ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

PROVOZNĚ EKONOMICKÁ FAKULTA

Katedra informačního inženýrství
obor Informatika

BAKALÁŘSKÁ PRÁCE

Virtuální privátní sítě - OpenVPN

Petr Zlatník

Vedoucí práce:
Ing. Martin Papík

Prohlášení

Prohlašuji, že jsem bakalářskou práci na téma „Virtuální privátní sítě - OpenVPN“ vypracoval samostatně za použití uvedených zdrojů a citované pasáže vyznačil spolu s uvedením citovaného pramene.

V Praze dne 23. 4. 2009

.....

Poděkování

Rád bych touto cestou poděkoval Ing. Martinu Papíkovi za cenné rady, pozitivní přístup, a ochotu při vedení mé bakalářské práce.

Virtuální privátní sítě – OpenVPN

Souhrn

Práce prezentuje základní principy technologie virtuálních privátních sítí a mapuje jednotlivé protokoly, které jsou používané k jejich realizaci. Také seznamuje s používanými prvky VPN sítí. Pozornost je věnována především SSL VPN, které se v poslední době masivně prosazují. Konkrétně implementaci OpenVPN, na níž je demonstrována jednoduchost instalace a konfigurace tohoto protokolu

Klíčová slova

Virtuální privátní síť - VPN, šifrování, autentifikace, integrita, firewall, IPSec, SSL/TLS, server, klient, tunel, IP, TCP, UDP, zapouzdření, směrovač, přemostění, maska, protokol, paket, hashování, certifikát, klíč, MPLS, GRE, OpenVPN, zasíťování

Virtual private Networks

Summary

Thesis presents basic principles of technology Virtual Private Networks and describes protocols, which is used to its implementation. Also introduce elements of VPN networks. Attention is intent especially on SSL VPNs, which is recently trusted forward. Specifically on OpenVPN implementation illustrating easy of installation, configuration and use these protocol.

Keywords

Virtual Private Network – VPN, cryptography, authentication, integrity, firewall, IPSec, SSL/TLS, server, client, tunnel, IP, TCP, UDP, encapsulation, router, bridging, mask, protocol, packet, hashing, certificate, key, MPLS, GRE, OpenVPN, networking

Obsah

1 Úvod

V dnešním globalizovaném světě je mnoho společností, které mají velké množství poboček rozmístěných po celém světě. Zároveň je jasné, že musí být centrálně spravovány, organizovány a vzájemně propojeny. V dřívějších dobách se spojení vzdálených lokalit realizovalo především za pomoci pronájmu dedikovaných linek. S příchodem internetu se objevila možnost přístupu k levnému a veřejnému přenosovému médiu, které však neposkytuje ochranu pro citlivá data. Aby ho bylo možné používat stejným způsobem jako dedikované spoje, bylo jej nutné zabezpečit proti aktivním i pasivním útokům.

Prvním masivně rozšířeným nástrojem pro vytváření virtuálních privátních sítí byl a stále je IPSec. Jeho největší nevýhodou je obtížná konfigurace v rámci OS a nutnost manuálního nastavení každého stroje. Kvůli tomu je správa často přenechávána bránám, které mají tuto technologii implementovanou. Jenomže v dnešní době se stále více pracovníků pohybuje „v terénu“ a proto je nezbytná instalace nějakého nástroje pro VPN do jejich počítače. Zmíněnou mezeru řeší VPN na bázi SSL resp. TLS. Operují v uživatelském režimu, nabízejí snadnou instalaci a konfiguraci. Většina komerčních VPN na této bázi dokonce umožňuje navázání a tvorbu šifrovaného spojení pomocí Java appletů bez nutnosti jakékoli instalace do klientského počítače.

Pro konstrukci virtuálních privátních sítí je třeba zajistit celou řadu služeb. Tvorba nespočívá pouze v symetrickém šifrování, ale obsahuje také prostředky pro zabezpečení autentifikace jednotlivých účastníků, ověření integrity dat, tvorbu digitálních podpisů a další.

Práce mapuje obecně technologii virtualizace privátních sítí, popisuje jednotlivé techniky a protokoly používané při tvorbě zabezpečeného vzdáleného spojení. Zároveň demonstruje jednoduchost *open source* řešení OpenVPN, které získalo v roce 2007 cenu pro nejlepší SSL implementaci virtuální privátní sítě v rámci volně šiřitelných programů. Začíná se masivně prosazovat a v dnešní době ho používá přes tři miliony uživatelů.

2 Cíl práce a metodika

Cílem práce je popsat principy virtuálních privátních sítí, rozlišit jednotlivé protokoly a uvést možnosti jejich využití v oblasti vzdálené konektivity. Hlavní část prezentuje vývoj, referuje o současné situaci a nastiňuje možnosti budoucího využití virtualizace privátních sítí na základě dosavadního vývoje a současných trendů. Tento materiál bude v elektronické podobě součástí dobrovolného semináře bezpečnosti a může tak sloužit k prezentaci VPN odborné veřejnosti.

Na úvod je vymezen pojem virtuální privátní sítě včetně nastínění důvodů jejich budování. Následně jsou uvedeny způsoby zabezpečení pomocí symetrického šifrování, asymetrického šifrování, hashování a digitálních podpisů, které souvisí se zaručením důvěrnosti, integrity, autentifikace, a ověřením původu.

V další části je zmíněna většina používaných i historicky překonaných modelů. Pozornost je věnována zejména technologii IPSec. Popsána je i technologie MPLS, které je předpovídána slibná budoucnost. SSL technologie je v této části záměrně vynechána z důvodu překrývání většiny vlastností s OpenVPN implementací, na níž je v práci věnován především zřetel.

V kapitole OpenVPN je vysvětlen vztah mezi pojmy SSL/TLS, krátce je nastíněn historický vývoj. Vysvětlen je důvod zapouzdření a jeho konkrétní provedení. Pozornost je věnována dvěma základním způsobům zasíťování za pomoci OpenVPN, zmíněny jsou jejich klady i zápory. Na závěr jsou shrnuty přínosy, výhody a nedostatky řešení.

V poslední části je uveden způsob instalace a konfigurace OpenVPN serveru na OS Linux. Na témže operačním systému je instalován klient a následně je proveden test spojení. Zmíněny jsou nejdůležitější parametry konfiguračních souborů a jejich využití. Zobrazen je i způsob instalace klienta ve Windows XP z důvodu jeho masového rozšíření u klientských strojů.

3 VPN

3.1 Definice pojmu VPN

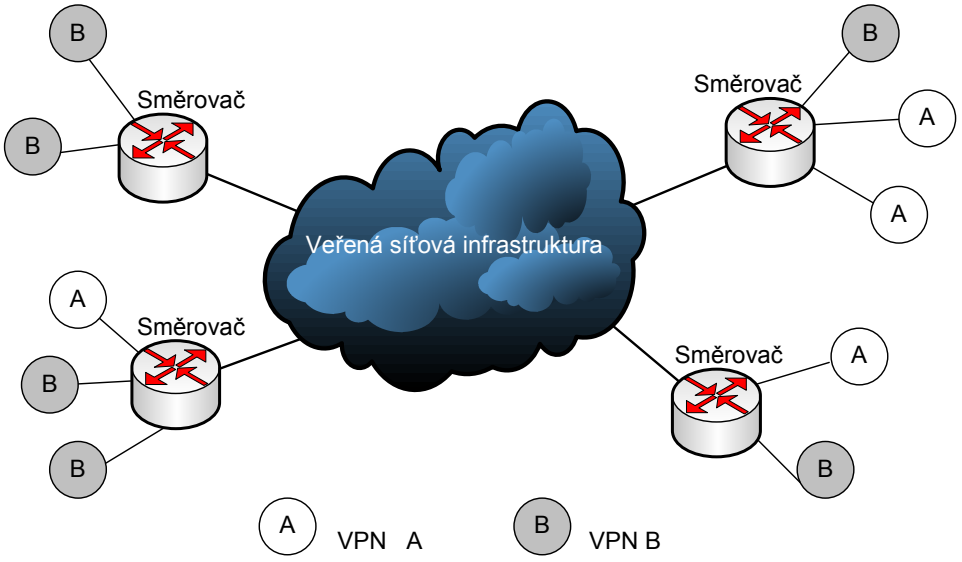
„Termín VPN nebo *Virtual Private Network* se používá k popisu širokého spektra řešení, i když sám předmět není přesně specifikován. Tato volnost v terminologii vede ke stavu, kdy je termín "VPN" používán pro označení i dosti různých technologií.“ [5]

„VPN je privátní sít, kde privátnost je vytvořena nějakou metodou virtualizace. VPN může být vytvořena v mnoha variantách - mezi dvěma koncovými systémy, mezi dvěma organizacemi, mezi několika koncovými systémy v rámci jedné organizace nebo mezi více organizacemi pomocí např. globální sítě Internet. Může být vytvořena také přímo mezi aplikacemi a samozřejmě také libovolnou kombinací všech uvedených možností.“ [5]

James Yonan (jeden z autorů OpenVPN) definuje VPN poněkud srozumitelněji:
VPN se sestává z množiny nástrojů, které umožňují bezpečné spojení dvou privátních sítí na odlišných místech s použitím veřejné síťové infrastruktury. Touto síťovou infrastrukturou je v dnešní době převážně Internet. [6]

„Ověřené a šifrované spojení na bázi VPN (samotný internet totiž nebyl navrhován s ohledem na bezpečnost) také poskytuje mnohem vyšší bezpečnost a ochranu než tradiční systémy vzdáleného přístupu chráněné heslem. VPN jsou tedy snazším, efektivnějším, levným a přitom vysoce bezpečným způsobem, jak zůstat v kontaktu se společností i mimo její sídlo.“ [9]

VPN je velmi výhodné pro vzdálené připojení cestujících uživatelů. Pomocí vytáčeného VPN spojení se mohou snadno a levně připojit prostřednictvím lokálního poskytovatele spojení a za použití VPN tunelu navázat spojení s firemní centrálou.



Obr. 1 běžný typ spojení privátních sítí

Obecný typ spojení několika privátních sítí prostřednictvím VPN demonstruje obrázek 1. Subsítě A i B nemají žádné informace o existenci té druhé.

3.2 Šifrování

Veškerá funkčnost VPN závisí silně na šifrování. S jeho pomocí je budováno a udržováno zabezpečené spojení.

Šifrování se skládá ze čtyř základních částí:

- Symetrických šifer
- Asymetrických šifer
- Hashování
- Digitálních podpisů

Které souvisí se čtyřmi dosahovanými cíly:

- Důvěrnost
- Integrita
- Autentifikace
- Nezřeknutí

3.2.1 Symetrické šifrovací nástroje-důvěrnost

Pokud je požadováno zabezpečení dat proti jejich odposlechu, musí se zašifrovat. Symetrické šifrování používá velmi rychlé šifrovací mechanismy pro zašifrování a rozšifrování dat. Obě strany tunelu používají stejný šifrovací mechanismus. Nejčastěji používanými jsou DES, 3DES, *Blowfish*, AES (*Rijndael*), RC5, RC6, *Serpent* a IDEA.

3.2.2 Hashování -Integrita

VPN používá pro přenos citlivých dat veřejné médium (Internet), což umožňuje jejich úmyslné i neúmyslné poškození či změnění. Z toho důvodu je potřeba zajistit, že data, která jsou odeslána, jsou daty, která jsou následně přijata. K tomuto zabezpečení je používáno hashování. Jedná se o jednosměrnou matematickou funkci šifrující zprávu jakékoliv velikosti do stejně dlouhých fragmentů. Z toho vyhází název hashování. Jedná se o jakýsi šifrovací sumář zprávy. Žádná zpráva by neměla mít stejný hash. Pokud je zpráva jakkoli změněna, výstup z funkce bude odlišný.

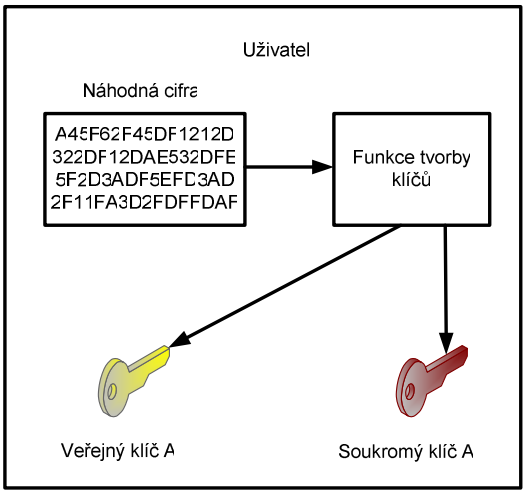
Než je odeslána zpráva je zpracována hashovací funkcí, pomocí níž vznikne hash, který je poslán se zprávou. Příjemce poté rozběhne stejnou funkci a porovná výsledek. Pokud jsou hashe totožné je jasné, že zpráva nebyla změněna.

3.2.3 Asymetrické šifrovací nástroje-kryptografie veřejného klíče

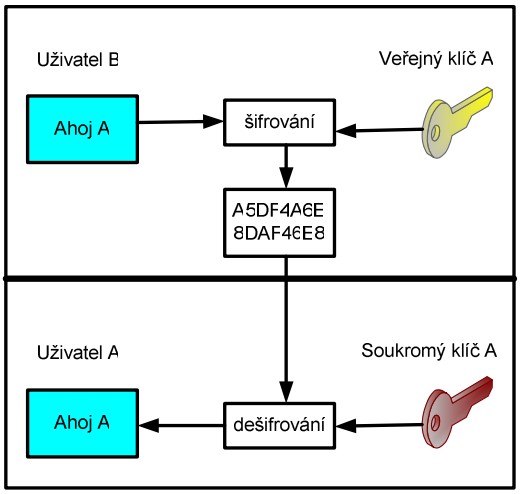
Šifrování prostřednictvím veřejného klíče bylo představeno v roce 1970. Jedná se o moderní metodu, která zaručuje bezpečnou komunikaci bez nutnosti použití předem smluveného hesla. Šifrování se sestává ze dvou částí, ze soukromého a veřejného klíče. Soukromý klíč je uchováván v tajnosti a veřejný klíč je distribuován tomu, kdo o něj požádá. Jedná se o systém asymetrického šifrování a všichni uživatelé používají stejné druhy šifrování pro veřejný klíč. Nejpoužívanějšími jsou RSA, *Diffie-Helman* a *ElGarnal*.

