

Konfigurace a propustnost jednotlivých dostupných VPN

Bc. Milan Hanuliak



Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav elektroniky a měření

Akademický rok: 2024/2025

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	Bc. Milan Hanuliak
Osobní číslo:	A23370
Studijní program:	N1032A020003 Bezpečnostní technologie, systémy a management
Specializace:	Bezpečnostní technologie
Forma studia:	Prezenční
Téma práce:	Konfigurace a propustnost jednotlivých dostupných VPN
Téma práce anglicky:	Configuration and Throughput of the Individual VPNs Available

Zásady pro vypracování

1. Zpracujte literární rešerši na dané téma.
2. Nakonfigurujte vybrané VPN protokoly (OpenVPN, IKEv2, L2TP, SSTP, WireGuard) ve vaší síti.
3. Provedte testy propustnosti jednotlivých VPN protokolů.
4. Analyzujte a porovnejte datovou propustnost jednotlivých VPN protokolů.
5. Vyhodnoťte vliv různých konfigurací VPN na celkový výkon sítě.

Forma zpracování diplomové práce: **tištěná/elektronická**
Jazyk zpracování: **Slovenština**

Seznam doporučené literatury:

1. VPN Technologies: Definitions and Requirements. Virtual Private Network Consortium – VPNC [online]. 2008 [cit. 2024-10-04]. Dostupné z: <http://www.vpnc.org/vpn-technologies.html>
2. Secure Remote Access (SSL VPN). Technology, Life and other stuff that come along [online]. 2006 [cit. 2024-10-04]. Dostupné z: <http://nirlog.com/2006/01/23/secure-remote-access-ssl-vpn/>
3. Srovnání VPN Protokolů: PPTP vs. L2TP vs. OpenVPN vs. SSTP vs. IKEv2 [online]. [cit. 2024-10-04]. Dostupné z: <https://cs.vpnmentor.com/blog/srovnani-vpn-protokolu-pptp-vs-l2tp-vs-openvpn-vs-sstp-vs-ikev2/>
4. SoftEther VPN [online]. [cit. 2024-10-04]. Dostupné z: <https://www.softether.org/>
5. RFC 5996 – "Internet Key Exchange Protocol Version 2 (IKEv2)" [online]. [cit. 2024-10-25]. Dostupné z: <https://tools.ietf.org/html/rfc5996>
6. RFC 4301 – "Security Architecture for the Internet Protocol" [online]. [cit. 2024-10-25]. Dostupné z: <https://tools.ietf.org/html/rfc4301>
7. RFC 2409 – "The Internet Key Exchange (IKE)" [online]. [cit. 2024-10-25]. Dostupné z: <https://tools.ietf.org/html/rfc2409>

Vedoucí diplomové práce: **Ing. Miroslav Matýšek, Ph.D.**
Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce: **29. listopadu 2024**
Termín odevzdání diplomové práce: **30. května 2025**

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan



Ing. Milan Navrátil, Ph.D. v.r.
ředitel ústavu

Ve Zlíně dne 29. listopadu 2024

PROHLÁŠENÍ AUTORA DIPLOMOVÉ PRÁCE

Beru na vědomí, že

- odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., v platném znění bez ohledu na výsledek obhajoby;
- diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému a bude dostupná k nahlédnutí;
- jedno vyhotovení diplomové práce v listinné podobě bude ponecháno Univerzitě Tomáše Bati ve Zlíně k uložení;
- na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – diplomovou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá; neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji, že

- jsem na diplomové práci pracoval(a) samostatně a použitou literaturu jsem řádně citoval(a); v případě publikace výsledků budu uveden(a) jako spoluautor;
- odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.

Ve Zlíně, dne

.....
podpis autora

ABSTRAKT

Táto práca sa zaoberá analýzou výkonnostných charakteristík vybraných VPN protokolov so zameraním na ich konfiguráciu a dátovú priepustnosť. Cieľom bolo experimentálne porovnať výkon protokolov OpenVPN, IKEv2, L2TP/IPsec, SSTP a WireGuard v jednotnom testovacom prostredí. Po spracovaní teoretickej rešerše boli všetky protokoly nakonfigurované a podrobené meraniam. Práca zároveň poukazuje na význam konfigurácie, typu protokolu a využitia UDP pri optimalizácii VPN výkonu. Zistenia poskytujú praktické odporúčania pre výber a nasadenie VPN riešení s dôrazom na priepustnosť, bezpečnosť a efektivitu.

Kľúčová slova: Wireguard, L2TP, SSTP, OpenVPN, IKEv2

ABSTRACT

This thesis analyzes the performance characteristics of selected VPN protocols, focusing on their configuration and data throughput. The aim was to experimentally compare the performance of OpenVPN, IKEv2, L2TP/IPsec, SSTP, and WireGuard protocols within a uniform testing environment. After reviewing theoretical research, all protocols were configured and subjected to measurements. The thesis also highlights the significance of configuration, protocol type, and UDP utilization in optimizing VPN performance. The findings offer practical recommendations for selecting and deploying VPN solutions, emphasizing throughput, security, and efficiency.

Keywords: Wireguard, L2TP, SSTP, OpenVPN, IKEv2

Na tomto mieste by som rád vyjadril úprimnú vďaku všetkým, ktorí mi pomohli pri vzniku tejto diplomovej práce.

Predovšetkým by som chcel poďakovať svojmu vedúcemu práce, Ing. Miroslavovi Matýskovi, Ph.D., za jeho cenné rady, odborné vedenie, trpezlivosť a motiváciu, ktoré mi poskytoval počas celého obdobia jej spracovania. Jeho podpora a konštruktívne pripomienky boli pre mňa neoceniteľné.

Veľké poďakovanie patrí taktiež mojej rodine a blízkym za ich neutíchajúcu podporu, povzbudenie a pochopenie počas celého štúdia a písania tejto práce. Ich morálna podpora bola pre mňa kľúčová.

OBSAH

ÚVOD.....	11
I TEORETICKÁ ČÁST	12
1 ÚVOD DO VPN	13
1.1 ZÁKLADNÉ POJMY VPN	13
1.1.1 Typy VPN	14
1.1.2 Princíp fungovania VPN	15
1.1.3 Výhody VPN	15
1.1.4 Výzvy a obmedzenia VPN	16
1.1.5 Vývoj a motivácia pre používanie VPN.....	17
1.2 ZOZNÁMENIE SA S OPENVPN	19
1.2.1 História OpenVPN	19
1.2.2 Hlavné vlastnosti OpenVPN	19
1.2.3 Bezpečnosť a autentifikácia v OpenVPN.....	21
1.2.4 Výhody a nevýhody OpenVPN.....	25
1.3 ZOZNÁMENIE SA S IKEV2/IPSEC.....	27
1.3.1 História IKEv2	27
1.3.2 Hlavné vlastnosti IKEv2	27
1.3.3 Fázy protokolu IKEv2	29
1.3.4 Funkcie a spolupráca protokolov ISAKMP, OAKLEY a SKEME	31
1.3.5 Výhody a nevýhody IKEv2.....	32
1.4 ZOZNÁMENIE SA S L2TP	33
1.4.1 História L2TP	33
1.4.2 L2TP Hlavné vlastnosti	33
1.4.3 Architektúra L2TP	34
1.5 ZOZNÁMENIE SA S SSTP	37
1.5.1 História SSTP	37
1.5.2 Hlavné vlastnosti SSTP	37
1.5.3 Výhody a nevýhody SSTP	39
1.6 ZOZNÁMENIE SA S WIREGUARD	39
1.6.1 História Wireguard	40
1.6.2 Hlavné vlastnosti Wireguard	40
1.6.3 Výhody a nevýhody Wireguard	41
2 PARAMETRE OVPLYVŇUJÚCE VÝKON VPN	43
2.1 ŠIFROVACIE ALGORITMY A ICH DOPAD NA PRIEPUSTNOSŤ	43
2.2 VPLYV LATENCIE, STRATY PAKETOV A TCP/UDP REŽIMU.....	43
II PRAKTICKÁ ČÁST.....	45
3 NÁVRH TESTOVACIEHO PROSTREDIA.....	46
3.1 LOGICKÁ TOPOLOGIA TESTOVACEJ SIETE	46
3.2 VIRTUÁLNY SIEŤOVÝ ADAPTÉR	48

3.3	FYZICKÁ ARCHITEKTÚRA	48
3.3.1	Hardvérové a softvérové vybavenie	49
4	IMPLEMENTÁCIA TESTOVACIEHO PROSTREDIA	51
4.1	INŠTALÁCIA VMWARE WORKSTATION PLAYER	51
4.2	VYTVORENIE VIRTUÁLNYCH STROJOV A ICH KONFIGURÁCIA	55
4.2.1	Inštalácia a konfigurácia sieťového pripojenia.....	60
4.2.2	Inštalácia a konfigurácia Ubuntu Server	64
4.2.3	Inštalácia a konfigurácia Ubuntu Klient.....	66
4.2.4	Inštalácia a konfigurácia Windows server.....	66
4.2.5	Inštalácia nástrojov pre testovanie priepustnosti (iPerf3)	69
4.3	KONFIGURÁCIA WIREGUARD	71
4.3.1	Inštalácia a príprava serverovej strany WireGuard	71
4.3.2	Inštalácia a príprava klientskej strany WireGuard	75
4.3.3	Overenie funkčnosti VPN tunela Wireguard.....	77
4.4	KONFIGURÁCIA OPENVPN.....	79
4.4.1	Inštalácia a príprava serverovej strany	79
4.4.2	Inštalácia a príprava klientskej strany	84
4.5	KONFIGURÁCIA IKEV2 (STRONGSWAN)	87
4.5.1	Konfigurácia a príprava serverovej strany	87
4.5.2	Konfigurácia klientskej strany IKEv2	92
4.6	KONFIGURÁCIA L2TP/IPSEC.....	95
4.6.1	Inštalácia a príprava serverovej strany	96
4.6.2	Konfigurácia klientskej strany.....	100
4.7	KONFIGURÁCIA SSTP	103
4.7.1	Inštalácia a príprava serverovej strany	103
4.7.2	Inštalácia a príprava klientskej strany	117
4.8.2	Výsledky testov priepustnosti	120
4.8.3	Analýza, porovnanie a zhodnotenie výsledkov	122
	ZÁVĚR.....	128
	SEZNAM POUŽITÉ LITERATURY	129
	SEZNAM OBRÁZKŮ	132
	SEZNAM TABULEK.....	134
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	135

ÚVOD

Virtuálna súkromná sieť je technológia, ktorá umožňuje vytvárať zabezpečené spojenie medzi dvoma bodmi v rámci nezabezpečeného verejného alebo zdieľaného sieťového prostredia, akým je internet. Virtuálna súkromná sieť vytvára šifrovaný tunel, prostredníctvom ktorého prebieha komunikácia, čím zabezpečuje dôvernosť, integritu a často aj autentifikáciu prenášaných dát. Bežne sa využíva na zabezpečenie vzdialeného prístupu k firemným sieťam, na ochranu súkromia používateľov pri používaní verejných Wi-Fi sietí, ale aj na obchádzanie geografických obmedzení prístupu k online službám.

V súčasnosti bola zaznamenaná rastúca potreba bezpečnej a efektívnej komunikácie prostredníctvom internetu. VPN sa stali neoddeliteľnou súčasťou moderného IT prostredia, kde bola zabezpečená ochrana dát a anonymita používateľov a bol umožnený bezpečný vzdialený prístup k sieťovým zdrojom. Vzhľadom na narastajúce bezpečnostné hrozby a potrebu optimalizácie výkonu sa otázka výberu vhodného VPN protokolu stala čoraz relevantnejšou.

V tejto práci bola riešená konfigurácia a priepustnosť dostupných VPN protokolov. Hlavným cieľom bolo porovnať výkon jednotlivých protokolov z hľadiska ich priepustnosti, latencie a efektivity pri rôznych konfiguráciách. Testovanie bolo realizované v kontrolovanom prostredí, pričom boli zohľadnené faktory, ako šifrovacie algoritmy, režimy prenosu dát a rôzne úrovne sieťovej záťaže.

I TEORETICKÁ ČÁST

1 ÚVOD DO VPN

VPN (Virtual Private Network) predstavovala sieťovú technológiu, ktorá umožnila vytvorenie bezpečného spojenia medzi vzdialenými uzlami prostredníctvom verejnej infraštruktúry, akou je Internet. Tento koncept bol založený na virtualizácii pripojenia, kde bol vytvorený šifrovaný tunel medzi dvoma alebo viacerými bodmi, ktorý zabezpečoval ochranu prenášaných údajov [1].

Podľa Fergusona a Hustona nebola stanovená jednoznačná definícia pojmu VPN, keďže tento termín bol často používaný nekonzistentne v rámci sieťového priemyslu. Boli citované ich slová: „*Termín VPN sa používa takmer rovnako neuvážene ako QoS (Quality of Service) na opis širokej škály problémov a riešení, pričom samotné ciele neboli správne formulované*“ [1, s. 1].

Napriek tomu mohla byť VPN definovaná ako „*komunikačné prostredie, v ktorom bol riadený prístup, aby bolo umožnené pripojenie len v rámci definovanej komunity záujmu, a ktoré bolo vytvorené nejakou formou rozdelenia spoločného základného komunikačného média*“ [1, s. 2].

1.1 Základné pojmy VPN

Pre úvod do problematiky VPN bolo nevyhnutné definovať a objasniť kľúčové termíny, ktoré tvorili ich základ. Táto kapitola sa preto zameriavala na analýzu elementárnych pojmov – sieť, privátna a virtuálna – a ich špecifický význam v kontexte VPN technológií [1].

Pre pochopenie konceptu VPN bolo potrebné rozobrať jednotlivé termíny, ktoré ho tvorili:

- **sieť** – sieť bola definovaná ako „*akýkoľvek počet zariadení, medzi ktorými bola umožnená komunikácia pomocou ľubovoľnej metódy*“ [1, s. 1]. Tieto zariadenia mohli zahŕňať počítače, smerovače, tlačiarne a ďalšie komponenty,
- **privátna** – podľa autorov „*súkromie v kontexte VPN znamenalo, že komunikácia medzi dvoma (alebo viacerými) zariadeniami bola určitým spôsobom utajená – zariadenia, ktoré sa na tejto komunikácii nezúčastňovali, nemali prístup k prenášaným údajom a často ani neboli informované o jej existencii*“ [1, s. 2],
- **virtuálna** – tento pojem označoval simuláciu niečoho, čo fyzicky neexistovalo. V prípade VPN šlo o vytvorenie logického spojenia medzi dvoma bodmi

prostredníctvom spoločnej infraštruktúry, pričom táto komunikácia bola izolovaná od ostatných používateľov tejto infraštruktúry [1].

1.1.1 Typy VPN

S ohľadom na rozmanitosť požiadaviek na zabezpečenie a spôsob pripojenia v rôznych prostrediach boli vyvinuté rôzne typy VPN. Tieto sa líšia svojou implementáciou, použitými protokolmi a úrovňou poskytovaného zabezpečenia. Pochopenie charakteristík jednotlivých typov VPN je kľúčové pre výber najvhodnejšieho riešenia v závislosti od konkrétnych potrieb a infraštruktúry. V nasledujúcej časti sú predstavené a stručne popísané základné kategórie VPN, ktoré sa líšia najmä použitými tunelovacími protokolmi a metódami zabezpečenia [2].

Podľa spôsobu implementácie a použitia je možné VPN rozdeliť do niekoľkých kategórií:

- **PPTP (Point-to-Point Tunneling Protocol)** – starší protokol, ktorý bol vyvinutý spoločnosťou Microsoft. Využíva sa kombinácia TCP (Transmission Control Protocol) a GRE (Generic Routing Encapsulation) pre tunelovanie dát. V súčasnosti je považovaný za zastaraný a nebezpečný z dôvodu zraniteľností v šifrovacom mechanizme MS-CHAPv2 (Microsoft Challenge Handshake Authentication Protocol version 2) [2],
- **IPSec (Internet Protocol Security)** – poskytuje sa silné zabezpečenie na sieťovej vrstve. Podporujú sa dva režimy: tunnel mode (pre VPN) a transport mode (pre end-to-end šifrovanie) [2],
- **SSL/TLS (Secure Sockets Layer and Transport Layer Security) VPN** – využíva sa šifrovací protokol SSL/TLS na ochranu dát. Za výhodu sa považuje jednoduchá integrácia do webových prehliadačov a podpora overovania certifikátov [2].
- **OpenVPN** – moderný a flexibilný VPN protokol, ktorý využíva SSL/TLS a podporuje širokú škálu šifrovacích algoritmov [2],
- **WireGuard** – nová generácia VPN protokolu, ktorý je optimalizovaný na výkon a jednoduchosť konfigurácie. Využívajú sa moderné kryptografické algoritmy ako ChaCha20 a Poly1305 na šifrovanie a autentifikáciu [2].

1.1.2 Princíp fungovania VPN

VPN umožňuje zabezpečenú komunikáciu prostredníctvom verejnej siete s využitím rôznych bezpečnostných mechanizmov. Hlavným cieľom VPN je poskytnúť súkromnú komunikáciu v rámci verejného prostredia. Medzi základné bezpečnostné aspekty patrí šifrovanie prenášaných údajov a autentifikácia pripojených zariadení [1].

Proces bezpečného prenášania dát cez verejnú sieť pomocou VPN zahŕňa šifrovanie a tunelovanie. Tento proces pozostáva z nasledujúcich krokov:

- **šifrovanie dát** – VPN klient šifruje odosielané dáta pred ich odoslaním cez internet [1], [2],
- **autentifikácia** – používateľ alebo zariadenie musí preukázať svoju identitu pomocou certifikátu alebo hesla [1], [2],
- **tunelovanie** – dáta sú zabalené do zabezpečeného tunela, ktorý ich chránil pred neoprávneným prístupom [1], [2],
- **dekódovanie na cieľovom zariadení** – prijímajúce zariadenie dešifruje dáta a sprístupňuje ich aplikáciám [1], [2].

1.1.3 Výhody VPN

VPN prináša množstvo výhod, vďaka ktorým predstavujú kľúčovú súčasť súčasnej digitálnej infraštruktúry. Ich hlavné využitie spočíva v zabezpečení komunikácie, ochrane súkromia a umožnení vzdialeného prístupu k sieťovým zdrojom. Napriek ich prínosom však existujú aj výzvy a obmedzenia, ktoré je potrebné zohľadniť pri nasadzovaní a používaní VPN technológií [1], [2].

Používanie VPN prináša viacero kľúčových výhod, z ktorých najvýznamnejšie boli:

- **zvýšená bezpečnosť** – VPN poskytuje šifrovanie prenášaných údajov, čím sa zabezpečuje ochrana citlivých informácií pred neoprávneným prístupom. Tento aspekt je obzvlášť dôležitý pri používaní verejných Wi-Fi sietí, kde môže dochádzať k ľahkému odpočúvaniu komunikácie útočníkmi. Väčšina moderných VPN riešení využíva AES-256 (Advanced Encryption Standard) alebo ChaCha20, ktoré patria medzi najbezpečnejšie dostupné šifrovacie algoritmy. Použitím VPN sa taktiež znižovalo riziko útokov typu Man-in-the-Middle, pri ktorých útočník zachytával a menil prenášané dáta [1], [2],

- **ochrana súkromia a anonymity** – použitím VPN sa maskuje skutočná IP (Internet Protocol) adresa používateľa, čím sa zabraňuje sledovaniu jeho online aktivít. VPN servery môžu byť umiestnené v rôznych krajinách, čo znamená, že používateľ môže simulovať svoje nachádzanie sa v inej geografickej oblasti. Táto vlastnosť je obzvlášť užitočná v krajinách s prísnou cenzúrou internetu, kde vlády monitorujú a obmedzujú prístup k určitým webovým stránkam [1], [2],
- **obchádzanie geografických obmedzení a cenzúry** – mnohé streamingové služby obmedzujú prístup k obsahu na základe geografickej polohy používateľa. VPN umožňuje pripojenie k serverom v iných krajinách, čím používatelia získavali prístup k blokovanému obsahu [1], [2],
- **vzdialený prístup k firemným zdrojom** – pre firmy predstavuje VPN nevyhnutný nástroj na umožnenie bezpečného prístupu zamestnancov k interným firemným systémom z rôznych lokalít. Tento model umožňuje efektívnu prácu na diaľku a zároveň zachováva vysokú úroveň bezpečnosti [1], [2],
- **zníženie nákladov na infraštruktúru** – VPN umožňuje spoločnostiam znížiť náklady na dedikované privátne siete tým, že sa využíva internet ako transportná vrstva. Toto riešenie je výhodné najmä pre organizácie s viacerými pobočkami, ktoré potrebujú bezpečnú komunikáciu medzi svojimi lokalitami. VPN tak poskytuje cenovo dostupnú alternatívu k tradičným metódam sieťového prepojenia [1], [2],
- **bezpečnosť pri používaní verejných Wi-Fi sietí** – pri pripojení na verejné Wi-Fi siete (napr. na letiskách, v hoteloch či kaviarňach) existuje vysoké riziko odpočúvania dát alebo útokov typu honeypot, kde útočník vytvára falošnú Wi-Fi sieť na zachytávanie citlivých informácií. VPN šifruje všetku komunikáciu, čím sa zabezpečuje ochrana používateľa pred takýmito hrozbami [1], [2].

1.1.4 Výzvy a obmedzenia VPN

Napriek tomu, že VPN ponúka významné výhody, ich implementácia a používanie prináša určité technické a prevádzkové výzvy, ktoré môžu ovplyvniť výkon, bezpečnosť a používateľskú skúsenosť.

Hoci VPN predstavuje efektívne riešenie pre zabezpečenie online komunikácie, ich nasadenie není bez technických a prevádzkových kompromisov. Medzi hlavné obmedzenia a nevýhody patria:

- **zníženie rýchlosti internetového pripojenia** – použitie VPN môže spomaliť rýchlosť pripojenia, keďže dáta sú šifrované a smerované cez vzdialené servery. Tento pokles výkonu závisí od rýchlosti šifrovacieho algoritmu, geografickej vzdialenosti VPN servera a šírky pásma poskytovateľa VPN. Moderné VPN protokoly, ako WireGuard, dokázali minimalizovať tento dopad v porovnaní so staršími riešeniami, ako boli L2TP/IPSec (Layer Two Tunneling Protocol) alebo SSTP (Secure Socket Tunneling Protocol) [3],
- **kompatibilita s rôznymi sieťami a zariadeniami** – nie všetky siete umožňujú používanie VPN. Niektoré verejné Wi-Fi siete blokujú IPSec, OpenVPN alebo WireGuard, aby zabránili šifrovanému prenosu dát. V takýchto prípadoch je možné použiť VPN cez port 443, ktorý je menej pravdepodobne blokovaný. Okrem toho, niektoré staršie zariadenia alebo operačné systémy nemusia podporovať moderné VPN protokoly [4],
- **bezpečnostné riziká pri nesprávnej konfigurácii** – nesprávne nastavená VPN môže paradoxne znížiť bezpečnosť siete. Napríklad pri nesprávnej konfigurácii split-tunnelingu môže dôjsť k úniku DNS (Domain Name System) požiadaviek alebo k smerovaniu nešifrovanej komunikácie mimo VPN tunel. Preto je dôležité používať správne firewall pravidlá a zabezpečiť DNS dotazy [3],
- **závislosť od poskytovateľa VPN** – niektoré komerčné VPN služby môžu uchovávať záznamy o aktivite používateľov alebo dokonca predávať dáta tretím stranám. Preto je dôležité vybrať si dôveryhodného poskytovateľa VPN, ktorý garantoval politiku „no-logs“ a podstupoval pravidelné bezpečnostné audity [3].

1.1.5 Vývoj a motivácia pre používanie VPN

V minulosti boli predstavované verejné dátové siete PDN (Public Data Networks) ako predchodcovia VPN, pričom najznámejším príkladom bol globálny internet. Verejná dátová sieť neobsahuje inherentnú politiku segregácie dátovej prevádzky, a preto je „*zodpovednosťou pripojených entít definovať a vynucovať túto politiku*“ [1, s. 3].

Hlavným dôvodom pre implementáciu VPN sú predovšetkým ekonomické aspekty. Využíva sa existujúca verejná infraštruktúra na prenos dát, čo je podstatne nákladovo efektívnejšie v porovnaní s výstavbou a údržbou vlastnej fyzickej siete. Z ekonomického hľadiska je výhodnejšie prevádzkovať viacero virtuálnych sietí na spoločnej fyzickej

infraštruktúre, než udržiavať rovnaký počet samostatných, fyzicky oddelených komunikačných systémov [1].

Okrem ekonomických výhod zohráva významnú úlohu aj otázka súkromia a bezpečnosti prenášaných informácií. V závislosti od konkrétnych požiadaviek na ochranu dát môže byť poskytovaná VPN s rôznou úrovňou zabezpečenia, od základného skrytia komunikácie až po sofistikované bezpečnostné mechanizmy, ako je šifrovanie a kontrolovaný prístup k sieti [1].

Virtuálne privátne siete neustále prechádzajú vývojom a adaptáciou na nové hrozby v oblasti bezpečnosti. S postupným rozširovaním internetových služieb a narastajúcim počtom kybernetických útokov sa stávajú VPN zásadnou technológiou pre ochranu osobných údajov a súkromia používateľov [1].

1.2 Zoznámenie sa s OpenVPN

OpenVPN je zaradený medzi najrozšírenejšie open-source VPN riešenia, ktoré sú predstavované ako flexibilná, bezpečná a široko podporovaná alternatíva k zavedeným VPN protokolom, akými sú IPsec alebo PPTP. Je vyvinutý s cieľom poskytovať bezpečný, jednoducho konfigurovateľný a efektívny nástroj na šifrovanú komunikáciu prostredníctvom internetu. OpenVPN využíva protokol SSL/TLS na autentifikáciu a zabezpečenie prenosu dát, pričom ponúka rozsiahle možnosti konfigurácie na rôznych platformách [2].

1.2.1 História OpenVPN

OpenVPN je prvýkrát predstavený v roku 2002 Jamesom Yonanom ako riešenie pre zabezpečenie sieťovej komunikácie. Na rozdiel od iných VPN protokolov kladie dôraz na flexibilitu a jednoduchú konfiguráciu. Postupne sa vyvinul na robustné VPN riešenie s podporou rozsiahlej škály šifrovacích algoritmov, autentifikačných mechanizmov a rôznych sieťových konfigurácií [2].

1.2.2 Hlavné vlastnosti OpenVPN

OpenVPN disponuje širokou škálou vlastností, ktoré ho odlišujú od iných VPN riešení. Jeho flexibilita, bezpečnostné mechanizmy a podpora viacerých platforiem z neho robia jeden z najuniverzálnejších VPN riešení, ktoré sú dostupné na trhu [2].

Podpora viacerých sieťových režimov – OpenVPN umožňuje nasadenie v rôznych sieťových režimoch, čo používateľom poskytuje rozsiahlejšiu kontrolu nad smerovaním a distribúciou dát:

- **TUN režim** – routovaná VPN, ktorá prenáša IP pakety medzi klientmi a serverom. Tento režim sa najčastejšie využíva v podnikových prostrediach a pre vzdialený prístup [2],
- **TAP režim** – bridged VPN, ktorá prenáša ethernetové rámce. Tento režim umožňuje prenos broadcast a multicast prevádzky, čo je dôležité pre aplikácie ako NetBIOS a vzdialené herné siete [2].

Podpora viacerých transportných protokolov – Na rozdiel od iných VPN riešení, OpenVPN umožňuje použitie UDP (User Datagram Protocol) aj TCP ako transportného protokolu.

- **UDP režim** – preferovaný pre väčšinu VPN pripojení, keďže ponúka nižšiu latenciu a vyššiu efektivitu [2],
- **TCP režim** – využívaný pri firemnom nasadení a prechádzaní cez striktné firewally, keďže dokáže lepšie zvládať paketové straty [2].

Možnosť tunelovania cez rôzne siete – OpenVPN dokáže tunelovať cez NAT (Network Address Translation) a firewally, čo z neho robí vhodné riešenie pre používateľov v obmedzených sieťových prostrediach.

- **Podpora NAT traversal** – OpenVPN umožňuje komunikáciu medzi zariadeniami, ktoré sú za NAT bránami [2],
- **Podpora pre HTTP (HyperText Transfer Protocol) a SOCKS proxy** – OpenVPN môže fungovať aj cez HTTP alebo SOCKS proxy, čím sa znižuje riziko blokovania VPN prenosov [2],
- **Port 443 kompatibilita** – umožňuje prevádzku OpenVPN na rovnakom porte ako HTTPS (HyperText Transfer Protocol Secure), čím je možné maskovať VPN prevádzku ako bežný internetový prenos [2].

Podpora IPv6 (Internet Protocol version 6) a mobilných platforiem – OpenVPN plne podporuje IPv6, čím zabezpečuje budúcu kompatibilitu so sieťovými infraštruktúrami.

- **IPv6 tunelovanie** – umožňuje vytvorenie VPN spojenia medzi IPv4 (Internet Protocol version 4) a IPv6 sieťami [2].
- **Podpora pre Android a iOS** – OpenVPN ponúka aplikácie pre mobilné zariadenia s plnou kompatibilitou s OpenVPN servermi [2].
- **Open-source a široká komunita**: OpenVPN je open-source projekt, čo znamená, že je neustále vyvíjaný komunitou a pravidelne aktualizovaný.
- **Rýchla odozva na bezpečnostné hrozby** – OpenVPN komunita aktívne pracuje na opravách zraniteľností a implementácii nových bezpečnostných štandardov [2].
- **Kompatibilita s rôznymi operačnými systémami** – OpenVPN je dostupný pre Windows, macOS, Linux, FreeBSD, Android a iOS [2].

1.2.3 Bezpečnosť a autentifikácia v OpenVPN

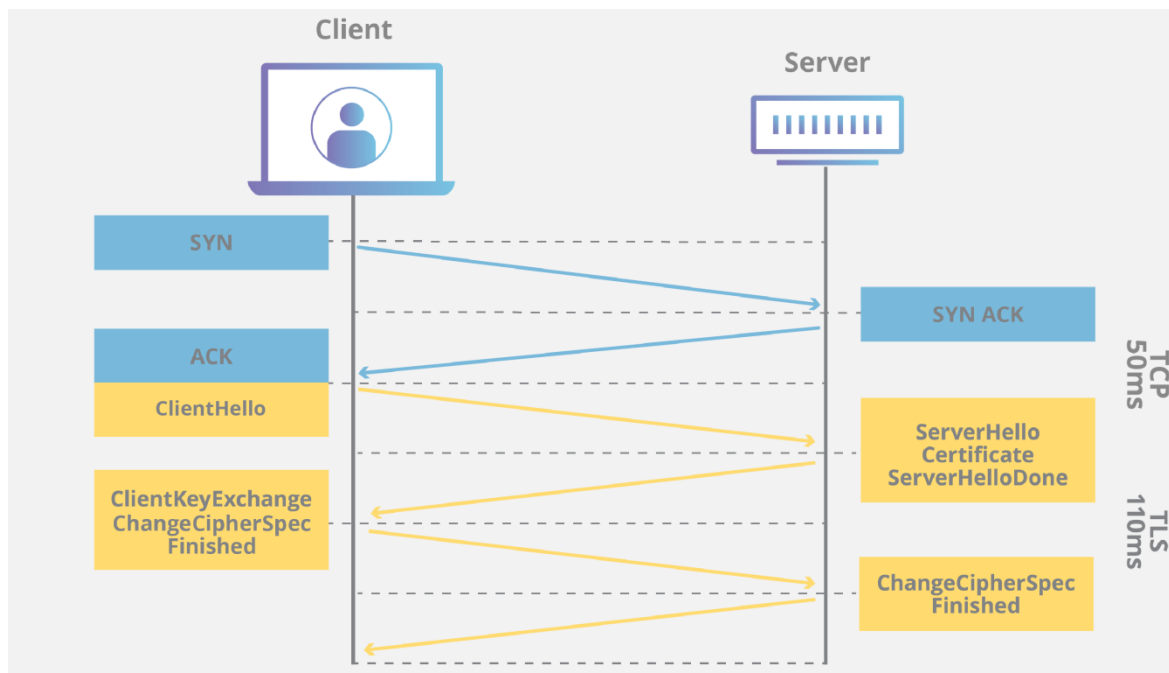
Bezpečnosť a flexibilné možnosti autentifikácie patria k fundamentálnym pilierom protokolu OpenVPN, vďaka čomu sa radí medzi popredné riešenia pre zabezpečenú sieťovú komunikáciu. Na rozdiel od niektorých iných VPN protokolov, OpenVPN stavia na robustnom rámci protokolu SSL/TLS, ktorý poskytuje silný základ pre autentifikáciu komunikujúcich strán a šifrovanie prenášaných dát. Táto kapitola sa podrobne venuje mechanizmom, ktoré OpenVPN využíva na zabezpečenie dôvernosti, integrity a autenticity VPN spojení [2].

Pre svoju činnosť si protokol OpenVPN zvolil osvedčený protokol SSL/TLS, ktorý je široko používaný na zabezpečenie komunikácie na internete, napríklad pri protokole HTTPS. V kontexte OpenVPN slúži SSL/TLS ako základný rámec pre nadviazanie bezpečného spojenia, výmenu kryptografických kľúčov a autentifikáciu medzi VPN klientom a serverom [2].

Komunikácia v OpenVPN je štruktúrovaná prostredníctvom dvoch logických kanálov, ktoré zabezpečujú riadenie spojenia a prenos dát:

- **riadiaci kanál** – zabezpečený protokolom SSL/TLS, tento kanál slúži na úvodnú autentifikáciu, vyjednanie šifrovacích parametrov a výmenu riadiacich informácií. Jeho bezpečnosť je kľúčová pre ochranu procesu nadviazania spojenia a citlivých informácií [2],
- **dátový kanál** – po úspešnej autentifikácii a dohode na šifrovacích parametroch sa tento kanál využíva na prenos samotných dát. Je šifrovaný symetrickou šifrou dohodnutou počas TLS handshake na riadiacom kanáli [2].

Kľúčovým mechanizmom pre zabezpečenie riadiaceho kanála je proces TLS handshake, ktorý zaisťuje bezpečnú výmenu kľúčov a autentifikáciu. Nasledujúci obrázok ilustruje typický priebeh tohto procesu.



Obrázok 1 TLS Handshake

(zdroj: [5])

Využitie protokolu SSL/TLS prináša pre OpenVPN viacero kľúčových výhod v oblasti bezpečnosti, vrátane silnej kryptografie, podpory rozmanitých autentifikačných metód a preukázanej odolnosti voči mnohým typom útokov.

OpenVPN disponuje flexibilnou škálou metód na overenie identity VPN klienta a servera, čo umožňuje prispôbiť úroveň zabezpečenia špecifickým potrebám a danému prostrediu:

- **TLS autentifikácia (tls-auth)** – direktíva `tls-auth` predstavuje voliteľný, avšak vysoko odporúčaný bezpečnostný mechanizmus, ktorý pridáva dodatočnú vrstvu ochrany na úrovni TLS handshake. Využitím statického (vopred zdieľaného) tajného kľúča sa overuje správnosť komunikačného partnera oboma stranami (klientom aj serverom) ešte pred samotnou výmenou certifikátov. Tento mechanizmus účinne bráni útokom typu DoS (Denial of Service), ktoré sa snažia zahltením servera znemožniť nadviazanie platných spojení. Taktiež zabraňuje neautorizovaným klientom, a to aj v prípade, že vlastní platný certifikát, nadviazať spojenie so serverom, ak im nie je známy zdieľaný tajný kľúč. Pre efektívnosť `tls-auth` je kľúčová správna konfigurácia a ochrana zdieľaného tajného kľúča [2],

- **certifikáty X.509** – autentifikácia prostredníctvom digitálnych certifikátov X.509 sa považuje za jednu z najsilnejších a najškálovateľnejších metód v OpenVPN, najmä pre rozsiahlejšie nasadenia. Tento mechanizmus je založený na princípoch asymetrickej kryptografie a infraštruktúry verejných kľúčov PKI (Public Key Infrastructure). Každý klient a server vlastní svoj unikátny digitálny certifikát, ktorý je podpísaný dôveryhodnou CA (Certificate Authority). Počas nadväzovania VPN spojenia si klient a server vymenia svoje certifikáty a overí sa ich platnosť voči certifikátu CA. Tento proces umožňuje bezpečnú vzájomnú autentifikáciu: server overuje identitu klienta na základe jeho certifikátu a voliteľne môže byť identita servera overená klientom, čím sa predchádza útokom typu MitM (Man-in-the-Middle). Využitie certifikátov X.509 prináša výhody v podobe silnej kryptografickej autentifikácie, jednoduchšej správy prístupových práv (odvolanie certifikátov pri kompromitácii) a škálovateľnosti pre rozsiahly počet klientov. Pre správne fungovanie je nevyhnutné vytvoriť a spravovať vlastnú CA alebo použiť dôveryhodnú externú CA [2],
- **Pre-Shared Key režim** – PSK (Pre-Shared Key) predstavuje jednoduchšiu metódu autentifikácie, pri ktorej klient aj server zdieľajú rovnaký tajný kľúč. Tento režim sa vyznačuje pomerne jednoduchou konfiguráciou a je vhodný pre menšie siete alebo testovacie účely. Avšak, z bezpečnostného hľadiska má PSK určité obmedzenia. V prípade kompromitácie zdieľaného kľúča sú ohrozené všetky spojenia, ktoré ho využívajú. Správa kľúčov sa stáva náročnejšou pri väčšom počte klientov a neposkytuje takú úroveň granularity v riadení prístupu ako certifikáty. Z tohto dôvodu sa PSK neodporúča pre produkčné prostredia s vysokými nárokmi na bezpečnosť. Pri využití PSK je mimoriadne dôležité zvoliť silný a komplexný kľúč [2],
- **autentifikácia pomocou používateľského mena a hesla** – OpenVPN umožňuje autentifikáciu klientov aj prostredníctvom tradičného používateľského mena a hesla. Táto metóda sa často integruje s existujúcimi autentifikačnými systémami, akými sú PAM (Pluggable Authentication Modules) na Linuxových systémoch, LDAP (Lightweight Directory Access Protocol), RADIUS (Remote Authentication Dial-In User Service) alebo SAML (Security Assertion Markup Language). Integrácia s týmito systémami

umožňuje centralizovanú správu používateľských účtov a ich prístupových práv k VPN. Pre zvýšenie bezpečnosti sa odporúča kombinovať autentifikáciu pomocou používateľského mena a hesla s ďalšími metódami, ako sú certifikáty alebo MFA (Multi-Factor Authentication). OpenVPN podporuje rôzne implementácie MFA, napríklad prostredníctvom TOTP (Time-based One-time Password) generovaných mobilnými aplikáciami [2].

