TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: M2612 – Elektrotechnika a informatika

Studijní obor: 3906T001 – Mechatronika

Bezpečná VPN

Secure VPN

Diplomová práce

Autor: **Petr Sedláček**

Vedoucí práce: Mgr. David Kmoch

V Liberci 18. 5. 2007

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé diplomové práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce.

Datum 18.5.2007

Podpis

Poděkování

Rád bych poděkoval Mgr. Davidu Kmochovi za vedení a podporu při tvorbě diplomové práce. Děkuji také svým nejbližším za morální a finanční podporu po celou dobu mého studia.

Abstrakt

Diplomová práce se zaměřuje na problematiku vytvoření a provozování virtuálních privátních sítí (VPN). Je zde uveden přehled používaných protokolů potřebných k úspěšnému navázání spojení. Jednotlivé protokoly jsou stručně popsány pro lepší orientaci v dané problematice. Součástí práce je vlastní návrh modelu virtuální privátní sítě s podrobným popisem instalace a konfigurace. Síť je zrealizována pomocí programů s otevřeným zdrojovým kódem, které využívají různé metody pro vytvoření VPN. V závěru práce jsou jednotlivé metody porovnány a zmíněny poznatky zaznamenané během instalace a následného provozu sítě.

Abstract

This diploma theses deals with questions of creating and keeping of Virtual Private Networks (VPN). The list of protocols necessary for successful connection is also presented. Individual protocols are briefly described for better exposure. Proper suggestion of a model VPN with detailed description of instalation and configuration is also part of this work. The network is created by means of open source software that are using various methods for establishing VPN. Indivdual methods are compared in the end of the work. There are also notes about instalation and running the network registered during the project.

Obsah

Seznam ilustrací

Úvod

Virtuální privátní síť (VPN - Virtual Private Network) je termín popisující vytvoření spojení mezi dvěma koncovými zařízeními. Tato zařízení se mohou od sebe nacházet libovolně daleko, avšak spolupracují spolu, jako by byla připojena do jedné stejné lokální sítě. Ve skutečnosti jejich komunikace může probíhat přes několik různých síťových uzlů. Přímé propojení takto vzdálených zařízení by bylo finančně nákladné, proto se nabízí využití již vybudované infrastruktury. Toto řešení je také ideální k přístupu například do firemní sítě pro mobilní uživatele, díky téměř všude dostupnému přístupu na Internet, který může sloužit jako tranzitní síť. Důležitým požadavkem je ale zachovat privátnost komunikace, kdy je potřeba probíhající komunikaci odstínit od okolí. Nabízí se několik možných řešení závislých na požadavcích na stupeň zabezpečení.

Cílem této práce je zmapování současné problematiky virtuálních privátních sítí a návrh bezpečného přístupu pomocí VPN s podrobným popisem jednotlivých kroků.

1. Přehled současné problematiky

1.1. Virtuální Privátní Síť

(VPN = Virtual Private Network = Virtuální privátní síť)

Hlavním cílem při vytvoření tohoto řešení je spojení dvou a více uzlů do jedné sítě. Uzel může být celá síťová struktura nebo pouze jeden počítač. Potřebné propojení dvou sítí nebo uzlů je možné realizovat pomocí veřejných a nedůvěryhodných vybudovaných infrastruktur, například Internet, ale je potřeba tuto komunikaci zašifrovat nebo jinak oddělit od okolní komunikace a zabránit tak nežádoucím odposlechům, změnám nebo ztrátám dat. Výsledkem není reálná síť, ale pouze virtuální připojení pomocí VPN programu.

Dalším významným požadavkem při tvorbě VPN je QoS[1] - Kvalitativní parametry přenosu. Jedná se o dostupnost a spolehlivost použité infrastruktury. Zde je potřeba zvážit potencionální požadavky v porovnání s finančními náklady. Není podmínkou využívat veřejné sítě, protože existují poskytovatelé spojových služeb nabízející propojení vzdálených uzlů pomocí jejich přenosového systému. Výhodou této plně privátní sítě je vyšší bezpečnost a spolehlivost na úkor značně vyšší provozní ceny.

Virtuální privátní síť není omezena jen pro vytvoření mezi dvěma koncovými počítači, může být také využita pro propojení několika lokálních sítí jednotlivých organizací. Je tak možné propojit libovolný počet poboček kdekoliv

1 Quality of Service

na světě a výsledkem je obdobný stav jako propojení počítačů v jedné místnosti.

1.2. Zabezpečení

Dnes je nejčastěji používáno jako přenosové médium síť Internet. Díky své dostupnosti a poměrně levnému připojení k sítí se stává velice vhodnou volbou. Snadný přístup ale přináší i svá rizika a při zvolení tohoto typu propojení je třeba dbát na prvky zabezpečení pro komunikaci mezi jednotlivými koncovými zařízeními.

Spojení vytvořené pomocí VPN bývá často bezpečné, nemělo by se ale zapomínat na koncové uzly připojené k lokální síti, které nemusí být dostatečně zabezpečeny.

Základním konceptem pro zabezpečení komunikace je využití firewallu pro přístup oprávněných uživatelů společně s certifikáty pro ověření pravosti, se kterými se komunikuje v zašifrované podobě prostřednictvím vytvořeného tunelu.

Jednotlivé prvky zabezpečení:

- ***Firewall***

Firewall je bezpečnostní prvek určený k řízení síťového provozu. Firewally mohou být jako softwarová nebo hardwarová řešení. Základní vlastností jsou definovaná pravidla přístupu pro jednotlivé programy nebo uživatele. Neautorizovaná spojení tak nemají do vnitřní sítě přístup. V dnešní době obsahují firewally navíc systémy[2] pro detekci útoků, které fungují jako účinné nástroje pro odhalování virů, spyware a případných jiných útoků.

- ***Šifrování***

Zabezpečení firewallem se vztahuje obvykle na koncové body komunikace. Většina VPN technologií podporuje nějaký typ šifrování, které slouží k zakódování dat během přenosu po nedůvěryhodné síti. K zvýšení

2 IDS - Intrusion Detection Systems

bezpečnosti bývá pravidlem šifrovací klíče měnit, pokud by tomu tak nebylo, případný útočník, který by komunikaci sledoval, by po čase mohl šifrovací klíč najít a komunikaci odposlouchávat nebo jinak zneužít. Pokud je klíč pravidelně měněn, má útočník pouze omezenou dobu na zjištění používaného klíče.

Pro zakódování se používají dva typy šifer:

Symetrické:

- Na obou stranách komunikace musí existovat nějaký stejný tajný klíč (heslo), podle kterého se budou data kódovat a dekódovat. Každý takto připojený uživatel musí klíč znát. Nevýhodou je použití tohoto typu šifrování u rozsáhlých sítí, kdy při snaze změnit tento klíč, se musí oznámit a případně změnit klíč u všech uživatelů.

Asymetrické:

- K zakódování a dekódování je zapotřebí dvou klíčů, privátního a veřejného klíče. Privátní klíč zná pouze sám uživatel a veřejný klíč je k dispozici uživatelům, se kterými má být komunikace navázána. Pokud chce uživatel s někým komunikovat, zašifruje data pomocí sdíleného klíče adresáta. Adresát dekóduje data prostřednictvím svého privátního klíče. Výhodou je složitost vzájemného odvození privátního a sdíleného klíče.

- **Přehled šifrovacích algoritmů:**

- Hašovací funkce - MD5 SHA-1 SHA-2 - šifrování založené na kontrolních výpočtech
- DES (Data Encryption Standard) (TripleDES, AES)- šifrování pomocí tajného klíče
- RSA - systém šifrování s veřejným klíčem

- ***Ověření pravosti***

Jedná se o ověření identity odesílatele a příjemce. Mnohé z VPN řešení používají autentizaci na principu sdíleného klíče. Autentizace se provádí obvykle

na začátku komunikace, někdy je provedena také v průběhu komunikace. Pomocí autentizace se také nechá odhalit odposlech s následnou změnou dat pomocí kontrolního součtu.

Používán je například protokol CHAP (The Challenge Handshake Authentication Protocol), kde se neposílají čistá autentizační data, ale tzv. *Chalange* - výzvy a *Hash* - mezivýpočty. Po zaslání *Chalange* serverem provede připojovaný uživatel kontrolní součet mezi příchozími a vlastními daty. Odešle kontrolní součet na server, a pokud se shodují, přístup je povolen. Výpočet se provádí pomocí MD5 algoritmu.

Vysoký důraz je také kladen na integritu dat, což je zárukou, že data během přenosu nebyla změněna.[3]

3 Pužmanová, R.: Bezpečnost ve VPN: IPSec versus SSL [online]. 2006, [cit. 2006-09-12]. Dostupné z: <http://www.dsl.cz/clanky.php?clanek=515>.

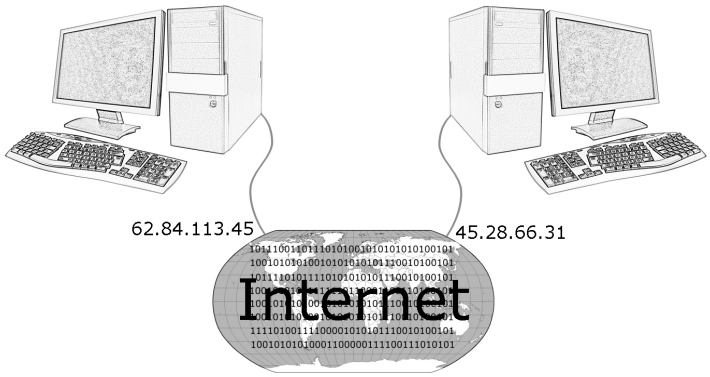
2. Dostupná řešení

Rozdělení jednotlivých metod řešení VPN nelze jednoznačně určit, lze je škálovat podle mnoha pravidel a vlastností. Možné dělení je podle stupně bezpečnosti, typu šifrování nebo podle využívaných protokolů.

Typické je například rozdělení podle způsobu komunikace:

Uživatel - Uživatel

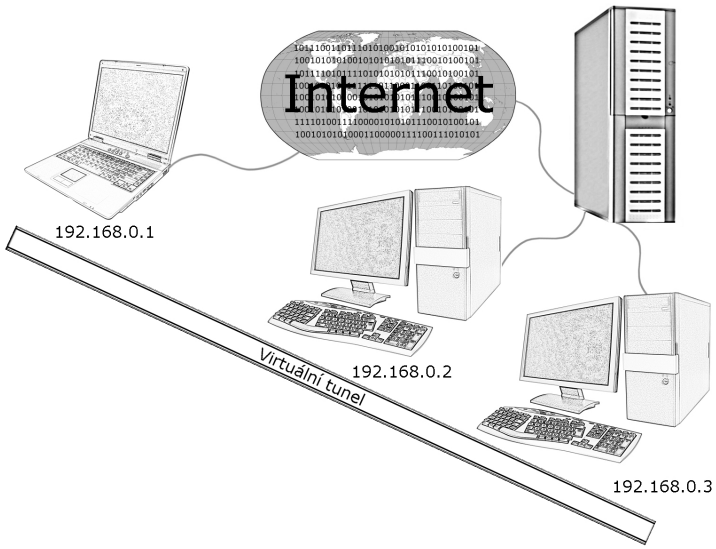
Propojení je realizováno prostřednictvím přímého spojení, tj. odpadá nutnost překládat IP adresy. U obou klientů musí být přímo dosažitelná IP adresa, která je známá druhé straně. Privátnost je zachována pouze v podobě zabezpečení dat šifrováním.