Použití asymetrického šifrování lze popsat následovně:
Pokud chce uživatel B zajistit bezpečný přenos dat k uživateli A, tak požádá A o jeho veřejný klíč a zašifruje pomocí něj svoje data. V okamžiku přijetí vezme uživatel A svůj

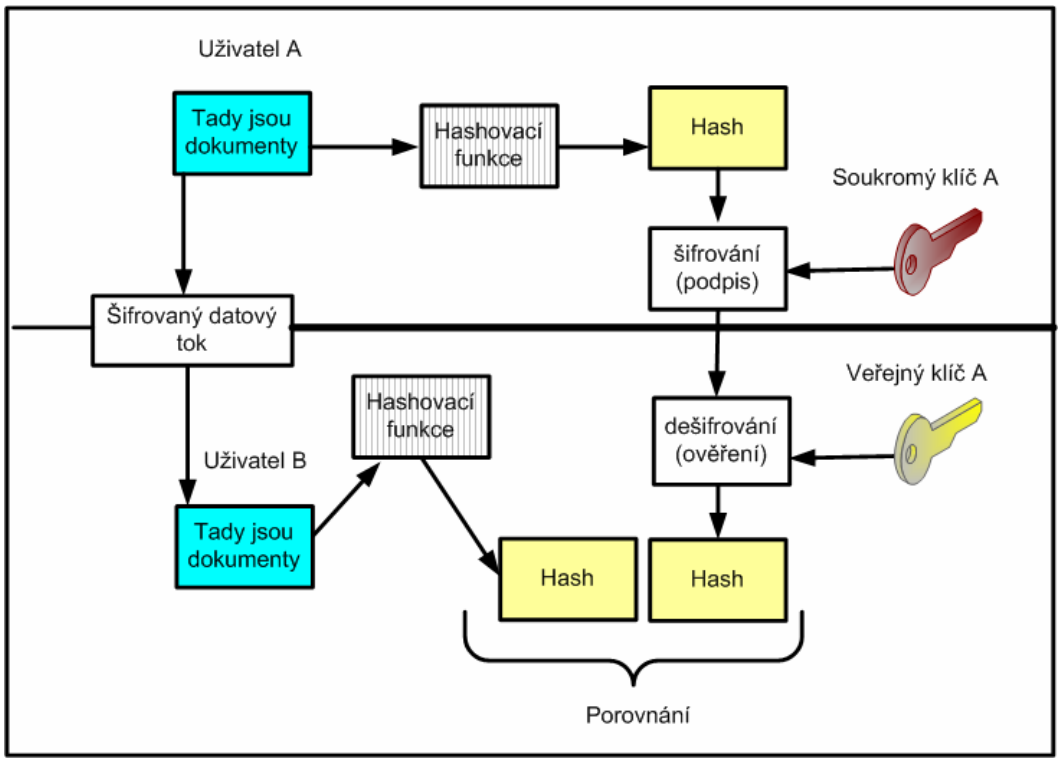
soukromý klíč a rozšifruje data. Z předcházejícího textu je zřejmé, že se jedná o dvojici klíčů tj. pouze uživatel, kterému náleží veřejný klíč párovaný s jeho soukromým klíčem, může rozšifrovat zprávu.



Obr. 2 tvorba klíčů



Obr. 3 kryptografie veřejného klíče



Obr. 4 digitální podpisy+integrita

3.2.4 Asymetrické šifrovací nástroje- digitální podpisy

V dnešní praxi se využívají digitální podpisy. Pokud odesílatel potřebuje poslat zprávu, rozběhne hashovací funkci, poté digitálně potvrdí hash prostřednictvím zakódování jeho privátním klíčem. Celý tento balík je následně zakódován symetrickým šifrováním k zaručení důvěrnosti. Příjemce poté rozšifruje hash pomocí veřejného klíče odesílatele a v případě, že funguje, je zaručeno nezřeknutí, tj. odesílatel nemůže popřít, že zprávu odeslal právě on. Následně je prověřena integrita viz 3.2.2.

3.2.5 Certifikáty

Jsou prostředkem k ověření identity prostřednictvím CA (*Certificate Authority*) použitím kryptografie veřejného klíče. Pokud je požadováno ověření identity je nutno učinit následující kroky:

- Generování požadavku k udělení Certifikátu
- Tvorba dvojice soukromého/veřejného klíče
- Poslání požadavku s veřejným klíčem k CA
- Odpověď, která je přidělená k veřejnému klíči bude odeslána žadateli (skládá se ze tří částí)
    - Certifikát CA obsahující veřejný klíč
    - Lokální certifikát identifikující VPN zařízení
    - Někdy též CLR (*certificate revocation list*) obsahující seznam odvolaných certifikátů prostřednictvím CA

Následně je načtena odpověď do VPN zařízení prostřednictvím TFTP (Thin File Transfer Protocol) přes příkazovou řádku. Načtení certifikátu do VPN zařízení přináší následující:

- Identita žadatele může být ověřena použitím lokálního certifikátu
- Certifikát CA může být použit k ověření identity jiných uživatelů
- CRL list může být použit k ověření neplatných certifikátů

CLR

Je používán k ujištění se, že se certifikát nestal neplatným. VPN zařízení jej používají k ověření platnosti před tvorbou VPN tunelu. Certifikát je ověřován v první fázi navazování zabezpečeného spojení. V případě nenačtení CRL do VPN se zařízení snaží o znovunačtení prostřednictvím LDAP (*Lightweight Directory Access Protocol*) nebo HTTP (*Hypertext Transfer Protocol*) podle definice uvnitř CA certifikátu. [1]

3.3 Modely koncových bodů VPN

Existuje několik možností tvorby koncových bodů VPN. Jejich konstrukce je ovlivněna především výběrem zařízení, jež má tímto koncovým bodem býti. Volba tohoto zařízení je určována množstvím průchozí komunikace.

Tři základní modely koncových bodů jsou následující:

- Koncový bod VPN na směrovači
- Koncový bod VPN na firewallu
- Koncový bod VPN na určeném zařízení (Server, Notebook, *Gateway* atd.)

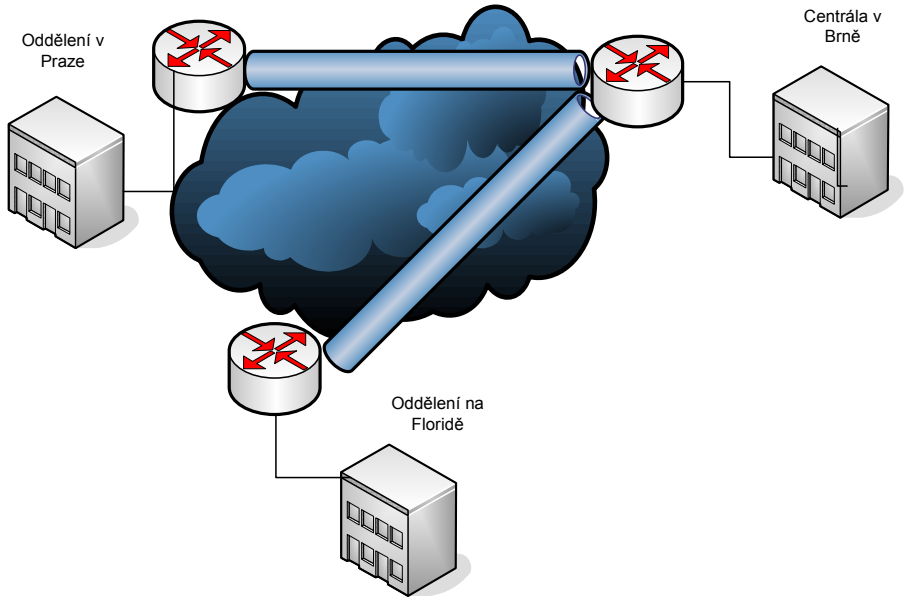
3.3.1 Koncový bod VPN na směrovači

Ukončení VPN komunikace na směrovači zajišťuje prověření veškeré příchozí komunikace firewallem. Tato topologie je nejlépe aplikovatelná na extranetové spoje, kde uživatelé nepožadují přístup k interní síti, ale potřebují zajistit přístup k serverům v DMZ (síťovém rozhraní), které by neměly být vystaveny běžné internetové komunikaci. Při větším počtu VPN spojů je směrovač šifrováním a dešifrováním zahlcen a šifrování se musí být přeneseno jinam.

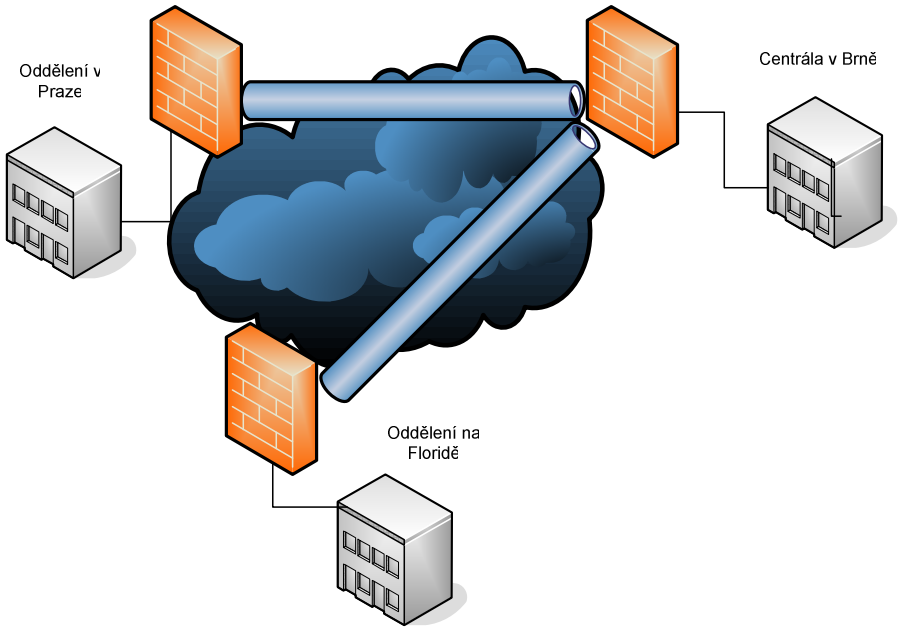
3.3.2 Koncový bod na firewallu

Ukončení VPN komunikace na firewallu umožňuje přímý přístup ze vzdálených sítí do jádra interní firemní sítě. Vzdálení uživatelé mají přístup ke všem vnitřním službám bez nutnosti druhotného ověření. Tento způsob ukončení je vhodný pro *LAN-to-LAN* spoje. Například propojení jednotlivých poboček s centrálním ředitelstvím. Stejným způsobem ale může být použit pro rozsáhlé WAN sítě, pokud je před firewallem směrovač pro

přímé spojení přes Internet. Nevýhoda tohoto propojení je možnost přetížení v případě velkého počtu oddělení připojených k centrále.



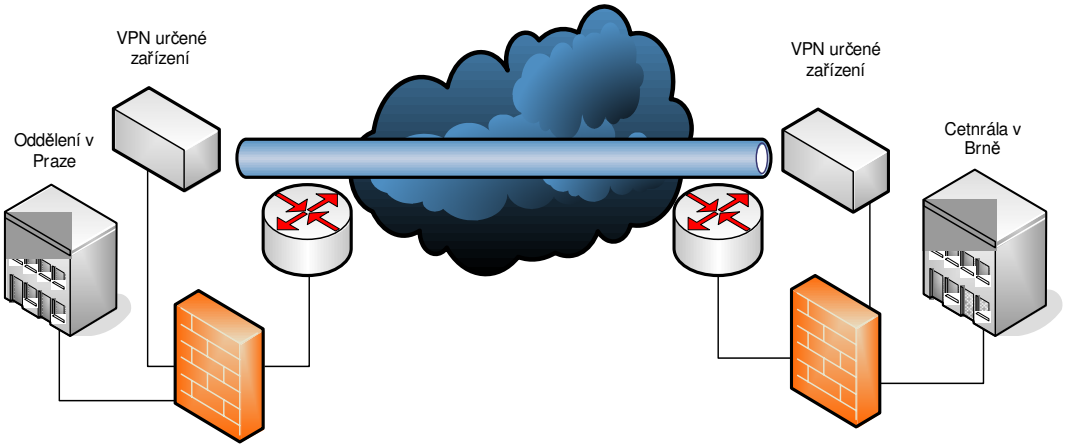
Obr. 5 koncový bod na směrovači



Obr. 6 koncový bod na firewallu

3.3.3 Koncový bod VPN na určeném zařízení (server)

Ukončení VPN komunikace na určeném zařízení je nejčastěji používáno pro *LAN-to-LAN* konektivitu. Jednou z výhod tohoto ukončení je možnost využití koncového zařízení (server) v kombinaci s bezdrátovými sítěmi. Omezení této struktury vychází předcházejících.



Obr. 7 koncový bod na určeném zařízení (server)

3.4 Členění VPN protokolů

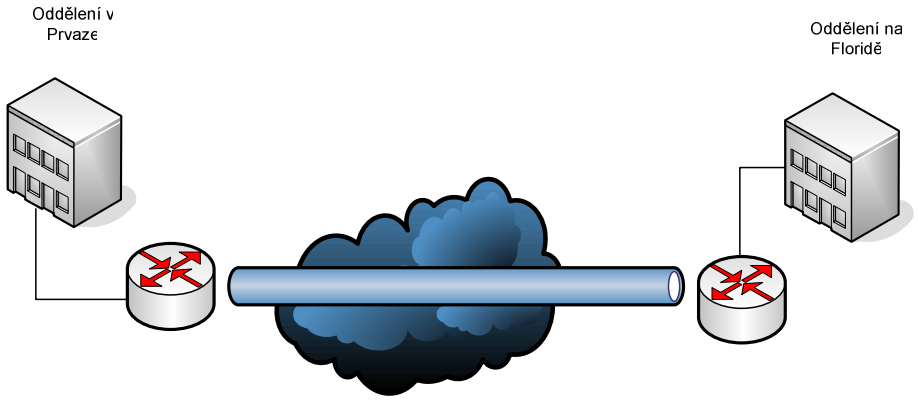
Při přenášení dat přes nezabezpečená média (Internet) je nutno zajistit datovou integritu a relevanci při zachování transparentnosti pro koncové uživatele. Z tohoto důvodu vzniknul Koncept VPN. Následně bylo pověřeno IETF (*Internet Engineering Task Force*) tvorbou standardizovaných protokolů pro zabezpečení dat a zajištění jejich spolehlivosti. Kromě IETF bylo vyvinuto několik protokolů i dalšími subjekty např. společností Microsoft.

