

Open source řešení SSL virtuálních privátních sítí

Open Source Solution of SSL Virtual Private Networks

Zadání bakalářské práce

Student: **Jakub Martiník**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2601R013 Telekomunikační technika

Téma: **Open source řešení SSL virtuálních privátních sítí**
Open Source Solution of SSL Virtual Private Networks

Zásady pro vypracování:

Cílem bakalářské práce je návrh, realizace a otestování různých řešení SSL virtuálních privátních sítí s využitím open source softwaru OpenVPN.

Osnova práce:

1. Popište software OpenVPN.
2. Navrhněte a v laboratorních podmínkách realizujte virtuální privátní síť s využitím OpenVPN.
3. Ověřte možnosti použití certifikátů X.509.
4. Ověřte možnosti použití OpenVPN v mobilních telefonech.

Seznam doporučené odborné literatury:

FAILNER, Markus, GRAF, Norbert. Beginning OpenVPN 2.0.9. Birmingham: Packt publishing, 2009. ISBN 978-1-847197-06-1.


KEIJSER, Jan Just. OpenVPN 2 Cookbook. Birmingham: Packt publishing, 2011. ISBN 978-1-84951-010-3.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **Ing. Petr Machník, Ph.D.**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013

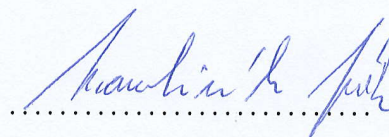

prof. RNDr. Vladimír Vašínek, CSc.
vedoucí katedry




prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava*.

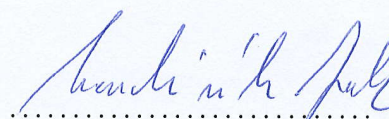
V Ostravě 22. dubna 2013



.....

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 22. dubna 2013



.....

Děkuji Ing. Petru Machníkovi, Ph.D., nejen za trpělivé vedení bakalářské práce, ale také za rady, podněty a připomínky, jež byly pro vznik práce velice přínosné.

Abstrakt

Bakalářská práce se zabývá bezplatným řešením SSL virtuálních privátních sítí a jejich realizaci pomocí softwaru spadající pod licenci open source. Zájemcům o výše uvedené téma nabízí informace o softwaru OpenVPN, historii, vývoji a možnostech jeho konfigurací s využitím různých metod zabezpečení přenášených dat. Součástí práce je uveden postup pro konfiguraci softwaru OpenVPN verze 2.2 se sílenými klíči a OpenVPN verze 2.2 s využitím SSL certifikátů. Uveden je také postup konfigurace mobilních zařízení fungující na operačních systémech Apple iOS 5.0 nebo vyšší a Android 4.0 nebo vyšší.

Klíčová slova: OpenVPN, certifikační autorita, spojení, tunel, virtuální rozhraní, síť

Abstract

The bachelors thesis deals with solution of SSL private virtual networks which is free of charge and its realization on software which goes to open source licence. The goal of this thesis is to offer an information about history, developement and possibilities of configuration OpenVPN system and possibilities of using different methods for protection of transmitted data, mainly for clients who are interested in this subject. The second part of the thesis is a proper process of configuration of OpenVPN version 2.2 software with pre-shared keys and OpenVPN verison 2.2 with SSL certificates. There is also a manual for mobile devices working on Apple iOS 5.0 or higher and Android 4.0 or higher.

Keywords: OpenVPN, certification Authority, connection, tunnel, virtual interface, network

Seznam použitých zkratk a symbolů

3DES	– Triple Encryption Standart
AES	– Advanced Encryption Standard
CA	– Certificate Authority
DHCP	– Dynamic Host Configuration Protocol
DynDNS	– Dynamic Domain Name System
FTP	– File Transfer Protocol
HMAC	– Hash-based Message Authentication Code
HTTP	– Hypertext Transfer Protocol
IETF	– Internet Engineering Task Force
IP	– Internet Protocol
IPsec	– Internet Protocol Security
ISO/OSI	– Open Systems Interconnection
ITU-T	– International Telecommunication Union
MD5	– Message-Digest Algorithm
NAT	– Network Address Translation
SHA	– Secure Hash Algorithm
SMTP	– Simple Mail Transfer Protocol
SSL	– Secure Sockets Layer
TAP	– Virtual interface
TCP	– Transmission Control Protocol
TLS	– Transport Layer Security
TUN	– Virtual interface
UDP	– User Datagram Protocol

Obsah

1	Úvod	4
2	Virtuální privátní síť	5
2.1	Virtuální privátní síť	5
2.2	Zabezpečení VPN	7
3	OpenVPN	9
3.1	Rozhraní TUN/TAP	9
3.2	Výhody software OpenVPN	9
3.3	Vývoj OpenVPN	12
3.4	Srovnání technologií IPsec a OpenVPN	14
4	Instalace a konfigurace OpenVPN	15
4.1	Instalace a konfigurace OpenVPN se sdílenými klíči	15
4.2	Návod pro konfiguraci OpenVPN se sdílenými klíči	15
4.3	Instalace a konfigurace OpenVPN s využitím SSL/TLS certifikátů	17
4.4	Návod pro konfiguraci OpenVPN s využitím SSL/TLS certifikátů	17
4.5	Konfigurace mobilních zařízení s operačními systémy Apple iOS a Android s využitím SSL/TLS	23
5	Závěr	29
6	Reference	30
	Přílohy	30
A	Konfigurační soubor pro server a klienty s využitím sdílených klíčů	31
A.1	Konfigurační soubor pro server <code>server.conf</code>	31
A.2	Konfigurační soubor pro klienty <code>client.conf</code>	33
B	Konfigurační soubor pro server a klienty s využitím SSL certifikátů	36
B.1	Konfigurační soubor pro server <code>server.conf</code>	36
B.2	Konfigurační soubor pro klienty <code>client.conf</code>	44

Seznam tabulek

1	Vývoj OpenVPN verze 1	12
2	Vývoj OpenVPN verze 2	13
3	Srovnání IPsec a OpenVPN	14
4	Charakteristika vygenerovaných souborů	22

Seznam obrázků

1	Ukázka propojení vzdálených lokálních sítí	6
2	Šifrovaný VPN tunel	6
3	Virtuální rozhraní TUN/TAP	10
4	OpenVPN se sdílenými klíči	15
5	OpenVPN s využitím SSL certifikátů	17
6	Záložka Subject a generování RSA klíče	20
7	Záložka Extensions	20
8	Úspěšné vygenerování certifikační autority	20
9	Výběr CA pro podpis certifikátů	21
10	Generování RSA klíče	21
11	Seznam certifikátů	21
12	Seznam klíčů	21
13	OpenVPN pro mobilní zařízení	24
14	Aplikace OpenVPN	24
15	Spuštěná aplikace	24
16	Nahrání souborů do aplikace OpenVPN	25
17	Spuštěná aplikace po nahrání souborů	26
18	Úspěšné nahrání konfigurace a certifikátů	26
19	Úspěšné připojení k OpenVPN serveru	26
20	Aplikace OpenVPN for Android	27
21	Umístění souborů do paměti telefonu	27
22	Spuštěná aplikace	28
23	Konfigurace certifikátů	28
24	Připojení k serveru	28

1 Úvod

Tato bakalářská práce se zabývá virtuálními privátními sítěmi a jejich konfiguracemi. Teoretická část práce popisuje základní vlastnosti virtuálních privátních sítí a jejich zabezpečení. Virtuální privátní síť slouží například k logickému propojení dvou firemních poboček, které jsou od sebe vzdáleny tak, že není možné realizovat fyzické propojení těchto poboček. Samozřejmě je kladen důraz na bezpečnost celé komunikace a přenášených dat. Popsán je software OpenVPN, historie vývoje a výhody softwaru. Software je zdarma dostupný pod licencí open source, a to pro většinu platforem.

Téma jsem zvolil zejména pro svůj zájem o počítačové sítě. Díky využití softwaru OpenVPN, který je vydáván pod licencí open source jsou potřebné náklady na realizaci virtuální privátní sítě minimální. Software OpenVPN spravuje obsáhlá komunita lidí, která ho neustále vylepšuje a inovuje.

Součástí práce je popis přesného postupu konfigurace VPN serveru i klientů s využitím OpenVPN software a různých zabezpečovacích mechanismů pro osobní počítače i mobilní zařízení. Úkolem práce je nastavit software OpenVPN se sdílenými klíči, kdy server i klient využívají stejný klíč pro šifrování a dešifrování, dále konfigurace OpenVPN s využitím SSL/TLS certifikátů, kde se využívá dvojice klíčů. V závěru se práce věnuje konfiguraci mobilních zařízení pro připojení k OpenVPN serveru.