Po úspešnej autentifikácii sa pre zabezpečenie dôvernosti prenášaných dát využíva šifrovanie. OpenVPN podporuje širokú škálu moderných a silných šifrovacích algoritmov, čo umožňuje prispôbiť úroveň ochrany podľa potrieb:

- **podpora rôznych šifrovacích algoritmov** – OpenVPN umožňuje výber z mnohých symetrických šifrovacích algoritmov, vrátane AES s rôznymi dĺžkami kľúčov (napr. AES-256), ChaCha20-Poly1305 (moderný a efektívny algoritmus s dobrou výkonnosťou aj na menej výkonných zariadeniach), Blowfish a ďalšie. Výber vhodného šifrovacieho algoritmu a dĺžky kľúča má priamy vplyv na úroveň bezpečnosti VPN spojenia. Odporúča sa využívať silné a aktuálne algoritmy, ktoré sa považujú za odolné voči známym útokom [2],
- **automatická rotácia šifrovacích kľúčov** – Pre zvýšenie bezpečnosti OpenVPN pravidelne obmieňa šifrovacie kľúče počas trvania VPN spojenia. Táto prax minimalizuje množstvo dát, ktoré by boli potenciálne ohrozené v prípade prelomenia jedného kľúča. Frekvencia rotácie kľúčov je konfigurovateľná [2],
- **TLS key negotiation** – Proces výmeny šifrovacích kľúčov pre dátový kanál prebieha bezpečne počas úvodného TLS handshake na riadiacom kanáli. Tento mechanizmus zaisťuje, že sa pre šifrovanie dát využívajú silné a náhodne generované symetrické kľúče, na ktorých sa dohodli klient a server [2],
- **PFS (Perfect Forward Secrecy)** – OpenVPN podporuje mechanizmy pre dosiahnutie PFS. PFS zaisťuje, že ak by bol v budúcnosti kompromitovaný dlhodobý súkromný kľúč servera, útočník nebude môcť dešifrovať minulé VPN relácie. To sa dosahuje využívaním krátkodobých Diffie-Hellmanových kľúčov pre každú reláciu [2].

Okrem šifrovania dát OpenVPN využíva aj hašovacie algoritmy napríklad SHA-256 (Secured Hash Algorithm) na zabezpečenie integrity prenášaných dát a overenie ich pôvodu, čím sa predchádza neoprávneným modifikáciám počas prenosu [2].

OpenVPN implementuje aj ďalšie bezpečnostné mechanizmy, ktoré prispievajú k celkovej robustnosti riešenia:

- **tls-crypt a tls-crypt-v2** – tieto príkazy poskytujú dodatočné šifrovanie a autentifikáciu na úrovni TLS handshake, čím maskujú metadáta riadiaceho kanála a zvyšujú jeho odolnosť voči pasívnemu odpočúvaniu a aktívnym útokom [2],
- **obmedzenie privilégií (user, group)** – konfiguračné direktívy user a group umožňujú spúšťať proces OpenVPN pod menej privilegovaným používateľom a skupinou, čím sa minimalizuje potenciálny dopad v prípade zneužitia zraniteľnosti [2],
- **chroot** – príkaz chroot obmedzuje prístup procesu OpenVPN len k určitej časti súborového systému, čo predstavuje ďalšiu vrstvu ochrany proti neoprávnenému prístupu k citlivým systémovým súborom [2],
- **ochrana proti replay útokom** – OpenVPN implementuje mechanizmy na detekciu a prevenciu replay útokov, pri ktorých sa útočník snaží opakovane odosielať zachytené platné pakety [2].

1.2.4 Výhody a nevýhody OpenVPN

OpenVPN patrí medzi najpoužívanjšie VPN riešenia vďaka svojej bezpečnosti, flexibilita a open-source charakteru. Napriek mnohým výhodám však existujú aj určité obmedzenia, ktoré je potrebné zohľadniť pri jeho implementácii.

Výhody OpenVPN:

- **silná bezpečnosť a šifrovanie** – OpenVPN využíva SSL/TLS na autentifikáciu a šifrovanie komunikácie, čo z neho robí bezpečné riešenie proti útokom typu Man-in-the-Middle. Podporuje rôzne šifrovacie algoritmy, vrátane AES-256, ChaCha20-Poly1305 a Blowfish, čím umožňuje vysoko bezpečné spojenia [2], [6],
- **flexibilita konfigurácie** – OpenVPN je možné konfigurovať v režime klient-server aj peer-to-peer, podporuje NAT traversal, tunelovanie cez HTTP/HTTPS proxy a viaceré sieťové konfigurácie (TUN a TAP režim) [2], [6],
- **podpora viacerých transportných protokolov** – OpenVPN môže fungovať cez UDP alebo TCP, čo z neho robí flexibilné riešenie pre rôzne sieťové prostredia [2], [6],

- **kompatibilita s viacerými platformami** – OpenVPN je dostupné pre Windows, macOS, Linux, Android, iOS, FreeBSD a ďalšie operačné systémy, čo umožňuje jeho široké nasadenie [2], [6],
- **open-source a široká komunita** – ako open-source projekt je zdrojový kód OpenVPN neustále auditovaný bezpečnostnými expertmi a pravidelne aktualizovaný komunitou. To znižuje riziko bezpečnostných zraniteľností a zaručuje transparentnosť [2], [6],
- **schopnosť obchádzať sieťové obmedzenia** – vďaka podpore portu 443 (HTTPS) môže OpenVPN maskovať svoju prevádzku ako bežnú HTTPS komunikáciu, čo umožňuje jeho použitie v sieťach s prísnyimi firewallmi [2], [6],
- **automatická rotácia šifrovacích kľúčov** – OpenVPN pravidelne obnovuje šifrovacie kľúče, čím zvyšuje bezpečnosť a minimalizuje riziko kompromitácie kľúčov [2], [6].

Nevýhody OpenVPN:

- **zložitejšia konfigurácia oproti iným VPN riešeniam** – OpenVPN vyžaduje manuálne nastavenie certifikátov, šifrovacích kľúčov a konfigurácie siete, čo môže predstavovať náročnosť pre menej skúsených používateľov [2], [6],
- **vyššia latencia oproti IPSec** – v dôsledku využitia protokolu SSL/TLS a potreby pravidelnej výmeny kľúčov môže OpenVPN vykazovať vyššiu latenciu v porovnaní s IPSec, najmä pri vysokorýchlostných spojeniach [2], [6],
- **vyššie nároky na systémové zdroje** – OpenVPN má tendenciu vykazovať vyššiu spotrebu CPU (Central Processing Unit) v porovnaní s WireGuard alebo IPSec, predovšetkým pri použití silného šifrovania napr. AES-256-GCM (Galois/Counter Mode) [2], [6],
- **závislosť od OpenSSL** – keďže OpenVPN využíva knižnicu OpenSSL, ak sa v OpenSSL objaví bezpečnostná zraniteľnosť, môže to mať dopad aj na OpenVPN [2], [6],
- **neexistencia natívnej podpory v OS** – na rozdiel od IPSec alebo WireGuard nie je OpenVPN štandardne integrovaný v operačných systémoch, čo si vyžaduje inštaláciu dodatočného softvéru [2], [6],

- **obmedzená škálovateľnosť** – OpenVPN nie je primárne optimalizovaný na gigabitové rýchlosti a môže narážať na limity pri obsluhovaní veľkého počtu súčasných pripojení [2], [6].

1.3 Zoznámenie sa s IKEv2/IPSec

Protokol IKEv2 (Internet Key Exchange Version 2) je vybavený širokou škálou vlastností, ktoré sú navrhnuté na zlepšenie bezpečnosti, efektivity a flexibility VPN spojení. Je navrhnutý s cieľom riešiť nedostatky predchádzajúcej verzie IKEv1 (Internet Key Exchange Version 1) a je vybavený spoľahlivými mechanizmami na vyjednávanie bezpečnostných parametrov [7].

1.3.1 História IKEv2

Protokol IKEv2 je navrhnutý ako reakcia na nedostatky staršej verzie IKEv1, ktorá bola využívaná na zabezpečenie komunikácie v rámci IPsec sietí. Nová verzia bola navrhnutá s cieľom zjednodušiť a zrýchliť proces vyjednávania bezpečnostných parametrov, je zvýšená stabilita spojenia a je poskytnutá lepšia podpora pre mobilné zariadenia a siete s prekladom adries [7].

1.3.2 Hlavné vlastnosti IKEv2

Protokol IKEv2 je považovaný za významný krok vpred v oblasti zabezpečenia VPN komunikácie. Je navrhnutý s cieľom prekonať limity svojho predchodcu IKEv1, ktorá bola využívaná na zabezpečenie komunikácie v rámci IPsec sietí.

Medzi kľúčové vlastnosti, ktorými bol IKEv2 charakterizovaný, patrili:

- **efektívne vyjednávanie bezpečnostných parametrov** – v protokole IKEv2 je implementovaná optimalizovaná výmena správ. Vďaka tomuto vylepšeniu je dosiahnuté rýchlejšie nadviazanie spojenia v porovnaní s IKEv1. Počet správ potrebných na vytvorenie SA (Security Association) bol zredukovaný na štyri, čím je znížená latencia a sieťová záťaž a zároveň je zvýšená celková efektivita spojení [7],
- **podpora NAT-T (Network Address Translation–Traversal)** – jedným z kľúčových vylepšení protokolu IKEv2 je zavedená natívna podpora NAT traversal. Vďaka tejto vlastnosti je umožnené fungovanie protokolu aj cez siete s NAT. Zaviedla sa kapsulácia ESP (Encapsulating Security Payload) paketov do UDP na

porte 4500, čím sa vyriešili obchádzané problémy so smerovaním IPsec v prostrediach s NAT [7],

- **zlepšený mechanizmus autentifikácie** – v protokole IKEv2 je podporované široké spektrum autentifikačných metód, čím je dosiahnutá vyššia flexibilita pri nasadení [7]:
 - **certifikáty X.509** – digitálne certifikáty X.509 sú poskytované pre silnú kryptografickú autentifikáciu medzi klientom a serverom,
 - **PSK** – PSK sú predstavované ako jednoduchšia metóda autentifikácie, vhodná pre menšie VPN siete, avšak s nižšou úrovňou bezpečnosti,
 - **EAP (Extensible Authentication Protocol)** – EAP je umožnený na integráciu s existujúcimi autentifikačnými systémami ako LDAP, RADIUS alebo Active Directory, čím je zvýšená kompatibilita s podnikovými sieťami.
- **podpora mobility a multihomingu** – v rámci protokolu IKEv2 je zahrnuté rozšírenie MOBIKE (Mobility and Multihoming Extension). Vďaka tomuto rozšíreniu je umožnená zmena IP adresy klienta bez prerušenia aktívneho VPN spojenia, čo je obzvlášť užitočné pre mobilné zariadenia prechádzajúce medzi rôznymi typmi sietí [7]:
 - **bez prerušenia spojenia** – VPN tunel je udržiavaný aktívny aj pri zmene sieťového pripojenia,
 - **rýchla rekonfigurácia** – je dosiahnutá rýchla rekonfigurácia IPsec tunela, čím sa minimalizuje doba výpadku spojenia pri zmene IP adresy.
- **redukcia sieťovej záťaže** – v protokole IKEv2 je dosiahnuté zníženie množstva sieťovej prevádzky v porovnaní s IKEv1 vďaka efektívnejšiemu protokolu výmeny kľúčov. Automatizovanejší a optimalizovanejší proces znižuje potrebu častého opakovaného vyjednávania spojení [7]:
 - **je znížená** potreba retransmisie správ, čo prispelo k nižšej latencii,
 - **je dosiahnuté** lepšie spracovanie výpadkov spojenia, čím sa minimalizuje potreba opätovného vyjednávania tunela.
- **vylepšená správa SA** – v protokole IKEv2 je zavedený zjednodušený spôsob správy SA. Na rozdiel od IKEv1, kde je potrebné manuálne nastavovať časové limity, sa o

životnosť SA stará každé zariadenie automaticky a v prípade potreby je iniciované jeho obnovenie [7]:

- **je iniciovaný** proces rekeyingu, čím sa predchádza nesúlodom v bezpečnostných politikách,
 - **je umožnený** automatický uzáver neaktívnych tunelov, čím sú šetrené systémové prostriedky.
- **Podpora IPv6 a budúcich štandardov** – protokol IKEv2 je plne kompatibilný s IPv6, čo umožňuje nasadenie aj v moderných sieťových infraštruktúrach. Je zabezpečená plná podpora tunelovania IPv6 cez IPv4 a naopak, čím sú umožnené kryptograficky bezpečné pripojenia aj v budúcich sieťových modeloch [7].

1.3.3 Fázy protokolu IKEv2

Protokol IKEv2 je implementovaný na proces nadviazania bezpečného spojenia v dvoch kľúčových fázach. Prvou z nich je IKEv2 Fáza 1, ktorej hlavným cieľom je vytvorenie bezpečnej a autentifikovanej IKEv2 bezpečnostnej asociácie medzi dvoma sieťovými peerami. Táto fáza je fundamentálnym krokom, ktorý umožňuje následné zabezpečenie datovej komunikácie v rámci IKEv2 Fázy 2 [7].

Primárnym účelom Fázy 1 je dosiahnuť dve zásadné veci:

- **autentifikácia sieťových partnerov** – v Fáze 1 je zabezpečené overenie identity oboch komunikujúcich strán, čím sa eliminuje riziko komunikácie s neautorizovanými entitami [7],
- **dohoda na kryptografických metódach pre Fázu 2** – počas Fázy 1 sa dohodnúť, ktoré šifrovacie algoritmy, hašovacie funkcie a metódy autentifikácie budú použité v nasledujúcej Fáze 2 na zabezpečenie prenášaných dát [7],

Pre dosiahnutie týchto cieľov sú vo Fáze 1 využité špecifické mechanizmy:

- **X.509 certifikáty pre autentifikáciu** – identita komunikujúcich strán je overená pomocou digitálnych X.509 certifikátov. Tieto certifikáty musia byť vopred zdieľané alebo mohli byť bezpečne vymenené prostredníctvom mechanizmov ako DNS alebo DNSSEC (Domain Name System Security Extensions) [7],
- **Diffie-Hellman výmena kľúčov pre šifrovanie** – na vytvorenie zdieľaného tajného kľúča je využitý algoritmus Diffie-Hellman. Táto metóda umožňuje dvom stranám

vytvoriť spoločný tajný kľúč cez nezabezpečený komunikačný kanál bez priamej výmeny [7].

Okrem autentifikácie a dohody na kryptografických metódach je definovaných aj niekoľko globálnych parametrov, ktoré ovplyvnia celkový proces zabezpečenia [7]:

- Názvy certifikátov verejných kľúčov
- PFS
- Dotknuté sieťové rozhrania
- Bezpečnostné protokoly a ich algoritmy
- Metóda autentifikácie

Fáza 1 protokolu IKEv2 je implementovaná dvoma odlišnými spôsobmi, ktoré sú ponúkané ako rôzne kompromisy medzi bezpečnosťou a rýchlosťou:

- **hlavný režim** – je považovaný za predvolenú a odporúčanú konfiguráciu pre Fázu 1. V ňom sú využité kryptografické kľúče založené na infraštruktúre verejných kľúčov na vzájomnú autentifikáciu oboch sieťových partnerov. Dôležité je, že v Hlavnom režime sú chránené prenášané informácie počas procesu vyjednávania, čím je zaistená vyššia úroveň bezpečnosti [7],
- **agresívny režim** – je považovaný za voliteľnú alternatívu, v ktorej sa uprednostňuje rýchlosť nadviazania spojenia pred úrovňou bezpečnosti. V Agresívnom režime je vymenený menší počet správ, čo vedie k rýchlejšiemu dokončeniu Fázy 1. Avšak, kľúčovým nedostatkom je, že v tomto režime sa prenášajú všetky informácie v čistej textovej podobe (plain text) bez akejkoľvek autentifikačnej ochrany [7].

Po úspešnom ukončení Fázy 1 a vytvorení bezpečnej IKEv2 SA sa zaháji IKEv2 Fáza 2. V tejto fáze sa vyjednávajú a implementujú konkrétne parametre pre novú bezpečnostnú asociáciu, ktorá je riadená protokolom ISAKMP (Internet Security Association and Key Management Protocol) a niekedy je nazývaná aj ISAKMP SA. Fáza 2 je považovaná za kľúčovú, pretože práve v nej sa dohodnutú detaily zabezpečenia pre samotnú komunikáciu, pri ktorej je spustený IPsec tunel. Je zdôraznené, že po úspešnom vytvorení bezpečnostnej asociácie Fázy 2 je ukončená IKEv2 SA a spojenie vytvorené v Fáze 1 [7].

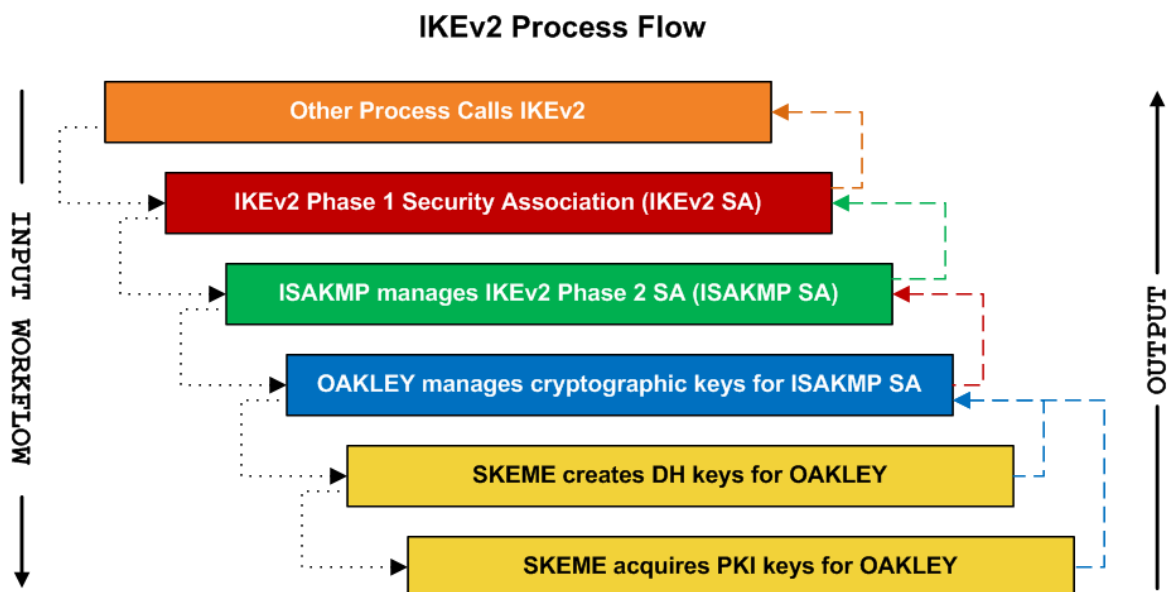
Bezpečnostné politiky a kryptografické kľúče používané v Fáze 2 sú úplne nezávislé od tých, ktoré sú implementované vo Fáze 1. Samotná implementácia bezpečnostnej asociácie Fázy 2 je riadená protokolom ISAKMP, ktorým sa odovzdávajú rôzne

bezpečnostné parametre nižším protokolom, ako boli OAKLEY (Oakley Key Determination Protocol) a SKEME (Secure Key Exchange Mechanism for Internet) [7].

1.3.4 Funkcie a spolupráca protokolov ISAKMP, OAKLEY a SKEME

Protokoly v rámci IKE/IKEv2 sú zložené z troch kľúčových zložiek: ISAKMP, OAKLEY a SKEME, pričom každej z nich je priradená špecifická úloha v procese správy bezpečnosti a distribúcie kľúčov [8].

- **ISAKMP** – tento protokol je zodpovedný za správu bezpečnostných asociácií (SAs). Slúži ako hlavný koordinátor komunikácie medzi dvoma stranami a zabezpečuje vytvorenie bezpečných komunikačných kanálov. Protokolom ISAKMP sa definujú politiky pre zabezpečenie pripojení, riadi autentifikáciu a umožňuje výmenu identifikátorov medzi účastníkmi. V prvej fáze IKEv2 sa prostredníctvom ISAKMP nastavujú bezpečnostné asociácie pre IKEv2 (IKEv2 SA), zatiaľ čo v druhej fáze sa vytvárajú a sú koordinované implementácie bezpečnostných asociácií, ako napríklad IPsec [8].
- **OAKLEY** – tento protokol je zameraný na správu parametrov kľúčov a distribúciu kryptografických kľúčov medzi stranami. Jeho úlohou je pomáhať pri vytváraní a distribúcii kľúčov nevyhnutných na šifrovanie komunikácie, spravovanie kľúčov D-H (Diffie-Hellman), ktoré sa používajú na zdieľanie tajných kľúčov medzi stranami. Protokol OAKLEY pôsobí v oboch fázach IKEv2, pričom v prvej fáze sa zameriava na distribúciu kľúčov a v druhej fáze sa zameriava na správu kľúčov potrebných na šifrovanie [8].
- **SKEME** – tento protokol je určený na generovanie kľúčov a správu ich obnovy počas komunikácie. Prostredníctvom tohto protokolu sa vytvárajú kryptografické kľúče a zabezpečuje ich obnova počas trvania spojenia. Protokol SKEME pôsobí v oboch fázach IKEv2, pričom v prvej fáze sa zameriava na generovanie kľúčov a v druhej fáze na poskytovanie nových kľúčov pre šifrovanie [8].



Obrázok 2 IKEv2 Process flow

(zdroj: [8])

Tieto tri protokoly – ISAKMP, OAKLEY a SKEME – spolupracujú na zabezpečení komunikácie a správnej distribúcií kľúčov, čím je umožnená bezpečná a šifrovaná výmena informácií medzi sieťovými partnermi [8].

1.3.5 Výhody a nevýhody IKEv2

Protokol IKEv2 sa vyznačuje vysokou rýchlosťou, efektivitou a silným zabezpečením, vďaka čomu je považovaný za ideálny pre moderné VPN riešenia. Zároveň však možno zaznamenať obmedzenia v kompatibilitate či konfigurácii [9].

Medzi kľúčové výhody protokolu IKEv2 patrí dosahovaná vysoká rýchlosť a stabilita počas prepínania sietí. Zabezpečené je aj efektívne využívanie zdrojov, čo sa prejavuje minimálnym dopadom na batériu a celkový výkon zariadení. Používateľom poskytuje silné zabezpečenie prostredníctvom implementácie výkonného šifrovania a autentifikačných mechanizmov. V neposlednom rade umožňuje rýchle obnovenie spojenia v prípade neočakávaných výpadkov siete [9].

Na druhej strane, pri používaní protokolu IKEv2 možno zaznamenať určité nevýhody. Jednou z nich sú potenciálne problémy s kompatibilitou, najmä pri systémoch, ktoré nemajú natívnu podporu pre tento protokol. Pre zariadenia bez priamej podpory sú niekedy nevyhnutné dodatočné a komplikované kroky konfigurácie. Fakt, že zdrojový kód protokolu nie je otvorený, niektorí používatelia vnímajú ako znižujúci transparentnosť celého riešenia.

Napokon, závislosť protokolu na špecifických sieťových portoch môže spôsobovať problémy s prístupnosťou v sieťach, kde sú tieto porty obmedzené alebo aktívne blokové [9].

1.4 Zoznámenie sa s L2TP

L2TP (Layer Two Tunneling Protocol) predstavuje tunelovací protokol, ktorý sa používa na vytváranie VPN spojení. Samotný protokol L2TP neposkytuje šifrovanie ani autentifikáciu, a preto sa často kombinuje s protokolom IPsec na zabezpečenie prenášaných dát [10].

1.4.1 História L2TP

Protokol L2TP vzniká koncom 90. rokov 20. storočia ako výsledok spolupráce spoločností Cisco Systems a Microsoft Corporation. Jeho špecifikácia je oficiálne publikovaná v roku 1999 v dokumente RFC 2639. Cieľom vývoja L2TP je spojiť výhody dvoch vtedajších populárnych tunelovacích protokolov: PPTP od Microsoftu, známeho svojou relatívnu jednoduchosťou, a L2F (Layer 2 Forwarding) od spoločnosti Cisco, ktorý ponúka lepšiu podporu pre viacnásobné protokoly a tunelovanie cez rôzne typy sietí. L2TP preberá od oboch protokolov kľúčové koncepty s cieľom vytvoriť jednotný a vylepšený štandard pre tunelovanie na druhej vrstve OSI modelu. Neskôr vychádza aj verzia L2TPv3, ktorá prináša ďalšie vylepšenia a rozšírenia funkcionality [10].

1.4.2 L2TP Hlavné vlastnosti

L2TP predstavuje dôležitý protokol v oblasti VPN technológií. Jeho návrh a implementácia prinášajú špecifické charakteristiky, ktoré ho odlišujú od iných riešení pre zabezpečenie sieťovej komunikácie.

- **Tunelovanie na druhej vrstve OSI modelu** – L2TP pracuje na dátovej spojujovej vrstve (Layer 2), kde enkapsuluje rôzne rámce, ako napríklad ethernetové, pre ich prenos cez IP siete. Vďaka tomu umožňuje prenos rôznorodých protokolov dátovej vrstvy, teoreticky vrátane IPX a AppleTalk, hoci v praxi sa primárne využíva pre IP [10].
- **Podpora viacerých protokolov** – Protokol poskytuje flexibilitu pre rôzne sieťové prostredia vďaka svojej schopnosti tunelovať rôzne protokoly na úrovni dátovej vrstvy. Napriek tejto univerzálnosti sa v súčasnosti takmer výlučne používa na prenos IP prevádzky v rámci VPN [10].

- **Možnosť prechodu cez NAT** – V kombinácii s IPsec L2TP efektívnejšie prekonáva problémy spojené s NAT v porovnaní s protokolom PPTP. Na úspešný prechod cez NAT zariadenia využíva protokoly IPsec ESP a NAT-T [10].
- **Podpora autentifikácie na úrovni tunela a spojenia:**
 - Na úrovni L2TP tunela podporuje autentifikačné protokoly ako PAP (Password Authentication Protocol) a CHAP (Challenge Handshake Authentication Protocol) počas nadväzovania spojenia medzi LAC (L2TP Access Concentrator) a LNS (L2TP Network Server),
 - V rámci samotného tunela sa na overenie používateľa používa PPP (Point-to-Point Protocol) s jeho vlastnými autentifikačnými protokolmi, ako sú PAP, CHAP, MS-CHAP (Microsoft Challenge-Handshake Authentication Protocol) a EAP,
 - Pre dosiahnutie silnejšej autentifikácie sa odporúča spoliehať sa na zabezpečenie poskytované protokolom IPsec [10].
- **Možnosť použitia s rôznymi médiami** – L2TP je primárne navrhnutý pre prenos cez IP siete, pričom pre riadiace správy využíva UDP port 1701. Hoci teoreticky existuje možnosť jeho použitia aj na iných médiách na úrovni dátovej vrstvy, v praxi sa takmer výlučne používa v spojení s IP. Protokol podporuje rôzne VPN topológie, vrátane point-to-point a vzdialeného prístupu [10].

1.4.3 Architektúra L2TP

Architektúra protokolu L2TP je charakteristická svojou dvojvrstvovou štruktúrou spojenia a jasným oddelením rolí medzi dvoma kľúčovými entitami: LAC a LNS. Tieto komponenty spolupracujú na vytváraní a správe VPN tunelov a relácií, cez ktoré sa prenášajú dáta [11].

LAC predstavuje vstupný bod pre klientov, ktorí sa chcú pripojiť k VPN sieti, a nachádza sa buď na strane poskytovateľa prístupu, alebo priamo u používateľa. Jeho primárnou funkciou je prijímať iniciálne spojenia od vzdialených klientov, ktoré sa často realizujú prostredníctvom protokolu PPP cez rôzne prístupové technológie, ako napríklad dial-up alebo DSL (Digital Subscriber Line)[11].

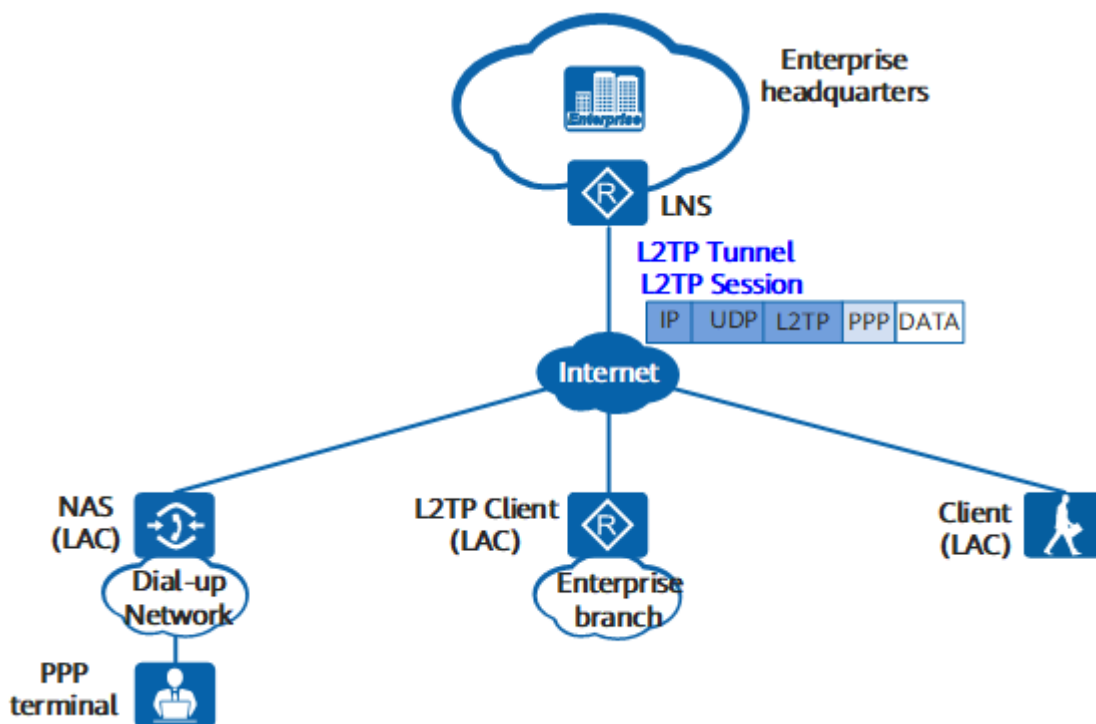
Po úspešnom nadviazaní PPP spojenia s klientom LAC enkapsuluje PPP rámce do L2TP paketov, ktoré obsahujú pôvodné PPP dáta spolu s L2TP hlavičkami umožňujúcimi ich prenos cez IP sieť k LNS [11].

Pre každé PPP spojenie od klienta LAC vytvára L2TP reláciu v rámci existujúceho L2TP tunela k LNS, pričom táto relácia slúži na logické oddelenie dátových tokov rôznych klientov. Následne LAC preposiela enkapsulované L2TP pakety cez IP sieť na LNS, pričom na riadiace správy a dátové prenosy využíva UDP port 1701 [11].

LNS predstavuje koncový bod L2TP tunelov a nachádza sa na strane siete, ku ktorej sa klienti pripájajú, napríklad v podnikovej sieti. Jeho hlavnou úlohou je prijímať L2TP pakety od jedného alebo viacerých LAC cez IP sieť. Po prijatí paketu LNS dekapuluje L2TP hlavičky a extrahuje pôvodné PPP rámce. LNS slúži ako logický koncový bod PPP spojenia, spracováva autentifikačné protokoly ako CHAP a MS-CHAP a konfiguruje sieťové protokoly, napríklad prideluje klientom IP adresy. Po úspešnej autentifikácii a konfigurácii LNS umožňuje klientovi prístup k zdrojom v cieľovej sieti [11].

L2TP tunel predstavuje virtuálne spojenie typu point-to-point, ktoré sa vytvára medzi LAC a LNS cez IP sieť. Tento tunel slúži ako logický kanál, cez ktorý sa prenášajú L2TP pakety. Vytváranie a správa tunela prebieha pomocou riadiacich správ L2TP, ktoré si LAC a LNS vymieňajú cez UDP port 1701. Tieto správy sú kľúčové pre nadviazanie tunela, udržiavanie jeho stavu počas trvania spojenia a jeho následné ukončenie. Dôležitou vlastnosťou L2TP tunela je možnosť multiplexovania relácií, čo znamená, že v rámci jedného tunela môže existovať viacero L2TP relácií, pričom každá relácia zodpovedá jednému PPP spojeniu od klienta. Táto funkcia umožňuje efektívne využitie tunela pre pripojenie viacerých používateľov súčasne [11].

L2TP relácia predstavuje logické spojenie, ktoré existuje v rámci vytvoreného L2TP tunela. Každá relácia priamo mapuje jedno PPP spojenie, ktoré prebieha medzi vzdialeným klientom (na strane LAC) a LNS. Jednou z kľúčových vlastností relácie je jej identifikácia prostredníctvom unikátneho ID (Identity) tunela a ID relácie. Tieto identifikátory umožňujú LAC a LNS správne smerovať prenášané dátové pakety k príslušnému PPP spojeniu. Okrem toho sa v rámci danej relácie tunela prenášajú pôvodné PPP dátové pakety, ktoré sú enkapsulované do L2TP dátových správ [11].



Obrázok 3 Ukážka L2TP

(zdroj: [12])

1.4.4 Výhody a nevýhody L2TP

Protokol L2TP, podobne ako iné protokoly určené na tunelovanie, ponúka špecifické výhody a nevýhody, ktorých zváženie je dôležité pri návrhu a implementácii VPN riešení. Tento protokol je vyvinutý s cieľom poskytnúť flexibilný a efektívny spôsob prenášania dát cez rôzne typy sietí [13].

Jednou z významných výhod je jeho široká podpora na rôznych zariadeniach a operačných systémoch. Predstavuje všestrannú voľbu pre rôzne sieťové prostredia a dobre sa integruje s rozličnými systémami. Je užitočný pre multi-protokolové VPN riešenia, keďže umožňuje zapuzdrenie rôznych sieťových protokolov. V kombinácii s protokolom IPsec poskytuje bezpečnú metódu tunelovania so silným šifrovaním. Pri prenose dát zaisťuje súkromie a integritu pomocou štandardov šifrovania, ako napríklad AES [13].

Medzi nevýhody patrí potreba ďalšieho protokolu, ako je IPsec, na zabezpečenie šifrovania, čo zvyšuje celkovú zložitosť konfigurácie. V prípade nesprávnej konfigurácie IPsec môžu existovať potenciálne zraniteľnosti. Dvojité enkapsulácia dát môže viesť k zhoršeniu výkonu siete. Protokol môže mať problémy s prechodom cez NAT a často

vyžaduje dodatočnú konfiguráciu pre úspešný prechod cez firewally. Na zabezpečenie jeho efektívnej prevádzky je nevyhnutná starostlivá konfigurácia a správa [13].

1.5 Zoznámenie sa s SSTP

SSTP (Secure Socket Tunneling Protocol) je VPN protokol vyvinutý spoločnosťou Microsoft, ktorý zabezpečuje šifrovaný prenos dát cez internet pomocou SSL/TLS. Na rozdiel od L2TP alebo PPTP využíva štandardný HTTPS port 443, čo mu umožňuje obchádzať firewally a proxy servery, ktoré blokujú tradičné VPN porty. SSTP je primárne integrovaný do operačného systému Windows, no je podporovaný aj v iných systémoch (Linux, macOS) prostredníctvom softvéru tretích strán. Hlavnou výhodou SSTP je jeho vysoká bezpečnosť a kompatibilita s modernými sieťovými prostrediami [14].