Protokoly můžeme členit do dvou hlavních skupin:

- *Site-to-site* protokoly
  - IPSec
  - GRE (*Generic Routing Protocols*)
  - MPLS
- Protokoly se vzdáleným přístupem
  - IPSec
  - SSL
  - L2TP
  - L2TP přes IPSec
  - PPTP

Jiné rozdělení je možné podle RM-OSI:

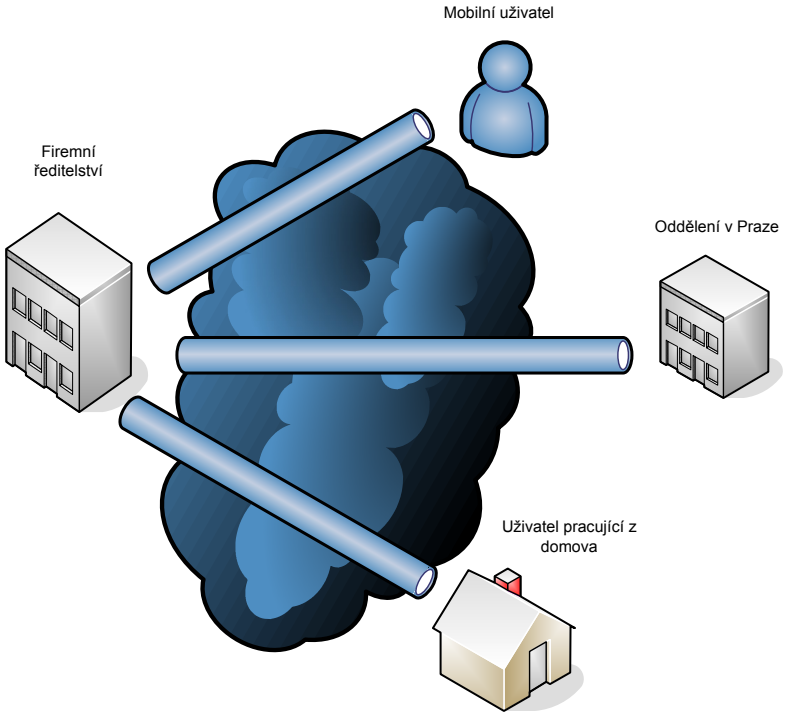
- Aplikační vrstva
  - WEP, WPA
  - L2TP, PPTP
  - SSL, SSH
- Spojová vrstva
  - MPLS
  - *Frame Relay*, ATM
  - 802.1q VLAN *Tagging*
- Síťová vrstva
  - IPSec
  - MPLS
- Fyzická vrstva
  - SDH/SONET



Obr. 8 *site-to-site* technologie

3.4.1 Site-to-site protokoly vs. Protokoly se vzdáleným přístupem

Site-to-site protokoly umožňují zajištění bezpečného spojení mezi dvěma a více odděleními, za použití veřejného média. Toto někdy eliminuje potřebu pronájmu dedikovaných linek.



Obr. 9 technologie se vzdáleným přístupem

Hlavní výhodou protokolů se vzdáleným přístupem je možnost připojení uživatelů z rozličných vzdálených lokací (domácnosti, hotely či internetové kavárny) a následného jednoduchého navázání se na centrální síť společnosti. Díky tomu nemusí organizace disponovat množstvím přístupových serverů pro uspokojení potřeb vzdálených uživatelů.

3.4.2 IPSec

Systém IPSec, definovaný podrobně v RFC 2401 až 2412, byl prvním VPN protokolem pro zabezpečení internetové komunikace. Je povinný pouze pro implementace v IPv6, ale používán je i v mnoha aplikacích IPv4. Srdcem IPSec je schopnost vzít privátní paket IP a buď jej zcela zapouzdřit do jiného IP paketu, nebo přidat do IP hlavičky dostatek informací pro zajištění jeho integrity.

Předtím, než mohou být pakety IPSec vyměňovány, musí se oba účastníci dohodnout na atributech zabezpečeného spojení, tzv. *Security association* (SA), která obsahuje sadu parametrů zahrnujících. [2]

- Kryptografické algoritmy
- Dobu platnosti klíče
- Druh zapouzdření
- Kompresi

Z bezpečnostních a praktických důvodů je k sestavování těchto parametrů použit protokol IKE (*Internet Key Exchange*). V případě oboustranné podpory IKE je možné dojednat SA za běhu, před vytvořením zabezpečené komunikace. Předpokladem pro jeho dojednání je vzájemná autentifikace. Ověření je prováděno pomocí tajných sdílených informací, nebo častěji pomocí mechanismu veřejného klíče, který byl popsán v předcházejících kapitolách.

Způsob zaručení zabezpečené komunikace je realizován následovně:

- Důvěrnost dat- pomocí symetrického šifrování
- Datová integrita – každý člen komunikace může ověřit, jestli byl přijatý paket změněn
- Datová autentifikace – příjemce může ověřit odesílatele
- *Anti replay* – příjemce může detekovat a odmítnout opakované pakety (ochrana proti *spoofing* a *man-in-the-middle* útokům)

IPSec zajišťuje ochranu prostřednictvím dvou hlavních protokolů. ESP (zapouzdření) a AH (autentifikace hlavičky). Všechny ostatní části standardu pouze implementují tyto protokoly a konfigurují požadované technické parametry. Použití AH nebo ESP na IP paket vkládá AH či ESP hlavičku mezi IP hlavičku obsah paketu. [1] Transformace je zobrazena na obrázcích níže.

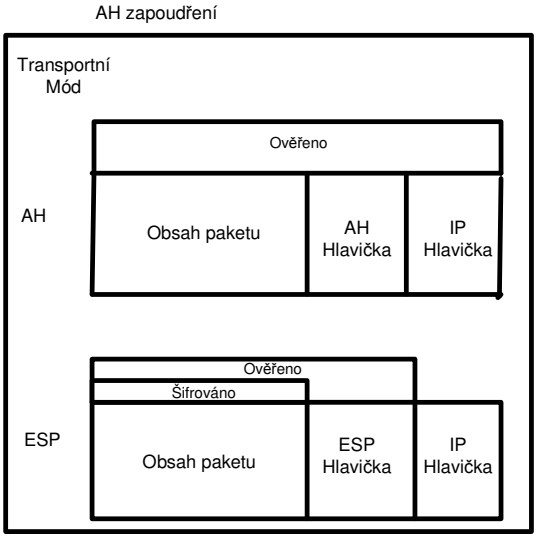
AH hlavička zaručuje:

- Datovou integritu
- Autentifikaci původu
- *Replay* ochranu

ESP hlavička zajišťuje:

- Ochranu proti odposlechům a šifrování
- Volitelně autentifikaci (stejným algoritmem jako AH)
- Volitelně *replay* ochranu

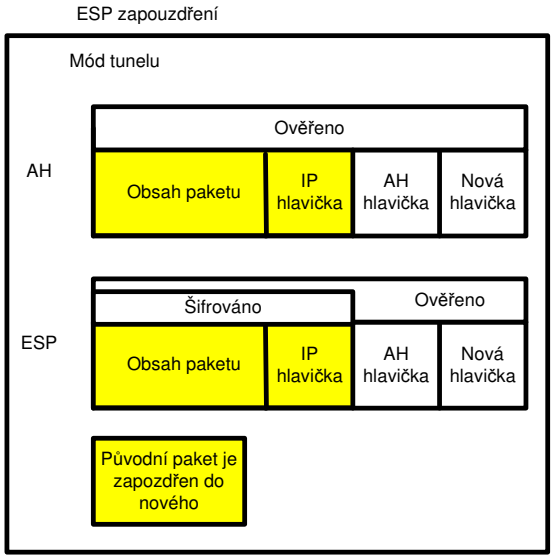
Původní ESP nezajišťoval autentifikaci a *anti-replay* ochranu, spoléhalo se na kombinaci použití ESP a AH pro jejich zabezpečení. Ale od chvíle kdy ESP začalo poskytovat stejné služby, jako AH se přestal druhý jmenovaný protokol používat.



Obr. 10 transportní režim

Transportní režim ESP:

Používá se pro ochranu obsahu IP paketů přenášených mezi dvěma bezpečnostními branami s IPSec zabezpečením. Při jeho použití vkládá bezpečnostní brána hlavičku ESP mezi data a IP hlavičku odkazující na předem dojednané SA a další ESP paket.



Obr. 11 tunelový režim

Tunelový režim ESP:

Zaručuje zabezpečení obsahu IP paketu včetně hlavičky. Jedná se o systém poskytující ochranu integrity, utajení a ověřování dat. Další nespornou výhodou tunelového režimu je možnost nepoužívání legitimních IP adres, z důvodu jejich skrytí při přenosu.

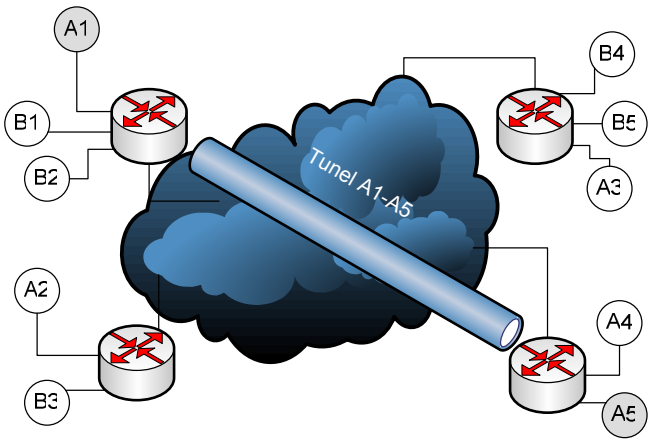
3.4.3 GRE (Generic Routing tunnels)

Efektivní cestou k vytváření virtuálních privátních sítí je tunelování. Při jeho realizaci je specifická část síťové komunikace přenášena speciálně vytvořeným tunelem. Nejčastější technologií tunelování pro spojení mezi dvěma směrovači je GRE (*Generic Routing Encapsulation*). Jedná se o klasický *overlay* model VPN.

„*Peer* model VPN je takový, ve kterém jsou směrovací výpočty prováděny vždy v každém uzlu na dráze paketu k cíli ("*hop-by-hop*") tj., jednotlivé uzly jsou rovnocenné ("*peer*"). Příkladem tohoto *peer* modelu jsou tradiční sítě, založené na směrovačích. Alternativní "*overlay*" model je takový, kde směrování na síťové vrstvě není založeno na směrování v každém mezilehlém uzlu na dráze paketu, ale na využití techniky vytváření přímých spojení ("*cut-through*") na úrovni spojové vrstvy mezi dvěma body sítě. Příkladem tohoto modelu jsou sítě založené na technologii ATM, *Frame Relay* nebo technice tunelování. ” [5]

Kromě výše zmíněného rozdílu je mezi oběma modely značný rozdíl ve škálovatelnosti. U *overlay* technologie může u rozlehlých sítí dojít k zahlcení směrovačů z důvodu vysoké výpočetní náročnosti.

Koncovými body GRE tunelů jsou standardně směrovače páteřní sítě. Přenášené pakety jsou vybaveny přídavnou GRE hlavičkou a adresou směrovače na konci tunelu. V koncovém bodu je zapouzdření odstraněno a paket následně putuje ke svému cíli s původní IP hlavičkou.



Obecná konstrukce GRE tunelů předpokládá implementaci bod-bod. Některé konstrukce ale umožňují také bod-více bodů tj., existenci několika konců.

Obr. 12 GRE tunel

Výhodou konceptu tunelování je vzájemné odstínění adresace v rámci společné sdílené sítě a privátního adresového prostoru. Všechny přístupové body jsou taktéž koncovými body přenosových tunelů a používají adresaci veřejné sítě. Zároveň zapouzdřené pakety

přenášené tímto tunelem používají adresy z adresového prostoru VPN. Díky tomu je možno použít privátní adresový prostor k adresaci na všech vzájemně oddělených privátních sítích.

Privátní síť může pracovat na libovolném protokolu, který je přenášen přes veřejnou síťovou infrastrukturu bez jakékoli změny a tím je simulována privátní síťová komunikace. Největším problémem GRE je výše zmíněná škálovatelnost řešení. Všechny tunely musí být totiž manuálně zkonfigurovány, což při větším počtu tunelů způsobuje vysokou administrativní zátěž.

3.4.4 Sítě s komutovaným přístupem

Odlišné virtuální privátní sítě, využívající tunelování představují sítě s komutovaným přístupem VPDN (*Virtual Private Dial Networks*). Nejpoužívanějšími implementacemi jsou L2TP a PPTP, přičemž L2TP vychází ze standardu L2F.

Rozlišují se dva odlišné způsoby vzniku spojení na základě požadavku:

- Klienta (dobrovolná inicializace)
- Serveru (povinná inicializace)

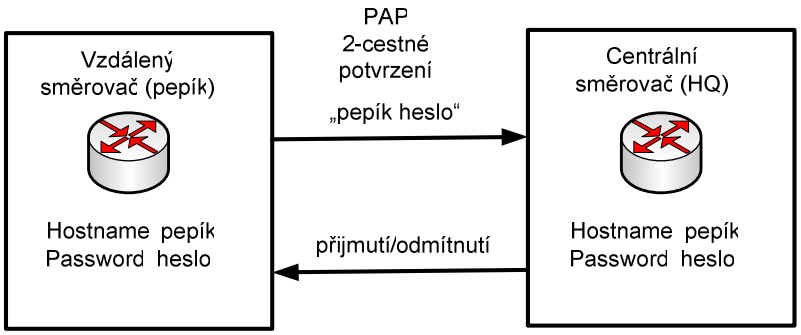
PPTP:
Byl vyvinut společností Microsoft jako alternativa IPSec VPN. Využívá protokolu PPP (*Point-to-Point Protocol*), kvůli tomu závisí veškeré implementace na instalaci PPP. Největším problémem PPTP jsou určité mezery v přidání zabezpečení připojení.

„První z nich jsou metody ověřování typicky podporované zařízeními PPP, jak od společnosti Microsoft, tak od třetích stran, jako např. PAP (*Password Authentication Protocol*), který vlastně není vůbec zabezpečený, protože posílá hesla v otevřené formě a CHAP (*Challenge Handshake Authentication Protocol*) včetně variant od Microsoftu a Shivy.“ [3]

„Druhou je důvěrnost, pro ochranu důvěrnosti užitečného obsahu paketu PPTP při průchodu nedůvěryhodnou sítí definoval volitelný šifrovací protokol. Protokol MPPE (*Microsoft Point-to-Point Encryption)* používá šifrovací algoritmus RC4 s heslem jako základem pro vygenerování klíčů relace pro šifrování.“ [3]

PAP:

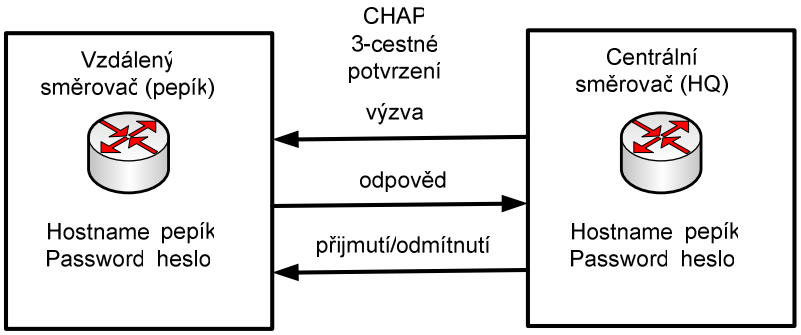
Při ověřování pomocí PAP se používá dvoucestné potvrzení, v němž je pár „username/password“ opakovaně vysílán protějšímu uzlu dokud není ověření potvrzeno nebo spojení ukončeno. Hesla jsou zasílána v nezašifrované podobě, Kvůli tomu musí vzdálený uzel hlídat případné opakované pokusy o spojení (útok typu pokus-omyl).



