```
apt-get purge openvpn && rm -r /etc/openvpn/ && rm -r /etc/ssl/demoCA/
```

Klient

- vpn_client.conf

```
remote 192.168.1.10 ### IP adresa serveru
dev tap
proto tcp-client
port 1194
tls-client
```

```
pull

mute 10
ca cacert.pem
cert cert.pem
key key.pem

comp-lzo
verb 3
```

- client_install.sh

```
echo "1) Instalace programů na klienta"
apt-get install -y openvpn openssl

echo "Přesunem konfigurak"
mv vpn_client.conf /etc/openvpn/vpn_client.conf

echo "Rozbalíme balík a pošleme *.pem soubory na správné místo"
tar -xvf client_*.tar; mv *.pem /etc/openvpn/

echo "Spustíme připojení klienta na server"
openvpn --config /etc/openvpn/vpn_client.conf
```

Konec

- **Vytvořil:** DuckD
- **Pomáhali a radili:** Krtko, fenquick, trako, lkopecky