1.5.1 História SSTP

SSTP je predstavený v roku 2008 ako súčasť Windows Vista Service Pack 1 a následne sa rozširuje do systémov Windows 7, 8, 10 a serverových edícií. Cieľom jeho vývoja je poskytnúť alternatívu k menej bezpečným protokolom, ako je PPTP, ktorý má známe zraniteľnosti, a zároveň riešiť problémy s NAT traversal a blokovaním portov, ktoré sprevádzajú kombináciu L2TP/IPsec. SSTP využíva štandardy SSL/TLS, ktoré sú dlhodobo overené v oblasti webovej bezpečnosti, čo z neho robí dôveryhodné riešenie pre podnikové prostredia [14].

1.5.2 Hlavné vlastnosti SSTP

SSTP sa vyznačuje niekoľkými kľúčovými vlastnosťami, ktoré ho odlišujú od iných VPN riešení. Kombinácia silného šifrovania, schopnosti obchádzať firewally a natívnej integrácie so systémom Windows z neho robí flexibilný nástroj pre zabezpečenú komunikáciu. Medzi hlavné funkcie a charakteristiky SSTP patria:

- **šifrovanie pomocou SSL/TLS** – SSTP využíva moderné šifrovacie štandardy TLS 1.2/1.3 s podporou pre AES-256 a SHA-384, čím zaisťuje dôvernosť a integritu prenášaných dát. Bezpečná výmena kľúčov a autentifikácia servera (a voliteľne aj klienta) je zabezpečená prostredníctvom SSL/TLS handshake a certifikátov X.509 [15],
- **port 443 a obchádzanie firewallov** – prenos dát prebieha cez TCP port 443, ktorý je štandardne otvorený pre webovú prevádzku. Táto vlastnosť umožňuje protokolu

SSTP obchádzať blokovanie tradičných VPN portov, ako je napríklad UDP 500 používaný protokolom IPsec, a efektívne fungovať aj v sieťach s prísnyimi firewallovými pravidlami, ako sú verejné Wi-Fi siete alebo podnikové prostredia [14],

- **integrácia s PPP** – pred samotným šifrovaním SSTP zapuzdruje prenášané dáta do PPP rámcov (Point-to-Point Protocol). Táto integrácia umožňuje prenos rôznych sieťových protokolov, vrátane IPv4, IPv6 a NetBIOS, cez vytvorený VPN tunel [15],
- **natívna podpora v systéme Windows** – SSTP je natívne integrovaný do operačného systému Windows už od verzie Vista SP1. Táto skutočnosť výrazne zjednodušuje jeho nasadenie v prostrediach, kde dominujú systémy od spoločnosti Microsoft, keďže klient na pripojenie nevyžaduje inštaláciu žiadneho dodatočného softvéru [15],
- **autentifikácia používateľov** – protokol SSTP podporuje viacero metód autentifikácie používateľov. Medzi ne patria MS-CHAP v2 (Microsoft Challenge-Handshake Authentication Protocol) a EAP (Extensible Authentication Protocol), ktorý umožňuje pokročilé autentifikačné scenáre, napríklad s využitím smart kariet. Taktiež je podporovaná integrácia s Active Directory pre centralizovanú správu používateľských účtov [15],
- **podpora NAT** – SSTP automaticky rieši preklad sieťových adries (NAT), čo predstavuje významnú výhodu pri pripojeniach z domácich alebo podnikových sietí, kde viaceré zariadenia zdieľajú jednu verejnú IP adresu [15],
- **TCP-based protokol** – na rozdiel od protokolov založených na UDP, ako je napríklad WireGuard, SSTP ako protokol založený na TCP garantuje doručenie všetkých prenášaných paketov. Táto spoľahlivosť však môže viesť k zníženiu výkonu (tzv. výkonnostnému overheadu) v dôsledku opätovného odosielania stratených paketov (TCP meltdown) [15],
- **proprietárny, ale čiastočne otvorený** – hoci je SSTP primárne vyvinutý spoločnosťou Microsoft, jeho implementácia je čiastočne otvorená, čo umožňuje kompatibilitu aj s inými platformami, ako sú Linux a macOS, prostredníctvom softvéru tretích strán [14].

1.5.3 Výhody a nevýhody SSTP

Podobne ako iné tunelovacie protokoly, aj SSTP má svoje charakteristické silné a slabé stránky, ktoré je potrebné brať do úvahy pri plánovaní a zavádzaní VPN riešení. Jeho vývojovým cieľom je zabezpečiť spoľahlivý a dôveryhodný prenos dát v rámci počítačových sietí [6].

Jednou z hlavných výhod je jeho hladká integrácia so systémami Windows. SSTP taktiež umožňuje obchádzanie firewallov a proxy serverov prostredníctvom SSL na porte 443. Vďaka tomu poskytuje spoľahlivý prístup aj v reštriktívnych sieťových prostrediach. Pre administrátorov systémov Windows sa vyznačuje relatívne jednoduchým nastavením [6].

Medzi nevýhody patrí jeho proprietárna povaha, keďže ide o protokol spoločnosti Microsoft, čo môže byť vnímané ako menej transparentné riešenie. Podpora mimo prostredí Windows je obmedzená. Intenzívne šifrovanie môže v niektorých prípadoch znižovať prenosovú rýchlosť. Napokon, pre najvyššie bezpečnostné požiadavky alebo v heterogénnych operačných prostrediach nemusí SSTP predstavovať ideálne riešenie [6].

1.6 Zoznámenie sa s Wireguard

WireGuard je moderný a bezpečný sieťový tunelovací protokol tretej vrstvy OSI modelu. Jeho pôvodná implementácia v jadre systému Linux mu zabezpečuje vysoký výkon a úroveň bezpečnosti. Neskôr bol portovaný aj na iné operačné systémy vrátane Windows, kde existuje implementácia v používateľskom priestore, ako aj novšia, natívna jadrová verzia WireGuardNT. Ambíciou WireGuardu je nahradiť etablované riešenia ako IPsec a OpenVPN, a to vďaka lepšej bezpečnosti, vyššej výkonnosti a jednoduchšiemu používaniu naprieč rôznymi platformami.

Charakterizuje ho minimalistický dizajn a relatívne malý objem zdrojového kódu, čo zjednodušuje bezpečnostný audit a overovanie spoľahlivosti. Základnou koncepciou protokolu je mapovanie verejného kľúča partnera na povolené zdrojové IP adresy. Na vytvorenie bezpečného spojenia využíva jednokolovú výmenu kľúčov (1-RTT) postavenú na moderných kryptografických algoritmoch. WireGuard tiež implementuje pokročilé metódy na ochranu proti útokom typu DoS [16].

1.6.1 História Wireguard

Vývoj protokolu WireGuard bol iniciovaný snahou prekonať zložitosť a obmedzenia existujúcich riešení pre zabezpečené tunelovanie, najmä protokolov IPsec a OpenVPN. Hoci IPsec bol z akademického hľadiska správne vrstvený, vyznačuje sa značnou komplexnosťou, rozsiahlym objemom kódu a náročnou konfiguráciou a implementáciou. OpenVPN, fungujúci v používateľskom priestore a založený na TLS, zase trpí nižšou výkonnosťou a zložitosťou spôsobenou širokou škálou podporovaných funkcií TLS [17].

Primárnou motiváciou pre vznik WireGuardu bola potreba vytvoriť VPN protokol, ktorý by bol bezpečnejší, výkonnejší a zároveň jednoduchší na použitie a správu. Vývojári sa zamerali na elimináciu zbytočnej komplexnosti, zjednotenie vrstiev a využitie moderných, kryptograficky silných a overených algoritmov. Inšpiráciu čerpali najmä zo spôsobu správy kľúčov v OpenSSH. Ich cieľom bolo navrhnúť protokol s malou kódovou základňou, čo zjednodušuje bezpečnostné audity a znižuje riziko zraniteľností.

WireGuard tak vznikol ako protokol s jasnou filozofiou: uprednostniť jednoduchosť, priehľadnosť a bezpečnosť pred komplexnou flexibilitou a univerzalitou starších riešení [17].

1.6.2 Hlavné vlastnosti Wireguard

WireGuard sa vyznačuje niekoľkými zásadnými charakteristikami, ktoré ho odlišujú od tradičných VPN riešení. Jeho dôraz na bezpečnosť, výkon a jednoduchosť sa prejavuje v architektonických rozhodnutiach a použitých kryptografických primitívoch. Medzi hlavné vlastnosti WireGuardu patria:

- **moderná a silná kryptografia** – WireGuard využíva súčasné, bezpečné a overené kryptografické algoritmy pre všetky aspekty zabezpečenia komunikácie. Patrí sem:
 - **Curve25519** pre bezpečnú výmenu kľúčov (eliptické krivky),
 - **ChaCha20-Poly1305** pre autentifikované šifrovanie dát, kombinujúce vysoký výkon (ChaCha20) s integritou (Poly1305),
 - **HKDF (HMAC-based Extract-and-Expand Key Derivation Function)** na bezpečné odvodzovanie kľúčov,
 - **BLAKE2s** ako rýchla a bezpečná hashovacia funkcia [16].
- **jednoduchá a rýchla výmena kľúčov (1-RTT)** – WireGuard implementuje jednokolový handshake (1 Round Trip Time), založený na štruktúre Noise IK

pattern, čo minimalizuje latenciu pri nadväzovaní spojenia. Na autentifikáciu strán sa používajú statické, pre-shared verejné kľúče, podobne ako v OpenSSH [16],

- **vysoký výkon a efektívna implementácia** – Pôvodná jadrová implementácia pre Linux maximalizuje výkon a znižuje režijné náklady. Pre Windows existuje natívna verzia WireGuardNT. Protokol využíva UDP, čo oproti TCP (napr. SSTP) prináša nižšiu latenciu a vyššiu priepustnosť, WireGuard má mimoriadne malý objem kódu menej ako 4000 riadkov v Linuxe – čo uľahčuje audit a znižuje riziko zraniteľností [16],
- **bezproblémový roaming** – WireGuard automaticky zisťuje a aktualizuje IP adresy peerov na základe autentifikovaných paketov. Vďaka tomu možno plynulo prepínať medzi sieťami (napr. Wi-Fi a mobilnými dátami) bez prerušenia spojenia [16],
- **silné bezpečnostné vlastnosti** – protokol integruje pokročilé bezpečnostné mechanizmy. PFS zaisťuje, že prípadná kompromitácia dlhodobých kľúčov neohrozí históriu prenesenej komunikácie. Implementovaná je aj ochrana proti replay útokom a ochrana proti Key-Compromise Impersonation. WireGuard taktiež zahŕňa mechanizmy na skrytie identity statických verejných kľúčov a vstavanú ochranu proti útokom typu Denial of Service (DoS) prostredníctvom inovatívneho využitia kryptografických cookies [16],
- **jednoduchá správa a konfigurácia** – WireGuard sa konfiguruje cez jednoduché virtuálne rozhranie (napr. `wg0`) pomocou štandardných nástrojov ako `ip` alebo `ifconfig`. Základ konfigurácie tvorí priradenie verejného kľúča k povoleným IP adresám peerov [16],
- **flexibilná podpora sietí** – WireGuard natívne podporuje IPv4 aj IPv6, vrátane možnosti enkapsulovať jeden protokol do druhého, čo zvyšuje flexibilitu pri rôznych sieťových scenároch [16],
- **nezávislý prístup ku distribúcii kľúčov** – Protokol sám nešpecifikuje metódu výmeny verejných kľúčov. Ich distribúciu ponecháva na iné mechanizmy napríklad manuálnu výmenu, zabezpečené prenosové kanály alebo PKI [16].

1.6.3 Výhody a nevýhody Wireguard

Tento moderný protokol poskytuje bezpečný, vysoko výkonný a jednoducho použiteľný spôsob prenášania dát cez počítačové siete, pričom jeho cieľom je nahradiť etablované

riešenia ako IPsec a OpenVPN. Medzi hlavné výhody protokolu WireGuard patrí jeho jednoduchý dizajn a malá kódová báza, ktorá uľahčuje audit a údržbu. Zabezpečenie na vysokej úrovni dosahuje vďaka použitiu modernej kryptografie. Rýchle nadviazanie spojenia umožňuje jednokolový handshake (1-RTT). Protokol sa vyznačuje efektívnym výkonom s nízkou latenciou a zároveň zabezpečuje vysokú priepustnosť prenášaných dát.

Na druhej strane, k nevýhodám patrí jeho relatívna novosť, čo znamená menej rozsiahle testovanie v reálnych prevádzkových podmienkach v porovnaní so staršími protokolmi. Predvolené nastavenia zahŕňajú ukladanie IP adries používateľov na serveri, čo si pre lepšie súkromie vyžaduje dodatočnú konfiguráciu. V niektorých prípadoch sa vyskytujú problémy s kompatibilitou s existujúcou sieťovou infraštruktúrou a podpora tohto protokolu je obmedzená u niektorých poskytovateľov VPN služieb. V určitých scenároch je potrebné prostredie s podporou viacerých protokolov [16], [17].

2 PARAMETRE OVPLYVŇUJÚCE VÝKON VPN

Výkon virtuálních privátných sítí představuje komplexní charakteristiku, kterou determinují viaceré faktory. Tieto faktory priamo ovplyvňujú rýchlosť prenosu dát (priepustnosť), odozvu siete (latenciu) a spoľahlivosť spojenia (stratu paketov) [18], [19].

2.1 Šifrovacie algoritmy a ich dopad na priepustnosť

Používané šifrovacie algoritmy zohrávajú zásadnú úlohu pri zabezpečovaní dôvernosti dát, ktoré sa prenášajú prostredníctvom VPN tunela. Avšak výpočtová náročnosť šifrovacích operácií môže signifikantne ovplyvňovať priepustnosť siete v dôsledku zaťaženia procesora.

- Silnejšie šifrovacie algoritmy, ako napríklad AES s 256-bitovým kľúčom, poskytujú vyššiu úroveň bezpečnosti, avšak vyžadujú rozsiahlejšie výpočtové zdroje na procesy šifrovania a dešifrovania dát. Tento fakt vedie k redukcii maximálnej dosiahnuteľnej prenosovej rýchlosti, najmä na zariadeniach s limitovaným výkonom, ako sú mobilné telefóny alebo staršie smerovače.
- Slabšie alebo menej výpočtovo náročné šifrovacie algoritmy, napríklad AES so 128-bitovým kľúčom alebo iné, menej bezpečné algoritmy, umožňujú dosiahnutie vyššej priepustnosti. Tento nárast výkonu je však dosiahnutý na úkor potenciálne nižšej úrovne zabezpečenia prenášaných dát.

Výber optimálneho šifrovacieho algoritmu preto predstavuje kompromis medzi požadovanou úrovňou bezpečnosti a akceptovateľným vplyvom na výkon siete. Súčasný VPN protokoly často implementujú možnosť výberu z rôznych šifrovacích algoritmov a ich konfiguračných nastavení [18].

2.2 Vplyv latencie, straty paketov a TCP/UDP režimu

Charakteristiky sieťového spojenia, ako sú latencia, strata paketov a použitý transportný protokol (TCP alebo UDP), majú signifikantný vplyv na celkový výkon VPN.

- Latencia predstavuje čas potrebný na prenos paketu z jedného sieťového bodu do druhého. Zvýšená latencia priamo ovplyvňuje interaktivitu aplikácií a môže znižovať vnímanú rýchlosť pripojenia. VPN spojenie pridáva určitú mieru latencie v dôsledku procesov enkapsulácie a spracovania paketov. Fyzická vzdialenosť medzi VPN klientom a serverom je taktiež významným faktorom, ktorý ovplyvňuje latenciu [19].

- Strata paketov ide o percentuálny podiel paketov, ktoré sa počas prenosu stratia a následne musia byť opätovne odoslané (v prípade TCP) alebo jednoducho chýbajú (v prípade UDP). Vysoká miera straty výrazne znižuje efektívnu priepustnosť a môže spôsobovať nestabilitu spojenia. VPN tunel je náchylnejší na stratu paketov v sieťach s nižšou spoľahlivosťou [19].
- TCP je spoľahlivý protokol orientovaný na spojenie, ktorý garantuje doručenie paketov v správnom poradí a rieši stratu paketov prostredníctvom mechanizmu opakovaného prenosu. Táto spoľahlivosť však vedie k vyššej réžii a potenciálne nižšej priepustnosti v porovnaní s UDP, najmä pri vysokých úrovniach straty paketov. Väčšina VPN protokolov má možnosť prevádzky cez TCP [19].
- UDP je bezstavový protokol, ktorý neposkytuje spoľahlivé doručenie ani zachovanie poradia paketov. Vyznačuje sa nižšou réžiou ako TCP a je vhodný pre aplikácie citlivé na latenciu, ako sú VoIP, online hry a streamovanie multimédií [19].

II PRAKTICKÁ ČÁST

3 NÁVRH TESTOVACIEHO PROSTREDIA

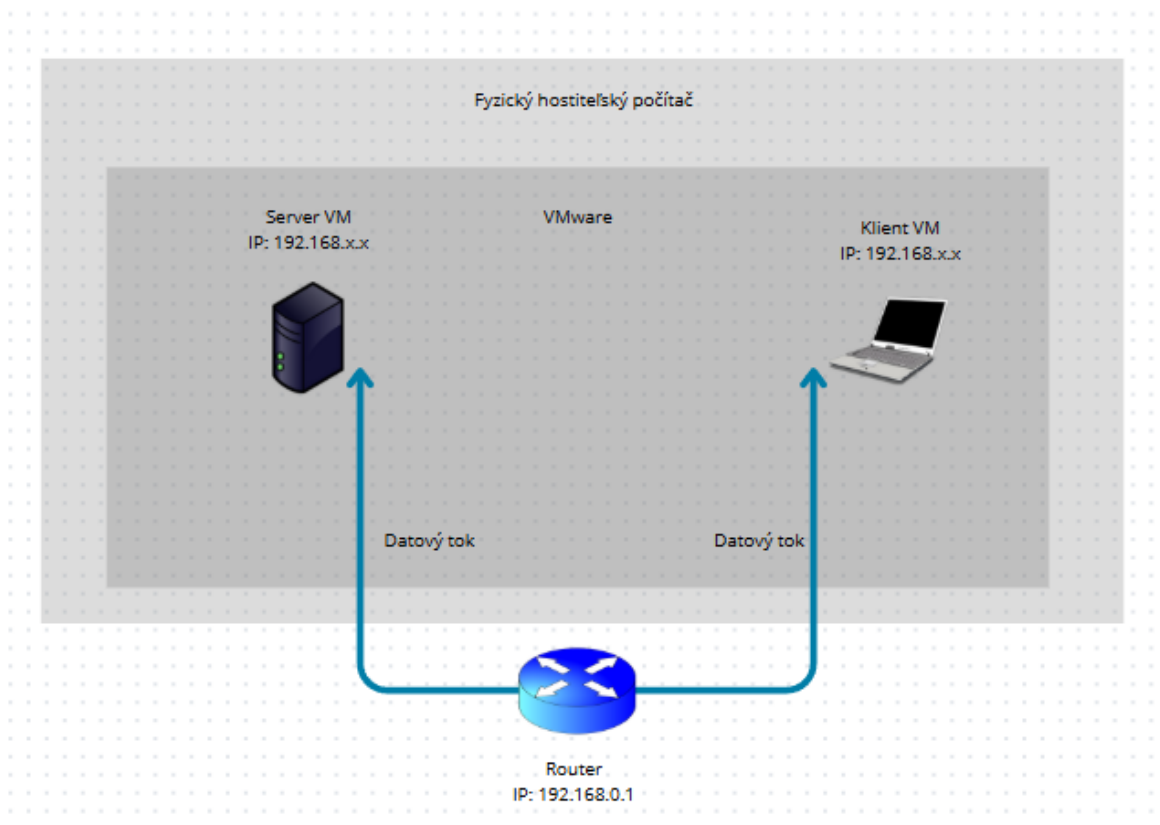
Testovacie prostredie pre meranie a analýzu výkonu VPN protokolov bolo navrhnuté a implementované ako virtuálna sieťová infraštruktúra s využitím hypervízora VMware Workstation Player. Toto riešenie umožnilo vytvorenie izolovaného a kontrolovaného prostredia, ktoré eliminovalo vplyvy externej siete a zaistilo konzistentné podmienky pre opakované testy. Všetky hlavné komponenty siete – konkrétne Ubuntu klient, Ubuntu server a Windows server – boli nastavené ako virtuálne stroje, ktoré bežali na jedinom fyzickom hostiteľskom počítači. Komunikácia medzi týmito virtuálnymi strojmi a smerovanie do širšej siete (vrátane internetu) prebiehalo prostredníctvom fyzického routera. Toto usporiadanie umožnilo efektívnu správu virtuálneho prostredia pri zachovaní realistického sieťového toku cez externé sieťové zariadenie.

3.1 Logická topológia testovacej siete

Na Obrázku 4 bola znázornená základná topológia testovacieho prostredia pred akoukoľvek konfiguráciou VPN protokolov alebo nadviazaním VPN tunela. Celá architektúra spočívala na fyzickom hostiteľskom počítači, ktorý hostil všetky virtuálne komponenty. Kľúčové komponenty tejto základnej topológie boli:

- **router** – zabezpečoval smerovanie prevádzky medzi virtuálnymi strojmi navzájom, ako aj ich pripojenie do externej siete a internetu. IP adresy pre virtuálne stroje boli v rámci príslušnej podsiete pridelené dynamicky prostredníctvom DHCP (Dynamic Host Configuration Protocol) servera integrovaného vo fyzickom routeri,
- **server VM (Virtual Machine)** – virtuálny stroj, ktorý mal primárne slúžiť ako platforma pre konfiguráciu VPN protokolov. V tejto základnej topológii bol k VMnet1 pripojený prostredníctvom virtuálneho sieťového adaptéra a získal IP adresu z rozsahu 192.168.x.x,
- **klient VM** – virtuálny stroj, z ktorého boli vykonávané testy priepustnosti. Bol taktiež pripojený k VMnet1 prostredníctvom virtuálneho sieťového adaptéra a získal IP adresu z rovnakého rozsahu 192.168.x.x.

V tejto počiatočnej konfigurácii prebiehala komunikácia medzi Server VM a Klient VM priamo cez router, bez akéhokoľvek šifrovania alebo tunelovania, simulujúc bežnú lokálnu sieťovú komunikáciu.



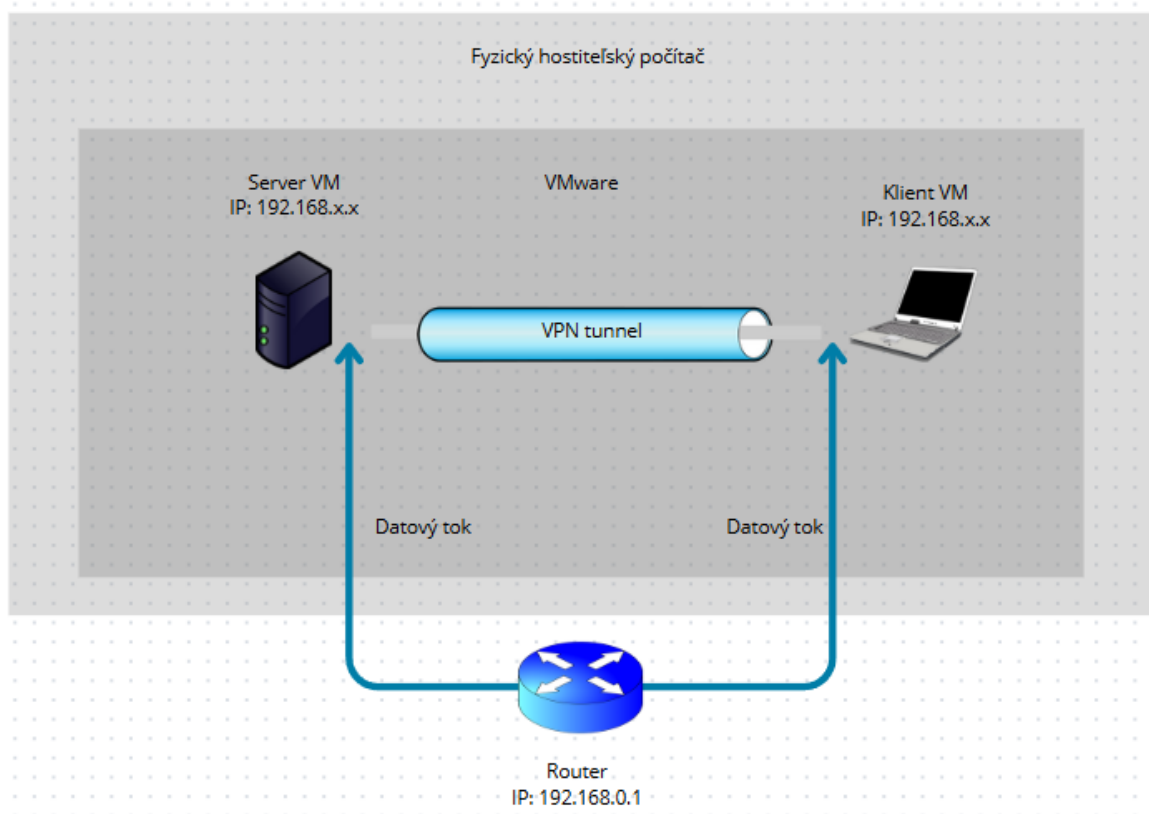
Obrázok 4 Topológia testovacej siete

(zdroj: vlastný)

Na (Obr. 5) bola predstavená topológia testovacieho prostredia s aktívnym VPN tunelom, ktorá bola priamo využívaná pre meranie výkonu jednotlivých VPN protokolov. Táto konfigurácia vychádzala zo základnej topológie z (Obr. 4), avšak s pridanou logickou vrstvou zabezpečenej komunikácie.

Po úspešnom nadviazaní VPN tunela si virtuálne stroje prideliťovali interné IP adresy, ktoré boli platné len v rámci tohto tunela (napríklad z podsiete 10.0.0.0/24). Tieto adresy sa líšili od IP adries pridelené routerom.

Na meranie priepustnosti bol použitý nástroj iPerf3. Dátový tok generovaný iPerf3 klientom na Klient VM bol šifrovaný a tunelovaný prostredníctvom aktívneho VPN spojenia a prenášaný k iPerf3 serveru bežiacemu na Server VM. Takto bolo zabezpečené, že všetky merania odrážali reálny výkon daného VPN protokolu vrátane réžie spôsobenej šifrovaním a tunelovaním. Týmto prístupom bolo možné presne porovnať priepustnosť rôznych VPN protokolov v kontrolovanom a izolovanom prostredí.



Obrázok 5 Topológia siete s aktívnym tunelom

(zdroj: vlastný)

3.2 Virtuálny sieťový adaptér

Sieťová komunikácia pre všetky virtuálne stroje, vrátane Ubuntu VPN klienta, Ubuntu servera a Windows VPN servera, bola spravovaná prostredníctvom fyzického routera. Virtuálne stroje boli nakonfigurované s typom sieťového pripojenia Bridged Mode v hypervízore VMware Workstation Player. Toto nastavenie zabezpečilo, že sa virtuálne stroje pripájali priamo k fyzickej sieti, čo im umožnilo komunikovať s ostatnými zariadeniami v lokálnej sieti a pristupovať na internet rovnako, ako keby to boli samostatné fyzické počítače. IP adresy pre virtuálne stroje boli v rámci príslušnej podsiete pridelované dynamicky prostredníctvom DHCP servera integrovaného vo fyzickom routeri.

3.3 Fyzická architektúra

Všetky tieto virtuálne komponenty bežali na jedinom fyzickom hostiteľskom počítači s dostatočnými hardvérovými prostriedkami na zabezpečenie stabilného a spoľahlivého behu viacerých virtuálnych strojov súčasne, bez vzájomného ovplyvňovania ich výkonu.

3.3.1 Hardvérové a softvérové vybavenie

Pre účely realizácie praktickej časti diplomovej práce bolo testovacie prostredie vybudované na jednom fyzickom hostiteľskom počítači, na ktorom boli vytvorené a prevádzkované všetky potrebné virtuálne komponenty. Cieľom bolo zabezpečiť dostatočný výpočtový výkon a stabilné prostredie pre konzistentné merania.

Testy boli vykonané na notebooku, ktorého kľúčové hardvérové špecifikácie boli nasledovné:

- **procesor** – AMD Ryzen 7 7435HS s taktom 3.10 GHz (Gigahertz),
- **RAM (Random-access memory)** – 16.0 GB (Gigabyte),
- **typ systému** – 64-bitový operačný systém, procesor pre platformu x64,
- **operačný systém** – Microsoft Windows 11 Pro, verzia 24H2, zostava operačného systému 26100.4061.

Ako virtualizačná platforma bol použitý VMware Workstation Player, ktorý umožnil vytvorenie a správu virtuálnych strojov v izolovanom sieťovom prostredí. Konkrétna verzia hypervízora 17.6.1 build-24319023 bola vybraná pre svoju stabilitu a kompatibilitu.

V rámci hypervízora VMware Workstation Player boli vytvorené vždy dva hlavné virtuálne stroje pre každý protokol: VPN Server a VPN Klient. Tieto virtuálne stroje boli konfigurované s dostatočnými hardvérovými prostriedkami, aby zvládli beh rôznych operačných systémov a zabezpečili stabilné prostredie pre meranie výkonu VPN protokolov.

Parametre VPN serveru:

- **pridelená RAM** – 4 GB,
- **pridelené CPU jadrá** – 2,
- **veľkosť virtuálneho disku** – pre operačné systémy na báze Linuxu bola pridelená veľkosť 20 GB a pre operačné systémy Microsoft Windows Server bola pridelená veľkosť 60GB,
- **operačné systémy** – pre rolu VPN servera boli striedavo použité rôzne operačné systémy, aby bola zabezpečená kompatibilita a optimálna implementácia pre každý testovaný VPN protokol. Konkrétne boli využité: Ubuntu Server 24.04.2 LTS (Long Term Support) a Microsoft Windows Server 2022.

Parametre VPN klienta:

- **pridelená RAM** – 4,
- **pridelené CPU jadrá** – 2,
- **veľkosť virtuálneho disku** – pre operačné systémy na báze Linuxu bola pridelená veľkosť 20 GB,
- **operačné systémy** – pre rolu VPN klienta boli na testovanie využité Ubuntu 18.04.6 LT.

Parametre routeru:

- **výrobca** – TP-Link,
- **model** – Archer C6,
- **procesor** – 1,2GHz,
- **LAN (Local Area Network)/WAN (Wide Area Network) porty** – s propustnosťou 1000Mbps (Megabits per second).

4 IMPLEMENTÁCIA TESTOVACIEHO PROSTREDIA

Implementácia testovacieho prostredia podrobne popisuje proces nastavenia a konfigurácie všetkých potrebných komponentov pre realizáciu meraní. Cieľom tejto časti je zabezpečiť reprodukovateľnosť testov a zrozumiteľnosť celého testovacieho prostredia.

4.1 Inštalácia VMware Workstation Player

Základom virtualizačného prostredia, ktoré bolo nevyhnutné pre realizáciu praktickej časti, predstavovala inštalácia hypervízora VMware Workstation Player. Tento bezplatný softvérový nástroj umožnil vytvorenie a správu izolovaných virtuálnych strojov, ktoré slúžili ako VPN server a VPN klient.

Inštalácia hypervízora prebehla na fyzickom hostiteľskom počítači so systémom Microsoft Windows 11 Pro. Inštalačný balíček bol stiahnutý z oficiálnych webových stránok spoločnosti VMware. Po stiahnutí bola spustená inštalačná aplikácia, ktorá riadila celý proces. Postup inštalácie bol dokumentovaný formou série screenshotov a zahŕňal nasledujúce kroky:

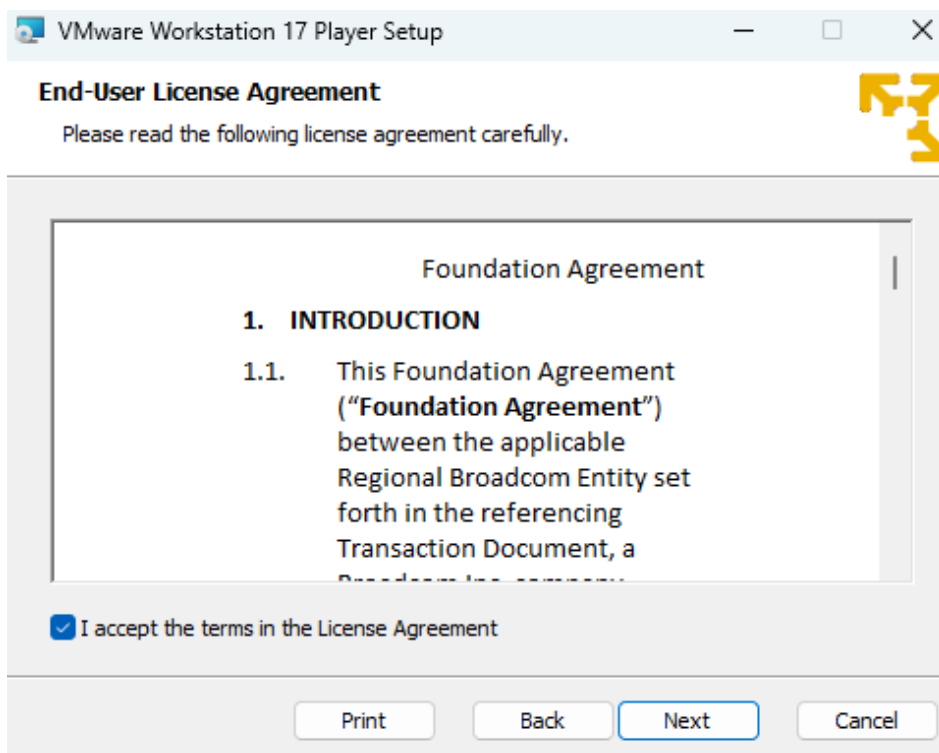
1. **Spustenie inštalátora** – Po spustení inštalačnej aplikácie sa objavilo úvodné okno, ktoré informovalo o začiatku procesu inštalácie. Pokračovalo sa kliknutím na tlačidlo s nápisom “Next”.



Obrázok 6 Spustenie inštalátora

(zdroj: vlastný)

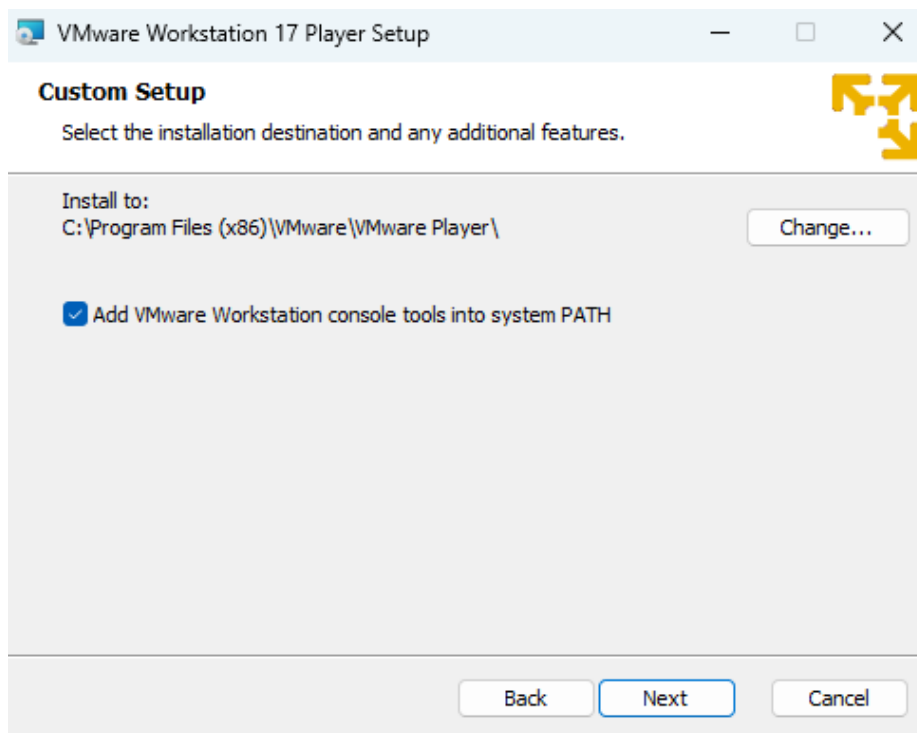
2. **Licenčná dohoda** – Následne bola zobrazená licenčná dohoda s koncovým používateľom. Pre pokračovanie inštalácie bolo potrebné odsúhlasiť podmienky dohody (I accept the terms in the license agreement) a kliknúť na tlačidlo “Next”.