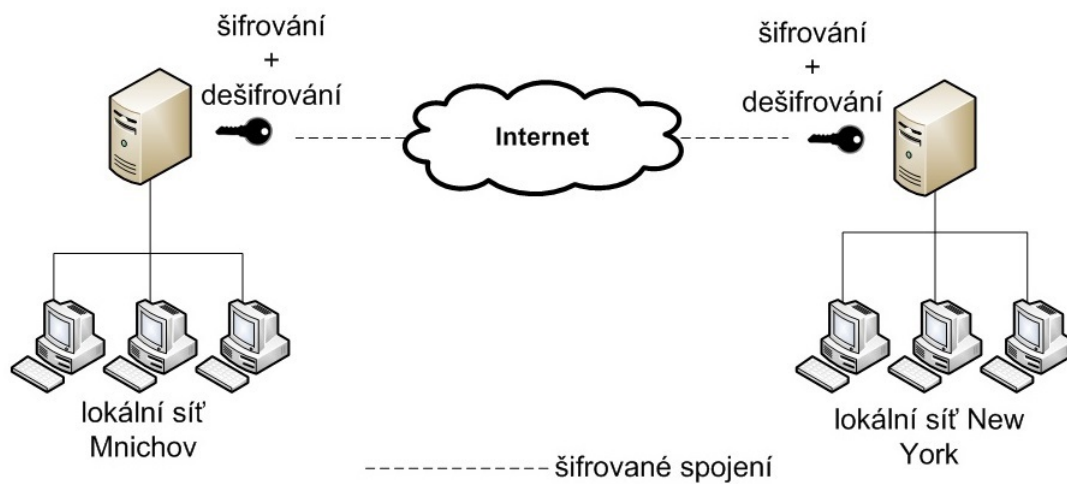
2 Virtuální privátní síť

2.1 Virtuální privátní síť

V roce 1990, v době vzestupu internetu, začaly zlevňovat internetové přípojky a začala oproti dosud používanému vytáčenému připojení narůstat rychlost internetu. Zrychlení internetových přípojek vedla k mnoha inovacím, jednou z nich jsou také Virtuální privátní síť. Virtuální privátní síť (VPN) je možnost přístupu do privátních lokálních sítí, ať jsme kdekoli na zeměkouli. VPN funguje prostřednictvím internetu, který představuje nezabezpečenou veřejnou síť, a proto nabízí šifrování přenášených dat, kterým se zamezí zneužití přenášených dat. Představme si, že máme firmu, která má více poboček po celém světě. Historicky tyto firmy mohly sdílet svá data pouze prostřednictvím pošty, telefonů později také pomocí faxu. Hlavní body vývoje virtuálních privátních sítí jsou tyto.[?]

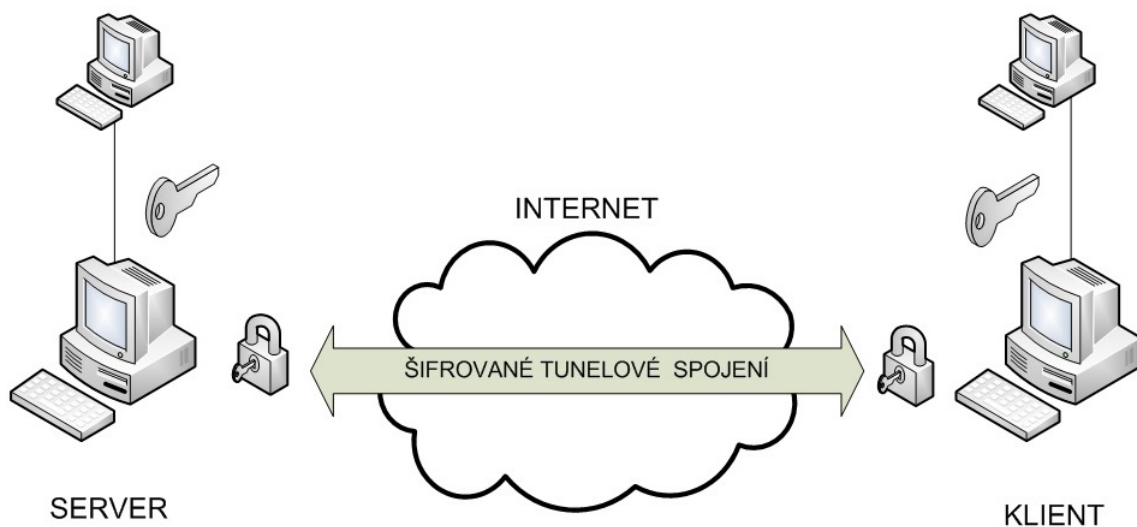
- Požadavky na flexibilitu a rychlou výměnu informací mezi pobočkami firem, umístěných kdekoli na světě.
- Zvyšování produktivity firem a spolupráci všech zaměstnanců, kteří pracují na různých pobočkách.
- Umožnění práce zaměstnancům odkudkoli je to možné, například z domu. Veškerá tato komunikace by měla být bez zbytečných zpoždění.
- Nbaídka bezpečné komunikace a zabránění odposlechu přenášených dat na internetu díky šifrovanému spojení.
- Implementace do mobilních zařízení jako jsou notebooky a chytré telefony pro mobilitu koncových uživatelů.

Díky propojení firemních poboček prostřednictvím virtuální privátní sítě, vzniká logické propojení mezi těmito pobočkami. Díky tomuto propojení mohou pobočky téměř okamžitě sdílet svá data. Toto propojení kompletně šifrováno. Topologie je zobrazena na obrázku 1



Obrázek 1: Ukázka propojení vzdálených lokálních sítí

VPN funguje na bázi klient-server, to znamená, že na jedné pobočce běží VPN server a na druhé VPN klient. Klienti se připojují na server a vzniká šifrované spojení, kterým proudí data. Tomuto spojení se říká tunel (Obrázek: 2). Díky tomuto tunelu proudí data vždy z jednoho konce na druhý, (podobně jako automobily, které nemohou z tunelu odbočit jiným směrem a musí jej opustit vždy až na druhém konci).



Obrázek 2: Šifrovaný VPN tunel

2.2 Zabezpečení VPN

2.2.1 IPsec

V roce 1995 byl IPsec vyvinut jako bezpečnostní rozšíření pro Internet Protocol (IP), který funguje na třetí vrstvě ISO/OSI referenčního modelu a byl standardizován Internet Engineering Task Force (IETF). IPsec je možné využít pro šifrování jakéhokoli protokolu běžícího na čtvrté až sedmé vrstvě ISO/OSI referenčního modelu. Například protokoly sedmé vrstvy (HTTP, SMTP, FTP), které jsou na třetí vrstvě ISO/OSI modelu šifrovány pomocí protokolu IPsec.

IPsec je protokolem třetí vrstvy ISO/OSI modelu. Tím vzniká problém, protože protokol IPsec není schopen šifrovat protokoly nižších vrstev, například Frame-Relay sítě nebo ethernet rámce.

Pro zaručení integrity (celistvosti) IP paketů využívá IPsec HMAC (Hash Message Authentication Code), aby tyto kódy získal, využívá k tomu hashování funkce MD5 a SHA na základě tajného klíče a obsahu paketu. HMAC je potom vložen do hlavičky IPsec protokolu a příjemce, pokud zná tajný klíč je schopen kontrolovat, zda nebyl paket změněn při cestě.

pro důvěryhodnost IP paketu používá IPsec symetrické šifrování. K tomu jsou dnes nejvíce používány algoritmy jako 3DES, AES a Blowfish.

2.2.2 SSL/TLS

Specifikace SSL byla vytvořena v 90. letech společností Netscape. V roce 2001 došlo k odkoupení patentu organizací IETF. Ten byl upraven a přejmenován na TLS (Transport Layer Security). SSL/TLS je dnes nejčastěji používáno pro autentizaci a šifrování. SSL/TLS je částí OpenSSL knihovny, která je přítomná na veškerých moderních operačních systémech. SSL/TLS je dnes používána například pro bezpečnou komunikaci s internetovým bankovníctvím, e-mailovými servery nebo pro bezpečné odesílání přístupových údajů.

OpenSSL je zcela zdarma přístupná, kryptografická knihovna, vedená pod licencí open source. Knihovna v sobě implementuje podporu SSL/TLS certifikátů X.509, hashovací funkce i symetrické a asymetrické šifrování.[1][4]

2.2.3 Certifikáty X.509

Standard ITU-T X.509, je součástí doporučení X.500. Ta definují adresářové služby. Norma X.509 pak definuje služby pro autentizaci uživatelů, kteří přistupují do adresářů a obsahuje i definice a formáty certifikátů, které se používají.

Certifikační autorita(CA) se stará o přidělování a správu certifikátů pro uživatele, například o generování veřejných nebo privátních certifikátů a jejich podepisování. Je totiž nepraktické, aby jedna CA obsluhovala všechny uživatele. Proto má každá organizační jednotka svou vlastní CA. Uživatelé patřící do této CA znají veřejný klíč své CA, a proto mohou ověřit, zda certifikát, který obdrželi, patří jejich CA.

2.2.3.1 Struktura X.509 certifikátu obsahuje:

- Číslo verze (version)
- Sériové číslo (serialNumber)
- ID podpisového algoritmu (signature)
- Vydavatel (issuer)
- Platnost (validity)
 - Od (notBefore)
 - Do (notAfter)
- Subjekt (subject)
- Informace o veřejném klíči subjektu (subjectPublicKeyInfo)
 - Algoritmus veřejného klíče subjektu (algorithm)
 - Veřejný klíč subjektu (subjectPublicKey)
- Jednoznačnou identifikaci vydavatele (issuerUniqueID)
- Jednoznačnou identifikaci subjektu (subjectUniqueID)
- Rozšíření certifikátu (extensions)
- ID podpisového algoritmu certifikátu (signatureAlgorithm)
- Digitální podpis certifikátu (signatureValue)

3 OpenVPN

OpenVPN je software, který vyvinul James Yonan v roce 2001, je stále vylepšován. Vytvořené spojení mělo velice slabé zabezpečení, které udalo hlavní směr vývoje softwaru OpenVPN. Ten od začátku kladl důraz na jeho bezpečnost a použitelnost. Žádné jiné VPN řešení nenabízí takové možnosti kombinací zabezpečení a spolehlivosti výhod.

