VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

LABORATORNÍ ÚLOHA VIRTUÁLNÍ SÍTĚ TYPU OPENVPN

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

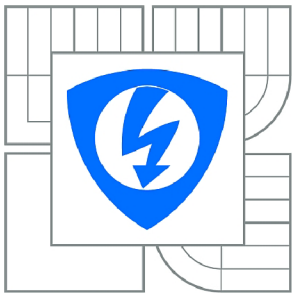
AUTOR PRÁCE
AUTHOR

JIŘÍ KORTUS

BRNO 2012

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

LABORATORNÍ ÚLOHA VIRTUÁLNÍ SÍTĚ TYPU OPENVPN

LABORATORY EXERCISE "OPENVPN VIRTUAL NETWORKS"

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

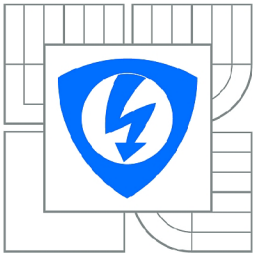
AUTOR PRÁCE — JIŘÍ KORTUS

AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. KAREL BURDA, CSc.

SUPERVISOR

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student:	Jiří Kortus	***ID:***	126760
Ročník:	3	***Akademický rok:***	2011/2012

NÁZEV TÉMATU:

Laboratorní úloha Virtuální sítě typu OpenVPN

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte a popište princip, základní komponenty a způsob fungování virtuálních sítí typu OpenVPN. Na tomto základě navrhněte koncept laboratorní úlohy s názvem Virtuální síť OpenVPN. Svůj návrh zdůvodněte, prakticky zrealizujte, otestujte a zhodnoťte. Pro navrženou úlohu zpracujte návod, ve kterém dbejte na pochopitelnost, srozumitelnost a přehlednost.

DOPORUČENÁ LITERATURA:

[1] Krčmář, P.: Linux: postavte si počítačovou síť. Grada Publishing. Praha 2008.
[2] - : OpenVPN Access Server System Administrator Guide. OpenVPN Technologies. Pleasanton 2010.

Termín zadání:	6.2.2012	***Termín odevzdání:***	31.5.2012

Vedoucí práce: doc. Ing. Karel Burda, CSc.
Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

ABSTRAKT

Tato práce v úvodu popisuje principy virtuálních privátních sítí (VPN) a stručně zmiňuje hlavní protokoly a přístupy pro jejich realizaci. Následně se podrobněji zaměřuje na seznámení čtenáře s charakterem a vlastnostmi sítí OpenVPN. Uvádí jednotlivé rysy a možnosti, diskutuje jejich výhody a nevýhody.

Dále se věnuje návrhu laboratorní práce zaměřené na použití OpenVPN. Diskutuje možnosti návrhu práce zejména z technického a didaktického hlediska. Na jejich základě představuje koncept laboratorní úlohy s jedním OpenVPN serverem a jedním OpenVPN klientem, která simuluje řešení v praxi se vyskytujícího problému s nežádoucím omezením připojení k Internetu špatně nebo příliš restriktivně nastaveným firewallem. Součástí konceptu je jak zadání, tak vlastní implementace virtuálních strojů použitých ve cvičení. Dále jsou popsány možnosti rozšíření navržené úlohy nebo možnosti použití jako základu pro nové úlohy.

KLÍČOVÁ SLOVA

VPN virtuální privátní síť OpenVPN laboratorní úloha SSL TLS Linux

ABSTRACT

This thesis describes the principles of virtual private networks (VPN) and briefly mentions the main protocols and methods for VPN deployment. After that, it aims on a more detailed description of the OpenVPN characteristics. It describes its properties and possibilities and discusses their advantages and disadvantages.

In the following part, it focues on a laboratory exercise concerning OpenVPN deployment. It discusses the possibilities of the laboratory exercise design mainly from the technical and didactic point of view. Considering them, a concept of the exercise with one OpenVPN server and one client is presented. The exercise simulates a solution of a practical, commonly emerging problem with unwanted Internet access restrictions caused by misconfigured or too restrictive firewalls. The concept comprises both theoretical part and directions, as well as images of the virtual machines used in the exercise. In addition, different possibilities of a further expansion of the exercise or its use as a base for another exercise are mentioned.

KEYWORDS

VPN virtual private network OpenVPN laboratory lab exercise SSL TLS Linux

KORTUS, Jiří *Laboratorní úloha Virtuální sítě typu OpenVPN*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2011. 59 s. Vedoucí práce byl doc. Ing. Karel Burda, CSc.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Laboratorní úloha Virtuální sítě typu OpenVPN“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu práce, panu doc. Ing. Karlu Burdovi, CSc., za konzultaci a cenné rady při psaní bakalářské práce.

Brno ....

(podpis autora)

OBSAH

SEZNAM OBRÁZKŮ

ÚVOD

V současných počítačových sítích se často setkáváme s potřebou propojit vzdálené sítě nebo umožnit přístup do sítě uživatelům, kteří se nacházejí mimo ni. Zároveň klademe požadavky na bezpečnost, nízkou finanční zátěž a další důležité aspekty související s touto problematikou.

V minulosti bylo běžné propojování vzdálených sítí pomocí vyhrazených okruhů (*pevných linek*), což bylo finančně nákladné a nepříliš flexibilní. S postupným rozvojem Internetu se začaly prosazovat virtuální sítě (VPN), které umožňují přístup do vzdálených sítí a jejich propojování, a to na základě různých technologických řešení. Jednotlivá řešení se více či méně liší a mají různé výhody nebo naopak omezení.

Tato práce se zabývá problematikou virtuálních sítí založených na technologii OpenVPN a zaměřuje se na její hlavní rysy a možnosti využití v laboratorním cvičení. V úvodu rozebírá vlastnosti sítí VPN a nejpoužívanější rodiny protokolů, resp. mechanismů, které bývají k realizaci virtuálních privátních sítí používány. U každé technologie jsou kromě charakteristiky rovněž stručně zmíněny výhody a nevýhody.

Následuje popis technologie OpenVPN se zaměřením na filozofii, na které je postavena, na bezpečnost a mechanismy, které k jejímu dosažení používá. Popsány jsou i dostupné metody autentizace a šifrování a základní principy, kterých využívají, stejně jako fungování OpenVPN z hlediska vlastního přenosu dat.

Dále je popsáno, jak lze OpenVPN konfigurovat (konfigurační soubory, grafická uživatelská rozhraní). Mezi popisované nástroje pro konfiguraci a správu OpenVPN je zahrnuta i rozšířená verze OpenVPN – *OpenVPN Access Server*, která nabízí komplexní prostředí pro rychlé nasazení OpenVPN.

Další část práce se věnuje návrhu laboratorní úlohy, jejímž cílem je nastavení jednoduché virtuální sítě založené na OpenVPN. Popisuje podmínky, které je nutné vzít při návrhu úlohy v úvahu, dále podmínky a zadání pro realizaci úlohy. Diskutuje možnosti obsahu úlohy a taktéž jsou uvedeny uvažované didaktické aspekty, jimiž se zadání práce řídí.

Na základě podmínek zadání a jejich analýzy je popsána vlastní realizace úlohy, a to jak z hlediska implementačního (konfigurace virtuálních strojů, na nichž je úloha postavena), tak z hlediska vlastního zadání úlohy. Zadání úlohy je dále rozšířeno o informace pro cvičící.

1 VPN

VPN (Virtual Private Network, virtuální privátní síť) je síť využívající zejména veřejnou telekomunikační infrastrukturu, jako například Internet, k zajištění přístupu vzdálených poboček nebo cestujících uživatelů k centrální síti organizace.[1]

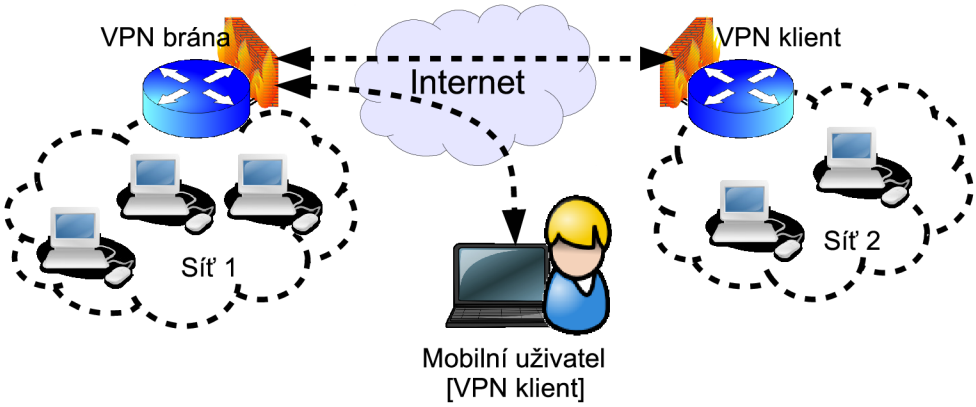
VPN tedy vytváří propojení jednotlivých sítí nebo koncových zařízení prostřednictvím komunikačního kanálu nenáležícího do těchto sítí (zejm. Internetu) tak, že se jeví, jako by byly propojeny přímo.

Historicky byly principy VPN realizovány za pomoci pronajímaných datových okruhů. Takové řešení však bylo finančně nákladné a neflexibilní. S rozvojem Internetu, tedy zejména se stále se rozšiřující dostupností, vyššími dostupnými přenosovými rychlostmi a klesajícími cenami za připojení, začal být k budování VPN používán Internet (nikoliv již vyhrazené datové okruhy)[1]. Dále bude pod termínem VPN označován pouze tento typ propojení.

V praxi jsou VPN využívány zejména k zabezpečenému propojení sítí jednotlivých poboček organizací nebo ke vzdálenému přístupu pracovníků do těchto sítí. Princip vzdáleného připojení do sítě a propojení sítí znázorňuje obr. 1.1, přerušovaná čára reprezentuje propojení pomocí VPN. Další možností využití VPN je vytvoření tunelu sloužícího jako plnohodnotné připojení do Internetu v prostředí, kde jsou nastaveny restrikce na využívání některých služeb (např. povolení jen určitých komunikačních portů na hotspotech apod.)[2].

1.1 Vlastnosti VPN

Výše uvedenou definici VPN lze rozšířit o další vlastnosti, které jsou od virtuálních privátních sítí typicky očekávány. Jedná se především o zajištění autentizace, autorizace, důvěrnosti a integrity:



Obr. 1.1: Princip virtuálních privátních sítí

Autentizace – ověření identity komunikujících stran (VPN serveru, VPN klienta), prokázání totožnosti. Obecně může autentizace být realizována např. uživatelským jménem a heslem, sdíleným klíčem, certifikátem či jinými prostředky.

Autorizace – umožnění přístupu do sítě VPN nebo k dílčím službám poskytovaným v této síti jen patřičným subjektům.

Důvěrnost – zajištění, že přenášená data nemohou být (v čitelné formě) získána třetí stranou. Důvěrnost je ve VPN zajištěna šifrováním komunikace.

Integrita – schopnost zjistit, že během přenosu nedošlo k jakékoliv změně přenášených dat, ať už chybou přenosu nebo jejich úmyslným pozměněním útočníkem.

Výše uvedené principy jsou důležité při realizaci VPN v prostředí, které z povahy nelze považovat za bezpečné (Internet), tedy v naprosté většině případů. Způsoby, jakými jsou jednotlivé body bezpečnosti zajištěny, se liší podle konkrétního typu VPN a jejího nastavení. Mohou být dále rozšiřovány, stejně tak ale mohou být některé v praxi vypuštěny, v závislosti na konkrétní aplikaci a daných požadavcích.

2 TUNELOVACÍ MECHANISMY

Existuje mnoho typů virtuálních privátních sítí s různou úrovní zabezpečení, využívající různé (zejména tunelovací) protokoly. Tunelovací protokol slouží k vytvoření *tunelu*, tedy virtuálního dvojbodového spojení mezi uzly v síti na linkové nebo síťové vrstvě. Tunel je schematicky znázorněn na obr. 2.1, tunel sloužící k propojení první a druhé sítě je znázorněn spojitou čárou, tok dat tunelem čárou přerušovanou.

Jedním či více takovými spojeními lze realizovat koncept virtuálních privátních sítí. K tunelování lze použít protokoly přímo navržené za tímto účelem (např. PPTP, L2TP apod.) nebo i jiné k tomuto účelu vhodné protokoly, pomocí kterých se následně komunikace zapouzdří (např. SSL). VPN (dle použitých protokolů) lze zařadit zejména do následujících skupin:

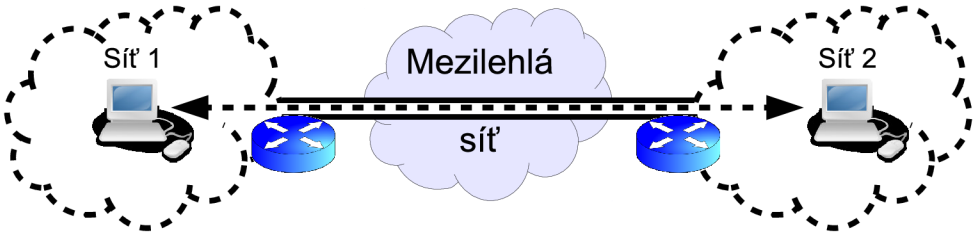
2.1 PPTP

PPTP (Point-to-Point Tunneling Protocol) je tunelovací protokol vyvinutý společností Microsoft ve spolupráci s dalšími firmami [3] dokumentovaný v doporučení RFC 2637. Je postaven nad protokolem GRE, do kterého zapouzdřuje pakety PPP, komunikace je řízena spojením přes protokol TCP. Nejběžnější implementací PPTP je klient/server společnosti Microsoft (od verze Windows 95 OSR2). Podporuje autentizaci protokoly PAP, CHAP, MSCHAPv1/v2 nebo EAP-TLS, data přenášená přes PPP jsou šifrována protokolem MPPE.

Nevýhodou je nízká bezpečnost (zranitelnost protokolů MSCHAPv1/v2) [3], naopak výhodou je nativní dostupnost v operačních systémech Microsoft Windows. K dispozici jsou i jiné implementace PPTP, např. PoPToP pro Linux.

2.2 L2TP

L2TP (Layer 2 Tunneling Protocol)[4] je tunelovací protokol vyvinutý za spolupráce firem Microsoft a Cisco, je popsán v doporučení RFC 2661. Slouží k tunelování dat



Obr. 2.1: Schematické znázornění síťového tunelu

na linkové vrstvě v IP sítích, k přenosu dat používá UDP datagramy. Protokol L2TP sám o sobě umožňuje pouze základní autentizaci a nemá mechanismy pro šifrování přenášených dat. V případě, že je zapotřebí silného autentizačního mechanismu nebo zabezpečení přenosu, lze použít L2TP v kombinaci s protokolem IPsec; mezi uzly je nejprve vytvořeno zabezpečené IPsec spojení, v němž se poté vytvoří L2TP tunel.