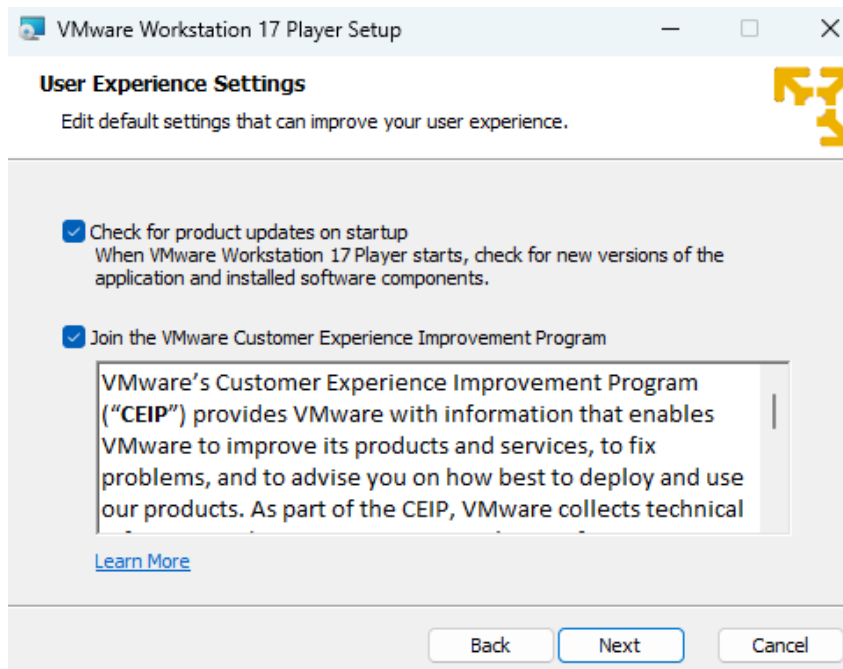
Obrázok 7 Licenčná dohoda

(zdroj: vlastný)

3. **Cesta inštalácie** – V ďalšom kroku bola ponechaná predvolená cesta inštalácie. Kliknutím na Next (Obr. 8) sa prešlo k nasledujúcemu nastaveniu.
4. **Aktualizácie produktu** – Pre zabezpečenie behu na najnovšej verzii bolo odkliknuté pole Check for product updates on startup (Obr. 9). Následne sa kliklo na “Next”.



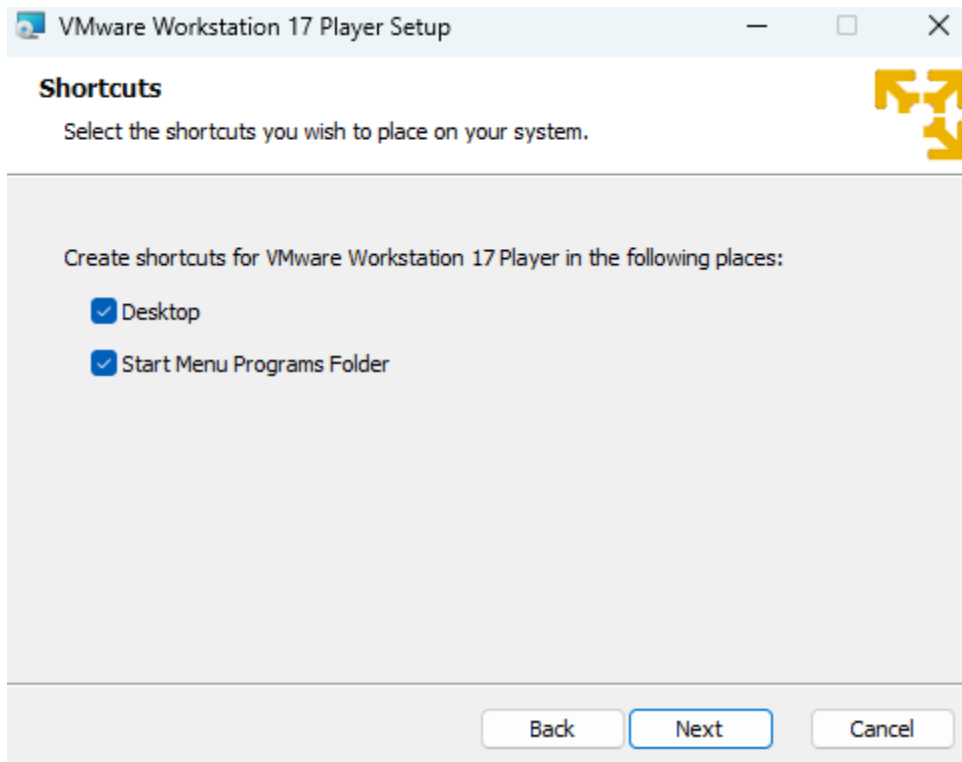
Obrázok 8 Cesta inštalácie

(zdroj: vlastný)

Obrázok 9 Aktualizácie produktu

(zdroj: vlastný)

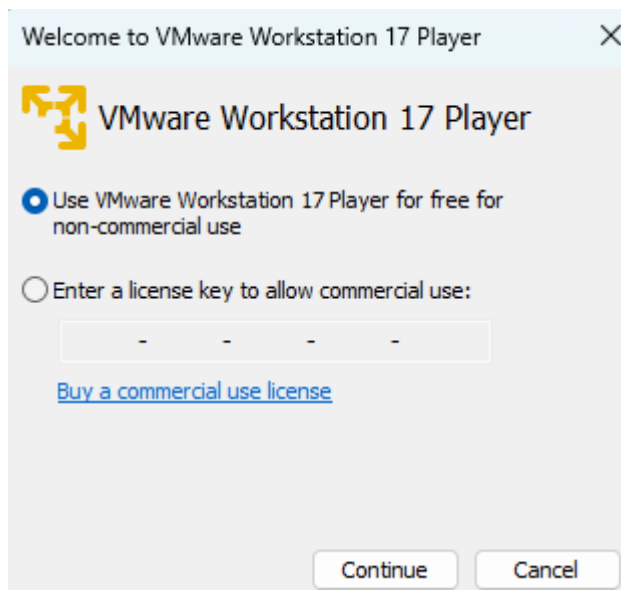
5. **Vytvorenie zástupcov** – Inštalačný proces ponúkol možnosť vytvorenia zástupcov aplikácie. Pre prehľadnosť boli možnosti vytvorenia zástupcov na ploche a v ponuke Štart ponechané ako predvolené (Obr. 10). Kliknutím na “Next” sa pokračovalo.



Obrázok 10 Vytvorenie zástupcov

(zdroj: vlastný)

6. **Typ licencie** – Keďže bol VMware Workstation Player používaný na nekomerčné účely v rámci študijnej práce, bola vybraná možnosť bezplatného používania Use VMware Workstation Player for free for non-commercial use (Obr. 11). Kliknutím na “Continue” sa potvrdila voľba a spustila inštalácia.



Obrázok 11 Typ licencie

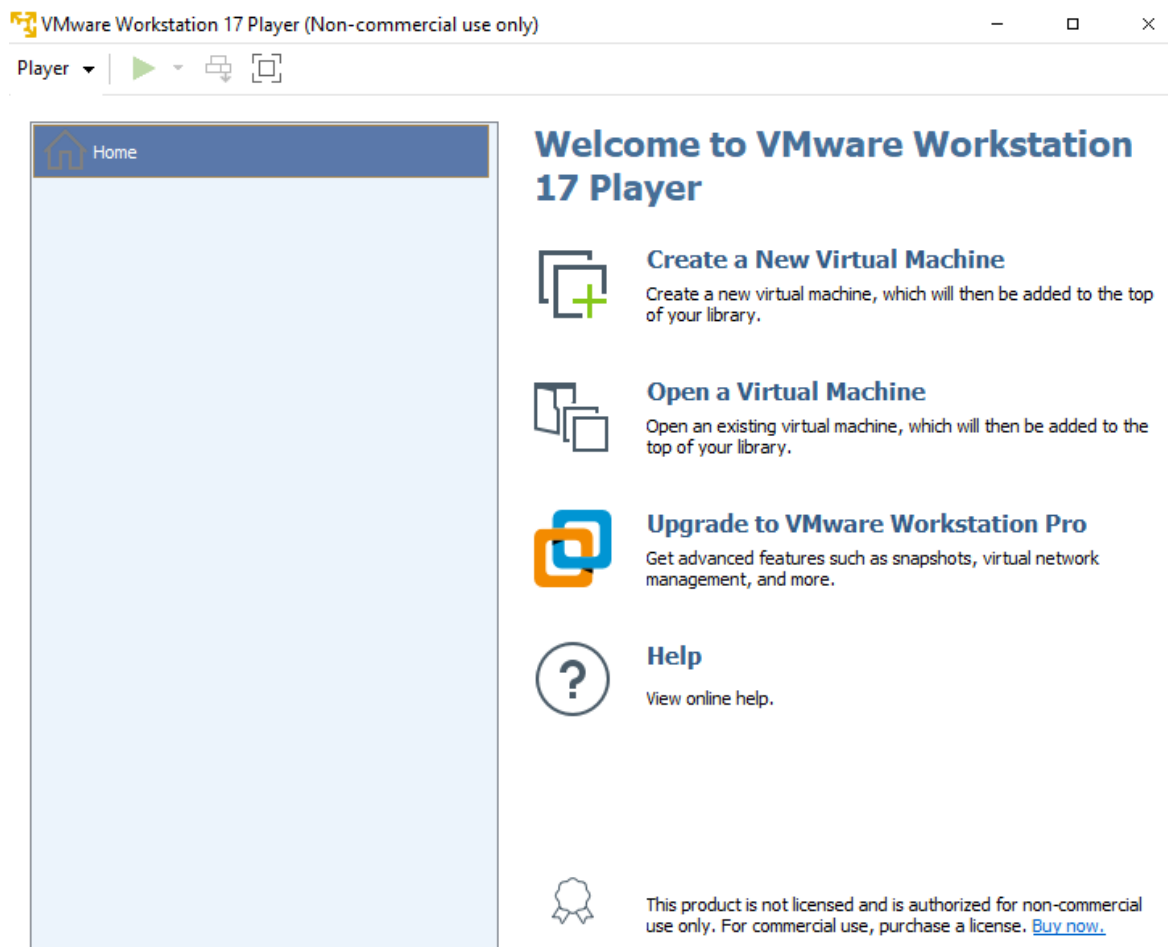
(zdroj: vlastný)

4.2 Vytvorenie virtuálnych strojov a ich konfigurácia

Po úspešnej inštalácii hypervízora VMware Workstation Player bolo nevyhnutné vytvoriť a správne nakonfigurovať virtuálne stroje, ktoré mali plniť rolu VPN servera a VPN klienta. Proces vytvárania bol pre oba typy virtuálnych strojov (Linuxové a Windows, Server a Klient) analogický, líšil sa iba vo výbere inštalačného ISO obrazu a pridelených hardvérových prostriedkoch, ako bolo špecifikované v sekcii 3.3.1.

Postup vytvorenia nového virtuálneho stroja bol nasledovný:

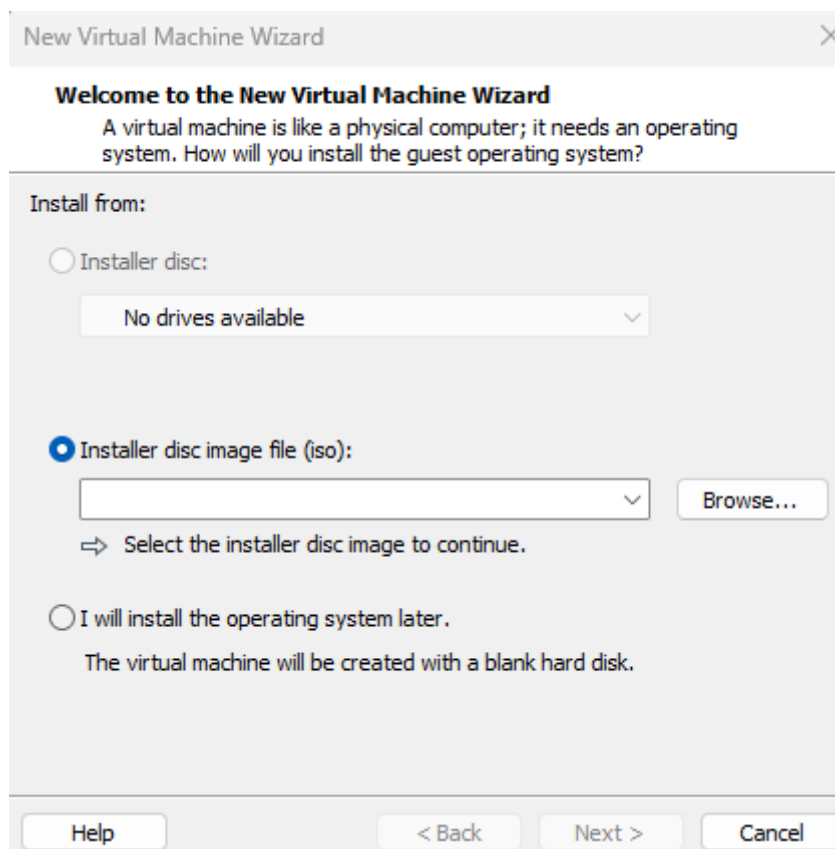
1. **Spustenie sprievodcu** – V hlavnom rozhraní aplikácie VMware Workstation Player bol spustený sprievodca pre vytvorenie nového virtuálneho stroja kliknutím na tlačidlo “Create a New Virtual Machine” (Obr. 12).



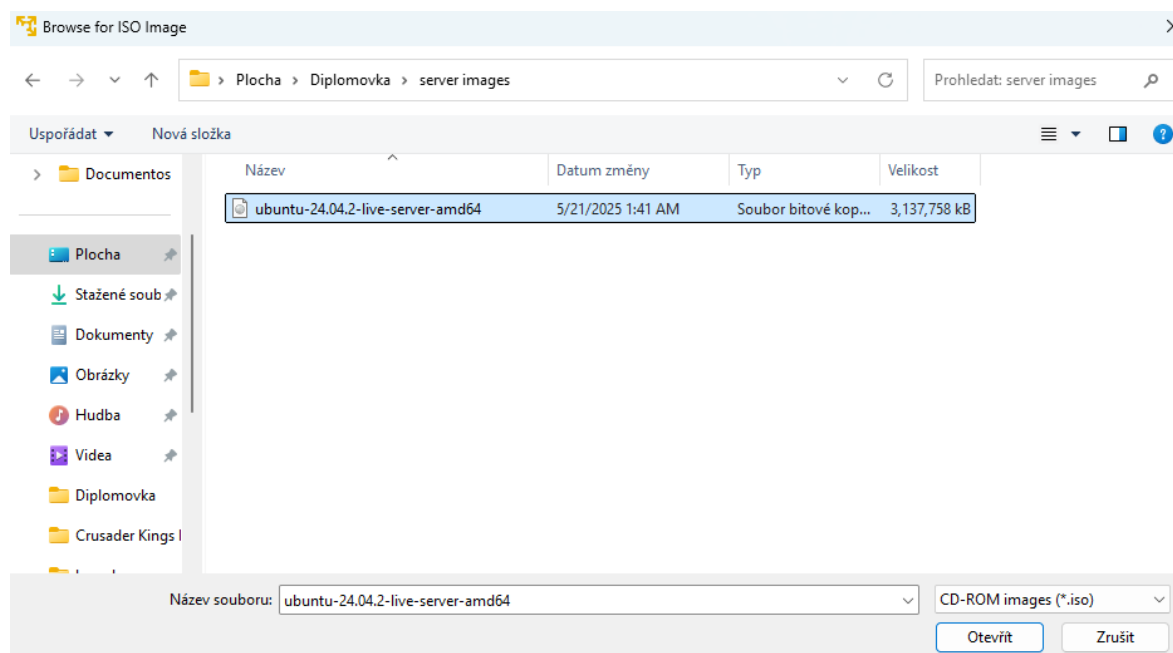
Obrázok 12 Spustenie sprievodcu

(zdroj: vlastný)

2. **Výber inštalačného média** – Následne bola zvolená možnosť Installer disc image file (iso), čo umožnilo vybrať inštalačný ISO obraz operačného systému z lokálneho úložiska. Kliknutím na “Browse” bolo otvorené dialógové okno, v ktorom bola navigovaná cesta k príslušnému ISO súboru. Po výbere iso súboru sa potvrdilo tlačidlom “Otevřít” a pokračovalo sa kliknutím na “Next” (Obr. 13) a (Obr. 14).



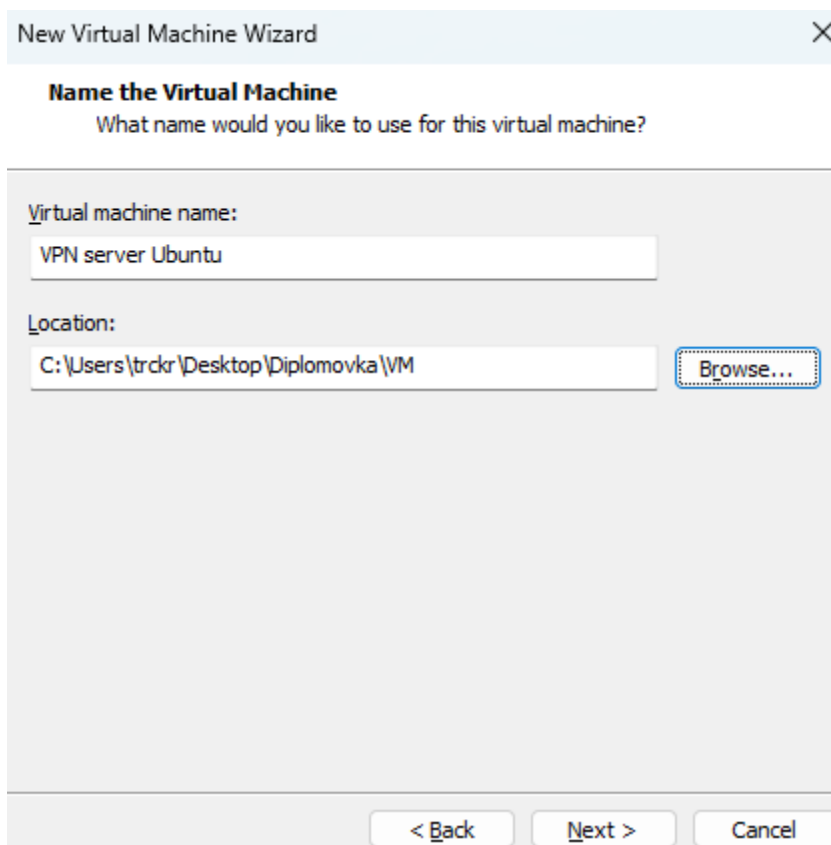
Obrázok 13 Výber inštalačného média

(zdroj: vlastný)

Obrázok 14 Výber inštalačného média pokračovanie

(zdroj: vlastný)

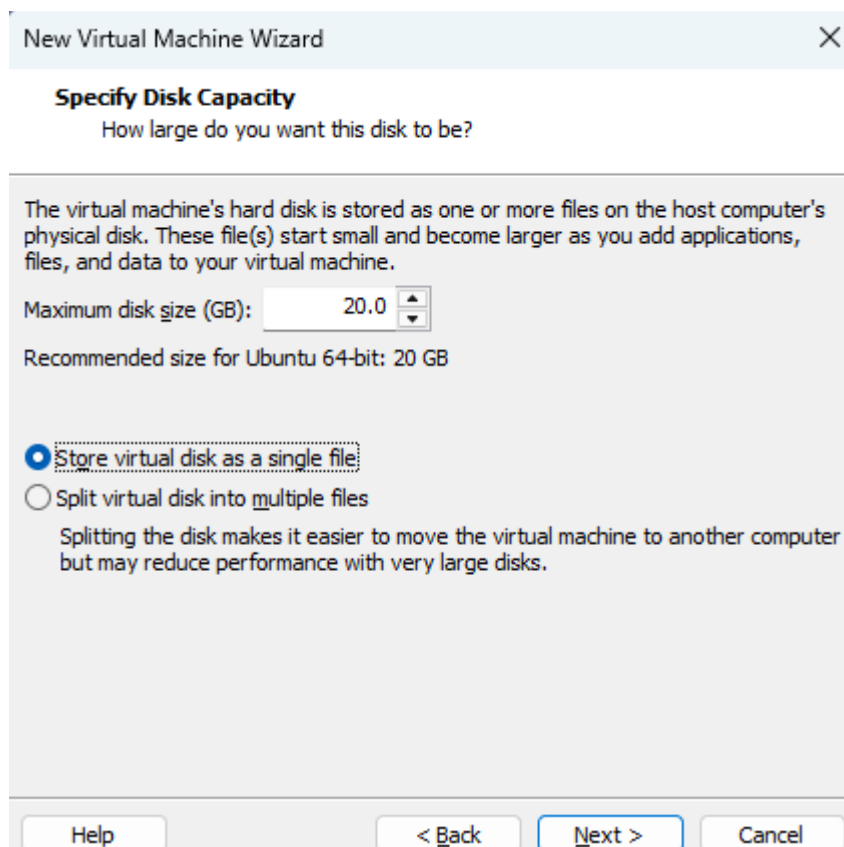
3. **Špecifikácia virtuálneho stroja** – V ďalšom kroku bol virtuálny stroj pomenovaný a bola špecifikovaná cesta k úložisku, kde mali byť súbory virtuálneho stroja uložené. Po zadaní týchto informácií sa pokračovalo tlačidlom “Next” (Obr. 15).



Obrázok 15 Špecifikácia virtuálneho stroja

(zdroj: vlastný)

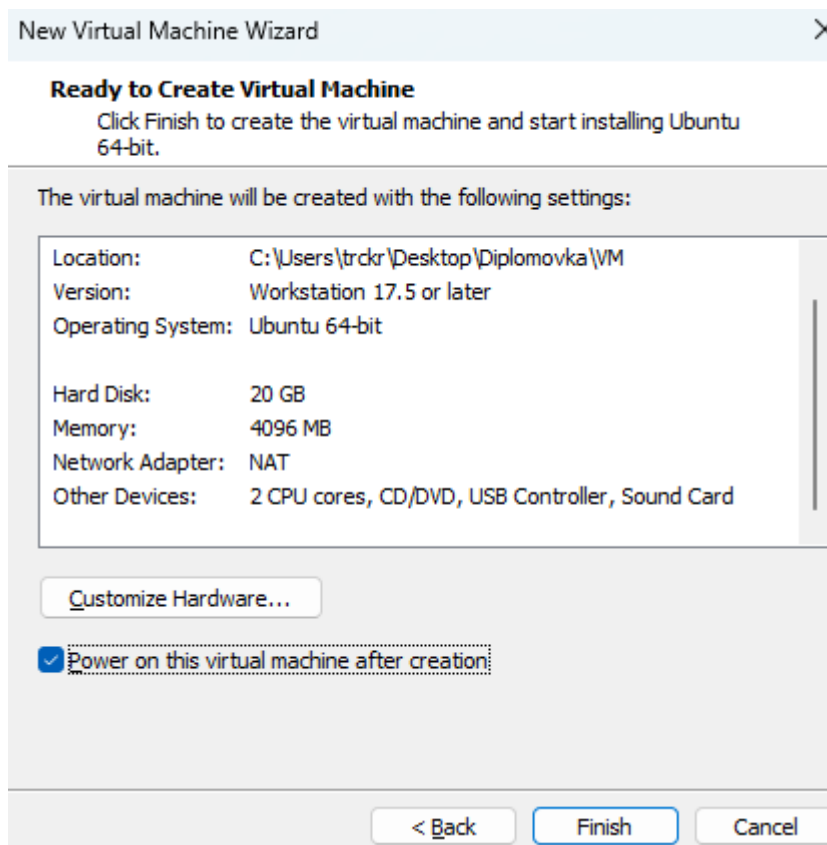
4. **Konfigurácia diskového priestoru** – Sprievodca umožnil nastaviť maximálnu veľkosť virtuálneho disku. V tejto fáze bola pre operačné systémy na báze Linuxu ponechaná odporúčaná veľkosť (20 GB), zatiaľ čo pre operačné systémy Microsoft Windows bola nastavená veľkosť 60 GB. Pre lepšiu správu a výkon bolo vybrané store virtual disk as a single file, po čom sa pokračovalo kliknutím na “Next” (Obr. 16).



Obrázok 16 Konfigurácia diskového priestoru

(zdroj: vlastný)

5. **Prispôsobenie hardvéru** – V poslednom kroku pred dokončením vytvárania bolo k dispozícii okno Customize Hardware (Obr. 17), kde bolo možné upraviť pridelenú RAM, počet procesorových jadier (CPU cores), typ sieťového adaptéra a ďalšie hardvérové parametre podľa potrieb jednotlivých virtuálnych strojov, ako bolo podrobne popísané v sekcii 3.3.1.



Obrázok 17 Prispôsobenie hardvéru

(zdroj: vlastný)

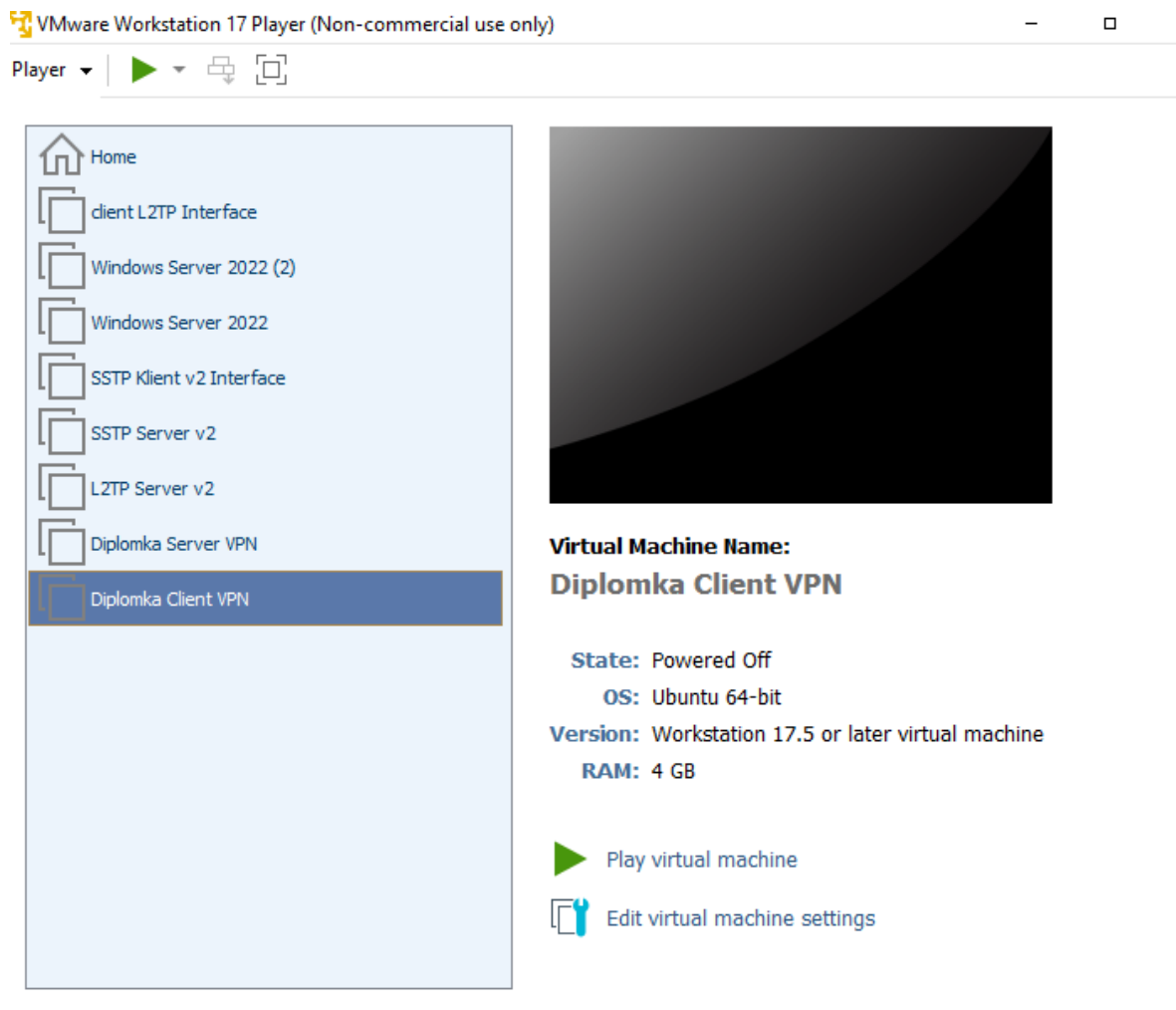
6. **Dokončenie** – Po úprave hardvérových nastavení bol proces vytvárania virtuálneho stroja dokončený kliknutím na tlačidlo “Finish”. Následne sa spustila inštalácia operačného systému vo vnútri novovytvoreného virtuálneho stroja.

4.2.1 Inštalácia a konfigurácia sieťového pripojenia

Po vytvorení základnej štruktúry virtuálnych strojov bolo nevyhnutné správne nakonfigurovať ich sieťové pripojenie, aby sa zabezpečila vzájomná komunikácia a zároveň pripojenie do externej siete a internetu. Kľúčovú úlohu tu zohrával fyzický router, ku ktorému boli virtuálne stroje pripojené v režime "Bridged".

Postup konfigurácie sieťového adaptéra pre každý virtuálny stroj bol nasledovný:

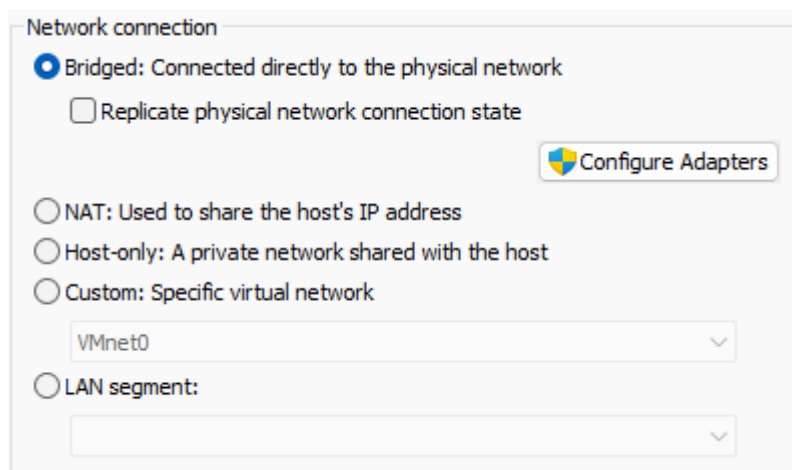
1. **Výber virtuálneho stroja** – V hlavnom rozhraní VMware Workstation Player bol vybraný konkrétny virtuálny stroj z knižnice virtuálnych strojov (Obr. 18).



Obrázok 18 Výber virtuálneho stroja

(zdroj: vlastný)

2. **Úprava nastavení VM** – Následne bolo kliknuté na možnosť Edit virtual machine settings, čím sa otvorilo konfiguračné okno virtuálneho stroja (Obr. 18).
3. **Výber sieťového adaptéra** – V ľavej časti konfiguračného okna bol zvolený komponent Network Adapter. Ak virtuálny stroj nemal adaptér, bolo ho potrebné pridať Add... a vybrať Network Adapter.
4. **Konfigurácia typu pripojenia** – V pravej časti okna, v sekcii Network connection, bola vybraná voľba "Bridged: Connected directly to the physical network". Táto možnosť zaistila, že adaptér virtuálneho stroja sa bude správať ako samostatné fyzické zariadenie v lokálnej sieti a bude priamo komunikovať s fyzickým routerom (Obr. 19).



Obrázok 19 Priradenie k routeru

(zdroj: vlastný)

5. **Uloženie zmien** – Zmeny boli potvrdené kliknutím na tlačidlo OK, čím sa aplikovala konfigurácia sieťového adaptéra pre daný virtuálny stroj.

Po nastavení sieťových adaptérov virtuálnych strojov do režimu Bridged a ich pripojení k fyzickej sieti bolo nevyhnutné overiť správnu funkčnosť sieťového spojenia. Tento krok bol kľúčový pre potvrdenie, že virtuálne stroje sa navzájom vidia, môžu komunikovať v rámci lokálnej siete a pristupovať na internet prostredníctvom fyzického routera, čo je predpokladom pre následné testovanie VPN protokolov. Zistenie IP adries virtuálnych strojov:

- **VM** – (s operačným systémom Linux) bola IP adresa zistená pomocou príkazu `ip a`. Vo výstupe príkazu `ip a` bolo potrebné hľadať riadok začínajúci na `inet` pri prislúchajúcom sieťovom adaptéri (typicky `ens33` alebo `eth0`), ktorý indikoval IP adresu pridelenú sieťovému adaptéru (Obr. 20),

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
lt qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 00:0c:29:5c:3b:5d brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.0.103/24 brd 192.168.0.255 scope global dynamic noprefixroute
ens33
    valid_lft 7157sec preferred_lft 7157sec
    inet6 fe80::a572:9d56:f21e:5f62/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
```

Obrázok 20 Príklad výpisu príkazu ip a

(zdroj: vlastný)

- VM – (s operačným systémom Windows) bola IP adresa zistená pomocou príkazu `ipconfig`. V príkazovom riadku alebo PowerShell.