Obr. 13 PAP

CHAP:

Při ověřování pomocí CHAP se používá třícestné potvrzení, které se provádí během 2. fáze relace a oproti PAP může být opakováno i během spojení. Šifrování hesla se provádí metodou jednocestné hashovací funkce (MD5 – *message digest* 5). CHAP také nedovoluje identifikaci volajícího uzlu bez předchozí výzvy.



Obr. 14 CHAP

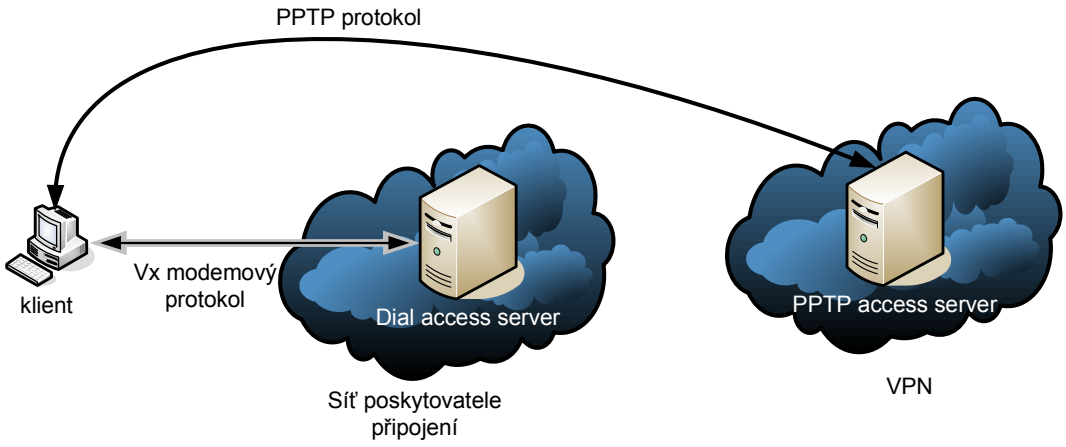
PPP relace se skládá ze tří částí:

- Sestavení spojení
- Ověření koncového uzlu (nepovinně)
- Zapouzdření do NCP (DHCP), NCP (IP)
- Zrušení spojení

V okamžiku sestavení IP připojení, nejčastěji přes PPP k nejbližšímu serveru pro vzdálený přístup (RAS) svého ISP (*Internet Service Provider*), použije klient zprávy PPTP běžící na vrcholu TCP pro sestavení připojení k serveru PPTP a dohodnutí adresy tunelu. PPP proces na serveru PPTP přiřazuje lokální i vzdálenou adresu výsledného tunelu PPP. Po úspěšném ověření uživatele je sestavena relace, na jejímž základě si mohou oba účastníci začít vyměňovat IP pakety. [4]

Existuje dokonce implementace podporující instalaci PPTP *open-source* serveru na linuxovém stroji, která samozřejmě umožňuje pracovat s klienty obsaženými ve Windows. Nazývá se PopTop a skládá se ze tří částí:

- Z jednoho daemonu
- Konfiguračního souboru
- Sady manuálových stránek a podpůrné dokumentace

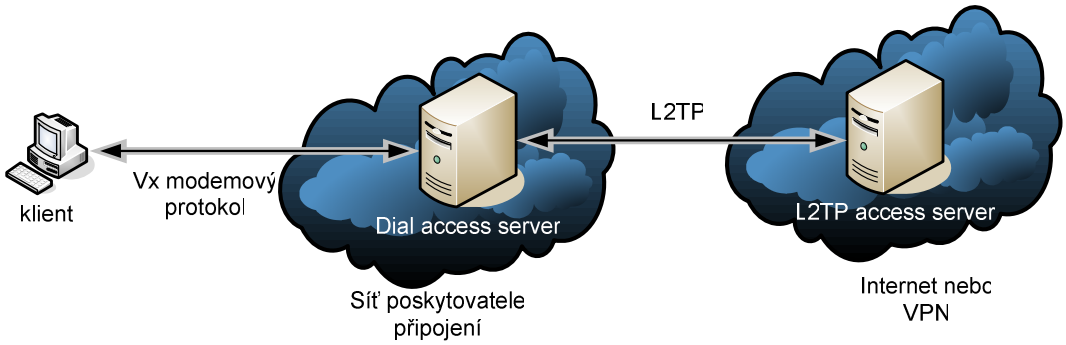


Obr. 15 PPTP

Inicializace tunelu s libovolně umístěným PPTP serverem probíhá bez účasti přístupového serveru. Spojení s žádaným PPTP serverem tvoří klient následně po ukončení spojení s přístupovým serverem.

L2TP:

Využívá se v kombinaci se zabezpečením pomocí IPSec protokolu. Takovéto připojení je poté označováno jako připojení L2TP/IPSec. Zaručuje základní služby typu zabezpečení a šifrování dat. Inicializace spojení probíhá následujícím způsobem. K přístupovému serveru se připojí uživatel a na základě jeho konfiguračního profilu proběhne autentifikace. V případě pozitivního ověření je dynamicky vytvořen L2TP tunel.



Obr. 16 L2TP

Srovnání obou modelů:

Výhodou PPTP je možnost výběru koncového bodu až po sestavení PPP spojení, což je velmi vhodné v případě často se měnícího cílového uzlu. Zároveň je zaručena transparentnost připojení, které tak může tak přesahovat více sítí.

Při použití L2TP modelu je po ukončení konektivity v síti poskytovatele spojení předáno dále do oblasti poskytovatele obsahu, kde leží koncový bod. Používáno je z důvodu hierarchické struktury veřejné telefonní sítě a v případech kdy velcí poskytovatelé obsahu přenechávají přístupové sítě.

3.4.5 Sítě na spojové vrstvě

Použití vlastního přenosového systému a síťové infrastruktury pro vytvoření virtuální privátní sítě patří mezi nejpřímější metody budování VPN, a umožňuje přitom budovat na tomto základě nezávislé VPN na vyšší přenosové vrstvě - diskrétní virtuální sítě na síťové vrstvě. VPN na síťové vrstvě můžeme považovat za blízkou (či přesnou) funkční analogii konvenčních privátních datových sítí. [5]

Virtuální spojení v sítích ATM a Frame Relay:

V tradičních privátních datových sítích je používána kombinace privátní komunikační infrastruktury a dedikovaných linek, na jejímž základě je vytvořena celistvá nezávislá komunikační infrastruktura. Virtuální privátní síť může tuto strukturu kopírovat nebo i přesahovat tzn., rozprostírá se i po pronajatých obvodech, jejichž základním rysem je synchronizace vysílání a příjmu dat.

Odlišný přístup je používán při konstrukci VPN na spojové vrstvě. Zde je předpokládána vysoká míra soběstačnosti a výdělečnosti, při využití veřejné síťové infrastruktury. Zároveň využívá společnou spojovou infrastrukturu, včetně přepínacích prvků při vzájemném odstínění a to jak přímé, tak nepřímé. Prvotní rozdíl mezi virtuálními linkami a dedikovanými obvody je v absenci časové synchronizace přenosů, přičemž dedikovaná přenosová cesta ani nemusí existovat. Počáteční bod také nezná předem velikost dostupné přenosové kapacity virtuálního obvodu z důvodu jeho závislosti na celkových požadavcích ostatních přenosů. Z toho důvodu dochází někdy k přetížení sítě.

„V sítích *Frame Relay* se používá pojem CIR (*Committed Information Rate*), sloužící jako referenční hodnota pro kontrolu velikosti přenosové rychlosti ve vstupním bodu sítě. Překročí-li rychlost dohodnutou hodnotu CIR, vstupní rámce jsou sice dále sítí akceptovány, jsou ale označeny jako DE (*Discard Eligible*). Takto označené rámce pak mohou být jako první zahozeny, dojde-li na jejich cestě sítí k přetížení (je překročena max. vstupní rychlost na přepínači a dojte k přetečení vyrovnávacích pamětí).“ [5]

Již zmíněnou výhodou sítí poskytovatelů přenosových služeb je velká flexibilita těchto sítí. Při pronajmutí virtuálních sítí jsou ve většině případů ve smlouvě uvedeny i jednotlivé parametry očekávaného vytížení poskytovaných služeb.

U sítí ATM (*Asynchronous Transfer Mode*) i u *Frame Relay* spojů není používána synchronizace datových přenosů přijímače a vysílače. Podobná vstupní funkce obstarává také kontrolu rychlosti vstupního proudu buněk, jenž mohou být označeny při překročení dohodnuté rychlosti indikované pomocí CLP (*Cell Loss Priority*), které jsou jako první zahozeny. Virtuální obvody na spojové vrstvě poskytují bezesporu kvalitní alternativu k sítím s dedikovanými obvody.

3.4.6 MPLS (MultiProtocol Label Switching)

Jedná se o relativně novou alternativu pro přenos IP paketů sítí WAN, která přepíná pakety definované po cestě na základě návěstí v doplněném záhlaví MPLS protokolu. Její velkou výhodou je aplikovatelnost pro různé protokoly a technologie (*Ethernet*, *Frame Relay*, ATM, IP, IPX …).

Jako většina standardizovaných protokolů i MPLS byl vypracován IETF. Její pomocí se podařilo přiblížit přenosové rychlosti IP ke spojově orientovaným technologiím viz, výše.
V budoucnosti je předpokládán přechod k technologiím G-MPLS.

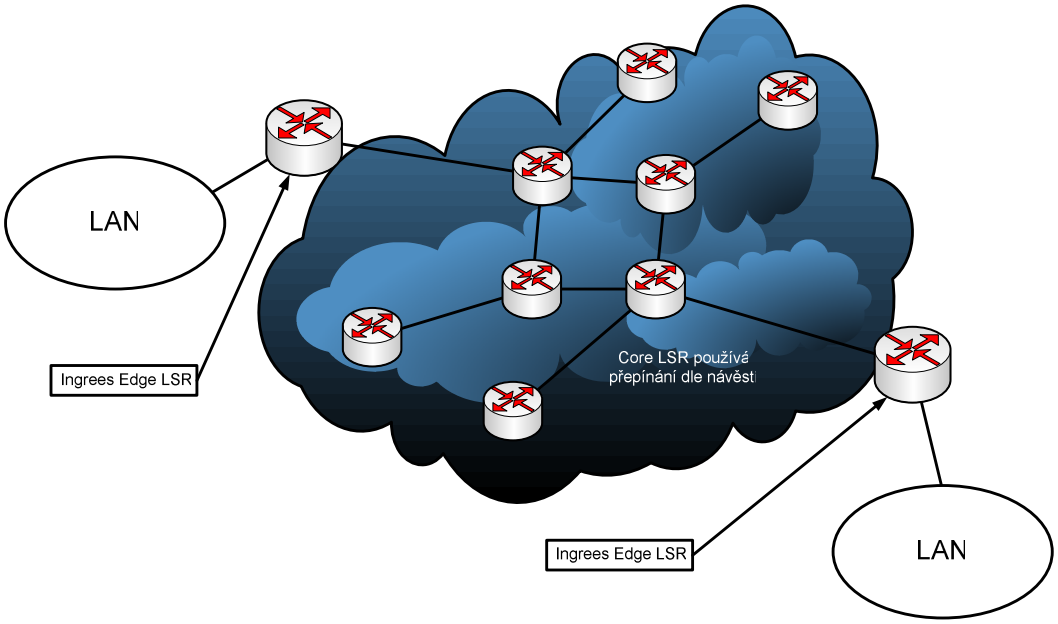
MPLS lze užít v oblastech:

- Přepínání a směrování
- Vytváření VPN
- Řízení provozu
- Zajištění QoS (*Quality of service*)

MPLS poskytuje řadu funkcí. Jednou z nich je analýza paketů směřujících do sítě a jejich klasifikace do FEC (*Forward Equivalence Class*) skupin. K nim vytváří návěstí a zároveň je distribuuje v MPLS síti. Nastavuje přepínací tabulku ve směrovačích a

vytváří přepínací cesty. Také zajišťuje přepínání paketů podle návěstí a vkládá MPLS záhlaví. [4]

Významnou výhodou je směrování pouze v hraničních routerech sítě a přepínání podle návěstí (*label Switching*), což zvyšuje přenosovou kapacitu. Také poskytuje kvalitu služby, zakrývá spojovou vrstvu a rozdíly mezi různými protokoly spojové vrstvy.



Obr. 17 MPLS

Edge LSR:

Hraniční směrovač sítě MPLS pracující ve funkci vstupního (*Ingress*) nebo výstupního (*Egress*) směrovače. Vstupní analyzuje informace v záhlaví IP paketu (adresa cílového uzlu, třída služby CoS, síť VPN), na jejichž základě přiděluje paket k definované skupině FEC. Pro zvolenou FEC je do záhlaví MPLS vloženo přiřazené návěstí. Výstupní odebírá záhlaví MPLS z přijatého paketu a odesílá původní IP paket na výstupní port. V průběhu přenosu se nemění záhlaví, pouze obsah pole TTL (*Time To Live*), které je dekrementováno v hraničním směrovači sítě. [4]

Core LSR:

Páteřní přepínač, který zajišťuje přepínání paketů podle návěstí v záhlaví MPLS mezi síťovými uzly, tzv. přepínání na vrstvě 2+. Záhlaví paketu nepodléhají během přepínání páteřními přepínači změně ani zjišťování (mění se pouze návěstí). [4]

Používané protokoly v MPLS:

- Aplikační
    - Signalizační MPLS, LDP, TDP, CR-LDP
- Transportní
    - Pro přenos signalizačních zpráv UDP/TCP
- Síťové
    - Směrovatelné (IP,IPX)
    - Směrovací (BGP, OSPF, IS-IS)
- Spojová (2)
    - ATM, Ethernet, PPP, *Frame Relay*
- Spojová (2+)
    - MPLS- přepínání podle návěstí
- Fyzická
    - ATM, Ethernet, *Packet over SDH*

4 OpenVPN

V první kapitole bylo záměrně vynecháno SSL VPN z důvodu překrývání většiny vlastností s OpenVPN implementací, která bude uvedena v této části.