Protokol L2TP je mj. podporován v operačních systémech MS Windows (nativně Windows 2000 a vyšší), Apple Mac OS X (verze 10.3 a vyšší) a v unixových operačních systémech.

2.3 IPsec

IPsec je bezpečnostní rozšíření IP protokolu [5], umožňující autentizaci a šifrování každého přenášeného paketu. IPsec může být pro tunelování použit v kombinaci s L2TP (jak bylo zmíněno výše), kdy pracuje v *transportním režimu* a jsou šifrována jen přenášená data (a/nebo je zajištěna jejich autentizace), nikoliv zdrojová/cílová adresa [5, Transport mode]. V tomto případě může být problematická přítomnost NAT v přenosové cestě.

Druhou možností je použití samotného protokolu IPsec v *tunelovacím režimu*. Zašifrován (a/nebo autentizován) je celý paket a následně je zapouzdřen do nového paketu s novou hlavičkou [5, Tunnel mode].

IPsec je nativně podporován v operačních systémech MS Windows (Windows 2000 a vyšší), pod linuxem může být použita podpora v jádře nebo podpora za pomoci dalších implementací mimo jádro, dále jsou podporovány *BSD systémy (mj. implementací v projektu KAME) a další OS včetně OS pro síťový hardware, např. Cisco IOS, Juniper Junos aj.

2.4 SSL/TLS

Protokol SSL (Secure Socket Layer) a TLS (Transport Layer Security) [6], který z SSL vychází, slouží k zajištění bezpečné komunikace přes Internet. K tomuto účelu využívají asymetrickou kryptografii (pro výměnu klíčů), symetrickou kryptografii (pro šifrování přenášených dat) a otisky zpráv (MAC, Message Authentication Code) pro zajištění integrity přenášených dat. SSL umožňuje jednostrannou nebo oboustrannou autentizaci pomocí certifikátů.

Protokol SSL byl navržen společností Netscape, první verze (2.0) byla veřejně vydána v roce 1995, následně byla v roce 1996 vydána verze 3.0 opravující řadu bezpečnostních nedostatků [6]. Protokol TLS byl v roce 1999 ve verzi 1.0 standardizován jako RFC 2246, v současnosti je poslední verzí TLS 1.2 (RFC 5246). Pojmy

TLS a *SSL* defacto označují stejný protokol v odlišné verzi (TLS 1.0 odpovídá označení SSL 3.1, TLS 1.1 odpovídá SSL 3.2 a TLS 1.2 je ekvivalent SSL 3.3), v tomto kontextu lze pojmy označit za volně zaměnitelné.

Protokoly pracují na aplikační vrstvě (z pohledu modelu TCP/IP), resp. relační vrstvě (z pohledu modelu ISO/OSI), a jsou typicky používány k zabezpečení aplikačních protokolů, kde se komunikace původně nezabezpečeným protokolem (např. HTTP, SMTP, IMAP) zabezpečí pomocí SSL/TLS (HTTPS, SMTPS, IMAPS). Kromě komerčních implementací jsou k dispozici i svobodné knihovny pro podporu protokolu SSL, např. OpenSSL, GnuTLS nebo PolarSSL.

2.4.1 SSTP

Realizaci VPN sítě (resp. tunelů) založenou na TLS umožňuje protokol SSTP (Secure Socket Tunneling Protocol)[7], kdy se skrz SSL 3.0 kanál vytvoří spojení protokolem PPP nebo L2TP. Byl vyvinut společností Microsoft se záměrem odstranit některé nedostatky protokolů PPTP nebo L2TP [8], zejména možnost jednoduchého blokování těchto protokolů na firewallech. Použití SSL jednak zajišťuje autentizaci a zabezpečení přenášených dat a jednak dovoluje lepší průchodnost VPN spojení přes firewally, pokud neblokují provoz šifrovaný SSL (standardně na portu 443).

Mimo výhody plynoucí z použití SSL lze zmínit nativní podporu v MS Windows Vista SP1 a vyšších (dále je protokol podporován systémy RouterOS a částečně pod Linuxem). Nevýhodou je použití protokolu TCP pro přenos dat VPN spojení (je vysvětleno dále) a orientace pouze na vzdálený přístup jednotlivých klientů do privátní sítě (nikoliv propojení vzdálených sítí – *site-to-site*).

2.4.2 OpenVPN

OpenVPN je významnou VPN aplikací, která je založena na protokolu SSL. Jedná se o svobodný software pod licencí GPL[9], s jehož vývojem začal v roce 2002 James Yonan a v současnosti je vyvíjen společností OpenVPN Technologies Inc. a opensource softwarovou komunitou (nejnovější stabilní verze je v době psaní tohoto textu – listopad 2011 – 2.2.1). Mimo opensource verze existuje navíc verze dostupná komerčně (OpenVPN Access Server [10]), která rozšiřuje možnosti OpenVPN a zjednodušuje správu zejména v instalacích s velkým počtem klientů.

Díky otevřené licenci může být OpenVPN bezpečnější (kdokoliv může najít a opravit bezpečnostní chybu ve zdrojových kódech) a může být používána zdarma. Funkcionalitu protokolu SSL zajišťuje knihovna *OpenSSL* (dostupná pod svobodnou licencí Apache)[11], jedná se o jednu z nejpoužívanějších svobodných kryptografických knihoven.

Mimo bezpečnost danou otevřenou licencí se OpenVPN snaží zajistit bezpečnost co nejvhodnějším návrhem architektury, zejména modulárním návrhem – např. využitím rozšířené SSL knihovny OpenSSL (nikoliv vlastní reimplementací protokolu SSL) nebo univerzálního ovladače pro vytvoření síťových rozhraní TUN/TAP. Dále využívá bezpečnostních mechanismů operačního systému, jako je ochrana přenášených dat a klíčů v paměti před uložením do odkládacího prostoru (na disk), možnost uzamčení do podstromu souborového systému (*chroot*), běh procesu pod uživatelem bez administrátorských práv. Implementace je provedena v uživatelském prostoru (*userspace*), díky tomu není zapotřebí při instalaci OpenVPN nebo povýšení na novou verzi zasahovat do jádra operačního systému. OpenVPN je multiplatformní software, který lze provozovat na unixových OS (Linux, *BSD, Solaris, MacOS X, aj.), MS Windows, ale i na dalších platformách včetně mobilních zařízení.

Možnou nevýhodou je především nutnost instalovat aplikaci OpenVPN, protože protokol OpenVPN nemusí být v operačním systému nativně podporován.

2.5 Topologie VPN

V souvislosti s VPN se někdy hovoří o typických způsobech použití z pohledu druhů entit (klient, server, peer), které jsou pomocí VPN propojeny.

Propojení více sítí, site-to-site. Pomocí VPN je mezi sebou spojeno více sítí (VPN spojení je realizováno mezi koncovými body, které dále na spojové nebo síťové vrstvě propojují sítě, k nimž náleží). Tato topologie může být využita např. pro propojení jednotlivých poboček firmy nebo jiné organizace.

Připojení více klientů do sítě. VPN spojení je realizováno mezi jednotlivými klienty a sítí, do které pomocí VPN přistupují. Topologie vhodná např. pro přístup zaměstnanců do firemní sítě, pokud pracují doma nebo když cestují. Pokud je VPN server, ke kterému se klient připojuje, ve směrovací tabulce klienta nastaven jako výchozí brána, je přes něj od klienta směrována veškerá komunikace. Toho lze s výhodou využít, pokud je klient připojen k Internetu v restriktivním prostředí (např. omezení typů služeb na firewallu); v takovém případě (pokud není blokován SSL provoz) může pak klient přistupovat do Internetu neomezeně (respektive s omezeními danými připojením do Internetu od VPN serveru). Kompletní směrování provozu přes VPN server je rovněž možné použít k lepšímu prosazování bezpečnostních politik.

Dvoubodové propojení, Point-to-Point (PtP). Může se jednat o VPN spojení mezi dvěma koncovými uzly nebo mezi dvěma uzly, prostřednictvím kterých jsou připojeny k nim přilehlé sítě (site-to-site propojení dvou sítí), v podstatě se jedná o jediný (zabezpečený) tunel. Nejjednodušší topologie vhodná pro potřeby propojení dvou koncových uzlů nebo dvou sítí (např. centrály firmy a pobočky, pokud nemá

poboček více).

Uvedené topologie nejsou vyčerpávající a mohou být (v závislosti na použitých technologiích) různě kombinovány.

3 OPENVPN

3.1 Filozofie/architektura

Jednoduchost návrhu. Filozofie OpenVPN je založena na co nejjednodušším přístupu s cílem vyhnout se přílišné komplexnosti (zde bývá jako protiklad zmiňován přístup IPsec)[12, kap. 3], protože složitý návrh může vést k nepřehlednosti, a tím být náchylnější na výskyt chyb. Jednoduchost návrhu rovněž usnadňuje pochopení, jak OpenVPN funguje, čímž snižuje riziko chyb v konfiguraci.

Modulární architektura. Dalším rysem je modulární architektura, tedy jednak možnost rozšiřovat funkce například pomocí dynamicky linkovaných knihoven (*DLL*) a jednak použití ověřených prostředků k vykonávání některých funkcí (zde se jedná zejména o knihovnu OpenSSL a ovladač virtuálních síťových rozhraní TUN/TAP).

Rozšiřitelnost, možnosti konfigurace. Funkcionalitu OpenVPN lze rozšířit jednak výše zmíněnými knihovnami DLL, další možností je využití skriptů v různých místech konfigurace. Pomocí skriptů je možné například řídit autentizaci nebo upravit nastavení firewallu. Dále lze některé konfigurační parametry upravovat specificky pro různé klienty.

3.2 Bezpečnost

OpenVPN je vyvíjena s cílem snadné konfigurovatelnosti, které ale není dosaženo na úkor bezpečnosti. Dále jsou zmíněny jednotlivé oblasti, kterými je v OpenVPN řešeno zabezpečení.

3.2.1 Komunikace

Protokol zabezpečení komunikace (SSL/TLS) byl (mimo jiné výhody) vybrán proto, že se jedná o ověřený protokol, který lze pokládat za průmyslový standard ve své oblasti [12, kap. 1]. Funkcionalita SSL/TLS je zajištěna knihovnou OpenSSL [1].

OpenSSL[11] je knihovna pod otevřenou licencí a zaměřuje se na implementaci funkcí definovaných protokoly SSL 2, 3 a TLS 1.0 (od verze 1.0.1 částečně podporuje TLS 1.2) a obecnou kryptografickou podporu. OpenSSL poskytuje tyto funkce:

- symetrické šifrování (algoritmy AES, Blowfish, 3DES a další)

[1] Od verze 2.3 je plánována podpora alternativní SSL/TLS knihovny [9] – PolarSSL. Díky této změně bude snazší provozovat OpenVPN na zařízeních s nízkou kapacitou operační paměti, resp. vnější paměti (typicky jednotky MB paměti flash) a nízkým výpočetním výkonem, jako jsou například domácí směrovače.

- hashování (algoritmy MD5, SHA1, SHA2 aj.)
- podpora kryptografie s veřejnými klíči (*PKI, Public Key Infrastructure*) – algoritmy RSA, DSA, algoritmus pro výměnu klíčů Diffie-Hellman, kryptografie založená na eliptických křivkách a práce s certifikáty

Komunikace je v OpenVPN ve výchozím stavu šifrována, šifrování se provádí pomocí symetrické šifry, jejíž typ a délku klíče lze nastavit. Symetrické šifrování je použito z výkonnostních důvodů (šifrování asymetrickou kryptografií je podstatně náročnější na výpočetní výkon).

První možností je šifrovat statickým klíčem – klíč je dopředu vygenerován a bezpečnou cestou přenesen na obě strany OpenVPN (klient a server, resp. na oba peery) a následně používán k šifrování při jakémkoliv přenosu dat přes OpenVPN. Zjevnou výhodou je nízká pracnost při zprovozňování takového řešení (není třeba řešit práci s certifikáty), oproti použití certifikátů však statický klíč nabízí nižší míru bezpečnosti. Pokud je klíč kompromitován, je nutné vygenerovat nový klíč a ten přenést na server i všechny klienty (příp. oba peery), což činí použití statického klíče nevhodné pro konfigurace s velkým počtem klientů. Dalším nedostatkem je, že útočník může dešifrovat veškerou komunikaci zachycenou před zjištěním klíče i komunikaci následnou.

Při použití PKI dochází k autentizaci komunikujích stran a v případě, že proběhne úspěšně, je ustanoven klíč pro symetrické šifrování, kterým jsou dále zabezpečena přenášená data. Pro zvýšení bezpečnosti je tento symetrický klíč periodicky měněn, čímž se zmenšuje množství dat, která může potenciální útočník analyzovat a v případě nalezení klíče dešifrovat (dešifrovat lze pouze data přenesená za dobu používání jednoho klíče). Rovněž je znemožněno dešifrování a změna (podvržení) dat přenesených po změně klíče. Změna klíče probíhá většinou po jedné hodině.

S udržováním PKI je spojena určitá režie (generování certifikátů nebo podepisování žádostí o vydání certifikátu, distribuce, revokace), ale poskytuje vyšší bezpečnost oproti statickému klíči. V situaci, kdy je kompromitován soukromý klíč některé z komunikujících stran stačí provést revokaci (zrušení platnosti) pouze certifikátu, ke kterému patří tento klíč, vygenerovat nový klíč a certifikát a přenést jej na patřičné místo.

Zajištění integrity a autenticity dat je realizováno otiskem zprávy (*MAC, Message Authentication Code*). MAC je hash dat přenášených v daném paketu, v kombinaci s ustanoveným klíčem pro šifrování dat. Odesílatel tedy použije data určená k přenosu a před ně připojí šifrovací klíč a vyrobí hash tohoto celku, čímž vznikne MAC. Příjemce Dešifruje přijatá data, přidá k nim šifrovací klíč, z tohoto klíče a přijatých dat opět udělá hash (tedy MAC zprávy) a ten porovná s MAC přijatým. Pokud jsou shodné, data nebyla po cestě poškozena ani jinak upravena.

Ještě vyšší stupeň bezpečnosti v tomto ohledu nabízí zabezpečení pomocí HMAC

(*Hash Message Authentication Code*)[13]. Princip zabezpečení je stejný jako u MAC, ale je použit oddělený šifrovací klíč, který musí existovat již před navázáním SSL relace (narozdíl od klíče ustanoveného při realizaci SSL relace). Použití HMAC dává možnost autentizace všech paketů, které OpenVPN přijme. Tato autentizace navíc může být provedena ještě před předáním paketu k dalšímu zpracování SSL knihovnou (čímž se eliminují útoky, které by mohly využívat případné bezpečnostní díry v této knihovně). Další výhodou je zvýšení odolnosti proti DoS útokům (*Denial of Service*, odepření služby) a proti skenování portů, které má za cíl zjistit, jaké UDP porty jsou otevřené[14].

