Vytvoření certifikační autority ručně s pomocí OpenSSL

Nejdříve si ukážeme jak si takovou certifikační autoritu sami vytvoříme jen s pomocí OpenSSL. Vše co potřebujeme je balíček s programem který se na Debian Etch jmenuje `openssl`. Pokud jej nemáme nainstalován, nainstlaujeme:

```
#aptitude install openssl
```

Nyní si vybereme kde budeme mít adresář s certifikační autoritou. Mnou uváděný postup provádí vytvoření a práci s certifikační autoritou pod uživatelem. Já jsem si pro ukázkovou certifikační autoritu vybral adresář `~/firma/ca`. Při tomhle postupu si jako užvatel můžeme udržovat několik certifikačních autorit. Vytvoříme tedy adresář a přepneme se do něj.

```
$mkdir -p ~/firma/ca

$cd ~/firma/ca
```

Nyní potřebujeme konfigurační soubor pro openssl. Protože si tento budeme upravovat pro každou certifikační autoritu, nakopírujeme si ukázkový soubor do právě vytvořeného adresáře. Jako ukázkový jsem použil ten standardně instalovaný s balíčkem `openssl` jenž se nachází v `/etc/ssl/openssl.cnf`.

```
$cp /etc/ssl/openssl.cnf .
```

Konfigurační soubor si upravíme. Nejdříve nastavíme adresář v kterém budou všechny soubory. Tento bude `~/firma/ca/ca`. Parametr jímž tak činíme se jmenuje `dir` a nachází se v sekci `[ CA_default ]`.

```
[ CA_default ]

dir = ./ca
```

Tato změna úplě stačí pro funkčnost. Ale je lépe podívat se i na několik dalších parametrů a nastavit si je přesně podle potřeb konkrétní certifikační autority. Jedná se v první řadě o parametry v sekci `[ req_distinguished_name ]`, které popisují komu je certifikát vydáván. Protože předpokládám že certifikáty vydávate pro vlastní potřebu, nebo potřebu firmy v které pracujete, má smysl tyto předvyplnit.

```
[ req_distinguished_name ]

countryName_default = CZ
```

```
stateOrProvinceName_default = .

localityName_default = Praha

0.organizationName_default = Firma, s.r.o.

organizationUnitName_default = IT oddeleni
```

Další nastavení které považuji za užitečné je zmena *policy* v souvislosti s parametrem `stateOrProvinceName`. Tento je totiž vyžadován, a v našem prostředí nemá smysl. Proto jej nastavím na`optional`. Parametr se nachází v sekci `[ policy_match ]`

```
[ policy_match ]

stateOrProvinceName = optional
```

Z řady parametrů ještě uvedu jeden, a to `default_days` který určuje počet dní na které je certifikát vydávan. Tedy počet dní ode dneška po které je certifikát platný. Po vypršení této doby je nutné certifikát prodloužit. Tento parametr se nachází v sekci `[ CA_defaults ]`

```
default_days=3600
```

Můžeme dle vlastní potřeby modifikovat i další parametry, tady laskavého čtenáře odkáži na dokumentaci k programu OpenSSL.

Máme vytvořen a k vlastnímu obrazu uprave konfigurační soubor a nyní si vytvoříme potřebné adresáře. Je to jednak samotný adresář `ca` ve kterém jsou všechny soubory certifikační autority, tak ji jeho podadresáře.

```
$mkdir ca

$mkdir ca/newcerts

$mkdir ca/private

$chmod 0700 ca/private
```

Dále si vytvoříme index certifikátů a inicializujeme jejich počitadlo.

```
$touch ca/index.txt
```

```
$echo 01 >ca/serial
```

Rovněž nastavíme přístupová práva aby se k naší certifikační autoritě nedostal nikdo jiný. Znalost klíče by mu titiž umožnila vydávat falešné certifikáty.

```
$chmod 0700 ~/firma/ca

$chmod 0700 ca
```

Máme vytvořeny adresáře a vytvoříme samotnou certifikační autopritu.

```
$cd ~/firma/ca

$openssl req -config openssl.cnf -new -x509 -out cacert.pem
-keyout cakey.pem -days 7500 -newkey rsa:4096
```

Při vytváření klíče certifikační autority jsme vyzváni k zadání přístupové fráze k tomuto klíči. Pomocí fráze je samotný klíč zašifrován, takže dostane-li se k němu nepovolaná osoba, nemůže jej bez znalosti fráze použít. Tuto frázi je ovšem nutno zadávat kdykoliv se klíč použije, tedy při podepisování každé žádosti o certifikát. Jsme li si jistí že se k našemu klíči nikod nedostane, a potřebujeme li podepisování automatizovat, můžeme zašifrování klíče pomocí přístupové fráze vypnout přepínačem `-nodes`.