```
PS C:\Users\Administrator> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::7413:da7b:a0b5:f3c8%7
    IPv4 Address. . . . . : 192.168.0.100
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1
PS C:\Users\Administrator>
```

Obrázok 21 Príklad výpisu ipconfig

(zdroj: vlastný)

Pre overenie vzájomnej konektivity pomocou príkazu `ping` na IP adresu druhej VM. Očakávané boli úspešné odpovede, potvrdzujúce konektivitu (Obr. 22),

```
PING 192.168.0.100 (192.168.0.100) 56(84) bytes of data.  
64 bytes from 192.168.0.100: icmp_seq=1 ttl=128 time=0.494 ms  
64 bytes from 192.168.0.100: icmp_seq=2 ttl=128 time=0.331 ms  
64 bytes from 192.168.0.100: icmp_seq=3 ttl=128 time=0.552 ms  
64 bytes from 192.168.0.100: icmp_seq=4 ttl=128 time=0.399 ms  
^C  
--- 192.168.0.100 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3076ms  
rtt min/avg/max/mdev = 0.331/0.444/0.552/0.085 ms
```

Obrázok 22 Overenie konektivity VM pomocou ping

(zdroj: vlastný)

Úspešné vykonanie týchto testov potvrdilo, že základná sieťová infraštruktúra bola funkčná a pripravená pre inštaláciu a konfiguráciu VPN protokolov.

4.2.2 Inštalácia a konfigurácia Ubuntu Server

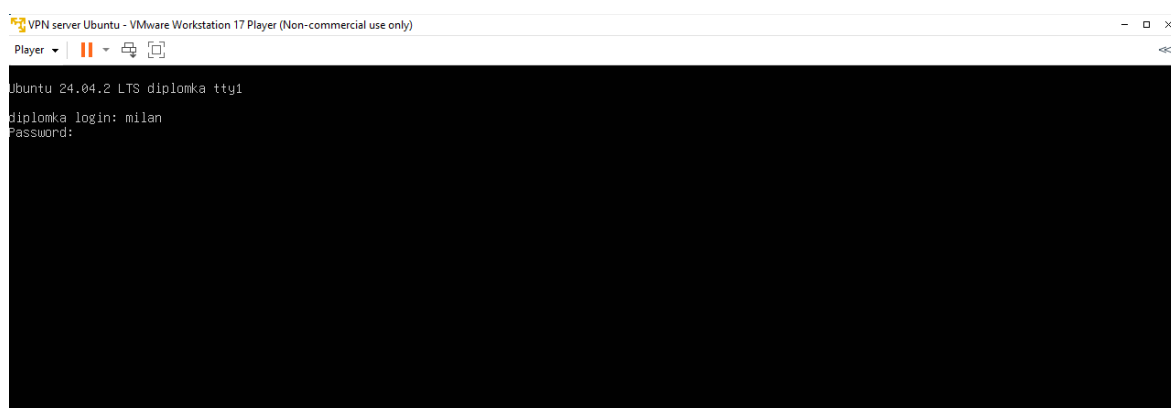
Po úspešnom vytvorení základnej štruktúry virtuálneho stroja, ktorý bol určený pre rolu VPN servera, nasledovala inštalácia operačného systému. Pre tento účel bol zvolený Ubuntu Server 24.04.2 LTS, ktorý bol nainštalovaný z príslušného ISO obrazu pripojeného k virtuálnemu stroju.

Postup inštalácie Ubuntu Servera zahŕňal nasledujúce kroky:

1. **Úvodná obrazovka inštalátora** – Po spustení virtuálneho stroja s pripojeným ISO obrazom Ubuntu Servera sa zobrazila úvodná obrazovka inštalátora. Pre spustenie inštalačného procesu bola vybraná možnosť Try or Install Ubuntu Server a voľba bola potvrdená stlačením klávesy Enter.
2. **Výber jazyka** – Následne bol zvolený preferovaný inštalačný jazyk a voľba bola potvrdená stlačením klávesy Enter.
3. **Výber typu inštalácie** – V ponuke inštalácie bolo pomocou medzerníka označené Ubuntu Server a prešlo sa na tlačidlo Hotovo, ktoré bolo potvrdené klávesou Enter.
4. **Konfigurácia siete** – Pre konfiguráciu sieťového adaptéra bola ponechaná predvolená možnosť, ktorá umožňovala automatické získanie IP adresy prostredníctvom DHCP. Následne sa pokračovalo na tlačidlo Hotovo a potvrdilo Enterom.

5. **Rozdelenie úložiska** – Pre jednoduchosť konfigurácie a na účely testovania bola v sekcii rozdelenia úložiska zvolená možnosť Use An Entire Disk a potvrdilo sa tlačidlom Hotovo. Tým sa operačnému systému pridelil celý virtuálny disk.
6. **Nastavenie profilu** – V ďalšom kroku bol nastavený používateľský profil, vrátane mena používateľa, mena servera a prihlasovacích údajov (používateľské meno a heslo). Po vyplnení požadovaných informácií sa pokračovalo na tlačidlo Hotovo a potvrdilo Enterom.
7. **Konfigurácia SSH** – Pre zabezpečenie možnosti vzdialeného prístupu k serveru prostredníctvom Secure Shell (SSH) bol zaškrtnutý checkbox Install OpenSSH server. Toto uľahčilo správu virtuálneho servera z hostiteľského počítača a klienta. Potvrdilo sa tlačidlom Hotovo.
8. **Server Snaps** – Sekcia Featured Server Snaps bola ponechaná s predvolenými možnosťami, čo znamená, že neboli inštalované žiadne ďalšie predpripravené softvérové balíky prostredníctvom Snaps. Následne sa potvrdilo tlačidlom Hotovo.
9. **Spustenie inštalácie a reštart** – Po potvrdení všetkých predchádzajúcich nastavení sa spustila samotná inštalácia operačného systému Ubuntu Server. Proces inštalácie prebiehal automaticky a po jeho dokončení sa zobrazil oznam o potrebe reštartu. Inštalácia bola ukončená stlačením tlačidla Reštartovať nyní (Reštartovať teraz).

Po reštarte bol virtuálny stroj s operačným systémom Ubuntu Server úspešne nainštalovaný a pripravený na ďalšiu konfiguráciu (Obr. 23).



Obrázok 23 Dokončená inštalácia

(zdroj: vlastný)

4.2.3 Inštalácia a konfigurácia Ubuntu Klient

Pre rolu VPN klienta bol zvolený operačný systém Ubuntu Desktop 24.04.2 LTS, ktorý poskytoval grafické rozhranie pre jednoduchšiu správu a testovanie. Inštalácia tohto operačného systému prebiehala taktiež z príslušného ISO obrazu po vytvorení štruktúry virtuálneho stroja.

Postup inštalácie Ubuntu Desktop zahŕňal nasledujúce kroky:

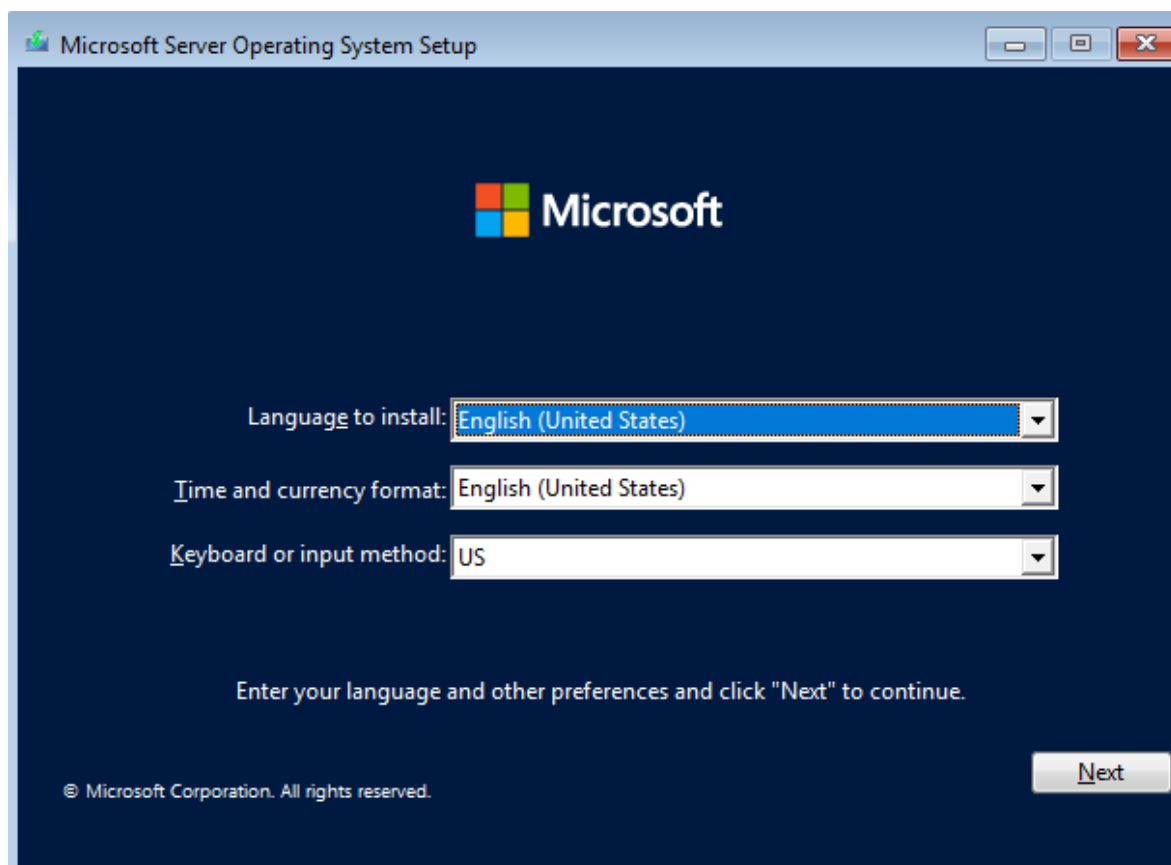
1. **Úvodná obrazovka inštalátora** – Po spustení virtuálneho stroja s pripojeným ISO obrazom Ubuntu Desktop sa objavila úvodná obrazovka. Systém automaticky začal načítavať inštalачné súbory a pripravovať prostredie pre inštaláciu.
2. **Konfigurácia používateľa** – Po načítaní sa zobrazilo okno s výzvou na zadanie používateľských údajov, vrátane mena používateľa, názvu počítača a hesla. Po vyplnení týchto informácií bolo kliknuté na tlačidlo Continue.
3. **Dokončenie inštalácie a reštart** – Po krátkej fáze inštalácie sa operačný systém automaticky nainštaloval a virtuálny stroj sa resetoval. Týmto bol proces inštalácie úspešne dokončený a VM klient bol pripravený na použitie.

4.2.4 Inštalácia a konfigurácia Windows server

Pre rolu VPN servera bol zvolený operačný systém Windows Server 2022 Standard Evaluation Desktop Experience. Tento systém poskytoval známe grafické rozhranie, ktoré uľahčovalo správu a testovanie. Inštalácia prebiehala z príslušného ISO obrazu po vytvorení štruktúry virtuálneho stroja.

Postup inštalácie Windows Servera zahŕňal nasledujúce kroky:

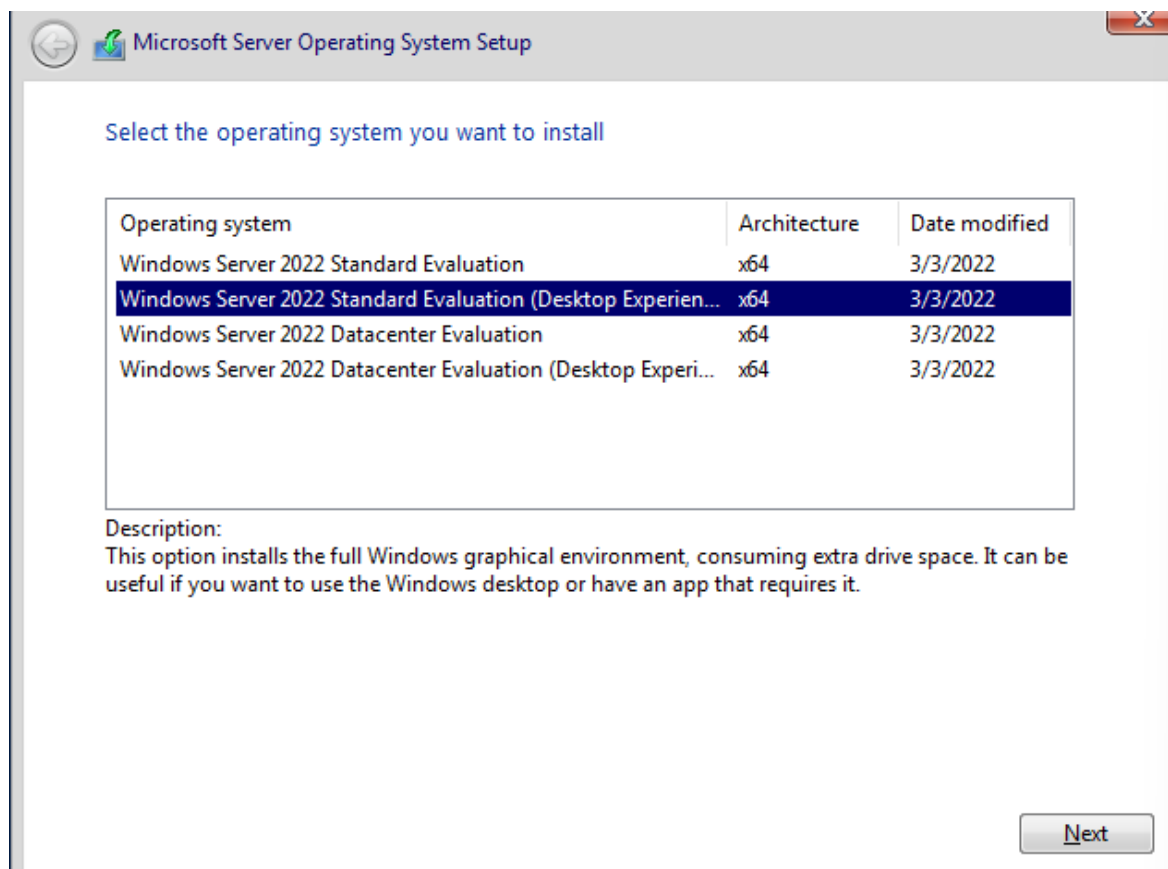
1. **Úvodná obrazovka inštalátora a výber jazyka** – Po spustení virtuálneho stroja s pripojeným ISO obrazom Windows Servera bola zobrazená úvodná obrazovka inštalátora s možnosťou výberu jazyka, času a formátu meny, a rozloženia klávesnice. Po potvrdení týchto nastavení kliknutím na tlačidlo Next bola zobrazená možnosť spustiť inštaláciu (Obr. 24).



Obrázok 24 Výber jazyku Windows server

(zdroj: vlastný)

- 2. Spustenie inštalácie a výber edície** – Na ďalšej obrazovke bolo potrebné kliknúť na tlačidlo **Install now** pre spustenie inštalačného procesu. Následne bol systémom ponúknutý výber edície operačného systému. Bola zvolená možnosť **Windows Server 2022 Standard Evaluation Desktop Experience**, ktorá zahŕňala grafické rozhranie, a pokračovalo sa kliknutím na **Next** (Obr. 25).

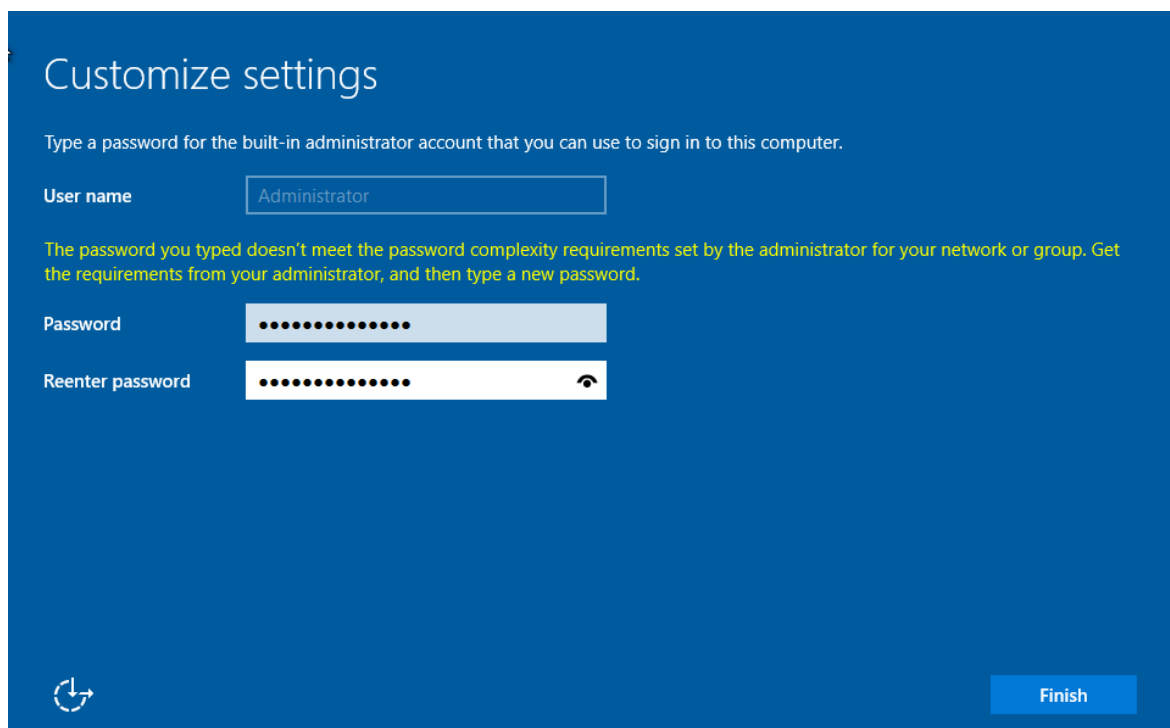


Obrázok 25 Volba edície operačného systému Windows

(zdroj: vlastný)

- 3. Výber disku** – V ďalšom kroku bol vybraný disk, na ktorý mal byť operačný systém nainštalovaný, a bolo kliknuté na Next. Po dokončení inštaláčného procesu a reštartovaní virtuálneho stroja bola zobrazená výzva na zadanie hesla pre účet administrátora. Po zadaní a potvrdení hesla bolo kliknuté na Finish (Obr. 26).

Týmto bol proces inštalácie úspešne dokončený a virtuálny stroj s Windows Serverom bol pripravený na použitie.



Obrázok 26 Výzva na zadanie hesla Windows

(zdroj: vlastný)

4.2.5 Inštalácia nástrojov pre testovanie priepustnosti (iPerf3)

Pre objektívne a kvantitatívne meranie priepustnosti VPN tunelov bol zvolený nástroj iPerf3. Jeho výber bol odôvodnený viacerými faktormi. iPerf3 je open-source nástroj s podporou rôznych operačných systémov (Linux, Windows, macOS), čo zaručilo jeho kompatibilitu s použitým testovacím prostredím. Umožňuje presné meranie priepustnosti TCP a UDP dátového toku, podporuje testovanie v rôznych režimoch (klient/server, obojsmerný prenos) a poskytuje detailné výstupy, ktoré sú kľúčové pre analýzu výkonu. Jeho jednoduchá obsluha a možnosť parametrizácie testov prispievali k reprodukovateľnosti a konzistentnosti meraní.

Inštalácia nástroja iPerf3 prebehla na všetkých virtuálnych strojoch. V prípade operačného systému Ubuntu Server a Ubuntu Klient bol iPerf3 nainštalovaný prostredníctvom správcu balíkov APT (Advanced Package Tool). Postup inštalácie bol nasledovný:

1. **Aktualizácia zoznamu balíkov** – Najprv bol aktualizovaný zoznam dostupných balíkov pre daný operačný systém vykonaním príkazu.

```
sudo apt update
```

2. **Inštalácia iPerf3** – Následne bol nástroj iPerf3 nainštalovaný pomocou príkazu.

```
sudo apt install -y iperf3
```

Prepínač -y automaticky potvrdil všetky otázky týkajúce sa inštalácie závislostí.

Pre potvrdenie základnej funkčnosti a získanie referenčných hodnôt priepustnosti siete bol pred konfiguráciou VPN protokolov vykonaný krátky test priepustnosti medzi virtuálnymi strojmi. Na serverovej VM bol spustený iPerf3 v režime servera pomocou príkazu.

```
iperf3 -s
```

Následne bol z klientskej VM spustený iPerf3 v režime klienta s odkazom na IP adresu servera príkazom.

```
iperf3 -c 192.168.225.130 -t 10
```

Čím sa merala priepustnosť po dobu 10 sekúnd. Úspešné meranie priepustnosti potvrdilo, že nástroj je funkčný a sieťová konektivita je bez problémov, čo predstavovalo východiskový bod pre meranie priepustnosti s aktívnym VPN tunelom.

V prípade operačného systému Windows Server bola inštalácia vykonaná stiahnutím a rozbalením distribučného balíka. Postup inštalácie bol nasledovný:

1. **Stiahnutie iPerf3** – Najprv bol stiahnutý balík iPerf3 určený pre Windows. Pre tento účel bola navštívená oficiálna stránka iPerf3, kde bola stiahnutá najnovšia stabilná verzia iPerf3 pre Windows.
2. **Rozbalenie súborov** – Po stiahnutí bol archív iPerf3 rozbalený do adresára na disku (napríklad C:\iPerf3 alebo C:\Program Files\iPerf3). Odporúčalo sa použiť krátku a ľahko zapamätateľnú cestu pre jednoduchší prístup cez príkazový riadok.
3. **Spustenie iPerf3** – Po rozbalení súborov boli vykonané nasledujúce príkazy v príkazovom riadku pre spustenie iPerf3 v režime servera.

```
cd C:\iperf3
```

```
iperf3.exe -s
```

Príkaz `cd C:\iperf3` zabezpečil navigáciu do adresára, kde bol iPerf3 rozbalený. Následne bol príkazom `iperf3.exe -s` spustený iPerf3 v režime servera, čím bol pripravený prijímať testovacie spojenia (Obr. 27).

```
C:\iperf3>iperf3.exe -s
-----
Server listening on 5201 (test #1)
-----
```

Obrázok 27 Ukážka spustenia iperf3 Windows

(zdroj: vlastný)

4.3 Konfigurácia Wireguard

Implementácia protokolu WireGuard v testovacom prostredí zahŕňala podrobnú konfiguráciu na oboch stranách. Tento proces zahŕňal inštaláciu potrebného softvéru, generovanie asymetrických kryptografických kľúčov pre každú stranu a následnú detailnú úpravu konfiguračných súborov pre nadviazanie bezpečného VPN tunela.

4.3.1 Inštalácia a príprava serverovej strany WireGuard

Inštalácia balíčka WireGuard prebehla prostredníctvom správcu balíkov APT. Pred samotnou inštaláciou bol aktualizovaný zoznam repozitárov pre zabezpečenie inštalácie najnovšej stabilnej verzie.

```
sudo apt update
sudo apt install -y wireguard
```

Príkaz `sudo apt update` slúžil na aktualizáciu zoznamu balíkov v repozitároch, zatiaľ čo `sudo apt install -y wireguard` zabezpečil stiahnutie a inštaláciu všetkých komponentov potrebných pre funkčnosť WireGuardu, pričom prepínač `-y` automaticky potvrdil všetky inštaláčne výzvy.

WireGuard používa na autentifikáciu a šifrovanie komunikácie asymetrické kľúče (privátny a verejný kľúč), ktoré musia byť vygenerované pre každého účastníka tunela. Pre serverovú stranu boli vygenerované vlastné kľúče.

1. **Vytvorenie adresára pre konfiguráciu** – Najprv bol vytvorený vyhradený adresár pre konfiguračné súbory WireGuardu, ktorý zabezpečoval správne oprávnenia.

```
sudo mkdir -p /etc/wireguard
sudo chmod 700 /etc/wireguard
```

Príkaz `sudo mkdir -p /etc/wireguard` vytvoril adresár `/etc/wireguard`, ak ešte neexistoval. Príkaz `sudo chmod 700 /etc/wireguard` následne nastavil oprávnenia tak, aby k tomuto adresáru mal prístup len užívateľ `root`, čo je kritické pre bezpečnosť privátnych kľúčov.

2. **Generovanie privátneho kľúča** – Bol vygenerovaný privátny kľúč pre WireGuard server.

```
wg genkey | sudo tee /etc/wireguard/privatekey > /dev/null
```

Príkaz `wg genkey` vygeneroval privátny kľúč, ktorý bol následne presmerovaný na štandardný vstup príkazu `sudo tee /etc/wireguard/privatekey`. Ten zapísal kľúč do súboru `/etc/wireguard/privatekey` a `> /dev/null` zabezpečil, že sa kľúč nevypíše na obrazovku terminálu, čím sa zvýšila bezpečnosť.

3. **Nastavenie oprávnení pre privátny kľúč** – Kľúčové bolo nastaviť správne oprávnenia pre súbor s privátnym kľúčom, aby k nemu mal prístup len root.

```
sudo chmod 600 /etc/wireguard/privatekey
```

Príkaz `sudo chmod 600 /etc/wireguard/privatekey` nastavil oprávnenia tak, aby súbor mohol čítať a zapisovať len vlastník (root), čím sa zabránilo neoprávnenému prístupu.

4. **Generovanie verejného kľúča z privátneho** – Z vygenerovaného privátneho kľúča bol odvodený zodpovedajúci verejný kľúč.

```
sudo bash -c wg pubkey </etc/wireguard/privatekey>  
/etc/wireguard/publickey
```

Tento príkaz prečítal obsah privátneho kľúča zo súboru `/etc/wireguard/privatekey` a pomocou `wg pubkey` z neho vygeneroval verejný kľúč, ktorý bol uložený do súboru `/etc/wireguard/publickey`.

5. **Nastavenie oprávnení pre verejný kľúč** – Pre verejný kľúč boli nastavené menej reštriktívne oprávnenia, pretože sa jedná o kľúč, ktorý sa zdieľa.

```
sudo chmod 644 /etc/wireguard/publickey
```

Príkaz `sudo chmod 644 /etc/wireguard/publickey` umožnil čítanie súboru pre všetkých užívateľov, zatiaľ čo zápis bol povolený len pre root.

6. **Zobrazenie vygenerovaných kľúčov (pre overenie)** – Pre účely konfigurácie a overenia boli zobrazené obsahy vygenerovaných kľúčov.

```
sudo cat /etc/wireguard/privatekey  
sudo cat /etc/wireguard/publickey
```

Tieto príkazy vypísali privátny a verejný kľúč na obrazovku terminálu, čo umožnilo ich skopírovanie pre ďalšiu konfiguráciu (Obr. 42).

```
milan@diplomka:~$ sudo cat /etc/wireguard/privatekey
GBNlp/4wzIfU6WPWN90x4CjYoze4yquRQqG+Shm/4Ek=
milan@diplomka:~$ sudo cat /etc/wireguard/publickey
SdzxkrHsz/4989PF0BnVofCK3P1z3XVV0W8Z5HkKnG4=
```

Obrázok 28 Výpis privátneho a verejného kľúča

(zdroj: vlastný)

Po inštalácii WireGuardu a vygenerovaní potrebných kryptografických kľúčov bolo nevyhnutné vytvoriť a správne nakonfigurovať konfiguračný súbor WireGuard servera. Tento súbor definuje parametre tunelového rozhrania, privátny kľúč servera, počúvací port a informácie o povolených klientoch. Konfigurácia bola vykonaná v súbore `/etc/wireguard/wg0.conf`.

7. Konfiguračný súbor – bol vytvorený a upravený pomocou textového editora Nano.

```
sudo nano /etc/wireguard/wg0.conf
```

Obsah konfiguračného súboru pre WireGuard server bol definovaný nasledovne [20]:

```
[Interface]
PrivateKey = GBNlp/4mzI1FlGWPWN90X4OJHnze4uqaRQqG+Shm/4Ek=
Address = 10.0.0.1/24
ListenPort = 55555

PostUp = iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A
        POSTROUTING -o ens33 -j MASQUERADE; iptables -A FORWARD -o ens33 -j
        ACCEPT

PostDown = iptables -D FORWARD -i wg0 -j ACCEPT; iptables -t nat -D
        POSTROUTING -o ens33 -j MASQUERADE; iptables -D FORWARD -o ens33 -j
        ACCEPT

[Peer]
PublicKey = ihxV5dgfrqKAYLS910QQ895d4WJqD6EDkpaGfsV11kk=
AllowedIPs = 10.0.0.2/32
```

Popis konfiguračných parametrov servera [20]:

- **[Interface]** – Táto sekcia definuje lokálne nastavenia WireGuard rozhrania servera,
- **PrivateKey** – Obsahuje privátny kľúč WireGuard servera, ktorý bol vygenerovaný v predchádzajúcom kroku. Tento kľúč je tajný a nesmie byť zdieľaný,
- **Address = 10.0.0.1/24** – Prideluje IP adresu 10.0.0.1 virtuálnemu tunelovému rozhraniu wg0 na serveri. /24 definuje masku siete, čo znamená, že tunel bude v podsieti 10.0.0.0/24,

- **ListenPort** = **55555** – Určuje UDP port, na ktorom WireGuard server počúva prichádzajúce pripojenia od klientov. Štandardne je to 51820, ale v tomto prípade bol použitý port 55555,
- **PostUp** – Tento parameter definuje príkazy, ktoré sa vykonajú po úspešnom nadviazaní WireGuard tunela. V tomto prípade ide o pravidlá iptables pre konfiguráciu firewallu a NAT,
- **iptables** – `-A FORWARD -i wg0 -j ACCEPT`: Povoľuje preposielanie dátových paketov prichádzajúcich cez WireGuard tunel (wg0) ďalej do iných sietí,
- **iptables** – `-t nat -A POSTROUTING -o ens33 -j MASQUERADE`: Zabezpečuje preklad sieťových adries (NAT) pre odchádzajúce pakety z VPN tunela. Pakety z 10.0.0.0/24 sú maskované za IP adresu sieťového rozhrania ens33, keď opúšťajú server smerom k hostiteľovi (ak by bola povolená trasa),
- **iptables** – `-A FORWARD -o ens33 -j ACCEPT`: Povoľuje preposielanie dátových paketov odchádzajúcich z tunela (wg0) cez rozhranie ens33 (smerom k klientovi cez VMnet1). Tieto pravidlá boli nastavené na umožnenie komunikácie medzi serverom a klientom v testovacom prostredí,
- **PostDown** – Tento parameter definuje príkazy, ktoré sa vykonajú po zrušení WireGuard tunela. Tieto príkazy rušia pravidlá iptables, ktoré boli nastavené pomocou PostUp (napr. `iptables -D` namiesto `iptables -A`),
- **[Peer]** – Táto sekcia definuje informácie o vzdialenom peerovi, v tomto prípade o WireGuard klientovi,
- **PublicKey** – Verejný kľúč WireGuard klienta. Tento kľúč slúži na identifikáciu a šifrovanie komunikácie smerujúcej ku klientovi,
- **AllowedIPs** – 10.0.0.2/32: Určuje, aké IP adresy sú priradené tomuto klientovi v rámci VPN tunela. Znamená to, že akékoľvek pakety, ktoré majú cieľovú IP adresu 10.0.0.2 (IP adresa klienta v tuneli), budú smerované cez tento peer (klienta). /32 označuje jednu konkrétnu IP adresu.

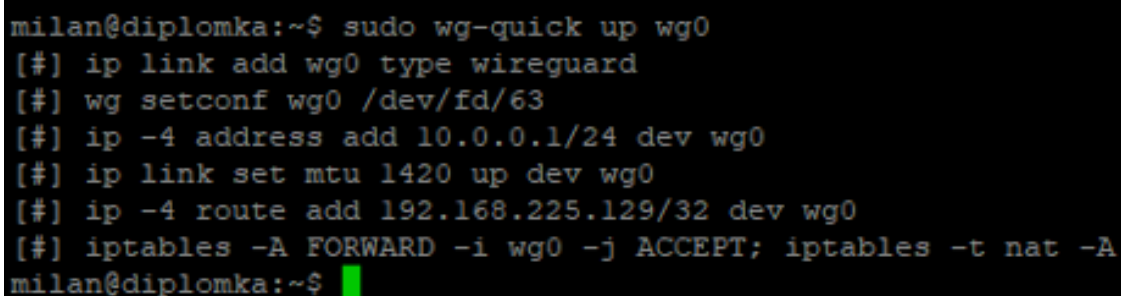
- 8. Povolenie IP preposielania na serveri** – Pre správne fungovanie WireGuard servera ako smerovača pre VPN prevádzku bolo nevyhnutné povoliť preposielanie IP paketov (IP forwarding) v jadre operačného systému.

```
sudo sysctl -w net.ipv4.ip_forward=1
```

- 9. Spustenie a overenie WireGuard služby** – Po vytvorení konfiguračného súboru a povolení IP preposielania bola WireGuard služba spustená.

```
sudo wg-quick up wg0
```

- **sudo wg-quick up wg0** – Spustil WireGuard tunel definovaný konfiguračným súborom /etc/wireguard/wg0.conf.



```
milan@diplomka:~$ sudo wg-quick up wg0
[#] ip link add wg0 type wireguard
[#] wg setconf wg0 /dev/fd/63
[#] ip -4 address add 10.0.0.1/24 dev wg0
[#] ip link set mtu 1420 up dev wg0
[#] ip -4 route add 192.168.225.129/32 dev wg0
[#] iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A
milan@diplomka:~$
```

Obrázok 29 Spustenie tunelu wireguard na servery

(zdroj: vlastný)

4.3.2 Inštalácia a príprava klientskej strany WireGuard

Konfigurácia klientskej strany protokolu WireGuard zahŕňala inštaláciu potrebného softvéru, použitie už vygenerovaných kryptografických kľúčov a vytvorenie konfiguračného súboru pre pripojenie k VPN serveru.

Inštalácia WireGuard na klientskej strane prebehla identickým spôsobom ako na serveri, prostredníctvom správcu balíkov APT.

```
sudo apt update
```

```
sudo apt install -y wireguard
```

Pre klienta boli vygenerované vlastné asymetrické kľúče (privátny a verejný kľúč), ktoré sú nevyhnutné pre autentifikáciu a šifrovanie komunikácie v rámci WireGuard tunela. Postup bol rovnaký ako na serverovej strane:

1. Vytvorenie adresára pre konfiguráciu.

```
sudo mkdir -p /etc/wireguard
```

```
sudo chmod 700 /etc/wireguard
```

2. Generovanie privátneho kľúča klienta.

```
wg genkey | sudo tee /etc/wireguard/privatekey > /dev/null  
sudo chmod 600 /etc/wireguard/privatekey
```

3. Generovanie verejného kľúča klienta z privátneho.

```
sudo bash -c wg pubkey </etc/wireguard/privatekey>  
/etc/wireguard/publickey  
sudo chmod 644 /etc/wireguard/publickey
```

4. Vytvorenie konfiguračného súboru klienta – Konfiguračný súbor pre WireGuard klienta bol vytvorený a upravený v súbore /etc/wireguard/wg0.conf pomocou textového editora Nano. Tento súbor definuje lokálne nastavenia klientskeho rozhrania a parametre pripojenia k WireGuard serveru. Predpokladalo sa, že privátny a verejný kľúč klienta boli už predtým vygenerované (rovnakým postupom ako na serveri) a ich hodnoty boli vložené do tohto konfiguračného súboru [20].

```
sudo nano /etc/wireguard/wg0.conf  
[Interface]  
PrivateKey = oLJq13m328MvKKIXGaJqA2RgCFUVwqIY2U9SiJI/gmkk=  
Address = 10.0.0.2/24  
  
[Peer]  
PublicKey = SdzxkrHsz/4989PF0BnVofCK3Plz3XVVOW8Z5HkKnG4=  
Endpoint = 192.168.225.128:55555  
AllowedIPs = 10.0.0.1/32  
#PersistentKeepalive = 25 # Voliteľné, pre udržanie tunelu aktívnym
```

Popis konfiguračných parametrov klienta [20]:

- **[Interface]** – Táto sekcia definuje lokálne nastavenia WireGuard rozhrania klienta,
- **PrivateKey**– Obsahuje privátny kľúč WireGuard klienta, ktorý bol vygenerovaný špecificky pre túto stranu,
- **Address = 10.0.0.2/24** – Prideluje IP adresu 10.0.0.2 virtuálnemu tunelovému rozhraniu wg0 na klientovi v rámci VPN siete,
- **[Peer]** – Táto sekcia definuje informácie o vzdialenom WireGuard serveri, ku ktorému sa klient pripája,

- **PublicKey** – Verejný kľúč WireGuard servera. Slúži na autentifikáciu servera a šifrovanie komunikácie smerujúcej na server,
- **Endpoint** = **192.168.225.128:55555** – Určuje verejnú IP adresu (alebo hostname) a port WireGuard servera, na ktorý sa klient pripája. IP adresa 192.168.225.128 je IP adresa Server VM v sieti, a 55555 je port, na ktorom WireGuard server počúva,
- **AllowedIPs** = **10.0.0.1/32** – Tento parameter informuje klienta, ktoré IP adresy sú dostupné cez tento WireGuard tunel. V tomto prípade, akýkoľvek sieťový prenos smerujúci na IP adresu 10.0.0.1 (IP adresa VPN Servera vo VPN tuneli) bude smerovaný cez WireGuard tunel,
- **#PersistentKeepalive** = **25** – Tento parameter (v tomto prípade zakomentovaný) je voliteľný a v produkčných prostrediach by mohol zabezpečiť pravidelné odosielanie keepalive paketov každých 25 sekúnd. To by mohlo pomôcť udržiavať tunel aktívny cez NAT zariadenia, ale v tomto testovacom prostredí nebol nevyhnutný.

Po vytvorení a uložení konfiguračného súboru bol WireGuard tunel na klientskej strane aktivovaný pomocou príkazu `wg-quick`.

```
sudo wg-quick up wg0
```

Tento príkaz spustil WireGuard rozhranie `wg0` na základe konfigurácie v súbore `/etc/wireguard/wg0.conf`.

4.3.3 Overenie funkčnosti VPN tunela Wireguard

Po úspešnej inštalácii a konfigurácii WireGuardu na oboch virtuálnych strojoch (server a klient), ako aj po ich spustení, bolo kľúčové overiť, či je VPN tunel skutočne aktívny a funkčný. Tento krok zahŕbal kontrolu stavu rozhrania a testovanie sieťovej konektivity v rámci vytvoreného VPN tunela.

Overenie stavu WireGuard rozhrania a nadviazaného Handshake na oboch virtuálnych strojoch Server VM aj Klient VM bol spustený príkaz.

```
sudo wg show
```

Tento príkaz poskytol komplexný prehľad o aktívnom WireGuard rozhraní, vrátane detailov o nadviazanom spojení.

- **Serverová strana** – Výstup příkazu zobrazil informace o rozhraní wg0 servera, vrátane jeho privátneho kľúča (zahashovaného), pridelenej IP adresy 10.0.0.1 a počúvacieho portu 55555. Kľúčové bolo zobrazenie sekcie peer, kde boli uvedené údaje o pripojenom klientovi, jeho verejný kľúč a predovšetkým indikácia **latest handshake** a hodnota **transfer**. Tieto informácie potvrdili, že server úspešne komunikuje s klientom a tunel je nadviazaný. (Obr. 30)

```
milan@diplomka:~$ sudo wg show
interface: wg0
  public key: SdzxkrHsz/4989PF0BnVofCK3P1z3XVV0W8Z5HkKnG4=
  private key: (hidden)
  listening port: 55555

peer: ihxV5dgfrqKAYLS910QQ895d4WJqD6EDkpaGfsVl1kk=
  endpoint: 192.168.225.129:46120
  allowed ips: 10.0.0.2/32
  latest handshake: 4 seconds ago
  transfer: 180 B received, 92 B sent
milan@diplomka:~$
```

Obrázok 30 wg show server

(zdroj: vlastný)

- **Klientska strana** – Podobne, výstup príkazu na klientovi zobrazil detaily o jeho rozhraní wg0 s IP adresou 10.0.0.1 a informácie o pripojenom peerovi (serveri). Prítomnosť **latest handshake"** a rastúca hodnota **transfer** aj na klientskej strane jednoznačne potvrdili obojsmernú funkčnosť a stabilitu WireGuard tunela. (Obr. 31)

```
milan@diplomkaclient:~$ sudo wg show
interface: wg0
  public key: ihxV5dgfrqKAYLS910QQ895d4WJqD6EDkpaGfsVl1kk=
  private key: (hidden)
  listening port: 46120

peer: SdzxkrHsz/4989PF0BnVofCK3P1z3XVV0W8Z5HkKnG4=
  endpoint: 192.168.225.128:55555
  allowed ips: 10.0.0.1/32
  latest handshake: 16 seconds ago
  transfer: 92 B received, 476 B sent
  persistent keepalive: every 25 seconds
milan@diplomkaclient:~$
```

Obrázok 31 wg show klient

(zdroj: vlastný)

Úspešné vykonanie týchto overovacích krokov potvrdilo, že WireGuard VPN tunel bol správne nakonfigurovaný, aktívny a schopný prenášať dáta medzi Klient VM a Server VM, čím bolo pripravené prostredie pre testovanie priepustnosti.

4.4 Konfigurácia OpenVPN

OpenVPN predstavuje robustné a flexibilné riešenie pre vytvorenie bezpečných VPN tunelov. Na rozdiel od WireGuardu, ktorý sa zameriava na minimalistický prístup a pevne stanovené kryptografické primitíva, OpenVPN ponúka širšie možnosti konfigurácie a podporuje rôzne kryptografické algoritmy. Jeho sila spočíva v jeho flexibilita, schopnosti fungovať nad TCP alebo UDP a silnej podpore komunity. Implementácia OpenVPN v tomto projekte bola zameraná na nastavenie servera aj klienta v sieti s využitím certifikátov pre autentifikáciu a TLS pre dodatočnú bezpečnosť.

4.4.1 Inštalácia a príprava serverovej strany

Inštalácia a konfigurácia OpenVPN servera prebiehala na s operačným systémom Ubuntu Server. Tento proces zahŕňal inštaláciu potrebných balíkov, generovanie kryptografických certifikátov a kľúčov pomocou nástroja Easy-RSA a finálnu konfiguráciu OpenVPN servera.”

Pre inštaláciu OpenVPN a nástroja Easy-RSA, ktorý slúži na správu CA a generovanie kľúčov, bolo potrebné, aby mal virtuálny stroj dočasne prístup k internetu.