James Yonan využil síťové virtuální rozhraní TUN/TAP, které je obsaženo v linuxovém jádru zvaném kernel. Využitím TUN/TAP virtuálních zařízení poskytl flexibilitu, jakou jiné VPN řešení dosud nenabízelo. Pomocí TUN/TAP virtuálního síťového rozhraní poté probíhala veškerá síťová komunikace. Jiné VPN řešení využívající SSL/TLS bylo nutné i nadále připojovat pomocí internetových prohlížečů

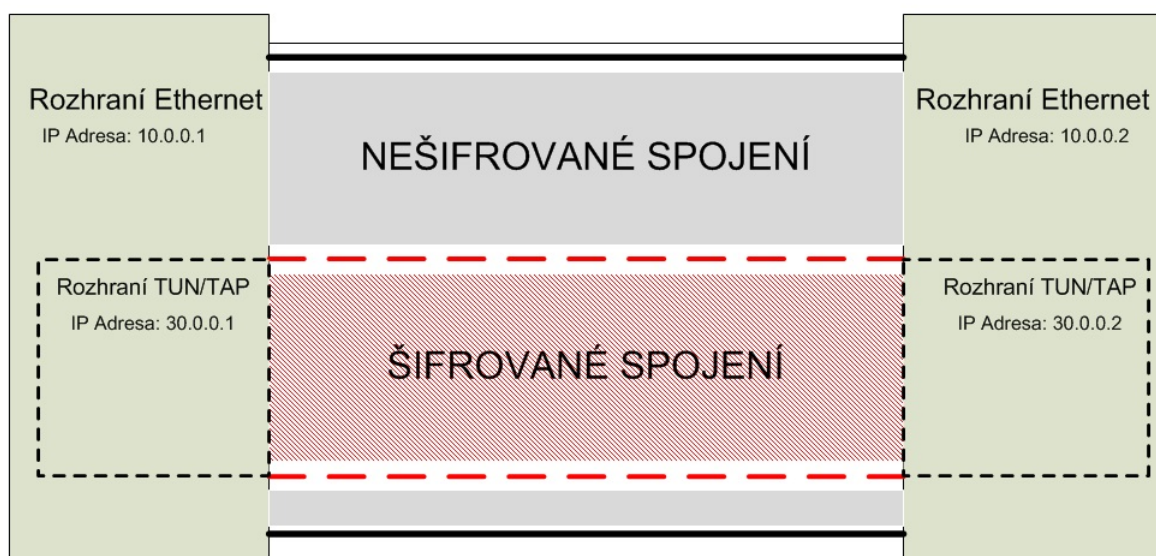
3.1 Rozhraní TUN/TAP

Rozhraní TUN/TAP je obsaženo ve všech novějších distribucích Linux/UNIX systémů a také Windows a Mac OS X. Využitím TUN/TAP rozhraní se značně zjednosuše celé zabezpečení přenosu. Virtuální TUN/TAP rozhraní se skrývají za fyzickými rozhraními. Díky tomuto řešení je možné rozlišit, který provoz bude šifrován a který bude posílán nešifrovaně. Například provoz, který bude směřován na firemní server budeme směřovat na virtuální rozhraní a celou komunikaci budeme šifrovat. Pokud si však budeme prohlížet veřejný obsah internetu, jako jsou například zpravodajské servery, není nutné tuto komunikaci šifrovat a budeme ji směřovat na fyzické rozhraní. Při konfiguraci je nutné dodržet, aby na obou stranách byl nakonfigurován stejný typ virtuálního rozhraní. Rozhraní není možné kombinovat z důvodu fungování na rozdílných vrstvách referenčního modelu ISO/OSI.

- Rozhraní TAP pracuje na druhé vrstvě referenčního modelu ISO/OSI. Fungování na druhé vrstvě nám umožní využít i jiné přenosové protokoly než je IP, např. Frame-Relay, PPP.
- Rozhraní TUN pracuje na třetí vrstvě referenčního modelu ISO/OSI. Tímto rozhraním není možné přenášet jiný než IP provoz.

3.2 Výhody software OpenVPN

S rozvojem VPN řešení přišly také problémy s kompatibilitou. VPN řešení různých společností využívaly také různé bezpečnostní mechanismy pro sestavování šifrovaného



Obrázek 3: Virtuální rozhraní TUN/TAP

spojení, které mezi sebou nebyly vždy kompatibilní, mnohdy ani nebyly standardizovány. Nebylo tedy možné navázat spojení s protější stranou, což vedlo ke značným problémům v komunikaci. OpenVPN využívá k zabezpečení spojení a přenosu dat stabilní SSL/TLS mechanismus, který je standardizovaný, a tudíž není tak složitý jako například IPsec.

3.2.1 Hlavní výhody OpenVPN jsou tyto:

1. OpenVPN nabízí možnosti implementace do druhé nebo třetí vrstvy ISO/OSI. VPN tunel na druhé vrstvě umožňuje posílat a šifrovat například ethernetové rámce, Frame-Relay apod. Tímto nabízí řešení pro sítě, které nefungují na protokolu IP. Komunikace na druhé vrstvě je ve většině jiných řešení problémová ne-li nemožná.
2. Pokud se klient připojí pomocí OpenVPN na lokální síť firmy, automaticky je veškerá komunikace chráněna a spadá pod pravidla firewallu, který se nachází na hranici této lokální sítě a internetu. Tímto řešením se poskytuje klientovi větší bezpečnost při práci na internetu, protože veškerá jeho komunikace je přenášena přes šifrované spojení a navíc chráněna pravidly firewallu.
3. OpenVPN řešení je podporováno pro navázání spojení téměř přes všechny firewally a proxy servery a to z důvodu využití jednoho portu pro komunikaci se serverem.

4. OpenVPN podporuje komunikaci typu klient – server. Server čeká na příchozí spojení od klientů, pokud proběhne autentizace klienta v pořádku, server sestaví šifrované spojení a klient komunikuje pouze přes šifrované spojení. Tímto je komunikace v obou směrech šifrována.
5. Pro připojení na OpenVPN server je nutné mít na firewallu povolen pouze jeden port, tímto portem se může připojit několik klientů.
6. OpenVPN je možné konfigurovat jak pro TCP spojení, tak UDP spojení. Tyto protokoly čtvrté vrstvy ISO/OSI modelu jsou podporovány.
7. OpenVPN podporuje protokol NAT, a to oběma směry. Server tedy může mít přidělenou privátní adresu, která není veřejně známa. Tímto se zvyšuje bezpečnost proti útokům na server, útočník není schopen bez velkého úsilí odhalit IP adresu serveru.
8. OpenVPN nabízí jednoduchou instalaci jak serveru, tak klienta na většinu dostupných operačních systémů. Konfigurace probíhá pomocí jednoduchých textových konfiguračních souborů.
9. OpenVPN také nabízí rozšíření pro používání skriptů vytvořených klientem nebo správcem OpenVPN serveru. Toto rozšíření nám umožňuje spouštět skripty například pro určitou skupinu klientů, kterým tak můžeme buď definovat nebo omezit přístup k určitým firemním serverům.
10. OpenVPN také umožňuje individuální nastavení pro skupiny klientů, které se připojují. Při připojení klienta na server, se klient zařadí do skupiny, která se přiřadí na základě informací v klientském certifikátu v proměně CN=. Tímto je možné rozlišit, zda se jedná o administrátora sítě nebo administrativního pracovníka.
11. OpenVPN umožňuje na základě informací v klientských certifikátech uplatňovat různá pravidla pro filtrování provozu na firemním firewallu.
12. Podpora na všech typech operačních systémů, Windows, UNIX, Mac OS X, jak v módu server, tak módu klient.
13. Podpora pro platformy mobilních telefonů s operačními systémy Windows mobile, Android, Apple iOS a Maemo.

3.3 Vývoj OpenVPN

3.3.1 OpenVPN verze 1

OpenVPN vstoupilo na scénu v 13. května 2001 s verzí označovanou 0.90. Tato verze podporovala UDP komunikaci, která mohla být šifrována pouze pomocí Blowfish šifry. Verze 0.91, která byla vydána o sedm měsíců později, už obsahovala více šifrovacích mechanismů. Verze 1.0 vydaná v březnu roku 2002 poskytovala základní SSL/TLS autentizaci a výměnu šifrovacích klíčů. Podstatné změny ve vývoji OpenVPN verze 1 je zobrazeny v tabulce 1.

Datum	Verze	Podstatné změny
květen 2001	0.90	První vydaná verze, která podporovala pouze protokol UDP a jediný šifrovací mechanismus
prosinec 2001	0.91	Přidáno více šifrovacích mechanismů
březen 2002	1.0	Přidána základní autentizace pomocí TLS mechanismu
duben 2002	1.1.0	Rozšíření podpory TLS/SSL, zvýšení bezpečnosti, vylepšená manuálová stránka
květen 2002	1.2.0	Rozšíření o textové konfigurační soubory, SSL/TLS mechanismus obsahoval delší klíče
září 2002	1.3.2	Přidán balík easy-rsa, který obsahuje skripty pro generování SSL certifikátů, podpora IPv6 u virtuálního rozhraní TUN
květen 2003	1.4.1	Přidána podpora pro UNIX jádro kernel verze 2.4
listopad 2003	1.5.0	Podpora TCP, podpora proxy serverů a rozšíření routovacích funkcí

Tabulka 1: Vývoj OpenVPN verze 1

3.3.2 OpenVPN verze 2

OpenVPN verze 2 se začala vyvíjet paralelně s OpenVPN verzí 1, a to v listopadu roku 2003. Vyšlo 29 testovacích verzí, poté 20 beta verzí a 21 verzí pro vydání. V dubnu 2005 vyšla oficiální verze OpenVPN verze 2.0. Největší rozdíly oproti verzi OpenVPN verze 1 jsou tyto.