FIXME: *Upozornil bych na použití přepínače `-des3` který zajistí že samotný klíč bude zašifrován pomocí přístupové fráze. Toto je důležité neb kdyby se někdo dostal k souboru s klíčem, potřebuje pro jeho použití ještě znát přístupovou frázi. Na druhou stranu to taky znamená, že kdykoliv budeme chtít naší autoritou podepsat připravenou žádost, musíme zadávat přístupovou frázi. Pokud jsme si zcela jisti, že se k našemu klíči nikdo nedostane a zadavání fráze je pro nás velká překážka pro automatické podepisování žádostí, odstraníme tento přepínač a použijeme místo něj jiný `-nodes`.*

Protože program **openssl** očekává konfigurační soubor na standardních místech jako je `/etc/ssl/openssl.cnf`, musíme mu oznámit že má použít náš. Tak učiníme parametrem `-config openssl.cnf`. Místo tohoto parametry je možné použít proměnnou prostředí `OPENSSL_CONF`. To s výhodou učiníme později až budeme práci s certifikáty zjodnodušovat vytvořením několik vlastních skriptů.

Dalším parametrem na který bych chtěl upozornit je `-days 7500`. Tento určuje dobu trvání certifikátu, tedy do kdy platí, v počtu dní ode dneška. V případě certifikátu certifikační autority zvolíme dostatečně velkou hodnotu. V uvedeném případě je to cca 20 let.

A poslední parametr který zmíním je `-newkey rsa:4096`. Tento určuje délku rsa klíče v bitech. Pro dnešní dobu je doporučované minimum 2048. Protože se jedná o klíč certifikační autoritiy, použiji raději klíč delší a to 4096 bitů dlouhý.

Vytvořené soubory umístníme na místo kde jsou očekávany a nastavíme jim přístupová práva.

```
$mv cacert.pem ca/# certifikát autority, veřejný soubor
```

```
$chmod 0400 cakey.pem# soukromý klíč autority, je neveřejný

$mv cakey.pem ca/private# a je očekáván v tomto adresáři
```

Tímto máme certifikační autoritu připravenu k použití, k podepisování žádostí o certifikáty.

Vytvoření certifikační autority: Nejdříve si ukážeme jak si takovou certifikační autoritu sami vytvoříme. Vše co potřebujeme je balíček s programem, který se na Linux distribuci Debian jmenuje openssl. Pokud jej nemáme nainstalován, provedeme následující příkaz:

```
# aptitude install openssl
```

Nyní si vybereme, kde budeme mít adresář s certifikační autoritou. Zde uváděný postup uvádí vytvoření a práci s certifikační autoritou pod uživatelem. Pro ukázkovou certifikační autoritu je vybrán adresář ~/firma/ca. Při tomto postupu si jako užvatel můžeme udržovat několik certifikačních autorit. Vytvoříme tedy adresář a přepneme se do něj.

```
# mkdir -p ~/firma/ca
# cd ~/firma/ca
```

Nyní potřebujeme konfigurační soubor pro openssl. Protože si tento budeme upravovat pro každou certifikační autoritu, nakopírujeme si ukázkový soubor do právě vytvořeného adresáře. Jako ukázkový je použit ten standardně instalovaný s balíčkem openssl jenž se nachází v /etc/ssl/openssl.cnf.

```
# cp /etc/ssl/openssl.cnf .
```

Konfigurační soubor si upravíme. Nejdříve nastavíme adresář ve kterém budou všechny soubory. To bude ~/firma/ca/ca. Parametr, jímž tak činíme, se jmenuje dir a nachází se v sekci [CA_default].

```
[ CA_default ]
dir = ./ca
```

