Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Srovnání VPN protokolů a jejich využívání v praxi

2023

David Novák

Vedoucí práce:
Mgr. Radek Janoštík, Ph.D.

Studijní program: Informační technologie,
kombinovaná forma

Bibliografické údaje

Autor:	David Novák
Název práce:	Srovnání VPN protokolů a jejich využívání v praxi
Typ práce:	bakalářská práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2023
Studijní program:	Informační technologie, kombinovaná forma
Vedoucí práce:	Mgr. Radek Janoštík, Ph.D.
Počet stran:	86
Přílohy:	elektronická data uložena na úložišti katedry informatiky
Jazyk práce:	český

Bibliographic info

Author:	David Novák
Title:	Copmaring of VPN protocols and it's practical usage
Thesis type:	bachelor thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2023
Study program:	Information Technologies, combined form
Supervisor:	Mgr. Radek Janoštík, Ph.D.
Page count:	86
Supplements:	electronical data stored in the storage of department of computer science
Thesis language:	Czech

Anotace

Bakalářská práce se zabývá technologií Virtuálních pritátních sítí (VPN). Součástí teoretické části práce je popis principu VPN, nejpoužívanějších protokolů, jejich analýza a následné srovnání. Zkoumán bude jejich výkon, bezpečnost a dostupnost klientů pro různé platformy. Na základě analýzy bude uvedeno vhodné užití daných protokolů v praxi. Zhodnocena bude rovněž náročnost nasazení na straně klienta / serveru a také porovnání výkonu a propustnosti na slabších zařízeních, jako jsou jednodeskové počítače či mobilní telefony.

Synopsis

The bachelor thesis deals with Virtual private network technologies (VPN). The theoretical part describes the principle of VPN, the most commonly used protocols, their analysis and comparison. The performance, security and availability of clients will be examined for each protocol. Based on the analysis, there will be mentioned the best practical use of each protocol. The difficulty of each protocol's implementation on client / server side will also be evaluated. The last part of the thesis will mention comparison of performance of each protocol while implemented on lower performance hardware as single board computers or mobile phones.

Klíčová slova: VPN, GRE, PPP, PPTP, IPSec, WireGuard, OpenVPN, iperf

Keywords: VPN, GRE, PPP, PPTP, IPSec, WireGuard, OpenVPN, iperf

Chtěl bych poděkovat především Mgr. Radku Janoštíkovi, Ph.D. za zpětnou vazbu, doporučení a připomínky při zpracovávání bakalářské práce. Dále bych chtěl poděkovat kolegovi Miroslavu Kokrdovi za užitečné rady při testování.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce podpis autora

Obsah

1 Úvod **11**

2 Virtuální privátní sítě **11**
2.1 Historie . . . 12
2.2 Definice . . . 12
2.3 Obecné rozdělení . . . 13
2.4 Fyzické rozdělení . . . 14
2.4.1 Hardware . . . 14
2.4.2 Software . . . 15
2.5 Bezpečnostní prvky . . . 16

3 Síťové modely **18**
3.1 Princip vrstvení . . . 18
3.2 Princip zapouzdržení . . . 20
3.3 OSI model . . . 20
3.4 TCP/IP model . . . 21

4 Síťový model OSI **22**
4.1 Fyzická vrstva . . . 22
4.2 Linková vrstva . . . 22
4.3 Síťová vrstva . . . 22
4.4 Transportní vrstva . . . 23
4.5 Relační vrstva . . . 23
4.6 Prezentační vrstva . . . 23
4.7 Aplikační vrstva . . . 24

5 Tunelovací protokoly **24**
5.1 GRE . . . 24
5.2 PPTP . . . 25
5.3 L2F . . . 26
5.4 L2TP . . . 27
5.5 IPSec . . . 28
5.6 OpenVPN . . . 31
5.7 WireGuard . . . 32

6 Nasazení zvolených protokolů **32**
6.1 PPTP . . . 33
6.2 IPSec . . . 36
6.3 OpenVPN . . . 45
6.4 WireGuard . . . 50

7 Testování protokolů **53**
7.1 Odezva . . . 53
7.2 Propustnost . . . 54
7.3 Hardwarová zátěž . . . 56
7.3.1 PPTP . . . 57
7.3.2 IPSec . . . 58
7.3.3 OpenVPN . . . 59
7.3.4 WireGuard . . . 59
7.3.5 Hardwarová zátěž - Shrnutí výsledků měření . . . 60
7.4 Stabilita . . . 62
7.4.1 PPTP . . . 62
7.4.2 IPSec . . . 64
7.4.3 OpenVPN . . . 65
7.4.4 WireGuard . . . 66
7.4.5 Shrnutí výsledků měření . . . 68
7.5 Bezpečnost . . . 70
7.5.1 PPTP . . . 70
7.5.2 IPSec/IKEv2 . . . 71
7.5.3 OpenVPN . . . 71
7.5.4 WireGuard . . . 71
7.6 Náročnost konfigurace . . . 72

8 Propustnost na slabších zařízeních **73**

9 Dostupnost klientů pro různé platformy **74**
9.1 Linux . . . 74
9.1.1 PPTP . . . 74
9.1.2 IPSec . . . 74
9.1.3 OpenVPN . . . 75
9.1.4 WireGuard . . . 75
9.2 Windows . . . 75
9.2.1 PPTP . . . 75
9.2.2 IPSec . . . 75
9.2.3 OpenVPN . . . 75
9.2.4 WireGuard . . . 76
9.3 iOS . . . 76
9.3.1 PPTP . . . 76
9.3.2 IPSec . . . 76
9.3.3 OpenVPN . . . 76
9.3.4 WireGuard . . . 76
9.4 Android . . . 76
9.4.1 PPTP . . . 76
9.4.2 IPSec . . . 77
9.4.3 OpenVPN . . . 77
9.4.4 WireGuard . . . 77

10 Doporučené užití jednotlivých protokolů 77

Závěr 79

Conclusions 81

A Obsah elektronických dat 83

Literatura 84

Seznam obrázků

1 Virtuální okruh zdroj: https://forum.huawei.com . . . 12
2 Remote-Access VPN zdroj: https://tp-link.com . . . 13
3 Site-to-Site VPN zdroj: https://www.techtutsonline.com/ . . . 14
4 Hardwarový firewall s funkcí VPN zdroj: https://www.fortinate.com 15
5 Obecný diagram MFA - Multifaktorové autentizace . . . 17
6 Protokolový zásobník - Referenční model ISO/OSI . . . 19
7 Zapouzdření - síťový model TCP/IP . . . 20
8 Srovnání síťových modelů ISO/OSI a TCP/IP . . . 21
9 GRE paket . . . 25
10 PPTP paket . . . 26
11 L2TP Paket v kombinaci s IPSec . . . 28
12 L2TP Paket bez kombinace s IPSec . . . 28
13 Šifrování na různých vrstvách ISO/OSI síťového modelu . . . 29
14 IPSec paket - transportní a tunelovací mód . . . 30
15 OpenVPN paket . . . 32
16 WireGuard paket . . . 32
17 PPTP - připojení klienta . . . 35
18 Přidání VPN profilu . . . 42
19 Linux - Konfigurace VPN profilu . . . 43
20 Přidání certifikátu . . . 44
21 Volba cesty k certifikátu . . . 44
22 Windows - Konfigurace VPN profilu . . . 45
23 Graf odezvy od CloudFlare DNS . . . 54
24 Graf propustnosti mezi VPN serverem a klientem . . . 56
25 Graf nečinnosti CPU - 5 klientů, 25 Mbit/s . . . 58
26 Graf průměrné utilizace RAM - 5 klientů, 100 Mbit/s . . . 61
27 Graf průměru ztrátovosti paketů - 1 klient . . . 68
28 Graf průměru ztrátovosti paketů - 3 klienti . . . 69
29 Graf průměru ztrátovosti paketů - 5 klientů . . . 70

Seznam tabulek

1 PPTP CPU utilizace . . . 57
2 IPSec CPU utilizace . . . 59
3 OpenVPN CPU utilizace . . . 59
4 WireGuard CPU utilizace . . . 60
5 Celkové srovnání CPU náročnosti . . . 60
6 Celkové srovnání vytížení RAM . . . 62
7 PPTP - Jitter (ms) - 1 klient . . . 63
8 PPTP - Jitter (ms) - 3 klienti . . . 63
9 PPTP - Jitter (ms) - 5 klientů . . . 63
10 IPSec - Jitter (ms) - 1 klient . . . 64

11 IPSec - Jitter (ms) - 3 klienti . . . 64
12 IPSec - Jitter (ms) - 5 klientů . . . 65
13 OpenVPN - Jitter (ms) - 1 klient . . . 65
14 OpenVPN - Jitter (ms) - 3 klienti . . . 66
15 OpenVPN - Jitter (ms) - 5 klientů . . . 66
16 WireGuard - Jitter (ms) - 1 klient . . . 67
17 WireGuard - Jitter (ms) - 3 klienti . . . 67
18 WireGuard - Jitter (ms) - 5 klientů . . . 67
19 Výkonnost slabších klientských zařízení . . . 74

Seznam zdrojových kódů

1 Instalace pptpd daemona a balíčku net-tools . . . 33
2 Konfigurace /etc/pptpd.conf . . . 33
3 Konfigurace /etc/ppp/pptpd-options . . . 34
4 Konfigurace /etc/ppp/chap-secrets . . . 34
5 Konfigurace /etc/sysctl.d/30-ipforward.conf . . . 34
6 Forwardovací pravidlo iptables . . . 34
7 Načtení aktuální konfigurace . . . 35
8 Spuštění pptpd služby . . . 35
9 Služba poslouchá na portu 1723 na všech rozhraních . . . 35
10 Instalace strongswan a přidružených balíčků . . . 36
11 Vytvoření adresářů, přiřazení práv . . . 36
12 Generování kořenového klíče . . . 36
13 Generování CA certifikátu a podepsání vygenerovaným klíčem . . . 37
14 Generování privátního klíče VPN serveru . . . 37
15 Generování serverového certifikátu . . . 37
16 Kopírování souborů do adresáře /etc/ipsec.d/ . . . 37
17 Konfigurační soubor /etc/ipsec.conf . . . 38
18 Konfigurační soubor /etc/ipsec.secrets . . . 39
19 Konfigurační soubor /etc/ipsec.secrets . . . 39
20 Povolení portů SSH, 500, 4500 . . . 39
21 Povolení ufw . . . 39
22 Zobrazení výchozího rozhraní . . . 40
23 Část konfigurační soubor /etc/ufw/before.rules . . . 40
24 Konfigurační soubor /etc/ufw/sysctl.conf . . . 41
25 Kopírování CA certifikátu na klientský systém . . . 41
26 Instalace openvpn a easy-rsa . . . 45
27 Kopírování easy-rsa souborů . . . 46
28 Editace souboru /etc/openvpn/vars . . . 46
29 Inicializace PKI . . . 46
30 Vytvoření CA certifikátu a klíče . . . 46
31 Vytvoření privátního klíče serveru . . . 46
32 Vytvoření serverového certifikátu . . . 46

33 Vytvoření DH klíče a ta klíče . . . 47
34 Kopírování certifikátů a klíčů . . . 47
35 Konfigurační soubor /etc/openvpn/server.conf . . . 47
36 Povolení IPv4 forwarding . . . 48
37 Načtení aktuální konfigurace . . . 48
38 Konfigurační soubor /etc/default/ufw . . . 48
39 Konfigurační soubor /etc/ufw/before.rules . . . 48
40 Povolení portu 1194, aktualizace konfigurace . . . 48
41 Konfigurační soubor /etc/default/ufw . . . 49
42 Vytvoření klientského kliče . . . 49
43 Vytvoření klientského certifikátu . . . 49
44 Kopírování certifikátu a klíčů . . . 49
45 Kopírování certifikátu a klíčů na klientskou stanici . . . 49
46 Konfigurační soubor /etc/openvpn/client.conf . . . 49
47 Připojení k VPN serveru . . . 50
48 Instalace WireGuardu . . . 50
49 Vytvoření soukromého klíče /etc/wireguard/private.key . . . 50
50 Změna oprávnění k souboru /etc/wireguard/private.key . . . 50
51 Vytvoření veřejného klíče /etc/wireguard/public.key . . . 51
52 Konfigurační soubor serveru /etc/wireguard/wg0.conf . . . 51
53 Konfigurační soubor /etc/sysctl.conf . . . 51
54 Načtení aktualní konfigurace . . . 51
55 Povolení portů . . . 51
56 Konfigurační soubor serveru /etc/wireguard/wg0.conf . . . 51
57 Vytvoření soukrokého klíče - klient /etc/wireguard/private.key . . . 52
58 Změna oprávnění k souboru - klient /etc/wireguard/private.key . . . 52
59 Vytvoření veřejného klíče - klient /etc/wireguard/public.key . . . 52
60 Konfigurační soubor /etc/wireguard/wg0-client.conf . . . 52
61 Konfigurační soubor /etc/wireguard/wg0.conf . . . 52
62 Spuštění serveru . . . 53
63 Spuštění klienta . . . 53
64 Měření odezvy . . . 54
65 Povolení portu 5001 na firewallu . . . 55
66 Měření propustnosti VPN linky - server . . . 55
67 Měření propustnosti VPN linky - klient . . . 55
68 Server očekává příchozí UDP spojení po dobu 100 sekund . . . 56
69 Měření zatížení CPU po dobu 60 sekund . . . 56
70 Spuštění měření na straně klientů . . . 57
71 Měření stabibility řešení - strana serveru . . . 62
72 Měření stabibility řešení - strana klienta . . . 62

1 Úvod

Technologie jako takové se neustále vyvíjejí a obecně lze říci, že jsou již po nějakou dobu nedílnou součástí života téměř každého z nás. S rozvojem informačních technologií a stále častějším využíváním *home office*, respektive práce z domova, je žádané, aby zaměstnanci měli zajištěný bezpečný přístup k datům a informacím, nacházejícím se ve firemním intranetu. Stále častější také nastává situace, kdy určitá společnost má své pobočky v různých městech či dokonce státech a k tomu aby mohli její zaměstnanci spolu efektivně komunikovat a spolupracovat, je na místě tyto místa logicky propojit. Zároveň jistá míra anomymity a zajištění bezpečnosti při komunikaci v Internetu je častým důvodem pro použití technologie virtuálních privátních sítí. Kvůli těmto a mnoha dalším je právě technologie VPN (Virtual Private Network) často užívána.

Nasazení VPN se může lišit v závislosti na účelu, který má plnit. K tomuto se využívá množina protokolů, které se liší ve způsobu konfigurace, technických možnostech či úrovni zabezpečení.

Bakalářská práce se zabývá porovnáním vybraných VPN protokolů, jejich doporučenému užití a celkovému zhodnocení z hlediska datové propustnosti, bezpečnosti a hardwarové náročnosti.

V první kapitole se práce zabývá obecnou definicí VPN, popisuje typy, rozdělení VPN a rovněž představuje jednotlivé bezpečnostní prvky, které tato technologie užívá.

Následující tři kapitoly jsou věnovány síťovým modelům ISO/OSI a TCP/IP, na základě kterých jsou VPN protokoly nasazovány. Rovněž podrobnému představení jednotlivých protokolů a jejich rozdělení dle síťového modelu ISO/OSI.

Předmětem šesté s sedmé kapitoly je samotné nasazení jednotlivých protokolů, provedení testů z hlediska odezvy, datové propustnosti, stability a hardwarové náročnosti. Zhodnocena je také celková bezpečnost řešení a náročnost konfigurace.

Předposlední kapitola se věnuje testování datové propustnosti jednotlivých protokolů na slabších zařízeních, zejména na mobilních telefonech a jednodeskovém zařízení Raspberry Pi. Zároveň je zhodnoceno zda tato varianta je vhodná pro produkční prostředí.

Poslední dvě kapitoly se věnuji dostupnosti klientů pro nejpoužívanější platformy a rovněž uvádí doporučené užití jednotlivých VPN protokolů.

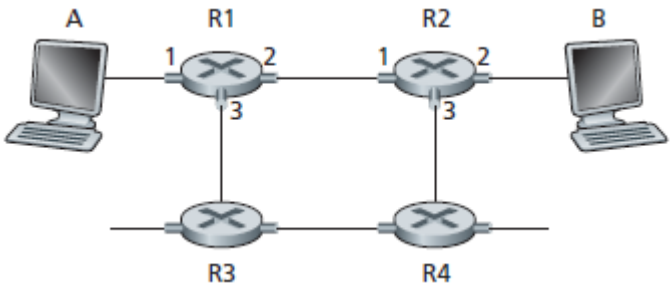
2 Virtuální privátní sítě

V následujících kapitolách bude popsána historie VPN technologie, její obecné rozdělení dle funkcionality, fyzického charakteru a rovněž budou popsány bezpečnostní prvky, které se s používáním VPN spojují.

2.1 Historie

Před příchodem Internetu, společnosti, které měly své pobočky ve vzájemně vzdálených lokalitách a zároveň chtěly, aby byla možná komunikace mezi nimi, musely vytvářet velmi rozsáhlé WAN (Wide Area Network) sítě tím, že propojovaly své lokální LAN (Local Area Network) sítě. Tohoto bylo dosaženo pronajmutím privátních linek. Těmito dříve bývaly například T1/T3 linky, které se lišily zejména přenosovou rychlostí. K přenosu bylo užíváno protokolů ATM (Asynchronous Transfer Mode), X.25 či jeho nástupce Frame Relay. Takto vytvořené sítě byly považovány za privátní.

Technologie, které implementují sítě VPN existují již poměrně dlouhou dobu. Původ můžeme nalézt ve virtuálních okruzích[1].



Obrázek 1: Virtuální okruh
zdroj: https://forum.huawei.com

Ve virtuálním okruhu jsou zdroje rezervovány pro časový interval přenosu dat mezi dvěma uzly. Tato síť je vysoce spolehlivým přenosovým médiem. Implementace virtuálních okruhů je nákladná, proto se v dnešní době příliš nevyužívá.

Základní pointou virtuálního okruhu je vytovření logického spojení mezi dvěma koncovými body, *zdroj* a *destinace*. K vytvoření takového spojení bývá často užito mnoha skoků mezi směřovači. Po ustanovení spojení mezi oběma body vzniká logické spojení, které se svou funkcí neliší od přímého propojení užitím fyzických portů. Takového logického spojení mohou již využívat konkrétní aplikace ke své komunikaci.

Fyzické propojení LAN sítí a vytvoření privátních sítí WAN je v dnešní době převážně nahrazováno užitím Internetu, pomocí kterého je umožněno logické propojení dvou koncových bodů. Tato varianta je snazší na nasazení a také výrazně levnější, co se týká finančních zdrojů potřebných pro realizaci.

2.2 Definice

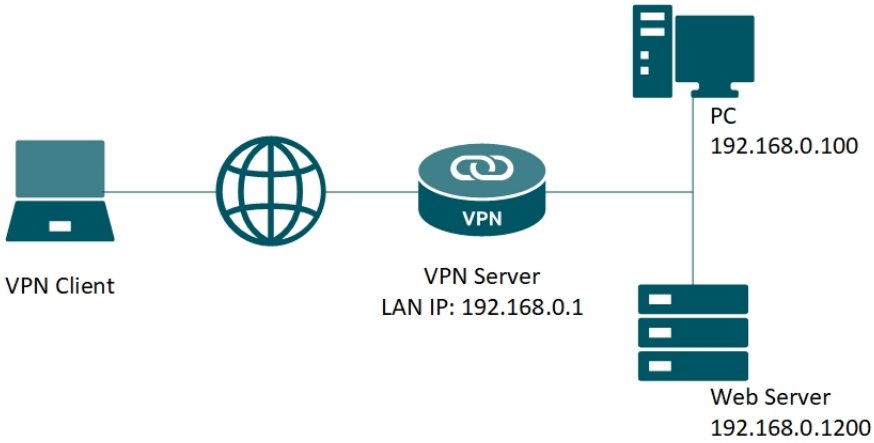
Technologii virtuálních privátních sítí lze chápat jako abstraktní rozšíření privátní sítě LAN, kdy za použití veřejné sítě, například sítě Internet, lze bezpečně přistupovat k datům či službám, které se nachází ve vzdálené LAN síti. K dosažení výše zmíněného je užito tunelovacích protokolů, které umožňují bezpečné,

šifrované spojení mezi klientem a serverem, či serverem a serverem, kdy veškerý či vybraný datový tok prochází zabezpečeným tunelem.

2.3 Obecné rozdělení

Kategorizovat VPN řešení lze více způsoby, obecně můžeme VPN dle užití rozdělit do dvou kategorii:

První možností užití VPN je připojení do vzdálené privátní sítě. Tohoto je užíváno zejména ke vzdálenému přístupu do firemní sítě, kdy po připojení je možno užívat prostředků, které vzdálená síť nabízí, např. přístup k firemním souborům či aplikacím, které jsou dostupné pouze z vnitřní sítě podniku. Tuto metodu vzdáleného přístupu nazávýme **Remote Access.** Nedílnou součástí tohoto typu VPN lze uvést užití *VPN jako služba*, kdy tímto pojmem rozumíme službu připojení do privátní sítě poskytovatele služby, za účelem vstupování do Internetu pod rozdílnou IP adresou, než je IP adresa přidelená od ISP (Internet Service Provider), což může za určitých podmínek zvyšovat jistou míru anonymity při užívání Internetu. Dalším důvodem proč používat VPN může být zašifrování provozu při používání veřejných síti, čímž rovněž dochází ke zvýšení celkové bezpečnosti, jelikož poskytovatel veřejného připojení či ostatní připojení klienti, nevidí, kam přistupujete. Důvodem pro užití VPN jako služby může být také obejití regionálních omezení dané země, kde jisté weby a služby nemusí být dostupné[2]. Příklad remote-access VPN lze vidět na Obrázku 2.



Obrázek 2: Remote-Access VPN
zdroj: https://tp-link.com

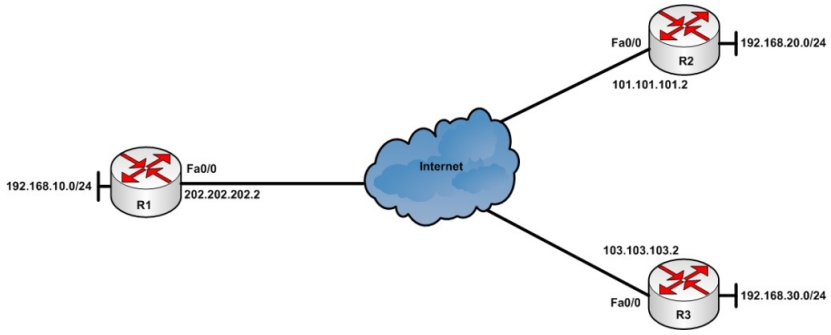
Druhou možnost užití VPN je takzvaná **Site-to-Site.** V tomto případě dochází ke připojení dvou nebo více firemních privátních sítí. Důvodem k tomuto je zejména umožnění vzájemné spolupráce firemním zaměstnancům, kdy fyzické pobočky mohou ležet v jiných městech, státech či kontinentech, ale veškerá, či

vybraná firemní data, jsou dostupná z kterékoliv z nich. Příklad Site-to-Site lze vidět na Obrázku 3. Variantu Site-to-Site lze dále dělit po dvou podkategorií:

- **Intranet** - intranetem lze uvažovat interní síť společnosti. Přístup do takové sítě je umožněn výhradně zaměstnancům společnosti. Často se v intranetu nacházejí vnitropodniková data, která ze své podstaty nejsou určena k dalšímu šíření mimo tuto síť.

- **Extranet** - síť uvnitř intranetu, do které je umožněn přístup externím společnostem či organizacím. Bývají zde zejména přístupná data generována v intranetu, se kterými externí společnosti mohou dále pracovat. Na extranet se lze dívat jako na určité API (Application Programming Interface), jelikož svou funkcionalitou je tomu velmi blízká.

.



Obrázek 3: Site-to-Site VPN
zdroj: https://www.techtutsonline.com/

2.4 Fyzické rozdělení

Mimo obecného rozdělení, tedy na Remote Access či Site-to-Site, můžeme dále dělit VPN dle fyzického stavu - Hardware a Software.

2.4.1 Hardware

Hardware VPN si můžeme představit jako samostatné zařízení, které poskytuje přístup do privátní sítě z veřejného Internetu. Toto zařízení má dedikovaný procesor a stará se o veškeré funkce VPN, jako je například autentizace, autorizace či šifrování. Hardwarové řešení lze obecně považovat za méně flexibilní, než řešení softwarové, zejména z hlediska konfiguračních možností, které nabízí a podstatně horší škálovatelnosti, což v tomto případě znamená zakoupení nového či dalšího zařízení. Hardwarová řešení se doporučují jako vhodná zejména ve velkých společnostech, kdy v těchto případech se o jejich konfiguraci a obsluhu starají specializované týmy pracovníků[3]. Hardwarová řešení se také často spojují s poskytováním vyšší úrovně zabezpečení, než řešení softwarová. Ve prospěch řešení pomocí

specializovaného hardwaru mluví také obecně vyšší přenosová rychlost[4]. Příkladem hardwarového VPN řešení bývají, mimo specializovaných zařízení, také routery či firewally s funkcí a podporou VPN. Příkladem takového zařízení může být produkční řešení od společnosti Fortinet, které nabízí, jak funkci firewallu, tak funkci VPN přístupového bodu. Zařízení lze vidět na Obrázku 4.