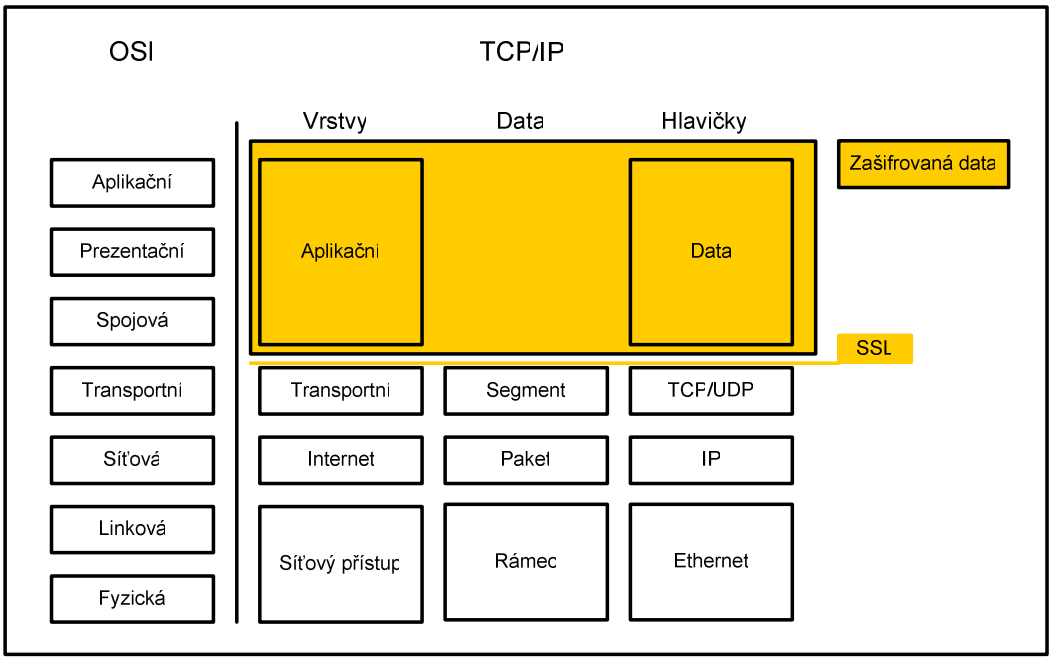
Po mnoho let byl IPSec jediný dostupný protokol k zabezpečení sítí VPN mezi pobočkami či mezi klientem a serverem. Nástup protokolu SSL toto naštěstí změnil. SSL již od počátku umožňuje zabezpečit konkrétní protokoly (HTTP), komunikace mezi aplikacemi a šifrovat data posílaná v TCP nebo UDP tunelech, používaných k vytváření sítí VPN. [10]

Specifikace SSL (*Secure Sockets Layer*) byla vytvořena v 90. letech společností Netscape. Nejprve byly publikovány dvě verze SSL: verze 2 (1994) a verze 3 (1995). Poté byl patent v roce 2001 koupen organizací IETF, následně upraven a přejmenován na TLS (*Transport Layer Security*) (RFC 2246). Zkratka SSL je běžně užívána k označení protokolů SSL i TLS. Poslední verze TLS je 1.1. [10]

V dnešní době je na trhu poměrně velké množství SSL VPN. Jedno z nejlepších a nejlevnějších řešení *open source* SSL VPN je OpenVPN. IPSec VPN jsou drahé a obtížně konfigurovatelné. IPSec je těžkopádný a pro běžného uživatele poskytuje přespříliš možností nastavení. Také operuje v módu jádra, což představuje riziko kritického selhání. OpenVPN odmítá komplexnost IPSec, užitím ověřeného SSL/TLS protokolu a kryptovacích knihoven k zabezpečení vyvážené a v některých případech lepší funkčnosti a jednoduší konstrukce. OpenVPN je SSL/VPN, které zaručuje stejnou funkčnost a ochranu jako IPSec předchůdce. [7]

Někdy je SSL VPN chápáno jako protokol, jenž šifruje datový přenos pro aplikace nebo pro přesměrování portu. V takovém případě se nejedná o VPN, ale o aplikační úroveň brány, firewall nebo SSL bránu. Virtuální privátní sítě odkazují na soukromé sítě přes veřejný internet kryptováním komunikace mezi dvěma privátními konci. Toto poskytuje stejnou konektivitu a bezpečí, kterou lze očekávat v lokální privátní síti. VPN zařízení je použito k vytvoření zašifrovaného, neaplikačně orientovaného tunelu mezi dvěma

stroji, které umožňuje těmto strojům nebo sítím vyměňovat informace bez ohledu na aplikaci či protokol. Výměna nevzniká na aplikačním základu, ale je založena na zabezpečeném spojení mezi stroji, nebo sítěmi a umožňuje přenášet libovolná data. [7]



Obr. 18 SSL šifrování

Počátky SSL VPN lze spatřovat v tun/tap adaptérech implementovaných v operačních systémech Unix. Rozhraní tun je virtuální síťový adaptér, který se pro operační systém tváří jako *point-to-point* síťový hardware. Namísto posílání bitů kabelem, je tun rozhraní posílá do uživatelského prostředí. Uživatelský program může otevřít tun zařízení stejným způsobem jako obyčejný soubor a číst i zapisovat dovnitř IP pakety. Tap rozhraní je velmi podobné ale emuluje místo *point-to-point* spojení ethernetovou konektivitu.

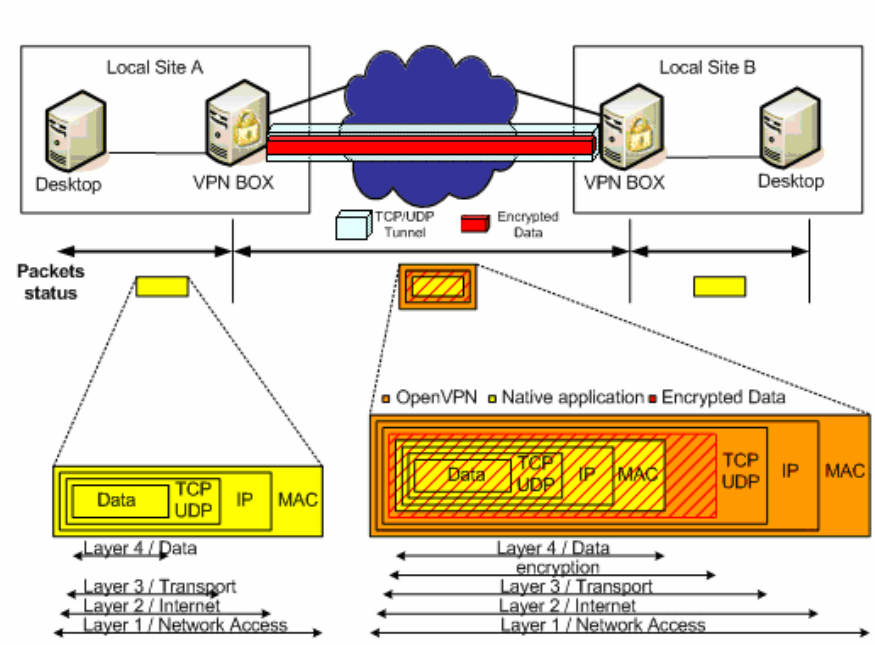
V případě instalovaného tun zařízení na stroji A i na stroji B stačí napsat jednoduchou síťovou aplikaci se dvěma vlákny, poté nakopírovat bity z tun zařízení do síťového socketu a následně zkopírovat bity ze síťového socketu do tun zařízení. V okamžiku spuštění těchto aplikací na obou strojích vzniká jednoduchá konstrukce VPN bez zabezpečení. [6]

Z A je možno navázat spojení s tun zařízením na B a z B možno navázat spojení tun zařízením A. Problémem této velmi jednoduché VPN (nazývané *clear*-text *tunnel*) je zabezpečení. [6]

V dnešní době je k dispozici několik *Open Source* VPN, které následují tun/tap model. Nejznámější jsou OpenVPN, VTun, Tinc, Cipe a další. Oproti IPSec implementaci řeší problematiku zabezpečení a přenosu dat odlišným způsobem.

4.1 Zabezpečení OpenVPN

Jednoduchou výše zkonstruovanou VPN, s jejíž pomocí byl vytvořen tunel přes TCP či UDP spojení lze zabezpečit například TCP konektivitou přes zabezpečený port s využitím SSH.



Obr. 19 OpenVPN zabezpečení [10]

TCP je spolehlivý aplikační protokol využívající nespolehlivou transportní vrstvu, což znamená, že webový prohlížeč (HTTP běží na TCP protokolu) očekává že TCP zvládne poruchy ve spojení mezi klientem a vzdáleným webovým severem. TCP zajišťuje integritu prostřednictvím opakovaného přenosu paketů, které jsou ztraceny kvůli

zahlcením nebo chybám v síti. TCP je spolehlivým mostem mezi aplikační a fyzickou síťovou vrstvou.

Jedna ze skvělých věcí v síťové komunikaci je možnost zapouzdření jednoho protokolu do jiného. V praxi se nejčastěji využívá zapouzdření IP (včetně TCP a UDP) do TCP, čímž vzniká jiný základní problém. TCP je vytvořeno pro běh přes nespolehlivé sítě. Zapouzdření TCP do TCP znamená, že je vnořována jedna spolehlivostní vrstva do další. Tj., Je produkován celý stupeň redundance. Tato redundance se projevuje v nižší efektivitě a robustnosti při vysokém vytížení sítě. Lepším řešením je zapouzdření TCP do UDP. IP bylo vytvořeno pro nespolehlivá fyzická média, které můžou trpět poruchami či zahlcením. Protože UDP je nespolehlivý protokol poskytuje IP přenosové médium, které je k uzavření přirozenější. [7]

V moderních, přenosných a snadno uživatelsky konfigurovatelných VPN jsou IP pakety z tun a tap virtuálních adaptérů zašifrovány a zapouzdřeny do UDP spojení a poté poslány vzdálenému hostiteli prostřednictvím internetu. Vzdálený hostitel rozšifruje, ověří pravost a odpouzdří IP pakety při přesunu do tun nebo tap virtuálního adaptéru na druhém konci.

VPN ochraňuje proti pasivním i aktivním útokům. Pasivní útočník odposlouchává, ale nemá žádnou schopnost přerušit či modifikovat datový přenos mezi dvěma koncovými body. Efektivním způsobem ochrany je šifrování. Aktivní útočník disponuje schopností vstoupit do přenosového procesu a modifikovat, přidávat nebo mazat datové pakety mezi dvěma stranami tunelu. Výše popsaný typ aktivních útoků se nazývá *Man-in-the-middle* útok.

Zabezpečení VPN není pouze o šifrování. Větší a obtížnější problém je ověření oprávnění (autentifikace). Autentifikace v kontextu VPN zahrnuje potvrzení každého paketu zabezpečovacím hashem. Tj., příjemce může ověřit, že je původní a z legitimního zdroje. Jak OpenVPN, tak IPSec používá HMAC konstrukce k autentifikaci paketů. HMAC ale nezajišťuje ochranu proti *replay* a *known plaintext* útokům. Ani jeden z následujících případů není v dnešní době možný. Jejich smyslem je ilustrovat,

že pouhé šifrování není dostačující, ale musí být kombinováno s HMAC, *replay* a *known plaintext* ochranou.

Replay útoky:
Předpokládáme, že útočník je schopný tapnutí se do T1 linky bankovního ústavu při nízkém provozu (ve 3 hodiny ráno). V momentě kdy sleduje šifrované bity protékající skrze linku se zařízením zvaným „*snort*“, přihlásí na webové stránky banky a udělá množství malých transferů, při nichž sleduje zašifrované pakety protékající skrze T1 linku banky. Pomocí časové analýzy je schopný získat přístup ke vzorovým zašifrovaným paketům, které reprezentují jeho peněžní transfery. V případě, že pošle na T1 linku velké množství takovýchto stejných paketů, nepotřebuje znát šifrovací algoritmus, pouze reprodukuje pakety. Jestliže banka užívá jen šifrování bez *replay* ochrany, může najít nevysvětlitelnou záplavu odpovídajících transferů následující ráno. Řešením tohoto problému je vložení jedinečného ID nebo časové známky do každého paketu před potvrzením. Receiver uchovává stopu této známky a ujišťuje se, že nikdy neakceptuje paket se stejnou časovou známkou dvakrát. OpenVPN i IPSec má implementovanou ochranu proti *replay* útokům použitím *Sliding Window* Algoritmu. [6]

Known Plaintext útoky
Jestliže *hacker* udělá pět transferů s odlišnými částkami peněz. Může analýzou textu šifry vyhledat místo, kde se nachází jeho transfery. Je schopný rozeznat byte týkající se peněžní částky transferu i v případě, že jsou jejich velikosti nesrozumitelně zašifrovány. Předpokládá, že jeden $ je 32 bitový *integer*. Vloží nějaký falešný paket na člen s dolarovou částkou. Neví, jaká konečná částka vznikne po rozšifrování, ale pokud udělá dostatečný počet pokusů, jeden z nich bude velký a ničivý. [6]

OpenVPN disponuje modulárním přístupem k šifrování. Většina šifrovacích funkcí je přeložena do OpenSSL knihovny. Vlastní ochranu proti pasivním útokům a známým typům aktivních útoků. Používá také již zmíněné ověření prostřednictvím veřejného klíče.

4.2 OpenVPN - třívrstvý bezpečnostní model

Známou pravdou je tvrzení říkající, že komplexnost je nepřítel počítačové bezpečnosti. Jednou z cest, jak redukovat dopad softwarové komplexnosti na celkovou bezpečnost software je přimět příchozí data procházet skrze nějaký druh bezpečné brány (např. firewall), která je jednoduší než aplikace běžící za ní. Cílem je redukce počtu řádek kódu, které může být dotčeno neautorizovanými pakety, což zároveň usnadňuje jejich kontrolu. OpenVPN expanduje na konceptu firewallu použitím `-tls-auth`. [7]

První vrstva zajišťuje:
Použití HMAC `-tls-auth` k eliminaci vkládání paketů do SSL/TLS.

Druhé vrstva zajišťuje:
Použití SSL/TLS pro dvousměrnou klient/server autentifikaci.

Třetí vrstva zajišťuje:
Přesunutí privilegovaného režimu OpenVPN daemona pomocí `user/group` k potlačení vloženého kódu.