Obr 2.1: Zjednodušený princip komunikace Uživatel - Uživatel

Uživatel - Server

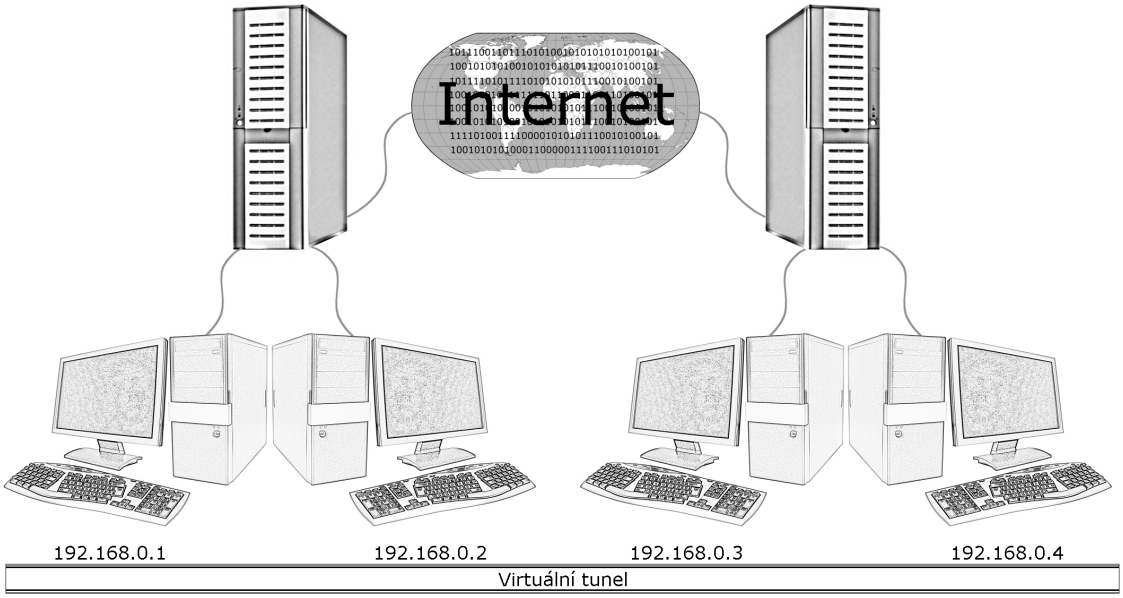
V tomto typu připojení již vzniká tzv. tunel mezi jednotlivými uživateli připojených do privátní sítě. Na Obr 2.2 je jednoduše znázorněn princip připojení. Mobilní uživatel využívá tranzitní síť Internet ke spojení se serverem. Uživatel i server mají libovolné IP adresy Internetu. Po navázaní spojení a úspěšné autorizaci získává uživatel přidělenou IP adresu firemní sítě. Tímto vzniká tunel určený pro komunikaci mezi uživatelem a jednotlivými počítači připojenými k serveru. Autentizace a bezpečnostní prvky potřebné k vytvoření tunelu s následným připojením do sítě jsou realizovány většinou pouze na straně mobilního uživatele a na straně serveru.



Obr 2.2: Zjednodušený princip komunikace Uživatel - Server

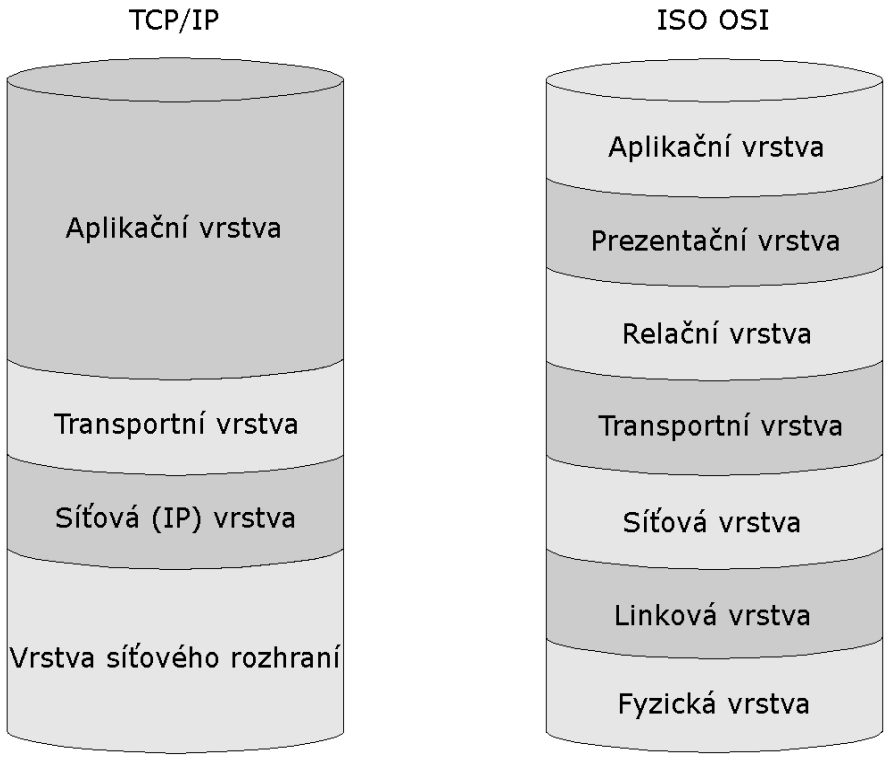
Server - Server

Složitější případy při budování tunelů mezi několika již vytvořenými síťovými infrastrukturami jsou podmíněny důsledným dodržováním adresní politiky. Princip je obdobný jako výše uvedené příklady. Požadováno je vytvoření tunelu, přes který budou servery komunikovat. Po úspěšné autentizaci a nastavení klíčů pro zabezpečenou šifrovanou komunikaci si navzájem vyměňují požadavky od počítačů ze své sítě. K cílovému počítači jsou data doručena v podobě, jako by byl odesílatel na stejné síti. O transformaci při odeslání a přijmutí upravených paketů pro přenos po Internetu se starají servery.



Obr 2.3: Zjednodušený princip komunikace Server - Server

Za obecně nejpřehlednější řešení považuji rozdělení podle síťového modelu TCP/IP:



Obr 2.4: TCP/IP protokolový zásobník a model ISO OSI

a) VPN na vrstvě síťového rozhraní

Pro vytvoření VPN na síťové vrstvě se často používala, a dnes ještě používá, infrastruktura tvořená sítěmi s protokolem Frame Relay nebo ATM.

Frame Relay

Standardizovaný protokol Frame Relay využívá přepojování rámců (proměnné délky do 8189 Byte) na vrstvě síťového rozhraní. Rychlost se pohybuje v řádu Mbit/s. Rámce se nezpracovávají na koncích datového spoje sítě, ale jsou kontrolovány a zpracovávány až u cílové stanice. Kontrolu o správném doručení do cílové stanice protokol nezajišťuje, problematiku přenechává na protokolech vyšších vrstev u cílové stanice.

K dispozici jsou služby se spojením po virtuálních okruzích PVC a SVC. PVC (Permanent Virtual Circuit) jsou virtuální okruhy pevně sestavené. SVC (Switched Virtual Circuit) komutované virtuální okruhy, které dynamicky vznikají podle aktuální potřeby.

ATM

Asynchronní režim transferu ATM (Asynchronous Transfer Mode) kombinuje paketový přenos se synchronním přenosem. Data jsou přenášena ve stejně velkých (53 Byte) datových jednotkách nazývaných buňky. K dispozici je také služba se spojením po virtuálních okruzích PVC a SVC. Rychlost přenosu se pohybuje v řádu stovek Mbit/s.

- ***MPOA***

(Multiprotocol over ATM)

Principem je oddělení zpracování směrování od přenosu paketů mezi jednotlivými subsítěmi. O výpočtovou část (správa adres, rozhodování apod.) se stará MPOA server. Jde o centrální směrovací server s ATM připojením. MPOA

Clients představují okrajová zařízení, která zajišťují fyzické směrování paketů podle pokynů směrovacího serveru.

Výhodou MPOA je dynamické vytváření virtuálních obvodů mezi koncovými uzly. Záporem je značně limitující omezení na ATM jako přenosové technologii.[4]

Sítě typu ATM začaly být nahrazovány novějšími technologiemi. V moderních paketových sítích jsou v dnešní době používány například MPLS, GMPLS (Generalized MPLS).

- ***MPLS***

(Multiprotocol Label Switching)

Jedná se také o protokol založený na oddělení procesu směrování od vlastního předávání paketů. Směrovač na okraji sítě MPLS příchozímu paketu přidělí značku, podle které je dále s paketem zacházeno v průchodu mezi směrovači uvnitř sítě. Takto je určena sekvence přepínačů, kterými paket projde, včetně výstupního směrovače ze sítě MPLS. Před výstupním směrovačem je značka odstraněna a pomocí směrovací tabulky výstupní směrovač doručí paket. Kromě značky, určující cestu k cílovému výstupnímu směrovači, si předávají koncové výstupní směrovače také informace, do jaké sítě má být paket zaslán, což zajišťuje správné doručení.

Pro uživatele je celá MPLS síť, kromě vstupních směrovačů, skryta. Pomocí značek je tak možné bezpečné oddělení komunikace.[5]

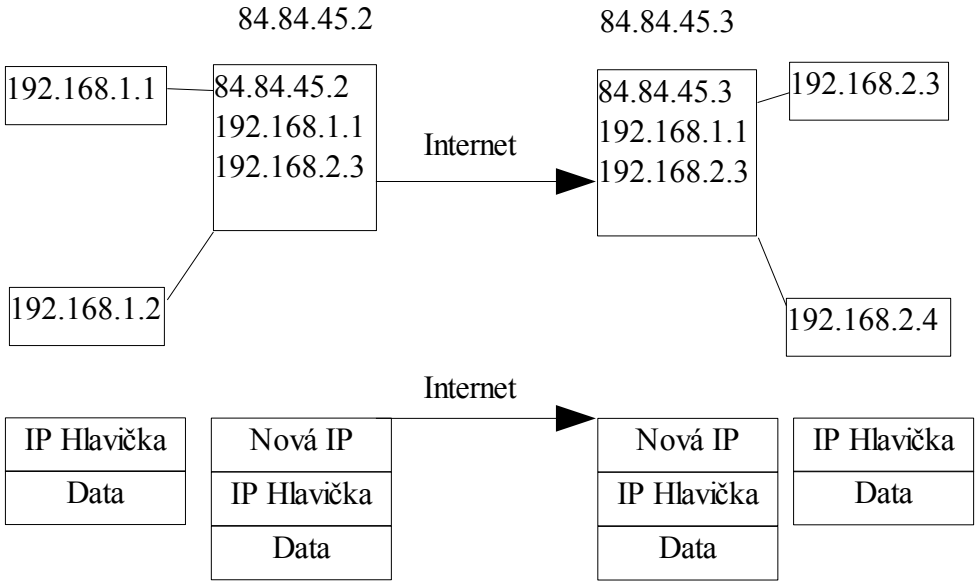
4 Luhový, K.: Virtuální privátní sítě VPN [online]. 2003 [cit. 2003-02-18]. Dostupné z: <http://www.svetsiti.cz/view.asp?rubrika=Tutorialy&temaID=219&clanekID=230>.

5 Pužmanová, R.: Vývoj paketových sítí a postavení MPLS [online]. 2006, [cit. 2006-07-24]. Dostupné z: <http://www.svetsiti.cz/view.asp?rubrika=Technologie&clanekID=302>.

Přehled dalších používaných protokolů:

- ***IPoverIP***

Jedná se o základní protokol umožňující přenášet pakety IP protokolu v obálce pod jiným IP paketem s vlastní hlavičkou. Následně je pak využíváno jiné směrovací politiky.



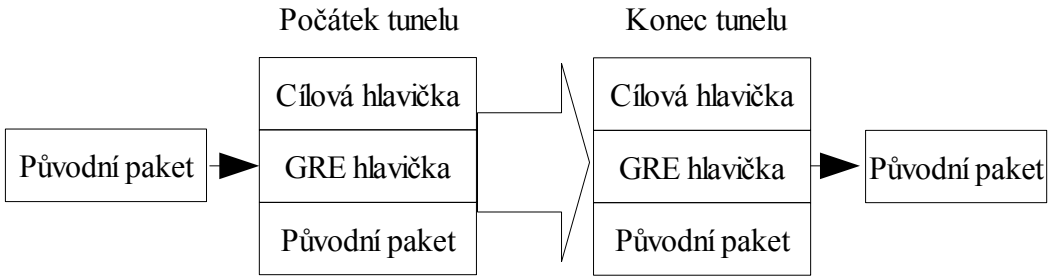
Před odesláním paketu do sítě je přidána nová IP hlavička s novými údaji. IP adresa odesílatele a adresáta je změněna tak, aby byla dostupná v přenosové síti.

Tento protokol je možné použít pro vybudování VPN v rámci vlastní sítě, například jen pro základní oddělení některých okruhů, rozhodně není vhodný pro vybudování VPN v síti Internet, kde je potřeba většího stupně zabezpečení.

- ***GRE***

(General Routing Encapsulation protocol)

Protokol umožňuje vytvořit tunel pro přenos paketů. Není vázaný pouze na přenos po IP protokolu.