Obrázek 4: Hardwarový firewall s funkcí VPN
zdroj: https://www.fortinate.com

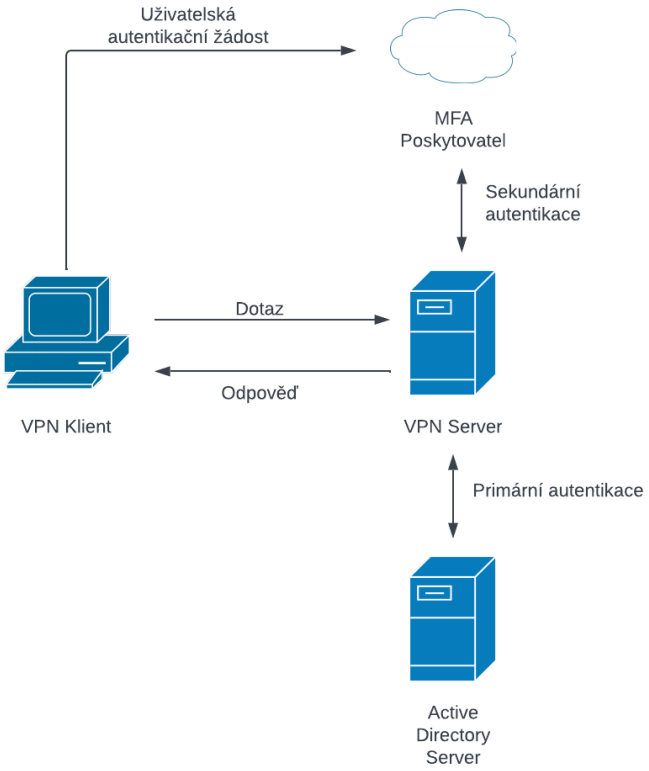
2.4.2 Software

Softwarové řešení poskytuje v mnoha směrech stejnou funkcionalitu jako řešení hardwarové. Jako hlavní benefit užití této varianty lze uvést poměrně velké množství softwarových řešení, kdy tyto se liší použitými tunelovacími protokoly, s tím související činností na různých úrovních ISO/OSI síťového modelu, úrovní zabezpečení a náročnosti samotné konfigurace. Škálovatelnost v tomto případě je mnohem snazší, jelikož v mnoha případech stačí změna konfigurace. Jako nejznámější řešení v této kategorii lze uvést například OpenVPN, WireGuard či řešení pomocí technologií IPSec. Samotná řešení budou podrobně popsána v kapitole 5. - Tunelovací protokoly.

2.5 Bezpečnostní prvky

Jelikož při užití VPN technologie je zpravidla užito veřejné sítě Internet, kterou obecně považujeme za nezabezpečenou, komunikaci mezi serverem a klientem či serverem a serverem, je nutné zabezpečit. Zabezpečení VPN se zajišťuje na několika úrovních:

- **Autentizace** - proces ověření, zda někdo nebo něco je opravdu tím, za koho se vydává. Autentizační technologie poskytují systém říženého přístupu na základě kontroly správnosti zadaných údajů, zejména ID a hesla. Metoda autentizace, která vyžaduje k udělení přistupu pouze přidělené ID a heslo se nazývá SFA (Single factor authentication). Dále je možnost užívat autentizační metodu 2FA (Two factor authentication), kdy tato metoda zvyšuje celkové zabezpečení, jelikož k udělení přístupu vyžaduje další bezpečnostní prvek (biometrický údaj, přístupový kód ...) Zejména v sektoru, kde dochází k práci s citlivými informacemi a daty se velmi často užívá autentikační metody MFA (Multifactor authentication), kdy k udělení přístupu je užito více než dvou ověřovacích prvků (například ID / heslo, přístupový kod a biometrický údaj)[5].



Obrázek 5: Obecný diagram MFA - Multifaktorové autentizace

- **Autorizace** – autorizací lze rozumět přidělování uživatelského přístupu ke specifickým zdrojím a funkcím. Jako vhodný příklad lze uvést oprávnění stáhnout soubor na sdíleném médiu, ke kterému mají přistup pouze zaměstnanci z daného oddělení v rámci společnosti. Autorizace může rovněž pracovat s různými úrovněmi oprávnění ve smyslu práci se soubory a adresáři - oprávnění Read, Write, Delete, Create, Execute či s přístupovými právy závislými na denním času, kdy například nebude umožněno nikomu přistoupit k citlivým informacím mimo stanovenou pracovní dobu. Soubor pravidel přístupu se obecně nazývá AC (Access Control) či ACL (Access Control List).

- **Důvěrnost** – důvěrnost v komunikaci lze chápat jako zabezpečení proti odposlouchávání datové komunikace, kdy k dosažení důvěrnosti se užívá

šifrování přenášených dat. K tomuto je užito šifrovacích metod, které se liší zejména v úrovni samotného zabezpečení.

- **Integrita** – zaručení, že během datového přenosu nebylo manipulováno s daty třetí, neautorizovanou stranu, či nebyla data modifikována chybou v průběhu datového přenosu. K tomuto jsou užívány hashovací algoritmy nebo digitální podpisy. Jako zástupce nejznámějších a zároveň nejpoužívanějších metod lze uvést různé verze MD (Message Digest) či SHA (Secure Hash Function).

- **Auditování** – kontrola a záznam činností, které jsou prováděny v průběhu datové komunikace. S tímto je spjato také logování událostí. Se zaznamenanými logy lze dále pracovat a hledat v nich příčinu určitého problému. Auditování může probíhat několika způsoby - po výskytu podezřelého chování či záznamu, periodicky v průběhu dne či v reálném čase.[1]

3 Síťové modely

Komunikační subsystém je složitá část hardwaru a softwaru. První pokusy o implementaci softwaru pro takové subsystémy byly založeny na jediném, složitém, nestrukturovaném programu s mnoha vzájemně se ovlivňujícími komponentami[6]. Výsledný software bylo velmi obtížné testovat a upravovat. K překonání tohoto problému vyvinula organizace ISO (International Organization for Standardization) *vrstevný přístup*. Ve vrstveném přístupu je koncept sítě rozdělen do několika vrstev a každé vrstvě je přiřazen konkrétní úkol. Mimo koncept vrstvení je také užit koncept *zapouzdření*. Oba tyto koncepty budou dále podrobněji popsány.

Nejpoužívanější síťové modely užívající obou zmiňovaných konceptů, jsou - model ISO/OSI a model TCP/IP.

3.1 Princip vrstvení

Metodický přístup k problému pomáhá zredukovat jeho celkovou komplexnost. Nezáleží tak ani na náročnosti samotného problému, metodický přístup může pomoci i při řešení jednoduchých úkolů, jako například při vaření oblíbeného jídla, ale zároveň i při podstatně komplexnějším úkolu, jako je ku příkladu vývoj a implementace účetního softwaru, který budou užívat stovky zaměstnanců společnosti z různých oddělení. Mimo jiných odvětví, kde se tento přístup užívá, je právě oblast sítí, potažmo VPN, jedno z technologických odvětví, který je na tomto principu přímo založeno.

V oblasti sítí se zavádí pojem *protokolový zásobník*[7]. Příklad protokolového zasobníku, konkrétně protokolového zásobníku referenčního síťového modelu OSI lze vidět na Obrázku 6.

Pojem protokolový zásobník můžeme chápat jako souhrn všech užitých vrstev a jejich funkcionalit, kterých daná architektura využívá. Velmi důležitým

pravidlem při implementaci tohoto principu je neumožnění jednotlivým vrstvám komunikovat s ostatními, mimo vrstvy sousední. Tímto se pro jednotlivé vrsty skryje nadbytečná komplexnost a jednotlivé vrstvy nejsou funkčně závislé na ostatních. Takové omezení komunikace je velmi výhodné, jednak z důvody zpřehlednění celé datové komunikace, tak z důvodu možnosti modifikace jednotlivých vrstev, bez toho aniž bychom funkčně jakkoliv narušili vrstvu jinou.

Princip vrstvení implementují oba zmíněné síťové modely. Konkrétní implementace jsou uvedeny v kapitole 3.3 ISO/OSI model a kapitole 3.4 TCP/IP model.

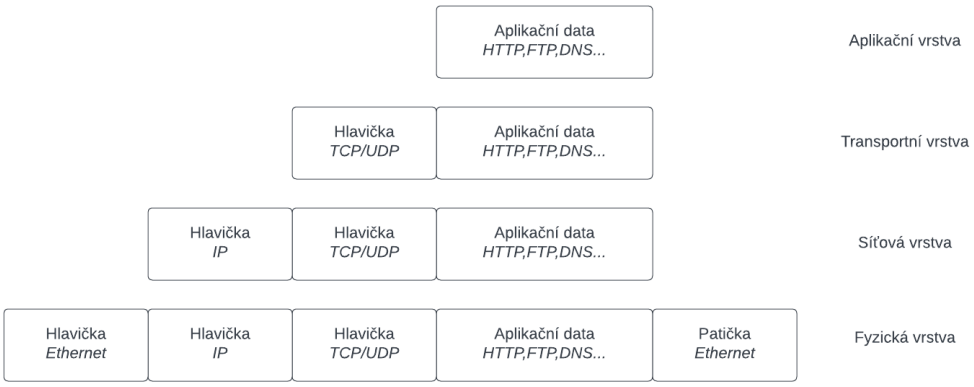
Obrázek 6: Protokolový zásobník - Referenční model ISO/OSI

3.2 Princip zapouzdržení

Jak bude dále popsáno, pojem zapouzdření je pro VPN a *tunelování* zásadní. Samotná funkce tunelování bude podrobněji popsána v kapitole 5 - Tunelovací protokoly.

Pokud se detailně podíváme na síťový provoz dle implementace TCP/IP či ISO/OSI, zjisíme, že každá vrstva v tomto modelu při své činnosti přidává k datům svou hlavičku, případně i patičku[8]. Zároveň při přechodu mezi jednotlivými vrstvami, každá vrstva zapouzdřuje data, která přišla z vrstvy předchozí, viz. obrázek 7.

Datová komunikace vždy probíhá směrem od nejvyšších vrstev k vrstvám nižším, následně ve směru opačném.



Obrázek 7: Zapouzdření - síťový model TCP/IP

3.3 OSI model

V průběhu let začaly být LAN sítě čím dál více oblíbené, což mělo za následek snahu o propojení těchto sítí, kdy častým problémem bylo užití rozdílných standardů, což v konečném důsledku velmi komplikovalo samotnou propojitelnost či ji zcela znemožňovalo.

V devadesátých letech dvacátého stolení se vláda USA, v té době dominantní hráč na poli zákazníků odebírající technologické produkty, rozhodla, že bude odebírat jen takové produkty a služby, které podporují a drží se modelu OSI, čímž značně podpořila snahu společností o dodržováním tohoto standardu.

Obecným cílem modelu OSI bylo poskytnutí společného konceptu pro vývoj standardů[9], které umožní propojení různých systémů a sítí.

Ačkoliv zcela funkční implementace modelu OSI existuje, jeho funkcí je zejména stanovení jistých pravidel, které by měly být dodržovány při vývoji systémů a zařízení využívající síťovou komunikaci.

Referenční model OSI lze vidět na Obrázku 6.

3.4 TCP/IP model

S ohledem na význam obou referenčních modelů, je model OSI pouze koncepčním modelem, sloužící především k popisu, diskuzi a pochopení jednotlivých síťových funkcí. TCP/IP byl však primárně navržen k řešení specifické sady reálných problémů. Model OSI je obecný, na protokolu nezávislý, přesto jej většina protokolů a systémů dodržuje, zatímco model TCP/IP je založen na standardních protokolech, které byly vyvinuty v souvislosti s rozvojem Internetu.

Narozdíl od modelu OSI, model TCP/IP nemá strukturu sedmi vrstev, ale pouze čtyř, kdy fyzická a linková vrstva je spojena v jednu - *vrstvu síťového rozhraní*. Třetí, síťová vrstva OSI se v modelu TCP/IP nazývá *internetová vrstva*. Čtvrtá, transportní vrstva, je identická v obou modelech a následné 3 vrstvy OSI modelu - relační, prezentační a aplikační je nazývána jako *aplikační*. Srovnání obou síťových modelů lze vidět na Obrázku 8.

Obrázek 8: Srovnání síťových modelů ISO/OSI a TCP/IP

4 Síťový model OSI

V této kapitole budou popsány veškeré vrstvy síťového modelu ISO/OSI, zároveň budou uvedeny příklady protokolů, které na jednotlivých vrstvách pracují.

4.1 Fyzická vrstva

Tato vrstva zahrnuje fyzické komponenty zapojené do přenosu dat, jako jsou kabely, konektory, síťové rozhraní či jiná přenosová média. Na této vrstvě se data převádí do *bitového toku*[10], což je řetězec jedniček a nul. Fyzická vrstva obou zařízení musí také souhlasit s konvencí signálu, aby bylo možné rozlišit jedničky od nul na obou koncových zařízeních. Propojující médium na této vrstvě může být měďený kabel, optický kabel, radiový signál, mikrovlný signál či jiný. K definici komponent fyzické vrstvy může být užito mnoha specifikací. Příkladem může být specifikace RJ-45, která definuje definuje tvar konektoru, počet užitých drátů a specifikuje piny, které mají být užity. Dalšími specifikacemi užitými ve fyzické vrstvě mohou být Ethernet, 802.3 či 802.5. Model OSI nijak nerozlišuje konkrétní přenosové médium.

4.2 Linková vrstva

Linková vrstva zajišťuje přenos dat z uzlů na uzel – spojení mezi dvěma přímo propojenými uzly. Jejím úkolem je zapouzdření dat do rámců. Definuje protokol pro navázání a ukončení spojení mezi dvěma fyzicky připojenými zařízeními, jako je například protokol PPP (Point-to-Point Protocol). Linková vrstva se obecně dělí na dvě podvrstvy – vrstvu řízení přístupu k médiím (MAC) a vrstvu řízení logického spoje (LLC). Vrstva MAC je zodpovědná za řízení toho, jak zařízení v síti získávají přístup k médiu a oprávnění k přenosu dat. Vrstva LLC je zodpovědná za identifikaci a zapouzdření protokolů síťové vrstvy a řídí kontrolu chyb a synchronizaci rámců. Jako konkrétní protokoly pracující na linkové vrstvě lze uvést již zmíněný PPP protokol, FDDI (Fiber Distributed Data Interface) či IEEE 802.3 (Ethernet).

4.3 Síťová vrstva

Síťová vrstva se stará o směrování paketů pomocí funkcí logického adresování a přepínání. Síť je médium, ke kterému lze připojit mnoho uzlů, kdy každý uzel má svoji přiřazenou adresu. Přenášená data v síťové vrstvě jsou nazývány *pakety* či *datagramy*, v závislosti na tom, jestli se jedná o spojení TCP (Transmission Control Protocol) či UDP (User Datagram Protocol). Jako nejpoužívanější protokol na síťové vrstvě můžeme uvažovat IP (Internet Protocol) či BGP (Border Gateway Protocol).

4.4 Transportní vrstva

Transportní vrstva poskytuje datový přenost *end-to-end*. Tímto rozumíme zajištění kompletní konektivity mezi dvěmi koncovými zařízeními. Přenos dat na této vrstvě může být orientovaný na spojení (connection oriented) nebo bez spojení (connectionless). Data na této vrstvě nazýváme *segmenty*. Služby zajišťující spolehlivý datový přenos jsou transparentní pro vyšší vrstvy OSI modelu. Obecně se jedná o služby:

- **Řízení toku dat** - spravuje datový přenos mezi zařízeními. Zajišťuje také, aby nedošlo k přehlcení zařízení, které data přijímá.
- **Multiplexing** - proces, ve kterém je více datových toků (v jednu chvíli může komunikovat více aplikací) kombinováno do jednoho. Cílem je co možná nejefektivnější využití daného přenosového média.
- **Kontorola chyb** - kontrola přenášených dat.
- **Oprava** - oprava chybně přenesených dat. Forma opravy může být například žádost o znovu zaslání ztacených segmentů.

Nejčastěji užívanými protokoly transportní vrstvy jsou TCP, který je orientovaný na spojení a protokol UDP, který je bez spojení.

4.5 Relační vrstva

Relační vrstva má za úkol vytváření, správu a ukončení spojení mezi dvěma hosty. Zároveň umožňuje dvěma entitám, které pracují na vyšších vrstvách - prezentační či aplikační vrstvě, synchronizovat a spravovat jejich datový přenos. Jakmile je spojení ustanoveno, hosti můžou komunikovat jednosměrně (*half duplex*) nebo obousměrně (*full duplex*).

Tato vrstva je také zodpovědná za správu tokenů, prostřednictvím kterých zabraňuje dvěma uživatelům současně přistupovat nebo se pokoušet o stejnou kritickou operaci[11]. Umožňuje také synchronizaci tím, že spravjí proces přidávání kontrolních bodů, které jsou považovány za synchronizační body k datovému toku.

Jako příklad protokolů pracující v relační vrstvě lze zmínit RPC (Remote Procedure Call), SQL (Structured Query Language), NetBIOS či ASP (AppleTalk Session Protocol).

4.6 Prezentační vrstva

Prezentační vrstva se stará o to, aby data pocházející z aplikační vrstvy jednoho hosta, byly čitelné aplikační vrstvou hosta druhého, kterému jsou data směřovány. Prezentační vrstva je zodpovědná také za šifrování dat, kompresi a konverzi dat.

Příkladem protokolů pracující na této vrstvě mohou být GIF (Graphic Interchange Format), JPEG (Joint Photographic Experts Group), ASCII či HTML (HyperText Markup Language)

4.7 Aplikační vrstva

Aplikační vrstva je rozhraní, které aplikace používají k tomu, aby se dostaly ke zdrojům na síti. Příkladem samotné funkce může být užití spustitelného protokolu FTP či Telnet, který umožňuje přístup k síťovým prostředkům. Tato síťová vrstva komunikuje přimo s aplikačními procesy, které jsou nad samotným OSI modelem. FTP (File Transfer Protocol), HTTP (HypetText Transfer Protocol), SMTP (Simple Mail Transfer Protocol) či SNMP (Simple Network Management Protocol) jsou příklady protokolů pracující na aplikační vrstvě.

5 Tunelovací protokoly

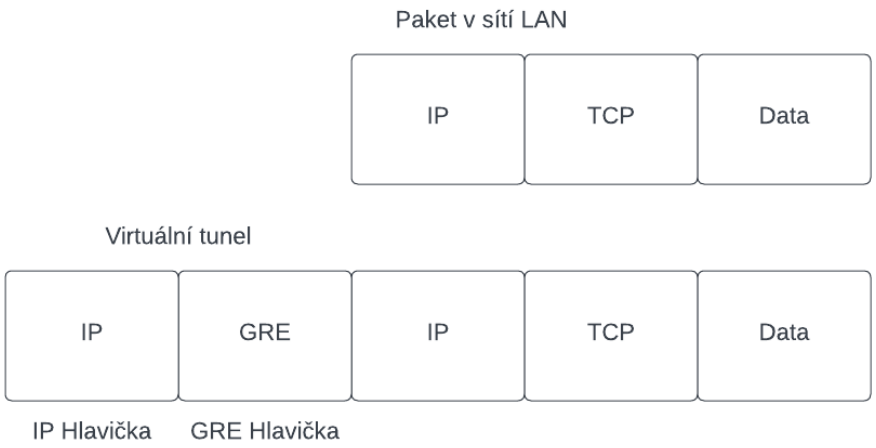
Tunelovací protokoly jsou síťové protokoly sloužící k vytvoření virtuálního spojení, tunelů, mezi dvěma uzly. Existuje možnost vytvoření tunelu a následné jeho využití k přenosu dat mezi uzly bez samotného šifrování, což ale nelze považovat za bezpečné VPN spojení, jelikož jsou soukromá data odesílána zcela nešifrovaně skrz privátní si veřejnou síť, tím pádem jednoduše čitelná. Pro zajištění bezpečného VPN spojení, jsou zasílaná data zapouzdřena, i zašifrována.

5.1 GRE

Protokol GRE (Generic Routing Encapsulation) byl vytvořen v roce 1994, specifikován v RFC (Request For Comments) 1701 a 1702. Vznikl na základě toho, že dřívější pokusy o vytvoření tunelů skrz Internet byly příliš specifické. Funguje na bázi zapouzdření paketů jednoho protokolu do protokolu druhého. K takto zapouzdřeným paketům je přidána hlavička GRE a hlavička cílové adresy, viz. Obrázek 9. V koncovém bodě tunelu je paket rozbalen a zpracován dle informací v původní IP hlavičce. Protokol GRE může zapouzdřit až 20 různých typů protokolů. Samotný protokol GRE nezajišťuje šifrování dat.

Samotné GRE tunelování obsahuje 3 typy protokolů:

- **Cestující protokol** - protokol, který je využit na lokální LAN.
- **Dopravující protokol** - samotný GRE protokol, jsou do něj zapouzdřena data z cestujícího protokolu.
- **Transportní protokol** - protokol, který je užit k transportu dat, např. IP (Internet Protocol). Jeho úkolem je zapouzdření dvou předchozích protokolů.



Obrázek 9: GRE paket

5.2 PPTP

Protokol PPTP (Point-to-Point Tunneling Protocol) byl vyvinut skupinou PPTP Forum, která měla za členy společnosti Ascend Communication, Microsoft Corporation, U.S. Robotics, 3Com, ECI Telematics a Copper Mountain Networks. PPTP je protokol, který pracuje na druhé, linkové vrstvě OSI modelu. Jeho funkce spočívá v zapouzdření PPP rámců do modifikované hlavičky GRE a vložení do IP paketu. Konkrétní strukturu paketu je možné vidět na Obrázku 10. PPTP využívá protokol TCP (Transmission Control Protocol) na portu 1723, z toho důvodu je provoz snadno detekovatelný a blokovatelný.

PPTP komunikace zahrnuje dva procesy, kdy pro spuštění následujícího, je nutná úspěšná kompletace předešlého procesu. Jedná se o procesy:

- **PPTP kontrolní spojení** - využívá připojení k Internetu ustanovaného protokolem PPP. Dále vytváří tunel mezi PPTP klientem a PPTP serverem, nazývaného *PPTP tunel*.

- **PPTP tunelování dat** - PPTP protokol vytvoří IP datagram, který obsahuje zašifrované PPP pakety, které jsou zaslané skrz PPTP tunel na PPTP server. Následně PPTP server IP datagram rozbalí, rozšifruje PPP pakety a směřuje je dále v privátní LAN.

Obrázek 10: PPTP paket

Společnost Microsoft Corporation měla na vývoj protokolu PPTP podstatně největší vliv, rovněž byla také byla první, která protokol v širokém měřítku implementovala. Protokol je z toho důvodu nativně integrován v OS Windows již od roku 1996, ačkoliv RFC 2637 dokumentace byla vydána až v roce 1999.

Samotný PPTP protokol využívá protokolu PPP k autentizaci. Využívá k tomu protokoly PAP (Password Authentication Protocol), CHAP (Challenge Handshake Authentication Protocol), MS-CHAP (Microsoft Handshake Authentication Protocol). Data v PPP rámcích jsou šifrována za pomocí šifrovacích alrogitmů RC4 a DES (Digital Encryption Standard). PPTP podporuje šifrování do 128-bitů.

Navzdory svému stáří a bezpečnostním zranitelnostem[12] se PPTP stále používá v některých síťových řešeních – většinou interních podnikových VPN. Výhoda protokolu PPTP spočívá v tom, že se snadno konfiguruje, má vysokou přenosovou rychlost a také z důvodu nativní integrace ve většině platforem. K jeho funkci není potřeba užití jakéhokoliv dodatečného softwaru, k připojení je potřeba mít pouze přihlašovací údaje a IP adresu či doménový název serveru.

5.3 L2F

L2F (Layer 2 Forwarding) je tunelovací protokol vyvinut společností Cisco, který používá virtuální sítě pro bezpečný přenos datových paketů. Funkčnost L2F je velmi podobná protokolu PPTP. L2F je součástí standardu Layer 2 Tunneling Protocol (L2TP) pod RFC 2661.

L2F jakkoliv nemodifikuje PPP rámce nebo rámce vyšších úrovní OSI, pouze poskytuje možnost přenesení rámce, u kterého je povolena *point-to-point* konektivita[13]. Ačkoliv je z převážné části využíváno přenosu přes TCP/IP sítě, je možno užít protokolu L2F i u jiných síťových infrastruktur, jako jsou například frame relay nebo virtuální okruhy X.25.

V roce 1999 Microsoft a Cisco sloučily své příslušné verze protokolu L2F a vytvořily L2TP.

Základní služby poskytované protokolem L2F:

- **Tunelování** - poskytuje možnost definování virtuálních tunelů mezi dvěma koncovými body za pomocí protokolu PPP, který data zapouzdří.
- **Autentizace** - protokol L2F je z velké části závislý na implementaci protokolu PPP a jeho implementaci protokolu PAP a protokolu CHAP, které slouží k autentizaci.
- **Podpora více protokolů** - umožňuje přenášet protokoly vyšších vrstev síťových modelů, které nelze jinak přes IP sítě přenést. Příkladem mohou být protokoly IPX/SPX, AppleTalk či NetBEUI.
- **Datová integrita** - L2F podporuje základní kontrolní součty.
- **Ochrana proti falšování** - základní ochrana je poskytnuta na základě výpočtu hodnoty klíčů při sestavování tunelu.