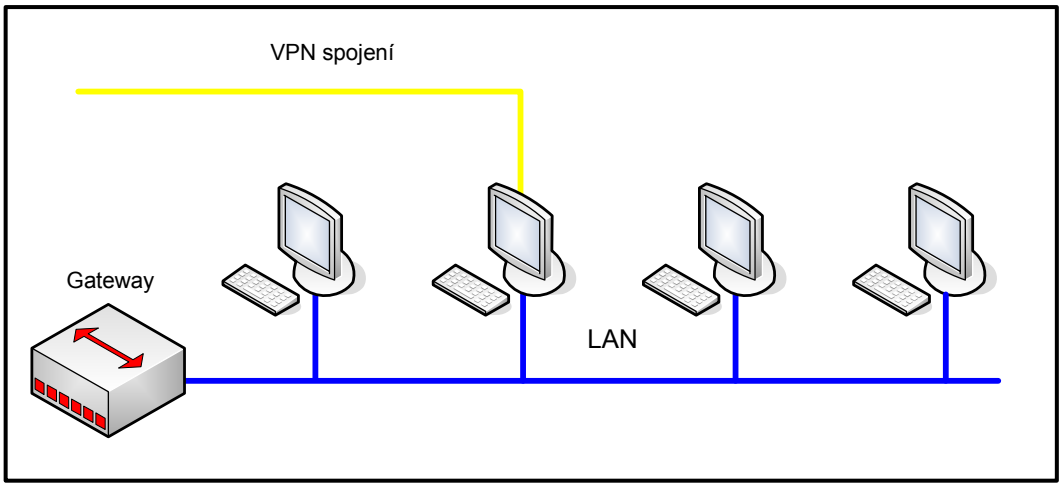
4.3 Zasíťování pomocí VPN

Existují dvě hlavní techniky pro síťování VPN. Jsou jimi směrování a přemostění. Přemostění je technika pro vytváření virtuálních rozlehlých ethernetových LAN běžících na jedné subsíti. Řeší problém rozsáhlých VPN využitím oddělených subsítí a nastavením směrování mezi nimi. [8]

Směrování umožňuje propojení oddělených nezávislých subsítí, které se nepřekrývají IP adresami. Při přijmutí poslaného paketu zkontroluje směrovač sítě umístění IP adres k určení, která z několika připojených sítí ho může přijmout, poté je tento paket poslán do vhodné sítě.

Síla směrování leží především v jeho efektivitě a škálovatelnosti. Pro funkčnost systému je nutné nastavit veškeré routovací tabulky zařízení, které se vyskytují na dané cestě

mezi komunikujícími konci. Zároveň při tomto způsobu propojení není funkční *broadcast*.



Obr. 20 OpenVPN směrování

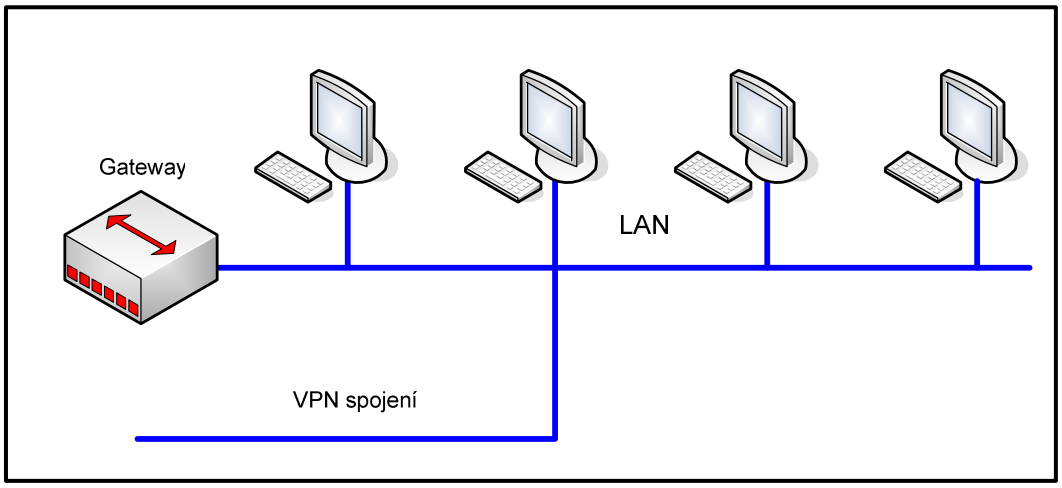
OpenVPN ve směrovacím módu vytvoří privátní síť sdílenou stroji, které jsou k ní připojeny zabezpečeným VPN tunelem. Popsaný způsob připojení je vhodný pro připojení ke stroji, na němž běží OpenVPN. Problém ovšem nastává v okamžiku potřeby připojení k jakémukoliv jinému stroji na lokální LAN síti. Žádné další zařízení na LAN totiž nemá informace o existenci běžící virtuální privátní sítě. Naštěstí lze tento problém vyřešit pomocí přidání routů ke stroji, ke kterému se chceme připojit. [8]

Jestliže je virtuální adaptér OpenVPN konfigurován pro směrování VPN spojů, je mu přidělena IP adresa mimo rozsah lokální LAN a zároveň je vytvořena oddělená virtuální subsíť pro připojení k jakémukoli vzdálenému stroji v této VPN síti. Z toho vyplývá, že virtuální subsíť musí nutně mít oddělený IP adresový prostor od fyzické LAN sítě. Jinak by stroj, na němž běží VPN nevěděl, jestli má poslat pakety do LAN, VPN nebo k internetové bráně. [8]

Alternativním řešením je použití přemostění. Při přemostění je možno přidělit vzdálenému uživateli IP adresu ze stejného rozsahu jako používají počítače na LAN. Díky tomu se vzdálený uživatel může připojit k jakémukoliv stroji umístěnému na této

LAN. Jelikož *bridge* funguje na rozdíl od směrovače na nižší vrstvě (stejně jako např. hub), projdou skrze něj veškeré ethernetové rámce (*broadcast*). Nevýhodou je možnost zahlcení sítě kvůli nefiltrování komunikace.

Přemostění je v kontrastu se směrováním mnohem snazší. Přemosťující síť je jednoduše mezičlánkem mezi oddělenými fyzickými sítěmi, které mohou mít stejné rozsahy IP adres. Při využití hubu jsou pakety přicházející na jakýkoliv port přemostěny a poslány na všechny porty. Přepínač je schopný adaptivního zjištění, jaké síťové karty jsou připojeny, kterým portům.



Obr. 21 OpenVPN přemostění

I přes to, že směrované spoje jsou užívanější a snáze konfigurovatelné jsou omezeny určitými limity. Naproti tomu přemostěné spoje jsou hůře konfigurovatelné a nejsou nativně podporované všemi operačními systémy. Pokud se je ale podaří zdárně nastavit, většinou zaručují veškeré požadované služby.

Pokud je cílem tvorba bezpečného mostu mezi serverem (linuxovým) a několika odlišnými klienty, tak tvorba probíhá následujícím způsobem:

- Vygenerování skupiny trvalých tap rozhraní
- Použitím `brctl` jsou přemostěny k reálnému ethernetovému adaptéru
- V okamžiku připojení klientů se jejich tap rozhraní přidělí IP adresa

Určení IP adres při přemostění ilustruje následující tabulka:

Nastavení	**Startovací parametr**	**hodnota**
Ethernetové rozhraní	eth	eth0
Lokální IP adresa	ip	192.168.8.4
Lokální maska sítě	eth_netmask	255.255.255.0
Lokální adresa *broadcast*	eth_broadcast	192.168.8.255
Adresový prostor klienta VPN		192.168.8.128 až 192.168.8.254
Virtuální *bridge* rozhraní	br	br0
Virtuální TAP rozhraní	tap	tap0

Tab. 1 nastavení IP při přemostění [6]

OpenVPN se snaží využít všech výhod, kterými disponuje SSL VPN:

- Portability
- Známého uživatelského prostředí
- Žádné modifikace jádra
- Šifrování je zaručeno OpenSSL knihovnou
- Komfortní používání s dynamickým přidělováním adres nebo NAT.
- Podporuje většinu operačních systémů (Linux Windows Mac OS X , BSD a Solaris)

5 Instalace a Konfigurace OpenVPN

OpenVPN bylo stvořeno pro vysokou portabilitu, tj., běh na většině operačních systémů (Windows 2000/XP, Linux, Solaris, BSD, a Mac OS X). Díky práci v rámci neprivilegovaného režimu je instalace snadná. Instalační soubor pro Windows lze stáhnout na stránkách openvpn.net, kde je dostupná kompletní dokumentace. Na stejných stránkách lze stáhnout také instalační soubory pro Linux, pokud již nejsou součástí balíčků, nebo v případě potřeby instalace odlišných verzí. Pro bezproblémovou instalaci je doporučována verze kernelu 2.4 a vyšší, v níž jsou zabudovány ovladače pro virtuální rozhraní pro tun a tap.

OpenVPN je stále ve vývoji. Aktuální stabilní verze 2.0.9 oproti předchozí 1.6 a nižším nevyžaduje použití odděleného UDP portu, konfiguračního souboru a tun/tap rozhraní pro každé VPN spojení. Díky tomu se dramaticky zjednodušuje správa více spojů. Použitím voleb `--ifconfig-pool`, `--push`, a `–pull` lze odeslat vzdálené adresy a stáhnout či odeslat mnoho konfiguračních voleb k usnadnění nastavení a údržby vzdálených strojů, což je výhoda zejména pro cestující pracovníky, kteří mění při připojení často IP adresy. Dostupná betaverze 2.1 má integrovanou podporu Windows Vista x86 i x64 architektury a v jejím instalačním balíku je přibaleno grafické uživatelské rozhraní. `TAP-Win32`, adaptér může být spuštěn v uživatelském režimu a další.

5.1 Instalace OpenVPN

5.1.1 Instalace a konfigurace Linuxového serveru

Instalace bude popsána pro distribuci Mandriva za použití RPM balíčku. Nejprve je nutno instalovat vlastní OpenVPN pomocí příkazu `urpmi openvpn`. Následně je třeba zkopírovat vzorové skripty, které se nejčastěji nachází v tomto adresáři:

```
cp -R /usr/share/openvpn/easy-rsa/ /etc/openvpn
```

(Volba –R specifikuje postup kopírování FIFO)

V cílovém adresáři se generují klíče, proto je dobré nastavit práva:

```
chmod 700 /etc/openvpn/easy-rsa
```

`V souboru /etc/openvpn/easy-rsa/2.0/vars` se nastavují proměnné pro tvorbu klíčů.

```
# Increase this to 2048 if you
# are paranoid.  This will slow
# down TLS negotiation performance
# as well as the one-time DH parms
# generation process.
export KEY_SIZE=1024

# In how many days should the root CA key expire?
export CA_EXPIRE=3650

# In how many days should certificates expire?
export KEY_EXPIRE=3650

# 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="Czech Republic"
export KEY_CITY="Praha"
export KEY_ORG="Zlatnik"
export KEY_EMAIL="p.zlatnik@gmail.com"
```

V prvním odstavci se nastavuje velikost *Diffie-Hellman* parametrů. Pro běžný provoz stačí 1024. Je zde i možnost 2048, která ale zpomalí navazování spojení.

V následujících dvou se určuje délka platnosti CA klíče resp. certifikátů.

Poslední odstavec obsahuje hodnoty, které budou zapsány v certifikátu.

Obr. 22 parametry pro tvorbu klíčů

Po upravení souboru ho je třeba načíst pomocí příkazů:

```
cd /etc/openvpn/easy-rsa/2.0
. ./vars
```

Vymazání vzorových klíčů:

```
./clean-al
```

Tvorba CA klíče:

```
./build-ca
```

Tvorba certifikátu pro server:

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

Certifikáty klientů (druhý je chráněn heslem):

```
./build-key client1
./build-key-pass client2
```

Tvorba *Diffie-Hellman* parametrů (v případě volby 2048 pár minut, jinak pár sekund):
`./build-dh`

Název	Požadováno pro	Účel	Tajné
ca.crt	Server+všichni klienti	Kořenový CA certifikát	Ne
ca.key	Pouze potvrzení klíče	Kořenový CA klíč	Ano
dh{n}.pem	Pouze server	Parametry Diffie Helman	Ne
server.crt	Pouze server	Certifikát serveru	Ne
server.key	Pouze server	Klíč serveru	Ano
client1.crt	Pouze klient 1	Certifikát klienta 1	Ne
client1.key	Pouze klient 1	Klíč klienta 1	Ano
client2.crt	Pouze klient 2	Certifikát klienta 2	Ne
client2.key	Pouze klient 2	Klíč klienta 2	Ano

Tab. 2 vytvořené certifikáty a klíče

Zkopírování certifikátů a klíčů:

```
cp /etc/openvpn/easy-
rsa/2.0/keys/{ca.crt,server.crt,server.key,dh1024.pem} /etc/openvpn
```

A nastavení přůíslušných práv:

```
chmod 600
/etc/openvpn/easy-rsa/2.0/keys/{ca.crt,server.crt,server.key,dh1024.pem}
/etc/openvpn
```

Dále je třeba vytvořit konfigurační soubor `/etc/openvpn/vpn_s.conf` (komentář je označen # resp. ";" (podrobněji jsou jednotlivé volby rozepsány v příloze)):

```
# udává režim serveru
mode server
# tls jako klient
tls-server
# určuje používaný port
port 1194
```

```
# TCP či UDP server
proto tcp-server
# “dev tun" vyrobí směrovaný IP tunel
dev tun
# konfigurace nastavení serveru a podporované VPN subsítě
server 10.0.1.0 255.255.255.0
# certifikát CA
ca /etc/openvpn/ca.crt
# certifikát serveru
cert /etc/openvpn/server.crt
# klíč serveru
key /etc/openvpn/server.key
# Diffie Hellman parametry
dh /etc/openvpn/dh1024.pem
# logy serveru
log-append /var/log/openvpn.log
# status serveru
status /var/run/vpn.status 10
# snížení privilegovanosti
user nobody
group nobody
# udržuje spojení
keepalive 10 120
# komprese prenasenych dat
comp-lzo
# podrobnost log souboru
verb 4
# skrytí opakovaných zpráv
mute 20
```