- Podpora připojení více klientů na jeden server. Klienti využívají TLS autentizaci a IP adresy pro komunikaci jim přiděluje OpenVPN server. Je možné připojit až 128 klientů pomocí stejného TCP nebo UDP portu.
- Síťové nastavení klientů je v kompletní režii OpenVPN serveru. Po úspěšném vytvoření šifrovaného tunelu mohou klienti obdržet různá síťová nastavení, pomocí protokolu DHCP.
- Možnost vzdálené správy pomocí telnetu.
- Došlo ke značnému vylepšení podpory ovladačů a OpenVPN softwaru pro operační systémy Windows a 64-bit operační systémy.

Postupný vývoj OpenVPN verze 2 je zobrazen v tabulce 2

Datum	Verze	Podstatné změny
červen 2005	2.0.1-rc3	Přidána funkce přidělování domén a DNS serverů pomocí protokolů DHCP
srpen 2005	2.0.1	Zvýšená obrana proti útokům na server
listopad 2005	2.1.beta7	Cesty k certifikátům a klíčům se nyní zadávají v textovém konfiguračním souboru.
únor 2007	2.1-rc4	Vylepšená podpora pro 64-bit operační systémy Windows, možnost využití grafického rozhraní pro správu OpenVPN serveru
září 2008	2.1.rc3	Využití knihovny OpenSSL 0.9.8 pro balíčky v operačním systému Windows, oprava chyb pro klienty využívající operační systém Windows

Tabulka 2: Vývoj OpenVPN verze 2

Dnes se nejčastěji používají OpenVPN verze 2.2.x nebo OpenVPN verze 2.3.x. Ve vývoji již je i OpenVPN verze 3, která je ale ve fázi testování. Srovnání nejčastěji používaných řešení virtuálních privátních sítí, jako jsou IPsec a OpenVPN jsem provedl v tabulce 3, kde jsem uvedl jejich kladné i záporné vlastnosti.[1][3]

3.4 Srovnání technologií IPsec a OpenVPN

IPsec VPN	OpenVPN
+ Podporováno na většině typů zařízení	– Dostupná na standardních typech operačních systému UNIX, Windows a Mac OS X
+ Známá a rozšířená technologie	– Docela nová technologie, stále se však rozrůstající a vylepšující
+ Mnoho grafických rozhraní pro správu	– Nemá oficiální grafické rozhraní, pouze grafické rozhraní třetích stran
– Složitá konfigurace a technologie	+ Jednoduchý modulární systém a konfigurace
– Nutné nastavení několika portů na firewallu	+ Nastavení pouze jednoho portu na firewallu
– Problém s dynamickými adresami na obou stranách	+ Díky DynDNS rychlé přepojení při změně dynamických adres
– Implementace IPsec různých výrobců může být nekompatibilní	+ Využívá standardizované šifrovací metody a mechanismy SSL/TLS
	+ Traffic shaping
	+ Kompatibilita s firewalley a proxy servery
	– Bezproblémová práce s NAT (zařízení na obou stranách mohou být za NATem)

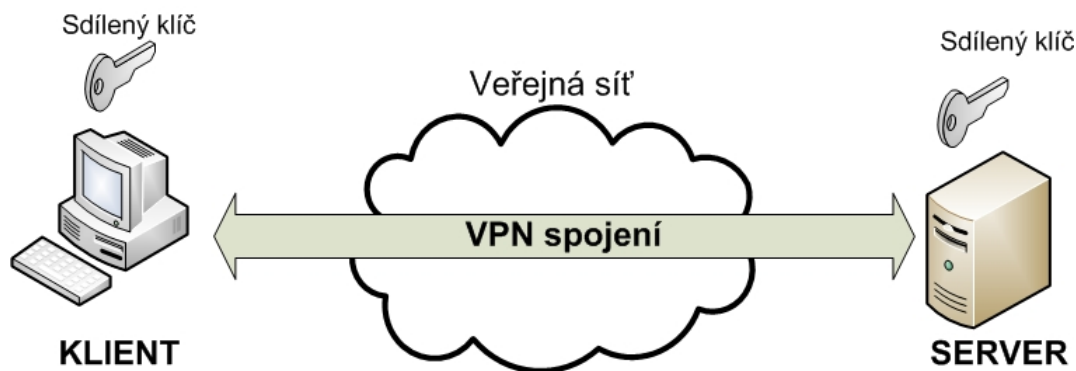
Tabulka 3: Srovnání IPsec a OpenVPN

4 Instalace a konfigurace OpenVPN

V této kapitole si ukážeme postupy pro konfiguraci OpenVPN serverů a klientů. Jako první si ukážeme konfiguraci OpenVPN se sdílenými klíči, kde server i klient využívají dvojici stejných klíčů pro šifrování a dešifrování přenášených dat. Bude následovat konfigurace týkající se OpenVPN s využitím SSL certifikátů a dále návody na konfiguraci mobilních zařízení pro připojení k OpenVPN serveru. Pro konfiguraci serveru a klientů jsem využil OpenVPN verzi 2.2.

4.1 Instalace a konfigurace OpenVPN se sdílenými klíči

Konfigurace pomocí symetrického šifrování spočívá v tom, že klient i server mají stejný šifrovací klíč. Tímto klíčem pak šifrují a zároveň dešifrují přenášená data. Toto řešení má své výhody i nevýhody. Mezi výhody patří jednoduchost šifrování i dešifrování a nižší náročnost na výpočetní výkon zařízení. Nevýhoda spočívá v bezpečnosti přenosu klíče ze serveru na klienta. Pokud bychom poslali klíč například emailem, který by útočník zachytil, mohlo by dojít k jednoduchému dešifrování přenosu. Nejbezpečnější zůstává fyzický přenos klíče na nějakém médiu, například USB flash disku. To však z pravidla nelze realizovat bez značného úsilí. Postup šifrování a dešifrování je znázorněn na obrázku.



Obrázek 4: OpenVPN se sdílenými klíči

4.2 Návod pro konfiguraci OpenVPN se sdílenými klíči

1. Nainstalujte software OpenVPN jak na klientskou stanici, tak na server. Instalaci proveďte na obou stanicích příkazem

```
apt-get install openvpn
```

2. Na stanici, která bude zastupovat úlohu serveru, vygenerujte klíč pro symetrické šifrování a dešifrování. Klíč vygenerujte uvedeným příkazem, přeneste na klient-skou stanici a vložte do složky `/etc/openvpn/`.

```
openvpn --genkey --secret /etc/openvpn/static.key
```

3. Vytvořte konfigurační soubor s názvem `server.conf`. Konfigurační soubor umístěte do adresáře `/etc/openvpn/server.conf` a dle potřeby změňte níže uvedené řádky. (Celý konfigurační soubor pro server naleznete v příloze A.1)

```
dev tun
```

```
# 10.1.0.1 Lokální IP adresa serveru
```

```
# 10.1.0.2 Lokální IP adresa klienta
```

```
ifconfig 10.0.0.1 10.0.0.2
```

```
# Název sdíleného klíče, který se bude používat
```

```
secret static.key # Statický klíč použitý pro šifrování a dešifrování
```

4. Vytvořte konfigurační soubor `client.conf` a umístěte jej do adresáře `/etc/openvpn/client.conf` a dle potřeby upravte následující řádky. (Celý konfigurační soubor naleznete v příloze A.2)

```
dev tun
```

```
remote 192.168.1.100 # IP adresa, na které je nainstalován VPN Server
```

```
# 10.1.0.2 Lokální IP adresa klienta
```

```
# 10.1.0.1 Lokální IP adresa serveru
```

```
ifconfig 10.0.0.2 10.0.0.1
```

```
# Název sdíleného klíče, který se bude používat
```

```
secret static.key # Statický klíč použitý pro šifrování a dešifrování
```

5. Pomocí uvedeného příkazu spusťte OpenVPN server.

```
openvpn --config /etc/openvpn/server.conf
```

6. Uvedeným příkazem proved'te připojení klientského zařízení k serveru.

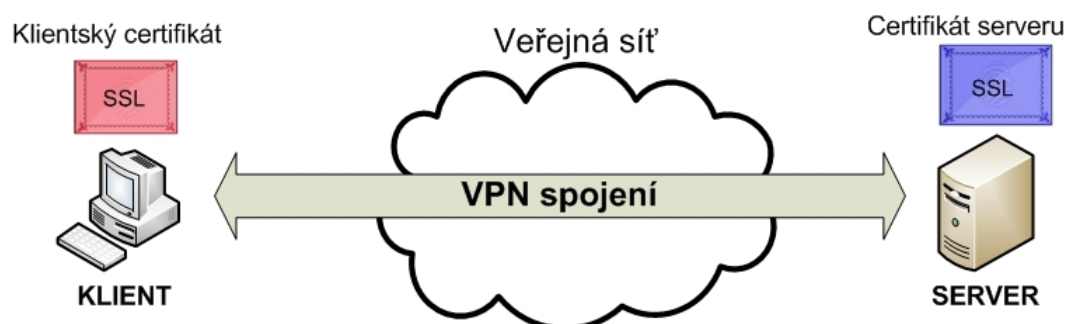
```
openvpn --config /etc/openvpn/client.conf
```

7. Konfiguraci ověřte příkazem `ifconfig`. Na výpisu se zobrazí virtuální rozhraní `TUN0-00`.

[2][3]

4.3 Instalace a konfigurace OpenVPN s využitím SSL/TLS certifikátů

Konfiguraci OpenVPN s využitím SSL/TLS certifikátů spočívá v tom, že námi nainstalovaná certifikační autorita (CA), vystavuje pro server a klienty dvojici klíčů. Jedním je veřejný klíč volně dostupný a druhým je privátní klíč. V tomto případě zastupují klíče vygenerované certifikáty. Při asymetrickém šifrování se využívá veřejného klíče protější strany, kterým šifrujeme přenášená data. Druhá strana pak dešifruje zašifrovaná data pomocí svého privátního klíče, který si chrání. Využitím asymetrického šifrování zvyšujeme bezpečnost při komunikaci a přenosu. Tento proces je ale více náročný na výpočetní výkon.