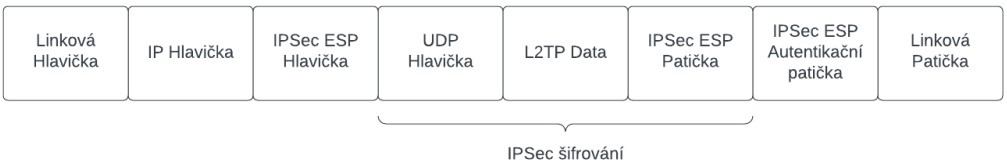
Ačkoliv konkurenční protokol PPTP byl a stále je velmi rozšířený, zároveň podporován velkou řadou zařízení, výhoda protokolu L2F optoti konkurečnímu protokolu PPTP spočívá v tom, že nevyžaduje síť postavou na IP, taktéž nevyžaduje žádnou konkrétního infrastrukturu a umožňuje běžet na jakékoliv technologii na Linkové vrstvě. K tomu poskytuje vyšší zabezpečení, namísto základního ID a hesla[14] .

5.4 L2TP

L2TP (Layer2 Tunneling Protocol) je kombinací protokolu PPTP a L2F. L2TP podporuje jakýkoli směrovaný protokol, jako jsou IP, IPX či AppleTalk. L2TP lze použít jako tunelovací protokol přes Internet nebo privátní intranet. Společnost IETF (Internet Engineering Task Force) spojila protokoly PPTP a L2F do jednoho společného standardu v RFC 2661. V roce 2005 byla představena nová verze L2TP - L2TPv3 s dalšími bezpečnostními funkcemi, vylepšeným zapouzdřením a schopností přenášet datová spojení po síti. Specifikace byla popsána v RFC 3931[15].

Funkce protokolu spočítá v tom, celý L2TP paket, data a L2TP hlavička, je odeslána pomocí protokolu UDP na portu 1701. Data jsou přenášena v L2TP tunelu, zapouzdřena v PPP rámcích. L2TP sám o sobě nepodporuje autentizaci či šifrování, proto se velmi často se používá společně s protokolem IPsec (IP Security) pro zajištění důvěrnosti, ověřování a integrity. Protokol IPSec bude podrobněji představen v kapitole 5.5 - IPSec.

Kombinace těchto dvou protokolů je obecně známá jako L2TP/IPsec. Spojení protokolů L2TP s IPSec je znázorněn na Obrázku 11.



Obrázek 11: L2TP Paket v kombinaci s IPSec

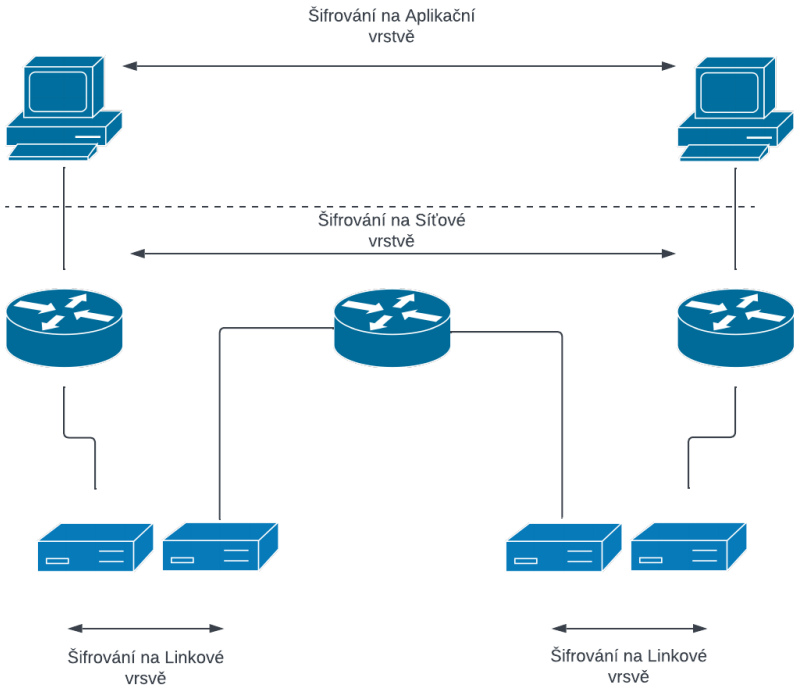
Protokol L2TP, stejně jako konkurenční protokol PPTP, pracuje na druhé vrstvě modelu OSI - linkové. Princip funkce protokolu L2TP je založen na modelu klient/server. Dva koncové body tunelu L2TP se nazývají LAC (Síťový koncentrátor) a LNS (Síťový server). LAC se nachází mezi LNS a vzdáleným systémem a přeposílá pakety na vzdálený server. Jakmile je vytvořen virtuální tunel mezi systémy, je síťový provoz zasílán oběma směry.

Zasílané pakety jsou dvou druhů - *kontrolní paket* a *datový paket*. Kontrolní pakety se zasílají sekvenčně, tudíž se považují za spolehlívé. U datových paketů kontrola doručení neprobíhá.

Obrázek 12: L2TP Paket bez kombinace s IPSec

5.5 IPSec

IPSec (IP Security) zahrnuje sadu protokolů, které jsou vytvořeny k poskytování zabezpečení na třetí vrstvě OSI - síťové. Oba již zmiňované protokoly, PPTP i L2TP, pracují na druhé vrstvě síťového modelu - linkové. Obrázek 13 znázorňuje funci šifrování na linkové vrstvě, síťové vrstvě a vrstvě nejvyšší - aplikační.



Obrázek 13: Šifrování na různých vrstvách ISO/OSI síťového modelu

V rámci protokolového zásobníku ISO/OSI, je síťová vrstva tou nejnižší, která může poskytnout *end-to-end* zabezpečení, což znamená, že může poskytnout ochranu pro jakákoliv data vyšší vrstvy síťového modelu zapouzdřenou v IP datagramu, bez nutnosti jakkoliv modifinovat danou aplikaci.

Základ IPSec tvoří 3 základní mechanismy:

- **AH** - (The Authentication Header Protocol) zajišťuje integritu přenášených dat pomocí hashovací funkce a sdíleného klíče. AH garantuje integritu přenášených dat, zároveň nezajišťuje šifrování dat. Podporuje *tunelovací mód* a *transportní mód*.
- **ESP** - (The Encapsulating Security Payload Protocol) je užíván k šifrování, autentizaci a k zajištění integrity dat. Stejně jako AH, tak i ESP podporuje oba přenosové módy.
- **IKE** - (Internet Key Exhange Protocol) - klíčovou rolí IKE je vyjednávání SA (Security Associations). SA jsou bezpečnostní zásady definované pro komunikaci mezi dvěma nebo více koncovými body. Při pokusu o vytvoření VPN tunelu nebo připojení, obě strany používají sadu algoritmů a vzájemně dohodnutých klíčů.

  Existují dva IKE standardy - IKE, který býl definován v RFC 2409 a novější, IKEv2, který byl definován v RFC 7296. IKE nejčastěji k autentizaci používá certifikáty infrastruktury veřejného klíče PKI (Public Key

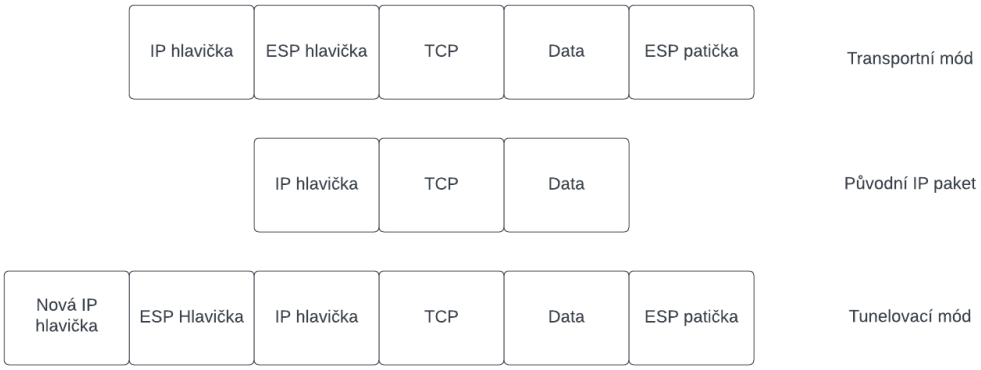
Infrastructure) X.509. K vytvoření sdílené relace bývá užit protokol pro výměnu klíčů - DH (Diffie-Hellman key exchange).

Jak již bylo uvedeno, pro oba AH i ESP protokoly platí, že mohou pracovat ve dvou módech - *tunelovacím* a *transportním*.

- **Tunelovací mód** - celý původní IP paket je zapouzdřen. Původnímu IP paketu je přidána nová IP hlavička, AH/ESP hlavička a v případě užití ESP i ESP patička. Tunelovací mód lze považovat obecně za bezpečnější, jelikož je celý původní paket šifrován. Nevýhodou je nutnost užití vyššího výpočetního výkonu z důvodu zapouzdření a šifrování celého původního paketu. Tunelovací mod je v IPSec nastaven jako výchozí.

- **Transportní mód** - v transportním módu jsou šifrována pouze data, původní hlavička IP paketu zůstává nezměněna. Přidány jsou pouze hlavičky ESP/AH.

Na Obrázku 14 je možné vidět příklady paketu užívající tunelovacího a transportního módu.



Obrázek 14: IPSec paket - transportní a tunelovací mód

Protokoly AH a ESP potřebují pro svou činnost mít definované SA. Před začátkem komunikace probíhá vyjednávání mezi oběma hosty ohledně užití vhodného protokolu, klíče a parametrů, které budou užity. Samotné vyjednávání SA má dvě fáze.

V první fázi dochází ke vzájemné autentikaci obou stran. Užito je buď Certifikátu nebo PKI a dochází k vytvoření ISAKMP SA (Internet Security Association and Key Management Protocol). K autentizaci je užito DH (Diffieho-Hellmanova výměna klíčů), kdy každá strana vygeneruje svůj privátní klič a veřejný klič. Následně zašle protistraně svůj veřejný klíč. Každá strana vygeneruje nový klíč kombinací svého privátního klíče a veřejného klíče protistrany. Tímto vzniknou dva identické klíče, které jsou použity pro šifrování komunikace

Pro úspěšné zahájení druhé fáze je nutné, aby byla první fáze úspěšně ukončena. Poté, co si obě strany sestaví tunel pomocí první fáze, pokračují fází druhou, ve které vyjednají konkrétní SA. Podobně jako u první fáze, si účastníci vyměňují návrhy, aby určili, které bezpečnostní parametry použít v SA. Návrh druhé fáze také zahrnuje bezpečnostní protokol – buď ESP nebo AH – a vybrané šifrovací a ověřovací algoritmy. Návrh může také specifikovat skupinu DH, pokud je požadováno PFS (Perfect Forward Secrecy).

Protokol IPsec je velmi komplexní a nabízí mnoho možností konfigurace. S tímto se také pojí náročnější konfigurace a samoné zprovoznění. S vyšší komplexnosti jsou také spojeny vyšší nároky na výpočetní výkon. IPsec je také snadno detekovatelný při použití transportního modu. IPSec je velmi oblíbené VPN řešení a při vhodné konfiguraci[16] a užití správně zvolených protokolů, se jedná o velmi bezpečné řešení, zajišťující rovněž vysokou datovou propustnost.

5.6 OpenVPN

OpenVPN je open-source VPN řešení a byl vytvořen v roce 2001 Jamesem Yonanem. Distribuován je pod GNU (General Public License). Je podporován mnoha operačními systémy - Windows, Linux, Mac OS X, FreeBSD, Solaris a dalšími.

Výchozí protokol a port pro OpenVPN je UDP a port 1194. Než IANA udělila OpenVPN oficiální přiřazení portu, starší klienti (2.0-beta16 a starší) užívali port 5000 jako výchozí.

OpenVPN podporuje zabezpečení SSL (Secure Socket Layer) / TLS (Transport Layer Security), ethernetové přemostění, přenos tunelu TCP nebo UDP přes proxy nebo NAT (Network Address Translation), podporu dynamických IP adres a DHCP. Nabízí rovněž vysokou škálovatelnost.[17].

Protokol je úzce spojen s knihovnou OpenSSL a odvozuje z ní většinu svých kryptografických schopností. Součástí knihovny OpenSSL je množina hashovacích algoritmů - MD4, MD5, MD2, SHA-1, SHA-2, SHA-3, MDC-2, BLAKE2, Whirpool, SM3 a mnoho dalších. Knihovla OpenSSL také obsahuje velký počet asymetrických algoritmů, jako například RSA, DSA, Diffie-Hellman, ED25519 a symetrických algoritmů - AES, DES, Blowfish, Camellia či ChaCha-Poly1305.

OpenVPN podporuje konvenční šifrování pomocí předem sdíleného tajného klíče (režim statického klíče) nebo zabezpečení veřejného klíče (režim SSL/TLS), pomocí klientských a serverových certifikátů. OpenVPN také podporuje nešifrované tunely TCP/UDP.

OpenVPN pracuje s dvěmi typy rozhraní - TAP a TUN.

- **TUN - network tunnel** - pracuje na třetí vrstvě modelu ISO/OSI. Využívá ke své funkcí výhradně protokol IP.

- **TAP - network tap** - pracuje na druhé vrstvě modelu ISO/OSI. TAP, narozdíl od rozhraní TUN, pracuje s ethernet rámci a umožňuje přenos i jiných protokolů, než je protokol IP.

Zajímavostí je také to, že samotný programový kod VPN neběží v jádře operačního systému, jako je to například u řešení IPSec, ale v uživatelském modu.

Samotnou strukturu OpenVPN paketu lze vidět na Obrázku 15.

Obrázek 15: OpenVPN paket

5.7 WireGuard

WireGuard je open-source komunikační protokol. Protokol WireGuard obsahuje podstatně méně řádků kódu než ostatní VPN řešení. OpenVPN společně s OpenSSL má asi 600 000 řádek kodu, zatímco celý WireGuard má jen 4000 řádků. To znamená, že je lépe auditovatelný, má méně problematických míst a je zároveň výkonnější.

WireGuard používá výhradně protokol UDP. Podporuje IPv6 v tunelu i mimo něj. Pracuje pouze na třetí vrstvě ISO/OSI, jak pro IPv4, tak pro IPv6.

Zdrojový kod WireGuardu je psán s ohledem na bezpečnost. Používá moderní kryptografii jako protokol Noise a algoritmy Curve25519, ChaCha20, Poly1305, BLAKE2, SipHash24 a HKDF.

Stejně jako IPSec, WireGuard je implementován přimo v jádře operačního systému, což znamená, že samotné VPN řešení je velmi rychlé. K ustanovení spojení je potřeba výměna IP adres a veřejných klíčů mezi protistranami. Aktuálně nejnovější verze - 1.0.20220627.

Strukturu WireGuard paketu je možné vidět na Obrázku 16.

Obrázek 16: WireGuard paket

6 Nasazení zvolených protokolů

V praktické části byly nasazeny zvolené VPN protokoly. Zvoleny byly protokoly PPTP, IPSec, OpenVPN a WireGuard. Protokol PPTP byl zvolen jako zástupce starších protokolů, zároveň z důvodu snadné konfigurace a možnosti nasazení na velkém množství zařízení, který tento protokol podporují. IPSec byl zvolen

z důvodu, že se jedná o velmi rozšířené řešení v soukromé i produkční sféře. OpenVPN je rovněž velmi oblíbenou volbou v produkčním prostředí, zejména z důvodu širokých konfiguračních možností. Posledním testovaným protokolem bude WireGuard, a to z jednak z důvodu, že se jedná o velmi mladý protokol, který se v posledních letech dostává do popředí, zejména díky své rychlosti, stabilitě a snadné konfiguraci.

K testování bylo užito architektury klient-server. Jako server sloužil virtualizovaný server na platformě Proxmox[18] ve verzi 7.3.3. Hypervizor běžel na CPU AMD Ryzen 9 3900X, měl k dispozici 32GB operační paměti RAM a 16TB pevné paměti.

Pro každé VPN řešení byl vytvořen samostatný virtuální stroj. Každý systém měl přiřazeno 4x vCPU, 4GB RAM a 32GB pevné paměti. Tato specifikace byla zvolena z důvodu, že by nemělo docházet v omezování funkcí VPN z důvodu nedostatku výpočetního výkonu.

Jako operační systém pro dané virtuální stroje byl zvolen Ubuntu Server ve verzi 22.04 LTS, zejména z důvodu velmi častého užití v soukromé i produkční sféře, dobře zpracované dokumentaci a vysoké stabilitě.

Jako klient byl primárně užit operační systém Ubuntu 22.04 LTS, zároveň ale jsou uvedeny i příklady připojení k VPN serveru na operačním systému Windows 11.

6.1 PPTP

Pro nasazení PPTP je užito programu pptpd, což je PPTP VPN daemon. Jedná se o nejpoužívanější open-source řešení pro operační systémy Linux.

```
# apt install pptpd net-tools
```

Zdrojový kód 1: Instalace pptpd daemona a balíčku net-tools

První konfigurační soubor, která je nutné upravit je */etc/pptpd.conf*, kde se nachází nastavení rozsahu IP pro lokální a vzdálenou adresaci. V tomto případě bude zvolena lokální IP adresa 10.0.0.1 a rozsah vzdálených adres, které budou užívat klienti - 10.0.0.10 - 10.0.0.20. Subnet v konfiguraci není uváděn.

```
localip 10.0.0.1
remoteip 10.0.0.10-20
```

Zdrojový kód 2: Konfigurace /etc/pptpd.conf

Po instalaci je nutné upravit konfigurační soubor */etc/ppp/pptpd-options*. Atributem *name* rozumíme název lokálního systému. Tento atribut je užit při autentikaci. Následně volíme jaké DNS servery (Domain Name System) budou užity. V tomto případě jsou zvoleny primární DNS servery Cloudflare a Google - 1.1.1.1 a 8.8.8.8. Tyto byly zvoleny zejména kvůli rychlosti a vysoké dostupnosti. Jako DNS servery lze použít i jiné veřejné DNS servery, DNS vašeho ISP či vlastní. Server bude konfigurován tak, aby přijímal autentizaci MS-CHAPV2 a šifrování MPPE128. Jiná autentizace bude serverem odmítnuta.

```
name pptpd
refuse-pap
refuse-chap
refuse-mschap
require-mschap-v2
require-mppe-128
ms-dns 1.1.1.1
ms-dns 8.8.8.8
```

Zdrojový kód 3: Konfigurace /etc/ppp/pptpd-options

Aby se klienti mohli připojit na VPN sever, je potřeba editovat soubor */etc/ppp/chap-secrets*. Sloupec *client* stanovuje uživatelské jméno, *server* určuje název serveru, *secret* nastavuje heslo uživatele. Poslední sloupec, *IP addresses*, stanovuje z jakých adres bude možnost na VPN server přistupovat. Pro jednoduchost bude zvolen znak *, který povoluje přístup ze všech IP adres.

```
# Secrets for authentication using CHAP
# client server secret IP addresses
vpnuser pptpd   heslo123        *
```

Zdrojový kód 4: Konfigurace /etc/ppp/chap-secrets

K tomu aby klienti mohli využívat přístup do Internetu či do vzdálené LAN sítě, je nutné povolit *IP přeposílání paketů*. Bude vytvořen nový soubor */etc/sysctl.d/30-ipforward.conf*. Tímto bude docíleno, že pakety, které přijdou od klientů, budou přeposílány dle pravidel *iptables*. Pravidla iptables budou upravena v následujícím kroku.

```
# Enable IPv4 packet forwarding
net.ipv4.ip_forward=1
```

Zdrojový kód 5: Konfigurace /etc/sysctl.d/30-ipforward.conf

Dále je nutné upravit pravidla *iptables*, aby docházelo k přeposílání paketů na jiné rozhraní, ze kterého je umožněn přístup do vnitřní LAN či Internetu. V tomto případě se jedná o volbu tabulky *nat*, přidání pravidla *POSTROUTING* na odchozí rozhraní *ens18*. Bude užito *MASQUERADE*, což umožňuje klientům vystupovat do Internetu či vnitřní LAN pod adresou serveru, čímž skryjí svou přiřaženou IP adresu, pokud jim byla přiřazena.

```
# iptables -t nat -A POSTROUTING -o ens18 -j
  MASQUERADE
```

Zdrojový kód 6: Forwardovací pravidlo iptables

Po dokončení konfigurace je nutné, aby se načetly nové konfigurace v adresáři */etc/sysctl.d/*

```
# sysctl --system
```

Zdrojový kód 7: Načtení aktuální konfigurace

Spuštění služby pptpd.service, která se stará o obsluhu pptpd daemona.

```
# systemctl start pptpd.service
```

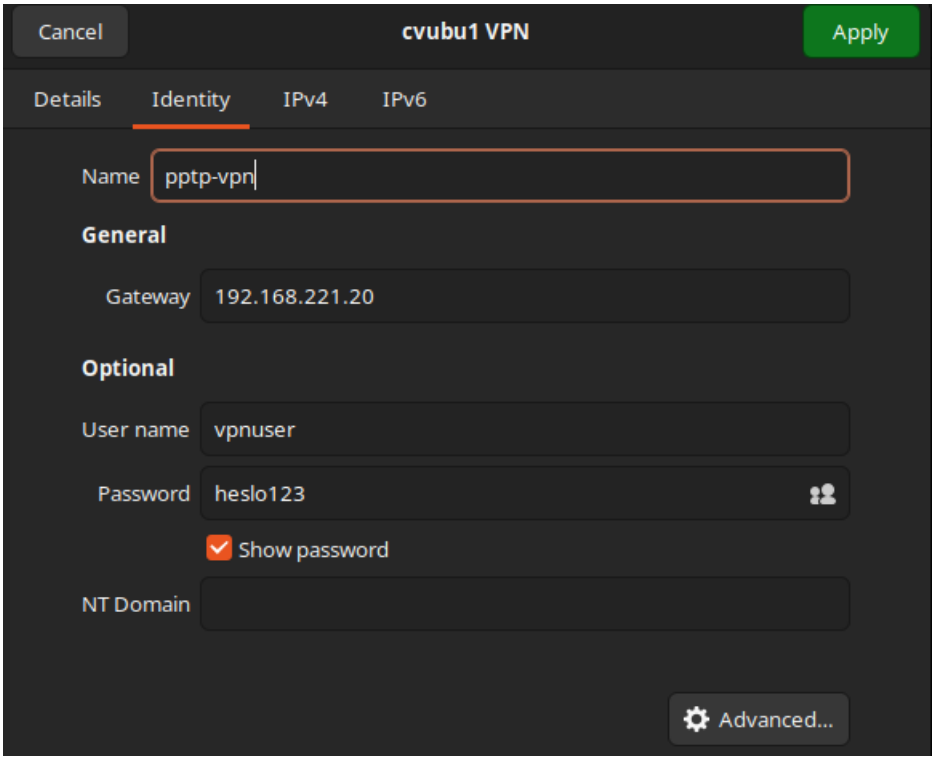
Zdrojový kód 8: Spuštění pptpd služby

Ověření, že server poslouchá na všech rozhraních na portu 1723.

```
# netstat -lpn | grep 1723
tcp 0    0 0.0.0.0:1723     0.0.0.0:*    LISTEN
   2221/pptpd
```

Zdrojový kód 9: Služba poslouchá na portu 1723 na všech rozhraních

Nyní je možné se k VPN serveru připojit, vyplněním IP adresy serveru, přihlašovacího jména a hesla 17.



Obrázek 17: PPTP - připojení klienta

6.2 IPSec

V případě konfigurace IPSec VPN je na výběr několik možných implementací. Jedná se o zejména *Openswan*, *Libreswan*, *Strongswan* či *ipsec-tools*. Vzhledem k tomu, že server běží na operačním systému Ubuntu Server, pro kterého je v základu užito řešení pomocí strongswan, bude užito právě tohoto. Dalším důvodem je velmi přehledná dokumentace a silná podpora i v jiných linuxových distribucích. Bude užita verze 5.9.5, která je standardně podporována v operačním systému Ubuntu Server 22.04.

Prvním krokem je instalace *strongswan* a balíčků potřebných pro jeho správnou funkci. Balíček *libcharon-extauth-plugins* se používá k zajištění toho, aby se různí klienti mohli autentizovat k VPN serveru pomocí sdíleného uživatelského jména a hesla. Balíček *libstrongswan-extra-plugins* zajišťuje širší podporu šifer. Součástí je i instalace balíčku *net-tools*, který obsahuje mimo jiné program netstat[19], sloužící mimo jiné k ověření příchozích / odchozích spojení.

```
# apt install strongswan strongswan-pki
  libcharon-extra-plugins libcharon-extauth-
  plugins libstrongswan-extra-plugins libtss2-
  tcti-tabrmd0 net-tools
```

Zdrojový kód 10: Instalace strongswan a přidružených balíčků

Jelikož bude užito implementace IPSec/IKEv2, jednou z možností je vygenerování samotné CA (Certificate Authority). V ideálním případě by CA měl být samostatný server, který slouží pouze k generaci certifikátů a klíčů. V testovaném případě bude CA přímo VPN server. Další možností je užití self-signed certifikátu. Rovněž bude vygenerován serverový certifikát.