```
sudo apt update
sudo apt install -y openvpn easy-rsa
```

Pre zabezpečenie OpenVPN komunikácie sú nevyhnutné digitálne certifikáty a kľúče, ktoré zabezpečujú autentifikáciu servera a klientov. Na ich správu bol použitý nástroj Easy-RSA [21].

1. **Vytvorenie adresára Easy-RSA** – Súbory Easy-RSA boli skopírované do domovského adresára a nastavené oprávnenia.

```
mkdir -p ~/easy-rsa
sudo cp -r /usr/share/easy-rsa/* ~/easy-rsa/
chmod 700 ~/easy-rsa
```

2. **Inicializácia PKI** – Prešlo sa do adresára Easy-RSA a inicializovala sa infraštruktúra verejných kľúčov.

```
cd ~/easy-rsa
./easyrsa init-pki
```

- 3. Generovanie koreňovej CA** – Bola vygenerovaná koreňová CA bez použitia hesla (nopass). CA je kľúčová pre podpisovanie všetkých ostatných certifikátov.

```
./easyrsa build-ca nopass
```

- 4. Generovanie serverového certifikátu a kľúča** – Bol vygenerovaný certifikát a privátny kľúč pre OpenVPN server. Použitý názov "server" je dôležitý pre neskoršiu konfiguráciu.

```
./easyrsa build-server-full server nopass
```

- 5. Generovanie klientskeho certifikátu a kľúča** – Bol vygenerovaný certifikát a privátny kľúč pre OpenVPN klienta.

```
./easyrsa build-client-full client1 nopass
```

- 6. Generovanie Diffie-Hellman parametrov** – Generovanie Diffie-Hellman parametrov (dh.pem) je nevyhnutné pre bezpečnú výmenu symetrických kľúčov počas nadväzovania spojenia. Tento proces môže trvať dlhšiu dobu.

```
./easyrsa gen-dh
```

- 7. Generovanie TLS autentifikačného kľúča (ta.key)** – Pre dodatočnú ochranu proti DDoS útokom a útokom typu TLS handshake boli vygenerované dodatočné TLS autentifikačné kľúče.

```
openvpn --genkey --secret pki/ta.key
```

Tento kľúč pridáva ďalšiu vrstvu zabezpečenia nad rámec samotných certifikátov.

Všetky vygenerované súbory (CA certifikát, serverový certifikát, serverový kľúč, DH parametre a TLS kľúč) boli skopírované do konfiguračného adresára OpenVPN servera (/etc/openvpn/server/).

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

Hlavný konfiguračný súbor pre OpenVPN server bol vytvorený a upravený v `/etc/openvpn/server/server.conf`.

```
sudo nano /etc/openvpn/server/server.conf
```

Obsah konfiguračného súboru `server.conf` [21]:

```
port 443
proto tcp
dev tun
ca /etc/openvpn/server/ca.crt
cert /etc/openvpn/server/server.crt
key /etc/openvpn/server/server.key
tls-crypt /etc/openvpn/server/ta.key
remote-cert-tls client
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist /var/log/openvpn/ipp.txt
push redirect-gateway def1 bypass-dhcp
push dhcp-options DNS 1.1.1.1
push dhcp-options DNS 9.9.9.9
keepalive 10 120
cipher AES-256-GCM
ncp-cipher AES-256-GCM
auth SHA256

user nobody
group nogroup
persist-key
persist-tun
status /var/log/openvpn/openvpn-status.log
verb 3
explicit-exit-notify 1
```

Popis konfiguračných parametrov servera [21]:

- **port 443** – Definuje UDP port, na ktorom OpenVPN server počúva. Port 443 sa často používa, pretože je bežne povolený cez firewally a môže sa vyhnúť blokovaniu,
- **proto tcp** – Určuje použitie TCP protokolu,
- **dev tun** – Vytvorí tunelové rozhranie (vrstva 3) pre VPN,
- **ca /etc/openvpn/server/ca.crt** – Cesta k certifikátu koreňovej certifikačnej autority,
- **cert /etc/openvpn/server/server.crt** – Cesta k certifikátu servera,

- **key /etc/openvpn/server/server.key** – Cesta k privátnemu kľúču servera,
- **tls-crypt /etc/openvpn/server/ta.key** – Aktivuje dodatočnú vrstvu šifrovania TLS handshake a zabráňuje skenovaniu portov,
- **remote-cert-tls client** – Zabezpečuje, že klient musí prezentovať certifikát podpísaný CA a označený ako klientsky certifikát (server overí, či je certifikát určený pre klienta),
- **server 10.8.0.0 255.255.255.0** – Definuje virtuálnu sieť 10.8.0.0/24, z ktorej OpenVPN server prideliť IP adresy klientom. Serveru bude pridelená IP adresa 10.8.0.1,
- **ifconfig-pool-persist /var/log/openvpn/ipp.txt** – Udržiava zoznam pridelených IP adries klientom pre prípad reštartu,
- **push "redirect-gateway def1 bypass-dhcp"** – Nariadi klientovi presmerovať všetku prevádzku (default gateway) cez VPN tunel,
- **push "dhcp-options DNS 1.1.1.1" a push "dhcp-options DNS 9.9.9.9"** – Nariadi klientovi použiť špecifikované DNS servery,
- **keepalive 10 120** – Udržiava spojenie aktívne. Pošle keepalive paket každých 10 sekúnd, a ak nedostane odpoveď 120 sekúnd, predpokladá odpojenie,
- **cipher AES-256-GCM** – Definuje symetrickú šifru AES-256 v režime GCM pre dátový kanál (moderná a bezpečná),
- **ncp-cipher AES-256-GCM** – Povoľuje OpenVPN, aby vyjednal s klientom šifru AES-256-GCM,
- **auth SHA256** – Definuje algoritmus SHA256 pre autentifikáciu dát (kontrolu integrity),
- **user nobody a group nogroup** – Znižuje oprávnenia OpenVPN procesu po jeho spustení, čo zvyšuje bezpečnosť,
- **persist-key a persist-tun** – Zabráňuje opakovanému čítaniu kľúčov a re-otváraníu tunelového rozhrania pri reštarte,
- **status /var/log/openvpn/openvpn-status.log** – Definuje cestu k logu stavu OpenVPN,
- **verb 3** – Nastavuje úroveň podrobnosti logovania (3 je stredná úroveň).

- **explicit-exit-notify 1** – Informuje klienta o odpojení [21].

Pre správne fungovanie OpenVPN servera bolo potrebné upraviť nastavenia jadra a firewallu.

1. **Povolenie IP preposielania** – V súbore `/etc/sysctl.conf` bolo odkomentované alebo pridané.

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

Zmeny boli aplikované spustením príkazu.

```
sudo sysctl -p
```

2. **Nastavenie NAT** – Pre umožnenie klientom prístupu k internetu cez VPN server (ak by bol server pripojený k internetu a klientom to bolo povolené) bolo nastavené pravidlo NAT.

```
sudo iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o ens33 -j MASQUERADE
```

Toto pravidlo prepíše zdrojovú IP adresu paketov prichádzajúcich z VPN siete (10.8.0.0/24) na IP adresu rozhrania ens33 servera.

3. **Konfigurácia firewallu** – Firewall UFW (Uncomplicated Firewall) bol nakonfigurovaný tak, aby povoloval potrebnú prevádzku a zároveň chránil server. V súbore `/etc/default/uw` bola odkomentovaná a nastavená hodnota `DEFAULT_FORWARD_POLICY` na `ACCEPT`, aby sa povolilo preposielanie paketov:

Následne bolo povolené prichádzajúce UDP spojenie na port 443.

```
sudo ufw allow 443/udp
```

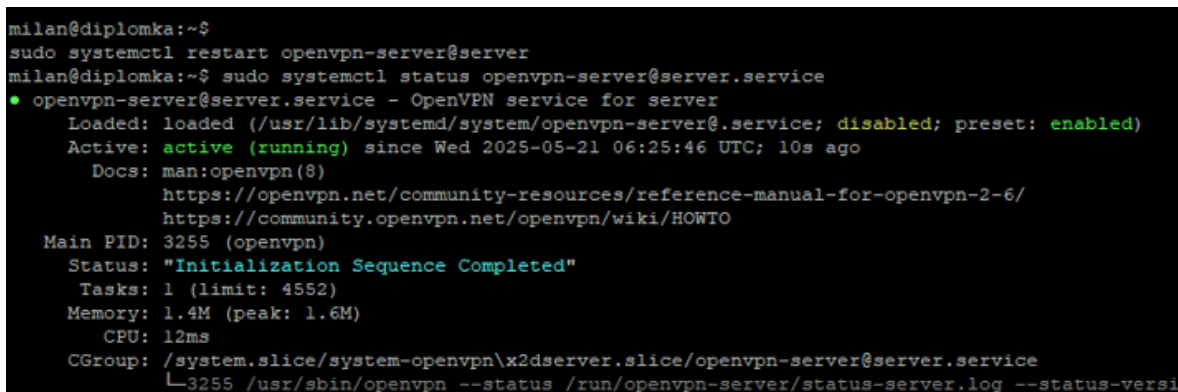
A firewall bol reštartovaný, aby sa zmeny prejavili.

```
sudo ufw disable
```

```
sudo ufw enable
```

Po všetkých konfiguráciách bola OpenVPN služba spustená.

```
sudo systemctl restart openvpn-server@server
```



```
milan@diplomka:~$
sudo systemctl restart openvpn-server@server
milan@diplomka:~$ sudo systemctl status openvpn-server@server.service
● openvpn-server@server.service - OpenVPN service for server
   Loaded: loaded (/usr/lib/systemd/system/openvpn-server@server.service; disabled; preset: enabled)
   Active: active (running) since Wed 2025-05-21 06:25:46 UTC; 10s ago
     Docs: man:openvpn(8)
           https://openvpn.net/community-resources/reference-manual-for-openvpn-2-6/
           https://community.openvpn.net/openvpn/wiki/HOWTO
  Main PID: 3255 (openvpn)
    Status: "Initialization Sequence Completed"
     Tasks: 1 (limit: 4552)
    Memory: 1.4M (peak: 1.6M)
       CPU: 12ms
   CGroup: /system.slice/system-openvpn\x2dserver.slice/openvpn-server@server.service
           └─3255 /usr/sbin/openvpn --status /run/openvpn-server/status-server.log --status-versi
```

Obrázok 32 Spustenie OpenVPN server

(zdroj: vlastný)

4.4.2 Inštalácia a príprava klientskej strany

Konfigurácia klientskej strany protokolu OpenVPN zahŕňala inštaláciu OpenVPN softvéru, prenos certifikátov a kľúčov zo servera a vytvorenie konfiguračného súboru klienta.

Pre inštaláciu OpenVPN na klientskej strane bolo potrebné, aby mal virtuálny stroj dočasne prístup k internetu.

```
sudo apt update
sudo apt install -y openvpn
```

Certifikáty a kľúče potrebné pre klienta (CA certifikát, klientsky certifikát, klientsky kľúč a TLS kľúč) boli prenesené zo servera na klienta pomocou nástroja scp (Secure Copy Protocol). Pre prenos bolo potrebné poznať používateľské meno a IP adresu klienta v sieti (192.168.225.129). Tieto súbory boli pôvodne vygenerované na serveri v adresári ~/easy-rsa/pki/.

```
scp ~/easy-rsa/pki/ca.crt milan@192.168.225.129:/home/milan/
scp ~/easy-rsa/pki/issued/client1.crt milan@192.168.225.129:/home/milan/
scp ~/easy-rsa/pki/private/client1.key milan@192.168.225.129:/home/milan/
scp ~/easy-rsa/pki/ta.key milan@192.168.225.129:/home/milan/
```

Po prenose boli súbory presunuté do správneho konfiguračného adresára OpenVPN klienta a nastavené oprávnenia.

```
sudo mkdir -p /etc/openvpn/client
sudo mv ~/ca.crt /etc/openvpn/client/
```

```
sudo mv ~/client1.crt /etc/openvpn/client/  
sudo mv ~/client1.key /etc/openvpn/client/  
sudo mv ~/ta.key /etc/openvpn/client/
```

Konfiguračný súbor pre OpenVPN klienta bol vytvorený a upravený v `/etc/openvpn/client/client.conf` pomocou príkazu `nano`.

```
sudo nano /etc/openvpn/client/client.conf
```

Obsah konfiguračného súboru `client.conf` [21]:

```
client  
dev tun  
proto tcp  
remote 192.168.225.128 443  
  
resolv-retry infinity  
nobind  
ca /etc/openvpn/client/ca.crt  
cert /etc/openvpn/client/client1.crt  
key /etc/openvpn/client/client1.key  
tls-crypt /etc/openvpn/client/ta.key  
  
cipher AES-256-GCM  
verb 3  
persist-key  
persist-tun
```

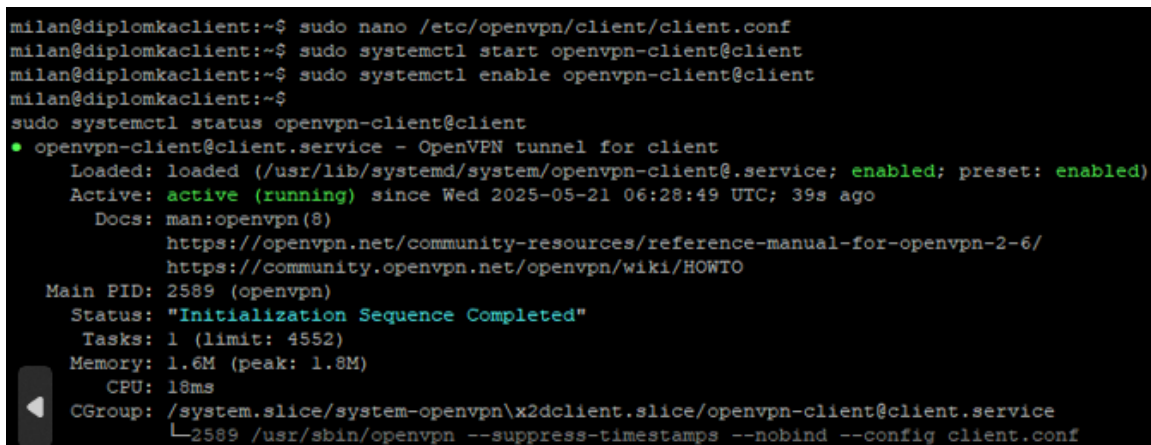
Popis konfiguračných parametrov klienta [21]:

- **client** – Označuje tento konfiguračný súbor ako klientsky,
- **dev tun** – Vytvorí tunelové rozhranie (vrstva 3) pre VPN,
- **proto tcp** – Určuje použitie UDP protokolu, ktorý musí zodpovedať protokolu na serveri,
- **remote 192.168.225.128 443** – Definuje IP adresu a port OpenVPN servera, ku ktorému sa klient pripája,
- **resolv-retry infinity** – Pokračuje v nekonečnom pokuse o opätovné riešenie DNS pre názov servera, ak je zadaný namiesto IP adresy,
- **nobind** – Zabráňuje klientovi viazať sa na špecifický lokálny port, čím sa umožňuje dynamická voľba portu,

- **ca /etc/openvpn/client/ca.crt** – Cesta k certifikátu koreňovej certifikačnej autority,
- **cert /etc/openvpn/client/client1.crt** – Cesta k certifikátu klienta,
- **key /etc/openvpn/client/client1.key** – Cesta k privátnemu kľúču klienta,
- **tls-crypt /etc/openvpn/client/ta.key** – Aktivuje dodatočnú vrstvu šifrovania TLS handshake, musí zodpovedať nastaveniu na serveri,
- **cipher AES-256-GCM** – Definuje symetrickú šifru pre dátový kanál, musí zodpovedať nastaveniu na serveri,
- **verb 3** – Nastavuje úroveň podrobnosti logovania,
- **persist-key a persist-tun** – Zabraňuje opakovanému čítaniu kľúčov a re-otváraní tunelového rozhrania pri reštarte.

Po nakonfigurovaní klienta bola služba OpenVPN spustená.

```
sudo systemctl start openvpn@client
sudo systemctl enable openvpn@client
sudo systemctl status openvpn@client
```



```
milan@diplomkaclient:~$ sudo nano /etc/openvpn/client/client.conf
milan@diplomkaclient:~$ sudo systemctl start openvpn-client@client
milan@diplomkaclient:~$ sudo systemctl enable openvpn-client@client
milan@diplomkaclient:~$ sudo systemctl status openvpn-client@client
● openvpn-client@client.service - OpenVPN tunnel for client
   Loaded: loaded (/usr/lib/systemd/system/openvpn-client@client.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-05-21 06:28:49 UTC; 39s ago
     Docs: man:openvpn(8)
           https://openvpn.net/community-resources/reference-manual-for-openvpn-2-6/
           https://community.openvpn.net/openvpn/wiki/HOWTO
  Main PID: 2589 (openvpn)
    Status: "Initialization Sequence Completed"
     Tasks: 1 (limit: 4552)
    Memory: 1.6M (peak: 1.8M)
       CPU: 18ms
    CGroup: /system.slice/system-openvpn\x2dclient.slice/openvpn-client@client.service
            └─2589 /usr/sbin/openvpn --suppress-timestamps --nobind --config client.conf
```

Obrázok 33 Spustenie OpenVPN klient

(zdroj: vlastný)

Pre overenie aktívneho VPN tunela a pridelených IP adries bolo skontrolované sieťové rozhranie tun0 a otestovaná konektivita. Očakávalo sa, že sa vytvorí nové rozhranie tun0 s pridelenou IP adresou z rozsahu 10.8.0.0/24. Následne bol vykonaný ping test na IP adresu OpenVPN servera v tuneli (10.8.0.1), čo potvrdilo základnú konektivitu.

```
milan@diplomkaclient:~$ ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=64 time=1.24 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=64 time=0.698 ms
64 bytes from 10.8.0.1: icmp_seq=3 ttl=64 time=0.771 ms
64 bytes from 10.8.0.1: icmp_seq=4 ttl=64 time=0.568 ms
^C
--- 10.8.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3092ms
rtt min/avg/max/mdev = 0.568/0.818/1.238/0.252 ms
```

Obrázok 34 ping serveru OpenVPN

(zdroj: vlastný)

4.5 Konfigurácia IKEv2 (StrongSwan)

IKEv2 je robustný a efektívny VPN protokol, ktorý je súčasťou rodiny protokolov IPsec. V porovnaní s OpenVPN ponúka často vyšší výkon a lepšiu stabilitu pri zmene sieťových podmienok, čo ho robí vhodným pre mobilných používateľov. Implementácia IKEv2 v tomto projekte bola realizovaná pomocou open-source softvérového balíka StrongSwan. Konfigurácia zahŕňala nastavenie servera aj klienta v sieti s využitím certifikátov pre autentifikáciu a EAP pre dodatočnú autentifikáciu používateľa.

4.5.1 Konfigurácia a príprava serverovej strany

Inštalácia a konfigurácia IKEv2 servera prebiehala na servery s operačným systémom Ubuntu Server. Proces zahŕňal inštaláciu StrongSwan balíkov, generovanie potrebnej certifikačnej autority, serverového certifikátu a kľúčov, ako aj nastavenie firewallu a IPsec konfigurácie.

Pre inštaláciu StrongSwan a súvisiacich balíkov, ktoré poskytujú nástroje pre správu PKI, bolo potrebné, aby mal virtuálny stroj dočasne prístup k internetu.

```
sudo apt update
sudo apt install -y strongswan strongswan-pki iperf3
```

Pre zabezpečenie IKEv2 komunikácie sú nevyhnutné digitálne certifikáty a kľúče, ktoré zabezpečujú autentifikáciu servera a klientov. Na ich generovanie a správu boli použité nástroje StrongSwan PKI.

1. **Vytvorenie adresárov PKI** – Boli vytvorené štandardné adresáre pre uloženie kľúčov a certifikátov PKI v domovskom adresári používateľa.

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

2. **Generovanie privátneho kľúča CA** – Bol vygenerovaný 4096-bitový RSA (Rivest–Shamir–Adleman) privátny kľúč pre koreňovú certifikačnú autoritu [22].

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

3. **Generovanie CA certifikátu** – Z privátneho kľúča CA bol vygenerovaný a podpísaný CA certifikát s platnosťou 3650 dní (10 rokov). Common Name (CN) bol nastavený na "VPN root CA" [22].

```
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
```

4. **Generovanie privátneho kľúča servera** – Bol vygenerovaný 4096-bitový RSA privátny kľúč pre IKEv2 server [22].

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

5. **Generovanie serverového certifikátu** – Z privátneho kľúča servera bol vygenerovaný a CA podpísaný serverový certifikát s platnosťou 1825 dní (5 rokov). Serverový certifikát bol vydaný s CN (Common Name) a SAN (Subject Alternative Names) zodpovedajúcimi IP adrese servera v sieti (192.168.225.128), čo je kľúčové pre overenie identity servera klientmi [22].

```
pki --pub --in ~/pki/private/server-key.pem --type rsa \
| pki --issue --lifetime 1825 \
--cacert ~/pki/cacerts/ca-cert.pem \
--cakey ~/pki/private/ca-key.pem \
--dn "CN=192.168.225.128" --san @192.168.225.128 --san 192.168.225.128 \
--flag serverAuth --flag ikeIntermediate --outform pem \
> ~/pki/certs/server-cert.pem
```

Všetky vygenerované súbory PKI (CA certifikát, serverový certifikát, serverový kľúč) boli skopírované do systémového adresára StrongSwan (/etc/ipsec.d/), kde ich StrongSwan automaticky načíta.

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

Tento súbor obsahuje privátne kľúče a heslá pre autentifikáciu. Bol upravený tak, aby obsahoval privátny kľúč servera a prihlasovacie údaje pre EAP autentifikáciu klientov.

```
sudo nano /etc/ipsec.secrets
```

Obsah súboru /etc/ipsec.secrets:

```
: RSA "server-key.pem"
```

```
milan : EAP "admin"
```

- : RSA "server-key.pem": Toto riadi StrongSwan, aby použil privátny kľúč servera (server-key.pem, ktorý sa nachádza v /etc/ipsec.d/private/) pre autentifikáciu servera.
- milan : EAP "admin": Definuje užívateľa milan s EAP heslom admin. Toto sa použije pre EAP autentifikáciu klientov [23].

Pre správne fungovanie IKEv2 servera bolo potrebné upraviť nastavenia firewallu a povoliť IP preposielanie. Pravidlá firewallu UFW (Uncomplicated Firewall) boli povolené porty 500 (IKE) a 4500 (NAT Traversal - pre IPsec, keď je za NATom) pre UDP prevádzku.

```
sudo ufw allow 500,4500/udp
```

Následne boli pridané pravidlá do /etc/ufw/before.rules pre povolenie ESP ktorý je kľúčový pre IPsec tunely.

```
sudo nano /etc/ufw/before.rules
```

Do súboru /etc/ufw/before.rules boli pridané nasledujúce riadky:

```
-A ufw-before-forward --match policy --pol ipsec --dir in --proto esp  
-s 10.10.10.0/24 -j ACCEPT  
-A ufw-before-forward --match policy --pol ipsec --dir out --proto esp  
-d 10.10.10.0/24 -j ACCEPT
```

Tieto pravidlá zabezpečujú, že zašifrovaná IPsec prevádzka je povolená cez firewall.

V súbore /etc/ufw/sysctl.conf boli upravené systémové parametre jadra pre správne fungovanie IPsec a preposielanie paketov.

```
sudo nano /etc/ufw/sysctl.conf
```

Do súboru /etc/ufw/sysctl.conf boli pridané/odkomentované nasledujúce riadky:

```
net/ipv4/ip_forward=1  
net/ipv4/conf/all/accept_redirects=0  
net/ipv4/conf/all/send_redirects=0  
net/ipv4/conf/default/accept_redirects=0  
net/ipv6/conf/all/accept_redirects=0  
net/ipv6/conf/default/accept_redirects=0  
net/ipv4/ip_no_pmtu_disc=1
```

Po vykonaní zmien vo firewallových pravidlách a nastaveniach sysctl, bol firewall reštartovaný, aby sa zmeny prejavili.

```
Sudo ufw disable
```

```
Sudo ufw enable
```

Hlavný konfiguračný súbor pre StrongSwan bol upravený v /etc/ipsec.conf.

```
sudo nano /etc/ipsec.conf
```

Obsah konfiguračného súboru ipsec.conf [24]:

```
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=300ms
    rekey=no
    left=%any
    leftid=192.168.225.128
    leftcert=server-cert.pem
    leftsendcert=always
    leftsubnet=0.0.0.0/0
    right=%any
    rightid=%any
    rightauth=eap-mschapv2
    rightsourcexp=10.10.10.100/24
    rightrdns=8.8.8.8,8.8.4.4
    rightsendcert=never
    eap_identity=%identity
    ike=chacha20poly1305-sha512-curve25519-prfsha512!
    esp=chacha20poly1305-sha512!
```

Popis konfiguračných parametrov servera:

- **config setup** – Globálna sekcia konfigurácie.
- **charondebug="ike 1, knl 1, cfg 0"** – Nastavuje úroveň logovania pre rôzne komponenty StrongSwan (IKE, kernel, configuration), čo je užitočné pre diagnostiku.

- **uniqueids=no** – Povoľuje viacerým klientom pripojiť sa s rovnakou identitou (napr. rovnakým certifikátom), čo je užitočné pre roaming užívateľov (aj keď pri EAP sa skôr overuje meno+heslo).
- **conn ikev2-vpn** – Definuje špecifické VPN spojenie s názvom ikev2-vpn.
- **auto=add** – Znamená, že toto spojenie bude automaticky načítané pri štarte StrongSwan, ale nebude automaticky nadviazané.
- **compress=no** – Vypína kompresiu dát, čo môže zlepšiť výkon pri testovaní priepustnosti.
- **type=tunnel** – Definuje typ spojenia ako tunel (vs. transport).
- **keyexchange=ikev2** – Špecifikuje použitie protokolu IKEv2 pre výmenu kľúčov.
- **fragmentation=yes** – Povoľuje fragmentáciu IKE správ, čo pomáha pri prechode cez niektoré NAT zariadenia.
- **forceencaps=yes** – Vynúti zapuzdrenie (encapsulation) IKE a ESP paketov do UDP, čo je dôležité pre NAT Traversal (UDP 4500).
- **dpdaction=clear, dpddelay=300ms** – Nastavenia DPD (Dead Peer Detection) pre detekciu "mŕtvych" peerov. clear znamená, že spojenie bude zrušené, ak peer neodpovie.
- **rekey=no** – Zakazuje automatické obnovovanie kľúčov. Pre testovacie účely v poriadku, v produkcii by sa odporúčalo.
- **left=%any** – Server bude akceptovať pripojenia z ľubovoľnej zdrojovej IP adresy (left je serverová strana).
- **leftid=192.168.225.128** – Identifikácia servera. Musí zodpovedať Common Name alebo Subject Alternative Name v serverovom certifikáte.
- **leftcert=server-cert.pem** – Určuje certifikát, ktorý server použije pre autentifikáciu.
- **leftsendcert=always** – Server bude vždy posilať svoj certifikát klientovi.
- **leftsubnet=0.0.0.0/0** – Definuje, ktoré siete sú dostupné za serverom cez tunel. 0.0.0.0/0 znamená presmerovanie všetkej prevádzky.
- **right=%any** – Klient môže mať akúkoľvek IP adresu (right je klientská strana).
- **rightid=%any** – Klient môže mať akúkoľvek identitu (použije sa EAP).

- **rightauth=eap-mschapv2** – Server bude autentifikovať klientov pomocou EAP-MSCHAPv2 (Microsoft Challenge-Handshake Authentication Protocol version 2).
- **rightsourcexp=10.10.10.100/24** – Rozsah IP adries, ktoré budú pridelené klientom z VPN poolu.
- **rightdns=8.8.8.8,8.8.4.4** – DNS servery, ktoré budú zaslané klientom.
- **rightsendcert=never** – Server nepožaduje, aby klient posielal svoj certifikát (autentifikácia prebehne pomocou EAP).
- **eap_identity=%identity** – Určuje, že IKEv2 bude od klienta vyžadovať EAP identitu.
- **ike=chacha20poly1305-sha512-curve25519-prfsha512!** – Definuje algoritmy pre IKE (fáza 1 - výmena kľúčov): šifrovanie, integrita, Diffie-Hellman skupina, PRF (pseudorandom function). Použitá je moderná ChaCha20-Poly1305 s Curve25519.
- **esp=chacha20poly1305-sha512!** – Definuje algoritmy pre ESP (fáza 2 - dátový tunel): šifrovanie a integrita. Použitá je ChaCha20-Poly1305 [24].

Po všetkých konfiguráciách bola StrongSwan služba spustená a povolená pre automatický štart po reboote.

```
sudo systemctl restart strongswan
sudo systemctl enable strongswan
sudo ipsec statusall
```

4.5.2 Konfigurácia klientskej strany IKEv2

Konfigurácia IKEv2 klienta zahŕňa inštaláciu StrongSwan softvéru, prenos CA certifikátu zo servera a vytvorenie konfiguračných súborov (ipsec.conf a ipsec.secrets).

Pre inštaláciu StrongSwan na klientskej strane bolo potrebné, aby mal virtuálny stroj dočasne prístup k internetu.

```
sudo apt update
sudo apt install -y strongswan
```

CA certifikát (ktorý bol vygenerovaný na serveri) je nevyhnutný pre klienta, aby mohol overiť pravosť serverového certifikátu. Tento certifikát bol prenesený zo servera na klienta (napríklad pomocou scp) a následne umiestnený do správneho adresára StrongSwan na klientovi.

```
scp ~/pki/cacerts/ca-cert.pem milan@192.168.225.129:/home/milan/
```

```
sudo mkdir -p /etc/ipsec.d/cacerts
sudo mv ~/ca-cert.pem /etc/ipsec.d/cacerts/ca-cert.pem
```

Následne bolo overené, či sa certifikát nachádza na správnom mieste:

```
cat /etc/ipsec.d/cacerts/ca-cert.pem
```

```
-----BEGIN CERTIFICATE-----
MIIE8DCCAtigAwIBAgIIOR3ulrmGml8wDQYJKoZIhvcNAQEMBQAwFjEUMBGA1UE
AxMLVlBOIHJvb3QgQ0EwHhcNMjUwNTU5WhcNMzUwNTE5MjUwNTU5WjAW
MRQwEgYDVQQDEwtWUE4gcm9vdCBDQTCCAIwDQYJKoZIhvcNAQEBBQADggIPADCC
AgoCggIBANDoXVWq63GeeuxKoslmUPfjCJ4Fwo3EK4FJZNoXbbdEM4/enZE4g/ne
k/0EGL/19Qk3XC97WtRvbHCkJPIHv7tJ+RBxDDDCDgIuWI7Frs2quXe+LQhWNVc2
gXezESAj0lDSTAmoXosblk3mFy7p2Af2Ev9FOTJE/olDXSMKslrvv2Ic+oPxjHgp
REXRk+AF2m5WiMkvPlDvkpuGSA+hP/d7Mi9Q67muzFaFnPjcGx0mKLLA5k6fr0le
CZylPgfwL/gEtJj3slmuMo79QhMwNeqs/e/A6YQdqxvgRcbDVX/4ch0lQ/FkUiEg
a0iKwWxjMNaxAECp3R/lRdoalui5y/+IW0KQ9kOljkR5E99Ma8l/UlWsJOlCbiml
Jr7T83sdiH35rB+Us8SzGNOFASJ2/BDNfNtuzCr9lMATff9/8SWfTReg2t3o3M0F
g8mOvaqkZ5wth2uBvUBbDejTYMj3F22/5NFD2b12hbaHsW6J4RZLQxdkdF9PR8fv
B5l0ntUeDO8IDSc++0rX23LzoVZwDdFO/cQZ2divMRLYvGPvnpSDji7kkSUHZroM
Fqbp8UmEshKlOhfhOgdgIiQmZm7Dlhd6HfvCRRS1Zku7XnjG3fbY0iR6sG7qPFMo
QlHUIjJYAgjeA245ZsUMllaOf0i9lq2NI2oSb4j2a/0KPx8sF4NpAgMBAAGjQjBA
MA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgEGMBOGA1UdDgQWBBROUYR6
BX5K9wz6w17G33rJlYfgyjANBgkqhkiG9w0BAQwFAAOCAGEA0ENXZk68oGKTqFqR
6qjiCLYnoIeq5igj/D49V/7lpY5mw6N53iMS6sNdliO8PPyHCflniQe6KGjT4gWy
7/Sz+bZrchEiNlY8fFdhTdaxsQZ/OXgs6mVTvvNmY7iDdOWv6ggOmSvpxnIoTZaa
6zxXUKJ804KkvJiaZQGgptitBHndP8bTXWBCw3xD+7JVBSUPbJe2VflT4knXuqih
etqjAI6D880HtQ/0SFGgEJHU4940x4Vum2MYPfUA7pgbotzsDmvkLLvAhvVKHWFd
zPYXr33jKx89VORTby2hpxbFKJq8KNillSk4vUNGbZZgTH3CtSVbQM5XtyxXur2
lrQ8TNuy9LcCA8SPjB5Fxor2Rsy08j7zolqF2E4ZwLSEYjTl6afEeZSMnvIPUPL
P3ruhEqDYedj/E0HFG3u5AieSzxOROO9CR6E0h6IHlOtNQed44h2lE37vEbCNvn2
GrwO6Jw9s8baKTRTOG24rfh/3wPkkHZO5in2/4ttumsC0wh4lyhU2/TCZgggGFNS
tIC4UtElTtA68vrrf9XHPfUxyYlV8R4xsPF2Tzbtcd7fzSyacpH7IYD54q4SwMu
Knbs5hADf0ec/KHeDdnr8MsPXyvkPwbPZwK6IyAI13TbPQSLR6vJfmlrfmncKcsrn
HJUysEUaZEcl7LSuElimdWaTTK8=
-----END CERTIFICATE-----
```

Obrázok 35 Zobrazenie certifikátu ca-cert.pem

(zdroj: vlastný)

Konfigurácia StrongSwan secrets (/etc/ipsec.secrets) na klientovi. Tento súbor obsahuje prihlasovacie údaje pre EAP autentifikáciu klienta.

```
sudo nano /etc/ipsec.secrets
```

Obsah súboru /etc/ipsec.secrets:

```
milan : EAP "admin"
```

- **milan : EAP "admin"** – Definuje prihlasovacie meno milan a heslo admin pre EAP autentifikáciu, ktoré sa použijú pri pripájaní k serveru.

Hlavný konfiguračný súbor pre StrongSwan klienta bol upravený v /etc/ipsec.conf

```
sudo nano /etc/ipsec.conf
```

Obsah konfiguračného súboru ipsec.conf [24]:

```
config setup
    charondebug="ike 1, knl 1, cfg 0"

conn ikev2-rw
    right=192.168.225.128
    rightid=192.168.225.128
    rightsubnet=0.0.0.0/0
    rightauth=pubkey
    leftsourceip=%config
    leftid=milan
    leftauth=eap-mschapv2
    eap_identity=%identity
    auto=start
```

Popis konfiguračných parametrov klienta:

- **config setup** – Globálna sekcia konfigurácie.
- **charondebug="ike 1, knl 1, cfg 0"** – Nastavuje úroveň logovania pre diagnostiku.
- **conn ikev2-rw** – Definuje VPN spojenie s názvom ikev2-rw.
- **right=192.168.225.128** – Definuje IP adresu vzdialeného VPN servera.
- **rightid=192.168.225.128** – Identita vzdialeného servera (musí zodpovedať leftid na serveri).
- **rightsubnet=0.0.0.0/0** – Definuje, že klient bude presmerovať všetku prevádzku na server.
- **rightauth=pubkey** – Klient bude overovať server pomocou jeho verejného kľúča (implicitne cez certifikát podpísaný známou CA).
- **leftsourceip=%config** – Klient získa IP adresu z DHCP poolu servera (t.j., z rozsahu 10.10.10.x).
- **leftid=milan** – Identita klienta, ktorá sa použije pre EAP autentifikáciu.

- **leftauth=eap-mschapv2** – Klient sa bude autentifikovať pomocou EAP-MS-CHAPv2 (meno a heslo).
- **eap_identity=%identity** – Klient bude odosielať svoju EAP identitu.
- **auto=start** – Znamená, že toto spojenie bude automaticky nadviazané pri štarte StrongSwan služby [24].

Po nakonfigurovaní klienta bola StrongSwan služba spustená.

```
sudo systemctl start strongswan-starter
sudo systemctl enable strongswan-starter
sudo ipsec statusall
```

Pre overenie aktívneho VPN tunela bolo na serverovej strane spustený príkaz.

```
sudo ipsec statusall
```

Výstup tohto príkazu potvrdil, že StrongSwan služba beží a konfigurácia ikev2-vpn je načítaná. Kľúčové bolo overenie prítomnosti a stavu SA. Očakávalo sa, že sa v sekcii Security Associations objaví záznam, napríklad ikev2-vpn[1]: ESTABLISHED, ktorý indkuje jedno aktívne (1up) a nadviazané IKEv2 spojenie, čím sa potvrdila úspešná komunikácia a vytvorenie tunela medzi serverom a klientom.