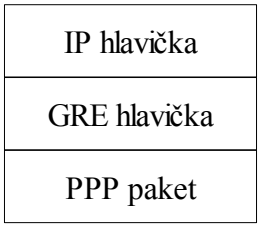


Na počátečním bodu tunelu je k původnímu paketu přidána GRE hlavička, která umožňuje příjemci na koncovém bodu tunelu zjistit informace o zabaleném paketu. GRE hlavička v koncovém bodu tunelu je odstraněna a původní paket pokračuje s původní IP adresou k příjemci. GRE patří mezi poměrně jednoduché a rozšířené standardy.

- ***PPTP***

(Point-to-Point Tunneling Protocol)

Využívá pro svou činnost protokolu PPP (point-to-point protocol). Při přenosu je původní paket obalen do PPP protokolu a pak ještě do upraveného GRE protokolu, naposled je přidána odpovídající IP hlavička pro přenosovou síť. PPTP neobsahuje žádné vlastní zabezpečení nad přenášenými daty, ale právě díky použitému PPP protokolu, který data může šifrovat, je zajištěna určitá ochrana dat.



Komunikace probíhá metodou klient-server. Výhodou protokolu je široká rozšiřitelnost ve většině operačních systémů.

- ***L2TP***

(Layer 2 Tunneling Protocol)

Vychází z protokolu L2F a ze specifikace PPTP.

Mezi hlavní vlastnosti protokolu patří:

Používání autentizačního schématu protokolu PPP.

Plná podpora IPSec.

Přenos PPP přes sítě jiného typu.

b) VPN na síťové vrstvě

Síťová vrstva obsahuje směrovací informace potřebné ke směrování IP protokolu, což je základ pro vytvoření VPN na síťové vrstvě. Vytvořené VPN se liší v závislosti na použitém protokolu, který udává stupeň zabezpečení, typ šifrování atd.

- ***IPSec***

Internetový protokol IPv4 neobsahuje spolehlivou ochranu přenášených dat, negarantuje ani pravost odesílatele. Protokol IPSec (IPSecurity) byl vytvořen ve snaze poskytnout potřebné zabezpečení. Ovládá jak vytvoření tunelu, tak i zabezpečení autenticity a šifrování přenášených dat.

IPSec není samotný protokol, ale skupina protokolů, které spolupracují na bezpečnosti jednotlivých IP datagramů.

Hlavní protokoly IPSec:

IKE (Internet Key Exchange)

- Protokol ošetřující výměnu a ověřování pravosti klíčů mezi systémy a zprostředkovávající šifrovací algoritmy.

AH (Authentication Header)

- Protokol zajišťující integritu a autentizaci dat.

- Zabraňuje útokům, které využívají zopakování přenášených dat.

ESP (Encapsulating Security Payload)

- Protokol, který také zajišťuje integritu a autentizaci dat nebo může být zvolen pro šifrování dat.

Protokol IPSec má dva základní režimy:

Transportní režim:

- Data jsou zabalena do IP paketu s adresou příjemce a odesílatele v přenosové síti. Transportní režim je používán u způsobu komunikace Uživatel - Uživatel.

Tunelový režim:

- Pracuje se pouze s datovou částí, IP hlavička zůstává. V praxi je použit tunelovací režim u sítí typu Server - Server.[6]

Protokol je používán pro vytvoření VPN s vysokým stupněm zabezpečení. *Při testování průniku do systému se bezpečnost VPN často přehlíží. Vede k tomu několik důvodů: Sítě VPN jsou mnohdy považovány za vnitřně bezpečné, protože využívají silné šifrování, systémy IPsec VPN nejsou běžně zachyceny při skenování portů a samotná složitost protokolů zastraší řadu lidí. Máme-li však správné nástroje a techniky, je relativně jednoduché odhalit a provést fingerprinting*[7] *těchto systémů.*[8] Protokol bude využit i v následující části práce při realizaci virtuální privátní sítě.

6 Pužmanová, R.: Bezpečnost ve VPN: IPSec versus SSL [online]. 2006, [cit. 2006-09-12]. Dostupné z: <http://www.dsl.cz/clanky.php?clanek=515>.

7 Technika pro zjištění informací o typu provozovaných služeb na serveru k možnému následnému zneužití.

8 Hills, R. Odhalení a fingerprinting systémů IPsec VPN. Hakin9, 2005, č.6, s. 52.

c) VPN na transportní a aplikační vrstvě

- ***SSL/TLS***

(Secure Socket Layer/Transport Layer Security)

Snahou je vytvoření šifrovaného tunelu založeného na protokolu SSL. Při použití tohoto protokolu nedochází ke změně IP nebo TCP hlavičky paketu. Šifrována jsou pouze přenášená data mezi koncovými zařízeními. SSL/TLS využívá asymetrické šifrovací technologie k zabezpečení identity uživatele pomocí soukromého a sdíleného klíče. Samotný protokol obsahuje také skupinu dílčích protokolů.

Hlavní protokoly SSL/TLS:

HP (Handshake Protocol)

- Pomocí tohoto protokolu se účastníci komunikace dohodnou na podobě šifrování a způsobu zabezpečení.

RLP (Record Layer Protocol)

- Pracuje s daty, provádí kontrolní výpočty. V případě odeslání data zašifruje a v opačném směru data dešifruje. Tímto protokolem je přenášen Handshake protokol.

CCSP (Change Cipher Specification Protocol)

- Po navázání komunikace prostřednictvím Handshake protokolu informuje o pravidlech komunikace.

AP (Alert Protocol)

- Informuje protistrany o vzniklých chybách spojení.[9]

9 Pužmanová, R.: Bezpečnost ve VPN: IPSec versus SSL [online]. 2006, [cit. 2006-09-12]. Dostupné z: <http://www.dsl.cz/clanky.php?clanek=515>.

3. Nastavení služby VPN na straně serveru

Pro srovnání byly vybrány dva typy VPN rozdílného typu zabezpečení. Projekt OpenVPN je založen na SSL/TLS a OpenSWAN využívá IPSec. Podrobněji bude rozebrána varianta vytvoření VPN prostřednictvím projektu OpenVPN, který se vyznačuje svou snadnou konfigurací v prostředí Linux i Windows.

3.1. OpenVPN (SSL/TLS)

Projekt OpenVPN je šířen pod licencí GNU/GPL.

Hlavní vlastnosti:

- Podpora mnoha platforem: Linux, Solaris, OpenBSD, FreeBSD, MacOS X, NetBSD, Windows 2000/XP
- Možnost použití sdíleného klíče a SSL certifikátů
- Podpora režimů klient – klient nebo klient – server
- Při použití komprese vysoká odolnost i na méně kvalitních linkách
- Používá protokol UDP, možno použít i TCP
- Průchodnost přes NAT routery
- Nevýhoda: není šifrována TCP/IP hlavička

a) Obecný popis

Veškerá komunikace probíhá prostřednictvím jediného portu. Snadno tak lze nakonfigurovat pravidla firewallu. Standardně je použit protokol UDP. Lze také použít TCP protokol.

OpenVPN daemon komunikuje přes TAP nebo TUN rozhraní. Pro komunikaci je potřeba provozovat na straně klienta i serveru stejný typ rozhraní. Nelze mít na straně serveru rozhraní TAP a přistupovat ze strany klienta prostřednictvím TUN rozhraní.

K dispozici je možnost volby mezi režimem *klient – klient* nebo režimem *klient – server*. U první možnosti je vytvořen mezi účastníky šifrovaný tunel a probíhá mezi nimi rovnocenné spojení. Volba klient – server umožňuje připojení několika počítačů k jednomu serveru.

b) Bezpečnost

K dispozici je několik úrovní zabezpečení. Je možné nastavit tunel bez jakéhokoliv šifrování vhodný pouze pro odstínění komunikace v lokální síti. Pro vytvoření tunelu v šifrované podobě se využívá certifikátů v podobě veřejného a soukromého klíče. Takto zašifrovaný tunel je pak možné použít v síti Internet.

c) Podrobný popis instalace

Distribuce: Mandriva Linux 2007 kernel 2.6.17

Verze: openvpn-2.0.7-1mdk.i586.rpm

Díky podpoře mnoha platforem je k dispozici několik variant instalace. Následující kroky jsou popsány pro distribuci Mandriva Linux 2007. Na domovských stránkách projektu OpenVPN[10] je uveden postup instalace i pro další distribuce a platformy. Konfigurace a vytvoření certifikačních souborů je téměř shodné na všech distribucích a nemělo by se výrazně odlišovat od následujícího postupu, proto zde uvádím postup pouze pro distribuci Mandriva.

Instalace OpenVPN nabývá několika možností včetně jednoduchého nainstalování pomocí rpm[11] balíčku. Nejjednodušší cestou je stažení archivu

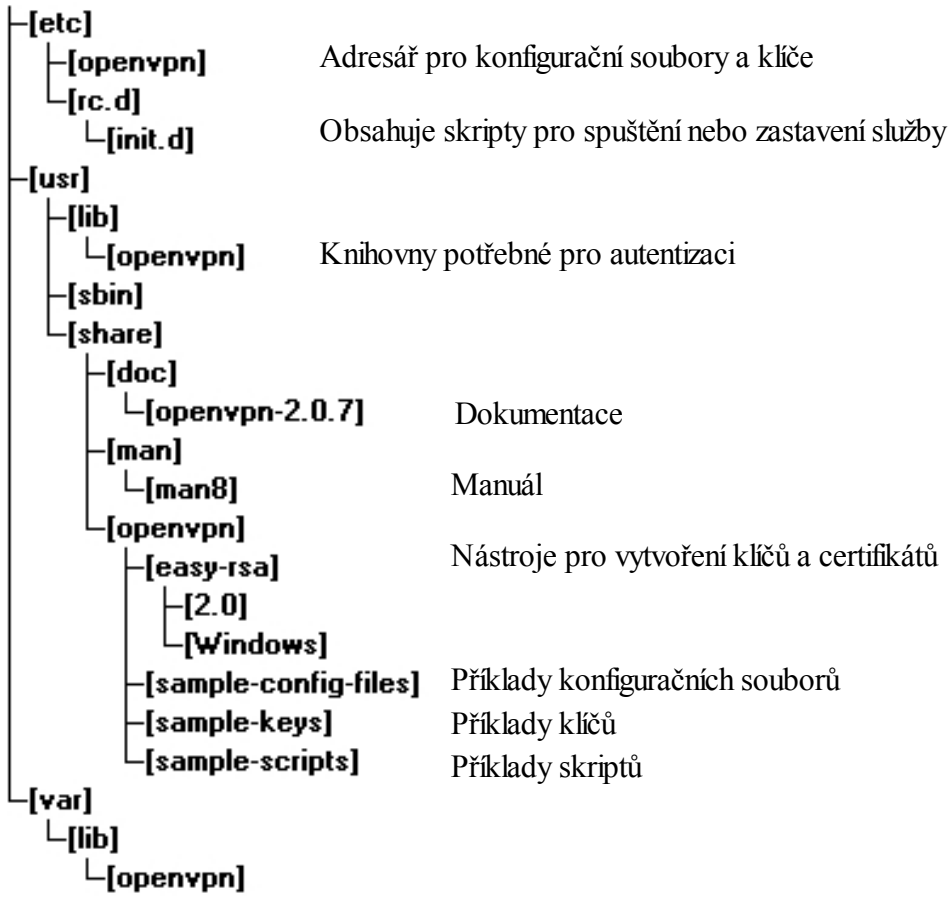
10 OpenVPN [online]. 2006 Dostupné z: <http://openvpn.net/>

11 RPM balíček je archiv, ve kterém se nachází instalovaný software.

z domovské stránky projektu[12] s následnou instalací. Samozřejmě je nutné mít nastavená práva pro instalaci software. Po rozbalení tarballu stačí napsat posloupnost příkazů uvnitř rozbaleného adresáře:

```
./configure

make

make install
```

Po dokončení instalace je připravena struktura adresářů obsahujících soubory nutné ke spuštění, konfiguraci a běhu OpenVPN. Po nainstalování by měla struktura vypadat nějak takto(Obr 3.1):



Obr 3.1: Stromová architektura vytvořená po instalaci

12 http://openvpn.net/download.html

Volba TUN / TAP rozhraní

Zvolený typ rozhraní určuje, jaký typ připojení bude VPN používat. Pro komunikaci typu klient – klient je vhodné rozhraní typu TUN. Pro volbu klient – server se nabízí TAP rozhraní. Vytvoří se tím virtuální Ethernet adaptér.

Vytvoření certifikátů a souborů s klíči

Certifikát musí být podepsán certifikační autoritou (CA) dokazující pravost uživatele. K dispozici je možné využít lokální certifikační autoritu umístěnou přímo na serveru.