Prvním krokem je vytvoření adresářů, kde budou generované certifikáty / klíče ukládány a přiřazení vhodných přístupových práv adresáři. V tomto případě se jedná o právo 700, což znamená právo číst, zapisovat či spouštět obsah adresáře pouze pro vlastníka adresáře[20]. V pozdějších krocích budou tyto adresáře, včetně jejich obsahu, přesunuty do adresářů obsahujících konfigurační soubory IPSec.

```
# mkdir -p ~/pki/{cacerts,certs,private}
# chmod 700 ~/pki
```

Zdrojový kód 11: Vytvoření adresářů, přiřazení práv

Dalším krokem je vytvoření kořenového, 4096 bitového RSA klíče, kterým bude podepsán CA certifikát. Platnost certifikátu byla zvolena standardní - 10 let.

```
# ipsec pki --gen --type rsa --size 4096 --
  outform pem > ~/pki/private/ca-key.pem
```

Zdrojový kód 12: Generování kořenového klíče

```
# ipsec pki --self --ca --lifetime 3650 --in ~/
  pki/private/ca-key.pem \
    --type rsa --dn "CN=VPN root CA" --outform
      pem > ~/pki/cacerts/ca-cert.pem
```

Zdrojový kód 13: Generování CA certifikátu a podepsání vygenerovaným klíčem

Následuje generování 4096 bitového klíče pro VPN server, což umožní autentizaci serveru při připojování klientů.

```
# ipsec pki --gen --type rsa --size 4096 --
  outform pem > ~/pki/private/server-key.pem
```

Zdrojový kód 14: Generování privátního klíče VPN serveru

Dalším krokem je vytvoření serverového certifkátu. Tohoto dosáhneme užitím veřejného klíče VPN serveru a podepsání CA klíčem. Atribut *dn* obsahuje hodnotu vnitřní adresy VPN serveru. Atribut *san* obsahuje také vnitřní IP adresu serveru. Pokud využíváme službu DNS, je možno do atributu *san* vložit doménový název serveru místo IP adresy. DNS užito není, proto je zvolena adresa IP. Parametr *–flag serverAuth* značí, se tento certifikát bude užit výhradně pro autentizaci serveru, ještě před samotným vytvořením tunelu. Parametr *–flag ikeItermediate* je užit kvůli podpoře starších macOS klientů.

```
# ipsec pki --pub --in ~/pki/private/server-key
  .pem --type rsa \
    | ipsec pki --issue --lifetime 1825 \
        --cacert ~/pki/cacerts/ca-cert.pem \
        --cakey ~/pki/private/ca-key.pem \
        --dn "CN=10.12.13.0" --san 10.12.13.0 \
        --flag serverAuth --flag
          ikeIntermediate --outform pem \
    >  ~/pki/certs/server-cert.pem
```

Zdrojový kód 15: Generování serverového certifikátu

Následujícím krokem je překopírování veškerých vygenerovaných klíčů a certifikátů do adresáře IPSec, ze kterého *strongswan* načítá konfigurační soubory.

```
# cp -r ~/pki/* /etc/ipsec.d/
```

Zdrojový kód 16: Kopírování souborů do adresáře /etc/ipsec.d/

Následuje samotná konfigurace IPSec. Atribut *charondebug* nastavuje jednotlivé logovací úrovně ike, kernelu a užitým pluginům. Atribut *uniqueids* nastavuje

užití unikatních ID klientů při pokusu o připojení. Hodnota *no* říká, že je umožněno více klientům užívat stejné přihlašovací údaje. Hodnota *yes* by při pokusu o připojení užitím stejných přihlašovacích údajů, původního užívatele odpojila a umožnila připojení klienta nového[21].

Následná sekce upravuje konkrétní nastavení IKEv2 spojení. Atribut *auto=add* říká, že se samotné připojení má načíst, nikoliv spojit. Atribut *compress=no* říká daemonu, že nemá přijímat či navrhovat kompresi. Atribut *type=tunnel* říká, že se má užít tunelovací mód. Následující atributy specifikují užití typ klíčů užitých při autentizaci, nastavení fragmentace a umožnění zapouzdření ESP hlaviček do UDP rámce, což zajišťuje lepší průchod přes firewally. Atributy *dpdaction*, *dpddelay* a *rekey* nastavují chování tunelu při neaktivitě klienta, jeho odpojení a expiraci spojení. Následují *left* atributy, které specifikují z jaké IP adresy se lze na VPN server připojit, adresu VPN serveru, cestu k certifikátu serveru, vyžadování žádosti o certifikát od klienta a stanovení subnetu VPN serveru. Jedná se o atributy, které konfigurují konkrétní nastavení VPN serveru. *Right* atributy konfigurují nastavení klientů, kteří se k VPN serveru budou připojovat. V závěru konfiguračního souboru jsou uvedeny podporované šifrovací/autentizační metody užité v IKE a ESP.

```
config setup
    charondebug="ike 1, knl 1, cfg 0"
    uniqueids=no

conn ikev2-vpn
    auto=add
    compress=no
    type=tunnel
    keyexchange=ikev2
    fragmentation=yes
    forceencaps=yes
    dpdaction=clear
    dpddelay=300s
    rekey=no
    left=%any
    leftid=192.168.221.23
    leftcert=server-cert.pem
    leftsendcert=always
    leftsubnet=0.0.0.0/0
    right=%any
    rightid=%any
    rightauth=eap-mschapv2
    rightsourceip=10.12.13.0/24
    rightdns=1.1.1.1,8.8.8.8
    rightsendcert=never
    eap_identity=%identity
```

```
    ike=chacha20poly1305-sha512-curve25519-
      prfsha512,aes256gcm16-sha384-prfsha384-
      ecp384,aes256-sha1-modp1024,aes128-sha1-
      modp1024,3des-sha1-modp1024!
    esp=chacha20poly1305-sha512,aes256gcm16-
      ecp384,aes256-sha256,aes256-sha1,3des-
      sha1!
```

Zdrojový kód 17: Konfigurační soubor /etc/ipsec.conf

V tuto chvíli je VPN server nakonfigurován, aby přijímal klientské požadavky na připojení. Dále je nutné vytvořit přístupové údaje, které budou k připojení k serveru využívány[22]. Tohoto dosáhneme editací souboru */etc/ipsec.secrets*. Prvně přidáme řádek *: RSA "server-key.pem"*, kde specifikujeme typ privátního klíče a cestu k privátnímu klíči. Následně je nutné vložit řádek s atributy *user*, který specifukuje název uživatele, *EAP*, což definuje typ přihlašovacích údajů a heslo.

```
# This file holds shared secrets or RSA private
    keys for authentication.

# RSA private key for this host, authenticating
    it to any other host
# which knows the public part.
: RSA "server-key.pem"
user : EAP "heslo123"
```

Zdrojový kód 18: Konfigurační soubor /etc/ipsec.secrets

Po konfiguraci souborů /etc/ipsec.conf a /etc/ipsec.secrets restartujeme strongswan službu, čímž dojde k načtení aktuální konfigurace.

```
# systemctl restart strongswan-starter
```

Zdrojový kód 19: Konfigurační soubor /etc/ipsec.secrets

Následujícím krokem je konfigurace firewallu a iptables pravidel. Jako zástupce firewallu bude užit nativní firewall *ufw*. (Uncomplicated firewall)[23].

Přidáme pravidla k povolení přístupu pomocí protokolu SSH (*Secure shell*) a UDP komunikaci pomocí protokolů 500 a 4500, což jsou standartní porty užívané protokolem IPsec. Důležité je také povolit samotný *ufw*.

```
# ufw allow OpenSSH
# ufw allow 500,4500/UDP
```

Zdrojový kód 20: Povolení portů SSH, 500, 4500

```
# ufw enable
```

Zdrojový kód 21: Povolení ufw

Dalším krokem je nastavení pravidla pro přeposílání paketů, komunikaci za použití NAT (Network Address Translation), což umožní klientům komunikovat do Internetu. K tomuto slouží soubor */etc/ufw/before.rules*. V tomto souboru přidáme pravidla, která určují komunikaci v rámci NAT, což je konfigurováno v sekci **nat*. Dalším úpravou konfiguračního souboru je přidání pravidel pro maximální velikosti paketu, což je konfigurováno v sekci **mangle*. Posledním krokem je přidání pravidla pro povolení přeposílání ESP paketů na jiné rozhraní, což umožní připojení klientům na VPN server. Tato konfigurace je vytvořena v sekci **filter*. Tyto tři sekce by měly být v úvodu konfiguračního souboru v uvedeném pořadí.

Konfigurace IP adres v konfuguračním souboru /etc/ufw/before.rules by měla odpovídat výchozímu rozhraní.

```
# ip route show default
```

Zdrojový kód 22: Zobrazení výchozího rozhraní

```
#
# rules.before
#
# Rules that should be run before the ufw
   command line added rules. Custom
# rules should be added to one of these chains:
#   ufw-before-input
#   ufw-before-output
#   ufw-before-forward
#
*nat
-A POSTROUTING -s 10.12.13.0/24 -o ens18 -m
   policy --pol ipsec --dir out -j ACCEPT
-A POSTROUTING -s 10.12.13.0/24 -o ens18 -j
   MASQUERADE
COMMI

*mangle
-A FORWARD --match policy --pol ipsec --dir in
   -s 10.12.13.0/24 -o ens18 -p tcp -m tcp --tcp
   -flags SYN,RST SYN -m tcpmss --mss 1361:1536
   -j TCPMSS --set-mss 1360
COMMIT

# Don't delete these required lines, otherwise
   there will be errors
*filter
:ufw-before-input - [0:0]
:ufw-before-output - [0:0]
```

```
:ufw-before-forward - [0:0]
:ufw-not-local - [0:0]
# End required lines
-A ufw-before-forward --match policy --pol
   ipsec --dir in --proto esp -s 10.12.13.0/24 -
   j ACCEPT
-A ufw-before-forward --match policy --pol
   ipsec --dir out --proto esp -d 10.12.13.0/24
   -j ACCEPT
```

Zdrojový kód 23: Část konfigurační soubor /etc/ufw/before.rules

Předposledním krokem je přidání pravidla pro povolení přeposílání paketů mezi rozhraními odkomentováním řádku v konfiguračním souboru */etc/ufw/sysctl.conf*. Finálním krokem je odkomentování řádků, které blokují přijímat a odesílat přeposílané ICMP (Internet Control Message Protocol) pakety.

```
net/ipv4/ip_forward=1
net/ipv4/conf/all/accept_redirects=0
net/ipv4/conf/all/send_redirects=0
```

Zdrojový kód 24: Konfigurační soubor /etc/ufw/sysctl.conf

Samotné připojení je možné z různých operačních systémů, jako jsou různé distribuce Linux, Windows, macOS či Android. Připojení zde bude popsáno z klienta s operačním systémem Ubuntu 22.04 a klienta s operačním systémem Windows 11.

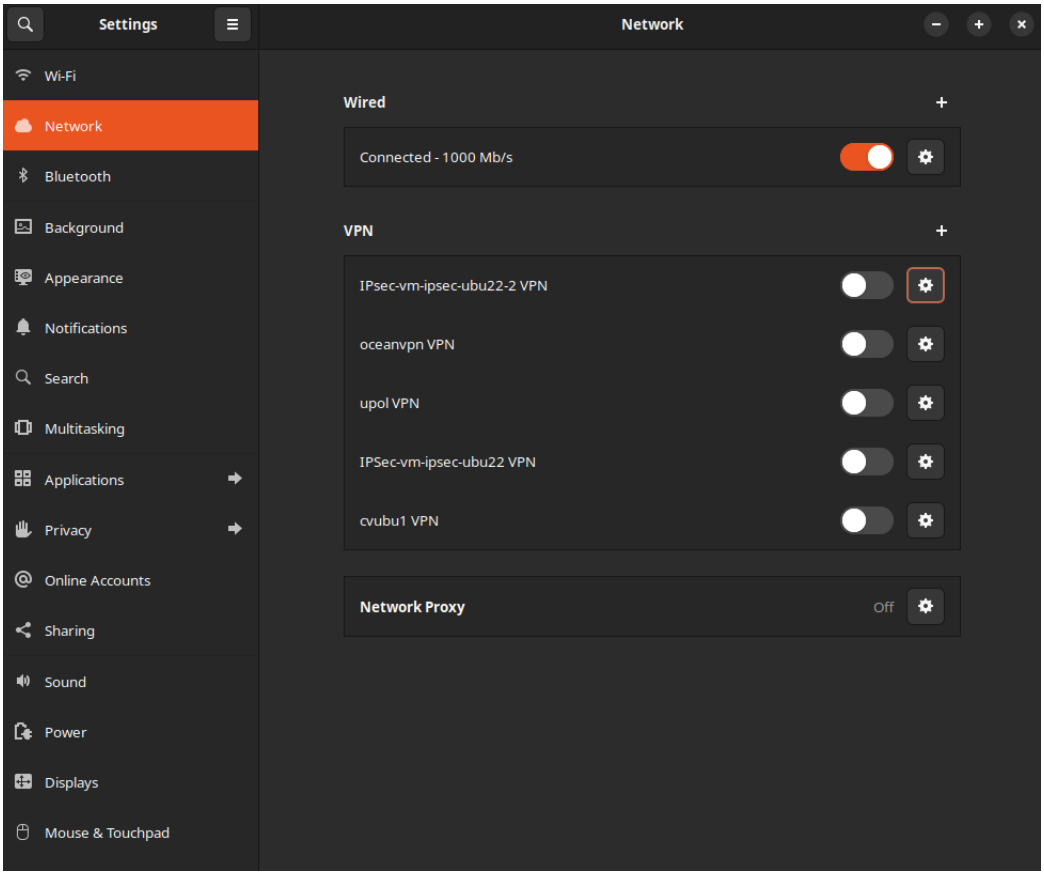
Před samotnou konfigurací klienta je nutné si zkopírovat certifikát CA - /etc/ipsec.d/cacerts/ca-cert.pem na klientský systém. Může být užito SCP (Open SSH Secure Copy), FTP (File Transfer Protocol), užití flash disku či jiného média. V tomto případě bude užito SCP, jak pro klienta OS Ubuntu, tak klienta OS Windows, zejména kvůli jednoduchosti a nativní podpoře u obou operačních systémů.

```
# scp 192.168.221.23:/home/dex/ca-cert.pem /
  home/dex/
```

Zdrojový kód 25: Kopírování CA certifikátu na klientský systém

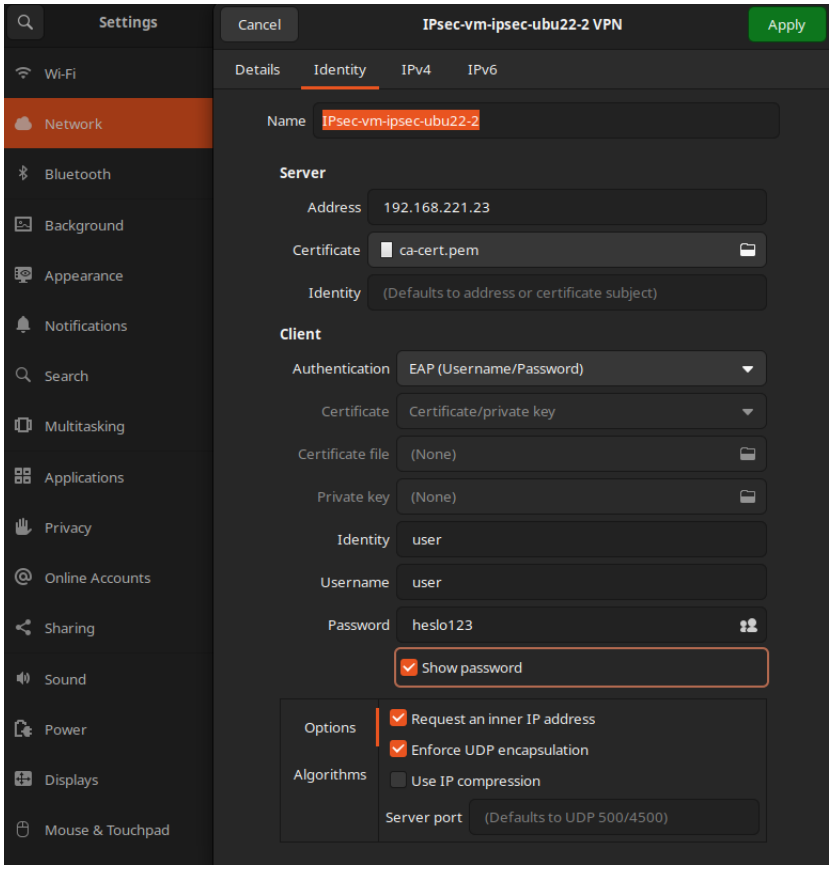
U klienta OS Ubuntu je užito výchozího desktop manažera - GNOME. Klient bude konfigurován následovně:

Settings > *Network* > tlačítko „+“ k přidání nového VPN profilu. Znázorněno na Obrázku 18.



Obrázek 18: Přidání VPN profilu

Dalším krokem je nutné pojmenování nově vytvořeného VPN profilu, zvolení cesty ke staženému CA cerifikátu, volba EAP jako typ přihlašovacích údajů, zadání uživatelského jména a hesla. Volitelně je také možnost zvolit přiřazení interní IP adresy a zapouzdření ESP do UDP paketu. Konkrétní konfiguraci lze vidět na Obrázku 19.



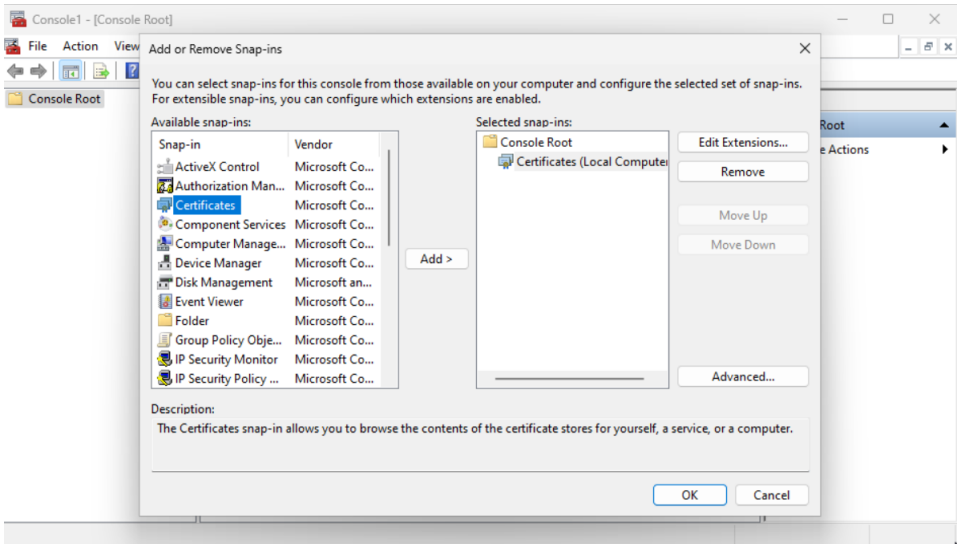
Obrázek 19: Linux - Konfigurace VPN profilu

Následně je umožněno klientu připojit se na VPN server.

U klienta Windows 11 bude bude užito následujícího postupu konfigurace:

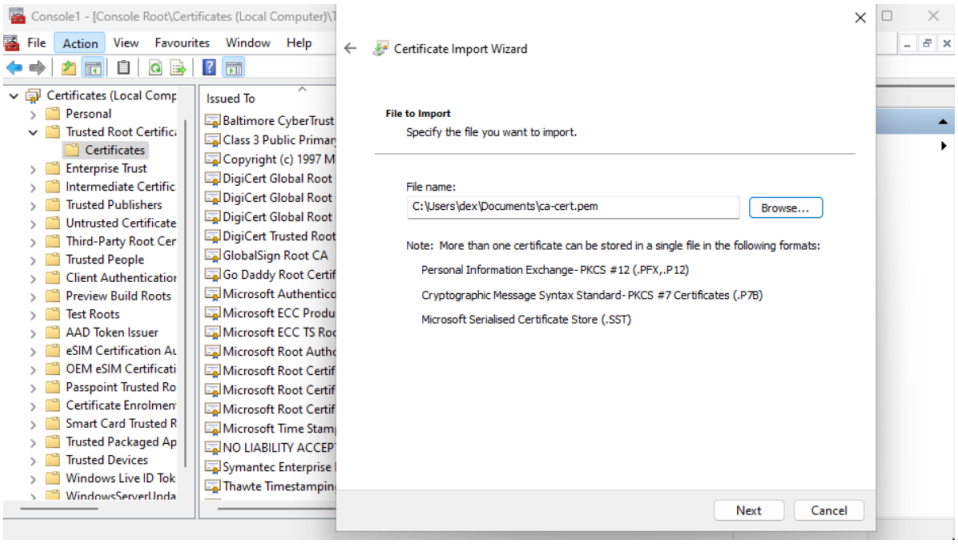
Prvním krokem je stisknutí klávesy *„winkey“* na klávesnici. Do vyhledávače poté zadat *„mmc.exe“*, čímž spustíme program *Microsoft Management Console*.

Dále zvolíme *File* > *Add or Remove Snap-in*, vybereme možnost *Certificates* a klikneme na *Add* 20.



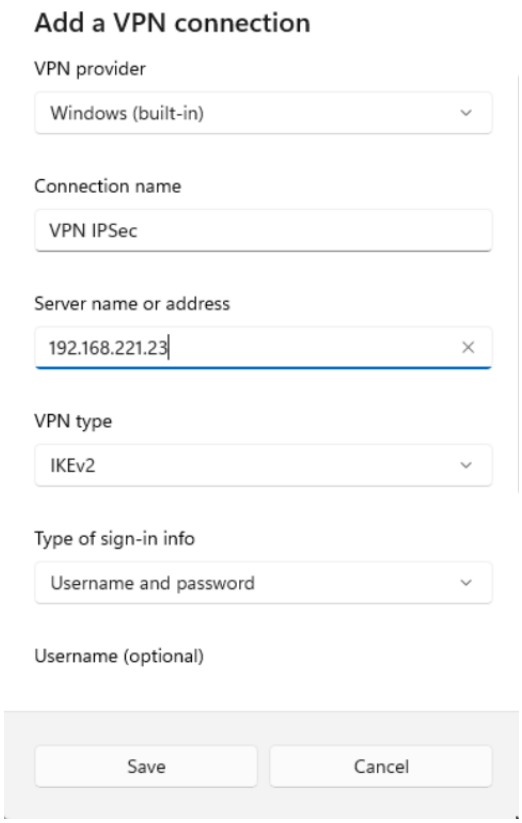
Obrázek 20: Přidání certifikátu

Poté vybereme možnost *Computer Account* a zvolíme *Next*. V dalším kroku vybereme *Local Computer* a následně *Finish*. V levém panelu rozklikneme položku *Trusted Root Certification Authorities* a zvolíme *Certificates*. V menu následně zvolíme *Action* > *All Tasks* > *Import*. Následně se objeví průvodce přidání cerfitikátu, kde se zvolí cesta ke staženému certifikátu ze serveru 21.



Obrázek 21: Volba cesty k certifikátu

Po úspěšném přidání je nutné vyplnit VPN konfiguraci na straně klienta. Znovu stistkneme klávesu *„winkey“* na klávesnici, ve vyhledávači vyhledáme *„VPN Settings“* a zvolíme tuto možnost. Na Obrázku 22 je znázorněno konkrétní nastavení.



Obrázek 22: Windows - Konfigurace VPN profilu

Nyní je možné se připojit na VPN server.