3.3 Autentizace

OpenVPN nabízí poměrně široké možnosti autentizace klientů do sítě VPN. Níže uvedené metody autentizace mohou být kombinovány a mohou tak zajistit ještě vyšší bezpečnost. Kombinace více metod (například použití autentizace certifikátem a uživatelským jménem/heslem) se nazývá *vícefaktorová autentizace*.

3.3.1 Autentizace založená na PKI

Infrastruktura veřejných klíčů – PKI – je založena na asymetrické kryptografii a na důvěryhodné třetí straně (certifikační autoritě, CA). V asymetrické kryptografii je šifrování prováděno (v závislosti na tom, kdo zprávu šifruje a komu je určena) soukromým a veřejným klíčem. Soukromý klíč musí být uchováván na bezpečném místě, zatímco veřejný klíč může být libovolně vystavován (resp. jeho všeobecná známost je žádoucí).

Pokud chce komunikující strana A poslat zašifrovanou zprávu straně B (tak, aby jen strana B ji byla schopná dešifrovat), zašifruje zprávu veřejným klíčem strany B a této straně zprávu odešle. Strana B pak je schopna zprávu dešifrovat použitím svého soukromého klíče. Pokud je proces a dešifrování proveden v opačném sledu (tedy zpráva je zašifrována soukromým klíčem strany B a následně dešifrována veřejným klíčem B), lze jím (v určité míře, pokud lze věřit spojení veřejného klíče B se stranou B) provést ověření, že zpráva skutečně pochází od strany B.

Vyšší míry autentizace lze dosáhnout použitím certifikátů a důvěryhodné třetí strany, tedy certifikační autority. V principu je certifikát množina informací o subjektu, jemuž certifikát náleží (např. jméno organizace, na jejíž jméno byl certifikát vystaven, sídlo organizace, organizační složka apod.) + veřejný klíč tohoto subjektu, a tyto informace jsou podepsány (je zašifrován jejich hash) soukromým klíčem certifikační autority (jejíž veřejný klíč je dobře znám). Kdokoliv si může za pomoci veřejného klíče CA ověřit, že certifikát byl podepsán CA (v případě, že by byl změněn,

by ověření podpisu selhalo). Je-li CA důvěryhodná organizace, pak lze i usuzovat, že data obsažená v certifikátu jsou pravdivá, a tedy že certifikát (a veřejný klíč) patří danému subjektu.

V praxi se pro některé účely (např. zabezpečené přenosy protokolem HTTPS) používají certifikáty vydané za poplatek akreditovanou CA, jejíž kořenový certifikát (a tudíž i veřejný klíč) je součástí softwaru podporujícího SSL/TLS (např. webového prohlížeče). Použití certifikátu vydaného akreditovanou CA je vhodné v případě, že ke službě, která se certifikátem prokazuje, přistupuje předem neurčený okruh uživatelů (u kterého lze dopředu předpokládat pouze znalost certifikátů akreditovaných certifikačních autorit).

Pokud lze okruh přistupujících uživatelů stanovit předem (nebo je akceptovatelné riziko, že informace v certifikátu nemusí být pravdivé), je možné založit si vlastní CA a certifikáty vydávat s její pomocí, přičemž CA si kořenový certifikát podepíše sama. Takové certifikáty se nazývají *self-signed* a např. v OpenVPN jsou běžně používány. Při použití self-signed certifikátu je ale zapotřebí odsouhlasit (ze strany uživatele) použití certifikátu, jehož CA není akreditovaná, případně si tento certifikát (nebo kořenový certifikát CA) uložit a tím jej považovat za důvěryhodný.

Při ztrátě nebo kompromitaci soukromého klíče může jeho vlastník kontaktovat CA, která zařídí zneplatnění přidruženého certifikátu (jeho revokaci). Zneplatnění provede tak, že přidá informace o tomto certifikátu do seznamu *CRL* (*Certificate Revocation List*). CRL je dostupný veřejně (v případě akreditovaných CA), v případě použití v OpenVPN postačí, aby byl dostupný OpenVPN serveru, který na jeho základě může odmítnout autentizaci klientů prokazujících se certifikátem, který je na CRL.

OpenVPN nabízí jako jednu z možností autentizaci pomocí certifikátů[14]. Na serveru a každém z klientů musí být pro použití tohoto způsobu autentizace přítomen kořenový certifikát CA, a soukromý klíč a certifikát náležící danému serveru/klientovi. Soukromý klíč může být chráněn heslem (zašifrován), aby byl chráněn před neautorizovaným použitím. V tom případě je nutné před použitím klíče zadat heslo k jeho dešifrování. Z tohoto důvodu (nutnosti interakce s uživatelem) není vhodné soukromé klíče chráněné heslem používat na serverech nebo obecně u služeb, kde je předpokládáno automatické použití klíče bez uživatelské interakce.

3.3.2 Autentizace pomocí jména a hesla

Další z možností autentizace klientů je použití uživatelského jména a hesla. Údaje poskytnuté klientem jsou poté OpenVPN serverem (volitelně) použity jedním z následujících způsobů:

- Mohou být pomocí architektury *PAM* (*Pluggable Authentication Module*) předány PAM modulu, který zajistí autentizaci oproti databázi uživatelských údajů (v případě unixových systémů může být použit například lokální soubor s uživatelskými jmény a hesly `/etc/shadow`) nebo možnosti autentizace dále dle nastavení upraví za pomoci dostupných PAM modulů.
- Mohou být předány k autentizaci přes adresářovou službu *LDAP* (*Lightweight Directory Access Protocol*) včetně *AD* (*Active Directory*) nebo přes *RADIUS* (*Remote Authentication Dial In User Service*). Autentizace pomocí uvedených služeb je možná pouze při použití modulů třetích stran nebo produktu OpenVPN Access Server.
- Při autentizaci může být spuštěn skript, na jehož standardní vstup jsou poskytnuty autentizační údaje. Skript na jejich základě rozhodne, zda autentizace proběhne úspěšně či nikoliv.

3.3.3 Autentizace čipovou kartou

K autentizaci pro OpenVPN spojení je možné použít čipovou kartu (*smartcard*). Přístup ke kartě je založen na platformně nezávislém standardu pro správu PKI – *PKCS#11* (*Public-Key Cryptography Standard #11*) [15].

3.4 Šifrování

OpenVPN šifruje přenášená data symetrickou šifrou, ve výchozím nastavení se jedná o blokovou šifru Blowfish s délkou klíče 128 bitů. Může být nastavena jakákoliv jiná symetrická šifra (a délka klíče v rozsahu podporovaném danou šifrou), kterou nabízí knihovna OpenSSL.

3.5 Systém

OpenVPN používá další bezpečnostní mechanismy i na úrovni operačního systému:

- Dovoluje nastavit běh pod neprivilegovaným uživatelem (pod unixovými systémy většinou uživatel *nobody*). OpenVPN po spuštění nejdříve provede inicializaci, ke které jsou potřeba oprávnění superuživatele a následně, kdy již pro vlastní běh OpenVPN tato práva nejsou třeba, změní vlastníka procesu na uživatele s omezenými právy. Pokud by došlo k úspěšnému útoku na proces OpenVPN, získá útočník v systému pouze práva neprivilegovaného uživatele, nikoliv superuživatele.
- Zabraňuje uložení přenášených dat a šifrovacích klíčů mimo operační paměť (odložení na disk, *swap*).

- Nabízí možnost uzamčení do podstromu souborového systému nastavením určeného adresáře jako kořene systému (*chroot*). V případě, že útočník získá kontrolu nad OpenVPN procesem, získá přístup pouze do daného podstromu souborového systému a nemůže tak získat nebo modifikovat data mimo něj.

3.6 Síť

3.6.1 Princip přenosu dat

OpenVPN implicitně pro přenos dat využívá protokol UDP, OpenVPN server naslouchá na portu 1194. V případě potřeby lze nastavit použití protokolu TCP, pokud pro to ale nejsou nutné důvody (např. blokování UDP přenosů na daném portu firewallem), je výhodnější použít protokol UDP. OpenVPN totiž realizuje z end-to-end pohledu logický spoj na druhé nebo třetí vrstvě ISO/OSI modelu, kde není předpokládáno zajištění spolehlivosti přenesení dat či uspořádání pořadí paketů. Tyto záležitosti řeší až transportní vrstva, popřípadě vrstvy vyšší. Použití UDP pro vytvoření nespolehlivé přenosové cesty je tedy z pohledu dat přenášených SSL tunelem v pořádku (pokud je tunelem přenášeno TCP spojení, zajistí si spolehlivost samo, pokud UDP spojení, spolehlivost není požadována nebo ji řeší vyšší vrstvy).

Jsou-li data SSL kanálu přenášena protokolem TCP, může při přenosu dat protokolem TCP skrz tento kanál dojít k podstatnému propadu rychlosti přenosu dat (tzv. *TCP meltdown*) [16]. Tento jev je způsoben mechanismem protokolu TCP zajišťujícím spolehlivé doručení, který nepředpokládá, že jsou další takové mechanismy použity na nižších vrstvách. Pakety jsou sekvenčně číslovány a příjemcem je potvrzováno jejich přijetí. Pokud po určenou dobu nedojde odesílateli potvrzení, že daný paket byl doručen, předpokládá jeho ztrátu vlivem zahlcení linky a paket zařadí do fronty k novému odeslání. Před tím ale počká určitou dobu (stanovenou časovačem), aby nedocházelo k dalšímu zahlcení linky. V případě, že se ztráta paketu opakuje, navyšuje dobu časovače (navyšování má exponenciální průběh). Pokud nastane výpadek, reaguje protokol TCP u spojení pro přenos dat SSL kanálu navýšením čekací doby před odesláním a zařazením paketu k opětovnému odeslání. V případě, že hodnota čekací doby u TCP spojení přenášeného uvnitř kanálu je nižší, nutně následuje nedoručení paketu, znovuzařazení do fronty k odeslání a navýšení hodnoty časovače. Tím dojde k zařazení více požadavků k opětovnému přenosu paketů TCP spojení uvnitř SSL kanálu, než jaké je schopné zpracovat vlastní TCP spojení SSL kanálu, a to může mít za následek celkové zpomalení přenosu nebo pády spojení.

OpenVPN spojení lze realizovat i přes HTTP nebo SOCKS proxy server. Při nastavení přenosu dat protokolem TCP nabízí OpenVPN možnost sdílení portu s jinou aplikací (podporuje protokoly HTTP a HTTPS). Pokud je potřeba, aby OpenVPN

server naslouchal na určitém portu (například 80 nebo 443) – typicky z důvodů filtrování provozu na jiných portech – a zároveň je nutné na stejném portu provozovat jinou aplikaci (např. webový server), lze tento problém vyřešit sdílením portů. Aplikaci, která port sdílí, je třeba nastavit tak, aby naslouchala na jiné IP adrese nebo na jiném portu a na původním portu pak naslouchá OpenVPN. Při příchozím požadavku na spojení vyhodnotí obsah paketu a pokud nedetekuje data OpenVPN spojení, předá tento paket aplikaci, s níž port sdílí.

V případě potřeby je v OpenVPN možné použít omezovač rychlosti (*shaper*), a nastavit jím maximální rychlost přenosu dat přes VPN spojení. Omezení rychlosti lze provést vždy pouze jedním směrem, pokud je třeba omezit rychlost v obou směrech, je nutné nastavit omezení pro každou stranu VPN spojení zvlášť.

3.6.2 Komprese přenášených dat

Data přenášená VPN spojením mohou být komprimována algoritmem *LZO* (označení je odvozeno od jmen jeho tvůrců – *Lempel, Ziv, Oberhumer*). OpenVPN zajišťuje (pokud takové chování není v konfiguraci vypnuto), že je komprese adaptivní. To znamená, že OpenVPN periodicky analyzuje vzorky přenášených dat a v případě, že by se v danou chvíli komprese nevyplatila (při přenosu nezkomprimovatelných nebo již zkomprimovaných dat), kompresi dočasně vypíná (dokud na základě dalšího vzorku dat nevyhodnotí, že kompresi se opět vyplatí použít).

3.6.3 Rozhraní TUN, TAP

Zakončení OpenVPN tunelů jsou řešena prostřednictvím univerzálních TUN, resp. TAP rozhraní. Jedná se o virtuální síťová rozhraní napojená na proces v uživatelském prostoru – data zapsaná aplikacemi na virtuální rozhraní jsou předána procesu (v tomto případě OpenVPN) a stejně tak data od tohoto procesu mohou být přečtena aplikacemi přes virtuální rozhraní. OpenVPN tak v principu (z pohledu přenosu dat) zajišťuje slučování řídicích dat kanálu a vlastních dat přenášených kanálem (užitečné zátěže), přičemž tato sloučená data jsou před odesláním šifrována a po přijetí dešifrována a oddělena.

Virtuální síťová rozhraní mohou být dvojího typu – *TUN* a *TAP*. Název rozhraní TUN vypovídá, že se jedná o zkratku slova *tunnel* a charakterizuje realizaci tunelu na 3. (síťové vrstvě), lze jím tedy přenášet libovolnou IP komunikaci. Využití rozhraní TUN implikuje použití směrované topologie, tedy vytvoření jedné či více podsítí s vlastním adresním rozsahem a směrování (případně překlad adres – *NAT*, *Network Address Translation*) mezi ním a zbylou částí sítě, do které se přes VPN přistupuje.

Název rozhraní TAP je odvozen od výrazu *tap* vyjadřujícího přímé (fyzické) napojení na síť (analogie například ke způsobu připojování stanic k Ethernetu 10Base-5). Rozhraní TAP podporuje přenos na 2. (linkové) vrstvě a emuluje vrstvu typu Ethernet. Tak lze realizovat VPN spojení logicky simulující přímé propojení konců VPN tunelu na úrovni Ethernetu.

Běžně je výhodnější použít rozhraní TUN, protože rozdělením na více podsítí jsou zmenšeny *broadcast* domény (části sítě, ve kterých je možné zasílat data všesměrově, tedy všem připojeným stanicím) a tím snížit zatížení VPN spojení. Další výhodou propojení na 3. vrstvě může být lepší rozčlenění připojených stanic z hlediska členění adresního rozsahu nebo restrikce přístupu do různých částí sítě firewallem. Naopak realizace propojení na 2. vrstvě je nezbytná, pokud má být tunelem zajištěna komunikace aplikací, které pro svou funkci používají všesměrové vysílání (sdílení v sítích MS Windows bez WINS nebo Samba serveru, starší hry s podporou hraní po síti apod.)[14].