V průběhu instalace byl vytvořen současně adresář (/usr/share/doc/openvpn/examples/easy-rsa), kde jsou umístěny skripty pro vytvoření certifikátů a souborů potřebných k autentizaci. Pro přehlednost je vhodné z tohoto umístění překopírovat adresář 2.0 do adresáře etc/openvpn/ a přejmenovat adresář například na easy-rsa. Následně je potřeba v tomto adresáři editovat soubor vars a doplnit potřebné údaje.

```
export EASY_RSA="/etc/openvpn/easy-rsa"
# umístění adresáře
export KEY_CONFIG="$EASY_RSA/openssl.cnf"
# umístění opnessl.cnf
export KEY_DIR="$EASY_RSA/keys"
# adresář kde budou klíče a certifikáty uloženy
export KEY_SIZE=1024
# velikost klíče pro Diffie Hellman
export CA_EXPIRE=3650
# doba platnosti certifikátu (udáváno ve dnech)
export KEY_EXPIRE=3650
# doba platnosti klíče (udáváno ve dnech)
export KEY_COUNTRY="CZ"
export KEY_PROVINCE="Czech"
export KEY_CITY="Liberec"
export KEY_ORG="Firma"
export KEY_EMAIL="muj@mail.cz"
# upřesňující informace
```

Po ukončení editace a uložení je potřeba spustit následující příkazy:

source ./vars

- zavede proměnné, které jsou uvedeny v souboru vars

./clean-all

- odstraní dosavadní soubory z adresáře keys

./build-ca

- vytvoří soubor ca.crt

./build-dh

- vytvoří soubor dh1024.pem (Diffie Hellman)

./build-key-server server

- vytvoří certifikát pro server (server.key)

./build-key client

- vytvoří certifikát určený pro klienta (client.key), důležité je vyplnění položky "common name", která by měla být pro každého klienta jiná

Nyní jsou potřebné soubory a certifikáty vytvořeny. Soubory server.key a client.key jsou tajné klíče, a proto by s nimi mělo být patřičně zacházeno. Soubory je možné kamkoliv umístit. Jedinou podmínkou je uvést správnou cestu do níže popsaného konfiguračního souboru. Pokud cesta nebude uvedena, předpokládá se, že soubory jsou umístěny v adresáři etc/openvpn/.

Konfigurační soubor *server.conf*

Hlavní parametry a chování serveru se udávají prostřednictvím konfiguračního souboru server.conf, který by se měl nacházet v adresáři etc/openvpn/. Po instalaci je připraven ukázkový konfigurační soubor, který je možno editovat v libovolném textovém editoru. Komentáře jsou tvořeny pomocí křížku (#) nebo středníku (;).

Příkazy používané v souboru server.conf

local IPadresa	#IP adresa na které daemon naslouchá
port 1194	#nastavení portu
proto udp / tcp	#volba protokolu
dev tap	#vytvoří rozhraní tap
ca ca.crt	# SSL/TLS root certifikát
cert server.crt	#certifikát
key server.key	#autentizační klíč
dh dh1024.pem	#Diffie-Hellman klíč
server 10.8.0.0 255.255.255.0	#nastavení serveru
ifconfig-pool-persist ipp.txt	#zaznam přiřazení IP adres klientů
server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100	#vytvoření mostu
learn-address ./script	#pravidla firewallu
push "redirect-gateway"	#brána pro klienty
client-to-client	#uživatel vidí uživatele (ne jen server)
duplicate-cn	#přístup jedním certifikátem pro všechny
keepalive 10 120	#ping každých 10 vteřin, pokud není odezva, čeká 120 sekund
cipher BF-CBC	#Blowfish (default)
cipher AES-128-CBC	#AES
cipher DES-EDE3-CBC	#Triple-DES
comp-lzo	#zapnutá komprese
max-clients 100	#max. počet klientů
user nobody	#práva uživatelů
group nobody	#práva skupiny
status openvpn-status.log	#logování (každou minutu)
log-append openvpn.log	#přidání
verb 3	#stupeň logování
mute 20	#opakovné zprávy

Pokud je konfigurační soubor nastaven a je společně s potřebnými klíči a certifikáty uložen v odpovídajícím adresáři (např. etc/openvpn/) je možné spustit daemona. Pro spuštění slouží příkaz:

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

Po správném zadání a nastavení v konfiguračním souboru by se měl zobrazit výpis podobný na obrázku Obr 3.2.

```
[root@Acer openvpn]# openvpn --config /etc/openvpn/server.conf
Sun Feb 18 10:23:02 2007 OpenVPN 2.0.7 i586-mandriva-linux-gnu [SSL] [LZO] [EPOLL] built on Apr 19 2006
Sun Feb 18 10:23:02 2007 Diffie-Hellman initialized with 1024 bit key
Sun Feb 18 10:23:02 2007 WARNING: file 'server.key' is group or others accessible
Sun Feb 18 10:23:02 2007 TLS-Auth MTU parms [ L:1542 D:138 EF:38 EB:0 ET:0 EL:0 ]
Sun Feb 18 10:23:02 2007 TUN/TAP device tun0 opened
Sun Feb 18 10:23:02 2007 /sbin/ifconfig tun0 10.8.0.1 pointopoint 10.8.0.2 mtu 1500
Sun Feb 18 10:23:02 2007 /sbin/route add -net 10.8.0.0 netmask 255.255.255.0 gw 10.8.0.2
Sun Feb 18 10:23:02 2007 Data Channel MTU parms [ L:1542 D:1450 EF:42 EB:135 ET:0 EL:0 AF:3/1 ]
Sun Feb 18 10:23:02 2007 UDPv4 link local (bound): 192.168.1.6:1194
Sun Feb 18 10:23:02 2007 UDPv4 link remote: [undef]
Sun Feb 18 10:23:02 2007 MULTI: multi_init called, r=256 v=256
Sun Feb 18 10:23:02 2007 IFCONFIG POOL: base=10.8.0.4 size=62
Sun Feb 18 10:23:02 2007 IFCONFIG POOL LIST
Sun Feb 18 10:23:02 2007 Test-Client,10.8.0.4
Sun Feb 18 10:23:02 2007 Initialization Sequence Completed
```

Obr 3.2: Výpis při spuštění OpenVPN na serveru

Ve výpisu jsou informace, které byly zadány v souboru server.conf. IP adresa serveru je 10.8.0.1. Server poslouchá na fyzické IP adrese 192.168.1.6 na portu 1194. Jakmile se připojí libovolný oprávněný uživatel, výpis pokračuje dále a informuje o průběhu autentizace s přihlašujícím se uživatelem.

```
Sun Feb 18 10:23:02 2007 Initialization Sequence Completed
Sun Feb 18 10:23:48 2007 MULTI: multi_create_instance called
Sun Feb 18 10:23:48 2007 192.168.1.1:1349 Re-using SSL/TLS context
Sun Feb 18 10:23:48 2007 192.168.1.1:1349 LZO compression initialized
Sun Feb 18 10:23:48 2007 192.168.1.1:1349 Control Channel MTU parms [ L:1542 D:138 EF:38 EB:0 ET:0 EL:0
]
Sun Feb 18 10:23:48 2007 192.168.1.1:1349 Data Channel MTU parms [ L:1542 D:1450 EF:42 EB:135 ET:0 EL:0
AF:3/1 ]
Sun Feb 18 10:23:48 2007 192.168.1.1:1349 Local Options hash (VER=V4): '530fdded'
Sun Feb 18 10:23:48 2007 192.168.1.1:1349 Expected Remote Options hash (VER=V4): '41690919'
Sun Feb 18 10:23:48 2007 192.168.1.1:1349 TLS: Initial packet from 192.168.1.1:1349, sid=c747c99d e7edf8
be
Sun Feb 18 10:23:48 2007 192.168.1.1:1349 VERIFY OK: depth=1, /C=KG/ST=NA/L=BISHKEK/O=OpenVPN-TEST/email
Address=me@myhost.mydomain
Sun Feb 18 10:23:48 2007 192.168.1.1:1349 VERIFY OK: depth=0, /C=KG/ST=NA/O=OpenVPN-TEST/CN=Test-Client/
emailAddress=me@myhost.mydomain
Sun Feb 18 10:23:48 2007 192.168.1.1:1349 Data Channel Encrypt: Cipher 'BF-CBC' initialized with 128 bit
 key
Sun Feb 18 10:23:48 2007 192.168.1.1:1349 Data Channel Encrypt: Using 160 bit message hash 'SHA1' for HM
AC authentication
Sun Feb 18 10:23:48 2007 192.168.1.1:1349 Data Channel Decrypt: Cipher 'BF-CBC' initialized with 128 bit
 key
Sun Feb 18 10:23:48 2007 192.168.1.1:1349 Data Channel Decrypt: Using 160 bit message hash 'SHA1' for HM
AC authentication
Sun Feb 18 10:23:48 2007 192.168.1.1:1349 Control Channel: TLSv1, cipher TLSv1/SSLv3 DHE-RSA-AES256-SHA,
 1024 bit RSA
Sun Feb 18 10:23:48 2007 192.168.1.1:1349 [Test-Client] Peer Connection Initiated with 192.168.1.1:1349
Sun Feb 18 10:23:48 2007 Test-Client/192.168.1.1:1349 MULTI: Learn: 10.8.0.6 -> Test-Client/192.168.1.1:
1349
Sun Feb 18 10:23:48 2007 Test-Client/192.168.1.1:1349 MULTI: primary virtual IP for Test-Client/192.168.
1.1:1349: 10.8.0.6
Sun Feb 18 10:23:49 2007 Test-Client/192.168.1.1:1349 PUSH: Received control message: 'PUSH_REQUEST'
Sun Feb 18 10:23:49 2007 Test-Client/192.168.1.1:1349 SENT CONTROL [Test-Client]: 'PUSH_REPLY,route 10.8
.0.1,ping 10,ping-restart 120,ifconfig 10.8.0.6 10.8.0.5' (status=1)
```

Obr 3.3: Výpis po připojení uživatele k serveru

Nastavení pro Firewall

Důležitým prvkem v udržení bezpečnosti bude přesné nadefinování pravidel firewallu. Nejvíce útoků na odhalení komunikace bude mířeno právě na koncové body vytvořeného tunelu, proto je třeba mít dobře nastavený firewall.

Pro zprovoznění komunikace je potřeba přidat další pravidla pro nově vzniklá zařízení a usměrnění komunikace mezi uživateli. Firewall představuje program iptables.