6.3 OpenVPN

Prvním krokem je instalace balíčku *openvpn* a také balíčku *easy-rsa*, který obsahuje sadu skriptů k obsluze PKI (Public Key Infrastructure). OpenVPN je užito ve verzi 2.5.5, easy-rsa ve verzi 3.0.8.

```
# apt install openvpn easy-rsa
```

Zdrojový kód 26: Instalace openvpn a easy-rsa

Následným krokek je vygenerování CA certifikátu serveru a klíče. Následuje zkopírování obsahu adresáře */usr/share/easy-rsa/* do adresáře */etc/openvpn/*. Soubor */etc/openvpn/vars.example* bude zkopírován do stejné cesty pod názvem *vars*. Soubor *vars* bude následně upraven.

```
# cp -r /usr/share/easy-rsa/ /etc/openvpn/
# cp -a /etc/openvpn/easy-rsa/vars.example /etc
  /openvpn/easy-rsa/vars
```

Zdrojový kód 27: Kopírování easy-rsa souborů

```
export KEY_COUNTRY="CZ"
export KEY_PROVINCE="CZ"
export KEY_CITY="Olomouc"
export KEY_ORG="UPOL"
export KEY_EMAIL="novada05@upol.cz"
export KEY_OU="OpenVPN"
```

Zdrojový kód 28: Editace souboru /etc/openvpn/vars

Dalším krokem je změna pracovního adresáře a inicializace PKI.

```
# cd /etc/openvpn/easy-rsa/
# ./easyrsa init-pki
```

Zdrojový kód 29: Inicializace PKI

Následuje vygenerování CA certifikátu a klíče.

```
# ./easyrsa build-ca
```

Zdrojový kód 30: Vytvoření CA certifikátu a klíče

Následně bude vygenerován privátní kliče serveru společně s certifikační žádostí. Tímto budou vygenerovány soubory:
/etc/openvpn/easy-rsa/pki/reqs/server.req,
/etc/openvpn/easy-rsa/pki/private/server.key

```
# ./easyrsa gen-req server nopass
```

Zdrojový kód 31: Vytvoření privátního klíče serveru

Jelikož je certifikační autoritou samotný VPN server, dalším krokem je podepsání certifikační žádosti a tím vytvoření serverového certifikátu. Tímto bude vygenerován certifikát */etc/openvpn/easy-rsa/pki/issued/server.crt*.

```
# ./easyrsa sign-req server server
```

Zdrojový kód 32: Vytvoření serverového certifikátu

Dalším krokem je vygenerování sdíleného DH klíče a *ta* klíče, potřebného k TLS spojení. Tímto budou vytvořeny soubory */etc/openvpn/easy-rsa/pki/dh.pem* a */etc/openvpn/easy-rsa/ta.key*.

```
# ./easyrsa gen-dh
# openvpn --genkey secret ta.key
```

Zdrojový kód 33: Vytvoření DH klíče a ta klíče

Následně překopírujeme vytvořené certifikáty a klíče do správné cesty s konfiguračními soubory OpenVPN.

```
# cp -a /etc/openvpn/easy-rsa/pki/issued/server
  .crt /etc/openvpn/
# cp -a /etc/openvpn/easy-rsa/pki/private/
  server.key /etc/openvpn/
# cp -a /etc/openvpn/easy-rsa/pki/dh.pem /etc/
  openvpn/
# cp -a /etc/openvpn/easy-rsa/pki/ca.crt /etc/
  openvpn/
# cp -a /etc/openvpn/easy-rsa/ta.key /etc/
  openvpn/
```

Zdrojový kód 34: Kopírování certifikátů a klíčů

Dalším krokem je samotná konfigurace VPN serveru. Vytvoří se soubor */etc/openvpn/server.conf*. Zde se nastaví, který port bude používán, zvolen bude protokol UDP a rozhraní TUN. Volí se zde rovněž cesta k CA certifiátu, serverovému certifikátu a serverovému klíči. Dále se volí IP rozsah serveru a DNS servery, které budou použity. Atribut *redirect-gateway def1 bypass-dhcp*, definuje přesměrování veškeré komunikace skrz VPN tunel. Dále je rovněž definováno logování a jeho verbosita. Důležitým parametrem je zde také *duplicate-dn*, což umožní více uživatelům připojení k VPN serveru za použití stejného klíče a certifikátu.

```
port 1194
proto udp
dev tun
ca ca.crt
cert server.crt
key server.key
dh dh.pem
duplicate-cn
server 10.8.0.0 255.255.0.0
push "redirect-gateway def1 bypass-dhcp"
push "dhcp-option DNS 208.67.222.222"
push "dhcp-option DNS 208.67.220.220"
keepalive 10 120
tls-auth ta.key 0
cipher AES-256-CBC
user nobody
group nogroup
```

```
persist-key
persist-tun
status /var/log/openvpn/openvpn-status.log
log /var/log/openvpn/openvpn.log
log-append /var/log/openvpn/openvpn.log
verb 3
explicit-exit-notify 1
```

Zdrojový kód 35: Konfigurační soubor /etc/openvpn/server.conf

Následně je nutné povolit IP přeposílání. V souboru */etc/sysctl.conf* odkomentujeme konfigurační řádek a načteme aktuální konfiguraci.

```
net.ipv4.ip_forward = 1
```

Zdrojový kód 36: Povolení IPv4 forwarding

```
# sysctl -p
```

Zdrojový kód 37: Načtení aktuální konfigurace

Ke správnému směřování je nutné upravit pravidla na firewallu. Konfigurační soubor */etc/default/ufw*.

```
DEFAULT_FORWARD_POLICY="ACCEPT"
```

Zdrojový kód 38: Konfigurační soubor /etc/default/ufw

Dále je nutné upravit směřovací pravidla v souboru */etc/ufw/before.rules*.

```
# START OPENVPN RULES
# NAT table rules
*nat
:POSTROUTING ACCEPT [0:0]
# Allow traffic from OpenVPN client to ens18
-A POSTROUTING -s 10.8.0.0/16 -o ens18 -j
   MASQUERADE
COMMIT
# END OPENVPN RULES
```

Zdrojový kód 39: Konfigurační soubor /etc/ufw/before.rules

Následuje povolení portu 1194 na firewallu a načtení aktuální konfigurace.

```
# ufw allow 1194/udp
# ufw reload
```

Zdrojový kód 40: Povolení portu 1194, aktualizace konfigurace

Tímto je serverová část nakonfigurována. Následuje spuštění služby *openvpn@server*.

```
# systemctl start openvpn@server
```

Zdrojový kód 41: Konfigurační soubor /etc/default/ufw

Následuje vygenerování certifikátů a kličů pro klienta. Budou vytořeny soubory:
/etc/openvpn/easy-rsa/pki/reqs/client.req,
/etc/openvpn/easy-rsa/pki/private/client.key.

```
# ./easyrsa gen-req client nopass
```

Zdrojový kód 42: Vytvoření klientského kliče

Podepsání klientského klíče, vytvoření klientského certifikátu. Bude vytvořen soubor */etc/openvpn/easy-rsa/pki/issued/client.crt*.

```
# ./easyrsa sign-req client client
```

Zdrojový kód 43: Vytvoření klientského certifikátu

Následně překopírujeme vytvořené certifikáty a klíče do cesty klientské konfigurace.

```
# cp -a /etc/openvpn/easy-rsa/pki/issued/client
  .crt /etc/openvpn/client/
# cp -a /etc/openvpn/easy-rsa/pki/private/ /etc
  /openvpn/client/
# cp -a /etc/openvpn/easy-rsa/pki/private/
  client.key /etc/openvpn/client/
# cp -a /etc/openvpn/easy-rsa/ta.key /etc/
  openvpn/client/
```

Zdrojový kód 44: Kopírování certifikátu a klíčů

Následuje zkopírování vygenerovaných klientských klíčů a certifikátu na klientskou stanici.

```
# scp /etc/openvpn/client/* root@192
  .168.221.112:/etc/openvpn/client
```

Zdrojový kód 45: Kopírování certifikátu a klíčů na klientskou stanici

Předposledním krokem je vytvoření samotného konfiguračního souboru na straně klienta. Konfigurační soubor bude v cestě */etc/openvpn/client.conf*. Direktiva je téměř totožná s konfigurací na straně serveru.

```
client
dev tun
proto udp
remote 192.168.221.25 1194
resolv-retry infinite
```

```
nobind
user nobody
group nogroup
persist-key
persist-tun
ca /etc/openvpn/client/ca.crt
cert /etc/openvpn/client/client.crt
key /etc/openvpn/client/client.key
remote-cert-tls server
tls-auth /etc/openvpn/client/ta.key 1
cipher AES-256-CBC
verb 3
```

Zdrojový kód 46: Konfigurační soubor /etc/openvpn/client.conf

Posledním krokem je připojení se k VPN serveru.

```
# systemctl start openvpn@client
```

Zdrojový kód 47: Připojení k VPN serveru

6.4 WireGuard

WireGuard bude užit ve verzi 1.0.20210914. Jedná se o výchozí verzi obsaženou ve standardním repozitáři operačního systému Ubuntu 22.04 LTS.

Prvním krokem je instalace samotného WireGuardu.

```
# apt install wireguard
```

Zdrojový kód 48: Instalace WireGuardu

Jakmile je WireGuard nainstalován, dalším krokem je vygenerování soukromého a veřejného klíče pro server. K vytvoření klíčů budou použity příkazy *wg genkey* a *wg pubkey*.

Zároveň je vhodné, stejně jako u předchozeních řešení, změnit oprávnění k soukromému klíči, protože ve výchozím nastavení je soubor čitelný pro každého uživatele na serveru.

```
# wg genkey | sudo tee /etc/wireguard/private.
  key
```

Zdrojový kód 49: Vytvoření soukromého klíče /etc/wireguard/private.key

```
# chmod go= /etc/wireguard/private.key
```

Zdrojový kód 50: Změna oprávnění k souboru /etc/wireguard/private.key

Dalším krokem je vytvoření veřejného kliče, který bude odvozen od klíče soukromého.

```
# cat /etc/wireguard/private.key | wg pubkey |
  sudo tee /etc/wireguard/public.key
```

Zdrojový kód 51: Vytvoření veřejného klíče /etc/wireguard/public.key

Následuje vytvoření konfiguračního souboru serveru. Soubor bude umístěn do cesty */etc/wireguard/* se jménem *wg0.conf*. Zde se nastaví IP adresa serveru, jeho privátní klíč, IP adresy povolených klientů, jejich veřejné klíče, port na kterém bude server poslouchat.

```
[Interface]
PrivateKey = eNc/Bj+d9tff5DFzGN3OP0cdka4xHz6Y/
  Gb5YPAb7mE=
Address = 10.8.0.1/24
ListenPort = 51820
SaveConfig = true
```

Zdrojový kód 52: Konfigurační soubor serveru /etc/wireguard/wg0.conf

Dalším krokem je povolení IPv4 přeposílání paketů. Stejně jako u předešlých řešení se upraví konfigurační soubor */etc/sysctl.conf*. Následuje načtení aktuální konfigurace.

```
net.ipv4.ip_forward=1
```

Zdrojový kód 53: Konfigurační soubor /etc/sysctl.conf

```
# sysctl -p
```

Zdrojový kód 54: Načtení aktualní konfigurace

Dalším krokem je nastavení firewallu. Je nutné povolit port 51820 a port 22, který využijeme ke kopírování veřejných klíčů ze serveru na klienta a opačně.

```
# ufw allow 51820/udp
# ufw allow OpenSSH
```

Zdrojový kód 55: Povolení portů

Následuje přidání směřovaích pravidel do souboru */etc/wireguard/wg0.conf*.

```
...
SaveConfig = true
PostUp = ufw route allow in on wg0 out on ens18
PostUp = iptables -t nat -I POSTROUTING -o
  ens18 -j MASQUERADE
PreDown = ufw route delete allow in on wg0 out
  on ens18
PreDown = iptables -t nat -D POSTROUTING -o
  ens18 -j MASQUERADE
```

Zdrojový kód 56: Konfigurační soubor serveru /etc/wireguard/wg0.conf

Dalším krokem je vytvoření konfiguračního souboru na straně klienta. Postup je identický jako tomu bylo na straně serveru.

```
# wg genkey | sudo tee /etc/wireguard/private.
  key
```

Zdrojový kód 57: Vytvoření soukrokého klíče - klient /etc/wireguard/private.key

```
# chmod go= /etc/wireguard/private.key
```

Zdrojový kód 58: Změna oprávnění k souboru - klient /etc/wireguard/private.key

```
# cat /etc/wireguard/private.key | wg pubkey |
  sudo tee /etc/wireguard/public.key
```

Zdrojový kód 59: Vytvoření veřejného klíče - klient /etc/wireguard/public.key

Následuje vytvoření klientského konfiguračního souboru */etc/wireguard/wg0-client.conf*.

```
[Interface]
PrivateKey = IHztu6n7O7Tp581/
  ZdpBK6sd9YpxCPa3e7x47BiuuUE=
Address = 10.8.0.2/24
[Peer]
PublicKey =
  ImNJDWUE4opL236PpWb62QUOw0zrdyXy0MHm3x286Bo=
AllowedIPs = 0.0.0.0/0
Endpoint = 192.168.221.27:51820
```

Zdrojový kód 60: Konfigurační soubor /etc/wireguard/wg0-client.conf

Posledním krokem je přidání klienta do konfiguračního souboru serveru. Konečný stav konfiguračního souboru */etc/wireguard/wg0.conf*:

```
[Interface]
Address = 10.8.0.1/24
SaveConfig = true
PostUp = ufw route allow in on wg0 out on ens18
PostUp = iptables -t nat -I POSTROUTING -o
  ens18 -j MASQUERADE
PreDown = ufw route delete allow in on wg0 out
  on ens18
PreDown = iptables -t nat -D POSTROUTING -o
  ens18 -j MASQUERADE
ListenPort = 51820
PrivateKey = eNc/Bj+d9tff5DFzGN3OP0cdka4xHz6Y/
  Gb5YPAb7mE=
```

```
[Peer]
PublicKey = 8
   g7GLplctRO5krHULDiYlCeh65N5B3DvDKxkiG/lXnY=
AllowedIPs = 10.8.0.2/32
```

Zdrojový kód 61: Konfigurační soubor /etc/wireguard/wg0.conf

Konfigurace na straně serveru i klienta je hotova, nyní stačí spustit službu na klientovi a serveru.

```
# wg-quick up wg0
```

Zdrojový kód 62: Spuštění serveru

```
# wg-quick up wg0-client
```

Zdrojový kód 63: Spuštění klienta

7 Testování protokolů

Testování zvolených VPN protokolů bude provedeno z několika hledisek - celková propustnost, odezva, hardwarová zátěž z pohledu CPU a užité paměti RAM. Zároveň bude otestovaná stabilita jednotlivých řešení.

Pro testování odezvy bude užito programu *ping*, kdy bude testována komunikace s DNS servery společnosti CloudFlare.

K testování je dále užito nástroje iperf[24] ve verzi 2.1.5. Tento nástroj slouží primárně k síťovému testování. Tento nástroj byl nainstalován na VPN servery i na klienty. Program *iperf* byl spuštěný na serveru s přepínačem *-s*, což znační, že se jedná o stranu serveru. Na klientovi bylo užito přepínače *-c*, což značí, že se jedná o stranu klienta. Bylo také užito přepínače *-t*, což uvádí dobu čekání na spojení na straně serveru. Tento přepínač na straně klienta značí, jak dlouho bude probíhat datová komunikace. Přepínač *-b* uvádí šířku pásma. Při testování datové propustnosti bylo užito přepínače *-d*, který říká, že se jedná o obousměrnou komunikaci. Hodnoty měření byly zaznamenávány každou sekundu.

Pro testování zátěže na hardware byl použitý balíček sysstat[25]. Z tohoto balíčku byl užit program *sar*, který slouží k monitorování zátěže mnoha hardwarových parametrů. Měření probíhalo na straně serveru. Při užití programu *sar* bylo užito přepínače *-r*, což monitoruje vytížení RAM, přepínač *-u* značí vytížení CPU.

7.1 Odezva

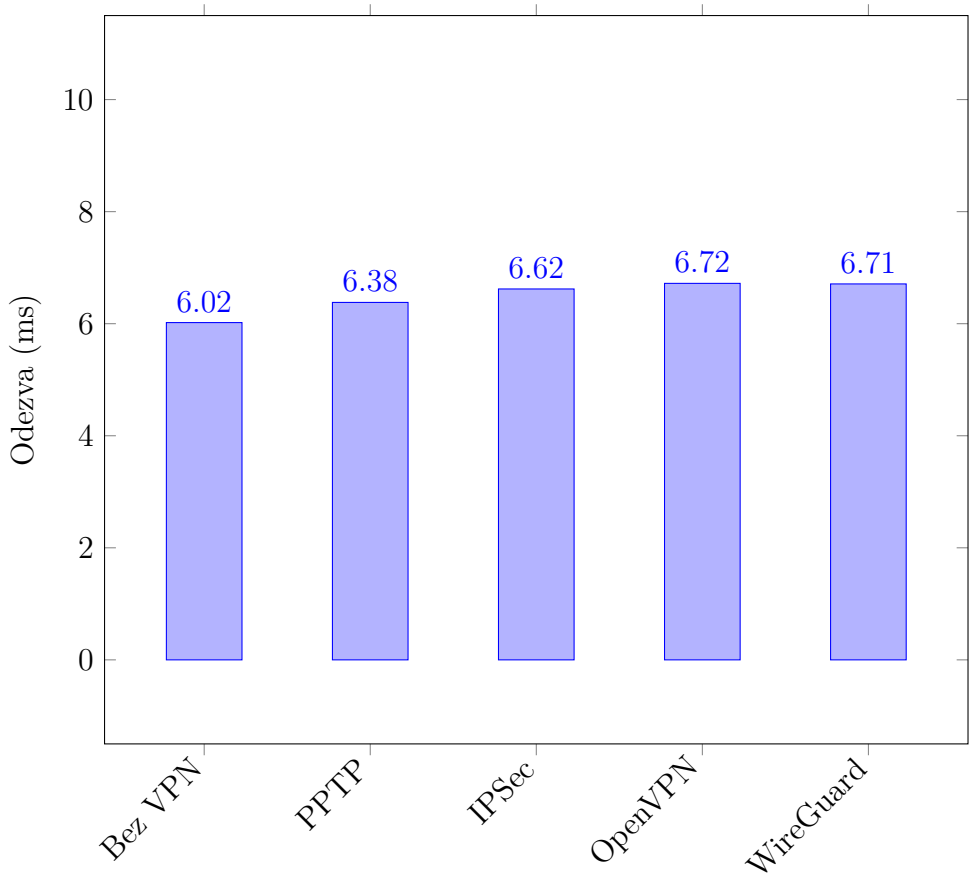
Byla měřena odezva od CloudFlare DNS - 1.1.1.1, pomocí nástroje *ping*. Hodnota byla měřena po dobu 60 sekund a z výsledných hodnot byl vypočten průměr, který lze vidět ve Grafu 23. Výsledek měření ukazuje, že mezi protokoly není

téměř žádný rozdíl, co se týče odezvy od DNS serveru 1.1.1.1. Naměřené hodnoty se u jednotlivých protokolů mohou v čase o setiny hodnoty měnit, tudíž na základě tohoto testu nelze jednoznačně říci, který z protokolů je v tomto ohledu nejlepším. Pro srovnání byla také přidána hodnota odezvy bez užití VPN, která je prakticky totožná s hodnotami při užití VPN tunelu. Takto shodné hodnoty lze přisuzovat i faktu, že samotné VPN servery se nacházejí ve stejné lokální síti.

```
# ping -c 60 1.1.1.1
```

Zdrojový kód 64: Měření odezvy

Obrázek 23: Graf odezvy od CloudFlare DNS



7.2 Propustnost

Propustnost linky mezi VPN serverem a klientem byla měřena utilitou *iperf*. Měření probíhalo po dobu 60 sekund, v Grafu 24 jsou uvedené průměrné hodnoty měření. Protokol PPTP v tomto měření dosahoval nejnižších hodnot a to průměrně 87 Mbit/s. Jako druhý nejpomalejší, ačkoliv s velkým rozdílem od protokolu PPTP, byl protokol OpenVPN s průměrnou hodnotou přenosu 477 Mbit/s.

IPSec v tomto byl ještě o něco rychlejší, jeho průměrná přenosová rychlost byla 585 Mbit/s. Jednoznačně nejrychlejším zde byl nejmladší protokol WireGuard, který je často označován jako nejrychlejší VPN protokol dnešní doby. V tomto testu jeho průměrná přenosová rychlost činila 818 Mbit/s. Pro srovnání byla také změřena přenosová rychlost mezi VPN serverem a klientem bez užití tunelu. Průměrné hodnota tohoto měření byla 850 Mbis/s. Rozdíl tohoto měření se svým výsledkem přiliš neliší od užití protokolu WireGuard. Vysoké přenosové rychlosti protokolů WireGuard a IPSec lze přisoudit i faktu, že na rozdíl od zbývajících VPN protokolů, kód těchto řešení běží přímo v jádru operačního systému.

Na všech VPN serverech byl povolen port 5001, což je výchozí port, na kterém poslouchá program *iperf*.

```
# ufw allow 5001
```

Zdrojový kód 65: Povolení portu 5001 na firewallu

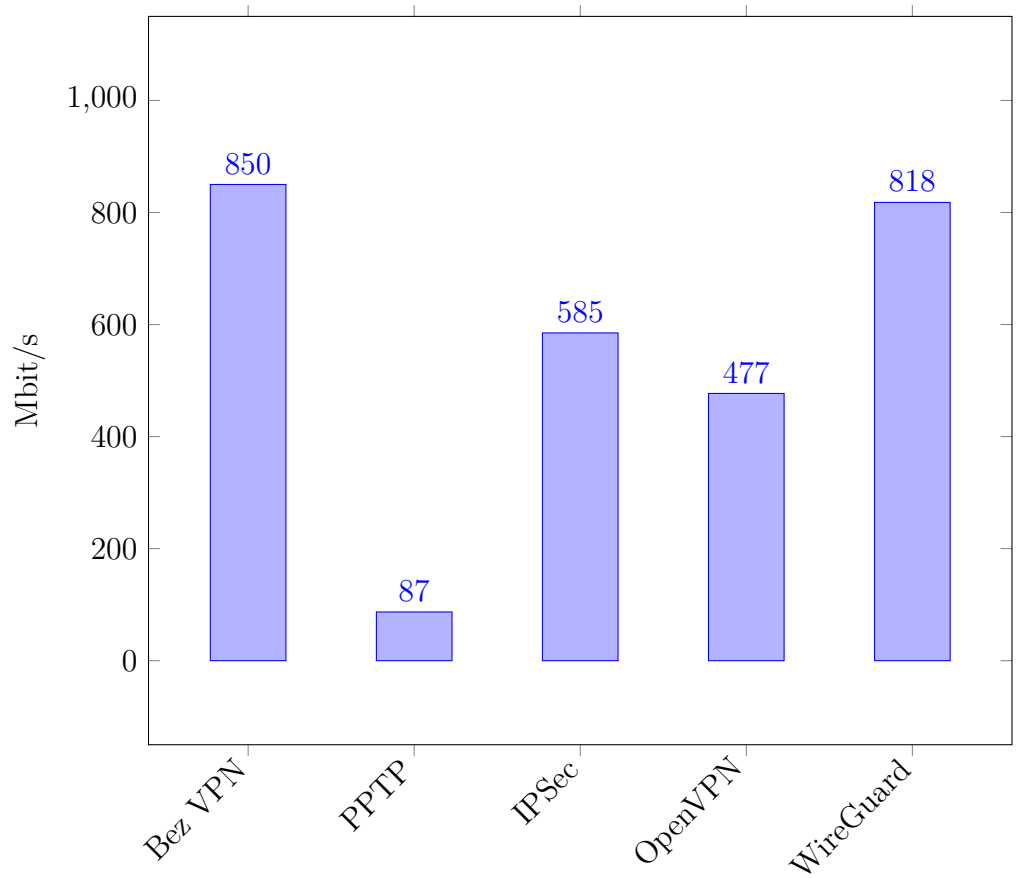
```
# iperf -s -t 1000
```

Zdrojový kód 66: Měření propustnosti VPN linky - server

```
# iperf -c <IP Serveru> -d -t 60
```

Zdrojový kód 67: Měření propustnosti VPN linky - klient

Obrázek 24: Graf propustnosti mezi VPN serverem a klientem



7.3 Hardwarová zátěž

Měřena byla hardwarová zátěž u jednotlivých protokolů. Konkrétně byla testována CPU a RAM utilizace v závislosti na počtu klientů a šířce pásma.

K měření bylo užito programu *sar* a programu *iperf*. V případě měření, kdy bylo na VPN server připojeno více klientů, bylo užito programu *Ansible*[26] ve verzi 7.4.0. Ansible byl nainstalován na samostatném serveru. Na tomto serveru byl konfigurován bezheslový přístup na všechny klienty za pomocí SSH klíčů. Ansible byl užit k zjenodušení testování, jelikož umožňuje spouštění příkazů na více klientech současně.

V první části měření bylo testování bylo měřeno zatížení CPU.

Na straně serveru bylo užito příkazů:

```
# iperf -s -u -t 100
```

Zdrojový kód 68: Server očekává příchozí UDP spojení po dobu 100 sekund

```
# sar -u 1 60
```

Zdrojový kód 69: Měření zatížení CPU po dobu 60 sekund

Na straně ansible serveru bylo užito příkazu:

```
# ansible all -m shell -a "iperf -c <IP adresa
  VPN serveru> -u -b <bandwidth> -t 60"
```

Zdrojový kód 70: Spuštění měření na straně klientů

Bylo měřeno celkové zatížení CPU při šířkách pásma 10 Mbit/s, 25 Mbit/s, 50 Mbit/s a 100 Mbit/s. Bylo testováno připojení jednoho, tří a pěti klientů při různých šířkách pásma. Cílem bylo zjistit, jak počet klientů a šířka pásma ovlivňují hardwarové zatížení VPN serverů.

Výsledkem jsou průměrné hodnoty získané z výstupu programu *sar*, konkrétně se jedná o hodnoty *idle*, které značí, jak často je server nečinný. Čím jsou hodnoty vyšší, tím je server méně zatížen.

7.3.1 PPTP

Prvním testovaným byl protokol PPTP. Při šířce pásma 10 Mbit/s byla průměrná hodnota nečinnosti CPU 99.13. Při 25 Mbit/s byla naměřena hodnota 97.79. Při šířce pásma 50 Mbit/s byla výsledná hodnota 91.30 a při 100 Mbit/s byl výsledek 89.66.