3.6.4 Adresování klientů

Klientům může OpenVPN přidělovat statické IP adresy (včetně možnosti definovat páry klient-adresa v externím souboru), nebo může přidělovat adresy dle dostupnosti z vyhrazeného rozsahu.

3.6.5 Předávání parametrů klientům

Server může klientům předávat konfigurační parametry a zajistit tak, že VPN spojení bude nastaveno správně. Mezi tyto parametry patří např. informace o směrování (tzn. jaké informace si klient má přidat do směrovací tabulky, aby mohl komunikovat se sítěmi připojenými k VPN), nastavení VPN serveru jako výchozí brány (pak je přes VPN server směrován veškerý provoz od klienta), vypínat/zapínat kompresi aj.

3.6.6 Rozkládání zátěže, redundance

OpenVPN klienty je možné nastavit tak, aby přistupovali k jednomu z množiny definovaných serverů. K serverům mohou přistupovat ve stanoveném pořadí – klient se pokusí připojit k prvnímu serveru, v případě neúspěchu ke druhému a analogicky k dalším serverům (jsou-li definovány), čímž lze vyřešit redundanci a zajistit tak možnost připojení klientů do VPN sítě i v případě výpadku serveru.

Klient může k serverům rovněž přistupovat v náhodném pořadí, což je výhodné při rozkládání zátěže mezi více OpenVPN serverů. Nutnou podmínkou pro správnou funkci sítě OpenVPN je v takovém případě synchronizace konfigurací všech serverů.

3.7 Konfigurace

3.7.1 Konfigurační soubory

Existuje několik způsobů, kterými lze OpenVPN konfigurovat. Základní možností je použití konfiguračních souborů, do kterých se v textové podobě zadají požadované konfigurační parametry. Tyto soubory mohou odkazovat na jiné soubory související s konfigurací, jako jsou seznamy přidělovaných IP adres, individuální konfigurační parametry (pro jednotlivé klienty) nebo např. skripty. Manuální úprava konfiguračních souborů je výhodná v případě, kdy postačí jednou OpenVPN nakonfigurovat a další změny se již nepředpokládají, nebo v případech, kdy je třeba upravit detailní nastavení parametrů, které by jinak nebylo možné provést (například při použití grafického rozhraní).

Konfigurační soubory OpenVPN mají pod Linuxem (jako většina konfiguračních souborů) příponu `.conf` a jsou umístěny v adresáři `/etc/openvpn`. Do tohoto adresáře (nebo podadresářů – pro lepší přehlednost) se umisťují i další soubory související s konfigurací (certifikáty, klíče aj.). Pod OS Windows je cesta hlavního adresáře s konfiguračními soubory `C:\Program Files\OpenVPN\config`. Konfigurační soubory mají v tomto případě příponu `.ovpn` a jsou asociovány pro otevření aplikací OpenVPN [14].

Pro použití konfiguračního souboru je možné jej zadat jako parametr aplikaci OpenVPN při spouštění (případně je toto provedeno spouštěcími skripty pro OpenVPN). Stejné konfigurace je možné dosáhnout i spuštěním OpenVPN klienta/serveru (v unixových systémech označujeme takovou aplikaci, zejména pokud běží na pozadí, *démon*) s jednotlivými parametry předanými přímo této aplikaci při jejím spouštění. Tuto možnost lze využít při testování OpenVPN.

V podstatné míře jsou konfigurační soubory platformně nezávislé, tj. například lze pod OS Windows použít konfigurační soubor vytvořený a používaný pod Linuxem. Může být nutné provést drobné úpravy, především změnu přípony souboru a úpravu zápisu cest obsažených v souboru, také je nezbytné zajistit správný převod konců řádků v konfiguračním souboru (konce řádků jsou zapisovány odlišným způsobem na unixových platformách a pod MS Windows). Převod lze provést například nástrojem `unix2dos`, resp. `dos2unix`.

Mimo tyto platformně specifické úpravy je důležité zmínit existenci voleb (konfiguračních direktiv), které jsou závislé na platformě, a na tuto skutečnost je třeba brát ohled při přenosu konfigurací mezi různými platformami. Příkladem je možnost uzamčení do určeného adresáře (chroot), která je specifická pro unixové systémy.

Kromě vlastního nastavení parametrů může být potřeba (v závislosti na konkrétním nastavení) provést ještě další úkony, které s konfigurací nepřímo souvisejí. Jedná

se především o generování klíčů nebo certifikátů, popř. zneplatnění existujícího certifikátu. Pro tyto účely obsahuje instalace OpenVPN sadu skriptů (`easy-rsa`), které zmíněné úkony činí rychlejšími a uživatelsky přívětivějšími.

3.7.2 Rozhraní pro správu

Správu OpenVPN (jak klienta, tak i serveru) lze provádět přes rozhraní s příkazovou řádkou [14] (*CLI*, *Command-Line Interface*) dostupné přes TCP spojení na adresu a port, kde naslouchá část aplikace poskytující přístup k CLI. Komunikace není šifrovaná a probíhá čistě v textové podobě, k připojení lze použít libovolnou klientskou aplikaci pro přístup ke vzdálenému terminálu (např. běžně rozšířenou aplikaci `telnet`), správa se provádí zadáváním příkazů přes CLI.

Přes rozhraní správy lze vypisovat stav OpenVPN démona (připojené klienty, množství přenesených dat apod.), shromažďovat záznamy a hlášení a odpojovat připojené klienty. Rozhraní je vhodné nejen pro vzdálené (nebo centralizované) ovládání OpenVPN serverů, ale i klientů (je využíváno některými grafickými rozhraními pro správu).

3.7.3 Grafické rozhraní

Konfigurování OpenVPN editací textových souborů nemusí být z uživatelského hlediska přívětivé nebo přehledné, zejména pro uživatele, kteří jsou zvyklí pracovat pouze v grafických prostředích operačního systému. Stejně tak může nastat situace, kdy nemají dostatečné znalosti, aby byli schopni provést konfiguraci editací konfiguračních souborů. Dalším možným hlediskem, proč může být textová konfigurace nevhodná, je požadavek na začlenění do grafického prostředí systému.

V těchto případech je řešením použití grafického rozhraní (*GUI*, *Graphical User Interface*) pro správu a konfiguraci, které je určitou nadstavbou usnadňující jak konfiguraci, tak běžně prováděné úkony s OpenVPN (připojení, odpojení, indikace stavu spojení, správa více připojení a podobně). Grafické rozhraní může být realizováno jako program nebo doplněk ke stávajícím GUI prostředkům pro správu sítě, nebo jako webové rozhraní.

K dispozici je celá řada GUI pro správu OpenVPN (jak klientů, tak i serverů) pro různé platformy, jejich seznam je k dispozici na `https://community.openvpn.net/openvpn/wiki/RelatedProjects`. Jednou z prvních grafických nadstaveb pro OpenVPN klienty pod MS Windows je *OpenVPN GUI* (`http://www.openvpn.se`), které umožňuje základní nastavení připojení k OpenVPN. OpenVPN GUI indikuje stav připojení, spouštění/zastavení/restart aplikace OpenVPN, pokud je nakonfigurována, aby běžela jako služba (obdoba démona pod unixovými systémy). Umožňuje

více připojení naráz (s možností automatického připojení po spuštění GUI), nastavení připojení lze upravovat v textovém editoru. Nabízí dialogy pro zadání hesla při autentizaci vůči OpenVPN serveru nebo pro dešifrování soukromého klíče a další vlastnosti. Od roku 2006 se však OpenVPN GUI nevyvíjí, a proto nelze tento projekt považovat za perspektivní.

Nedostatky OpenVPN GUI řeší projekt *openvpn-gui* (`https://sourceforge.net/projects/openvpn-gui`), a také dále rozšiřuje možnosti nastavení a ovládání OpenVPN klienta pod OS Windows.

Z GUI pro Linux lze jmenovat především doplněk pro podporu OpenVPN ve správci síťových připojení *Gnome Network Manager*, který je výchozím v prostředí Gnome. V prostředí KDE lze různá VPN připojení (včetně OpenVPN) spravovat nástrojem *KVpnc*. Nabízí relativně široké možnosti konfigurace a mj. také průvodce nastavením VPN připojení, který může představovat užitečnou pomůcku pro méně pokročilé uživatele.

4 OPENVPN ACCESS SERVER

OpenVPN Access Server je rozšířenou, placenou[1] verzí OpenVPN vyvíjenou společností OpenVPN Technologies Inc. [10]. Představuje komplexní aplikaci obsahující OpenVPN server a webové rozhraní sloužící jednak ke snadné administraci serveru a jednak k přístupu klientů. Kromě rozhraní pro zrychlení a zjednodušení správy nabízí i další možnosti, které nejsou v komunitní verzi OpenVPN dostupné, nebo jsou dostupné jen za použití doplňků třetích stran. Svým charakterem se tak OpenVPN Access Server zaměřuje na použití ve firmách nebo jiných organizacích, ve kterých je předpokládán přístup většího množství klientů, čemuž je vhodné přizpůsobit systém správy.

Oproti komunitní verzi OpenVPN nabízí navíc Access Server nativní podporu autentizace přes protokol LDAP (včetně *Active Directory*) nebo RADIUS. Je navržen pro zajištění dostupnosti i při výpadku jednoho serveru – umožňuje replikaci konfigurace na sekundární OpenVPN server zabezpečeným kanálem pomocí protokolu *SSH* (*Secure Shell*) a sdílení IP adresy mezi dvěma servery. Každý ze serverů má vlastní IP adresu a navíc je stanovena jedna společná (virtuální), na které je dostupný OpenVPN server. Za normálních okolností pod touto adresou vystupuje primární server a v případě jeho výpadku tuto adresu přebere server sekundární, tato funkcionalita využívá protokolu *UCARP* (*Userland Common Address Redundancy Protocol*).

Access Server nabízí jednak obecné možnosti nastavení OpenVPN serveru (takže není nutné jej konfigurovat manuálně editací konfiguračních souborů) a dále taktéž správu uživatelů (přidávání, odebírání, blokování, rozdělování do skupin apod.). Přístupy uživatelů do VPN sítě a do webového rozhraní jsou zaznamenávány, lze je prohlížet prostřednictvím webového rozhraní.

Z pohledu klienta nabízí Access Server přes webové rozhraní (po přihlášení) ke stažení klientský software pro OS Windows a Mac OS X. Klient pro Linux není nabízen, ale webové rozhraní nabízí odkaz na stránky společnosti, kde je k dispozici návod na instalaci klientského software. Z webového rozhraní lze také stáhnout kompletní vygenerované konfigurační soubory (obsahují jak konfigurační parametry, tak i patřičné certifikáty a soukromý klíč), a to buď specifické pro konkrétního přihlášeného uživatele, nebo obecný konfigurační soubor použitelný všemi uživateli. Tímto způsobem se uživatel může snadno připojit i z počítače, na kterém doposud OpenVPN nepoužíval, má-li k tomu potřebná oprávnění.

[1] OpenVPN je možné ze stránek společnosti stáhnout zdarma, zkušební verze obsahuje licenci pro současné připojení dvou klientů k serveru. Větší počet současně připojených klientů je možný pouze při patřičném rozšíření licence (v současné době je minimum pro zakoupení licence 10 klientů).

5 LABORATORNÍ ÚLOHA

5.1 Výchozí podmínky

Cílem této práce je na základě nastudované problematiky virtuálních sítí typu OpenVPN navrhnout a zrealizovat laboratorní úlohu na konfiguraci virtuální sítě OpenVPN. Úloha bude využita ve cvičeních z předmětu *Návrh, správa a bezpečnost sítí*. Při návrhu úlohy bylo nutné vycházet z následujících podmínek:

- Pro řešení úlohy je k dispozici pracoviště s jedním, nanejvýš dvěma počítači s OS MS Windows XP, které jsou připojeny do sítě.
- Čas na realizaci úlohy je 90 minut.
- U studentů lze předpokládat alespoň částečnou znalost operačního systému Linux, protože jej používají i v jiných úlohách.
- Úloha má být postavena na virtuálních strojích, které poběží ve virtualizačním prostředí VMware.
- Studenti nebudou z úlohy vypracovávat laboratorní zprávu.
- Cvičící nemusí disponovat hlubší znalostí konkrétně dotčené problematiky (je zapotřebí zaměřit se na možné problémy při realizaci úlohy a na to, jakým způsobem kontrolovat správnost provedení úlohy).

První součástí práce je návod k laboratorní úloze (zahrnující teoretický úvod – popis problematiky, možnosti praktického využití a následně zadání pro konfiguraci sítě OpenVPN. Tato část je doplněna informacemi pro cvičící, kde jsou stručně rozebrány možné problémy a jejich řešení, a dále tématické otázky pro studenty (včetně vzorových odpovědí).

Druhou částí práce jsou pak vlastní obrazy virtuálních strojů použitých při cvičeních.

Použití virtuálních strojů je výhodné z následujících důvodů:

- K realizaci úlohy stačí jeden fyzický stroj.
- Obrazy virtuálních strojů je možné kdykoliv v případě potřeby (například při poruše) přenést na jiný počítač.
- V případě poškození nebo znefunkčnění virtuálního stroje jej lze snadno uvést do původního stavu.
- Obrazy mohou být v případě potřeby využity jako základ podobné úlohy nebo dále rozšířeny.

5.2 Předmět laboratorní úlohy

5.2.1 Aplikační scénář

Vzhledem k omezením daným zejména dostupným časem pro vypracování bylo třeba zvolit vlastní předmět úlohy s ohledem na toto omezení, a přitom v zadání obsáhnout důležité rysy sítí OpenVPN. Za tímto účelem se jeví jako vhodné použití jednoduché síťové topologie pracující s OpenVPN, typické pro určitý aplikační scénář. Mezi tyto typické scénáře patří například propojení dvou odlehlých sítí, propojení dvou strojů nebo přístup klienta do sítě organizace. Dále je možné volit mezi různými způsoby šifrování a autentizace (síť bez šifrování, použití statického klíče, použití certifikátů) a případně specifikovat detailní požadavky na nastavení sítě.

OpenVPN nabízí bohaté možnosti konfigurace, při nasazování OpenVPN mimo vlastní konfiguraci s touto problematikou souvisí i další nutné úkony, jako jsou úpravy firewallu (nutné pro vlastní funkci OpenVPN – nesmí být firewallem blokována, umožnění rozkládání zátěže, řešení překladů adres nebo další specifické úpravy). Navržená laboratorní úloha se soustřeďuje na základní nastavení OpenVPN, které je i prakticky (s mírnými úpravami) použitelné, případně bez velkých obtíží upravitelné pro jiný aplikační scénář.

Úloha simuluje situaci, ve které máme k dispozici připojení k Internetu, které je však omezeno firewallem, který propouští pouze provoz na určitých portech, zbylý provoz blokuje. Dále máme k dispozici stroj, přes který lze dále směrovat provoz, a na kterém můžeme provozovat OpenVPN server. Po konfiguraci OpenVPN klienta a OpenVPN serveru a realizaci VPN spojení (úloha uvažuje spojení na síťové vrstvě) tak lze efektivně obejít blokování provozu na firewallu (směrem do internetu), s čímž se v praxi lze setkat.