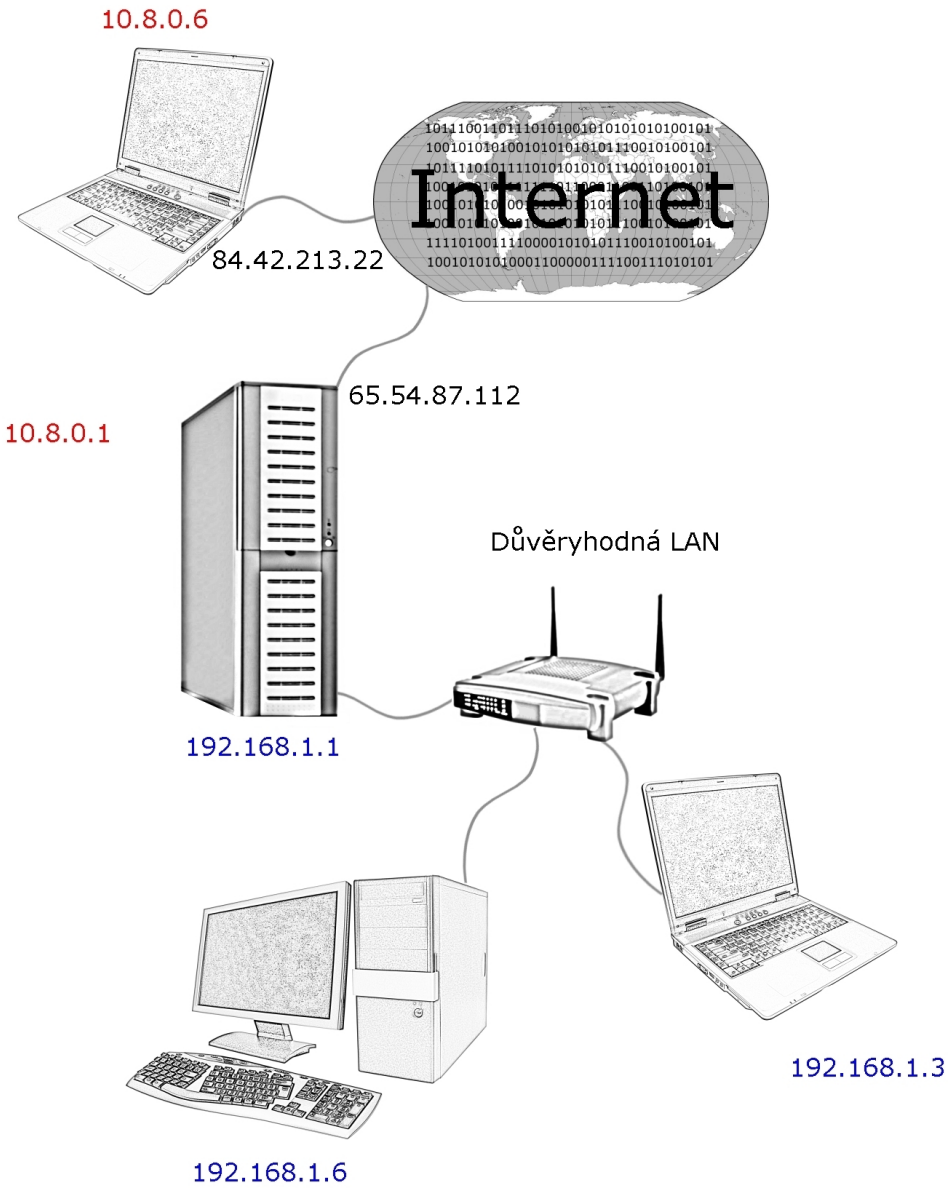
Příkazy pro nastavení:

```
# předpokládám nastavení OUTPUT ACCEPT,  INPUT+FORWARD DROP

iptables -A INPUT -p udp --dport 1196 -j ACCEPT
#povolení UDP portu

iptables -A INPUT -i tun+ -j ACCEPT
iptables -A FORWARD -i tun+ -j ACCEPT
iptables -A INPUT -i tap+ -j ACCEPT
iptables -A FORWARD -i tap+ -j ACCEPT
# povolení TUN/TAP rozhraní

iptables -A INPUT -i eth1 -j ACCEPT
iptables -A FORWARD -i eth1 -j ACCEPT
# povolení z podsítě
```

Nyní by měla být konfigurace serveru nastavena a komunikace by měla probíhat podle Obr 3.4. Uživatel připojující se prostřednictvím internetu k síti LAN musí nejdříve navázat spojení se serverem. IP adresa uživatele může být libovolná (záleží na nastavení firewallu, pokud by se uživatel připojoval z jednoho místa s pevnou IP adresou, je možné nastavit ve firewallu pevné pravidlo a zajistit tak vyšší bezpečnost). Na straně serveru je potřeba mít přímo viditelnou IP adresu pro připojující se uživatele. Po vykonání autentizační procedury je přidělena uživateli IP adresa pro virtuální rozhraní. Server a klient spolu komunikují prostřednictvím šifrovaného tunelu.

Na serveru je nastaveno přemostění virtuálního rozhraní na fyzické ethernetové rozhraní umožňující komunikaci s uživateli připojenými do důvěryhodné sítě LAN. Vzdálený uživatel může také využívat sdílené služby na jednotlivých počítačích v LAN (tisk, sdílené dokumenty apod.).

Samozřejmostí je možnost připojit se ke službám běžících na serveru (FTP, pošta atd.). Pokud budou nastavena pravidla firewallu tak, že k těmto službám budou mít přístup pouze uživatelé z vytvořeného virtuálního rozhraní a sítě LAN, získáme tak další a hlavně vyšší stupeň zabezpečení při využívání těchto služeb.



Obr 3.4: Model komunikace podle nastavení serveru

3.2. OpenSWAN (IPSec)

a) Obecný popis

Projekt OpenSWAN je implementace protokolu IPSec pro Linux. Poskytuje široké možnosti nastavení při konfiguraci. Je možné vytvořit plnohodnotné připojení mezi servery a také umožnit přístup do sítě mobilním uživatelům. Problémy s průchodností přes NAT byly vyřešeny způsobem NAT-T[13], který přidá UDP hlavičku s UDP porty určujícími adresaci klienta.

b) Bezpečnost

Projekt OpenSWAN využívá protokol IPSec, což je zárukou vysokého stupně zabezpečení. Podporovány jsou certifikáty X.509 i šifra RSA s veřejným klíčem.

c) Popis instalace

Distribuce: Mandriva Linux 2007 kernel 2.6.17

Verze: openswan-2.4.5-2mdv2007.0.i586.rpm

Obdobně jako u OpenVPN projektu nabízí také OpenSWAN několik možností instalace odlišných na konkrétní distribuci. Ve většině případů je opět instalace velice podobná. Na domovských stránkách projektu OpenSWAN[14] je možné potřebné instalační soubory získat. Konfigurace souborů jsou téměř shodné na všech distribucích a neměly by se výrazně odlišovat od následujícího postupu, proto zde uvádím postup pouze pro distribuci Mandriva.

Instalace nabývá několika možností včetně jednoduchého nainstalování pomocí rpm balíčku. Samozřejmostí je nastavenost práv pro instalaci software.

13 Network Address Translation-Traversal

14 OpenSWAN [online]. 2006 Dostupné z: <http://www.openswan.org/>

Pokud je instalace, obdobná jako u OpenVPN, dokončena, následuje vytvoření konfiguračního souboru na straně serveru. Soubor je potřeba mít umístěn v adresáři /etc pod názvem ipsec.conf. V konfiguračním souboru jsou používány následující parametry a příkazy:

```
Interfaces = "ipsec0=eth0"
# udává, které síťové rozhraní je asociováno s IPSec
conn jmeno_tunelu
# nadefinování tunelu
type=tunnel
#bude používán tunelovací mód
left=1.2.3.4
# (lokální) veřejná IP adresa na jedné straně tunelu
leftsubnet=192.168.0.0/24
# za IP adresou 1.2.3.4 se nachází podsíť 192.168.0.0/24
leftnexthop=192.168.0.5
# většinou výchozí brána
leftupdown=/etc/ipsec.d/leftup
# skript spuštěný po aktivaci tunelu
right=194.15.7.21
# veřejná IP adresa vzdáleného konce tunelu
rightsubnet=192.168.1.0/24
#obdoba leftsubnet
rightupdown=/etc/ipsec.d/rightup
# obdoba leftupdown
auto=start
#po spuštění IPSec dojde k definování tunelu
```

authby=rsasig

autentizace proběhne pomocí digitálního RSA podpisu, další možností je volba secret, pro sdílené klíče

leftrsasigkey=5SkANy ....

veřejný RSA klíč na levé straně tunelu[15]

rightrsasigkey=3gHkL...

veřejný RSA klíč na levé straně tunelu[16]

pokud je zvoleno authby=secret jsou používány tyto další parametry

ike=3des-md5

nastavuje se metoda šifrování a hashování

esp=3des-md5

nastavení parametrů pro quick mód (stejny jako esp)

keyexchange=ike

#sestavení tunelu pomocí ike

Pokud je vše správně nastaveno a soubor uložen, je možné spustit daemona příkazem:

etc/init.d/ipsec start

Správnou funkčnost a vytvoření tunelu můžeme otestovat příkazem:

etc/init.d/ipspec status

15 Ke zjištění slouží příkaz: ipsec showhostkey --left

16 Ke zjištění slouží příkaz: ipsec showhostkey --right

4. Přístup z klientských počítačů

4.1. Hlavní požadavky

Vycházím z principu komunikace uvedeného na Obr 3.4, kdy uživatelé se připojují prostřednictvím virtuální privátní sítě k serveru. Uvedený popis konfigurace bude vytvořen pro platformu Linux (distribuce Mandriva) a Microsoft Windows XP Home SP2.

Hlavním cílem bylo dosažení snadného připojení klienta k serveru s minimální konfigurací a nutností nastavování. Požadovaným kroky je pouze vykonání patřičné autentizace pro navázání spojení, nastavení firewallu a síťového zařízení. Řízení komunikace a přiřazování IP adres potřebných ke komunikaci je zajištěno serverem.

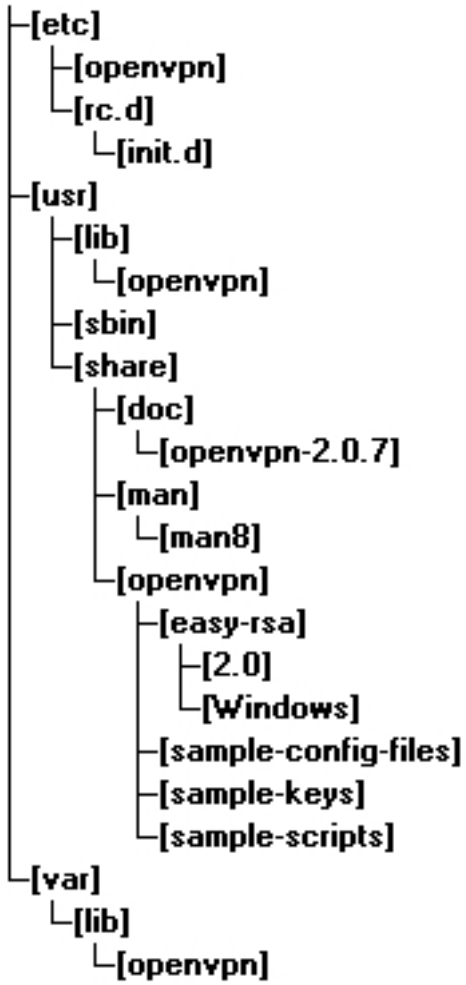
4.2. Přístup pomocí VPN z prostředí Linux

a) OpenVPN klient

Instalace OpenVPN klienta se v podstatě nijak neliší od instalace na straně serveru. Po nainstalování rpm balíčku nebo při rozbalení tarballu s následným vykonáním příkazů:

```
./configure
make
make install
```

vzniká již známá adresářová struktura Obr 4.1.



Obr 4.1: Stromová architektura vytvořená po instalaci

Konfigurační soubor client.conf

Hlavní změny a nastavení chování klienta připojujícího se do sítě musí být nastaveny v konfiguračním souboru client.conf.

Příkazy používané v souboru client.conf

```
client                          #klientský mód
dev tap                         #vytvoří rozhraní tap
proto udp                       #volba protokolu
remote 192.168.1.6 1194         #adresa serveru a číslo portu
resolv-retry infinite           #čas
nobind                          #specifické lokální porty
persist-key
persist-tun
ca ca.crt                       #certifikat
cert client.crt                 #certifikat
key client.key                  #klíč, heslo, lze nastavit pomocí Change
                                Password
comp-lzo                        #zapnutá komprese
verb 3                          #stupeň logování
mute 20                         #opakované zprávy
```

Zabezpečení

Pomocí klíče

Klíč získaný ze strany serveru je možné umístit kamkoliv. Umístění musí být ale zohledněno v konfiguračním souboru.

Pomocí certifikátu

Soubory[17] ca.crt, client.key a client.crt.

17 Vytvořené na straně serveru

Nastavení pravidel pro firewall

K bezproblémové komunikaci po vytvořeném šifrovaném spojení se serverem je potřeba také upravit pravidla firewall. Použité příkazy pro nastavení jednotlivých podsítí vytvořené VPN:

```
# předpokládám nastavení OUTPUT ACCEPT,  INPUT+FORWARD DROP

iptables -A INPUT -p udp --dport 1196 -j ACCEPT
#povolení UDP portu

iptables -A INPUT -i tun+ -j ACCEPT
iptables -A FORWARD -i tun+ -j ACCEPT
iptables -A INPUT -i tap+ -j ACCEPT
iptables -A FORWARD -i tap+ -j ACCEPT
# povolení TUN/TAP rozhraní
```

b) OpenSWAN klient

Instalace na straně klienta se shoduje s instalací na serveru. Propojené jsem měl dva servery, proto není označení klient zcela přesné. Na obou stranách tunelu byly umístěny servery, za kterými byly umístěny další podsítě, které jsem propojoval. Vytvořený tunel byl pouze mezi servery a jen po tranzitní síti Internet. Konfigurační soubory mohou být na obou stranách shodné, OpenSWAN sám pozná, která strana tunelu je pravá a levá (right, left) podle lokální adresy. Stačí pouze změnit případné rozdíly v síťových rozhraních. Průběh instalace a konfigurace je uveden v nastavení na straně serveru, proto zde nebudu instalaci více popisovat.