Tato změna úplně stačí pro funkčnost. Ale je lépe podívat se i na několik dalších parametrů a nastavit si je přesně podle potřeb konkrétní certifikační autority. Jedná se v první řadě o parametry v sekci [req_distinguished_name], které popisují komu je certifikát vydáván. Protože předpokládáme, že certifikáty vydáváme pro vlastní potřebu, nebo potřebu firmy, ve které pracujeme, má smysl tyto předvyplnit.

```
[ req_distinguished_name ]
countryName_default = CZ s
tateOrProvinceName_default = .
localityName_default = Ostrava
 0.organizationName_default = VSB
organizationUnitName_default = KAT440
```

alší nastavení, které se považuje za užitečné je změna politiky v souvislosti s parametrem stateOrProvinceName. Tento je totiž vyžadován, ale v našem prostředí nemá smysl. Proto jej nastavíme na optional. Parametr se nachází v sekci [policy_match].

```
[ policy_match ]
stateOrProvinceName = optional
```

Z řady parametrů ještě uvedeme jeden, a to default_days, který určuje počet dní na které je certifikát vydávan. Tedy počet dní ode dneška po které je certifikát platný. Po vypršení této doby je nutné certifikát prodloužit. Tento parametr se nachází v sekci [CA_defaults].

```
default_days=3600
```

Můžeme dle vlastní potřeby modifikovat i další parametry, jak na to je popsáno v dokumentaci k programu OpenSSL. Máme vytvořen a k vlastnímu obrazu upraven konfigurační soubor a nyní si vytvoříme potřebné adresáře. Je to jednak samotný adresář ca ve kterém jsou všechny soubory certifikační autority, tak i jeho podadresáře.

```
# mkdir ca
# mkdir ca/newcerts
# mkdir ca/private
 # chmod 0700 ca/private
```

Dále si vytvoříme index certifikátů a inicializujeme jejich počitadlo.

```
# touch ca/index.txt
 # echo 01 >ca/serial
```

Rovněž nastavíme přístupová práva, aby se k naší certifikační autoritě nedostal nikdo jiný. Znalost klíče by mu totiž umožnila vydávat falešné certifikáty.

```
# chmod 0700 ~/firma/ca
# chmod 0700 ca
```

Máme vytvořeny adresáře a vytvoříme samotnou certifikační autoritu.

```
# cd ~/firma/ca
# openssl req -config openssl.cnf -new -x509 -out cacert.pem –keyout cakey.pem -days 7500 -newkey rsa:4096
```

Při vytváření klíče certifikační autority jsme vyzváni k zadání přístupové fráze k tomuto klíči. Pomocí fráze je samotný klíč zašifrován, takže dostane-li se k němu nepovolaná osoba, nemůže jej bez znalosti fráze použít. Tuto frázi je ovšem nutno zadávat kdykoliv se klíč použije, tedy při podepisování každé žádosti o certifikát. Jsme-li si jistí, že se k našemu klíči nikodo nedostane, a potřebujeme-li podepisování automatizovat, můžeme zašifrování klíče pomocí přístupové fráze vypnout přepínačem -nodes. Protože program OpenSSL očekává konfigurační soubor na standardních místech jako je /etc/ssl/openssl.cnf, musíme mu oznámit že má použít náše umístění. To učiníme parametrem -config openssl.cnf. Dalším parametrem, na který upozorníme je -days 7500. Ten určuje dobu trvání certifikátu, tedy do kdy platí, v počtu dní ode dneška. V případě certifikátu certifikační autority zvolíme dostatečně velkou hodnotu. V uvedeném případě je to cca 20 let. A poslední parametr který zmíníme je -newkey rsa:4096. Ten určuje délku rsa klíče v bitech. Pro dnešní dobu je doporučované minimum 2048. Protože se jedná o klíč certifikační autoritiy, použijieme raději klíč delší a to 4096 bitů dlouhý. Vytvořené soubory umístníme na místo, kde jsou očekávany, a nastavíme jim přístupová práva.

```
# mv cacert.pem ca/ # certifikát autority, veřejný soubor
# chmod 0400 cakey.pem # soukromý klíč autority, je neveřejný
# mv cakey.pem ca/private # a je očekáván v tomto adresáři
```

Tímto máme certifikační autoritu připravenu k použití a k podepisování žádostí o certifikáty.