Spuštění a následná kontrola logu:

```
[root@localhost openvpn]# /etc/init.d/openvpn start
Startuji openvpn:                                          [  OK  ]
[root@localhost openvpn]# tail -f /var/log/openvpn.log
Mon Apr  6 10:49:53 2009 GID set to nogroup
Mon Apr  6 10:49:53 2009 UID set to nobody
Mon Apr  6 10:49:53 2009 Listening for incoming TCP connection on [undef]:1194
Mon Apr  6 10:49:53 2009 Socket Buffers: R=[87380->131072] S=[16384->131072]
Mon Apr  6 10:49:53 2009 TCPv4_SERVER link local (bound): [undef]:1194
Mon Apr  6 10:49:53 2009 TCPv4_SERVER link remote: [undef]
Mon Apr  6 10:49:53 2009 MULTI: multi_init called, r=256 v=256
Mon Apr  6 10:49:53 2009 IFCONFIG POOL: base=10.0.1.4 size=62
Mon Apr  6 10:49:53 2009 MULTI: TCP INIT maxclients=1024 maxevents=1028
Mon Apr  6 10:49:53 2009 Initialization Sequence Completed
```

Obr. 23 spuštění serveru

Ještě lze zkontrolovat, jestli opravdu přibylo virtuální síťové rozhraní příkazem `ifconfig`:

```
tun0      Link encap:NEZNÁM  HWadr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet adr:10.0.1.1  P-t-P:10.0.1.2 Maska:255.255.255.255
          AKTIVOVÁNO POINTOPOINT BĚŽÍ NEARP MULTICAST  MTU:1500  Metrika:1
          RX packets:17 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17 errors:0 dropped:0 overruns:0 carrier:0
          kolizí:0 délka odchozí fronty:100
          RX bytes:1596 (1.5 KiB)  TX bytes:1428 (1.3 KiB)
```

Obr. 24 tun rozhraní serveru

5.1.2 Instalace a konfigurace Linuxového klienta

Nejprve je třeba nainstalovat OpenVPN, viz konfigurace serveru. Poté je nutno vytvořit konfigurační soubor `/etc/openvpn/vpn_c.conf` s podobnou strukturou jako má server.

```
# adresa serveru
remote 192.168.3.62
# tls jako klient
tls-client
# určuje používaný port
port 1194
# tcp klient
proto tcp-client
# “dev tun" vyrobí směrovaný IP tunel
dev tun
# umožňuje stažení konfigurací ze serveru
pull
# certifikát CA
ca ca.crt
# certifikát klienta
cert client1.crt
# klíč klienta
key client1.key
# sníženi privilegovanosti
user nobody
group nobody
# logy klienta
log-append /var/log/openvpn.log
# status klienta
status /var/run/vpn.status 10
# komprese přenášeních dat
comp-lzo
# podrobnost log souboru
verb 4
```

Po jeho vytvoření je třeba bezpečnou cestou přenést soubory `ca.crt`, `client1.crt` a `client1.key` do adresáře `/etc/openvpn`. Tyto soubory se nachází v adresáři `/etc/openvpn/easy-rsa/2.0/keys` serveru. Poté je možno známým příkazem spustit OpenVPN.

```
[root@localhost petr]# /etc/init.d/openvpn start
Startuji openvpn:                                          [  OK  ]
[root@localhost petr]# tail -f /var/log/openvpn.log
  <AF>=Use '-A <af>' or '--<af>'; default: inet
  List of possible address families (which support routing):
    inet (DARPA Internet) inet6 (IPv6) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
    x25 (CCITT X.25)
Mon Apr  6 21:54:20 2009 /sbin/route add -net 10.0.1.1 netmask 255.255.255.255 gw 10.0.1.5
Mon Apr  6 21:54:20 2009 GID set to nogroup
Mon Apr  6 21:54:20 2009 UID set to nobody
Mon Apr  6 21:54:20 2009 Initialization Sequence Completed
```

Obr. 25 spuštění klienta

Kontrola tun rozhraní:

```
tun0      Link encap:NEZNÁM  HWadr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet adr:10.0.1.6  P-t-P:10.0.1.5 Maska:255.255.255.255
          AKTIVOVÁNO POINTOPOINT BĚŽÍ NEARP MULTICAST  MTU:1500  Metrika:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          kolizí:0 délka odchozí fronty:100
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Obr. 26 tun rozhraní klienta

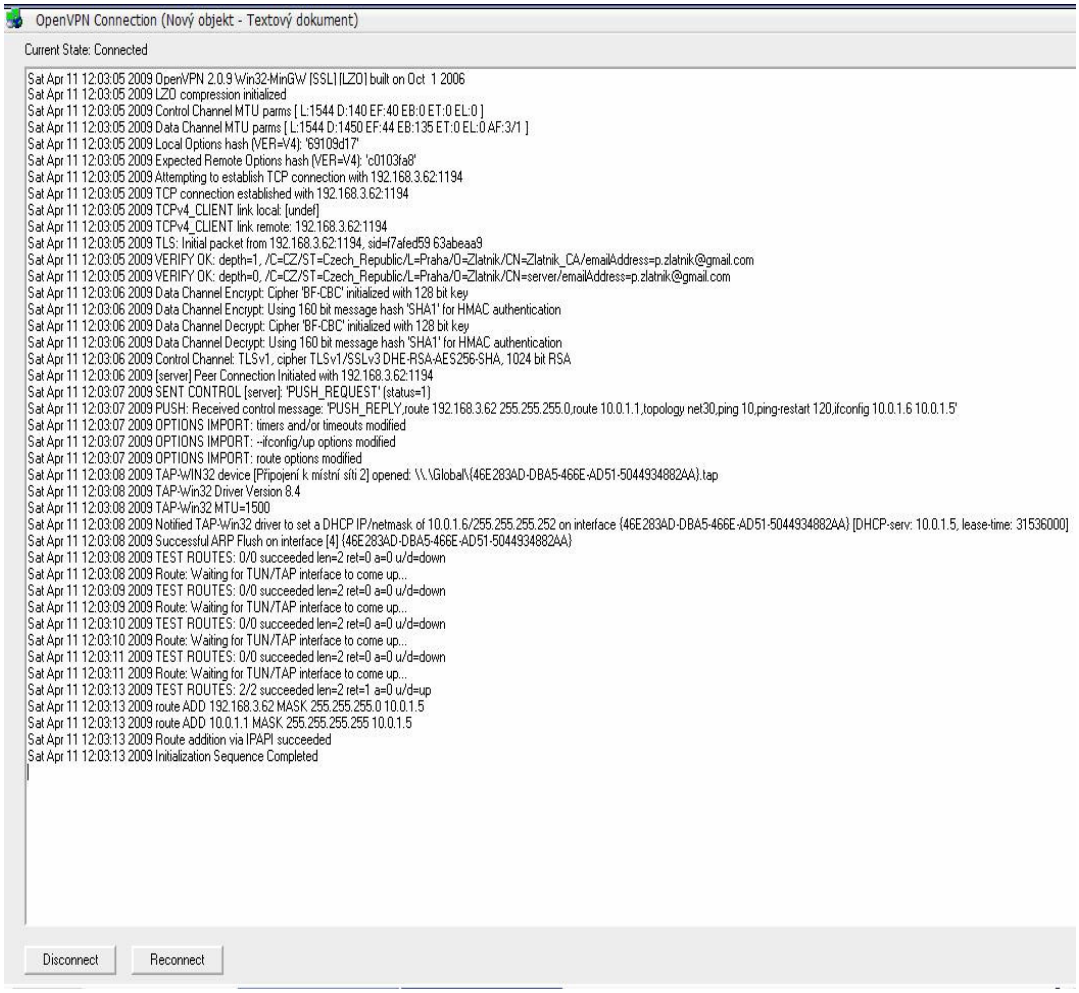
Test spojení:

```
[root@localhost petr]# ping 10.0.1.1
PING 10.0.1.1 (10.0.1.1) 56(84) bytes of data.
64 bytes from 10.0.1.1: icmp_seq=1 ttl=64 time=1.99 ms
64 bytes from 10.0.1.1: icmp_seq=2 ttl=64 time=22.4 ms
64 bytes from 10.0.1.1: icmp_seq=3 ttl=64 time=2.11 ms
64 bytes from 10.0.1.1: icmp_seq=4 ttl=64 time=2.63 ms
64 bytes from 10.0.1.1: icmp_seq=5 ttl=64 time=2.07 ms
64 bytes from 10.0.1.1: icmp_seq=6 ttl=64 time=11.2 ms
64 bytes from 10.0.1.1: icmp_seq=7 ttl=64 time=3.13 ms
64 bytes from 10.0.1.1: icmp_seq=8 ttl=64 time=2.08 ms
64 bytes from 10.0.1.1: icmp_seq=9 ttl=64 time=6.99 ms
64 bytes from 10.0.1.1: icmp_seq=10 ttl=64 time=2.12 ms
64 bytes from 10.0.1.1: icmp_seq=11 ttl=64 time=27.8 ms
64 bytes from 10.0.1.1: icmp_seq=12 ttl=64 time=2.53 ms
64 bytes from 10.0.1.1: icmp_seq=13 ttl=64 time=2.07 ms
64 bytes from 10.0.1.1: icmp_seq=14 ttl=64 time=2.10 ms
^C
--- 10.0.1.1 ping statistics ---
14 packets transmitted, 14 received, 0% packet loss, time 13017ms
rtt min/avg/max/mdev = 1.995/6.533/27.896/8.079 ms
```

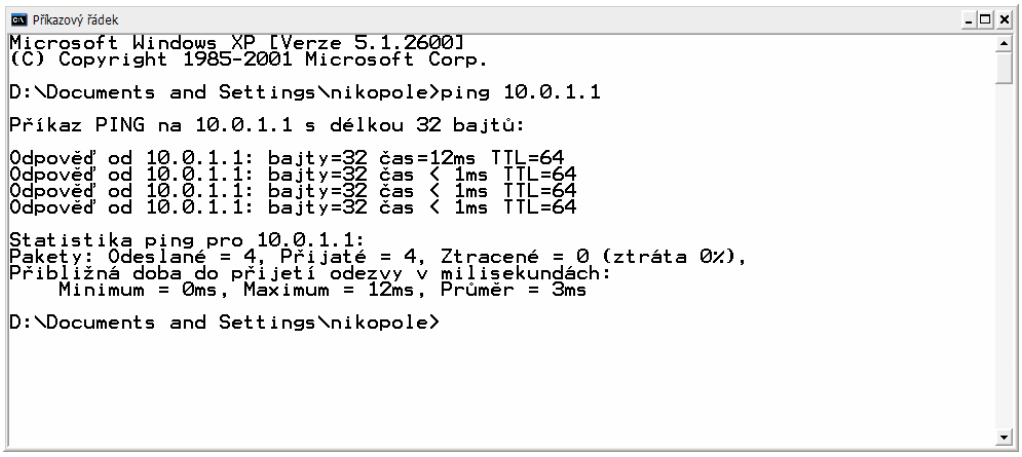
Obr. 27 test spojení Linux

5.1.3 Instalace a konfigurace klienta pod Windows XP SP2

Při instalaci klienta pro Windows je nutno stáhnout instalační soubor ze stránek openvpn.net. Pro pohodlnější spouštění je dobré instalovat také OpenVPN GUI například z openvpn.se. Po instalaci je třeba do start --> všechny programy --> openvpn --> openvpn configuration file directory přenést bezpečnou cestou soubory `ca.crt`, `client1.crt` a `client1.key`, které se nachází v adresáři `/etc/openvpn/easy-rsa/2.0/keys` serveru. Zároveň ve stejném adresáři vytvořit konfigurační soubor `vpn_c.ovpn` se stejnou strukturou jako u klienta na Linuxu, vyjma příkazů `log-append`, `status`, user `nobody` a `group nobody`. Po spuštění OpenVPN GUI lze zkontrolovat login a otestovat spojení:



Obr. 28 login z Windows GUI



Obr. 29 test spojení Windows XP

5.2 Zacílení OpenVPN (porovnání s IPSec)

Jak je vidět instalace a konfigurace není nijak složitá. Zároveň ale OpenVPN nabízí mnoho služeb, které je možno k základní funkcionalitě přidat. Samozřejmostí je podpora směrování i přemostění, velkou výhodou je oproti IPSec řešení dobrá použitelnost pro cestující pracovníky, kterým stačí pouze předat konfigurační soubor, certifikáty a klíče. Odpadá tak nutnost samostatné konfigurace každého klientského stroje. Další už zmíněnou nespornou výhodou je také běh v uživatelském módu, který vylučuje možnost kritického selhání.

Jak už název napovídá OpenVPN je *open source* řešení, tj., užívání programu je zcela zdarma, dostupný je i zdrojový kód. Platí se pouze v případě, že je tento zdrojový kód upraven a následně určen ke komerčním účelům. Nabízí se otázka: „Proč tedy vůbec používat IPSec?“ Logicky pokud SSL zajišťuje stejnou ochranu a placené implementace navíc ani nevyžadují instalaci klienta, protože se spouští automaticky pomocí Java appletu, mohl by se zdát IPSec přežitkem. Ale v případě, že je třeba šifrovat veškerý síťový provoz mezi dvěma sítěmi, je IPSec ideálním řešením. Šifrování totiž probíhá na úrovni IP protokolu a díky tomu je pro aplikace a síťové prvky transparentní. Také lze těžko přistupovat k nějaké síti, kde už je IPSec implementován, kvůli jeho logické nekompatibilitě s SSL. Proto je někdy používán i v případech, kdy by bylo spojení

možné řešit za pomoci SSL. Velmi zjednodušeně by mohl být rozdíl mezi IPSec a SSL popsán takto:

- IPSec je univerzální v tom, kam můžu přistupovat
- SSL je univerzální v tom, odkud můžu přistupovat

Atributy	IPSec VPN	SSL VPN
Podporované aplikace:		
Mainframe aplikace	Ano	Ano
Klient/server	Ano	Ano
HTTP	Ano	Ano
Sdílení souborů	Ano	Ano
Pracovní prostředí:		
Přístup bez klienta	Ne	Placené implementace
Podpora bezdrátových zařízení	Ano	Ano
Konektivita s pomocí Java apletů	Ne	Placené implementace
Podporovaný přístup:		
Z PC ve firmě	Ano	Ano
Z domova nebo kavárny	Různě	Ano
Z veřejného PC	Ne	Placené implementace
Z PDA	Ano	Ano
Další důležité atributy:		
Rentabilní instalace konfigurace a správa	Ne	Ano
Jednoduchý provoz v jakékoli síti bez nastavování	Ne	Ano
Snadno průchozí NAT a firewall	Ne	Ano
Běh v režimu	Jádra	Neprivilegovaném
Nejvhodnější použití:		
Site-to-site konktivita	Ano	Spíše ne
Mobilní přístup	Spíše ne	Ano

Tab. 3 porovnání SSL a IPSec [11]

6 Závěr

Virtualizace se čím dál více prosazuje do světa informačních technologií a privátní sítě nejsou v tomto směru výjimkou. SSL VPN přináší plnohodnotné řešení v oblasti vzdálené konektivity s nespornými přednostmi a to zejména pro cestující pracovníky.

Lídrem v uvedené oblasti je bezesporu *open source* řešení OpenVPN, které nabízí propracovanou konstrukci VPN s mnoha volbami, ale zároveň uchovává jednoduchost, což je základní směrování, kterým se musejí v budoucnu ubírat veškeré masově rozšířené technologie. Je zřejmé, že je velmi obtížně možné, aby ve velkých společnostech se stovkami uživatelů specializovaný pracovník konfiguroval veškeré stroje manuálně. V tomto ohledu se jeví zajímavě konstrukce komerčních SSL řešení, která umožňují tvorbu spojení pomocí Java appletů, bez nutnosti jakékoliv instalace do klientského stroje. Je nutno podotknout, že v případě OpenVPN je u klientského stroje instalace velmi snadná a nevyžaduje žádné speciální znalosti, ale přeci jen se instalovat musí.

Pro velké *site-to-site* spoje se zdá být ale stále výhodnějším řešením využití IPSec protokolu z důvodu jeho transparentnosti a větší robustnosti. Na koexistenci obou protokolů pružně reagují výrobci VPN bran, kteří do svých zařízení začali implementovat podporu pro obě technologie, a tak je v dnešní době už pouze na uvážení společnosti, kterou z technologií se rozhodne používat. Nebo, což je asi nejlepším řešením, obě technologie kombinovat.

Často jsou virtuální privátní sítě chápány jako určitá náhrada dedikovaných linek, které z velké části opravdu nahrazují a se zvyšujícími rychlostmi internetové konektivity je budou asi nahrazovat čím dál více. Ve velkém rozsahu se ale také používají k zabezpečení přenosu realizovaného pomocí dedikovaných spojů, a to z důvodu sdílení těchto linek více odděleními jedné společnosti, nebo prostě z důvodu zvýšení bezpečnosti.

Elektronická verze práce je v upravené podobě k dispozici na *e-learningovém* serveru moodle.czu.cz v rámci kurzu „Dobrovolný seminář bezpečnosti“. Doufám, že přispěje jeho účastníkům k lepší orientaci v popsané tématice a umožní snazší nastavení v případě použití OpenVPN implementace.

Technologie VPN je silným, rychle se vyvíjejícím prostředkem vzdálené konektivity. Mezi hlavní přednosti SSL implementace lze zařadit jednoduchost, portabilitu a relativně nízké náklady na konstrukci i správu. V budoucnu lze očekávat ještě masivnější rozšíření, které zapříčiňují současné trendy globalizace.

V této práci jsem prakticky prověřil OpenVPN. Uvedl jsem popis, konfiguraci a porovnání vůči IPSec. Rád bych se touto problematikou zabýval i ve své případné budoucí diplomové práci.

7 Seznam literatury a použitých zdrojů

1 LUCAS Mark, SINGH Abhishek, CANTRELL Chris
Firewall policies and VPN configurations
Rockland : Syngress, c2006. 465 s.
ISBN 1-59749-088-1

2 JAZIB Frahim, QIANG Huang
SSL remote access VPNs
Indianopolis : Cisco Press, c2008. 349 s.
ISBN 1-58705-242-2

3 HONTAŇÓN Ramón
LINUX - praktická bezpečnost
Praha: Grada Publishing a.s., c2003. 438 s.
ISBN 80-247-0652-0

4 Bezpalec Pavel
Přednášky z předmětu datové sítě
ČVUT FEL

5 Luhový Karel
Virtuální privátní sítě VPN [online], c2003
[cit. 2009-03-1]
http://www.svetsiti.cz/view_list.asp?rubrika=Tutorialy&temaID=219

6 Yonan James, Dinha Francis
The User-Space VPN and OpenVPN [online], c2008
[cit. 2009-03-4]
http://openvpn.net/

7 Hosner Charlie
OpenVPN and the SSL VPN Revolution.
SANS: Bethesda [online], c2004
[cit. 2009-03-5]
http://www.sans.org/reading_room/whitepapers/vpns/1459.php

8 Gibson Steve
Routing versus Bridging [online], c2008
[cit. 2009-03-10]
http://www.grc.com/vpn/routing.htm

9 Attel
VPN - Virtuální privátní síť [online], c2005
[cit. 2009-03-15]
http://www.attel.cz/typova-reseni/vpn-virtualni-privatni-sit/

10 Openmaniak
OpenVPN – Tutoriál [online], c2006
[cit. 2009-02-20]
http://openmaniak.com/openvpn.php

11 Ajoomal Asociados
IPSec vs. SSL VPNs for Secure Remote Access [online], c2005
[cit. 2009-02-10]
http://www.ajoomal.com/descargas/aventail/IPSec_vs1%20_SSL_VPNs_For_Secure_Remote_Access_-_English_(A4).pdf

8 Seznam obrázků

9 Seznam tabulek

10 Seznam příloh

Příloha 1 konfigurační soubor serveru

Příloha 2 konfigurační soubor klienta

Příloha 1

Konfigurační soubor serveru:

zdroj: [6]