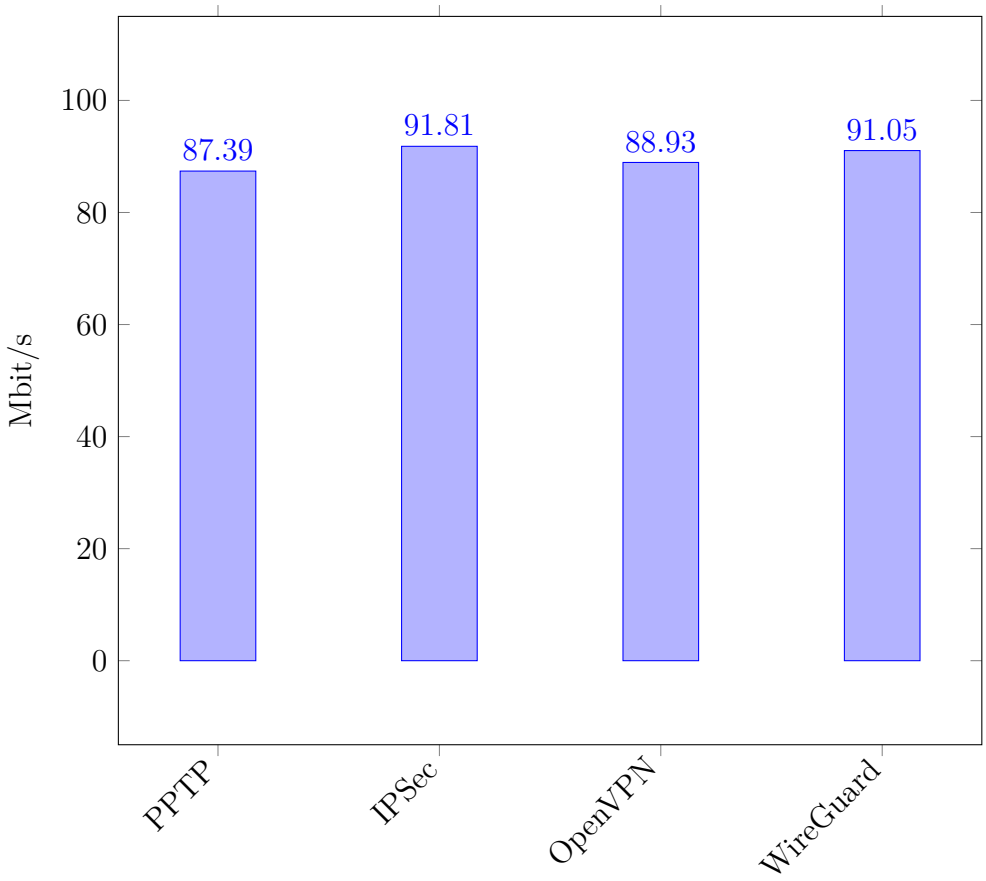
Výsledné hodnoty se u protokolu PPTP snižovaly s postupným přidáváním klientů a zvyšováním šířky pásma. Nejnižší hodnoty byly naměřeny u tří a pěti klientů při šířce pásma 50 Mbit/s a 100 Mbit/s. Nejnižší hodnota byla naměřena při pěti připojených klientech a šířce pásma 100 Mbit/s. Jednalo se o průměrnou hodnotu 20.31. Veškeré naměřené hodnoty lze vidět v Tabulce 1.

Nečinnost CPU - Průměrné hodnoty				
Počet klientů	10 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
1	99.13	97.79	91.30	89.66
3	97.22	93.01	85.58	73.11
5	95.95	87.39	74.22	20.31

Tabulka 1: PPTP CPU utilizace

Pro srovnání zátěže protokolu PPTP s ostatními protololy bylo zvolen datový tok 25 Mbit/s s pěti připojenými klienty. Výsledek měření lze vidět na Obrázku 25. Na základě výsledků při užití pěti klientů a šířce pásma 25 Mbit/s, lze říci, že PPTP je ze všech testovaných protokolů nejvíce hardwarově náročný, ačkoliv rozdíly jsou v tomto případě v rámci nižších jednotek. Na druhou stranu, při měření šířky pásma 100 Mbit/s s pěti připojenými klienty, si protokol PPTP vedl výrazně nejhůře ze všech testovaných protokolů, s výslednou hodnotou 20.31.

Obrázek 25: Graf nečinnosti CPU - 5 klientů, 25 Mbit/s



7.3.2 IPSec

Druhým testovaným byl protokol IPSec. Při šířce pásma 10 Mbit/s byla průměrná hodnota nečinnosti CPU 99.24. Při 25 Mbit/s byla naměřena hodnota 95.72. Při šířce pásma 50 Mbit/s byla výsledná hodnota 92.40 a při 100 Mbit/s byl výsledek 82.92.

Výsledné hodnoty se u protokolu IPSec snižovaly s postupným přidáváním klientů a zvyšováním šířky pásma. Nejnižší hodnoty byly naměřeny u současně připojených tří a pěti klientů při užití šířky pásma 50 Mbit/s a 100 Mbit/s. Stejně jako u PPTP, nejnižší hodnota byla naměřena při pěti připojených klientech a šířce pásma 100 Mbit/s. Jednalo se o průměrnou hodnotu 77.27. Veškeré naměřené hodnoty lze vidět v Tabulce 2.

Nečinnost CPU - Průměrné hodnoty				
Počet klientů	10 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
1	99.24	95.72	92.40	82.92
3	97.90	94.96	89.90	79.92
5	96.91	91.81	83.66	72.27

Tabulka 2: IPSec CPU utilizace

7.3.3 OpenVPN

Dalším testovaným protokolem byl protokol OpenVPN. Při šířce pásma 10 Mbit/s byla průměrná hodnota nečinnosti CPU 98.95. Při 25 Mbit/s byla naměřena hodnota 98.00. Při šířce pásma 50 Mbit/s byla výsledná hodnota 95.96 a při 100 Mbit/s byl výsledek 92.20.

Stejně jako u protokolu PPTP či IPSec, nejnižší hodnoty byly naměřeny u současně připojených tří a pěti klientů při užití šířky pásma 50 Mbit/s a 100 Mbit/s. Nejnižší hodnota byla naměřena při pěti připojených klientech a šířce pásma 50 Mbit/s. Jednalo se o průměrnou hodnotu 77.45. Při připojení pěti klientů s užitím šířky pásma 100 Mbit/s, nebylo možné změřit celkové zatížení z důvodu přetížení serveru, který nezvládal dostatečně rychle odbavovat požadavky. Výsledky lze vidět v Tabulce 3.

Nečinnost CPU - Průměrné hodnoty				
Počet klientů	10 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
1	98.95	98.00	95.96	92.20
3	97.22	93.10	86.77	74.83
5	95.53	88.93	77.45	-

Tabulka 3: OpenVPN CPU utilizace

7.3.4 WireGuard

Posledním testovaným protokolem byl protokol WireGuard. Bylo provedeno měření s jedním připojeným klientem s užitím šířky pásma 10 Mbit/s, 25 Mbit/s, 50 Mbit/s a 100 Mbit/s. Při šířce pásma 10 Mbit/s byla průměrná hodnota nečinnosti CPU 99.50. Při 25 Mbit/s byla naměřena hodnota 98.17. Při šířce pásma 50 Mbit/s byla výsledná hodnota 96.40 a při 100 Mbit/s byl výsledek 94.91.

Výsledné hodnoty se u protokolu WireGuard snižovaly s postupným přidáváním klientů a zvyšováním šířky pásma, stejně jako u ostatních testovaných protokolů. Nejnižší hodnoty byly naměřeny u současně připojených tří a pěti klientů při užití šířky pásma 50 Mbit/s a 100 Mbit/s. Nejnižší hodnota byla naměřena při pěti připojených klientech a šířce pásma 100 Mbit/s. Jednalo se o průměrnou hodnotu 71.35. Veškeré naměřené hodnoty lze vidět v Tabulce 4.

Nečinnost CPU - Průměrné hodnoty				
Počet klientů	10 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
1	99.50	98.17	96.40	94.91
3	97.24	94.79	89.91	81.17
5	96.46	91.05	81.70	71.35

Tabulka 4: WireGuard CPU utilizace

7.3.5 Hardwarová zátěž - Shrnutí výsledků měření

V Tabulce 5 je možno vidět celkové srovnání hodnot z předchozích měření u všech testovaných protokolů. Zeleně je značena hodnota nejvyšší, červeně je značena hodnota nejnižší. Tímto vidíme, že protokol PPTP je v mnoha ohledech protokolem nejvíce hardwarově náročným, zároveň v žádném z testovaných měření neměl výsledek nejlepší v porovnání s ostatními protokoly. Protokol IPSec se na druhou stranu jeví jako dobrý protokol z hlediska hardwarové náročnosti, zejména při vysokém vytížení, kdy při připojení pěti klientů měl nejnižší hardwarové zatížení ve všech šířkách pásma. Protokol OpenVPN v žádném z testů nevynikal, ale vyloženě ani neztrácel. Jedinou výjimkou je test s pěti připojenými klienty a s užitím šířky pásma 100 Mbit/s. V takovém testu nebylo možné stanovit průměrné hardwarové zatížení z důvodu opakovanému ukončování spojení mezi klientem a serverem. Protokol WireGuard se v tomto testu jeví jako ideální volba, pokud se jedná o spojení s jedním klientem v jakékoliv šířce pásma.

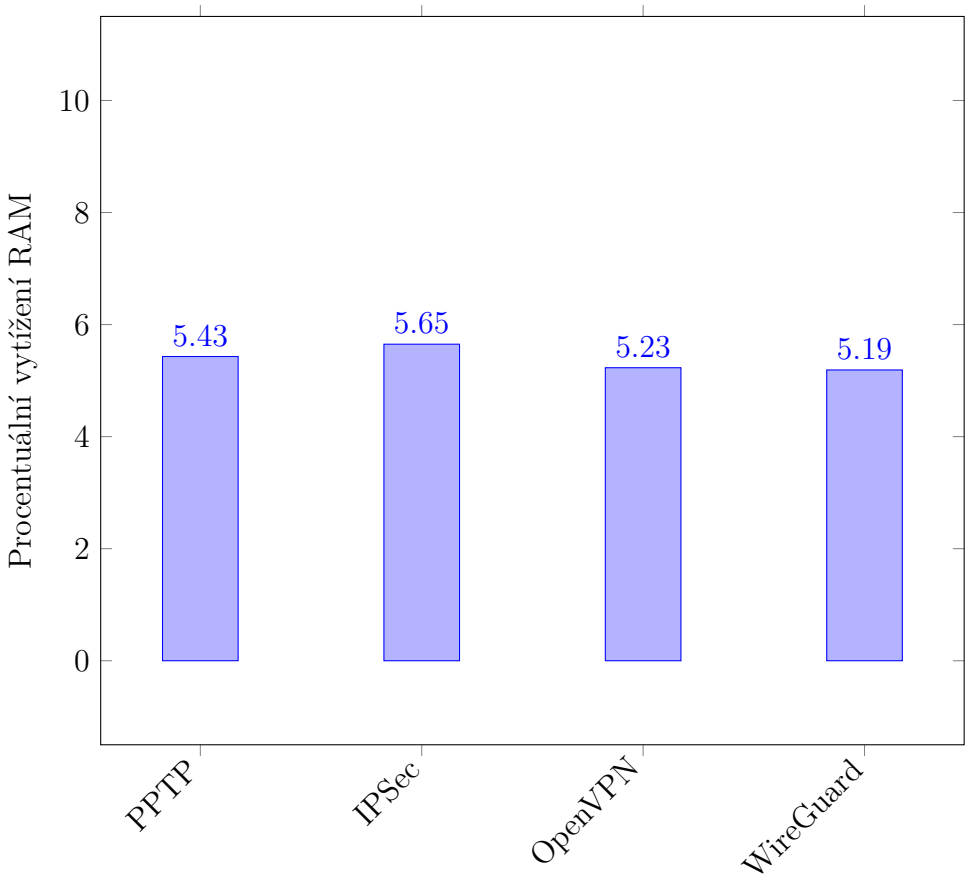
Nečinnost CPU - Průměrné hodnoty					
Protokol	Počet klientů	10 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
PPTP	1	99.13	97.79	91.30	89.66
	3	97.22	93.01	85.58	73.11
	5	95.95	87.39	74.22	71.22
IPSec	1	99.24	95.72	92.40	82.92
	3	97.90	94.96	89.90	79.92
	5	96.61	91.81	83.66	72.27
OpenVPN	1	98.95	98.00	95.96	92.20
	3	97.23	93.10	86.77	74.83
	5	95.53	88.93	77.45	-
WireGuard	1	99.50	98.17	96.40	94.91
	3	97.24	94.79	89.91	81.17
	5	96.46	91.05	81.70	71.35

Tabulka 5: Celkové srovnání CPU náročnosti

Druhou částí testování hardwarové náročnosti bylo testování zatížení RAM. Stejně jako u testu zatížení CPU, bylo testováno současné připojení jednoho, tří a pěti klientů. Pro testování bylo užito šířek pásma 10 Mbit/s, 25 Mbit/s, 50 Mbit/s a 100 Mbit/s.

Výsledky těchto testů ukázaly, že jak počet připojených klientů, tak šířka pásma, nemá na vytížení RAM podstatný vliv. Rozdíly mezi protokoly byly minimální, jak je možno vidět ve Grafu 26.

Obrázek 26: Graf průměrné utilizace RAM - 5 klientů, 100 Mbit/s



V Tabulce 6 je možno vidět celkové porovnání vytížení RAM za všechny testované případy. Data ukazují, že vytížení RAM je téměř na stejné úrovni u všech testovaných protokolů bez ohledu na počet připojených klientů či užitou šířku pásma. Na základě zjištěných dat lze říci, že celkové využití RAM na straně VPN serveru, není příliš ovlivněno používáním konkrétního řešení.

Procentuální využití RAM - Průměrné hodnoty					
Protokol	Počet klientů	10 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
PPTP	1	5.36	5.38	5.32	5.39
	3	5.43	5.45	5.42	5.46
	5	5.40	5.50	5.39	5.56
IPSec	1	5.02	4.96	5.10	5.12
	3	5.22	5.13	5.38	5.46
	5	5.59	5.65	5.65	5.58
OpenVPN	1	4.92	4.89	4.90	4.96
	3	4.69	4.91	5.02	5.06
	5	5.20	5.22	5.23	-
WireGuard	1	4.94	4.89	4.87	4.88
	3	5.01	5.03	5.01	5.03
	5	5.11	5.14	5.19	5.16

Tabulka 6: Celkové srovnání vytížení RAM

7.4 Stabilita

Dále byla testována stabilita jednotlivých VPN řešení. Bylo užito programu *iperf*. Měřen byl *jitter* a průměrná ztrátovost paketů při různých šířkách pásma. Pro komunikaci bylo užito protokolu UDP. Každé VPN řešení bylo testováno s jedním, třemi a pěti klienty. K testování bylo rovněž užito programu *Ansible*, jehož účelem bylo spouštění příkazů na více klientech současně.

K testování bylo užito následujících příkazů:

```
# iperf -s -u -t 100
```

Zdrojový kód 71: Měření stabibility řešení - strana serveru

```
# ansible all -m shell -a "iperf -c <IP adresa
  VPN serveru> -u -b <bandwidth> -t 60"
```

Zdrojový kód 72: Měření stabibility řešení - strana klienta

7.4.1 PPTP

Při tomto testu bylo sledována hodnota *jitter* a také ztrátovost paketů v testovaných šířkách pásma a při zvoleném počtu souběžně připojených klientů.

Při připojení jednoho klienta na VPN server byla naměřena v šířce pásma 10 Mbit/s hodnota jitteru 0.007 ms. Při šířce pásma 25 Mbit/s byla naměřena průměrná hodnota dokonce nižší a to 0.005 ms. Při dvojnásobné šířce pásma, 50 Mbit/s bylo naměřeno hodnoty 0.003 ms a při šířce pásma 100 Mbit/s bylo naměřeno hodnoty 0.009 ms. Ve všech těchto měřeních byla nulová ztrátovost paketů. Rovněž nedocházelo k doručení paketů v jiném pořadí, než byly odeslány. Výsledné hodnoty měření s jedním klientem je možné vidět v Tabulce 7.

Při měření připojení tří klientů se hodnoty *jitter* příliš nelišily od měření pouze s jedním klientem. Výsledné hodnoty se třemi připojenými klienty lze vidět v Tabulce 8.

Výsledky měření při pěti připojených klientech byly velmi podobné dvěma předchozím měřením. Hodnoty *jitter* se pohybovaly mezi 0.005 ms a 0.021 ms. Zásadní rozdíl byl zaznamenán při pěti připojených klientech, kdy každý klient využíval šířku pásma 100 Mbit/s. V tomto případě se výsledné hodnoty zásadně zvýšily, a to až k hodnotám 100ms. Výsledné hodnoty s pěti připojenými klienty lze vidět v Tabulce 9.

PPTP - Jitter (ms) - 1 klient					
Protokol	Klient	10 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
PPTP	1	0.007	0.005	0.003	0.009

Tabulka 7: PPTP - Jitter (ms) - 1 klient

PPTP - Jitter (ms) - 3 klienti					
Protokol	Klient	10 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
PPTP	1	0.005	0.006	0.007	0.011
	2	0.006	0.006	0.007	0.011
	3	0.004	0.004	0.007	0.012

Tabulka 8: PPTP - Jitter (ms) - 3 klienti

PPTP - Jitter (ms) - 5 klientů					
Protokol	Klient	10 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
PPTP	1	0.006	0.007	0.007	97
	2	0.009	0.011	0.010	100
	3	0.008	0.008	0.005	97
	4	0.010	0.005	0.008	99
	5	0.021	0.010	0.005	96

Tabulka 9: PPTP - Jitter (ms) - 5 klientů

U měření celkové stability připojení byla také sledována celková ztrátovost paketů a jejich doručení ve spravném pořadí. V případě jednoho klienta byla naměřena nulová ztrátovost paketů ve všech zkoumaných šířkách pásma, zároveň byly veškeré pakety doručeny ve správném pořadí.

Obdobné výsledky byly naměřeny při třech a pěti klientech v nižších šířkách pásma. Při třech připojených klientech se do maximální šířky pásma 25 Mbit/s hodnoty ztrátovosti paketů držely na nule. Pakety do této šířky pásma byly doručovány ve správném pořadá. Při užití větší šířky pásma, byly hodnoty ztrátovosti vyšší, zároveň se zvýšil počet paketů, které nedorazily ve správném pořadí.

V případě měření paketů, které byly přijaty ve špatném pořadí, se jednalo o tisíciny procent. Měřením nebyla zjištěna souvislost mezi počtem klientů, šířkou pásma a celkovým počtem paketů, které dorazily ve špatném pořadí.

Výsledné hodnoty ztrátovosti jsou uvedeny ve srovnání s ostatními VPN protokoly v závěrů sekce.

7.4.2 IPSec

Při tomto testu, stejně jako u protokolu PPTP, byla sledována hodnota *jitter*, ztrátovost paketů v testovaných šířkách pásma a při zvoleném počtu souběžně připojených klientů.

Při připojení jednoho klienta na VPN server byla naměřena v šířce pásma 10 Mbit/s hodnota jitteru 0.004 ms. Při šířce pásma 25 Mbit/s byla naměřena průměrná hodnota dokonce nižší a to 0.002 ms. Při dvojnásobné šířce pásma, 50 Mbit/s, bylo naměřeno hodnoty 0.002 ms a při šířce pásma 100 Mbit/s bylo naměřeno hodnoty 0.006 ms. Ve všech těchto měřeních, stejně jako u protokolu PPTP, byla zjištěna nulová ztrátovost paketů. Rovněž nedocházelo k doručení paketů v jiném pořadí, než byly odeslány. Výsledné hodnoty měření s jedním klientem je možné vidět v Tabulce 10.

Při měření připojení tří klientů se hodnoty *jitter* příliš nelišily od měření pouze s jedním klientem. Výsledné hodnoty se třemi připojenými klienty lze vidět v Tabulce 11.

Výsledky měření při pěti připojených klientech byly velmi podobné dvěma předchozím měřením. Hodnoty *jitter* se pohybovaly mezi 0.002 ms a 0.071 ms. Výsledné hodnoty při pěti připojených klientech lze vidět v Tabulce 12.

IPSec - Jitter (ms) - 1 klient					
Protokol	Klient	10 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
IPSec	1	0.004	0.002	0.002	0.006

Tabulka 10: IPSec - Jitter (ms) - 1 klient

IPSec - Jitter (ms) - 3 klienti					
Protokol	Klient	10 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
IPSec	1	0.002	0.005	0.002	0.006
	2	0.006	0.004	0.007	0.020
	3	0.002	0.002	0.003	0.011

Tabulka 11: IPSec - Jitter (ms) - 3 klienti

IPSec - Jitter (ms) - 5 klientů					
Protokol	Klient	10 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
IPSec	1	0.004	0.005	0.006	0.071
	2	0.005	0.004	0.004	0.043
	3	0.004	0.005	0.005	0.053
	4	0.004	0.004	0.009	0.048
	5	0.005	0.003	0.005	0.038

Tabulka 12: IPSec - Jitter (ms) - 5 klientů

U měření celkové stability připojení byla také sledována celková ztrátovost paketů a jejich doručení ve spravném pořadí. V případě jednoho klienta byla naměřena nulová ztrátovost paketů ve všech zkoumaných šířkách pásma, zároveň byly veškeré pakety doručeny ve správném pořadí.

Obdobné výsledky byly naměřeny při třech a pěti klientech, kdy celková ztrátovost paketů byla nulová. Jedinou výjimkou bylo připojení pěti klientů a využití šířky pásma 100 Mbit/s. V takovém případě docházelo ke průměrné ztrátovosti paketů ve výši 1%. V případě měření paketů, které byly přijaty ve špatném pořadí, bylo ve všech testech zjištěna nulová hodnota. Všechny odeslané byly doručeny v pořadí, ve kterém byly odeslány.

Výsledné hodnoty ztrátovosti jsou uvedeny ve srovnání s ostatními VPN protokoly v závěrů sekce.

7.4.3 OpenVPN

Při tomto testu, stejně jako u protokolu PPTP a IPSec, byla sledována hodnota *jitter*, ztrátovost paketů v testovaných šířkách pásma a při zvoleném počtu souběžně připojených klientů.

Při připojení jednoho klienta na VPN server, byla naměřena v šířce pásma 10 Mbit/s hodnota jitteru 0.004 ms. Při šířce pásma 25 Mbit/s byla naměřena průměrná hodnota dokonce nižší a to 0.011 ms. Při dvojnásobné šířce pásma, 50 Mbit/s, bylo naměřeno hodnoty 0.004 ms a při šířce pásma 100 Mbit/s bylo naměřeno hodnoty 0.062 ms. Ve všech těchto měřeních, stejně jako u protokolu PPTP a IPSec, byla zjištěna nulová ztrátovost paketů. Rovněž nedocházelo k doručení paketů v jiném pořadí, než byly odeslány. Při měření připojení tří klientů se hodnoty *jitter* příliš nelišily od měření pouze s jedním klientem.

OpenVPN - Jitter (ms) - 1 klient					
Protokol	Klient	10 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
IPSec	1	0.004	0.011	0.004	0.062

Tabulka 13: OpenVPN - Jitter (ms) - 1 klient

OpenVPN - Jitter (ms) - 3 klienti					
Protokol	Klient	10 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
IPSec	1	0.008	0.005	0.014	0.033
	2	0.005	0.004	0.018	0.072
	3	0.008	0.008	0.017	0.033

Tabulka 14: OpenVPN - Jitter (ms) - 3 klienti

OpenVPN - Jitter (ms) - 5 klientů					
Protokol	Klient	10 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
IPSec	1	0.005	0.017	0.008	-
	2	0.007	0.005	0.035	-
	3	0.009	0.005	0.022	-
	4	0.007	0.013	0.020	-
	5	0.008	0.008	0.036	-

Tabulka 15: OpenVPN - Jitter (ms) - 5 klientů

U měření celkové stability připojení byla také sledována celková ztrátovost paketů a jejich doručení ve spravném pořadí. V případě jednoho klienta byla naměřena nulová ztrátovost paketů ve všech zkoumaných šířkách pásma, zároveň byly veškeré pakety doručeny ve správném pořadí.

Při měření ztrátovosti paketů při třech a pěti klientech a šířky pásma do 25 Mbit/s, nebylo zjištěna ztrátovost jediného paketu. Při testování tří klientů a šířky pásma 50 Mbit/s, byla naměřena ztrátovost v rámci tisícin procenta. Při měření tří klientů a šířce pásma 100 Mbit/s, byla zjištěna průměrná ztrátovost 9%. Při užití pěti klientů a šířky pásma 50 Mbit/s, byla naměřena průměrná ztrátovost prakticky nulová, konkrétně 0.054%. Při užití pěti klientů a šířky pásma 100 Mbit/s, byla naměřena ztátovost paketů v průměru 13%.

V případě měření paketů, které byly přijaty ve špatném pořadí, bylo ve všech testech zjištěna nulová hodnota.

V závislosti na zjištěných datech lze říci, že protokol OpenVPN lze považovat za spolehlivější než protokol PPTP, zároveň ale méně spolehlivý než protokol IPSec. Hodnoty *jitteru*, stejně jako u PPTP a IPSec, dosahují tisícin procenta.

Výsledné hodnoty ztrátovosti jsou uvedeny ve srovnání s ostatními VPN protokoly v závěrů sekce.

7.4.4 WireGuard

Při tomto testu, stejně jako u ostatních testovaných protokolů, byla sledována hodnota *jitter*, ztrátovost paketů v testovaných šířkách pásma a při zvoleném počtu souběžně připojených klientů.

Při připojení jednoho klienta na VPN server, byla naměřena v šířce pásma 10 Mbit/s hodnota jitteru 0.010 ms. Při šířce pásma 25 Mbit/s byla naměřena

průměrná hodnota dokonce nižší a to 0.019 ms. Při dvojnásobné šířce pásma, 50 Mbit/s bylo naměřeno hodnoty 0.004 ms a při šířce pásma 100 Mbit/s bylo naměřeno hodnoty 0.006 ms. Ve všech měřeních byla zjištěna nulová ztrátovost paketů. Rovněž nedocházelo k doručení paketů v jiném pořadí, než byly odeslány. Výsledné hodnoty měření s jedním klientem je možné vidět v Tabulce 16.