Obrázek 5: OpenVPN s využitím SSL certifikátů

4.4 Návod pro konfiguraci OpenVPN s využitím SSL/TLS certifikátů

1. Nainstalujte software OpenVPN na server i klientskou stanici pomocí příkazu

```
apt-get install openvpn
```

2. Vygenerujte kořenovou certifikační autoritu (CA), která se bude starat o správu certifikátů. Balíček obsahující skripty a nastavení pro CA naleznete na stránkách projektu OpenVPN.net. Složku uložte do adresáře `/etc/openvpn/`.
3. V textovém souboru `vars`, který naleznete `/etc/openvpn/easy-rsa-master/easy-rsa/2.0/vars` upravte následující řádky.

```
# These are the default values for fields
# which will be placed in the certificate.
# Don't leave any of these fields blank.
export KEY_COUNTRY="CZ"
export KEY_PROVINCE="CZ"
export KEY_CITY="Ostrava"
export KEY_ORG="VSB"
export KEY_EMAIL="mar0178@vsb.cz"
export KEY_OU="BC"
```

4. Pomocí uvedených příkazů spusťte skripty, které naleznete `/etc/openvpn/easy-rsa-master/easy-rsa/2.0/`

```
source ./vars
./clean-all
./build-ca
```

Posledním příkazem `./build-ca` spusťte generování kořenové certifikační autority. Je-li operace úspěšná vytvoří se soubory `ca.crt` a `ca.key`. Ukázka výstupu při generování kořenové CA je uvedena níže.

```
ai:easy-rsa \# ./build-ca
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information
that will be incorporated
into your certificate request.
```

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

```
Country Name (2 letter code) [CZ]:
State or Province Name (full name) [Morava]:
Locality Name (eg, city) [Ostrava]:
Organization Name (eg, company) [VSB]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:
OpenVPN-CAEmail Address [me@myhost.mydomain]:
```

5. Pomocí skriptů vygenerujte certifikáty a klíče pro server a klienta. Pro vygenerování využijte skripty nacházející se ve složce `/etc/openvpn/easy-rsa-master/easy-rsa/2.0/`.

```
./build-key-server server
```

Výstupem uvedeného příkazu budou soubory: `server.crt` a `server.key`

```
./build-key client1
```

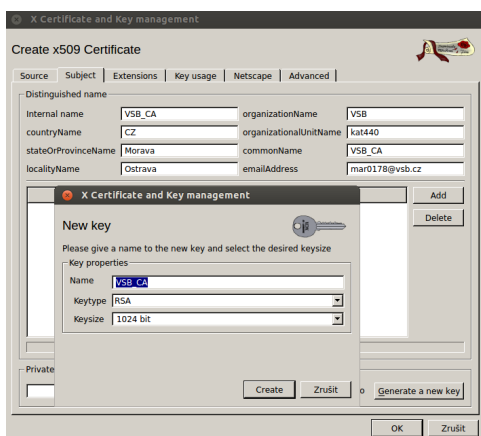
Výstupem druhého příkazu budou dva soubory: `client1.crt` a `client1.key`. Tyto soubory bezpečně přeneste na klientskou stanici a vložte do složky `/etc/openvpn/`.

Pro vytvoření certifikační autority CA a certifikátů pro server a klienty můžete využít program XCA. Program je volně dostupný pro Windows a UNIX systémy. Nabízí grafické rozhraní pro generování certifikátů a jejich správu.

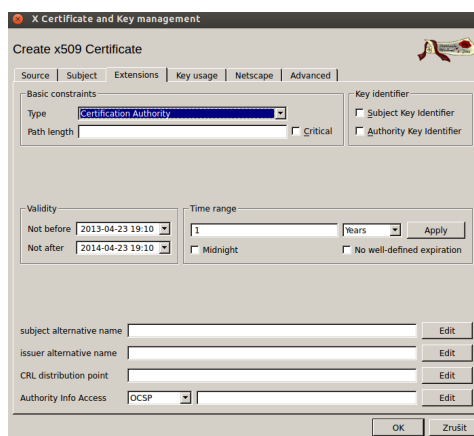
- (a) Program XCA je volně dostupný a je ho možné nainstalovat pomocí příkazu níže.

```
apt-get install xca
```

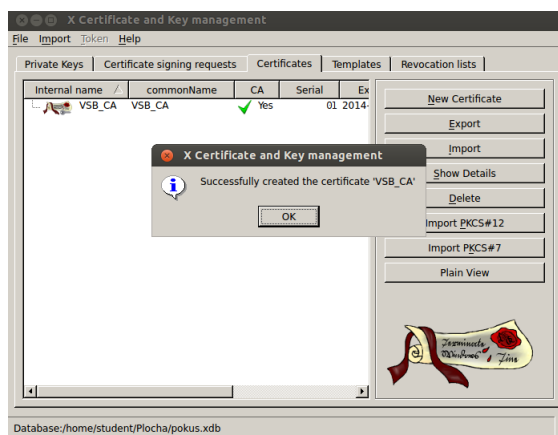
- (b) Po spuštění programu vyberte umístění pro novou databázi certifikátů. Při vytváření databáze budete požádáni o zadání hesla k databázi certifikátů.
- (c) Vytvořte certifikační autoritu CA. Pod tuto certifikační autoritu budou spadat vygenerované certifikáty pro server a klienty. Certifikační autoritu CA, vytvořte na záložce Certificates (Obrázek:6), zvolte položku New Certificate. Vyplňte příslušné údaje na záložce Subject a vygenerujte RSA klíč pomocí Generate a new key pro certifikační autoritu CA. Na záložce Extensions7 zvolte, že se jedná o certifikační autoritu a potvrďte tlačítkem OK.



Obrázek 6: Záložka Subject a generování RSA klíče

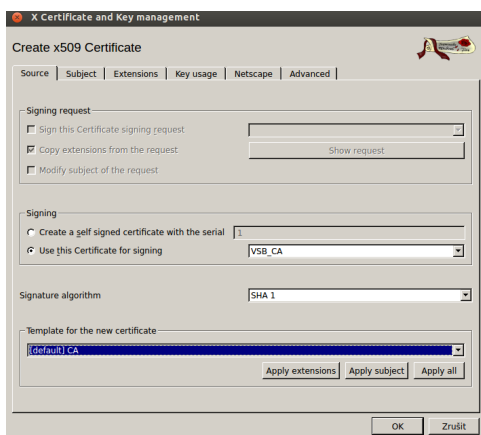


Obrázek 7: Záložka Extensions

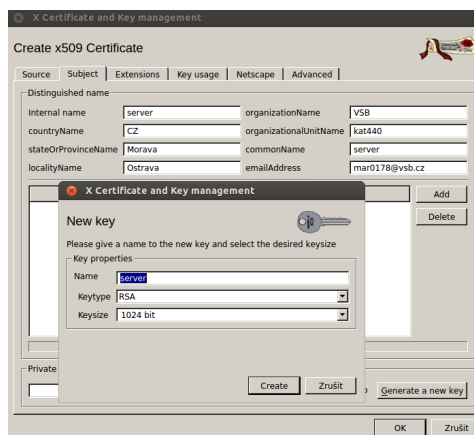


Obrázek 8: Úspěšné vygenerování certifikační autority

- (d) Po úspěšném vytvoření certifikační autority můžete generovat certifikáty pro server a klienty. Certifikáty vygenerujete na záložce Certificates, pomocí New Certificate. Na záložce Source zvolte certifikát pro podepsání a to námi vygenerovanou certifikační autoritu (Obrázek:9). Na záložce Subject vyplňte příslušné údaje, vygenerujte RSA klíč (Obrázek:10) a potvrďte tlačítkem OK. Tento postup opakujte i pro generování certifikátů pro klienty.

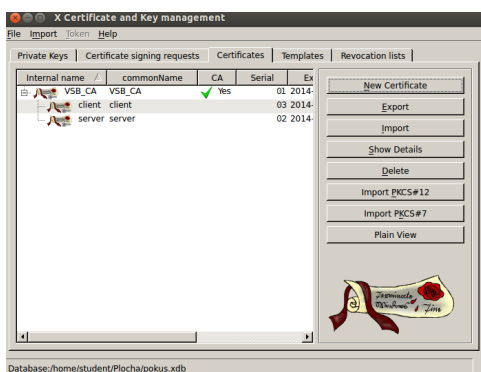


Obrázek 9: Výběr CA pro podpis certifikátů

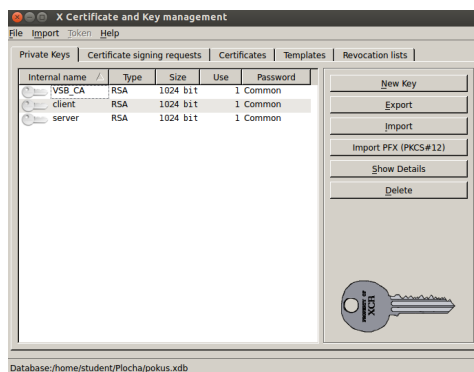


Obrázek 10: Generování RSA klíče

- (e) Na obrázku 11 lze vidět seznam certifikátů a na obrázku 12 lze vidět seznam klíče pro server a klienty.