Vytvoření certifikátu podepsaného naší certifikační autoritou:

Vytvoříme žádost o klíč:

openssl req -config openssl.cnf -new -des3 -out request.pem –keyout key.pem -days 1098 -newkey rsa:2048

který necháme podepsat certifikační autoritě:

openssl ca -config openssl.cnf -in request.pem -out cert.pem

Pokud potřebujeme vytvořit certifikát pro www server, zadáme při vytváření žádosti do pole Common Name adresu tohoto serveru. Například *www.vsb.cz*.

OpenVPN

OpenVPN je jednoduše použitelná, robusní a velmi konfigurovatelná implementace VPN, umožňující bezpečně propojit dva počítače v lokálních sítích skrze veřejnoiu síť (Internet) nebo i více sítí mezi sebou. Komunikace přes OpenVPN je šifrována pomocí SSL, nejedná se tudíž o klasické schéma VPN s využitím IPsec, tak, jak je známé. Projekt OpenVPN je Open-source a je šířen pod licencí GNU/GPL. Popis instalace a nastavení OpenVPN je v této ukázce popsán vzhledem k OS Linux Debian, ale podporované operační systémy jsou i MS Windows 2000/XP, MacOS X, OpenBSD, FreeBSD NetBSD a také Solaris. Kompletní výpis vlastností OpenVPN je k nalezení na stránkách projektu.

Instalace

V Linux distribuci Debian je nástroj OpenVPN součástí balíčků repozitáře proto instalace probíhá jednoduchým příkazem:

aptitude install openvpn

Tunel bez zabezpečení:

Tento tunel není nijak zabezpečen, komunikace probíhá protokolem UDP na portu 5000. Do příkazového řádku definujeme nejdříve pomocí příkazu —remote vzdálený po- čítač pomocí domény či jeho IP adresy, poté pomocí –dev rozhraní a pomocí —ifconfig nastavíme ip adresy rozhraní. Příkaz –verb nastavuje podrobnost výpisů. Co se týče rozhraní, rozlišuje OpenVPN dvě – TUN a TAP. Rozdíl mezi těmito rozhraními je ten, že TAP rozhraní funguje jako bridge, tzn. Na druhé vrstvě OSI/ISO modelu. TUN rozhraní se poté používá pro routování, tzn. funguje na třetí IP vrstvě OSI/ISO modelu. Lomítkov zápisu znamená zalomení na další řádek kvůli velikosti stránky. Ve skutečnosti je zápis na jednom řádku.

Na počítači 1 spustíme:

openvpn --remote pocitac2.cz --dev tun1 \ --ifconfig 192.168.2.1 192.168.2.2 --verb 9

Na počítači 2 spustíme:

openvpn --remote pocitac1.cz --dev tun1 \ --ifconfig 192.168.2.2 192.168.2.1 --verb 9

Funkčnost otestujeme pomocí programu ping na druhou stranu tunelu:

pocitac 1: ping 192.168.2.2
pocitac 2: ping 192.168.2.1

Tunel zabezpečený pevným klíčem (symetrické šifrování):

Nejdřív si vygenerujeme klíč, kterým zabezpečíme spojení:

openvpn --genkey --secret klic.key

Klíč musíme mít na obou počítačích, na kterých končí tunel. Přenos klíče by měl proběhnout nějakým bezpečným způsobem.

První počítač:

openvpn --remote pocitac2.cz --dev tun1 \ --ifconfig 192.168.2.1 192.168.2.2 --verb 5 --secret klic.key

Druhý počítač:

openvpn --remote pocitac2.cz --dev tun1 \ --ifconfig 192.168.2.1 192.168.2.2 --verb 5 --secret klic.key

Tunel zabezpečený soukromým a veřejným klíčem s použitím TLS/SSL (asymetrické šifrování):

Předchozí varianty jsou popsány abychom ukázali, že jsou možné a funkční, ale je těžké pro ně najít vhodné využití. Proto se pro bezpečné připojení z libovolného místa častěji používá TLS/SSL.