4.3. Přístup pomocí VPN z prostředí MS Windows

a) OpenVPN klient

Verze: openvpn-2.0.9-gui-1.0.3-install.exe

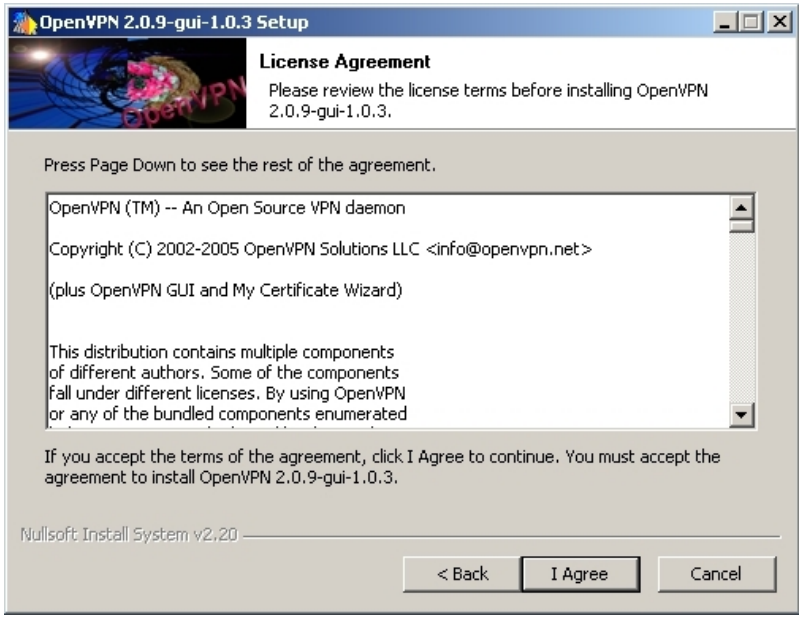
Potřebný balíček je možné stáhnout na stránkách http://openvpn.net/ v sekci download. K dispozici je verze s grafickým uživatelským rozhraním nebo jen konzolová verze. Principielně jsou obě verze shodné.

Základem je vytvoření virtuálního síťového adaptéru, který bude propojen se serverem prostřednictvím vytvořeného tunelu. Spojení se řídí podle konfiguračního souboru client.ovpn.

Po stažení souboru stačí standardně dodržovat postup instalace jako u jiných aplikací. Instalátor provede všechny potřebné kroky bez nutnosti nějakých složitých zásahů ze strany uživatele. Na úvodní obrazovce zvolte Další (Next).

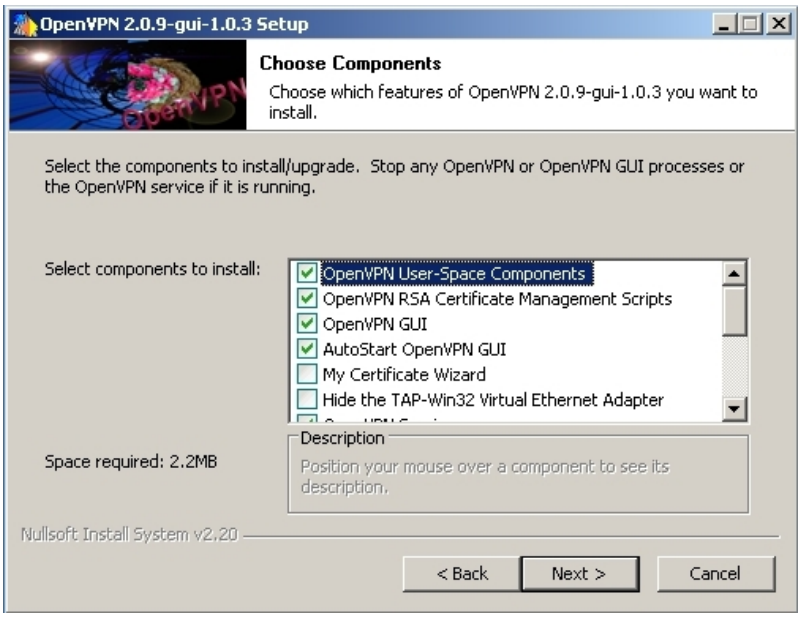
Obr 4.2: Průběh instalace

Následně je potřeba přečíst a souhlasit s licenčním ujednáním (I Agree).



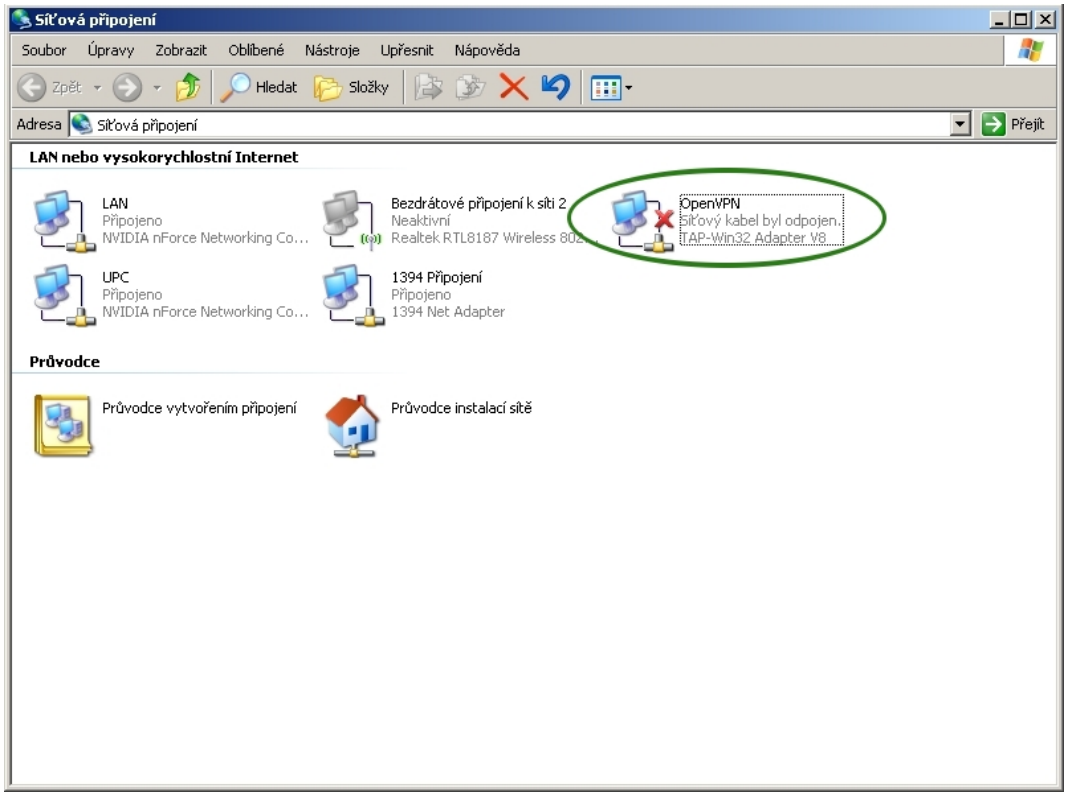
Obr 4.3: Průběh instalace

Po odsouhlasení se dostáváme k oknu pro volbu součástí. Důležité části jsou zahrnuty v defaultním nastavení.



Obr 4.4: Průběh instalace

Po dokončení instalace je vytvořeno nové síťové rozhraní TAP -Win32 Adapter V8.

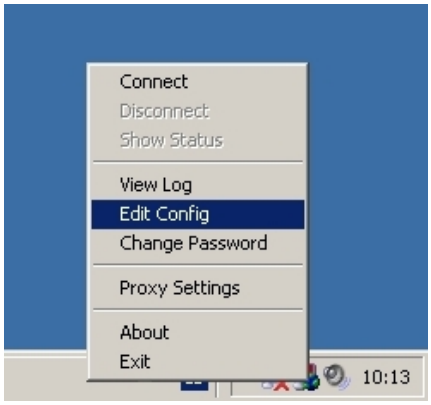


Obr 4.5: Síťová připojení

Po nainstalovaní se rovněž zobrazuje grafické rozhraní pro OpenVPN v systémové liště v podobě dvou dočasně červených monitorů.

Obr 4.6: Grafické rozhraní

Po kliknutí pravým tlačítkem myši se rozbalí nabídka umožňující editaci konfiguračního souboru client.ovpn. Ostatní položky nejsou zatím důležité, protože bez nakonfigurování souboru client.ovpn se není možné připojit k serveru.



Obr 4.7: Nabídka možností

Zabezpečení

Pomocí klíče

Klíč získaný ze strany serveru je možné umístit kamkoliv. Umístění musí být ale zohledněno v konfiguračním souboru.

Pomocí certifikátu

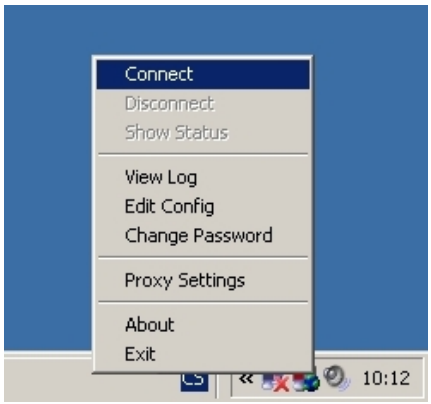
Soubory[18] ca.crt, client.key a client.crt.

Příkazy používané v souboru *client.ovpn*

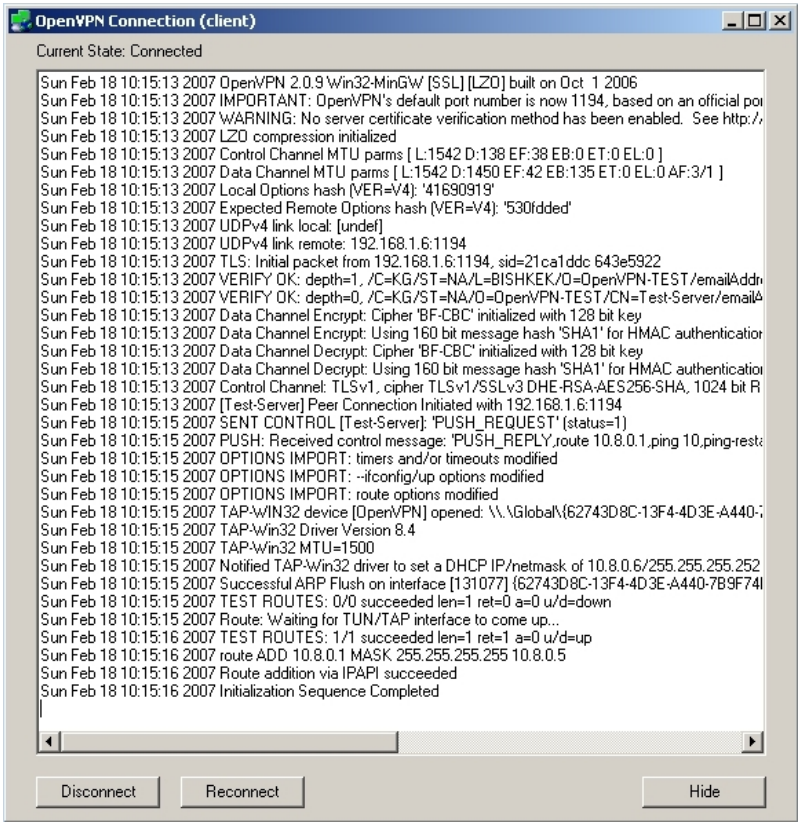
client	#klientský mód
dev tap	#vytvoří rozhraní tap
proto udp	#volba protokolu
remote 192.168.1.6 1194	#adresa serveru a číslo portu
resolv-retry infinite	#čas
nobind	#specifické lokální porty
persist-key	
persist-tun	
ca ca.crt	#certifikat
cert client.crt	#certifikat
key client.key	#klíč, heslo, lze nastavit pomocí Change Password
comp-lzo	#zapnutá komprese
verb 3	#stupeň logování
mute 20	#opakované zprávy

18 Vytvořené na straně serveru

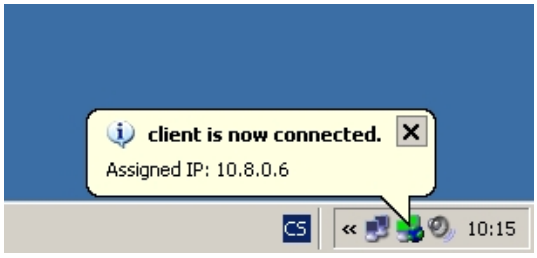
Po správném nakonfigurování je možné zvolit volbu Connect. Následně se otevře Status okno zobrazující informace o připojení k serveru. Po úspěšném navázání spojení je na systémové liště oznámena přidělená IP adresa.