Při měření připojení tří klientů se hodnoty *jitter* zásadně nelišily od měření pouze s jedním připojeným klientem. Výsledné hodnoty se třemi připojenými klienty lze vidět v Tabulce 17.

Výsledky měření při pěti připojených klientech byly velmi podobné dvěma předchozím měřením. Hodnoty *jitter* se pohybovaly mezi 0.004 ms a 0.025 ms. Ve srovnání s ostatními testovanými protokoly, protokol WireGuard dosahoval nejnižších hodnot *jitter*. Výsledné hodnoty při pěti připojených klientech lze vidět v Tabulce 18.

WireGUard - Jitter (ms) - 1 klient					
Protokol	Klient	10 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
WireGuard	1	0.010	0.019	0.004	0.006

Tabulka 16: WireGuard - Jitter (ms) - 1 klient

WireGuard - Jitter (ms) - 3 klienti					
Protokol	Klient	10 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
WireGuard	1	0.004	0.017	0.005	0.006
	2	0.015	0.006	0.006	0.004
	3	0.0013	0.018	0.006	0.002

Tabulka 17: WireGuard - Jitter (ms) - 3 klienti

WireGuard - Jitter (ms) - 5 klientů					
Protokol	Klient	10 Mbit/s	25 Mbit/s	50 Mbit/s	100 Mbit/s
WireGuard	1	0.004	0.009	0.008	0.004
	2	0.019	0.006	0.008	0.008
	3	0.021	0.007	0.007	0.009
	4	0.007	0.010	0.012	0.008
	5	0.008	0.025	0.005	0.010

Tabulka 18: WireGuard - Jitter (ms) - 5 klientů

Při měření ztrátovosti paketů, nebyla v žádném z testů zjištěna ztráta jediného paketu. Rovněž se u protokolu WireGuard neobjevilo přijetí paketů v jiném než původním pořadí.

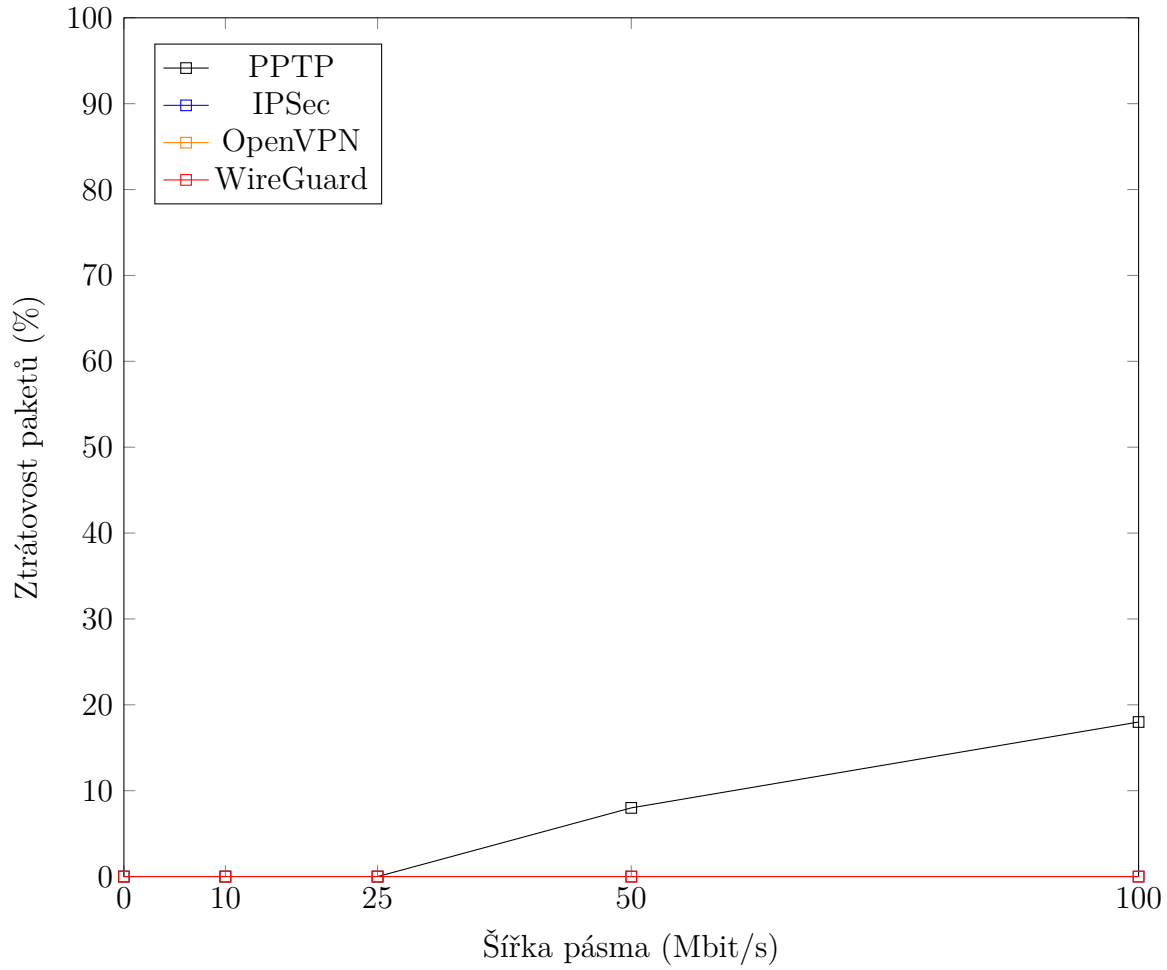
Výsledné hodnoty ztrátovosti jsou uvedeny ve srovnání s ostatními VPN protokoly v závěrů sekce.

7.4.5 Shrnutí výsledků měření

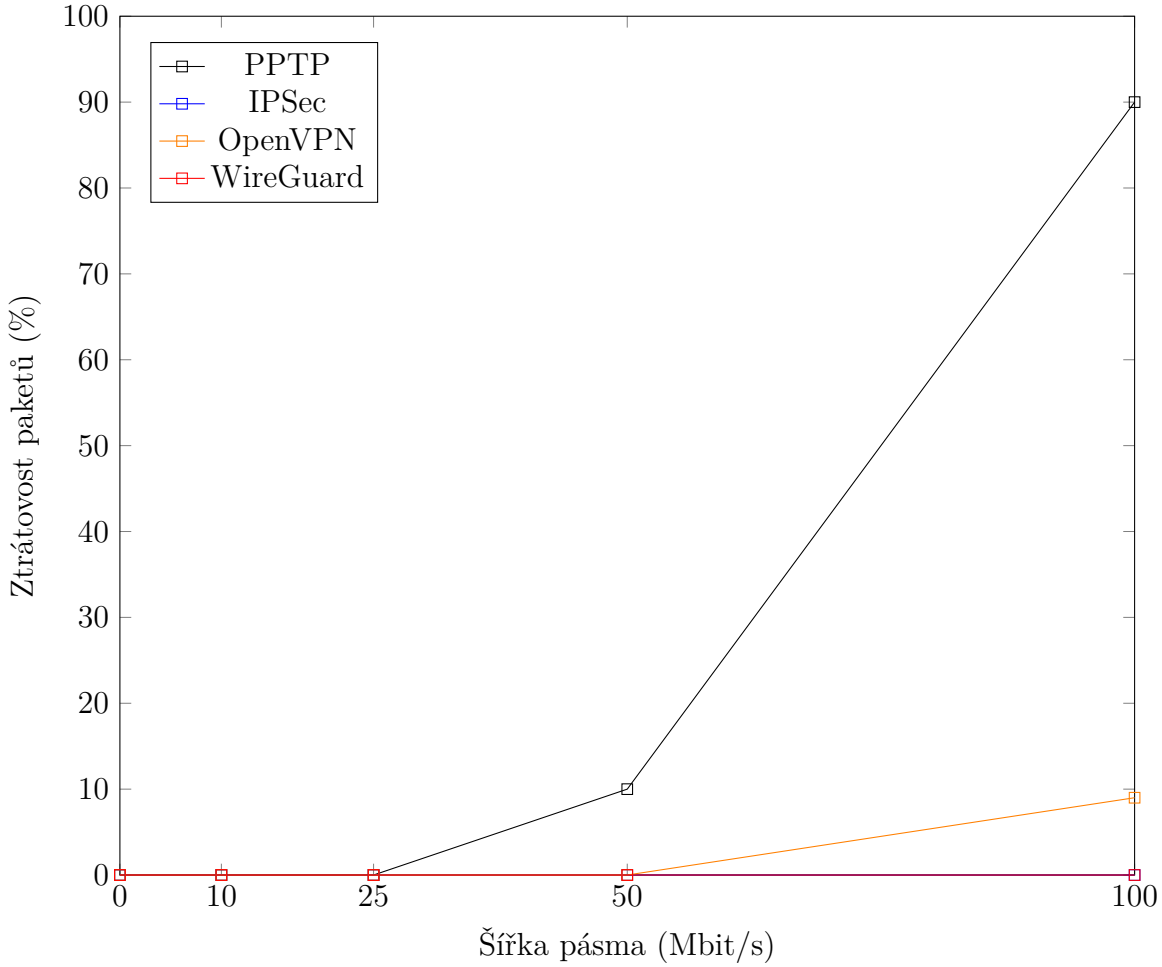
Na základě provedených měření stability lze konstatovat, že všechny testované protokoly dosahují obdobných hodnot *jitter* v jednotkách tisícin milisekund. Protokol WireGuard je co se týče stability nejlepším řešením, jelikož u něj v žádném z testů nedošlo ke ztrátovosti paketů či přijetí paketů ve špatném pořadí. Jako druhý nejstabilnější protokol lze označit protokol IPSec, který dosáhl rovněž hodnot *jitteru* v nižších jednotkách tisícin milisekund a ztrátovost paketů byla ve většině testů nulová. Rovněž se ho netýkalo přijetí paketů v nesprávném pořadí. U protokolů PPTP a OpenVPN byla rovněž naměřena hodnota *jitter* v nižších jednotkách tisícin milisekund, ačkoliv protokol PPTP měl hodnoty průměrně nižší než protokol OpenVPN. U obou těchto protokolů byla zaznamenána ztrátovost paketů, i doručování paketů v nesprávném pořadí.

Srovnání protokolů z hlediska ztrátovosti při jednom, třech a pěti připojených klientech lze vidět ve Grafech 27, 28, 29.

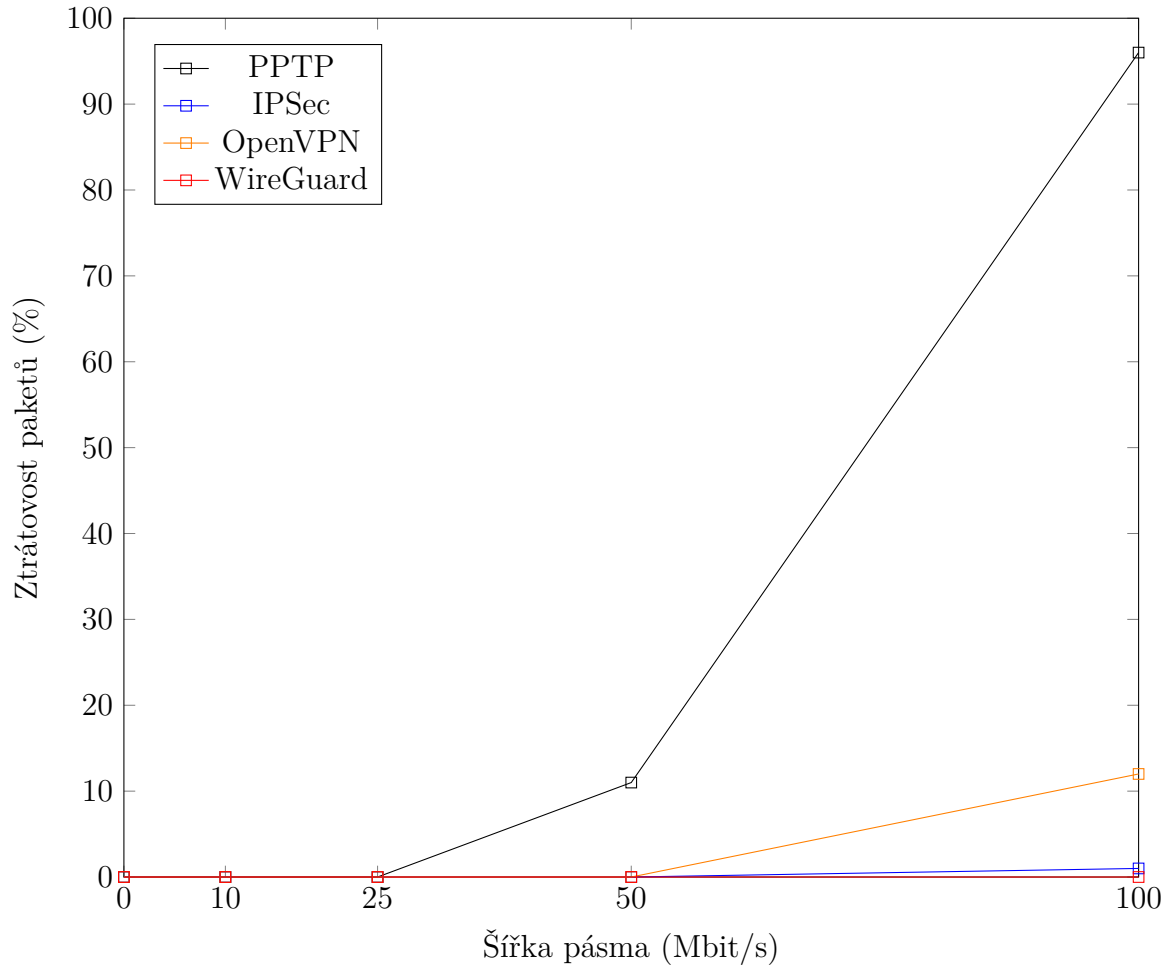
Obrázek 27: Graf průměru ztrátovosti paketů - 1 klient



Obrázek 28: Graf průměru ztrátovosti paketů - 3 klienti



Obrázek 29: Graf průměru ztrátovosti paketů - 5 klientů



7.5 Bezpečnost

Tato kapitola se zabývá porovnáním protokolů na základě celkové bezpečnosti řešení. Zkoumána byla užitá autorizace, šifrování a známé bezpečnostní zranitelnosti.

7.5.1 PPTP

PPTP je z vybraných protokolů tím nejstarším. Vyvíjen byl v devadesátých letech minulého století, kdy v té době nebyla bezpečnost vnímána jako až tak důležitý prvek v síťové komunikaci, jak je tomu nyní.

PPTP k autentizaci využívá protokol MS-CHAP, později MS-CHAPv2. MS-CHAP v obou verzích nelze považovat za bezpečný, jelikož jeho hashovací algoritmus je možné jednoduše prolomit[27]. Tímto útočník může získat hashovaná hesla, která lze následně v krátkém čase rozšifrovat[28].

PPTP používá protokol MPPE k šifrování dat. MPPE má mnoho známých

slabin, které útočníci používají k provádění útoků typu bit-flipping, brute-force a DDoS (Distributed Denial of Service)[29].

Ačkoliv je PPTP jednoduše konfigurovatelný, poměrně rychlý, zároveň podporovaný na většině zařízení, z hlediska bezpečnosti není dobrou volbou kvůli mnoha zranitelnostem. Protokol PPTP je možné užít, pokud není bezpečnostní stránka řešení důležitým faktorem.

7.5.2 IPSec/IKEv2

IPSec řešení lze označit za bezpečné při použití správné kombinace autorizačních a šifrovacích protokolů. Kombinace IPSec/IKEv2 užívá protokol ESP k šifrování hlaviček paketů, SA pro vyjednávání šifrovacích klíčů a šifrovacích algoritmů.

IKEv2 používá šifrovací protokoly jako Blowfish, Camellia, AES 256-bit, které lze považovat za bezpečné.

Další možnou zranitelností může být, stejně jako u ostatních VPN protokolů, užití slabého hesla, které lze prolomit technikou *brute-force*.

Obecně lze kombinaci IPSec/IKEv2 označit za bezpečnou. I z toho důvodu je toto řešení často užíváno v produkčním prostředí[30].

7.5.3 OpenVPN

Protokol OpenVPN je zástupcem open-source protokolů a je stále aktivně vyvíjen a komunitně podporován. OpenVPN je také velmi široce konfigurovatelný. V případě autorizace je možno užít předsdílený klíč, certifikát nebo užít kombinaci jména a hesla. Je také možné užít PKI a od verze 2.0 rovněž kombinaci certifikátu, jména a hesla.

K šifrování je užito knihovny OpenSSL. Nejčastěji užívanými šifrovanými protokoly jsou AES 256-bit a ChaCha[31].

Obecně je OpenVPN považován za velmi bezpečný protokol, avšak záleží na konkrétní konfiguraci, jelikož jak již bylo zmíněno, OpenVPN je vysoce konfigurovatelný, tudíž samotná bezpečnost protokolu se do jisté míry odvíjí od zvolené konfigurace.

7.5.4 WireGuard

WireGuard je ze zvolených protokolů tím nejmladším. WireGuard je považován za bezpečný VPN protokol, jak z důvodu poměrně krátkého zdrojového kodu v porovnání s konkurencí, tak v užití pouze nejbezpečnějších šifrovacích algoritmů, jako jsou ChaCha20, Curve25519, BLAKE2s, SipHash24, HKDF a další. Bezpečnostní audit, který proběhl v roce 2018 nezjistil žádné zranitelnosti ve zdrojovém kódu[32]. Autorizace probíha formou sdílení veřejných klíčů mezi VPN serverem a klientem.

Protokol je také hůře zjistitelný na síti, jelikož nereaguje na žádné pakety, které nepoznává, takže skenování sítě neodhalí, že na systému WireGuard běží.

Navíc spojení mezi klientem a serverem, kteří mohou fungovat jako klienti i servery současně, utichne, když nedochází k výměně dat.

Slabinou protokolu WireGuard je fakt, že klientovi při konfiguraci přiřazujeme vždy stejnou IP adresu, tudíž zde může nastat problém se soukromím. Toto omezení ale některé komerční řešení vyřešily vlastní implementací protokolu[33].

7.6 Náročnost konfigurace

Za řešení s nejsnazší konfigurací lze označit řešení pomocí protokolu PPTP. Toto řešení je poměrně jednoduché, kdy pro jeho funkci stačí nastavit lokální IP adresu serveru, stanovit rozsah IP adres, které budou přiděleny klientům, vytvořit přihlašovací údaje klientů a nastavit DNS. Poté je řešení již funkční. Jsou zde samozřejmě i dotatečné možnosti konfigurace, jako je volba autorizace či šifrování, ale v porovnání s ostatními protokoly, jsou tyto možnosti značně omezené.

Druhým, také poměrně snadným řešení z hlediska konfigurace, je protokol WireGuard. Zde je nutné nastavit IP adresu serveru a klienta, porty na kterých bude server a klient komunikovat a vytvořit privátní a veřejné klíče pro obě strany. Pro správnou funkcni je zároveň nutné upravit pravidla firewall. Poté je řešení již funkční. Protokol WireGuard nabízí velmi dobrou konfigurovatelnost a mnoho konfiguračních možností v porovnání s protokolem PPTP, ale při použití základní konfigurace, bez dodatečných úprav, je samotná konfigurace poměrně snadná a rychlá.

Jako konfiguračně náročnějším protokolem než-li PPTP či WireGuard, lze označit protokol OpenVPN. Zde je nutné generovat několik certifikátů a klíčů, jak na straně serveru, tak na straně klienta. Samotný konfigurační soubor na straně klienta i serveru je poměrně obsáhlý s velkým množstvím atributů a konfiguračních možností. Stejně jako u již zmíněných protokolů, je zde také nutné upravit pravidla firewall dle potřeby. Nevýhodou a zároveň i výhodou protokolu OpenVPN jsou opravdu široké možnosti konfugirace. Další výhodou řešení OpenVPN je velmi dobře zpracovaná dokumentace a také to, že z důvodu, že je tento protokol velmi oblíben, zejména z důvodu bezpečnosti a konfiguračních možností, je možné mnohá řešení problémů s konfigurací nalézt na diskuzních fórech či jiných zdrojích.

Posledním, subjektivně nejnáročnějším protokolem z hlediska konfigurace, je protokol IPSec v kombinaci s IKEv2. Svou konfigurací se příliš neliší od protokolu OpenVPN. Stejně jako u OpenVPN, tak i u protokolu IPSec/IKEv2, je nutné vygenerování klíčů a certifikátů pro server i klienta. Nutností je také úprava samotných firewall pravidel. Jedním z důvodu proč bylo řešení IPSec/IKEv2 označeno jako náročnější na konfigurací v porovnání s druhým nejnárovnějším, OpenVPN, je samotný konfigurační soubor, který není tak snadno čitelný a jednoznačný, jako u protokolu OpenVPN. Dalším důvodem je hůře zpracovaná dokumentace v porovnání s ostatními VPN protokoly.

8 Propustnost na slabších zařízeních

Ačkoliv nejčastěji užívaným VPN klientem býva uživatelská stanice, klientem může být i mobilní telefon či jednodeskové zařízení. V tomto případě byla testována datová propustnost na zvolených zařízeních. Jako zástupce OS Andriod byl zvolen mobilní telefon Samsung A52S 5G, jakožto zástupce střední třídy telefonů a Xiaomi Redmi Note 7, jakožto zástupce telefonů starších pěti let, zároveň telefon patřící do nižší třídy. Jako zástupnce iOS byl zvolen mobilní telefon iPhone 13 Mini. V případě jednodeskového zařízení bylo užito Raspberry Pi 3, verze s 2GB operační paměti.

Při testování, v případě mobilních telefonů, bylo užito aplikace *Speedtest by Ookla*[34]. U Rasperry Pi bylo užito CLI (Command Line) alternativy - *speedtest-cli*, vyvinuté stejnou společností.

K testování bylo užito domácí WiFi sítě využívající pásma 2.4 Ghz. 5 Ghz pásmo nebylo užito z důvodu nepodpory u zařízení Raspberry Pi. Při testování byly všechny zařízení ve stejné vzdálenosti od směřovače z důvodu vytvoření stejných testovacích podmínek pro všechna zařízení.

Před samotným testování propustnosti VPN protokolů na zvolených zařízeních, byla měřena celková propustnost bez užití VPN.

Při testu propustnosti bez užití VPN byly výsledky zvolených zařízení velmi podobné. Největší propustnost byla naměřena u mobilního telefonu iPhone 13 Mini a to 51.11 Mbit/s. Druhým nejvýkonějším byl Samsung A52 5G s výsledkem 50.59 Mbit/s. Na třetím místě se umístilo jednodeskové zařízení Raspberry Pi s naměřenou hodnotou 48.47 Mbit/s. Nejméně výkonným v testu bez užití VPN se ukázal mobilní telefon Xiaomi Redmi Note 7 s naměřenými 45.52 Mbit/s.

V jednotlivých testech datové propustnosti výsledky prakticky kopírovaly výsledky měření bez užití VPN. Jediným rozdílem byla výkonnost Raspberry Pi, která bez užití VPN dosahovala výsledků totožných s testovanými mobilními telefony, ale při užití VPN se již projevovala nedostatečná výkonnost a ve všech testech se Raspberry Pi umístila na posledním místě, místy se značnými výkonovými rozdíly od ostatních testovaných zařízení.

Za nejvýkonější telefony z hlediska datové propustnosti při užití VPN lze označit iPhone 13 Mini a Samsung A52S 5G, kteří se v jednotlivých testech umisťovali na prvních místech.

Celkové výsledky měření lze vidět v Tabulce 19. Zelené hodnoty značí nejvýkonější zařízení v daném testu. Červené hodnoty značí nejméně výkonné zařízení u jednotlivého VPN protokolu.

Výkonnost slabších klientských zařízení				
Protokol	Xiaomi	Samsung A52S	iPhone 13 Mini	Raspberry Pi
Bez VPN	45.52	50.69	51.11	48.47
PPTP	45.11	47.38	47.97	25.86
IPSec	45.39	48.11	47.56	31.12
OpenVPN	43.12	47.11	49.03	33.45
WireGuard	45.47	49.02	47.11	38.94

Tabulka 19: Výkonnost slabších klientských zařízení

Při užití Raspberry Pi jako klienta, lze vidět, že jeho celková datová propustnost razantně klesá s užitím VPN. Při užití VPN jako serveru s více klienty lze předpokládat, že zařízení nebude schopné obsluhovat všechny klienty dostatečně rychle, bez značné ztráty datové propustnosti. V otázce zda je Raspberry vhodné jako VPN řešení pro produkční prostředí, lze na základě testů s určitou jistotou konstatovat, že tomu tak není z důvodů nedostatečné výkonnosti.

9 Dostupnost klientů pro různé platformy

Dostupnost klientů byla zkoumána s ohledem na několik operačních systémů. Konkrétně se jedná o Linux, Windows 10/11, iOS a Android.

Obecně lze říci, že dostupnost klientů na všech zkoumaných operačních systémech není v dnešní době problém a veškeré zmiňované operační systémy mají svého klienta pro daný VPN protokol. V případě operačních systémů Linux a Windows, které jsou primárně určeny pro desktopová zařízení, mají zpravidla mimo GUI i CLI variantu klienta.