Obrázek 11: Seznam certifikátů



Obrázek 12: Seznam klíčů

- (f) Certifikáty a klíče je možné vyexportovat označením příslušného certifikátu nebo klíče. Pomocí tlačítka Export vybereme umístění pro vyexportované soubory a potvrďte tlačítkem OK

9. Upravte konfigurační soubor pro klienta. V konfiguračním souboru upravte níže uvedené řádky a soubor uložte do adresáře `/etc/openvpn/client1.conf`. (Celý konfigurační soubor naleznete v příloze B.2)

```
dev tun           #Typ virtuálního rozhraní
proto tcp         #Spojově orientované spojení
remote 192.168.1.24 1194 #Adresa VPN serveru a port
ca ca.crt         #Veřejný klíč kořenové CA
cert client1.crt  #Veřejný klíč klienta
key client1.key   #Tajný klíč klienta,
                  který je nutno udržet v tajnosti
```

10. Pomocí příkazu níže spusťte VPN server.

```
openvpn -config /etc/openvpn/server.conf
```

11. Pomocí příkazu níže se připojete z klientské stanice k VPN serveru.

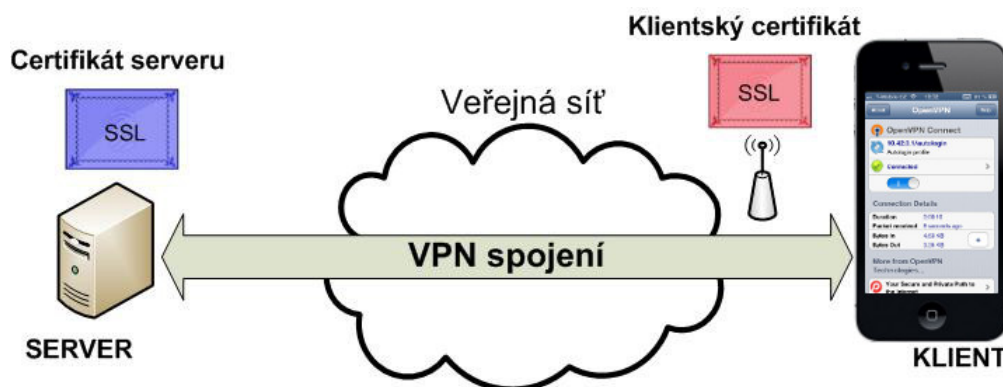
```
openvpn -config /etc/openvpn/client1.conf
```

12. Konfiguraci ověřte příkazem `ifconfig`. Na výpisu se zobrazí virtuální rozhraní `TUN0-00`.

[2][3]

4.5 Konfigurace mobilních zařízení s operačními systémy Apple iOS a Android s využitím SSL/TLS

V této kapitole se seznámíme s konfigurací mobilních zařízení s operačními systémy Apple iOS a Android. Pro připojení klientů k OpenVPN serveru využijeme volně dostupnou aplikaci OpenVPN pro obě platformy mobilních zařízení. Topologie, na které jsem testoval konfiguraci je vyobrazena na obrázku 13

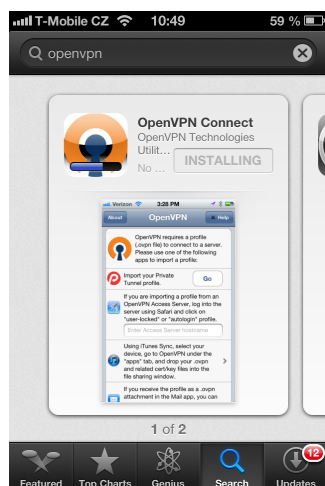


Obrázek 13: OpenVPN pro mobilní zařízení

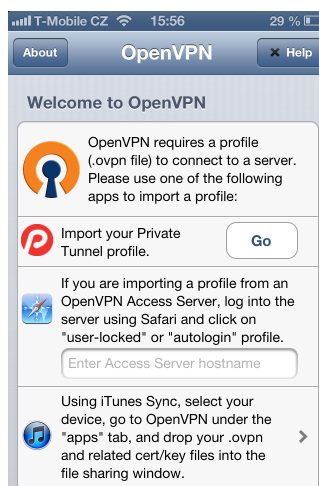
4.5.1 Konfigurace mobilních zařízení s operačním systémem Apple iOS

Níže uvedený postup konfigurace je použitelný pro iPhone 3GS, iPhone 4, iPhone 4S, iPhone 5, iPod touch 3. - 5. generace a tabletu iPad. Proto, by konfigurace mohla proběhnout, je nutné mít nainstalován Apple iOS verze 5 nebo vyšší.

1. Pro konfiguraci OpenVPN serveru a generování certifikátů můžeme využít návod uvedený v kapitole 4.3
2. Nainstalujeme aplikaci OpenVPN z aplikace App Store, která se nachází v mobilním zařízení. Aplikace OpenVPN je vyobrazena na obrázku 14.



Obrázek 14: Aplikace OpenVPN



Obrázek 15: Spuštěná aplikace

3. Vytvoříme konfigurační soubor pro klienta, ve kterém upravíme následující řádky a uložíme jej pod názvem `client.ovpn`. (Celý konfigurační soubor nalezneme v příloze B.2)

```
dev tun                                #Typ virtuálního rozhraní
proto tcp                               #Spojově orientované spojení
remote 192.168.1.24 1194               #Adresa VPN serveru a port
ca ca.crt                              #Veřejný klíč kořenové CA
cert client1.crt                       #Veřejný klíč klienta
key client1.key                        #Tajný klíč klienta,
                                        který je nutno udržet v tajnosti
```

4. V počítači musíme mít uložený klientský certifikát `client1.crt`, klientský tajný klíč `client1.key` a certifikát certifikační autority `ca.crt`.
5. Připojíme zařízení k počítači a spustíme volně dostupný program iTunes. V programu otevřeme připojené mobilní zařízení a přesuneme se na záložku Aplikace. Ze seznamu aplikací vybereme aplikaci OpenVPN a klikneme na tlačítko přidat. Vybereme soubory `client1.crt`, `client1.key`, `ca.crt` a `client.ovpn` a klikneme na tlačítko synchronizovat. (Obrázek:16)



Obrázek 16: Nahrání souborů do aplikace OpenVPN

6. Otevřeme aplikaci OpenVPN a dokončíme nastavení aplikace. Klikneme na zelený symbol +, uvedený na obrázku 17. Konfigurace se nahraje, objeví se posuvný přepínač a stav připojení Disconnected, který je vyobrazen na obrázku 18.



Obrázek 17: Spuštěná aplikace po nahrání souborů



Obrázek 18: Úspěšné nahrání konfigurace a certifikátů

7. Pomocí posuvného přepínače se připojíme k VPN server a stav připojení se změní na Connected. V horním stavovém řádku se objeví ikona VPN stejně jako na obrázku 19. V detailu připojení můžeme dobu připojení a počet přenesených bytů.



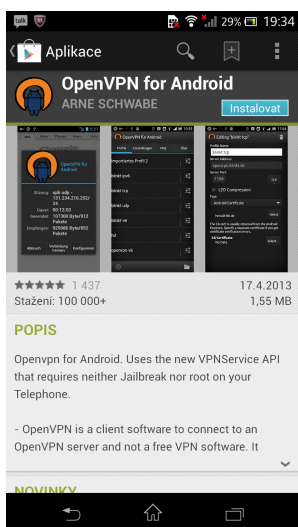
Obrázek 19: Úspěšné připojení k OpenVPN serveru

4.5.2 Konfigurace mobilních zařízení s operačním systémem Android

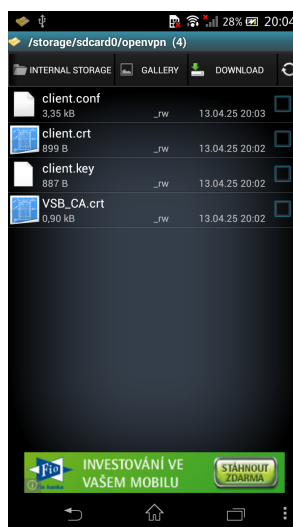
Tento postup konfigurace se vztahuje k operační systém Android 4.0 a vyšší. Aplikace OpenVPN pro operační systém Android 4.0 a vyšší je zdarma. Při konfiguraci mobil-

ních zařízení s operačním systémem Android do verze 4.0 je nutné využít alternativní software, který většinou placený.

1. Pro konfiguraci OpenVPN serveru a generování certifikátů pro server i klienty můžeme využít postup v kapitole 4.3.
2. Z aplikace Google Play nainstalujeme aplikaci OpenVPN for Android, která je zdarma a je zobrazena na obrázku 20.