5.2.2 Potřebné úkony

Na základě zmíněného scénáře, který má úloha realizovat, bude studentem provedena konfigurace OpenVPN serveru a klienta (server běží na jednom virtuálním stroji – označen jako `ovpn-server`, klient na druhém – označen jako `ovpn-klient`). Součástí úlohy je dále ustanovení vlastní certifikační autority a vygenerování klíčů a certifikátů jak pro server, tak pro klienta, které jsou použity pro autentizaci a šifrování komunikace. Konfigurace je prováděna manuální úpravou konfiguračních souborů [1].

[1] Bylo by možné využít webové rozhraní OpenVPN Access Serveru, nicméně při úpravě konfiguračního souboru jsou veškeré parametry pohromadě na jednom místě, a z tohoto důvodu by měla být konfigurace (její podstata) pochopitelnější. Navíc se patrně jedná o řešení, se kterým se v praxi lze častěji setkat.

Při konfiguraci zadání vychází ze vzorových konfiguračních souborů dodávaných s OpenVPN. Výhodou je, že tyto soubory obsahují potřebné konfigurační direktivy (z nichž je v tomto případě využita jen část) a zejména poměrně obsáhlé komentáře (v anglickém jazyce), což zlepšuje orientaci a případně lépe objasňuje význam některých nastavení. Mimo to mohou studenti po splnění úkolů zadaných v laboratorním cvičení (pokud jim zbyl čas) dále zkoumat jiné možnosti konfigurace a jejich vliv na chování sítě.

Shrnutí jednotlivých bodů úlohy je následující:

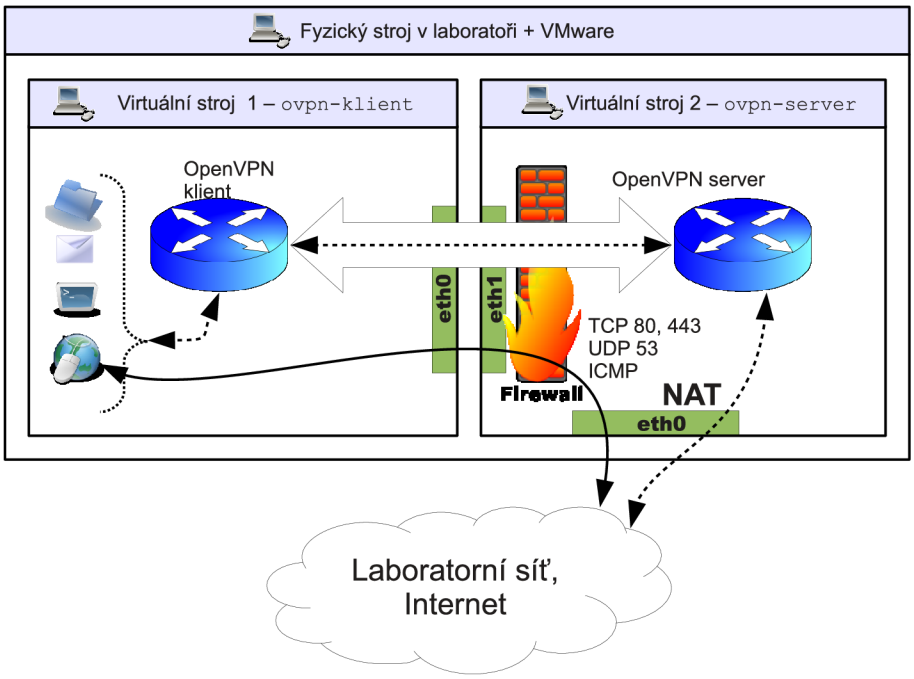
- Prozkoumání výchozího stavu sítě a přítomných omezení
- Založení certifikační autority, vygenerování klíčů a certifikátů
- Nastavení OpenVPN serveru
- Nastavení OpenVPN klienta
- Prozkoumání stavu po realizaci VPN spojení

5.2.3 Topologie

Virtuální stroj `ovpn-klient` obsahuje rozhraní *eth0* (IP adresa 192.168.1.2/24), jehož prostřednictvím je spojen s rozhraním *eth1* stroje `ovpn-server` (IP adresa 192.168.1.1/24). Stroj `ovpn-server` dále obsahuje rozhraní *eth0* (jeho adresa je přidělována dynamicky od virtualizačního nástroje VMWare Player), je nastavena jako výchozí brána a na data směrovaná tímto strojem je aplikován překlad adres. Dále je zde uplatněn firewall omezující provoz do Internetu ze stroje `ovpn-klient` přes rozhraní *eth1* stroje `ovpn-server`.

Topologie virtuální sítě je vyobrazena na obr. 5.1. Souvislá čára mezi aplikacemi na stroji `ovpn-klient` a Internetem charakterizuje výchozí situaci, kdy je komunikace s vnější sítí možná, avšak firewall povolí pouze služby na určitých portech (zde znázorněn zákaz služeb kromě protokolů DNS, NTP, HTTP a HTTPS, které jsou běžně provozovány na UDP portech 53 a 123 a TCP portech 80 a 443, povolen je také protokol ICMP).

Spojení vyznačené přerušovanou čarou vyjadřuje situaci po zprovoznění OpenVPN klienta a serveru a jejich propojení – v této situaci je veškerý provoz ze stroje `ovpn-klient` směrován do laboratorní sítě či Internetu přes stroj `ovpn-server` bez omezení. Komunikace je umožněna VPN spojením na síťové vrstvě, které využívá průchozí port na firewallu.



Obr. 5.1: Topologie virtuální sítě v laboratorní úloze

5.3 Implementace

5.3.1 Operační systém, prostředí

Jako operační systém pro nasazení na virtuálních strojích byl zvolen Ubuntu Linux 11.04. Místo výchozího grafického prostředí (Gnome 3/Unity) bylo nainstalováno prostředí LXDE, a to jednak z důvodu kontroverzních názorů uživatelů na přívětivost ovládání prostředí Unity, a zejména z důvodu nízkých požadavků prostředí LXDE na systémové prostředky.

Byly provedeny základní úpravy prostředí a systému tak, aby se zlepšila použitelnost a aby systém lépe vyhovoval nasazení ve virtuálním stroji jakožto základ laboratorní úlohy. Do systému byly doinstalovány veškeré potřebné balíčky s programy a knihovnami.

Jako virtualizační prostředí byl (dle zadání) zvolen program VMware ve verzi VMware Player, který je zdarma dostupný na stránkách společnosti VMware. Konfigurace virtuálního stroje byla upravena tak, aby provedené změny (vytvořené, upravené soubory apod.) nebyly po vypnutí stroje uloženy[2].

[2]Toto chování lze vypnout zakomentováním volby *scsi0:0.mode = "independent-nonpersistent"* v konfiguračním souboru `ovpn-klient.vmx`, resp. `ovpn-server.vmx`.

5.3.2 Síť, firewall

Nastavení sítě je shrnuto v následující tabulce:

Stroj	Rozhraní	Adresa
ovpn-klient	eth0	192.168.1.2/24
	tun0	DHCP – OpenVPN (z rozsahu 192.168.10.0/24)
ovpn-server	eth0	192.168.1.1/24
	eth1	DHCP – VMware
	tun0	DHCP – OpenVPN (z rozsahu 192.168.10.0/24)