Obr 4.8: Volba připojení



Obr 4.9: Průběh připojení

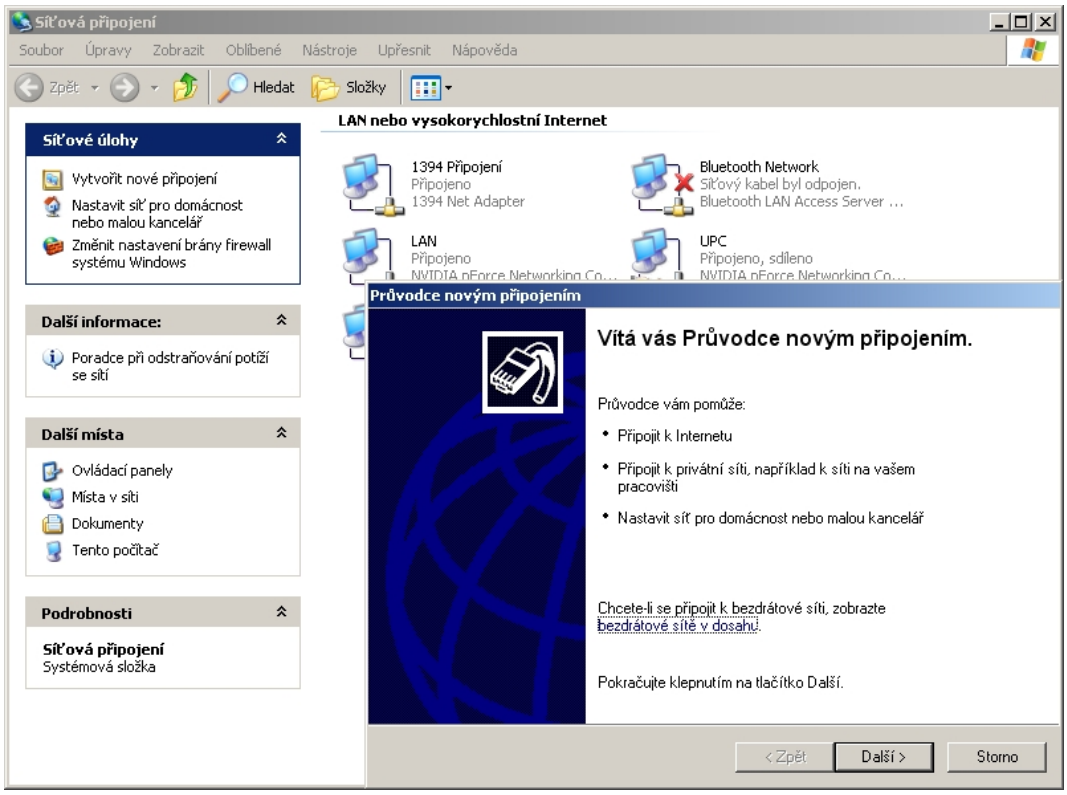


Obr 4.10: Oznámení připojení

b) FreeS/WAN klient

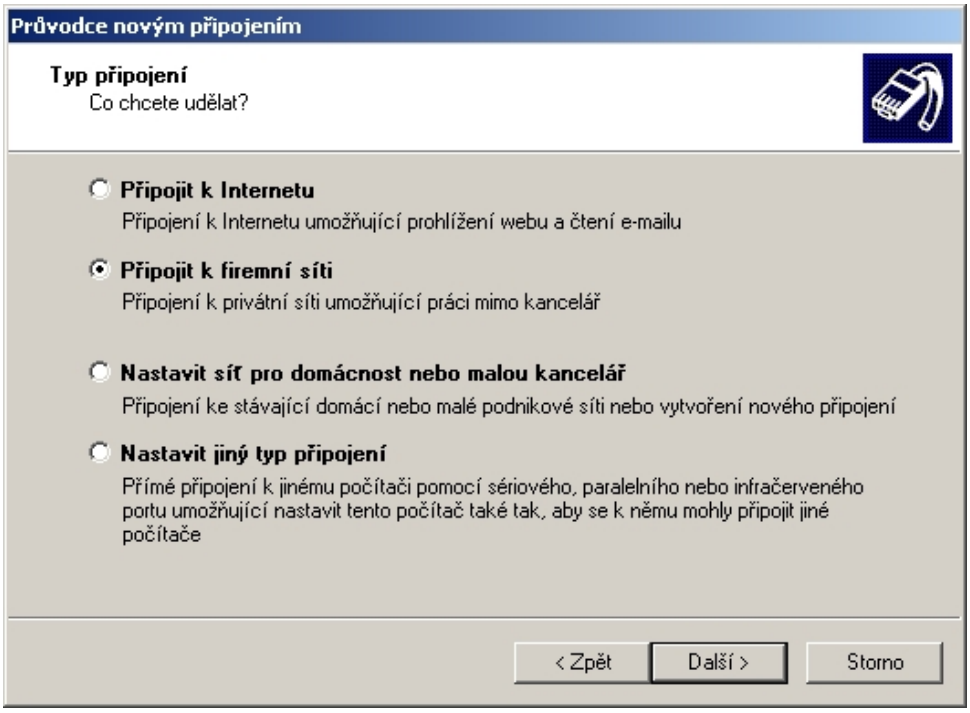
Pro navázání komunikace využijeme klienta standardně obsaženého v Microsoft Windows využívajícího L2TP protokolu. Pro vytvoření nového připojení se nabízí využít průvodce, který nás povede k úspěšnému navázání spojení.

Z nabídky *Start* přes možnost *Připojit* se dostaneme k dialogu *Síťová připojení*, kde vybereme volbu *Vytvořit nové připojení*.



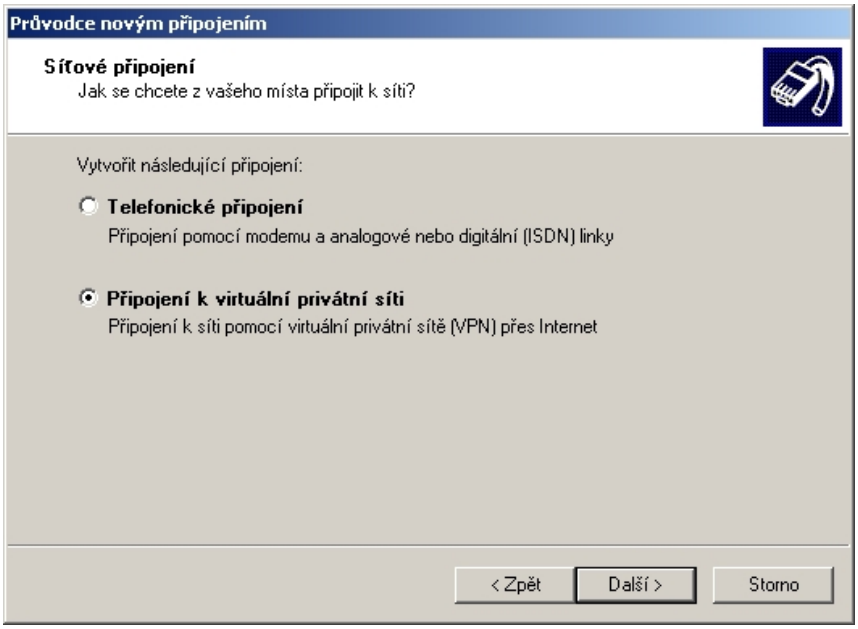
Obr 4.11: Síťová připojení

Po zvolení tlačítka *Další* se otevře okno, kde vybereme možnost *Připojení k firmení síti*.



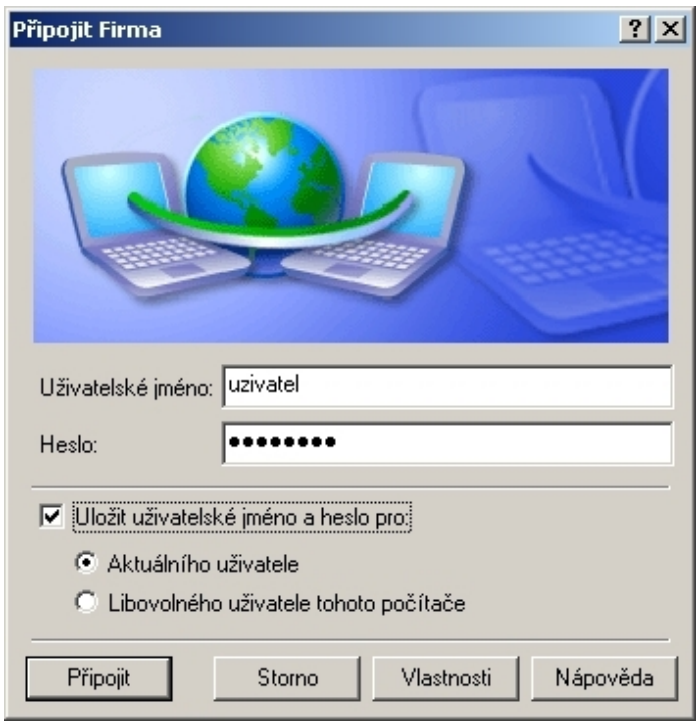
Obr 4.12: Průvodce novým připojením

Po zvolení a pokračování v instalaci v nasledujícím okně zvolíme *Připojení k virtuální privátní síti.*



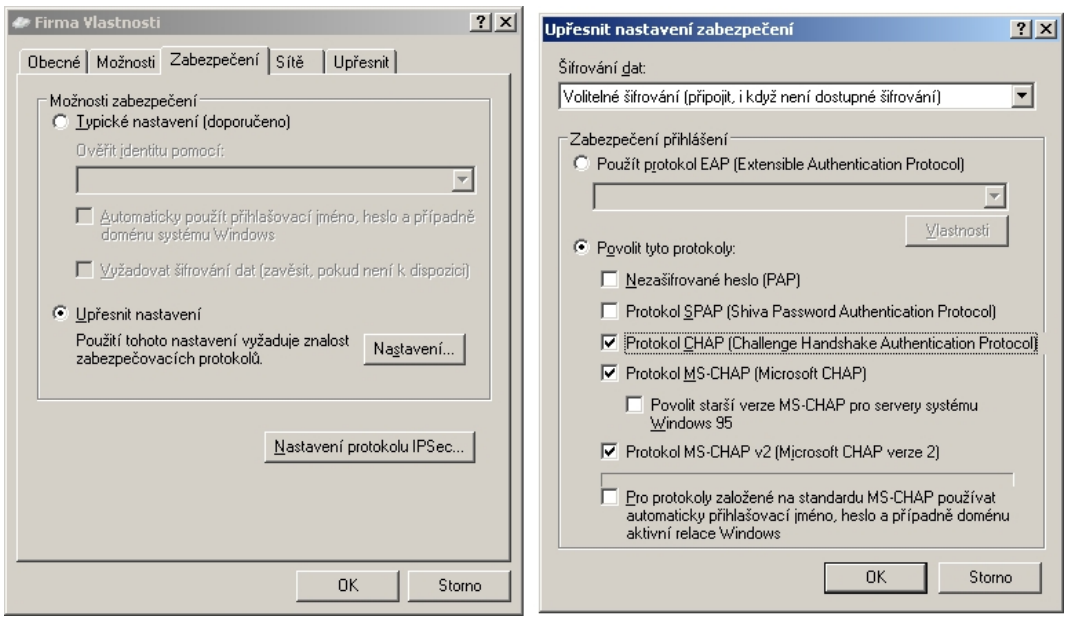
Obr 4.13: Volba typu sítě

Následují okna s dotazem na název sítě, který slouží pouze pro pojmenování sítě na klienstkém počítači a dotaz na IP adresu serveru. Po vyplnění se zobrazí okno s možností připojení do sítě. Před navázaním komunikace se musí ještě zvolit typ autentizace a další upřesňující nastavení, která provedeme pomocí volby *Vlastnosti.*



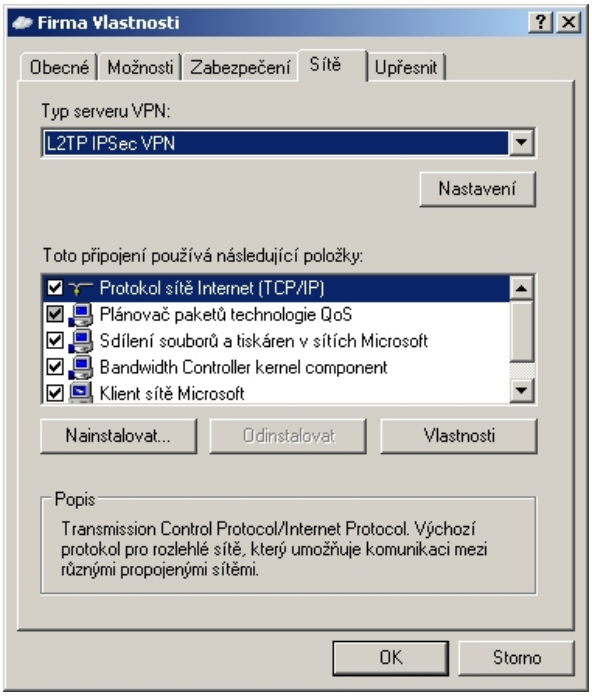
Obr 4.14: Nabídka připojení

Následně je k dispozici dialogové okno ze záložkami. Začneme na záložce Security, kde zvolíme checkbox na možnost *Upřesnit nastavení* a klikneme na tlačítko *Nastavení* . Zvolíme a začkrtneme patřičné volby potřebné k úspěšné autorizaci.