Pro to abychom mohli používat následující nastavení potřebujeme SSL certifikáty a klíče podepsané od jedné CA. Balíček OpenVPN obsahuje skripty s jednoduchou CA (adresář easyrsa/), kterou použijeme. Pro vytvoření CA, klíčů a certifikátů je také možné využít OpenSSL knihovnu podle příkladů uvedených výše v textu. V adresáři easy-rsa/ najdeme soubor vars, který si musíme upravit v závislosti na tom, kam jsme si easyrsa/ nakopírovali. Doporučuje se, aby jsme si přečetli README, které se nachází v tomto adresáři, v README se totiž nachází kompletní popis výroby klíče. Všechny soubory, které zde budou popsány, jsou v adresáři /etc/openvpn, který má práva 700 a majitele uživatele root.

SSL certifikáty – Server

Na server si umístíme certifikát CA (ca.crt), certifikát a klíč serveru (server.crt a server.key), které jsme si vygenerovali podle README v adresáři easyrsa/, také bychom neměli zapomenout na soubor dh1024.pem nebo dh2048.pem, záleží na tom, který jsme si vyrobili.

server.conf - Tento soubor obsahuje nastavení OpenVPN na straně serveru (pokud se podíváme do dokumentace k OpenVPN, zjistíme, co která položka znamená).:

```
dev tun
ifconfig 192.168.2.1 192.168.2.2
tls-server
dh dh2048.pem
#nebo dh dh1024.pem
ca ca.crt
cert server.crt
key server.key
 port 5000
comp-lzo
persist-tun
persist-key
verb 3
```

Parametry ifconfigu jsou ifconfig , jsou to koncové adresy tunelu. Server běží na UDP portu 5000.

SSL certifikáty – Klient

Na klientský počítač umístíme certifikát CA (ca.crt), certifikát a klíč klienta (klient.crt a klient.key), které jsme si vygenerovali podle README v adresáři easy-rsa/.

klient.conf - Tento soubor obsahuje nastavení OpenVPN na straně serveru (pokud se podíváme do dokumentace k OpenVPN, zjistíme, co která položka znamená).:

```
dev tun
remote 147.32.119.186 # VPN server
ifconfig 192.168.2.2 192.168.2.1
tls-client
ca ca.crt
cert klient.crt
key klient.key
port 5000
comp-lzo
persist-tun
persist-key
verb 3
```

Jakmile máme nastaveny konfigurační soubory, můžeme VPN tunel spustit. Nejdříve je nutné spustit VPN tunel na straně server:

```
# openvpn server.conf
```

Měla by proběhnout inicializační sekvence, podobná následujícímu příkladu:

Sun Feb 6 20:46:38 2010 OpenVPN 2.0_rc12 i686-debian-linux [SSL] [LZO] [EPOLL] built on Feb 5 2010
Sun Feb 6 20:46:38 2010 Diffie-Hellman initialized with 2048 bit key
Sun Feb 6 20:46:38 2010 TLS-Auth MTU parms [L:1542 D:138 EF:38 EB:0 ET:0 EL:0]
Sun Feb 6 20:46:38 2010 TUN/TAP device tun1 opened
Sun Feb 6 20:46:38 2010 /sbin/ifconfig tun1 192.168.2.1 pointopoint 192.168.2.2 mtu 1500
Sun Feb 6 20:46:38 2010 Data Channel MTU parms[L:1542 D:1450 EF:42 EB:23 ET:0 EL:0 AF:3/1]
Sun Feb 6 20:46:38 2010 UDPv4 link local (bound): [undef]:1194 Sun Feb 6 20:46:38 2010 UDPv4 link remote: [undef]
Sun Feb 6 20:46:38 2010 MULTI: multi_init called, r=256 v=256
Sun Feb 6 20:46:38 2010 IFCONFIG POOL: base=10.8.0.4 size=62
Sun Feb 6 20:46:38 2010 IFCONFIG POOL LIST
Sun Feb 6 20:46:38 2010 Initialization Sequence Completedconf

Poté spustíme VPN tunel in a straně klienta, který by měl obě strany relace propojit:

openvpn klient.conf

Zda je tunel kompletní a funknčí ověříme pomocí programu ping na adresu virtuáolního rozhraní TUN.

klient: ping 192.168.2.1