Na stroji `ovpn-server` je povoleno směrování paketů a je na něm nastaven firewall pomocí skriptu `/etc/init.d/firewall` spouštěného při startu systému:

```
#!/bin/bash

ipt="/sbin/iptables"

$ipt -F
$ipt -F -t nat

$ipt -A FORWARD -i eth1 -p tcp --dport http -j ACCEPT
$ipt -A FORWARD -i eth1 -p tcp --dport https -j ACCEPT
$ipt -A FORWARD -i eth1 -p udp --dport domain -j ACCEPT
$ipt -A FORWARD -i eth1 -p udp --dport ntp -j ACCEPT
$ipt -A FORWARD -i eth1 -p icmp -j ACCEPT
$ipt -A FORWARD -i eth1 -j DROP

$ipt -A INPUT -i eth1 -p tcp --dport http -j ACCEPT
$ipt -A INPUT -i eth1 -p tcp --dport https -j ACCEPT
$ipt -A INPUT -i eth1 -p tcp --dport ssh -j ACCEPT
$ipt -A INPUT -i eth1 -p udp --dport domain -j ACCEPT
$ipt -A INPUT -i eth1 -p icmp -j ACCEPT
$ipt -A INPUT -i eth1 -m state --state ESTABLISHED -j ACCEPT
$ipt -A INPUT -i eth1 -j DROP

$ipt -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Skript zajistí, že komunikace, která přichází z rozhraní `eth1` (tzn. od stroje `ovpn-klient` ve výchozím nastavení), bude povolena pouze pokud se bude jednat o TCP provoz s cílovým portem 80 nebo 443 (resp. 22, pokud bude cílem stroj

`ovpn-server`), UDP provoz s cílovým portem 53 nebo 123 nebo o data protokolu ICMP.

5.4 Přínos pro studenty

Laboratorní práce má za cíl seznámit studenty s virtuálními sítěmi realizovanými technologií OpenVPN. Studenti si mohou vyzkoušet konfiguraci, která reprezentuje řešení problému, který se v praxi vyskytuje. Řešení není vyčerpávající, ale dává studentům základní návod, který lze v případě potřeby prakticky použít a případně dále uzpůsobit potřebnému použití.

5.5 Didaktický pohled

Úloha je navržena primárně s cílem umožnit studentům (jak bylo zmíněno výše) konfiguraci sítě, která odráží jeden z typických aplikačních scénářů sítí VPN. Cílem je, aby student pochopil praktický přínos virtuálních sítí typu OpenVPN. Tomu musí pochopitelně předcházet teoretický úvod do problematiky VPN sítí a základní charakteristika a výhody sítí OpenVPN, aby se student v látce rámcově orientoval. Nedílnou součástí zadání laboratorního cvičení jsou i potřebné konfigurační postupy a popis potřebných parametrů.

Mimo zadání byly vypracovány i stručné informace o úloze pro cvičící. Tyto informace obsahují postup instalace strojů, vzorové konfigurace OpenVPN klienta a serveru, problémy, které mohou při cvičení nastat a jejich řešení, a dále návrhy otázek, které mohou být studentům kladeny, a odpovědi na ně. Přílohou této práce je text pro laboratorní úlohu a informace pro cvičící ve formátu OpenDocument (na přiloženém disku DVD). Tyto texty mohou být přímo použity ve cvičeních, případně podle potřeby pozměněny.

6 ZADÁNÍ LABORATORNÍ ÚLOHY

6.1 Teoretický úvod

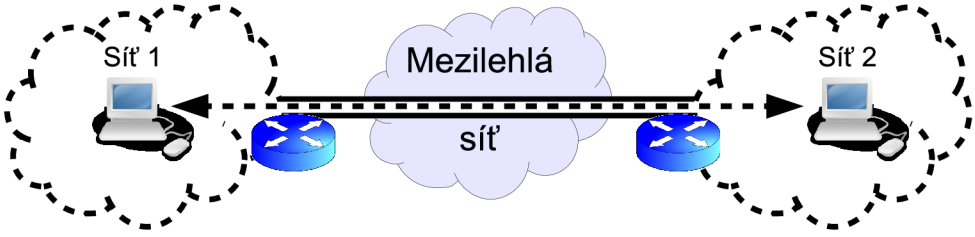
6.1.1 Virtuální privátní sítě – VPN

V počítačových sítích se nezřídka setkáváme s potřebou propojit oddělené sítě (či jednotlivá zařízení) tak, aby bylo možné mezi nimi komunikovat stejným způsobem, jako kdyby byly přímo propojeny do jednoho celku. V praxi se může jednat například o propojení sítí jednotlivých poboček firmy v různých městech nebo státech nebo například o zajištění přístupu cestujícího pracovníka do firemní sítě. Nemusí se nutně jednat o propojování na úrovni celých sítí, v některých případech může stejně tak být záměrem i pouhé propojení dvou zařízení (počítačů), přičemž požadujeme, aby toto propojení bylo zabezpečené.

K těmto účelům se využívá virtuálních privátních sítí (VPN, Virtual Private Networks), nebo tzv. tunelů. Tunel představuje virtuální dvoubodové spojení dvou zařízení v síti (může se jednat jak o koncové uzly, tak i o směrovače zajišťující komunikaci s přilehlými sítěmi), které je realizováno prostřednictvím sítě třetí strany (nejčastěji přes Internet). Virtuální privátní síť pak lze připodobnit množině vzájemně propojených tunelů.

Schématické přiblížení, jak funguje princip tunelu, lze nalézt na obr. 6.1. Sítě *Síť 1* a *Síť 2* si můžeme představit jako dva nezávislé celky (např. ve dvou různých městech). Ty jsou mezi sebou propojeny prostřednictvím směrovačů, které realizují tunel, tedy virtuální propojení, přes Internet. Pro zjednodušení si můžeme představit, že situace je prakticky analogická, jako kdybychom zmíněné dvě sítě propojili kabelem.

VPN (obr. 6.2) fungují v principu velmi podobně, jen je většinou propojeno více sítí nebo více koncových zařízení. Obrázek znázorňuje příklad, kdy pomocí Internetu jsou propojeny dvě sítě (opět mohou komunikovat, jako by byly přímo fyzicky propojeny), a navíc je do vnitřní sítě stejným způsobem umožněn přístup mobilním uživatelům. Pokud veškeré přenosy (včetně přenosů mimo vnitřní síť) od



Obr. 6.1: Princip síťového tunelu

klientů budeme směrovat přes VPN server, můžeme lépe řídit, kam mají klienti přístup. Na druhou stranu jim rovněž můžeme umožnit plnohodnotný přístup do Internetu, pokud jsou například omezováni poskytovatelem připojení k Internetu pouze na využívání některých služeb. Touto situací se budeme v úloze zabývat.

6.1.2 Klíčové vlastnosti VPN

Hlavním požadavkem na VPN je zajištění výše zmíněného transparentního propojení sítí – data směřující z jedné sítě jsou na hranici této sítě prostřednictvím aplikace zajišťující VPN spojení zabalena (enkapsulována) do nových paketů s odlišnými vlastnostmi (typ paketu, zdrojová a cílová adresa), odeslány směrem k cílové síti, na jejíž hranici jsou data opět z paketů rozbalena a doručena ke skutečnému cíli.

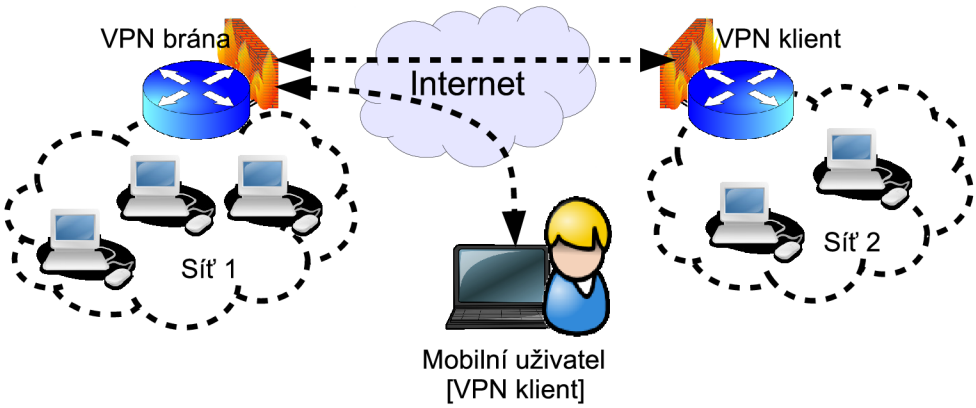
Mimo tyto základní funkce však virtuální privátní sítě mohou nabízet (a typicky také nabízejí) i další možnosti:

- **Autentizaci** – ověření totožnosti komunikujících stran
- **Autorizaci** – řízení přístupu do sítě na základě identity komunikující strany
- **Ověření integrity přenášených zpráv** – ověření, že data po cestě nebyla pozměněna, ať už úmyslně (útočníkem) nebo neúmyslně (chybou při přenosu)
- **Zajištění důvěrnosti zpráv** – jinými slovy, zašifrování přenášených dat, což je typicky jeden z klíčových požadavků na VPN

Autorizace může být řešena přímo aplikací pro VPN nebo jinak, například za pomoci firewallu. Ostatní vlastnosti jsou založeny na kryptografických principech.

6.1.3 OpenVPN

Existují různé možnosti, jak VPN realizovat. Jednou z těchto možností je využití balíku OpenVPN, jehož konfigurací se budeme v této laboratorní úloze zabývat.



Obr. 6.2: Princip virtuálních privátních sítí

Jedná se o svobodný, volně šiřitelný software, postavený nad kryptografickou knihovnou OpenSSL. OpenVPN má kromě své dostupnosti a otevřenosti další výhody, zejména v oblasti bezpečnosti. Filozofií OpenVPN je neduplikování funkcí (tedy využití komponent, které již byly k danému účelu vytvořeny a jsou prověřené), zajištění co nejvyššího stupně bezpečnosti a zároveň poskytnutí vysoké flexibility.

Pomocí OpenVPN lze propojit jak dvě zařízení či sítě v režimu peer-to-peer (dvoubodové spojení), tak i více sítí/zařízení v režimu klient-server. Spojení nemusí být šifrované, ale běžně se šifrování využívá – může se jednat jak o šifrování symetrickým klíčem (tedy stejným klíčem na všech uzlech, které zajišťují VPN spojení), tak i o asymetrické šifrování založené na veřejné infrastruktuře klíčů (PKI, Public Key Infrastructure). Komunikace mezi sítěmi může probíhat na linkové vrstvě nebo na vrstvě síťové (doporučuje se použít propojení na síťové vrstvě, pokud neexistuje důvod pro využití spojení na linkové vrstvě – např. potřeba provozovat v síti aplikace, které ke komunikaci využívají broadcast a fungují pouze v rámci jedné podsítě).

OpenVPN nabízí širokou škálu konfiguračních parametrů, a to jak globálně, tak i individuálně pro jednotlivé připojené klienty. Lze zvolit, jaký protokol bude využit pro přenos dat mezilehlou sítí (běžně se doporučuje použití UDP, protože OpenVPN zprostředkovává spojení na linkové či síťové vrstvě, které je z podstaty nespolehlivé, spolehlivost zajišťuje až protokol TCP či protokoly vyšších vrstev). Rovněž je možné nastavit kompresi přenášených dat, což může umožnit zvýšení propustnosti spojení. Mezi dalšími vlastnostmi lze zmínit i předávání (push) konfiguračních parametrů jednotlivým VPN klientům.

V této úloze se budeme zabývat konfigurací OpenVPN klienta a serveru, přičemž server bude vystupovat v roli výchozí brány a veškerý provoz bude procházet VPN spojením a bude serverem směrován dál. Propojení bude šifrované, využijeme infrastruktury PKI a sady certifikátů a soukromých klíčů.

6.1.4 Asymetrická kryptografie, PKI

Princip asymetrické kryptografie lze zjednodušeně shrnout následovně: Vlastníme soukromý a veřejný kryptografický klíč. Soukromý klíč držíme v tajnosti, zatímco veřejný klíč dáme volně k dispozici komukoliv, kdo o něj má zájem. Pokud zašifrujeme data soukromým klíčem, lze je dešifrovat klíčem veřejným. Pokud nikdo jiný nemá k soukromému klíči přístup, pak lze ověřit, že jsme původci odeslané zprávy (tzn. lze ji dešifrovat naším veřejným klíčem).

Opačného principu lze využít, pokud známe cizí veřejný klíč a chceme druhé straně důvěrně poslat zprávu. Zašifrujeme ji veřejným klíčem druhé strany a pouze ona ji může (pomocí svého soukromého klíče) dešifrovat. Asymetrická kryptografie

je však oproti symetrické (ve které jak odesilatel, tak příjemce používá k šifrování i dešifrování stejný klíč) mnohem náročnější na výpočetní výkon, a proto se používá pouze ve fázi navazování spojení. Dále proběhne bezpečné ustanovení společného klíče (k tomuto se využívá mechanismus Diffie-Hellman), kterým je další komunikace symetricky šifrována. Pro zvýšení bezpečnosti může být tento klíč během přenosu po určitém časovém intervalu změněn.

Infrastruktura PKI je založena na použití asymetrické kryptografie, certifikátů a důvěryhodné třetí strany. Certifikát je množina informací o jeho vlastníkovi (např. jméno, organizace, geografické informace) a jeho veřejný klíč, a následně digitální podpis důvěryhodné třetí strany (otisk – hash informací zašifrovaný soukromým klíčem důvěryhodné třetí strany). Máme-li k dispozici veřejný klíč důvěryhodné třetí strany, můžeme pomocí něj ověřit platnost certifikátu, a tudíž i identitu entity, která se jím prokazuje (tato entita musí samozřejmě znát i soukromý klíč spojený s certifikátem).

Důvěryhodná třetí strana je entita, jejíž totožnost známe a můžeme s jistotou (či vysokou pravděpodobností) předpokládat, že jí vydané informace lze považovat za pravdivé. Jedná se o certifikační autoritu (CA), která může vydávat certifikáty na komerční bázi (ty jsou vhodné, pokud předem nevíme, s kým přesně budeme komunikovat, resp. nemáme nad komunikujícími stranami kontrolu). Další možností je ustanovení vlastní CA a podepisování certifikátů s její pomocí, což je např. pro použití v OpenVPN vhodné řešení (nad OpenVPN serverem i klienty máme kontrolu), v tomto případě není třeba za vydávání certifikátů platit třetí straně.

Balík OpenVPN obsahuje nástroje pro vybudování vlastní certifikační autority a vydávání certifikátů, které lze poměrně jednoduše používat.

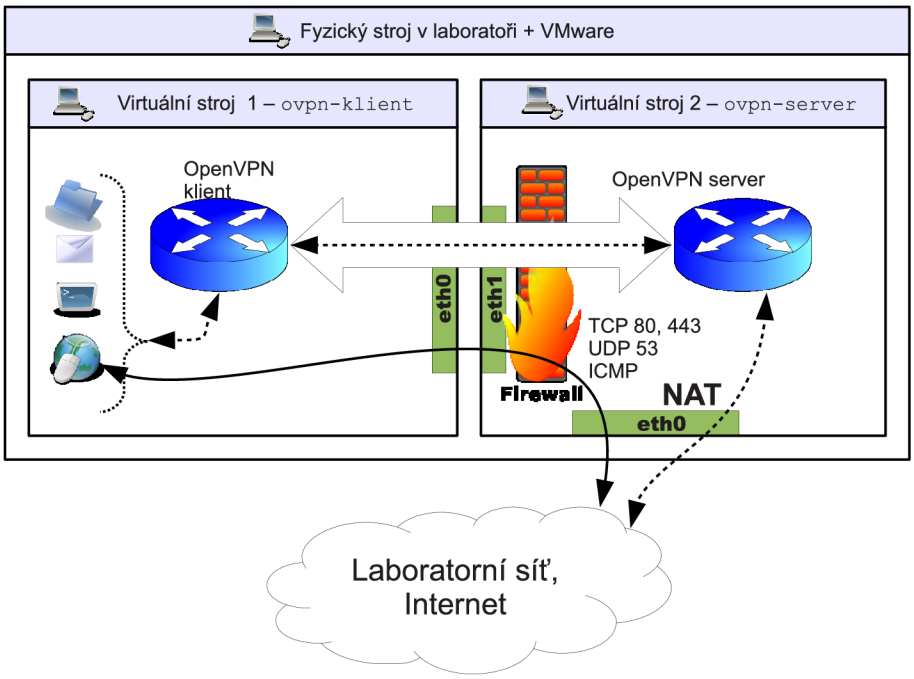
6.2 Úkoly

- Prozkoumejte výchozí stav sítě, zejména na stroji `ovpn-klient`.
- Založte certifikační autoritu, vygenerujte certifikáty a soukromé klíče pro stroj `ovpn-server` a `ovpn-klient`.
- Nastavte OpenVPN na obou strojích a zrealizujte mezi nimi VPN spojení.
- Prozkoumejte, jak se změnilo chování sítě oproti výchozímu nastavení.

6.3 Pracovní postup

6.3.1 Seznámení s prostředím

Úloha je založena na dvou virtuálních strojích s OS Ubuntu Linux 11.10 v prostředí VMWare. Jeden ze strojů bude vystupovat jako OpenVPN klient (stroj `ovpn-klient`),



Obr. 6.3: Topologie virtuální sítě

druhý jako server (stroj `ovpn-server`). Ve výchozím nastavení jsou oba stroje propojeny sítí 192.168.1.0/24, stroj `ovpn-server` má navíc druhé rozhraní s dynamicky přidělovanou adresou (adresu přiděluje program VMWare). Komunikace ze stroje `ovpn-klient` je směrována přes stroj `ovpn-server`, na něm je navíc omezena firewallem. Firewall propouští ze stroje `ovpn-klient` směrem do vnější sítě komunikaci pouze pro protokol **TCP port 80 (HTTP)**, **TCP port 443 (HTTPS)**, **UDP port 53 (DNS)**, **UDP port 123 (NTP)** a rovněž je povolen protokol **ICMP**.

Tato konfigurace simuluje praktickou situaci, ve které disponujeme počítačem připojeným k Internetu (zde `ovpn-klient`), ale jsme omezeni na využívání pouze určitých služeb. Dále máme k dispozici druhý počítač s neomezeným přístupem do Internetu (zde `ovpn-server`), na kterém můžeme nakonfigurovat OpenVPN. OpenVPN nakonfigurujeme i na prvním stroji, zrealizujeme VPN spojení (na portu, který firewallem není blokován) a veškerou komunikaci směrujeme na OpenVPN server (a z něj dále do Internetu). Tím obejdeme původní omezení.

Zapojení je znázorněno na obr. 6.3. Spoje kreslené plnou čarou znázorňují výchozí situaci – komunikace mimo síť, do které je stroj `ovpn-klient` připojen, probíhá přes stroj `ovpn-server`, možnosti přístupu ke službám jsou však omezené. Situace po konfiguraci je znázorněna spoji s přerušovanou čarou – komunikace směřující mimo síť je předávaná přes VPN spojením mezi stroji, a následně do vnější sítě; OpenVPN spojení využívá jednoho z neblokovaných portů na firewallu. Spojení stroje `ovpn-klient` do vnější sítě a Internetu již není omezeno.

6.3.2 Spuštění virtuálních strojů

Spusťte program VMWare ve dvou instancích. V jedné z nich spusťte virtuální stroj `ovpn-server`, ve druhé `ovpn-klient`. Po naběhnutí operačního systému se do strojů přihlaste – uživatelské jméno je `student`, heslo `student`. Na každém ze strojů spusťte terminál (konzoli), pomocí něhož budete stroj konfigurovat.

6.3.3 Výchozí stav sítě

1. Zjistěte, jaká síťová rozhraní jsou ve strojích přítomna a jak jsou nakonfigurována (použijte příkaz `ifconfig`).
2. Zjistěte, jak je nakonfigurováno směrování (příkaz `route -n`), vyzkoušejte dostupnost (`ping`) a trasování cesty na server ležící v Internetu (příkazem `traceroute` *`adresa.trasovaneho.serveru`*). Jako adresu serveru ležícího na Internetu můžete použít např. `www.linux.cz` nebo `ftp.linux.cz`.
3. Vyzkoušejte na stroji `ovpn-klient` možnost přistupovat na webové servery v internetu protokolem HTTP nebo HTTPS (buď příkazem `wget` *`www.adresa.serveru`* `-O -` nebo ve webovém prohlížeči). Stejným způsobem otestujte možnost připojit se na FTP server v Internetu (příkazem `ftp` nebo v prohlížeči), případně další služby. Jaké služby fungují a jaké nikoliv?

6.3.4 Generování certifikátů

1. Na stroji `ovpn-server` je nejprve třeba založit certifikační autoritu. Úkony související se založením certifikační autority a nastavením OpenVPN je třeba provádět s právy superuživatele, proto se přihlaste příkazem `su` jako superuživatel; heslo je stejné jako pro neprivilegovaného uživatele.
   Pokud bychom certifikační autoritu zakládali a používali v praxi, bylo by z bezpečnostních důvodů založit ji na zvláštním stroji určeném pouze pro činnost CA! V našem případě tedy CA založíme na serveru pouze z důvodu jednoduchosti takového řešení.
   Pro založení a správu CA je v balíku OpenVPN k dispozici sada nástrojů *easy-rsa*. Nástroje se v distribuci Ubuntu nacházejí v adresáři `/usr/share/doc/openvpn/examples/easy-rsa/2.0/`, pro přehlednost je nakopírujte do adresáře `/etc/openvpn/easy-rsa`:

   ```
   cd /etc/openvpn
   cp -r /usr/share/doc/openvpn/examples/easy-rsa/2.0 easy-rsa
   cd easy-rsa
   ```

2. V textovém editoru (např. `vim`, `nano`) otevřete soubor `vars`, který obsahuje nastavení proměnných pro skripty sloužící ke generování klíčů. Vyhledejte řádky, ve kterých dochází k nastavení hodnot proměnných `KEY_COUNTRY`, `KEY_PROVINCE`, `KEY_CITY`, `KEY_ORG`, `KEY_EMAIL` (nacházejí se na konci souboru). Proměnné jsou přednastaveny na výchozí hodnoty, jsou v nich obsaženy identifikační údaje CA, tedy země, část země (oblast) a město, kde CA sídlí, dále název organizace a kontakt na správce CA. Nastavte proměnné na adekvátní hodnoty, například takto:

```
export KEY_COUNTRY="CZ"
export KEY_PROVINCE="Jihomoravsky kraj"
export KEY_CITY="Brno"
export KEY_ORG="FEKT VUT"
export KEY_EMAIL="admin@moje-ca.cz"
```

3. Založení certifikační autority zahrnuje vytvoření kořenového certifikátu CA a vygenerování soukromého klíče, kterým mohou být podepisovány certifikáty vydávané CA. Inicializujte nastavení proměnných z předchozího kroku, proveďte vymazání přítomných klíčů (zatím žádné klíče nejsou přítomny, ale příkaz pro vymazání klíčů zároveň slouží k potřebné inicializaci adresáře pro ukládání klíčů) a vytvořte kořenový certifikát CA a soukromý klíč CA:

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

   Dojde k vygenerování soukromého klíče CA `ca.key` a následně bude požadováno vyplnění informací pro certifikát CA. Některé položky jsou již předvyplněné na hodnoty nastavené v předchozím kroku – jsou uvedeny v hranatých závorkách. Pokud chcete použít přednastavenou hodnotu, potvrďte ji stiskem klávesy Enter. Pokud je třeba nabídnutou hodnotu změnit, zadejte požadovanou hodnotu a opět ji potvrďte klávesou Enter. Mimo hodnoty přednastavené v předchozím kroku je požadováno vyplnění hodnot *Organizational Unit Name* (jméno organizační jednotky, můžete vyplnit například `UTKO` nebo ponechat hodnotu prázdnou), *Common Name* (obecné jméno – můžete ponechat nabízenou hodnotu, nebo zadat vlastní název CA) a jméno spojené s certifikátem. Po zadání všech hodnot dojde k vytvoření kořenového certifikátu CA `ca.crt` v adresáři `keys`.
4. V dalším kroku vytvoříte soukromý klíč (`server.key`) a certifikát (`server.crt`) pro OpenVPN server. K tomu stačí spustit příkaz

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

a opět zadat požadované informace vztahující se k certifikátu OpenVPN serveru. Ponechte předvyplněné hodnoty, jako hodnotu organizační jednotky můžete opět vyplnit např. `UTKO` nebo položku ponechat prázdnou. Skript se bude dotazovat na případné heslo (*challenge password*), opět ponechte prázdné (ochrana klíče heslem je pro server z povahy jeho využití nežádoucí)! Poslední dotazovanou položkou je volitelné jméno společnosti (*optional company name*) – ponechte prázdné, nebo zadejte hodnotu dle vlastního uvážení. Nakonec stačí potvrdit podepsání certifikátu pro server zadáním písmena `y` a stiskem klávesy Enter (potvrzení je skriptem požadováno ve dvou krocích, proto je proveďte dvakrát).

5. Podobně jako v předchozím kroku vytvořte soukromý klíč (`ovpn-klient.key`) a certifikát (`ovpn-klient.crt`) pro OpenVPN klienta, při vyplňování údajů a potvrzení vytvoření certifikátu postupujte jako v předchozím kroku:

   ```
   ./build-key ovpn-klient
   ```

6. Aby bylo možné použít protokol pro výměnu klíčů Diffie-Hellman, je třeba pro něj nejprve vygenerovat parametry (generování může trvat i několik desítek sekund):

   ```
   ./build-dh
   ```

7. Zkopírujte vytvořený certifikát CA, parametry DH, certifikát serveru a soukromý klíč serveru z adresáře `/etc/openvpn/easy-rsa/keys` do konfiguračního adresáře OpenVPN na serveru (`/etc/openvpn`):

   ```
   cd /etc/openvpn/easy-rsa/keys
   cp ca.crt server.crt server.key dh1024.pem /etc/openvpn
   ```

8. Zkopírujte vytvořený certifikát CA, certifikát klienta a soukromý klíč klienta do konfiguračního adresáře OpenVPN na klientovi (taktéž `/etc/openvpn`, zkopírování na druhý stroj se provede pomocí nástroje `scp`):

   ```
   scp ca.crt ovpn-klient.crt ovpn-klient.key \
    192.168.1.2:/etc/openvpn
   ```

6.3.5 Nastavení OpenVPN serveru

1. Pro konfigurační a další potřebné soubory aplikace OpenVPN je určen adresář `/etc/openvpn`. Konfigurace OpenVPN serveru se provádí nastavením konfiguračních direktiv v souboru `/etc/openvpn/server.conf`. Ve výchozím stavu soubor neexistuje a je potřeba jej nejprve vytvořit. Je možné jeho obsah

vytvořit ručně, výhodnější a rychlejší je však použít vzorový soubor s konfigurací (`/usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz`) a ten upravit tak, aby vyhovoval našim potřebám. Soubor je třeba zkopírovat do konfiguračního adresáře OpenVPN a rozbalit:

```
cd /etc/openvpn
cp /usr/share/doc/openvpn/examples/sample-config-files/\
 server.conf.gz ./
gunzip server.conf.gz
```

2. Dalším krokem je úprava konfiguračních direktiv v souboru `/etc/openvpn/server.conf`. Výhodou je, že prakticky všechny potřebné direktivy jsou ve výchozím souboru obsažené (tzn. stačí je pouze upravit) a rovněž u nich je vždy umístěn popis s jejich významem a možnostmi nastavení. Symbol mříže (#) uvozuje komentář, středník slouží k deaktivaci následující konfigurační direktivy nebo taktéž uvozuje komentář. V konfiguračním souboru jsou pro nás důležité tyto direktivy:
   - **port *XYZ*** – OpenVPN server bude naslouchat na portu číslo *XYZ*. Nastavte port 53.
   - **proto *PROTOKOL*** – Určuje, jakým protokolem budou data VPN spojení přenášena (UDP, TCP) – ve výchozím nastavení je zvolen protokol UDP, toto nastavení ponechte.
   - **dev *TYP*** – Určuje, jaký typ rozhraní bude určen pro přenos dat. Typ může být buď `tun`, pak bude spojení VPN komunikovat na síťové vrstvě, nebo `tap`, v takovém případě bude komunikace probíhat na linkové vrstvě. Opět ponechte výchozí nastavení.
   - **ca**, **cert**, **key**, **dh** – Tyto direktivy slouží k zadání cest k souborům obsahujícím kořenový certifikát CA, certifikát serveru, soukromý klíč serveru a parametry DH. Ponechte výchozí nastavení.
   - **server *SÍŤ MASKA*** – Nastavení adresy a masky virtuální privátní sítě. Z tohoto rozsahu budou přiděleny adresy serveru a VPN klientu. Nastavte adresu sítě `192.168.10.0` a masku `255.255.255.0`. Můžete zvolit i vlastní nastavení, avšak uvědomte si, jaké rozsahy adres jsou v síti použity, aby nedošlo ke konfliktu!
   - **push "*parametry*"** – Server může při sestavení spojení klientům předat konfigurační parametry sítě, které klienti u sebe aplikují. V našem případě se bude jednat o nastavení OpenVPN serveru jako výchozí brány a nastavení adresy DNS serveru na `8.8.8.8`:

     ```
     push "redirect-gateway"
     push "dhcp-option DNS 8.8.8.8"
     ```

- **user *U*, group *G*** – Umožňuje, aby OpenVPN po inicializaci běžel pod daným uživatelem *U* a skupinou *G*. Server startuje s právy superuživatele, po inicializaci se může těchto práv vzdát a běžet pod obyčejným uživatelem/skupinou, což je z bezpečnostního hlediska výhodnější. Tyto položky aktivujte (odkomentujte).
- **comp-lzo** – Zapíná kompresi přenášených dat algoritmem LZO, čímž umožňuje za určitých okolností zvýšit propustnost VPN spojení.
- **status *CESTA*** – Definuje cestu k souboru, v němž bude možné sledovat stav OpenVPN serveru. Zde nastavte `/var/log/openvpn-status.log`.

Provedené změny v souboru uložte.

3. Nyní, když je konfigurace hotová, zbývá spustit OpenVPN server:

```
service openvpn start
```

Pokud dojde během startu k chybě, zkontrolujte nastavení v konfiguračním souboru, přítomnost všech potřebných souborů pro šifrování/autentizaci (certifikát CA, certifikát serveru, soukromý klíč serveru, soubor s parametry DH). Dále zkontrolujte případná chybová hlášení OpenVPN v systémovém záznamu (soubor `/var/log/syslog`).

6.3.6 Nastavení OpenVPN klienta

1. Konfigurace klienta (na stroji `ovpn-klient`) je velmi podobná konfiguraci serveru (konfiguraci je opět nutné provádět pod superuživatelem). Opět je nejdříve nutné nakopírovat do konfiguračního adresáře vzorový soubor s konfigurací:

```
cd /etc/openvpn
cp /usr/share/doc/openvpn/examples/sample-config-files/\
 client.conf ./
```

2. V konfiguraci klienta (`/etc/openvpn/client.conf`) jsou podstatné tyto direktivy:
   - **client** – Volba způsobí, že se aplikace OpenVPN bude chovat jako klient.
   - **remote *SERVER PORT*** – Adresa a port OpenVPN serveru, ke kterému se klient bude připojovat. V tomto případě zadejte adresu OpenVPN serveru a dále port, který byl nastaven v konfiguraci serveru:

     ```
     remote 192.168.1.1 53
     ```

   - **cert**, **key** – Stejně jako u serveru se jedná o nastavení certifikátu a soukromého klíče, tentokrát však pro OpenVPN klienta:

```
cert ovpn-klient.crt
key ovpn-klient.key
```

- Stejně jako na serveru zde odkomentujte položky `user` a `group`.

3. Aktivujte službu OpenVPN, čímž dojde k vytvoření spojení se serverem (pokud jste správně provedli konfiguraci):

   ```
   service openvpn start
   ```

   V případě, že dojde během startu k chybě, zkontrolujte nastavení v konfiguračním souboru, přítomnost všech potřebných souborů pro šifrování/autentizaci (certifikát CA, certifikát klienta, soukromý klíč klienta) a dále případná chybová hlášení v systémovém záznamu (soubor /var/log/syslog).

6.3.7 Stav sítě při VPN spojení

1. Prozkoumejte (jak na klientovi, tak na serveru) zprávy OpenVPN démona v systémovém záznamu:
   `less /var/log/syslog`
2. Zjistěte, jak se na klientovi změnila dostupná síťová rozhraní a jak se změnilo směrování. Vyzkoušejte trasování cesty na server umístěný na Internetu a porovnejte s pozorováním, které jste provedli na začátku cvičení (použijte stejný postup).
3. Vyzkoušejte na klientovi funkčnost služeb, které ve výchozím stavu sítě nefungovaly (např. FTP).
4. Na serveru zobrazte informace o připojených OpenVPN klientech a informace prostudujte:
   `less /var/log/openvpn-status.log`

7 INFORMACE PRO CVIČÍCÍ

7.1 Instalace virtuálních strojů

Virtuální stroje pouze stačí nakopírovat na příslušný počítač, na kterém budou provozovány, a následně je rozbalit. Poté lze spustit oba virtuální stroje otevřením příslušných `.vmx` souborů.

Podstatné je při prvním spuštění stroje na dotaz programu VMware, zda byl daný virtuální stroj zkopírován, nebo přesunut, **zvolit, že stroj byl přesunut**. V případě druhé volby by program VMware vytvořil pro virtuální stroj novou síťovou kartu s novou MAC adresou, a tím by došlo ke změně jeho konfigurace, a tudíž i k nastavení IP adres na virtuálním stroji.

7.2 Vzorové konfigurace

7.2.1 server.conf

V konfiguraci OpenVPN serveru (v souboru `/etc/openvpn/server.conf`) je třeba upravit/přidat oproti výchozímu stavu tyto položky:

```
port 53
server 192.168.10.0 255.255.255.0
push "redirect-gateway"
push "dhcp-option DNS 8.8.8.8"
user nobody
group nogroup
status /var/log/openvpn-status.log
```

7.2.2 client.conf

V konfiguraci OpenVPN klienta (v souboru `/etc/openvpn/client.conf`) by měly být změněny/nastaveny oproti výchozímu stavu tyto položky:

```
remote 192.168.1.1 53
user nobody
group nobody
cert ovpn-klient.crt
key ovpn-klient.key
```

7.3 Možné problémy

7.3.1 Server

V případě výskytu problémů na straně serveru je vhodné podívat se na výpis hlášení OpenVPN serveru v systémovém záznamu (logu) – `/var/log/syslog`, z nich lze většinou poměrně snadno zjistit, v čem je příčina problému. Pokud se server nepodaří spustit (objeví se chybová hláška při pokusu o spuštění), je vhodné zkontrolovat, zda jsou v adresáři `/etc/openvpn` přítomny soubory soukromého klíče, certifikátu CA, parametrů DH a certifikátu serveru.

Další problémy spojené se stranou serveru mohou být např.

- překlep v konfiguraci, zapomenutý parametr,
- nastavený/odkomentovaný parametr v konfiguraci, se kterým nemělo být manipulováno,
- chybně nastavený port serveru,
- vybraný rozsah adres pro virtuální spojení z nevhodného rozsahu, dochází k překrytí (pokud student zvolí rozsah sám).

Při generování certifikátů/klíčů by pravděpodobně neměly nastat problémy, pokud nastanou, pak patrně z nepozornosti. V takovém případě je vhodné zopakovat pečlivě celý postup podle zadání a řídit se pokyny skriptů pro generování certifikátů/klíčů.

7.3.2 Klient

Stejně jako na straně serveru je vhodným výchozím bodem pro hledání příčin problému systémový záznam. Dále je vhodné zkontrolovat, že

- byly v konfiguračním souboru správně nastaveny potřebné parametry podle zadání úlohy,
- do adresáře `/etc/openvpn` byly nakopírovány všechny potřebné soubory pro autentizaci, tzn. certifikát CA, certifikát klienta, soukromý klíč klienta,
- IP adresa (a zejména port) nastavený na straně klienta odpovídá nastavení na straně serveru,
- rozhraní, kterým je klient připojen k serveru (`eth0`), je aktivní a spojení mezi virtuálními stroji (v rozsahu 192.168.1.0/24) funguje (nedošlo např. omylem k přenastavení).

7.4 Otázky a odpovědi k tématu OpenVPN

Pro ujištění, že student látku probíranou v tomto cvičení pochopil, popř. pro jeho hlubší zamyšlení nad problematikou, mu mohou být pokládány otázky související se cvičením. Následuje seznam navrhovaných otázek, které by k tomuto účelu mohly být použity, spolu s odpověďmi naznačujícími směr, kterým by se měly úvahy studenta nad daným problémem ubírat.

- *Jaký port byl zvolen pro provoz OpenVPN serveru a proč? Bylo by použití tohoto portu v praxi problematické – pokud ano, za jakých okolností?*
  Byl nastaven port UDP 53, a to z důvodu, že se jedná o jeden z portů, které nejsou na firewallu blokovány. Bylo by možné použít i jiný neblokovaný port (např. TCP 80). V praxi by použití portu UDP 53 problematické bylo v případě, že by na stejném stroji měl běžet i DNS server (port 53 je za normálních okolností využit právě pro DNS).
- *Proč je pro data přenášená VPN spojením použit protokol UDP? Lze použít protokol TCP, mělo by to výhody nebo nevýhody?*
  Ve výchozím nastavení OpenVPN je použit nespolehlivý protokol UDP (toto nastavení je doporučeno) – jeho nespolehlivost zde není na závadu, protože VPN spojení nahrazuje nespolehlivé spojení na síťové, příp. linkové vrstvě. Spolehlivost je pak v případě potřeby zajištěna protokoly vyšších vrstev (zejm. TCP). Protokol TCP pro VPN tunel lze taktéž použít (někdy je to i nutné – pokud není k dispozici, např. z důvodu blokování na firewallu, vhodný UDP port). Potenciálně však mohou nastat problémy s výkonem VPN spojení, protože spolehlivost bude zajištěna jak vnějším spojením VPN tunelu, tak TCP spojením uvnitř tunelu, a vzhledem k tomu, že vnitřní spojení počítá s nespolehlivým přenosovým médiem, může docházet k problémům.
- *K čemu v konfiguraci slouží parametr* `comp-lzo`*? Má vliv na datovou propustnost VPN spojení? Pokud ano, za jakých okolností?*
  Parametr aktivuje kompresi přenášených dat algoritmem LZO. Může tak dojít ke zvýšení propustnosti (ke snížení by dojít nemělo – pokud OpenVPN zjistí, že po kompresi by bylo nutné přenášet více dat než bez ní, komprese se nepoužije). Podmínkou však je, že jsou přenášena komprimovatelná data (nejlépe data, která ještě komprimována nebyla), např. při přenášení dat typu JPEG nebo MPEG nebude propustnost o tolik vyšší jako např. při přenosu textu.
- *Existují omezení na volbu adresního rozsahu, ze kterého jsou přidělovány adresy pro VPN spojení? Co je v tomto případě dobré brát v úvahu (zamyslete se, odkud se VPN klienti mohou připojovat)?*
  Adresní rozsah pochopitelně nesmí kolidovat s žádným rozsahem použitým

v síti. Taktéž je dobré zvážit, jaké adresní rozsahy bývají často použity na směrovačích v domácnostech nebo malých firmách, odkud se klienti mohou také připojovat (typicky například 192.168.0.0/24). Z toho plyne, že pro VPN spojení je vhodné zvolit jiný rozsah (v tomto cvičení je doporučen rozsah 192.168.10.0/24).

- *Je pro provoz VPN nezbytné použití asymetrické kryptografie? Pokud ano, proč; pokud ne, existují jiné možnosti?*
  Použití asymetrické kryptografie není z hlediska funkce VPN nezbytné. Stejně tak lze použít kryptografii symetrickou (na všech klientech a serveru je použit stejný šifrovací klíč) nebo je možný (i když v praxi zcela nevhodný) i provoz bez šifrování. Asymetrická kryptografie je složitější na správu než symetrická, má však podstatné výhody (vzájemná autentizace klienta i serveru, nebo situace při kompromitaci jednoho z klíčů – stačí revokovat patřičný klientský certifikát a vydat nový, zásah se týká jen jednoho stroje).
- *K čemu slouží parametry* `user` *a* `group` *v konfiguraci OpenVPN?*
  Tyto parametry určují, s jakými oprávněními (pod jakým uživatelem a skupinou) poběží OpenVPN démon po inicializaci (prvotně je spuštěn s právy superuživatele). Jedná se o bezpečnostní návrh – v případě, že by démon obsahoval chybu a byl kompromitován, získá útočník přístup do systému pouze s oprávněními obyčejného uživatele a nikoliv rovnou superuživatele.

8 TESTOVÁNÍ, DALŠÍ MOŽNÉ ROZŠÍŘENÍ

8.1 Testování

Úloha byla otestována s cílem ověřit korektnost textu zadání a na základě tohoto testu byl text zadání dále upraven, aby reflektoval nalezené nedostatky. Dále byla otestována funkčnost virtuálních strojů (a celé úlohy) v různých prostředích, a to pod OS Ubuntu Linux 11.10 v programu VMware Player 4.0.2 a Windows 7 Professional 4.0.3. Byl proveden i test na školním stroji s Windows XP Professional, kde se však ukázalo, že pro vytvořené obrazy virtuálních strojů je potřeba novější verze programu VMware Player než verze nainstalovaná.

Test na školním stroji v prostředí Ubuntu/VMware 4.0.3 se ukázal jako bezproblémový. Po instalaci aktuální verze programu VMware na školní stroje by tedy mělo nasazení proběhnout bez jakýchkoliv potíží.

8.2 Další možné rozšíření

Navržená úloha může být dále rozšířena (např. pokud by se po praktickém nasazení ve výuce ukázalo, že je pro studenty z časového hlediska snadno zvládnutelná) nebo může být využita jako základ pro jinou úlohu využívající OpenVPN (přinejmenším virtuální stroje by se pravděpodobně daly využít pouze s malými změnami).

Mezi možnými rozšířeními se nabízí sledování provozu na síti vhodným nástrojem (např. `wireshark` nebo `tcpdump`), porovnání stavu přímého propojení strojů (bez šifrování) a propojení pomocí VPN (šifrované spojení). Dále by bylo možné využít možností nastavení OpenVPN - například individuálních nastavení pro jednotlivé OpenVPN klienty, možnost povolení/zákazu vzájemné komunikace mezi klienty, šifrování symetrickým klíčem, propojení v režimu peer-to-peer nebo pokročilé možnosti autentizace (např. pomocí lokální systémové databáze uživatelů nebo LDAP).

Jinou možností pro návrh nové úlohy je využití produktu OpenVPN Access Server. Dalším uvažovaným námětem je použití nevhodně připravené konfigurace, kterou by studenti měli za úkol uvést do řádného stavu (tím by si procvičili řešení problémů) nebo zavést umělá omezení (např. ztrátovost linky propojující virtuální stroje) s cílem vyladit spojení, aby podávalo co nejlepší výkon.

V případě, že by úloha byla podstatně rozšířena, mohlo by být vhodné rozčlenit ji do samostatných celků, na kterých by studenti mohli pracovat paralelně po dvojicích (příp. větších skupinách) na dvou počítačích. Doplnění stávající úlohy nebo vypracování nové úlohy navazující na ni bude však vhodné zvážit, až bude úloha vyzkoušena přímo ve výuce s větším množstvím studentů. Nasazením ve výuce se

nejlépe ukáže, zda jsou studenti schopni v jednom cvičení zpracovat větší rozsah látky, nebo zda naopak stávající koncept úlohy obsahuje místa, která budou pro studenty složitá.

9 ZÁVĚR

Virtuální privátní sítě představují často používaný prvek při výstavbě sítí. Zajišťují propojení vzdálených sítí nebo přístup do vzdálených sítí obecně, v principu napodobují situaci, kdy jsou komunikující strany propojeny přímo.

Existuje celá řada technologií, které lze k realizaci VPN použít, jednou z nich je i OpenVPN. Jedná se o otevřenou a volně dostupnou technologii s velmi širokými možnostmi uplatnění. Ve srovnání s jinými technologiemi má některé nedostatky, avšak také mnoho výhod, které je vhodné zvážit v případě nasazování VPN technologií do sítě nebo při přechodu ze stávající technologie na jinou. Filozofií OpenVPN je jednoduchost a z ní plynoucí bezpečnost, a to při zachování rozmanitých možností nastavení; při návrhu byl také kladen důraz na praktickou použitelnost.

Pro OpenVPN existují doplňky třetích stran, které rozšiřují možnosti přizpůsobení, konfigurace a správy. Dále existuje komerční verze – OpenVPN Access Server, která je určena pro firmy a organizace a poskytuje komplexní balík pro jednoduchou správu a nastavení serveru OpenVPN včetně správy uživatelů.

Tato práce se kromě popisu charakteristik OpenVPN dále věnuje návrhu laboratorního cvičení sloužícího pro seznámení studentů s virtuálními privátními sítěmi typu OpenVPN. Cvičení bylo navrženo s ohledem na výchozí podmínky; zaměřuje se na prakticky využitelnou situaci, ve které je využito OpenVPN klienta a serveru s cílem obejít omezení v síti (blokování služeb firewallem).

Součástí je jak vlastní zadání (včetně teoretického úvodu), tak i informace pro cvičící a obrazy virtuálních strojů, na nichž cvičení bude probíhat. Zadání bylo na virtuálních strojích úspěšně otestováno, takže je zajištěna korektnost všech uvedených postupů. Virtuální stroje byly otestovány i na školních počítačích, problematická byla verze nainstalovaného software VMware Player; po instalaci aktuální verze by nasazení úlohy mělo být bezproblémové.

Úloha může být využita v rámci předmětu *Návrh, správa a bezpečnost sítí*; v rámci práce byly rovněž navrženy možné úpravy a rozšíření, úloha může taktéž posloužit jako základ pro jiné laboratorní úlohy podobného charakteru.

LITERATURA

[1] Virtual private network. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 31 December 2004, last modified on 1 November 2011 [cit. 2011-11-12]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Virtual_private_network>.

[2] KRČMÁŘ, Petr. *Linux : postavte si počítačovou síť*. Praha : Grada Publishing, 2008. 184 s.

[3] Point-to-Point Tunneling Protocol. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 22 December 2005, last modified on 20 February 2010 [cit. 2011-11-13]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Pptp>.

[4] Layer 2 Tunneling Protocol. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 8 March 2007, last modified on 8 March 2007 [cit. 2011-11-13].
Dostupné z WWW: <http://en.wikipedia.org/wiki/L2TP>.

[5] IPsec. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 8 March 2002, last modified on 26 November 2011 [cit. 2011-11-28]. Dostupné z WWW: <http://en.wikipedia.org/wiki/IPsec>.

[6] Transport Layer Security. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 7 December 2001, last modified on 22 November 2011 [cit. 2011-11-23]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Transport_Layer_Security>.

[7] Secure Socket Tunneling Protocol. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 23 January 2007, last modified on 4 October 2011 [cit. 2011-12-01]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Secure_Socket_Tunneling_Protocol>.

[8] FONTANA, John. *Techworld.com* [online]. 22 January 2007 [cit. 2011-11-11]. Microsoft develops new tunneling protocol. Dostupné z WWW: <http://news.techworld.com/networking/7814/microsoft-develops-new-tunneling-protocol/>.

[9] Openvpn. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 13 November 2004, last modified on 3 June 2010 [cit. 2011-11-15]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Openvpn>.

[10] *OpenVPN Access Server : System Administrator Guide* [online]. Pleasanton : OpenVPN Technologies, 2010 [cit. 2011-11-30]. Dostupné z WWW: <http://www.openvpn.net/images/pdf/OpenVPN_Access_Server_Sysadmin_Guide_Rev.pdf?Name=Value>.

[11] *OpenSSL : The Open Source Toolkit for SSL/TLS* [online]. 2009 [cit. 2011-11-28]. Dostupné z WWW: <www.openssl.org>.

[12] FEILNER, Markus; GRAF, Robert. *Beginning OpenVPN 2.0.9*. Birmingham : Packt Publishing Ltd., 2009. 356 s.

[13] HOSNER, Charlie. *OpenVPN and the SSL VPN Revolution* [online]. Bethesda : SANS Institute, 2004 [cit. 2011-11-10]. Dostupné z WWW: <http://www.sans.org/rr/whitepapers/vpns/1459.php>.

[14] *OpenVPN* [online]. 2008 [cit. 2011-12-01]. HOWTO. Dostupné z WWW: <http://openvpn.net/index.php/open-source/documentation/howto.html>.

[15] PKCS11. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 25 October 2006, last modified on 11 November 2011 [cit. 2011-12-05]. Dostupné z WWW: <http://en.wikipedia.org/wiki/PKCS11>.

[16] TITZ, Olaf. *Olaf Titz* [online]. 2001 [cit. 2011-12-01]. Why TCP Over TCP Is A Bad Idea. Dostupné z WWW: <http://sites.inka.de/bigred/devel/tcp-tcp.html>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

3DES	Triple Data Encryption Standard
AD	Active Directory
AES	Advanced Encryption Standard
BSD	Berkeley Software Distribution
CA	Certifikační autorita
CHAP	Challenge-Handshake Authentication Protocol
CLI	Command Line Interface
CRL	Certificate Revocation List
DLL	Dynamic-Link Library
DoS	Denial of Service
DSA	Digital Signature Algorithm
EAP-TLS	Extensible Authentication Protocol-Transport Layer Security
GRE	Generic Routing Encapsulation
GUI	Graphical User Interface
HMAC	Hash Message Authentication Code
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IMAP	Internet Message Access Protocol
IMAPS	Internet Message Access Protocol Secure
IP	Internet Protocol
ISO	International Organization for Standardization
L2TP	Layer 2 Tunneling Protocol
LDAP	Lightweight Directory Access Protocol
LZO	Lempel, Ziv, Oberhumer

MAC	Message Authentication Code
MD5	Message-Digest Algorithm 5
MPPE	Microsoft Point-to-Point Encryption
MSCHAP	Microsoft Challenge-Handshake Authentication Protocol
NAT	Network Address Translation
OSI	Open System Interconnection
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PKCS#11	Public-Key Cryptography Standard #11
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunneling Protocol
RADIUS	Remote Authentication Dial In User Service
RFC	Request for Comments
RSA	Rivest Shamir Adleman
SHA	Secure Hash Algorithm
SMTP	Simple Mail Transfer Protocol
SMTPS	Simple Mail Transfer Protocol Secure
SSH	Secure Shell
SSL	Secure Sockets Layer
SSTP	Secure Sockets Tunneling Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UCARP	Userland Common Address Redundancy Protocol
UDP	User Datagram Protocol

VPN	Virtual Private Network
WINS	Windows Internet Name Service

SEZNAM PŘÍLOH

Součástí této práce jsou následující přílohy v elektronické podobě, které jsou uloženy na přiloženém disku DVD:

- Zkomprimované obrazy virtuálních strojů pro prostředí VMware
- Text bakalářské práce ve formátu PDF
- Zadání laboratorní úlohy ve formátu OpenDocument (LibreOffice Writer)
- Informace pro cvičící ve formátu OpenDocument (LibreOffice Writer)