```
Virtual IP pools (size/online/offline):
 10.10.10.0/24: 254/1/0
Listening IP addresses:
 192.168.225.128
 10.0.0.1
Connections:
ikev2-vpn:  %any...%any  IKEv2, dpddelay=300s
ikev2-vpn:  local:  [192.168.225.128] uses public key authentication
ikev2-vpn:  cert:  "CN=192.168.225.128"
ikev2-vpn:  remote: uses EAP_MSCHAPV2 authentication with EAP identity '%any'
ikev2-vpn:  child:  0.0.0.0/0 === dynamic TUNNEL, dpdaction=none
Security Associations (1 up, 0 connecting):
ikev2-vpn[1]: ESTABLISHED 65 seconds ago, 192.168.225.128[192.168.225.128]...192.168.225.129[milan]
ikev2-vpn[1]: IKEv2 SPIs: c8021a16fef4eal2_i f0lee0e785708a22_r*, rekeying disabled
ikev2-vpn[1]: IKE proposal: CHACHA20_POLY1305/PRF_HMAC_SHA2_512/CURVE_25519
ikev2-vpn[1]:  INSTALLED, TUNNEL, reqid 1, ESP in UDP SPIs: c42d2866_i ca8e105f_o
ikev2-vpn[1]:  CHACHA20_POLY1305, 0 bytes_i, 0 bytes_o, rekeying disabled
ikev2-vpn[1]:  0.0.0.0/0 === 10.10.10.1/32
```

Obrázok 36 Výstup ipsec statusall

(zdroj: vlastný)

4.6 Konfigurácia L2TP/IPSec

L2TP/IPSec je často používaný VPN protokol, ktorý kombinuje L2TP pre vytváranie tunelov a IPsec pre šifrovanie a autentifikáciu. Hoci je všeobecne podporovaný a dostupný na väčšine moderných operačných systémov, jeho konfigurácia môže byť komplexnejšia v porovnaní s inými protokolmi, ako je napríklad WireGuard. V tomto projekte bola

implementácia L2TP/IPSec realizovaná pomocou xl2tpd pre L2TP tunelovanie a strongswan pre IPsec zabezpečenie, s použitím pre-shared key pre autentifikáciu a PPP (Point-to-Point Protocol) pre autentifikáciu používateľov.

4.6.1 Inštalácia a príprava serverovej strany

Inštalácia a konfigurácia L2TP/IPSec servera prebiehala na VM s operačným systémom Ubuntu Server. Tento proces zahŕňal inštaláciu potrebných balíkov, konfiguráciu IPsec a L2TP démonov, nastavenie používateľských účtov a úpravu firewallu.

```
sudo apt update
sudo apt install -y strongswan xl2tpd ppp lsof
```

Konfigurácia IPsec (/etc/ipsec.conf). IPsec je zodpovedný za šifrovanie L2TP prevádzky. Konfiguračný súbor /etc/ipsec.conf definuje pravidlá pre IPsec tunel.

```
sudo nano /etc/ipsec.conf
```

Obsah konfiguračného súboru ipsec.conf [24]:

```
config setup
    uniqueids=no
    charondebug="ike 1, knl 1, cfg 1, net 1, esp 1, dmh 1, mgr 1"

conn L2TP-PSK
    keyexchange=ikev1
    authby=secret
    ike=aes256-sha256-modp2048!
    esp=aes256-sha256-modp2048!
    type=transport
    left=%any
    leftprotoport=17/1701
    right=%any
    rightprotoport=17/1701
    auto=add
```

Popis konfiguračných parametrov IPsec:

- **config setup** – Globálna sekcia konfigurácie.
- **uniqueids=no** – Povoľuje viacerým klientom pripojiť sa s rovnakou identitou.
- **charondebug="ike 1, knl 1, cfg 1, net 1, esp 1, dmh 1, mgr 1"** – Nastavuje detailnú úroveň logovania pre diagnostiku rôznych komponentov StrongSwan.

- **conn L2TP-PSK** – Definuje IPsec spojenie pre L2TP s použitím pre-shared key (PSK).
- **keyexchange=ikev1** – Určuje použitie IKEv1 protokolu pre výmenu kľúčov (štandard pre L2TP/IPSec).
- **authby=secret** – Indikuje, že autentifikácia bude prebiehať pomocou pre-shared key (PSK).
- **ike=aes256-sha256-modp2048!** – Definuje algoritmy pre IKE fázu 1 (šifrovanie, integrita, DH skupina). Použitá je silná kombinácia AES-256, SHA-256 a MODP-2048.
- **esp=aes256-sha256-modp2048!** – Definuje algoritmy pre ESP fázu 2 (dátový tunel). Použitá je rovnaká silná kombinácia ako pre IKE.
- **type=transport** – Nastavuje typ IPsec spojenia ako transport, čo znamená, že šifruje existujúce pakety (L2TP) namiesto vytvárania nového tunela.
- **left=%any, right=%any** – Určuje, že server (left) a klient (right) môžu mať ľubovoľnú IP adresu.
- **leftprotoport=17/1701, rightprotoport=17/1701** – Špecifikuje, že IPsec bude zabezpečovať prevádzku na UDP porte 1701 (L2TP) pre obe strany. Protokol 17 je UDP.
- **auto=add** – Znamená, že toto spojenie bude automaticky načítané pri štarte StrongSwan, ale nebude automaticky nadviazané (klient iniciuje) [24].

V tomto súbore je definovaný pre-shared key, ktorý musia poznať server aj klient pre úspešnú IPsec autentifikáciu.

```
sudo nano /etc/ipsec.secrets
```

Obsah súboru /etc/ipsec.secrets:

```
: PSK "admin"
```

- **: PSK "admin"** – Nastavuje pre-shared key na "admin". Toto je kľúč, ktorý bude StrongSwan používať pre autentifikáciu IPsec spojenia [23].

Tento súbor konfiguruje samotný L2TP tunel a pridelovanie IP adres klientom.

```
sudo nano /etc/xl2tpd/xl2tpd.conf
```

Obsah konfiguračného súboru xl2tpd.conf [23]:

```
[global]
port = 1701

[lns default]
ip range = 192.168.200.10-192.168.200.20
local ip = 192.168.200.1
require chap = yes
refuse pap = yes
require authentication = yes
name = L2TP-VPN
ppp debug = yes
pppoptfile = /etc/ppp/options.xl2tpd
length bit = yes
```

Popis konfiguračných parametrov L2TP:

- **[global]** – Globálna sekcia konfigurácie xl2tpd,
- **port = 1701** – L2TP bude počúvať na UDP porte 1701,
- **[lns default]** – Definuje predvolený L2TP Network Server (LNS).
- **ip range = 192.168.200.10-192.168.200.20** – Definuje rozsah IP adres, ktoré budú pridelované L2TP klientom,
- **local ip = 192.168.200.1** – IP adresa, ktorú bude mať L2TP server v rámci tejto virtuálnej L2TP siete,
- **require chap = yes, refuse pap = yes, require authentication = yes** – Nastavenia pre PPP autentifikáciu. Vyžaduje CHAP (Challenge-Handshake Authentication Protocol) a odmieta slabší PAP. Vyžaduje celkovú autentifikáciu,
- **name = L2TP-VPN** – Názov L2TP servera, ktorý sa zobrazí klientom,
- **ppp debug = yes** – Zapína debug režim pre PPP,
- **pppoptfile = /etc/ppp/options.xl2tpd** – Určuje cestu k súboru s ďalšími PPP možnosťami,
- **length bit = yes** – Povolenie bitu dĺžky v L2TP hlavičkách[23],

Tieto súbory riadia PPP komunikáciu a autentifikáciu používateľov.

```
sudo nano /etc/ppp/options.xl2tpd
```

Obsah súboru options.xl2tpd [23]:

```
debug
require-mschap-v2
refuse-mschap
refuse-pap
ms-dns 8.8.8.8
ms-dns 1.1.1.1
asyncmap 0
hide-password
name vpnuser
proxyarp
lcp-echo-interval 30
lcp-echo-failure 4
mtu 1410
mru 1410
```

- **debug** – Zapína PPP debug logovanie,
- **require-mschap-v2, refuse-mschap, refuse-pap** – Definuje metódy PPP autentifikácie. Vyžaduje silnejší MS-CHAPv2 a odmieta staršie a menej bezpečné MS-CHAP a PAP,
- **hide-password** – Skrýva heslo v logoch,
- **name vpnuser** – Predvolené meno pre autentifikáciu L2TP démona voči klientovi,
- **proxyarp** – Povoľuje ARP proxy, čo umožňuje serveru odpovedať na ARP požiadavky pre klientov,
- **lcp-echo-interval 30, lcp-echo-failure 4** – Nastavenia LCP (Link Control Protocol) pre detekciu "mŕtvych" spojení,
- **mtu 1410, mru 1410** – Nastavuje MTU (Maximum Transmission Unit) a MRU (Maximum Receive Unit) pre tunel. Hodnota 1410 je bežná pre L2TP/IPSec.

Používateľské účty v (/etc/ppp/chap-secrets). Tento súbor obsahuje používateľské mená a heslá pre PPP autentifikáciu klientov (CHAP).

```
sudo nano /etc/ppp/chap-secrets
```

Obsah súboru chap-secrets [24]:

```
# Secrets for authentication using CHAP
# client          server  secret          IP addresses
vpnpuser l2tpd    admin      *
```

- **vpnpuser l2tpd admin *** – Definuje používateľa vpnpuser s heslom admin pre server l2tpd. * znamená, že klient môže dostať ľubovoľnú IP adresu z definovaného rozsahu.

Povolenie IP preposielania v súbore `/etc/sysctl.conf` bolo odkomentované alebo pridané a zmeny boli aplikované spustením príkazu.

```
net.ipv4.ip_forward=1
sudo sysctl -p
```

Nastavenie NAT pre umožnenie klientom prístupu k iným sieťam cez VPN server bolo nastavené pravidlo NAT. Toto pravidlo premení zdrojovú IP adresu paketov prichádzajúcich z L2TP siete na IP adresu rozhrania `ens33` servera.

```
sudo iptables -t nat -A POSTROUTING -o ens33 -j MASQUERADE
```

Firewall UFW bol nakonfigurovaný tak, aby povoľoval potrebnú prevádzku pre L2TP/IPSec. Boli povolené UDP porty 500 (IKE), 4500 (NAT Traversal pre IPsec) a 1701 (L2TP).

```
sudo ufw allow 500,4500,1701/udp
```

A firewall bol reštartovaný, aby sa zmeny prejavili.

```
sudo ufw disable
sudo ufw enable
```

Po všetkých konfiguráciách boli služby `ipsec` a `xl2tpd` spustené a povolené pre automatický štart po reboote.

```
sudo systemctl restart ipsec
sudo systemctl enable ipsec
sudo systemctl restart xl2tpd
sudo systemctl enable xl2tpd
```

4.6.2 Konfigurácia klientskej strany

Konfigurácia L2TP/IPSec klienta bola vykonaná na s operačným systémom Ubuntu Desktop s využitím grafického rozhrania `NetworkManager`, ktoré zjednodušuje pripojenie k tomuto typu VPN.

Pre umožnenie konfigurácie L2TP/IPsec v grafickom prostredí `NetworkManager` bolo potrebné nainštalovať príslušný plugin.

```
sudo apt update
sudo apt install network-manager-l2tp-gnome -y
```

Následne bol NetworkManager reštartovaný, aby sa plugin načítal.

```
sudo systemctl restart NetworkManager
```

Proces nastavenia VPN pripojenia prebiehal prostredníctvom grafického užívateľského rozhrania NetworkManageru:

1. Prístup k nastaveniam siete:

- Kliklo sa na ikonu siete vpravo hore.
- Vybralo sa Network Settings.
- Prešlo sa na záložku "VPN".

2. Pridanie nového VPN pripojenia:

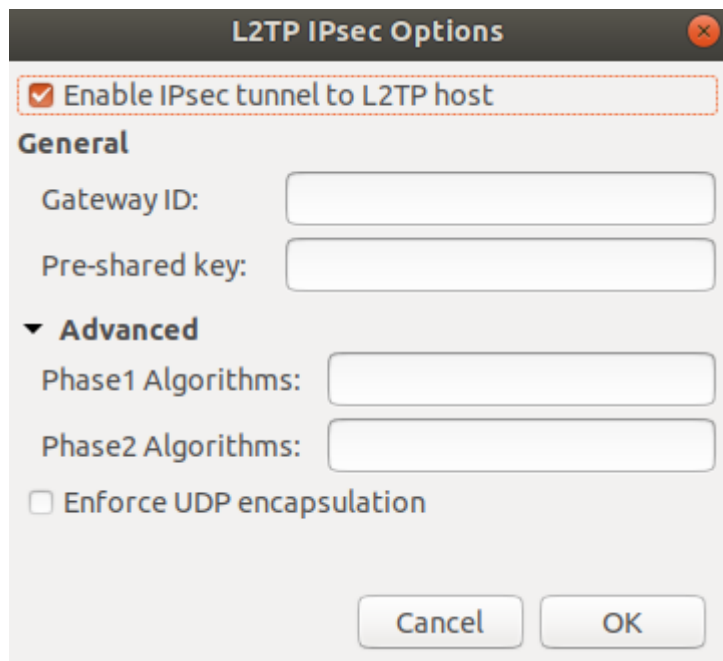
- Kliklo sa na tlačidlo "+" pre pridanie nového pripojenia.
- Zo zoznamu typov VPN sa vybral "Layer 2 Tunneling Protocol (L2TP)".

3. Pre základné nastavenia boli vyplnené nasledujúce parametre:

- **Názov (Name)** – napr. VPN-L2TP-Test (pre jednoduchú identifikáciu).
- **Brána (Gateway)** – IP adresa VPN servera.
- **Užívateľské meno (Username)** – vpnuser.
- **Heslo (Password)** – admin

4. Nastavenia IPsec:

- Bolo zaškrtnuté **"Enable IPsec tunnel to L2TP host"**.
- Do poľa **"Pre-shared key"** bolo zadané **admin** (musí sa zhodovať s PSK na serveri).
- Do sekcie **Phase 1** bol zadán algoritmus IKE z nášho serveru aes256-sha256-modp2048.
- Do sekcie **Phase 2** bol zadán algoritmus ESP z nášho serveru aes256-sha256-modp2048 (Obr. 37).

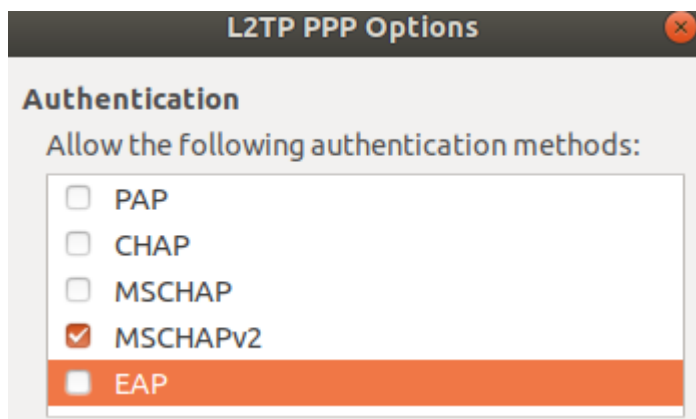


Obrázok 37 Ukážka rozhrania s pokročilými nastaveniami L2TP

(zdroj: vlastný)

5. Nastavenia PPP (PPP Settings):

- V sekcii Authentication Methods bolo zaškrtnuté len MS-CHAPv2.
- Ostatné metódy (PAP, CHAP, MS-CHAP) boli odškrtnuté, aby sa vynútilo použitie silnejšej autentifikácie.



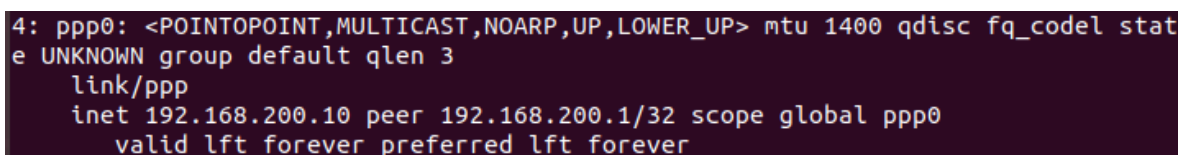
Obrázok 38 Nastavenia PPP

(zdroj: vlastný)

6. **Uloženie a pripojenie** – Po vyplnení všetkých nastavení sa konfigurácia uložila a VPN spustila našu konfiguráciu.

Po pokuse o pripojenie bolo dôležité overiť, či sa L2TP/IPSec tunel úspešne nadviazal.

1. **Kontrola stavu pripojenia v NetworkManager** – V NetworkManager sa malo zobrazit', že VPN pripojenie je "ON".
2. **Kontrola pridelených IP adries na klientovi** – Na klientovi sa otvoril terminál a spustením príkazu `ip` a sa overilo, či sa vytvorilo nové virtuálne rozhranie s pridelenou IP adresou z rozsahu 192.168.200.10-192.168.200.20.



```
4: ppp0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1400 qdisc fq_codel state UNKNOWN group default qlen 3
    link/ppp
    inet 192.168.200.10 peer 192.168.200.1/32 scope global ppp0
        valid lft forever preferred lft forever
```

Obrázok 39 Ukážka výstupu `ip` a L2TP

(zdroj: vlastný)

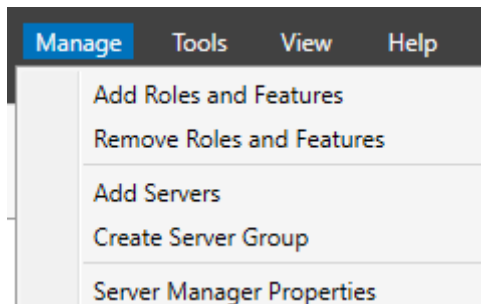
4.7 Konfigurácia SSTP

SSTP je VPN protokol vyvinutý spoločnosťou Microsoft, ktorý poskytoval bezpečné tunelovanie dát cez HTTPS. Vďaka využitiu portu 443 a protokolu TCP bol SSTP často schopný obísť firewally, ktoré by iné VPN protokoly mohli blokovať. Jeho integrácia s SSL/TLS zabezpečovala robustné šifrovanie a autentifikáciu. Hoci bola pôvodne plánovaná komplexná implementácia všetkých VPN protokolov na báze operačných systémov Linux, v prípade SSTP sa ukázalo, že táto stratégia nebola optimálna. Linuxové implementácie SSTP boli limitované, pričom mnohé z relevantných repozitárov sa prestali aktualizovať a ich funkčnosť bola značne obmedzená. Vzhľadom na to, že SSTP je natívne a plne podporovaný operačnými systémami Microsoft, bolo rozhodnuté pre implementáciu a konfiguráciu SSTP servera na platforme Windows Server, čo zaručovalo stabilitu, plnú funkčnosť a jednoduchšiu správu.

4.7.1 Inštalácia a príprava serverovej strany

V nasledujúcej časti budeme pridávať rolu Active Directory Certificate Services.

1. **Otvorenie Správcu servera** – Proces bol zahájený otvorením Server Manager. Z hlavného panelu Správcu servera bola zvolená možnosť Manage a následne bola vybraná položka Add Roles and Features z rozbaľovacieho menu.



Obrázok 40 Ukážka Add Roles and Features

(zdroj: vlastný)

2. **Pred spustením** – Po spustení Add Roles and Features bola dôkladne prečítaná úvodná stránka „Before You Begin“. Po oboznámení sa s informáciami bolo pokračované kliknutím na tlačidlo Next.
3. **Výber typu inštalácie** – V ďalšom kroku bol zvolený typ inštalácie Role-based or feature-based installation, čo je štandardný prístup pre pridávanie služieb na server. Následne bolo opäť kliknuté na Next.
4. **Výber cieľového servera** – Zo zoznamu serverov bol vybraný konkrétny cieľový server (napr. WIN-73CGR0P15MH), na ktorý sa mala služba AD CS (Active Directory Certificate Services) nainštalovať. Výber bol potvrdený kliknutím na Next.
5. **Výber rolí servera** – V zozname Server Roles bola zaškrtnutá možnosť Active Directory Certificate Services.
6. **Pridanie požadovaných funkcií pre Active Directory Certificate Services** – Po výbere roly AD CS sa objavilo kontextové okno s požiadavkou na pridanie súvisiacich funkcií. Bolo zaistené, že možnosť Include management tools bola zaškrtnutá. Kliknutím na tlačidlo Add Features boli tieto súčasti pridané.
7. **Výber funkcií** – Na stránke Features boli skontrolované a potvrdené všetky potrebné funkcie. V tomto prípade boli automaticky pridané nevyhnutné funkcie pre AD CS. Bolo zabezpečené, aby boli nainštalované/zaškrtnuté aj funkcie .NET Framework 3.5 Features a „NET Framework 4.8 Features, ktoré sú často nevyhnutné pre správnu funkčnosť. Následne bolo pokračované kliknutím na Next.

8. **Active Directory Certificate Services** – Na ďalšej stránke bol prečítaný prehľad informácií o službe AD CS. Po preštudovaní obsahu sa prešlo na ďalší krok kliknutím na Next.
9. **Výber rolových služieb** – V rámci sekcie Role services boli zaškrtnuté konkrétne požadované rolové služby:
 - **Certification Authority,**
 - **Certification Authority Web Enrollment.**

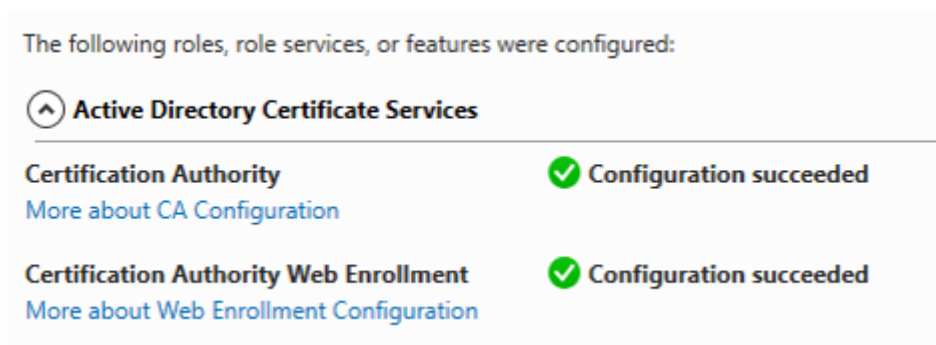
Následne bolo opäť kliknuté na Next.

10. **Potvrdenie** – Na stránke Confirmation boli skontrolované všetky zvolené možnosti inštalácie. Inštalácia bola zahájená kliknutím na tlačidlo Install.

Po úspešnej inštalácii požadovaných rolí Active Directory Certificate Services bolo nevyhnutné pristúpiť k ich podrobnej konfigurácii. Tento proces zahŕňal nastavenie certifikačnej autority a webovej registrácie, čím sa zabezpečila plná funkčnosť a dostupnosť certifikačných služieb.

1. **Priebeh inštalácie a výsledky** – Po úspešnom dokončení inštalačného procesu sa na stránke Results zobrazil stav inštalácie. Po dokončení inštalácie bolo nevyhnutné nakonfigurovať roly AD CS. To bolo realizované kliknutím na odkaz Configure Active Directory Certificate Services on the destination server.
2. **Sprievodca konfiguráciou AD CS** – V prvom kroku Sprievodcu konfiguráciou AD CS bolo zabezpečené, že vybraný užívateľ disponuje administrátorskými oprávneniami. Potom bolo pokračované kliknutím na Next.
3. **Roly služby na konfiguráciu** – Bolo potvrdené, že na konfiguráciu sú vybrané roly Certifikačná autorita a Webová registrácia certifikačnej autority. Kliklo sa na Next.
4. **Typ nastavenia** – Pre novú inštaláciu certifikačnej autority bola zvolená možnosť Enterprise CA, vzhľadom na prostredie s doménou Active Directory. Následne bolo pokračované kliknutím na Next.
5. **Typ CA** – Vzhľadom na to, že sa jednalo o prvú CA v hierarchii, bol zvolený typ Root CA. Kliklo sa na Next.
6. **Súkromný kľúč** – Bola vybraná možnosť Create a new private key. Kliklo sa na „Ďalej“.

7. **Kryptografia** – Predvolené kryptografické možnosti boli akceptované, pretože neboli špecifické požiadavky na zmenu. Následne bolo kliknuté na Next.
8. **Názov CA** – Bol špecifikovaný spoločný názov pre certifikačnú autoritu. V tomto prípade bol akceptovaný predvolený názov, ktorý zvyčajne zodpovedá názvu hostiteľa servera. Kliklo sa na Next.
9. **Doba platnosti** – Bola nastavená doba platnosti certifikátu CA. Predvolená hodnota 5 rokov bola ponechaná. Kliklo sa na Next.
10. **Databáza certifikátov** – Boli špecifikované umiestnenia databázy certifikátov. Predvolené umiestnenia boli akceptované. Následne bolo kliknuté na Next.
11. **Potvrdenie – Webová registrácia** – Sprievodca potvrdil konfiguráciu pre rolu Webová registrácia certifikačnej autority. Konfigurácia bola dokončená kliknutím na tlačidlo Configure.
12. **Výsledky** – Na stránke Results sa zobrazil stav konfigurácie. Po úspešnej konfigurácii sa zobrazila správa Configuration succeeded pre obe roly Certifikačná autorita a Webová registrácia certifikačnej autority. Proces bol ukončený kliknutím na Close.



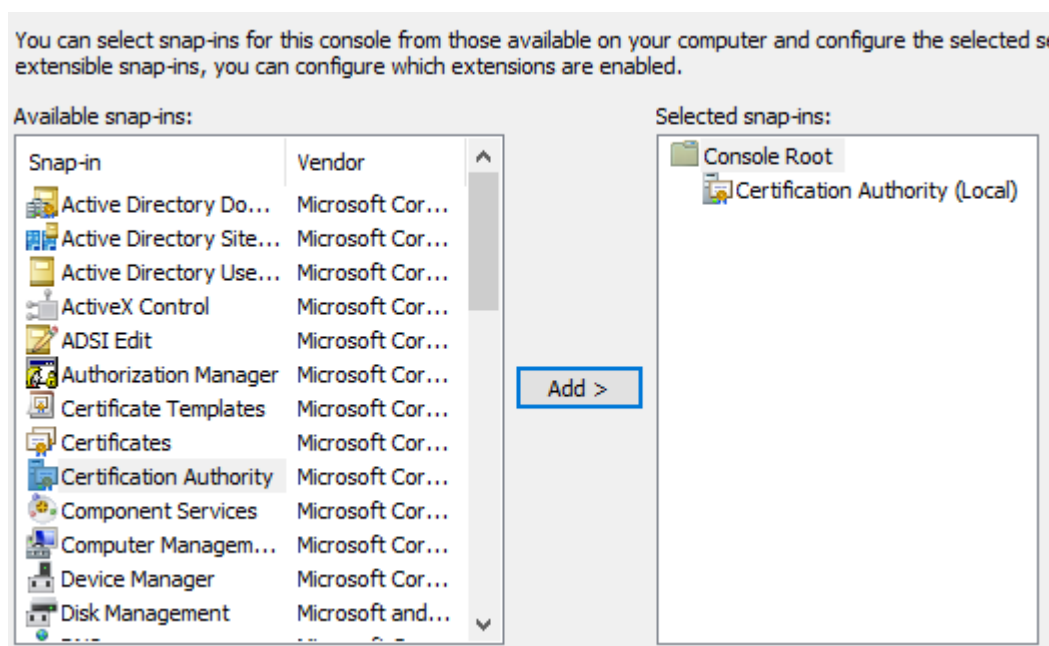
Obrázok 41 Výsledok konfigurácie AD CS

(zdroj: vlastný)

Po úspešnej inštalácii požadovaných rolí Active Directory Certificate Services bolo nevyhnutné prístupť k ich podrobnej konfigurácii. Tento proces zahŕňal nastavenie certifikačnej autority a webovej registrácie, čím sa zabezpečila plná funkčnosť a dostupnosť certifikačných služieb.

Následující část se zaměřuje na konfiguraci certifikační autority s cílem publikovat seznam odvolaných CRL Certificate Revocation List a certifikát certifikační autority do Active Directory.

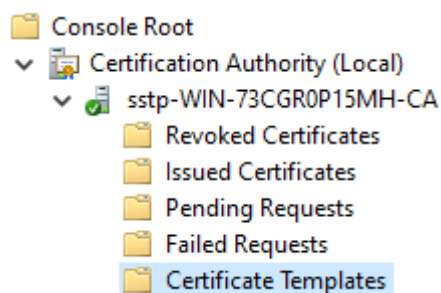
1. **Otvorenie MMC (Microsoft Management Console)** – Pre prístup k nástrojom správy certifikačnej autority bola otvorená konzola MMC. To sa dosiahlo spustením príkazu mmc z okna Spustiť (Run).
2. **Pridanie modulu snap-in certifikačnej autority** – V rámci konzoly MMC bolo z menu File zvolené Add/Remove Snap-in.. Zo zoznamu dostupných modulov snap-in bol vybraný Certification Authority a pridaný do vybranej konzoly kliknutím ADD.
3. **Výber spravovanej certifikačnej autority** – Po pridaní modulu snap-in bolo zvolené spravovať Local computer, aby sa konfigurovala novoinštalovaná certifikačná autorita. A kliklo sa na Finish.



Obrázok 42 Dokončený snap-in

(zdroj: vlastný)

4. **Prístup k vlastnostiam certifikačnej autority:** V strome konzoly MMC sa rozbalila položka Certification Authority (Local). Následne bolo kliknuté pravým tlačidlom myši na Certificate Templates a z kontextového menu bola vybraná položka Manage.



Obrázok 43 Strom MMC

(zdroj: vlastný)

5. **Zoznam šablón** – V zozname šablón bola vybraná šablóna „IPsec“. Kliknutím pravým tlačidlom myši na ňu bola zvolená možnosť Duplicate Template. Otvorilo sa nové okno s vlastnosťami duplikovanej šablóny. V sekcii General bola duplikovaná šablóna premenovaná na „SSTPVPN“. Následne sa prešlo na záložku Request Handling. Tu bola zaškrtnutá možnosť Allow private key to be exported. Ďalej sa pokračovalo na záložku Subject Name. Tu bola zaškrtnutá možnosť Supply in the request. Potom sa prešlo na záložku Extensions. Bola vybraná položka Application Policies a kliklo sa na tlačidlo Edit. V zobrazenom okne bolo kliknuté na Add. V novom okne, ktoré sa otvorilo, bola zaškrtnutá možnosť Server Authentication a kliklo sa na OK. Všetky otvorené okná s vlastnosťami šablóny boli následne pozatvárané potvrdením zmien.
6. **Návrat do konzole MMC** – Stále pod uzlom, bolo kliknuté pravým tlačidlom myši na Certificate Templates a zvolené New následne Certificate Template to Issue. V zobrazenom okne bola vyhl'adaná a vybraná práve vytvorená šablóna „SSTPVPN“ a jej vydanie bolo potvrdené kliknutím na „OK“.

Po úspešnej konfigurácii certifikačnej authority a šablóny certifikátu bolo ďalším krokom nastavenie prostredia Active Directory pre integráciu s certifikačnými službami, vrátane vytvorenia špecifického používateľského účtu a konfigurácie jeho oprávnení.

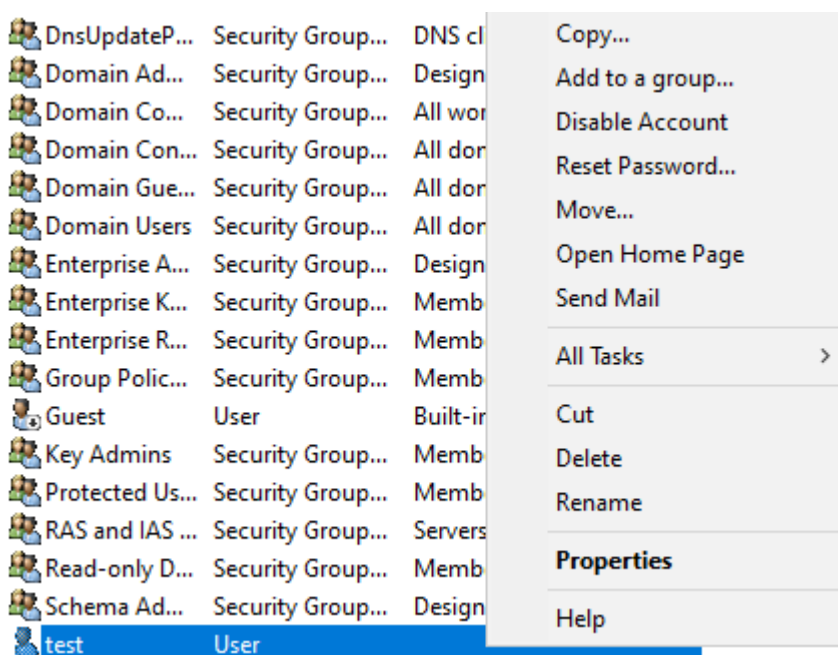
1. **Otvorenie ADUC (Active Directory Users and Computers) a vytvorenie nového používateľa** – Pre správu používateľov a počítačov v doméne bola otvorená konzola Active Directory Users and Computers. Následne, v rámci zložky Users, bolo kliknutím pravým tlačidlom myši zvolená možnosť New a následne User. V zobrazenom dialógovom okne boli vyplnené nasledujúce informácie pre nového používateľa:

- **Meno a priezvisko** – „test“.
- **Prihlasovacie meno používateľa (User logon name)** – „test“ Po zadaní týchto údajov bolo kliknuté na „Ďalej“ (Next).

V ďalšom kroku sa zadalo heslo pre nového používateľa. Bola zaškrtnutá možnosť Password never expires, aby sa zabezpečila kontinuita pri testovaní. Následne bolo kliknuté na Next a proces bol dokončený kliknutím na Finish.

- Nastavenie oprávnení pre používateľa** – Po vytvorení používateľského účtu „test“ bolo nevyhnutné nastaviť špecifické oprávnenia, ktoré mu umožňovali prihlásenie prostredníctvom certifikátu.

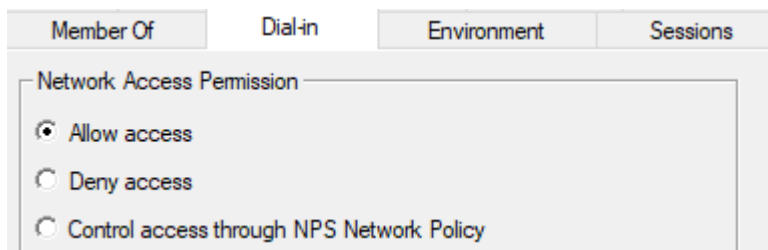
- Kliklo sa pravým tlačidlom myši na novovytvoreného používateľa „test“ a zvolili sa „ Properties“.



Obrázok 44 Vlastnosti užívateľa

(zdroj: vlastný)

- V okne vlastností bola prechádzaná záložka Dial-in. Na tejto záložke bola vybraná možnosť Allow access, čím sa používateľovi udelilo oprávnenie pre prístup cez dial-up alebo VPN, ktoré je často spojené s overovaním prostredníctvom certifikátov (Obr.58).

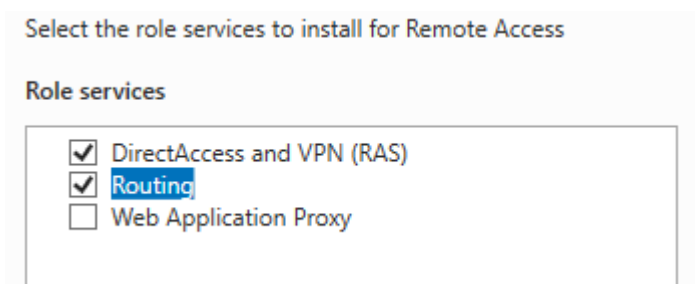


Obrázok 45 Allow access Dial-in

(zdroj: vlastný)

Po dokončení predchádzajúcich častiach konfigurácie bolo potrebné nainštalovať rolu Remote Access, ktorá je kľúčová pre implementáciu VPN pripojení, ako je SSTP VPN.

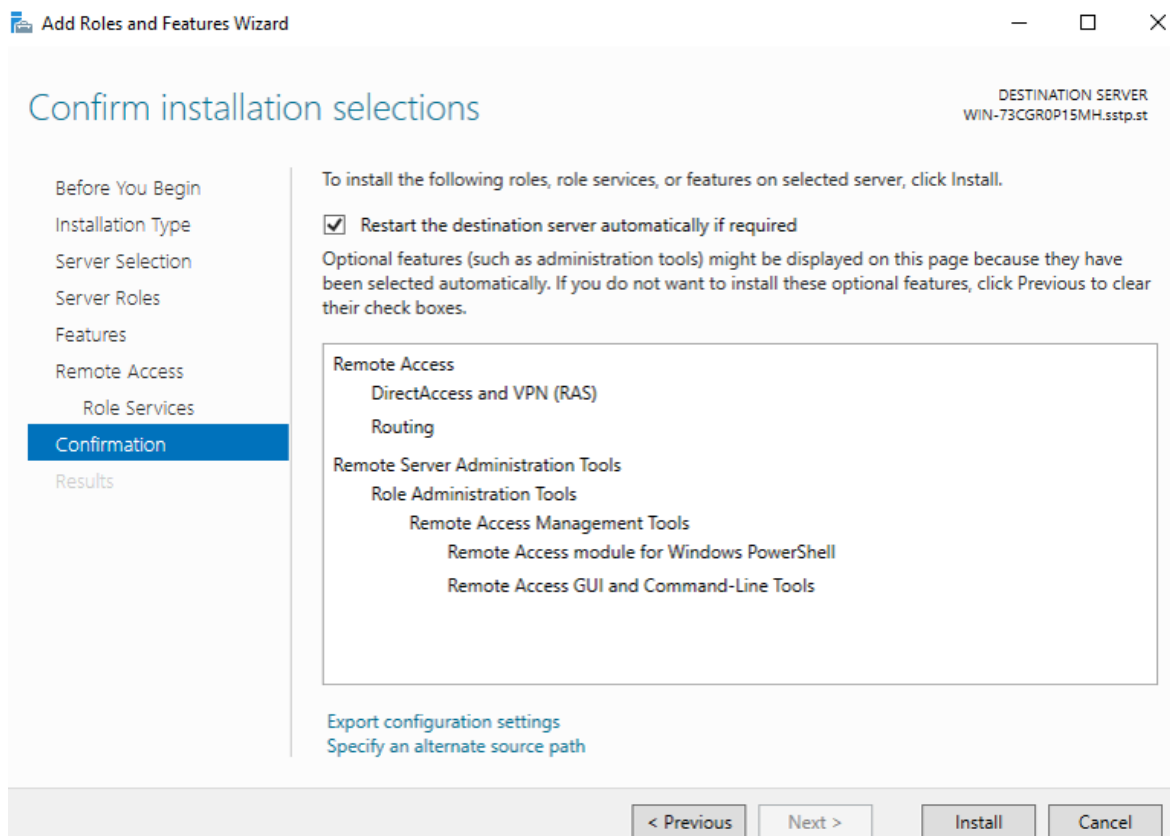
1. **Spustenie Sprievodcu pridaním rolí a funkcií** – Proces bol opäť zahájený otvorením Správcu servera a zvolením Manage -> Add Roles and Features.
2. **Výber typu inštalácie a cieľového servera** – Na stránke Before You Beginsa prešlo ďalej, následne bol zvolený typ inštalácie Role-based or feature-based installation a bol vybraný cieľový server.
3. **Výber role servera** – V sekcii Server Roles bola zaškrtnutá možnosť Remote Access.
4. **Výber rolových služieb** – V rámci roly „Vzdialený prístup“ boli v sekcii Role services zaškrtnuté nasledujúce možnosti:
 - DirectAccess a VPN (RAS)
 - Routing po výbere týchto služieb bolo kliknuté na „Ďalej“.



Obrázok 46 Role services Remote Access

(zdroj: vlastný)

- 5. Potvrdenie a inštalácia** – Na stránke Confirmation bola zaškrtnutá možnosť Restart the destination server automatically if required. Následne bolo kliknuté na Install, čím sa spustila inštalácia vybraných rolí a funkcií.



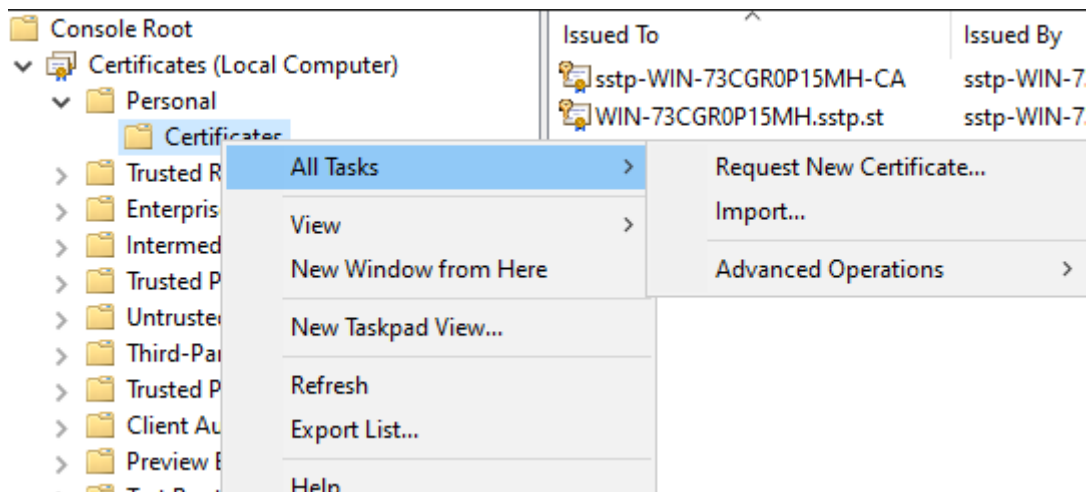
Obrázok 47 Potvrdenie a inštalácia Remote Access

(zdroj: vlastný)

Po inštalácii roly Remote Access bolo nutné získať certifikát pre server, ktorý bude použitý pre zabezpečené pripojenia, ako je SSTP VPN.

- 1. Otvorenie Microsoft Management Console (MMC)** – Pre prístup k správe certifikátov bola otvorená konzola MMC spustením príkazu mmc.
- 2. Pridanie modulu Certificates Snap-in** – V konzole MMC bolo z menu zvolené Add or Remove Snap-ins... V zozname dostupných modulov snap-in bol vybraný Certificates a kliklo sa na Add.
- 3. Výber účtu pre správu certifikátov** – V zobrazenom okne bolo zvolené Computer account a kliklo sa na Next. Potom bola vybraná možnosť Local computer a proces bol dokončený kliknutím na Finish.

4. **Žiadosť o nový certifikát** – V strome konzoly MMC sa rozbalila položka Certificates a následne Personal. Kliknutím pravým tlačidlom myši na Certificates bolo zvolené All Tasks -> Request New Certificate...



Obrázok 48 Žiadosť o nový certifikát

(zdroj: vlastný)

5. **Sprievodca žiadosťou o certifikát** – Otvorilo sa okno Select Certificate Enrollment Policy, kde sa pokračovalo kliknutím na Next. Následne sa zobrazilo okno Request Certificates. Tu sa objavila informácia More information is required to enroll for this certificate. Bolo na ňu kliknuté. Otvorilo sa ďalšie okno, v ktorom bol pre typ vybraný Common name a do poľa Value bolo zadané „sstp.server“. Táto hodnota bola pridaná kliknutím na Add a okno bolo zatvorené tlačidlom Apply (Obr. 49).
6. **Po návrate do okna** – Request Certificates bola zaškrtnutá šablóna „SSTPVPN“, ktorú sme predtým vytvorili, a kliklo sa na Enroll.
7. **Výsledok inštalácie certifikátu** – V ďalšom okne Certificate Installation Results bol zobrazený stav Succeeded. Proces bol dokončený kliknutím na Finish (Obr. 64).

the user or computer that is receiving the certificate.

Subject name:

Type:
Common name ▼

Add >

Value:

< Remove

CN=sstp.server

Alternative name:

Type:
Directory name ▼

Add >

Value:

< Remove

OK Cancel Apply

Obrázok 49 Pridaná hodnota sstp.server

(zdroj: vlastný)

Certificate Installation Results

The following certificates have been enrolled and installed on this computer.

Active Directory Enrollment Policy

☒ SSTPVPN ✓ **STATUS: Succeeded** [Details ^](#)

The following options describe the uses and validity period that apply to this type of certificate:

Key usage: Digital signature
Key encipherment

Application policies: Server Authentication
IP security IKE intermediate

Validity period (days): 730

[View Certificate](#)

Finish

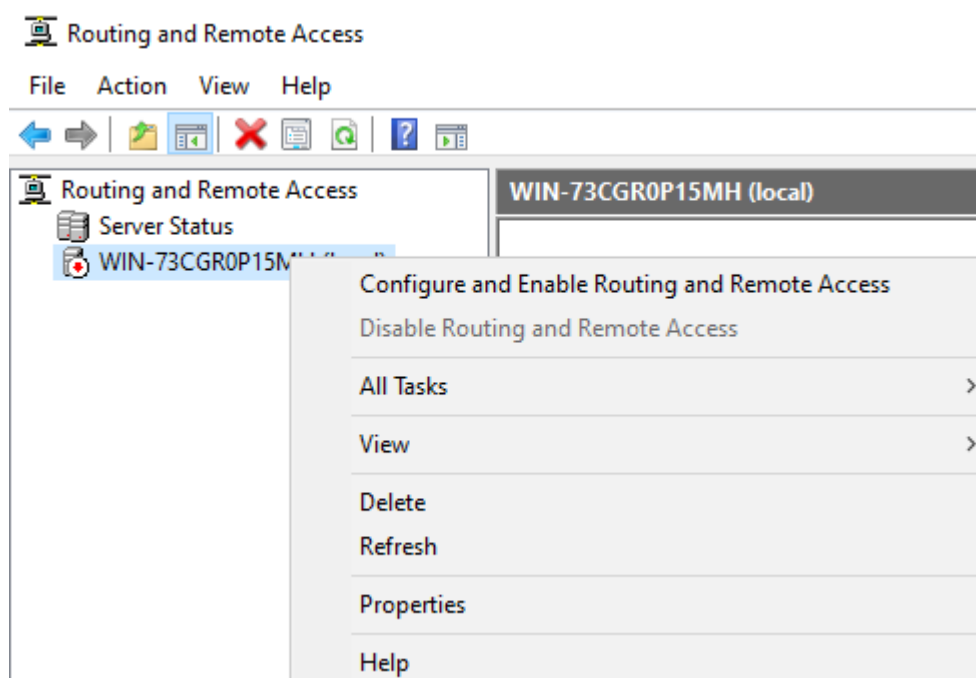
Obrázok 50 Výsledok inštalácie certifikátu

(zdroj: vlastný)

Týmto krokom bol server vybavený potrebným certifikátom pre zabezpečení komunikáciu, čo je kľúčové pre nasadenie SSTP VPN.

Po úspešnom získaní certifikátu pre server bolo možné pristúpiť k inicializácii a základnej konfigurácii služby Routing and Remote Access, ktorá bola predtým nainštalovaná.

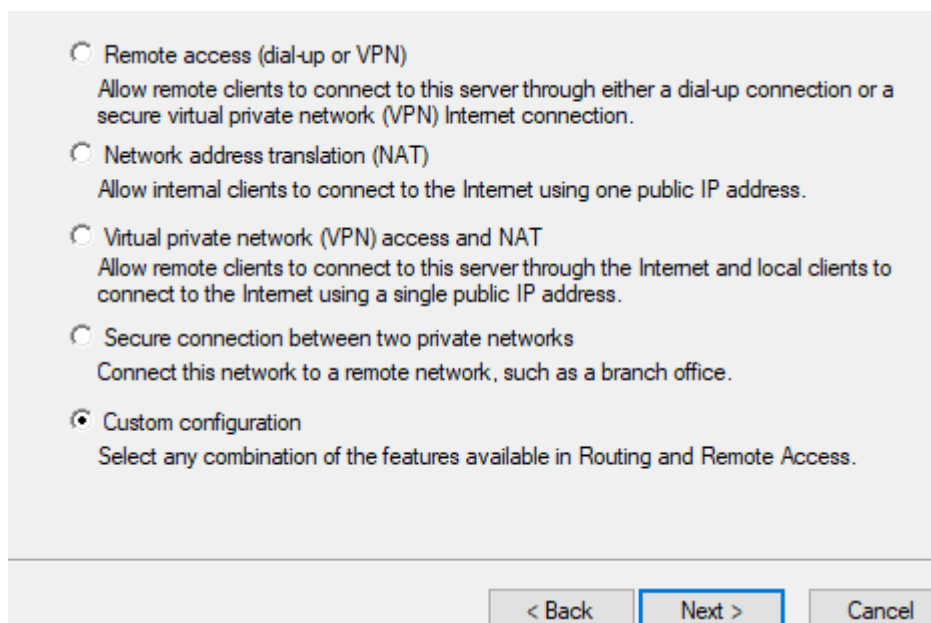
1. **Otvorenie konzoly Routing and Remote Access** – Bola otvorená konzola Routing and Remote Access. V tejto konzole bol viditeľný názov servera, na ktorom bola služba nainštalovaná.
2. **Konfigurácia a povolenie Routing and Remote Access** – Kliklo sa pravým tlačidlom myši na názov servera a z kontextového menu bola vybraná Configure and Enable Routing and Remote Access (Obr. 51).



Obrázok 51 Výber konfigurácie Routing a access

(zdroj: vlastný)

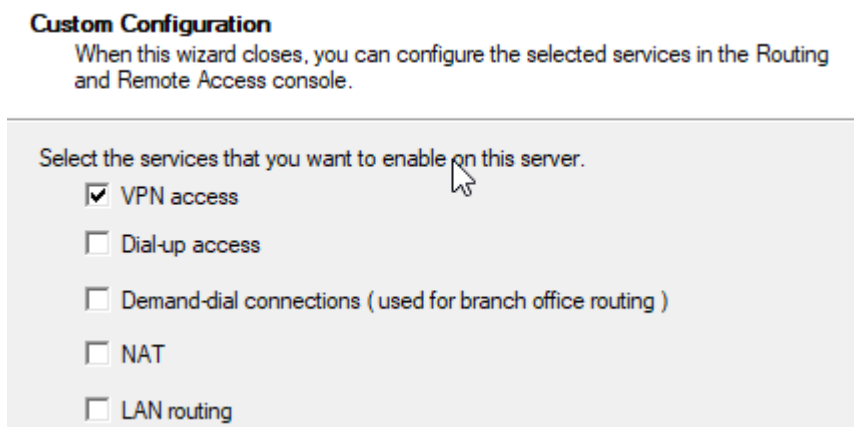
3. **Sprievodca konfiguráciou** – Bol spustený sprievodca konfiguráciou. Prešlo sa úvodnými obrazovkami, až kým sa nedosiahlo okno Configuration.
4. **Výber vlastnej konfigurácie** – V okne konfigurácie bola zaškrtnutá možnosť Custom configuration. Následne bolo kliknuté na Next (Obr. 52).



Obrázok 52 Výber vlastnej konfigurácie

(zdroj: vlastný)

- 5. Výber služieb vlastnej konfigurácie** – V okne ďalšom okne bola zaškrtnutá možnosť VPN access. Kliklo sa na Next.



Obrázok 53 Výber služby vlastnej konfigurácie

(zdroj: vlastný)

- 6. Dokončenie konfigurácie** – V ďalšom okne sa kliklo na Finish, čím sa aplikovali zvolené nastavenia a spustil sa proces inicializácie služby.

Po inicializácii a nastavení základnej služby Routing and Remote Access nám ostávalo už len zmeniť základné nastavenia serveru.

1. **Otvorenie konzoly Routing and Remote Access** – Bola otvorená konzola Routing and Remote Access. V tejto konzole bol viditeľný názov servera, na ktorom bola služba nainštalovaná.
2. **Konfigurácia a povolenie Routing and Remote Access** – Kliklo sa pravým tlačidlom myši na názov servera a z kontextového menu bola vybraná možnosť Configure and Enable Routing and Remote Access.
3. **Sprievodca konfiguráciou** – Bol spustený sprievodca konfiguráciou. Prešlo sa úvodnými obrazovkami, až kým sa nedosiahlo okno Configuration.
4. **Výber vlastnej konfigurácie** – V okne konfigurácie bola zaškrtnutá možnosť „Custom configuration“. Následne bolo kliknuté na Next.
5. **Výber služieb vlastnej konfigurácie** – V okne „Custom configuration“ bola zaškrtnutá možnosť VPN access. Kliklo sa na Next.
6. **Dokončenie konfigurácie** – V ďalšom okne sa kliklo na Finish, čím sa aplikovali zvolené nastavenia a spustil sa proces inicializácie služby. Po dokončení týchto nastavení sa server spustil.
7. **Zmena certifikátu SSL pre Routing and Remote Access** – Po spustení servera bolo opäť kliknuté pravým tlačidlom myši na názov servera v konzole Routing and Remote Access a zvolili sa Properties. V zobrazenom okne sa prešlo do záložky Security. V časti SSL Certificate Binding bol certifikát zmenený na predtým získaný certifikát „sstp.server“.
8. **Nastavenie IPv4 a rozsahu IP adries** – Následne sa v okne „Properties“ prešlo do záložky „IPv4“. Bola zaškrtnutá možnosť Enable IPv4 Forwarding, čím sa povolilo smerovanie IPv4 prevádzky. Potom bol nastavený rozsah statických IP adries, ktoré bude táto VPN služba prideľovať pripojeným klientom. Všetky zmeny boli potvrdené kliknutím na tlačidlá OK alebo Apply.

Pre zabezpečenie riadenia prístupu a sieťových politík bol posledným krokom inštalácia roly Network Policy and Access Services.

1. **Pridanie roly Network Policy and Access Services** – Proces inštalácie bol zahájený prostredníctvom Server Manager -> Manage -> Add Roles and Features. V sprievodcovi sa pokračovalo rovnako ako pri predchádzajúcich konfiguráciách rolí,

až kým sa v časti „Server Roles“ nevybrala možnosť „Network Policy and Access Services“.

Po dokončení všetkých týchto častí bola týmto krokom úspešne ukončená konfigurácia SSTP serveru.

4.7.2 Inštalácia a príprava klientskej strany

Pre konfiguráciu a správu VPN pripojenia bol na klientovi nainštalovaný NetworkManager-sstp. Tento balík pridáva podporu pre protokol do NetworkManagera, čím umožňuje jednoduchú konfiguráciu VPN pripojení na inštaláciu tohto balíka boli použité.