```
# Komentáře jsou označeny “#“ nebo “;“

# Jakou lokální IP adresu má OpenVPN
# používat? (volitelně)
;local a.b.c.d

# tls jako server
tls-server

# udává režim serveru
mode server

# Jaký TCP/UDP port má OpenVPN používat?
# V případě využívání několika instancí OpenVPN
# na stejném stroji, je nutno použít pro každou
# odlišné číslo portu.
# Tento port je nutno otevřít pro firewall
port 1194

# TCP či UDP server?
proto tcp-server
;proto udp

# "dev tun" vyrobí směrovaný IP tunel,
# "dev tap" vyrobí ethernetový tunel.
# v případě požadavku ethernetového přemostění
# je nutno použít "dev tap0", zároveň
# je vytvořeno tap0 virtuální rozhraní
# přemostěné pomocí ethernetového rozhraní.
# Na systémech vyjma Windows je možno přidělit
# explicitní číslo jednotky, jako tun0.
# Na Windows, je doporučeno použít "dev-node"
# Na většině systémů, VPN nebude fungovat
# do doby než je částečně či úplně vypnut
# firewall pro TUN/TAP rozhraní.
;dev tap
dev tun

# Windows vyžaduje pojmenování TAP-Win32 adaptéru
# z panelu síťového připojení, jestliže je využíván
# více než jeden.  Na XP SP2 nebo vyšší, je nutno
# selektivně zakázat firewall pro TAP adaptér.
# OS mimo Windows to většinou nepožadují.
;dev-node MyTap
```

```
# SSL/TLS kořenový certifikát (ca), certifikát
# (cert) a privátní klíč (key).  Každý klient
# a server musí mít vlastní cert a key soubor.
# Server a všichni klienti používají stejný ca soubor.
# Pro generování RSA certifikátu a privátních klíčů
# je použit "easy-rsa" adresář, v němž je sada skriptů.
# Je nutno nezapomenout zadat jedinečné Common
# jméno certifikátu pro server a všechny klienty.
ca /etc/openvpn/ca.crt
cert /etc/openvpn/server.crt
key /etc/openvpn/server.key # tento soubor by měl být utajen

# Diffie hellman parametry.
# Vlastní jedinečné parametry jsou generovány:
# openssl dhparam -out dh1024.pem 1024
# Pokud jsou použity 2048 bit klíče
# je nutno nahradit 2048 za 1024
dh /etc/openvpn/dh1024.pem

# Konfigurace nastavení serveru a podporované VPN subsítě.
# pro OpenVPN za účelem nastavení adres klientů.
# Server použije pro sebe 10.0.1.1, zbylé adresy
# budou dány k dispozici klientům. Každý klient
# bude moci navázat spojení se serverem na 10.0.1.1.
# V případě přemostění je nutno následující řádek
# zakomentovat.
server 10.0.1.0 255.255.255.0

# Udržování virtuální IP adresy klienta
# v následujícím souboru. Jestliže OpenVPN spadne nebo
# je restartováno, znovunavázání spojení s klienty je určeno
# stejnou virtuální IP adresou,
# jenž byla před ukončením spojení přiřazena.
;ifconfig-pool-persist ipp.txt

# Konfigurace nastavení serveru pro ethernetové přemostění.
# Je nutno manuálně nastavit IP/masku sítě
# na přemosťujícím zařízení
# Poté je nutno nastavit rozsah IP pro subsíť.
# (start=10.0.1.2 end=10.0.1.100) vymezující
# rozsah pro připojení klientů.
;ifconfig 10.0.1.1 255.255.255.0
;ifconfig-pool 10.0.1.2 10.0.1.100 255.255.255.0
```

```
# Přidělení záznamu ze směrovacích tabulek
# umožní klientu připojení k subsíti
# umístěné za serverem.
;push "route 192.168.10.0 255.255.255.0"
;push "route 192.168.20.0 255.255.255.0"
# Určitá specifická síťová nastavení Windows
# mohou být poskytována klientům, např. DNS
# či WINS servery.
;push "dhcp-option DNS 10.0.1.1"
;push "dhcp-option WINS 10.0.1.1"

# Pokud je požadováno „vidění“ se klientů navzájem
# (defaultně klienti vidí pouze server).
# K zajištění, že klienti uvidí pouze server je
# je také nutné zabezpečit TUN/TAP rozhraní
# pomocí firewallu.
;client-to-client

# Pokud je požadováno připojení několika klientů se
# stejným certifikátem/klíčem, přičemž každý má vlastní
# common jméno. Jedná se spíše o testovací nastavení
# Pro ostrý provoz je silně doporučováno aby
# měl každý klient vlastní pár certifikát/klíč.
;duplicate-cn

# Příkaz keepalive zapříčiní odesílání ping
# zpráv zpět a vpřed přes tunel, k zajištění
# zpětné odezvy, v případě odpojení jedné strany.
# Ping je zasílán každých 10 sekund. Odpojení
# druhé strany je předpokládáno, jestliže není přijat ping
# v časovém intervalu 120 ti sekund.
keepalive 10 120

# Výběr šifrovacího mechanismu.
# Konfigurační položka musí být zkopírována
# do klientova konfiguračního souboru.
;cipher BF-CBC        # Blowfish (default)
;cipher AES-128-CBC   # AES
;cipher DES-EDE3-CBC  # Triple-DES

# Umožnění komprese ve VPN spoji.
# Jestliže je povolen následující příkaz
# musí být komprese také povolena
# v konfiguračním souboru klienta
comp-lzo
```

```
# Nastavení maximálního počtu
# souběžně připojených klientů
;max-clients 100

# Po inicializaci OpenVPN deamona
# je dobré snížit jeho privilegovanost
# (na systémech kromě Windows)
user nobody
group nobody

# Volba persist se snaží zabránit připojení
# některých prostředků po restartu, které by
# neměly být přístupné z důvodu snížení práv.
;persist-key
;persist-tun

# Výstup short status souboru ukazuje
# aktuální připojení, aktualizované
# a přepsané každých 10 sekund.
status /var/run/vpn.status 10

# Standardně se logovací zprávy vypisují na syslog
# (na Windows, jdou do "\Program Files\OpenVPN\log" directory).
# Lze použít log nebo log-append k přepisu defaultního nastavení.
# "log"přesune log soubor na OpenVPN startup,
# "log-append" se k němu připojí
# lze samozřejmě použít jenom jednu volbu.
;log /var/log/openvpn.log
log-append /var/log/openvpn.log

# Nastavení podrobnosti log souboru
# 0 tichý režim, vyjma fatálních chyb
# 4 doporučeno pro běžné uživatele
# 5 a 6 pro opravení problémů s konektivitou
# 9 extrémně podrobné
verb 3

# Skrytí opakovaných zpráv.  Nejčastěji je
# používáno, že bude 20 opakovaných výpisů
# stejné zprávy bude vypsáno do log souboru.
mute 20
```

Příloha 2

Konfigurační soubor klienta:

Zdroj: [6]

```
# Specifikace klienta
# jsou přebírány některé
# konfigurační soubory serveru
mode-client

# tls jako klient
tls-client

# umožňuje stažení konfigurací ze serveru
pull

# Použití stejných nastavení jako na
# serveru
;dev tap
dev tun

# Windows vyžaduje pojmenování TAP-Win32 adaptéru
# z panelu síťového připojení, jestliže je využíván
# více než jeden. Na XP SP2 nebo vyšší, je nutno
# selektivně zakázat firewall pro TAP adaptér.
;dev-node MyTap

# Připojování je uskutečňováno k TCP
# UDP serveru? Použije se stejné
# nastavení jako na serveru
proto tcp
;proto udp

# Hostname/IP a port serveru.
# Pro rovnoměrné vytížení
# je možno využívat několik serverů
remote 192.168.3.62 1194
;remote 192.168.3.63 1194

# Pro rovnoměrné vytížení lze vybrat
# náhodně hostitele.  Nebo
# ho přesně specifikovat
;remote-random

# Umožňuje přidělit hostitelské jméno
# OpenVPN serveru na neurčitou dobu.
# Používané na strojích, které nejsou
# trvale připojeni internetu např. notebooky
;resolv-retry infinite
```

```
# Většina klientů se neváže
# na specifické místní číslo portu
;nobind

# Snížení privilegovanosti po inicializaci, vyjma Windows
user nobody
group nobody

# Snaha o uchování některých informací po restartu.
;persist-key
;persist-tun

# Pokud je spojení realizováno přes HTTP proxy
# zadává se proxy server/IP a
# číslo portu následovně.
;http-proxy-retry
;http-proxy [proxy server] [proxy port #]

# Bezdrátové sítě produkují množství
# duplicitních paketů. Pro skrytí
# warningů duplikovaných paketů.
;mute-replay-warnings

# SSL/TLS parametry
# Používá se oddělený pár .crt/.key souborů
# pro každého klienta. Jeden ca soubor může
# může být používán pro všechny klienty.
ca ca.crt
cert client.crt
key client.key

# Ověření certifikátu serveru ověřením,
# jestli má pole nsCertType
# nastavené na "server".  Jedná se
# o důležitou ochranu proti potenciálním
# man-in-the middle útokům.
#
# Pokud je požadováno použití uvedené vlastnosti
# je nutné generovat certifikáty serveru s nsCertType
# polem nastaveným na "server". Toto obstarává
# build-key-server skript v easy-rsa složce.
;ns-cert-type server
```

```
# Výběr šifrovacího algoritmu.
# Zde musí být specifikován stejný
# mechanismus jako na straně serveru.
;cipher BF-CBC

# Umožnění komprese na VPN spoji.
# Může být využita pouze, pokud je
# povolena v konfiguračním souboru serveru.
comp-lzo

# nastavení podrobnosti výpisu.
verb 3

# Skrytí opakovaných zpráv.
mute 20
```