9.1 Linux

V případě Linuxu jsou dostupní klienti pro všechna vybraná VPN řešení. Na výběr jsou komerční řešení pomocí GUI, ale také zejména pomocí CLI, která bývají častěji open-source.

9.1.1 PPTP

Klient pro protokol PPTP bývá v základu již součástí OS, záleží na konkrétní distribuci. V případě OS Ubuntu 22.04 je součástí klient pro GUI, společně s CLI klientem. Pri užití CLI varianty klienta je možné užít program *pptp* či *pptpsetup*. Oba programy jsou součástí oficiálního repozitáře operačního systému.

9.1.2 IPSec

V případě IPSec je možné využít několik klientů, jak GUI, tak CLI. Z GUI klientů lze zmínit komerční řešení *FortiVPN* a *Cisco AnyConnect*. CLI klient je

zde na výběr ze trojice *strongswan*, *libreswan* a *openswan*. V těchto případech se jedná o velmi podobná řešení, která v určitém vývojovém cyklu byla rozdělena v závislosti na vývojovém týmu a společnosti, která jej vyvíjela. U všech tří variant se jedná o open-source. Také je možné užít programu *softether*, což je multiplatformní aplikace. Aplikace je rovněž open-source.

9.1.3 OpenVPN

OpenVPN se v Linuxu užívá výhradně pomocí CLI, kdy je možné využít programu *openvpn* či GUI konfigurační varianty řešené balíčkem *network-manager-openvpn*.

9.1.4 WireGuard

WireGuard je rovněž podporován pouze ve své CLI variantě a to za pomocí balíčku *wireguard* a *wireguard-tools*.

9.2 Windows

Pro OS od společnosti Microsoft, jsou rovněž dostupní klienti pro všechna vybraná VPN řešení. Jedná se zejména o komerční řešení pomocí GUI. CLI varianta v případě OS Windows nebývá často preferována.

9.2.1 PPTP

Klient pro protokol PPTP je v základu již součástí OS, jelikož se jedná o protokol, který byl vyvinut rovněž společností Microsoft. Preferována je varianta pomocí GUI, ačkoliv je zde možnost i varianty CLI.

9.2.2 IPSec

I v případě OS Windows je zde možno zvolit z několika klientů, zejména pomocí GUI. Windows 10/11 podporuje IPSec již v základu, tudíž není nutné instalovat jakéhokoliv klienta. Mimo této možnosti lze zmínit komerční řešení *FortiVPN* či *Cisco AnyConnect*. Stejně jako v Linuxu, i zde je možné užít programu *softether*, což je multiplatformní aplikace, která má i své grafické rozhraní. Důvodem pro užití jiného klienta, než-li toho, který je již součástí OS samotného, bývá potřeba konkrétní konfigurace, kterou základní klient nemusí umožňovat.

9.2.3 OpenVPN

K užití OpenVPN u systému Windows se užívá zejména GUI varianty klienta. Jedná se o oficiální program od společnosti OpenVPN - *OpenVPN Client Connect*.

9.2.4 WireGuard

WireGuard v případě užití na na OS Windows užívá výhradně svou grafickou varitantu uživatelského prostředí. Stejnojmenná aplikace *WireGuard* je k dispozici na oficiálních stránkách společnosti.

9.3 iOS

Apple a jeho operační systém iOS rovněž nabízí klienty pro všechny testované VPN protokoly. Jelikož se jedná o operační systém, který je využíván čistě mobilními telefony, CLI klienti nejsou v tomto případě dostupní.

9.3.1 PPTP

Operační systém iOS má již v základu integrovanou podporu protokolu PPTP. Není nutno tedy instalovat jakýkoliv další software.

9.3.2 IPSec

IPSec je u tohoto operačního systému podporován již v základu, tudíž není nutné instalovat jakéhokoliv klienta. Mimo této možnosti lze zmínit komerční řešení *FortiVPN* či *Cisco AnyConnect*. V případě iOS je také možno využit open-source klienta *strongswan*.

9.3.3 OpenVPN

Pro používání protokolu OpenVPN na operačním sytému iOS je dostupný opens-source klient *OpenVPN Connect*. Klient je dostupný ke stažení na Apple Store.

9.3.4 WireGuard

Stejně jako OpenVPN, protokol WireGuard nabízí oficiálního klienta dostupného na Apple Store - *WireGuard iOS*.

9.4 Android

Stejně jako iOS, Android také nabízí klienty pro všechny testované VPN protokoly. V případě operačního systému Andriod se také jedná o systém, který je využíván výhradně mobilními telefony, CLI klienti nejsou v tomto případě dostupní.

9.4.1 PPTP

Android má již v základu integrovanou podporu protokolu PPTP. Není nutno tedy instalovat jakýkoliv další software.

9.4.2 IPSec

IPSec je u tohoto operačního systému podporován již v základu, tudíž není nutné instalovat jakéhokoliv klienta. Mimo této možnosti lze zmínit komerční řešení *FortiVPN* či *Cisco AnyConnect*. Stejně jako u iOS, tak i Android nabízí možnost využít open-source řešení *strongswan*, který je ke stažení na Play Store.

9.4.3 OpenVPN

Android nabízí opens-source klienta *OpenVPN for Android*. Klient je dostupný ke stažení na Play Store.

9.4.4 WireGuard

Stejně jako OpenVPN, protokol WireGuard nabízí oficiálního klienta dostupného na Apple Store - *WireGuard iOS*.

10 Doporučené užití jednotlivých protokolů

Na základě provedených testů a zjištění lze říci, že každý protokol může mít své vhodné užití, kde může být vhodnějším řešením, než-li ostatní.

V případě PPTP je velkou výhodou jeho podpora, která je rozšířená na většinu zařízení. Nespornou výhodou je také jednoduchost konfigurace a ve většině případů zde také není nutnost instalace jakéhokoliv klienta. Nevýhodou je opravdu slabé zabezpečení a horší výkon ve srovnání s ostatními zkoumanými VPN protokoly. Vhodné užití by mohlo být v domácích podmínkách, či při testování, kde bezpečnost není důležitým faktorem. PPTP protokol je nevhodný pro firemní užití, primárně z jeho nedostatečného zabezpečení.

Protokol IPSec je rychlým, v kombinaci IPSec/IKEv2 také velmi bezpečným protokolem, který zároveň není příliš hardwarově náročný i při vyšší zátěži. Zároveň jeho stabilita je na vvysoké úrovni. Nevýhodou je zde složitější konfigurace. Pro svou rychlost, stabilitu a vysokou bezpečnost, je vhodným pro užití ve firemní síti, kde je důležitým faktorem bezpečnost, rychlost a stabilita. Z uvedených důvodů je protokol často volbou při *site-to-site* připojení.

OpenVPN je rovněž velmi rychlý a stabilní, ačkoliv nedosahuje přenosových rychlostí protokolu IPSec. Z bezpečnostního hlediska je protokol na vysoké úrovni, jelikož podporuje řadu bezpečnostních prvků, nejnovější šifrovací metody a hashovací funkce. Jelikož se jedná o open-source aplikaci, dochází ke kontrolám kodu ze strany mnoha uživatelů a společností, což potencionálně také zlepšuje celkové zabezpečení. OpenVPN je rovněž velmi dobře konfigurovatelný a nabízí nejvíce konfiguračních možností ze všech testovaných protokolů. OpenVPN je vhodné užít jak v domácím, tak zejména ve firemním prostředí, kde může být preferován pro specifické konfigurace, které ostatní VPN protokoly nemusí neumožňovat.

Posledním testovaným protolem byl protokol WireGuard. Nejmladší z protokolů dosahoval nejlepších výsledků v testování celkové výkonnosti a stability. Bezpečnost v podání protokolu WireGuard je také na velmi vysoké úrovni, zejména z důvodu podpory nejnovějších šifrovacích metod a také faktu, že se jedná o open-source aplikaci, pro kterou platí, to samé jako například pro zmiňovaný OpenVPN. Vhodné užití je zejména v kombinaci s operačním systémem Linux, jelikož samotný WireGuard běži v jeho jádře. Protokol WireGuard je vhodné používat jak v domácím prostředí, tak ve firemním, s ohledem na velmi snadnou konfiguraci, široké konfigurační možnosti a vysokou úroveň zabezpečení.

Závěr

Cílem bakalářské práce bylo podrobné seznámení s technologií virtuálních privátních sítí, srovnat vybrané protokoly dle vybraných kritérií a určit jejich vhodné užití.

Na základě teoretické části lze konstatovat, že nejstarší protokol PPTP není vhodný pro firemní použití kvůli jeho nedostatečnému zabezpečení. Tento protokol je možné užít v domácím či testovacím prostředí, kde není kladen důraz na zabezpečení komunikace. Protokoly IPSec/IKEv2, OpenVPN či WireGuard lze doporučit pro použití i ve firemním prostředí, jelikož splní požadavky na bezpečnost z důvodu užití moderních šifrovacích a hashovacích metod a podpory dalších bezpečtnostních prvků, jako je například 2FA.

V praktické části byly zvolené protokoly nakonfigurovány a následně otestovány. Zkoumána byla jejich odezva, datová propustnost, hardwarová zátěž, stabilita spojení a celková bezpečnost. Následně byla zhodnocena také náročnost konfigurace na straně klienta a serveru.

Dle provedených testů bylo zjištěno, že co se týče celkové odezvy jednotlivých VPN protokolů, nejsou zde podstatné rozdíly. V testu propustnosti byl zjištěn protokol WireGuard jako nejvýkonější, s jistým odstupem od IPSec či OpenVPN. Ač byl protokol PPTP označován jako rychlý, v testu propustnosti byl zdaleka nejméně výkonný. Co se týče hardwarové náročnosti z hlediska CPU, je možné konstatovat, že protokoly IPSec či WireGuard jsou nejméně náročné, v závislosti na daném testovacím scénáři. Při testu celkového zatížení RAM nebyly mezi protokoly zjištěny výraznější rozdíly. V testu celkové stability spojení lze označit protokol WireGuard jako nejstabilnější, jelikož u něj nedocházelo k žadným ztrátám paketů či příjmu paketů ve špatném pořadí. Protokol IPSec byl v testu stability také velmi dobrý, jeho ztrátovost paketů se pohybovala maximálně v řádu desetin procenta. Protokoly PPTP a OpenVPN měly ztrátovost vyšší a příjem paketů v jiném pořadí se zde objevoval také častěji. S ohledem na zabezpečení zvolených VPN protokolů, lze na základě zjištění konstatovat, že s výjimkou protokolu PPTP, lze všechny protokoly považovat za bezpečné. Je tomu z důvodu užití moderních šifrovacích metod v kombinaci s možností užití dalších bezpečtnostních prvků.

V následující kapitole byla zjišťována datová propustnost na vybraných slabších zařízeních, konkrétně mobilních telefonech a jednodeskovém zařízení. Měřením bylo zjištěno, že hlavní limitací v propustnosti na daném zařízení, není v případě mobilních telefonů konkrétní VPN řešení, ale hardwarové možnosti telefonu. V případě novějších a hardwarově výkonnějších telefonů, byla naměřená propustnost vždy vyšší, bez ohledu na protokol, než u telefonu staršího, méně hardwarově výkonného. Co se týče výsledků měření na Raspberry Pi, lze konstatovat, že tato varianta není příliš vhodná k použití jako VPN klient či server, právě z výkonových důvodů.

V poslední kapitole byla zjišťována dostupnost klientů pro zvolené VPN protokoly v závislosti na operačním systému. Výsledkem je fakt, že žádný z vybra-

ných VPN protokolů nelimituje nedostupnost klienta. V mnoha případech instalace klienta není ani nutná z důvodu podpory protokolu operačním systémem již v základu.

Závěřem lze říci, že s výjimkou nejstaršího protokolu PPTP, zejména z důvodu nedostatečného zabezpečení, lze všechny zvolené VPN protokoly používat v domácím i firemním prostředí. Protokoly dosahují vysokých přenosových rychlosti, vysoké stability a rovněž při správné konfiguraci, i velmi dobrého zabezpečení. Limitace z pohledu klientů pro různé platformy zde prakticky neexistuje. Při volbě protokolu, který bude užit je důležité zohlednit konkrétní cíl, kterého chceme dosáhnout. Pokud by se jednalo o propojení dvou systémů běžící na operačním systému Linux, je vhodným kandidátem protokol WireGuard, jelikož je velmi rychlý a běží přímo v jádře operačního systému. Pokud by se jednalo o propojení dvou vzdálených systémů, například konfigurací na firewallu, může být vhodným řešením právě IPSec, jelikož je často přímo na těchto zařízeních podporován, zároveň má vysokou přenosovou rychlost a vysokou úroveň zabezpečení při kombinaci s IKEv2. OpenVPN lze užít při nutnosti specifické konfigurace, kterou nemusí ostatní protokoly podporovat.

Ačkoliv mohou být zvolené protokoly rozdílné svou funkcí a obecnému pojetí konfigurace, lze konstatovat, že volba daného protokolu by měla být založena na konkrétním cíli, kterého chceme dosáhnout, s přihlédnutím k technickým možnostem daného VPN protokolu a zkušenostem, které již s protokolem máme.

Conclusions

The goal of the Bachelor thesis was getting to know virtual private technologies, compare selected solutions and choose the ideal use case for each one.

Based on the theoretical part we can assume that the oldest protocol - PPTP is not an ideal for production environment, mainly for the security issues that the protocol is facing. PPTP can be used in lab or home environment, where the security is not such an important part. Protocols IPSec/IKEv2, OpenVPN or WireGuard can be recommended to use in production environment as they match the security standards, mainly because of using the latest encryption methods or support of other security elements, e.g. 2FA.

In the practical part of the thesis, the selected protocols were configured and tested. Multiple factors were tested - response time, data bandwidth, hardware requirements and security. Also the complexity of the configuration was evaluated.

Based on the tests, it was found that the response time of the protocols was almost the same. In the test of data bandwidth it was found that WireGuard protocol is the fastest one with the certaion gap over IPSec or OpenVPN. As the PPTP was said to be simple and fast one, the tests proved otherwise, meaning PPTP protocol was the slowest from all the selected ones.

Regarding hardware demands from the CPU point of view, it can be stated that the IPSec or WireGuard protocols are the least demanding, depending on the given test scenario. In the total RAM load test, no significant differences were found between the protocols. In the test of the overall stability of the connection, the WireGuard protocol can be characterized as the most stable, as there were no packet losses or packet reception in the wrong order. The IPSec protocol was also very good in the stability test, its packet loss was at most in the order of tenths of a percent. The PPTP and OpenVPN protocols had a higher loss rate, and reception of packets in a different order appeared more often here. With regard to the security of the selected VPN protocols, based on the findings, it can be concluded that with the exception of the PPTP protocol, all protocols can be considered safe. This is due to the use of modern encryption methods in combination with the possibility of using other security elements.

In the following chapter, data throughput was determined on selected weaker devices, namely mobile phones and single-board devices. The measurement revealed that the main limitation in throughput on a given device, in the case of mobile phones, is not a specific VPN solution, but the phone's hardware capabilities. In the case of newer and more powerful hardware phones, the measured throughput was always higher, regardless of the protocol, than for an older, less powerful phone. As for the measurement results on the Raspberry Pi, it can be stated that it is not very suitable for use as a VPN client or server, mainly for performance reasons.

In the last chapter, the availability of clients for selected VPN protocols was determined depending on the operating system. The result is the fact that none

of the selected VPN protocols limit the unavailability of the client. In many cases, the installation of the client is not even necessary due to the fact that the protocol is already supported by the operating system.

In conclusion, we can say that with the exception of the oldest PPTP protocol, mainly due to insufficient security, all selected VPN protocols can be used in both home and business environments. The protocols achieve high transmission speeds, high stability and, with the correct configuration, also very good security. There is practically no limitation from the point of view of clients for different platforms. When choosing the protocol that will be used, it is important to take into account the specific goal that we want to achieve. If it were to connect two systems running on the Linux operating system, the WireGuard protocol is a suitable candidate, as it is very fast and runs directly in the operating system kernel. If it were to connect two remote systems, for example through a configuration on a firewall, IPSec may be a suitable solution, as it is often directly supported on these devices, and at the same time has a high transmission speed and a high level of security when combined with IKEv2. OpenVPN can be used when a specific configuration is required, which may not be supported by other protocols.

Although the chosen protocols may differ in their function and general concept of configuration, it can be stated that the choice of a given protocol should be based on the specific goal that we want to achieve, taking into account the technical capabilities of the given VPN protocol and the experience we already have with the protocol.

A Obsah elektronických dat

doc/
: Obsahuje text práce ve formátu .pdf, zdrojové texty v archivu .zip.

data/
: Obsahuje jednotlivé výsledky měření při testování.

Literatura

[1] Mairs, John L.; Mueller, Michael; Sieben, Jackie. *Vpns: A Beginner's Guide*. 2001. ISBN 0072191813.

[2] Chan, Conrad; Dao, Anthony; Hou, Justin; Jin, Tony; Tuong, Calvin. Dostupný také z: ⟨https://cs.stanford.edu/people/eroberts/cs181/projects/2010-11/FreeExpressionVsSocialCohesion/china_policy.html⟩.

[3] *VPN hardware vs. VPN Software*. [online]. 2022 [cit. 2022-11-3]. Dostupný z: ⟨https://openvpn.net/solutions/use-cases/vpn-hardware-vs-vpn-software/⟩.

[4] Risukhin, Artem. Hardware VPN vs software VPN. *Macpaw* [online]. 2021, [cit. 2022-11-3]. Dostupný z: ⟨https://macpaw.com/how-to/hardware-vpn-vs-software-vpn/⟩.

[5] Shacklett, Mary E.; Rosencrance, Linda. *What is authentication?* [online]. 2021 [cit. 2022-11-3]. Dostupný z: ⟨https://www.techtarget.com/searchsecurity/definition/authentication⟩.

[6] Balasooriya, Umesha. Computer Network Models. *medium.com* [online]. 2022, [cit. 2023-5-28]. Dostupný z: ⟨https://umeshabalasooriya.medium.com/computer-network-models-94e64546d5c1⟩.

[7] Mairs, John L.; Mueller, Michael; Sieben, Jackie. 2001. Layering Principles, s. 4–4. ISBN 0072191813.

[8] *VPNS illustrated: Tunnels, vpns, and IPsec*. Snader, J. C. 2006. 2.3 Encapsulation, s. 11–11.

[9] *VPNS illustrated: Tunnels, vpns, and IPsec*. Snader, J. C. 2006. 2.3 Encapsulation, s. 4.

[10] Reed, Jessica. What is a Bit Stream. *EasyTechJunkie* [online]. 2023, [cit. 2023-2-25]. Dostupný z: ⟨https://www.easytechjunkie.com/what-is-a-bit-stream.htm⟩.

[11] Critical Section in Synchronization. *geeksforgeeks* [online]. [Cit. 2023-5-26]. Dostupný z: ⟨https://www.geeksforgeeks.org/g-fact-70/⟩.

[12] Gordon, Don. PPTP VPN Security Risks. *myworkdrive* [online]. [Cit. 2023-5-22]. Dostupný z: ⟨https://www.myworkdrive.com/vpn-alternative/pptp-security-risks/⟩.

[13] Rodier, Rob. Point-to-Point (P2P) Connectivity: What You Need To Know. *lightyear.ai* [online]. [Cit. 2023-5-26]. Dostupný z: ⟨https://lightyear.ai/blogs/point-to-point-connectivity⟩.

[14] *VPNS illustrated: Tunnels, vpns, and IPsec*. Snader, J. C. 2006. L2F, s. 274.

[15] Ashraf, Zeeshan, 2018. Virtual Private Networks in Theory and Practice, s. 25. ISBN 9786202309899.

[16] CONSTANTINESCU, Vlad. Most Popular VPN Connection Protocols. *BitDefender* [online]. 2022, [cit. 2023-3-17]. Dostupný z: ⟨https://www.bitdefender.com/blog/hotforsecurity/most-popular-vpn-connection-protocols-explained/⟩.

[17] OpenVPN. Community Downloads. *OpenVPN* [online]. [Cit. 2023-3-19]. Dostupný z: ⟨https://openvpn.net/community-downloads/⟩.

[18] Proxmox. *proxmox.com* [online]. [Cit. 2023-5-28]. Dostupný z: ⟨https://proxmox.com/en⟩.

[19] linux.die.net. netstat manual page. *linux.die.net* [online]. [Cit. 2023-4-2]. Dostupný z: ⟨https://linux.die.net/man/8/netstat⟩.

[20] linux.die.net. chmod manual page. *linux.die.net* [online]. [Cit. 2023-4-2]. Dostupný z: ⟨https://linux.die.net/man/1/chmod⟩.

[21] linux.die.net. ipsec.conf manual page. *linux.die.net* [online]. [Cit. 2023-4-2]. Dostupný z: ⟨https://linux.die.net/man/5/ipsec.conf⟩.

[22] linux.die.net. ipsec.secrets manual page. *linux.die.net* [online]. [Cit. 2023-4-2]. Dostupný z: ⟨https://linux.die.net/man/5/ipsec.secrets⟩.

[23] Ubuntu. UFW - Community Help Wiki. *help.ubuntu.com* [online]. [Cit. 2023-4-3]. Dostupný z: ⟨https://help.ubuntu.com/community/UFW⟩.

[24] linux.die.net. IPerf manual page. *linux.die.net* [online]. [Cit. 2023-4-10]. Dostupný z: ⟨https://linux.die.net/man/1/iperf⟩.

[25] Godard, Sebastien. sysstat manual page. *man7.org* [online]. [Cit. 2023-4-10]. Dostupný z: ⟨https://man7.org/linux/man-pages/man5/sysstat.5.html⟩.

[26] Ansible. *ansible.com* [online]. [Cit. 2023-5-28]. Dostupný z: ⟨https://ansible.com⟩.

[27] Scott, Olivia. The PPTP VPN protocol: Is it safe? *Infosec Institute* [online]. 2019, [cit. 2023-5-8]. Dostupný z: ⟨https://resources.infosecinstitute.com/topic/the-pptp-vpn-protocol-is-it-safe/⟩.

[28] Schneier, B.; Mudge. Cryptanalysis of Microsoft's Point-to-Point Tunneling Protocol (PPTP). *Schneier on Security* [online]. 1998, [cit. 2023-5-8]. Dostupný z: ⟨https://www.schneier.com/academic/archives/1998/11/cryptanalysis_of_mic.html⟩.

[29] Hensen, Kristi. Everything You Need To Know About PPTP protocol in 2023. *PrivacyHub* [online]. 2023, [cit. 2023-5-8]. Dostupný z: ⟨https://www.cyberghostvpn.com/en_US/privacyhub/pptp-vpn/⟩.

[30] Vojinovic, Ivana. What is IKEv2 VPN Protocol Is it Secure? *DataProt* [online]. 2023, [cit. 2023-5-8]. Dostupný z: ⟨https://dataprot.net/guides/what-is-ikev2-vpn/⟩.

[31] Dahan, Marc. What is OpenVPN? Is OpenVPN safe? *Comparitech* [online]. 2021, [cit. 2023-5-8]. Dostupný z: ⟨https://www.comparitech.com/blog/vpn-privacy/what-is-openvpn/⟩.

[32] Formal Verification - WireGuard. *wireguard.com* [online]. [Cit. 2023-5-28]. Dostupný z: ⟨https://www.wireguard.com/formal-verification/⟩.

[33] Sutherland, Richard. Is the new WireGuard protocol secure? *Tomsguide* [online]. 2021, [cit. 2023-5-8]. Dostupný z: ⟨https://www.tomsguide.com/how-to/is-the-new-wireguard-protocol-secure⟩.

[34] Speetest by Ookla. *play.google.com* [online]. [Cit. 2023-6-12]. Dostupný z: ⟨https://play.google.com/store/apps/details?id=org.zwanoo.android.speedtest&hl=en_US&pli=1⟩.

[35] Brown, Aaron. *Best VPN: What's the best VPN service? NordVPN, Express VPN, Surfshark, and more RATED*. 2022. Name - Warner Media LLC; Netflix Inc; Copyright - Copyright Express Newspapers PLC Nov 16, 2022; Last updated - 2022-11-17. Dostupný také z: ⟨https://www.proquest.com/newspapers/best-vpn-whats-service-nordvpn-express-surfshark/docview/2736934061/se-2⟩.

[36] Hummel, Robert L. How It Works: Virtual Private Network. *PC World.Com*. 2000, s. 1–3. Copyright - (PC World (c) 2000; Last updated - 2011-10-21. Dostupný také z: ⟨https://www.proquest.com/trade-journals/how-works-virtual-private-network/docview/200761305/se-2⟩.

[37] VPN hardware vs. VPN Software. *Surfshark* [online]. 2022, [cit. 2022-11-3]. Dostupný z: ⟨https://surfshark.com/blog/hardware-vpn/⟩.

[38] Amin, Romj. What Is Geo Blocking and How Does it Work? *Techjury* [online]. 2022, [cit. 2022-11-4]. Dostupný z: ⟨https://techjury.net/blog/what-is-geo-blocking/⟩.

[39] Aliza Vigderman, Gabe Turner. *NordVPN Review* [online]. 2022 [cit. 2022-11-4]. Dostupný z: ⟨https://www.security.org/vpn/nordvpn/review/⟩.