Obrázek 20: Aplikace OpenVPN for Android

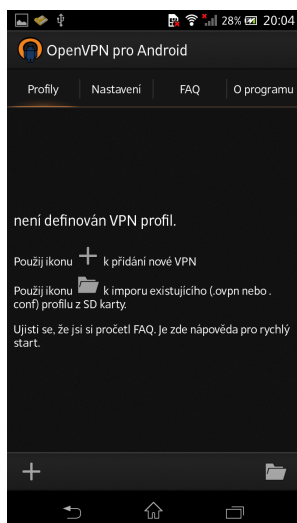


Obrázek 21: Umístění souborů do paměti telefonu

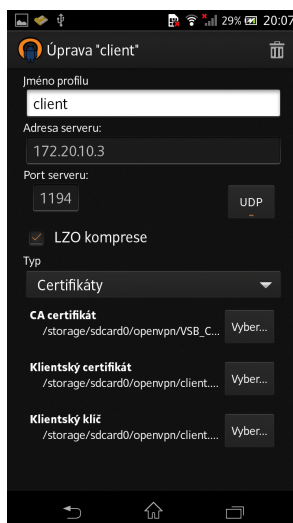
3. V souboru `client.conf` upravíme následující řádky. (Celý soubor nalezneme v příloze B.2)

```
dev tun #Typ virtuálního rozhraní
proto udp #Spojově orientované spojení
remote 172.10.0.3 1194 #Adresa VPN serveru a port
ca ca.crt #Veřejný klíč kořenové CA
cert client.crt #Veřejný klíč klienta
key client.key #Tajný klíč klienta,
který je nutno udržet v tajnosti
```

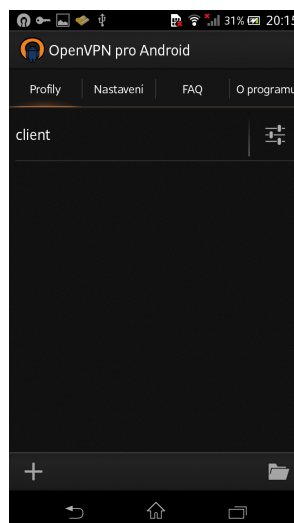
4. Umístíme soubory `client.conf`, `client.crt`, `client.key`, `ca.crt` do paměti mobilního zařízení, konkrétně do složky `openvpn`, stejně jako je vyobrazeno na obrázku 21.



Obrázek 22: Spuštěná aplikace



Obrázek 23: Konfigurace certifikátů



Obrázek 24: Připojení k serveru

5. Spustíme aplikaci OpenVPN for Android a provedeme nastavení. Nainportujeme do aplikace konfigurační soubor `client.conf`, a to pomocí symbolu složky v pravém dolním rohu. Symbol složky je vyobrazen na obrázku 22.
6. Dojde-li k úspěšnému nahrání konfiguračního souboru pro klienta, zobrazí se konfigurace, která je uvedena na obrázku 23. V této konfiguraci připojíme soubory `client.crt`, `client.key`, `ca.crt`, uložené ve složce `openvpn` v paměti telefonu.
7. Poté co nastavíme cesty k certifikátům, vrátíme se na úvodní stránku aplikace a připojíme se k VPN serveru kliknutím na řádek `client`. Pokud je připojení k VPN serveru úspěšné, zobrazí se v levém horním rohu symbol klíče.(Obrázek: 24)

[2][3]

5 Závěr

Bakalářská práce je věnována bezplatnému řešení virtuálních privátních sítí pod licencí open source. Byly splněny a realizovány všechny body zadání bakalářské práce.

V teoretické části bakalářské práce je popsán princip fungování virtuální privátní sítě a dostupné zabezpečovací mechanismy. Popsán je také software OpenVPN jeho historie, vývoj, výhody a rozšíření. Provedl jsem porovnání virtuálních privátních sítí, které fungují na protokolu IPsec a OpenVPN.

Praktickou část bakalářské práce jsem realizoval na operačním systému linux Ubuntu 12.04. Operační systém jsem využil jak pro konfiguraci serveru, tak i klientů. Konfigurace byla velmi přehledná a jednoduchá, díky využití textových konfiguračních souborů, kterými se nastavují veškeré parametry serveru i klientů.

Při konfiguraci mobilních zařízení jsem využil dvojici zařízení, kdy jedno funguje na operačním systému Apple iOS a druhé na operačním systému Android 4.0 nebo vyšší. Při konfiguraci mobilních zařízení s operačním systémem Apple iOS je nutné využít operačního systému verze 5.0 nebo vyšší. Tato verze se nachází ve většině přístrojů. Certifikáty a konfigurační soubor se vkládají pomocí programu iTunes nebo prostřednictvím emailů. Tento postup však není bezpečný.

Operační systém Android je dnes nejvíce zastoupeným operačním systémem na trhu mobilních zařízení, avšak ne všechny zařízení podporují operační systém verze 4.0 a vyšší. Proto je nutné využít alternativní zpoplatněné programy pro připojení k OpenVPN serveru. Výhodou při konfiguraci OpenVPN na operačním systému Android, je absence speciálního programu pro nahrání certifikátů a konfiguračního souboru do zařízení. Po připojení mobilního zařízení k počítači se nahrají certifikáty a konfigurační soubor do paměti telefonu.

Tato bakalářská práce mi pomohla rozšířit si znalosti v oblasti počítačových sítí a bezpečnosti při komunikaci na internetu. Praktická část práce může sloužit jako návod pro konfiguraci OpenVPN serveru či klientů, s různými metodami zabezpečení. Další informace o software OpenVPN a jeho rozšíření naleznete v publikacích [1, 2].

6 Reference

- [1] Markus Feilner, Graf Norbert, *Beginning OpenVPN 2.0.9*, Birmingham: Packt Publishing, 2009, ISBN 978-1-847197-06-1.
- [2] KEIJSER, Jan Just. *OpenVPN 2 Cookbook*, Birmingham: Packt Publishing, 2011. ISBN 978-1-84951-010-3.
- [3] OpenVPN. [online] [2013-3-16] <http://openvpn.net>
- [4] OpenManiak. [online] [2012-12-20] <http://openmaniak.com/cz/openvpn.php>

A Konfigurační soubor pro server a klienty s využitím sdílených klíčů

A.1 Konfigurační soubor pro server `server.conf`

```
#
# Sample OpenVPN configuration file for
# office using a pre-shared static key.
#
# '#' or ';' may be used to delimit comments.

# Use a dynamic tun device.
# For Linux 2.2 or non-Linux OSes,
# you may want to use an explicit
# unit number such as "tun1".
# OpenVPN also supports virtual
# ethernet "tap" devices.
dev tun

# 10.1.0.1 is our local VPN endpoint (office).
# 10.1.0.2 is our remote VPN endpoint (home).
ifconfig 10.1.0.1 10.1.0.2

# Our up script will establish routes
# once the VPN is alive.

# Our pre-shared static key
secret static.key

# OpenVPN 2.0 uses UDP port 1194 by default
# (official port assignment by iana.org 11/04).
# OpenVPN 1.x uses UDP port 5000 by default.
# Each OpenVPN tunnel must use
# a different port number.
# lport or rport can be used
```

```
# to denote different ports
# for local and remote.
; port 1194

# Downgrade UID and GID to
# "nobody" after initialization
# for extra security.
; user nobody
; group nobody

# If you built OpenVPN with
# LZO compression, uncomment
# out the following line.
; comp-lzo

# Send a UDP ping to remote once
# every 15 seconds to keep
# stateful firewall connection
# alive. Uncomment this
# out if you are using a stateful
# firewall.
; ping 15

# Uncomment this section for a more reliable detection when a system
# loses its connection. For example, dial-ups or laptops that
# travel to other locations.
; ping 15
; ping-restart 45
; ping-timer-rem
; persist-tun
; persist-key

# Verbosity level.
# 0 -- quiet except for fatal errors.
# 1 -- mostly quiet, but display non-fatal network errors.
# 3 -- medium output, good for normal operation.
```

```
# 9 -- verbose, good for troubleshooting
verb 3
```

A.2 Konfigurační soubor pro klienty `client.conf`

```
#
# Sample OpenVPN configuration file for
# home using a pre-shared static key.
#
# '#' or ';' may be used to delimit comments.

# Use a dynamic tun device.
# For Linux 2.2 or non-Linux OSes,
# you may want to use an explicit
# unit number such as "tun1".
# OpenVPN also supports virtual
# ethernet "tap" devices.
dev tun

# Our OpenVPN peer is the office gateway.
remote 1.2.3.4

# 10.1.0.2 is our local VPN endpoint (home).
# 10.1.0.1 is our remote VPN endpoint (office).
ifconfig 10.1.0.2 10.1.0.1

# Our up script will establish routes
# once the VPN is alive.

# Our pre-shared static key
secret static.key

# OpenVPN 2.0 uses UDP port 1194 by default
# (official port assignment by iana.org 11/04).
# OpenVPN 1.x uses UDP port 5000 by default.
```

```
# Each OpenVPN tunnel must use
# a different port number.
# lport or rport can be used
# to denote different ports
# for local and remote.
; port 1194

# Downgrade UID and GID to
# "nobody" after initialization
# for extra security.
; user nobody
; group nobody

# If you built OpenVPN with
# LZO compression, uncomment
# out the following line.
; comp-lzo

# Send a UDP ping to remote once
# every 15 seconds to keep
# stateful firewall connection
# alive. Uncomment this
# out if you are using a stateful
# firewall.
; ping 15

# Uncomment this section for a more reliable detection when a system
# loses its connection. For example, dial-ups or laptops that
# travel to other locations.
; ping 15
; ping-restart 45
; ping-timer-rem
; persist-tun
; persist-key

# Verbosity level.
```

```
# 0 -- quiet except for fatal errors.  
# 1 -- mostly quiet, but display non-fatal network errors.  
# 3 -- medium output, good for normal operation.  
# 9 -- verbose, good for troubleshooting  
verb 3
```