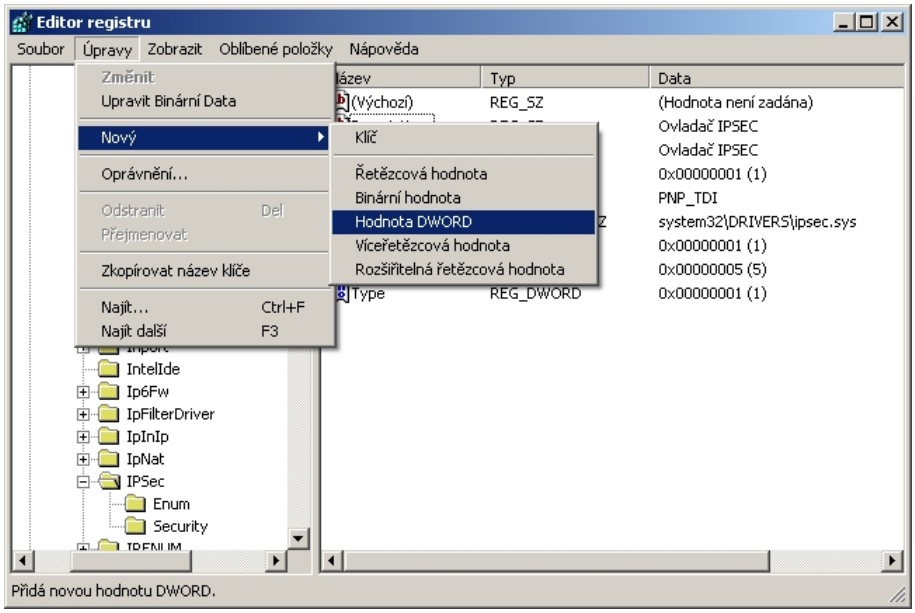
Obr 4.15: Nastavení zabezpěčení

V záložce Sítě pak již jen zbývá zvolit typ serveru na L2TP IPSec VPN a zvolit *OK*

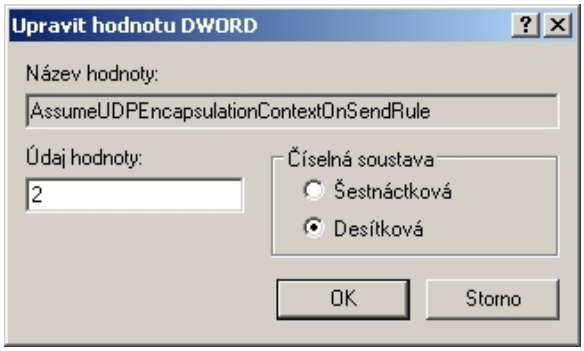


Obr 4.16: Typ serveru

Pokud je uživatel umístěn za NAT[19], je nutné provést přidání položky do seznamu registrů. Jedná se o hodnotu DWORD(2) s názvem *AssumeUDPEncapsulationContextOnSendRule* u místěnou v *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IPSec.* Do registrů se dostanem z položky *Spustit* v nabídce *Start* a zadáním příkazu *regedit*.



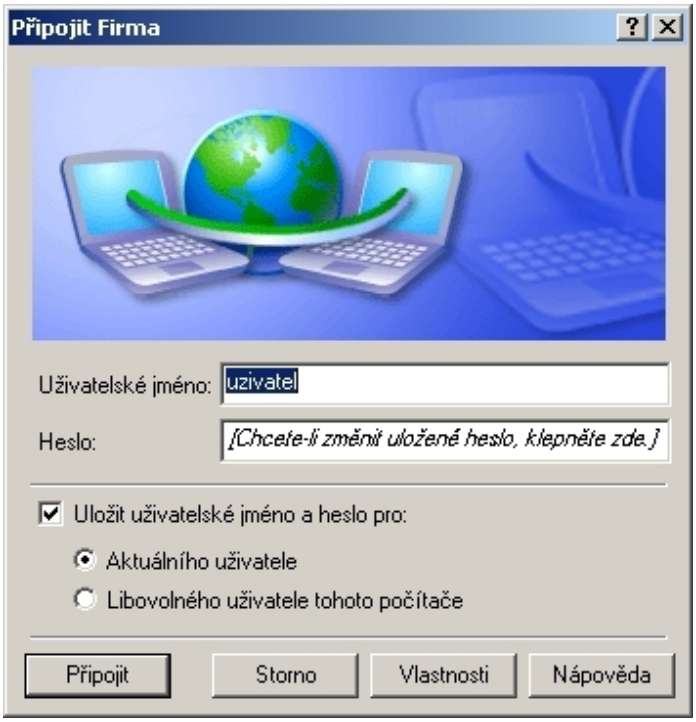
Obr 4.17: Přidání položky



Obr 4.18: Hodnota DWORD

19 Network address translation - překlad síťových adres

Nyní by mělo být vše nastaveno a v dialogu je možné zvolit tlačítko pro připojení.



Obr 4.19: Dialog připojení

5. Porovnání

OpenVPN

Zástupce SSL VPN řešení disponuje snadnou instalací a pohodlnou konfigurací. Je umožněno zvolit mezi několika stupni zabezpečení. Výhodou je vysoká průchodnost sítě včetně proxy serverů či NAT zařízení.

OpenVPN bych volil pro vytvoření virtuální privátní sítě, kdy je vytvořen jeden server, ke kterému se připojují mobilní uživatelé. Vycházím z předpokladu, že většina uživatelů bude připojena za proxy servery nebo NAT zařízeními, se kterými si IPSec bez spolupráce dalších protokolů neporadí.

OpenSWAN

Projekt OpenSWAN plně využívá výhod IPSec protokolu. IPSec protokol zajišťuje nejen bezpečný a šifrovaný tunel určený pro komunikaci, ale také sleduje probíhající komunikaci a zaznamená případné pokusy o prolomení šifrování. Nespornou výhodou je také možnost navázání komunikace i s hardwarovými klienty v podobě zařízení firmy Cisco. Podpora IPSec je implementována i přímo do Windows, a tak připojení vzdálených uživatelů připojujících se z této platformy není podmíněna instalací klienstkého software.

OpenSWAN je vhodný pro vytvoření tunelu mezi koncovými zařízeními[20] s přímo viditelnými adresami. Šifrovaný tunel je vytvořen pouze přostřednictvím IPSec, a tak je zaručena vysoká bezpečnost při probíhající komunikaci.

20 server – server nebo server – klient

6. Závěr

V této diplomové práci jsem nastínil současnou problematiku vytvoření virtuální privátní sítě. Uvedl jsem seznam často používaných metod a protokolů se stručným popisem jejich vlastností. Jednotlivé protokoly nejsou detailně probrány, neboť rozbor by vydal na několikastránkovou dokumentaci pouze k jedinému protokolu.

Při návrhu jsem se zaměřil jen na dostupné projekty s otevřeným zdrojovým kódem. Sítě byly vytvořeny pomocí odlišných protokolů, ve snaze porovnat složitost implementace a konfigurace, pro vytvoření stejného modelu virtuální privátní sítě. Právě díky rozdílnosti využívaných technologií nelze jednoznačně určit, která metoda je nejvýhodnější. Záleží na konkrétních požadavcích a představách sítě. Při tvorbě firemní sítě bych volil provozování projektů současně, a tak spojil výhody obou využívaných technologií pro dílčí způsoby komunikace.

Oba projekty jsou již několik let vyvíjeny a zdokonalovány, ale není možné vytvořit v bezpečnosti nepřekonatelnou síť.

Literatura

Monografie:

[1] Dostálek, L. - Kabelová, A. Velký průvodce protokoly TCP/IP a systémem DNS. 4. vyd. Brno: CP Books, a.s., 2005. ISBN 80-722-6675-6.

[2] Feilner, M. OpenVPN: Building and Integrating Virtual Private Networks. 1.vyd. Birmingham: Packt Publishing, 2006. ISBN 1-904811-85-X.

[3] Pužmanová, R. TCP/IP v kostce. 1. vyd. České Budějovice: Kopp, 2004. ISBN 80-7232-236-2.

[4] Scott, Ch. - Wolfe, P. - Erwin, M. Virtual Private Networks. 2. vyd. Sebastopol: O´Reilly, 2000. ISBN 1-56592-529-7.

[5] Tiller, J.S. A Technical Guide to IPSec Virtual Private Networks. 1. vyd. Boca Raton: CRC Press LLC, 2001. ISBN 0-8493-0876-3.

Periodika

[6] Hills, R. Odhalení a fingerprinting systémů IPsec VPN. Hakin9, 2005, č.6, s. 52-62.

Elektronické publikace

[7] Hladík, R.: OpenVPN - VPN jednoduše [online]. 2004, [cit. 2004-10-11]. Dostupné z: <http://www.root.cz/clanky/openvpn-vpn-jednoduse/>.

[8] Luhový, K.: Virtuální privátní sítě VPN [online]. 2003 [cit. 2003-02-18]. Dostupné z: <http://www.svetsiti.cz/view.asp?rubrika=Tutorialy&temaID=219&clanekID=230>.

[9] OpenSWAN [online]. 2006 Dostupné z: <http://www.openswan.org/>

[10]OpenVPN [online]. 2006 Dostupné z: <http://openvpn.net/>

[11]Pužmanová, R.: Vývoj paketových sítí a postavení MPLS [online]. 2006, [cit. 2006-07-24]. Dostupné z: <http://www.svetsiti.cz/view.asp?rubrika=Technologie&clanekID=302>.

[12]Pužmanová, R.: Bezpečnost ve VPN: IPSec versus SSL [online]. 2006, [cit. 2006-09-12]. Dostupné z: <http://www.dsl.cz/clanky.php?clanek=515>.

Přílohy

Seznam příloh

Příloha 1: soubor Server.conf

```
mode server
tls-server
local 65.54.87.112

port 1194
proto udp
dev tap

ca ca.crt
cert server.crt
key server.key
dh dh1024.pem

server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
inactive 3600

push "route-delay 3 5"
push "route-gateway 10.8.0.1"
push "route 192.168.1.0 255.255.255.0"
push "dhcp-option DNS 192.168.1.1"
push "dhcp-option WINS 192.168.1.1"

client-to-client

learn-address ./script
comp-lzo
status openvpn-status.log
verb 3
mute 20
```

Příloha 2: soubor Client.conf

```
client

dev tap
proto udp

remote 65.54.87.112 1194

resolv-retry infinite

pull
tls-client

nobind

ca ca.crt
cert client.crt
key client.key

comp-lzo
verb 3
mute 20
```

Příloha 3: soubor IPSec.conf

```
config setup
        interfaces="ipsec0=eth1"
        klipsdebug=none
      plutodebug=none


conn tunel
        type=tunnel

        left=65.54.87.112
        leftsubnet=192.168.1.0/24
        leftnexthop=192.168.1.1
        leftupdown=/etc/ipsec.d/leftup

        right=84.42.213.22
        rightsubnet=192.168.2.0/24
        rightnexthop=192.168.2.1
        rightupdown=/etc/ipsec.d/leftup

        authby=rsasig
        leftrsasigkey=GdFHc
        rightrsasigkey=QwHjL

        auto=start

include /etc/ipsec.d/examples/no_oe.conf
```