B Konfigurační soubor pro server a klienty s využitím SSL certifikátů

B.1 Konfigurační soubor pro server `server.conf`

```
#####  
# Sample OpenVPN 2.0 config file for #  
# multi-client server. #  
# #  
# This file is for the server side #  
# of a many-clients one-server #  
# OpenVPN configuration. #  
# #  
# OpenVPN also supports #  
# single-machine single-machine #  
# configurations (See the Examples page #  
# on the web site for more info). #  
# #  
# This config should work on Windows #  
# or Linux/BSD systems. Remember on #  
# Windows to quote pathnames and use #  
# double backslashes, e.g.: #  
# "C:\\Program Files\\OpenVPN\\config\\foo.key" #  
# #  
# Comments are preceded with '#' or ';' #  
#####  
  
# Which local IP address should OpenVPN  
# listen on? (optional)  
;local a.b.c.d  
  
# Which TCP/UDP port should OpenVPN listen on?  
# If you want to run multiple OpenVPN instances  
# on the same machine, use a different port  
# number for each one. You will need to  
# open up this port on your firewall.
```

```
port 1194

# TCP or UDP server?
;proto tcp
proto udp

# "dev tun" will create a routed IP tunnel,
# "dev tap" will create an ethernet tunnel.
# Use "dev tap0" if you are ethernet bridging
# and have precreated a tap0 virtual interface
# and bridged it with your ethernet interface.
# If you want to control access policies
# over the VPN, you must create firewall
# rules for the the TUN/TAP interface.
# On non-Windows systems, you can give
# an explicit unit number, such as tun0.
# On Windows, use "dev-node" for this.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun

# Windows needs the TAP-Windows adapter name
# from the Network Connections panel if you
# have more than one.  On XP SP2 or higher,
# you may need to selectively disable the
# Windows firewall for the TAP adapter.
# Non-Windows systems usually don't need this.
;dev-node MyTap

# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key).  Each client
# and the server must have their own cert and
# key file.  The server and all clients will
# use the same ca file.
```

```
#
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys. Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca VSB_CA.crt
cert server.crt
key server.key # This file should be kept secret

# Diffie hellman parameters.
# Generate your own with:
# openssl dhparam -out dh1024.pem 1024
# Substitute 2048 for 1024 if you are using
# 2048 bit keys.
dh dh2048.pem

# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.8.0.1 for itself,
# the rest will be made available to clients.
# Each client will be able to reach the server
# on 10.8.0.1. Comment this line out if you are
# ethernet bridging. See the man page for more info.
server 10.8.0.0 255.255.255.0

# Maintain a record of client virtual IP address
# associations in this file. If OpenVPN goes down or
# is restarted, reconnecting clients can be assigned
# the same virtual IP address from the pool that was
# previously assigned.
ifconfig-pool-persist ipp.txt
```

```
# Configure server mode for ethernet bridging.
# You must first use your OS's bridging capability
# to bridge the TAP interface with the ethernet
# NIC interface. Then you must manually set the
# IP/netmask on the bridge interface, here we
# assume 10.8.0.4/255.255.255.0. Finally we
# must set aside an IP range in this subnet
# (start=10.8.0.50 end=10.8.0.100) to allocate
# to connecting clients. Leave this line commented
# out unless you are ethernet bridging.
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100

# Push routes to the client to allow it
# to reach other private subnets behind
# the server. Remember that these
# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server.
;push "route 192.168.10.0 255.255.255.0"
;push "route 192.168.20.0 255.255.255.0"

# To assign specific IP addresses to specific
# clients or if a connecting client has a private
# subnet behind it that should also have VPN access,
# use the subdirectory "ccd" for client-specific
# configuration files (see man page for more info).

# EXAMPLE: Suppose the client
# having the certificate common name "Thelonious"
# also has a small subnet behind his connecting
# machine, such as 192.168.40.128/255.255.255.248.
# First, uncomment out these lines:
;client-config-dir ccd
;route 192.168.40.128 255.255.255.248
```

```
# Then create a file ccd/Thelonious with this line:
#   iroute 192.168.40.128 255.255.255.248
# This will allow Thelonious' private subnet to
# access the VPN.  This example will only work
# if you are routing, not bridging, i.e. you are
# using "dev tun" and "server" directives.

# EXAMPLE: Suppose you want to give
# Thelonious a fixed VPN IP address of 10.9.0.1.
# First uncomment out these lines:
;client-config-dir ccd
;route 10.9.0.0 255.255.255.252
# Then add this line to ccd/Thelonious:
#   ifconfig-push 10.9.0.1 10.9.0.2

# Suppose that you want to enable different
# firewall access policies for different groups
# of clients.  There are two methods:
# (1) Run multiple OpenVPN daemons, one for each
#     group, and firewall the TUN/TAP interface
#     for each group/daemon appropriately.
# (2) (Advanced) Create a script to dynamically
#     modify the firewall in response to access
#     from different clients.  See man
#     page for more info on learn-address script.
;learn-address ./script

# If enabled, this directive will configure
# all clients to redirect their default
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
# the TUN/TAP interface to the internet in
# order for this to work properly).
# CAVEAT: May break client's network config if
```

```
# client's local DHCP server packets get routed
# through the tunnel.  Solution: make sure
# client's local DHCP server is reachable via
# a more specific route than the default route
# of 0.0.0.0/0.0.0.0.
;push "redirect-gateway"

# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses.  CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
;push "dhcp-option DNS 10.8.0.1"
;push "dhcp-option WINS 10.8.0.1"

# Uncomment this directive to allow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.
;client-to-client

# Uncomment this directive if multiple clients
# might connect with the same certificate/key
# files or common names.  This is recommended
# only for testing purposes.  For production use,
# each client should have its own certificate/key
# pair.
#
# IF YOU HAVE NOT GENERATED INDIVIDUAL
# CERTIFICATE/KEY PAIRS FOR EACH CLIENT,
# EACH HAVING ITS OWN UNIQUE "COMMON NAME",
# UNCOMMENT THIS LINE OUT.
;duplicate-cn

# The keepalive directive causes ping-like
```

```
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120

# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
#
# Generate with:
#   openvpn --genkey --secret ta.key
#
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
;tls-auth ta.key 0 # This file is secret

# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
;cipher BF-CBC          # Blowfish (default)
;cipher AES-128-CBC    # AES
;cipher DES-EDE3-CBC   # Triple-DES

# Enable compression on the VPN link.
# If you enable it here, you must also
# enable it in the client config file.
comp-lzo

# The maximum number of concurrently connected
# clients we want to allow.
;max-clients 100
```

```
# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
#
# You can uncomment this out on
# non-Windows systems.
;user nobody
;group nobody

# The persist options will try to avoid
# accessing certain resources on restart
# that may no longer be accessible because
# of the privilege downgrade.
persist-key
persist-tun

# Output a short status file showing
# current connections, truncated
# and rewritten every minute.
status openvpn-status.log

# By default, log messages will go to the syslog (or
# on Windows, if running as a service, they will go to
# the "\Program Files\OpenVPN\log" directory).
# Use log or log-append to override this default.
# "log" will truncate the log file on OpenVPN startup,
# while "log-append" will append to it. Use one
# or the other (but not both).
;log          openvpn.log
;log-append   openvpn.log

# Set the appropriate level of log
# file verbosity.
#
# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
```

```
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
verb 3

# Silence repeating messages. At most 20
# sequential messages of the same message
# category will be output to the log.
;mute 20
```

B.2 Konfigurační soubor pro klienty `client.conf`

```
#####
# Sample client-side OpenVPN 2.0 config file #
# for connecting to multi-client server.    #
#                                           #
# This configuration can be used by multiple #
# clients, however each client should have #
# its own cert and key files.              #
#                                           #
# On Windows, you might want to rename this #
# file so it has a .ovpn extension         #
#####

# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client

# Use the same setting as you are using on
# the server.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
```

```
dev tun

# Windows needs the TAP-Windows adapter name
# from the Network Connections panel
# if you have more than one.  On XP SP2,
# you may need to disable the firewall
# for the TAP adapter.
;dev-node MyTap

# Are we connecting to a TCP or
# UDP server?  Use the same setting as
# on the server.
;proto tcp
proto udp

# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote 172.20.10.3 1194
;remote my-server-2 1194

# Choose a random host from the remote
# list for load-balancing.  Otherwise
# try hosts in the order specified.
;remote-random

# Keep trying indefinitely to resolve the
# host name of the OpenVPN server.  Very useful
# on machines which are not permanently connected
# to the internet such as laptops.
resolv-retry infinite

# Most clients don't need to bind to
# a specific local port number.
nobind
```

```
# Downgrade privileges after initialization (non-Windows only)
;user nobody
;group nobody

# Try to preserve some state across restarts.
persist-key
persist-tun

# If you are connecting through an
# HTTP proxy to reach the actual OpenVPN
# server, put the proxy server/IP and
# port number here. See the man page
# if your proxy server requires
# authentication.
;http-proxy-retry # retry on connection failures
;http-proxy [proxy server] [proxy port #]

# Wireless networks often produce a lot
# of duplicate packets. Set this flag
# to silence duplicate packet warnings.
;mute-replay-warnings

# SSL/TLS parms.
# See the server config file for more
# description. It's best to use
# a separate .crt/.key file pair
# for each client. A single ca
# file can be used for all clients.
ca VSB_CA.crt
cert client.crt
key client.key

# Verify server certificate by checking
# that the certificate has the nsCertType
# field set to "server". This is an
# important precaution to protect against
```

```
# a potential attack discussed here:
# http://openvpn.net/howto.html#mitm
#
# To use this feature, you will need to generate
# your server certificates with the nsCertType
# field set to "server". The build-key-server
# script in the easy-rsa folder will do this.
;ns-cert-type server

# If a tls-auth key is used on the server
# then every client must also have the key.
;tls-auth ta.key 1

# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
;cipher x

# Enable compression on the VPN link.
# Don't enable this unless it is also
# enabled in the server config file.
comp-lzo

# Set log file verbosity.
verb 3

# Silence repeating messages
;mute 20
```