```
sudo apt update  
sudo apt install network-manager-sstp network-manager-sstp-gnome
```

Pre zabezpečenie dôveryhodnosti komunikácie bolo nevyhnutné získať a nainštalovať certifikát certifikačnej autority servera. Bol otvorený webový prehliadač a do adresného riadku bola zadaná URL (Uniform Resource Locator) adresa webovej služby certifikačnej autority: <http://192.168.0.109/certserv>. Zobrazila sa výzva na zadanie prihlasovacích údajov. Boli použité poverenia užívateľa, ktoré boli vytvorené na doménovom radiči servera. Po úspešnom prihlásení na webovej stránke bolo kliknuté na možnosť Request a certificate a následne na Download CA certificate. Certifikát s názvom certnew.cer sa stiahol do domovského adresára klienta.

Certifikát stiahnutý z webovej služby bol vo formáte DER (Distinguished Encoding Rules), binárnom formáte. Pre správne spracovanie a použitie v prostredí Linuxu, najmä pre nástroje OpenSSL a systémové úložiská certifikátov, bol potrebný formát PEM (Privacy-Enhanced Mail), ktorý je textový a base64-enkódovaný. Bol otvorený terminál na Ubuntu klientovi a bol vykonaný nasledujúci príkaz pre konverziu:

```
openssl x509 -inform DER -in ~/certnew.cer -out ~/certnew.crt
```

Pre kontrolu správnej konverzie a čitateľnosti certifikátu v textovom formáte bol použitý príkaz:

```
cat ~/certnew.crt
```

Výstup obsahoval čitateľné riadky s hlavičkami -----BEGIN CERTIFICATE----- a -----END CERTIFICATE----- s base64 kódovanými dátami certifikátu medzi nimi, čo potvrdilo úspešnú konverziu (Obr. 54).

```
sstpklient@ubuntu:~$ cat certnew.crt
-----BEGIN CERTIFICATE-----
MIIDCzCCAAlgAwIBAgIQNAkvVythx5VFjiW53U3X6zANBgkqhkiG9w0BAQsFADBM
MRIwEAYKCZImiZPyLGGQBgRYCc3QxFDA5BgoJkiaJk/IsZAEZFgRzc3RwMSAwHgYD
VQQDExdzc3RwLVdJTj03M0NHUjBQMTVNSC1DQTAeFw0yNTA1MjYxOTA2MDhaFw0z
MDA1MjYxOTE2MDhaMEwxEjAQBgoJkiaJk/IsZAEZFgRzc3RwMSAwHgYD
ARkWBHNzdHxIDAeBgNVBAMTF3NzdHAtV0l0LTczQ0dSMFAxNU1ILUNBMiIBIjAN
BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAt7kLT+S1uWl7nXFVj/GY0y3bMk/Y
0R9srhi9D+4aYU8/ApmCJJhp9AxiqMRDtdqJotp6lK26Uch2e34TYtVq+hTHGsM0
6pW4y4xSTHrVmf5m9CdNqyl5ByZIAz+5q/OUZ5uRvWZsLaZQyRe+a1BsZBEwtw/v
AGJy7o6MjtNRTjves97pvkbMXQEIQj6UXr90CApF/mi4lInGeoYXSt6t/moP2u9I
0vxXD2VTsb7ly0zV2754Bro/tmV8oro4XIdj17hE7uGTVQXdxTilzlvTpNZHWfXSs
uwSLBtutV18erIKOGKTu6VqMrmsWX41TumiANqCECW9iqpxJEqDYqa2nQIDAQAB
o1EwTzALBgNVHQ8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUuFvT
d0+NNjciV6zJ7ESBRmRrswEAYJKwYBBAGCNxUBBAMCAQAwDQYJKoZIhvcNAQEL
BQADggEBAI1VCwTfH5CDHIRRoAU3Tn1SwQPos9EHuETQLitwavFdc+5nnC/gRwIC
bGRHKR5ceYiYQtmcBCLKem9xqKoUaXgdwS8g5ghgYXZBgIn4JTeTyi2uwS0awut8
ZBfTcnmeTz7fzQM9Ag8A2a0oD8j9Q3pwd00DFg/CC2yJBocR59hyF530vU+3H0ua
gG57t/rG/MMMTAQpTRf1yduZrUnprDH+91wLxEa0eW1K8SjXckVorun4irRG57Fi
+4fSQbKOUTi1lbuqof4QGzq/zHpPdYznByBD00gEnA/22qHLWHN5j9VxCpfy9I40
Sow0Zo28YX0x4ltXiHGT1EBu90I2ebs=
-----END CERTIFICATE-----
```

Obrázok 54 Výstup cat certnew.crt

(zdroj: vlastný)

Aby operačný systém a aplikácie dôverovali certifikátu certifikačnej autority, bolo potrebné ho umiestniť do systémového úložiska dôveryhodných certifikátov. Skonvertovaný .crt súbor bol skopírovaný do štandardného adresára pre vlastné CA certifikáty v Ubuntu pomocou príkazu:

```
sudo cp ~/certnew.crt /usr/local/share/ca-certificates/certnew.crt
```

Nakoniec bolo aktualizované systémové úložisko certifikátov, aby sa nový certifikát zaregistroval a stal sa dôveryhodným pre systém a aplikácie:

```
sudo update-ca-certificates
```

Tento príkaz prečítal certifikáty z adresára /usr/local/share/ca-certificates/ a vytvoril potrebné symbolické odkazy v /etc/ssl/certs/, čím ich sprístupnil pre systémové knižnice a aplikácie využívajúce systémové dôveryhodné certifikáty.

Po úspešnom získaní a inštalácii CA certifikátu sa pristúpilo ku konfigurácii samotného VPN pripojenia v grafickom rozhraní Ubuntu.

1. **Otvorenie nastavení siete** – Otvorili sa systémové Settings a následne sa prešlo do sekcie Network.
2. **Pridanie nového VPN pripojenia** – V časti VPN sa kliklo na tlačidlo + pre pridanie nového pripojenia.

3. Výber typu VPN – Z vyskakovacieho okna sa vybrala možnosť Secure Socket Tunneling Protocol (SSTP). Tým sa otvorilo nové konfiguračné okno pre SSTP VPN.

4. Zadanie detailov pripojenia:

- Do poľa "Gateway" bola zadaná IP adresa VPN.
- V sekcii pre autentifikáciu boli zadané prihlasovacie údaje, ktoré boli vytvorené na serveri pre prístup k VPN.

5. Konfigurácia SSL Tunnelu – Následne sa kliklo na tlačidlo Advanced. V sekcii SSL Tunnel bol nájdený a vybraný predtým stiahnutý a prekonvertovaný CA certifikát (certnew.crt), ktorý sa nachádzal v systémovom úložisku certifikátov.

6. Uloženie a spustenie VPN – Po skontrolovaní všetkých nastavení sa kliklo na Apply pre uloženie konfigurácie. Následne bolo VPN pripojenie spustené prepnutím príslušného prepínača v sekcii Network vedľa názvu novovytvoreného VPN pripojenia. Tým sa nadviazalo šifrované spojenie s VPN serverom.

Pre kontrolu úspešného nadviazania VPN spojenia sa na serveri zadal príkaz.

ip a

Vo výstupe tohto príkazu bolo vidieť novovytvorené VPN rozhranie a pridelenú IP adresu, čo potvrdzovalo aktívny VPN tunel.

4.8 Testovanie a analýza priepustnosti VPN protokolov

Táto kapitola sa zameriava na praktické testovanie priepustnosti jednotlivých VPN protokolov a analýzu získaných dát s cieľom vyhodnotiť ich vplyv na celkový výkon siete.

4.8.1 Metodika testovania

Cieľom testovania bolo zmerať datovú priepustnosť rôznych VPN protokolov v kontrolovanom prostredí a analyzovať vplyv ich konfigurácie na celkový výkon siete.

Testovacie prostredie pozostávalo zo servera a klienta bežiacich na virtuálnych strojoch. Na oboch stranách bol pre meranie priepustnosti použitý nástroj iPerf3. Na každom testovanom serveri bol spustený iPerf3 v režime servera pomocou príkazu.

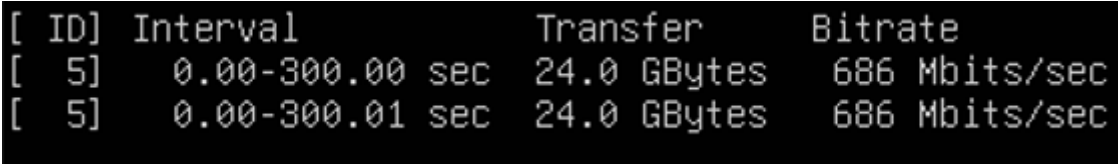
```
Iperf3 -s
```

Následne z klienta boli posielané pakety smerom na server po dobu 300 sekúnd pomocou príkazu.

```
iperf3 -c [IP_adresa_servera] -t 300
```

4.8.2 Výsledky testov priepustnosti

V tejto podkapitole sú prezentované výsledky meraní datovej priepustnosti pre jednotlivé VPN protokoly. Merania boli vykonané pomocou nástroja iPerf3 podľa metodiky popísanej v kapitole 4.8.1. Pre každý protokol je uvedená dosiahnutá priepustnosť.

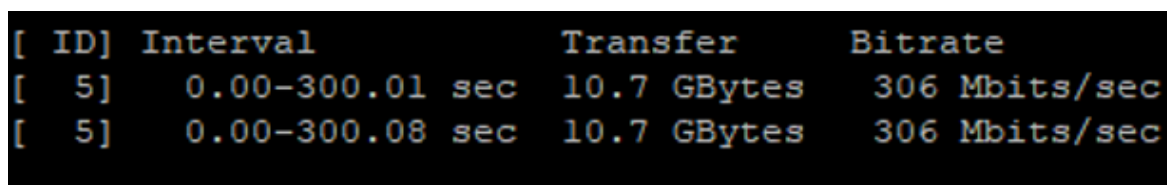


[ID]	Interval	Transfer	Bitrate
[5]	0.00-300.00 sec	24.0 GBytes	686 Mbits/sec
[5]	0.00-300.01 sec	24.0 GBytes	686 Mbits/sec

Obrázok 55 Výsledok merania Wireguard

(zdroj: vlastný)

Ako je vidieť na (Obr.55), pri prenose dát cez WireGuard VPN tunel bola dosiahnutá priepustnosť 686 Mbits/s (Megabits za sekundu) čo zodpovedalo prenosu 24.0 GBytes (GigaBytes) dát za 300 sekúnd.

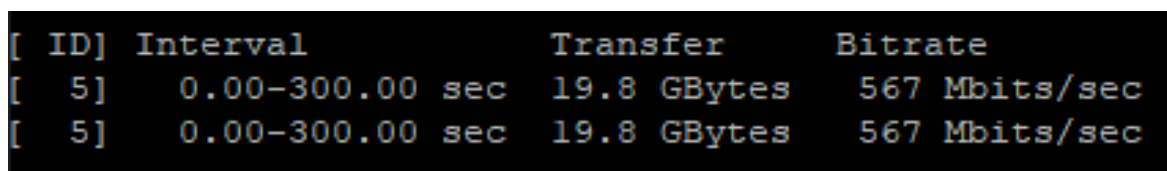


[ID]	Interval	Transfer	Bitrate
[5]	0.00-300.01 sec	10.7 GBytes	306 Mbits/sec
[5]	0.00-300.08 sec	10.7 GBytes	306 Mbits/sec

Obrázok 56 Výsledok merania OpenVPN

(zdroj: vlastný)

Ako je vidieť na (Obr. 56) pre OpenVPN bola nameraná efektívna priepustnosť 306 Mbit/s (Megabit za sekundu), pričom sa prenieslo 10.7 GBytes dát v rámci 300-sekundového testovacieho intervalu.

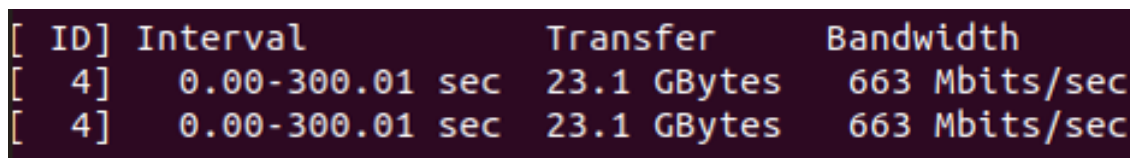


[ID]	Interval	Transfer	Bitrate
[5]	0.00-300.00 sec	19.8 GBytes	567 Mbits/sec
[5]	0.00-300.00 sec	19.8 GBytes	567 Mbits/sec

Obrázok 57 Výsledok merania IKEv2

(zdroj: vlastný)

Pre protokol IKEv2 bolo meranie priepustnosti vizualizované na (Obr. 57). Protokol IKEv2 preukázal robustný výkon s dosiahnutou priepustnosťou 567 Mbit/s. Počas testu sa prenieslo 19.8 GBytes dát v priebehu 300 sekúnd.



[ID]	Interval	Transfer	Bandwidth
[4]	0.00-300.01 sec	23.1 GBytes	663 Mbits/sec
[4]	0.00-300.01 sec	23.1 GBytes	663 Mbits/sec

Obrázok 58 Výsledok merania L2TP

(zdroj: vlastný)

Meranie priepustnosti pre protokol L2TP/IPsec, vykonané v rovnakých podmienkach ako predchádzajúce protokoly, je zobrazené na (Obr 58). Pre tento protokol bola dosiahnutá vysoká priepustnosť 663 Mbit/s, pri prenose 23.1 GBytes dát za stanovených 300 sekúnd.

ID]	Interval	Transfer	Bitrate
5]	0.00-300.00 sec	6.41 GBytes	184 Mbits/sec
5]	0.00-300.00 sec	6.41 GBytes	184 Mbits/sec

Obrázok 59 Výsledok merania SSTP

(zdroj: vlastný)

Posledným testovaným protokolom bola SSTP VPN. Jej výsledok merania priepustnosti je zobrazený na (Obr. 59). Pre SSTP VPN bola nameraná priepustnosť 184 Mbit/s, s prenosom 6.41 Gbytes dát za 300 sekúnd.

Tabuľka 1 Súhrn nameraných priepustností pre jednotlivé VPN protokoly

Protokol	Priepustnosť (Mbits/s)	Velikost' prenesených dát za 300 sekúnd
WireGuard	686	24.0
OpenVPN	306	10.7
IKEv2	567	19.8
L2TP/IPsec	663	23.1
SSTP	184	6.41

(zdroj: vlastný)

Pre uľahčenie prehľadu a priameho porovnania výkonu jednotlivých VPN protokolov sú všetky namerané hodnoty priepustnosti zhrnuté v Tabuľke 1. Táto vizualizácia poskytuje okamžitý prehľad o relatívnom výkone každého protokolu v testovacom prostredí.

4.8.3 Analýza, porovnanie a zhodnotenie výsledkov

V tejto podkapitole je vykonaná podrobná analýza a porovnanie výsledkov meraní datovej priepustnosti jednotlivých VPN protokolov, ktoré boli prezentované v predošlej sekcii. Cieľom je identifikovať faktory, ktoré mohli ovplyvniť dosiahnutý výkon, vrátane použitých kryptografických algoritmov, ich implementácie a špecifik operačného systému, následne vysvetliť zistené rozdiely medzi nameranými hodnotami.

Tabuľka 2 Kryptografické konfigurácie testovaných VPN protokolov

VPN Protokol	Šifrovací algoritmus	Algoritmus autentifikácie	Kľúčová výmena	Transportný protokol
WireGuard	Chacha20-Poly1305	Poly1305	Curve25519	UDP
OpenVPN	AES-256-GCM	GCM implicitne	TLS/DHE	TCP
IKEv2	ChaCha20-Poly1305	SHA512	Curve25519	UDP
L2TP/IPsec	AES-256	SHA256	MODP2048(Diffie-Hellman Group 14)	UDP
SSTP	TLS_AES_256_GCM	SHA384 (TLS 1.3)	TLS Handshake	TCP

(zdroj: vlastný)

Pre objektívnu analýzu výkonu bolo považované za dôležité poznať konkrétne kryptografické algoritmy a funkcie použité pre každý VPN protokol počas testovania. Tieto konfigurácie sú zhrnuté v Tabuľke 2.

Z nameraných výsledkov Tabuľka 1 bol zistený výrazný rozdiel v dosiahnutej priepustnosti medzi jednotlivými VPN protokolmi. Predpokladá sa, že tento rozdiel je ovplyvnený kombináciou ich návrhu, konkrétnej implementácie a použitých kryptografických algoritmov.

- **WireGuard** dosiahol najvyššiu priepustnosť **686 Mbit/s**. Tento výnimočný výkon je priamym dôsledkom jeho návrhových princípov. Protokol WireGuard sa vyznačuje minimalistickým kódom a využívaním moderných kryptografických primitívov, ako je **ChaCha20** pre šifrovanie a **Poly1305** pre autentifikáciu **AEAD (Authenticated Encryption with Associated Data)**. Tieto algoritmy sú navrhnuté s dôrazom na vysoký výkon a efektívnu hardvérovú akceleráciu. Navyše, jeho optimalizácia pre beh priamo v jadre operačného systému (Linux Kernel) minimalizuje latenciu a réžiu spojenú s prepínaním kontextu medzi jadrom a

používateľským priestorom, čo sa priamo prejavilo vo vysokej priepustnosti a nulovom počte retransmisií. Skutočnosť, že nameraná hodnota 1.06 Gbit/s mierne presahuje nominálnu kapacitu 1 Gbit/s ethernetovej linky, je pripisovaná meracím špecifikám nástroja iPerf3, ktorý meria dátovú priepustnosť na aplikačnej vrstve (užitočné dáta) bez započítania režie hlavičiek nižších sieťových vrstiev (napr. Ethernet, IP, UDP hlavičky) a medzipaketových medzier, ktoré tvoria časť celkovej kapacity fyzickej linky. Toto svedčí o extrémnej efektívnosti protokolu WireGuard v maximalizácii využitia dostupnej sieťovej kapacity.

- **L2TP/IPsec** s priepustnosťou **663 Mbit/s** sa umiestnil na druhom mieste, tesne za WireGuardom. Protokol IPsec, ktorý zabezpečuje L2TP tunel, sa javí ako vysoko efektívny, čo je pripisované rozsiahlej podpore hardvérovej akcelerácie pre **AES-256 šifrovanie** a **SHA256 autentifikáciu** (najmä prostredníctvom AES-NI inštrukcií v moderných procesoroch). To mu umožňuje veľmi rýchle spracovanie šifrovaných dát, minimalizujúc zaťaženie CPU a maximalizujúc priepustnosť aj napriek komplexnosti IPsec protokolu. Pre kľúčovú výmenu bola použitá **skupina Diffie-Hellman MODP (Modular Exponentiation) 2048 (Group 14)**, ktorá je považovaná za štandardnú a bezpečnú voľbu pre vytváranie silných zdieľaných tajomstiev.
- **IKEv2** dosiahol priepustnosť **567 Mbit/s**. Tento protokol je súčasťou rodiny IPsec a predpokladá sa, že využíva podobné efektívne kryptografické sady. Konkrétne použitie **ChaCha20-Poly1305 pre šifrovanie** a **SHA512 pre autentifikáciu** (v kombinácii s Curve25519 pre kľúčovú výmenu) naznačuje snahu o moderné a výkonné kryptografické riešenie. Hoci jeho priepustnosť bola o niečo nižšia ako pri L2TP/IPsec a WireGuarde, je to považované za veľmi dobrý výsledok, ktorý potvrdzuje jeho vhodnosť pre vysokovýkonné nasadenia a jeho robustnosť.
- **OpenVPN** s priepustnosťou **306 Mbit/s** sa umiestnil pod IKEv2. Hoci je OpenVPN mimoriadne flexibilný, široko podporovaný a bezpečný (vďaka použitiu **AES-256-GCM** ako šifrovacieho algoritmu, ktorý poskytuje aj autentifikáciu), jeho výkon môže byť často obmedzený v dôsledku jeho architektúry. Napriek použitiu **UDP protokolu** ako transportnej vrstvy, je primárnym faktorom, ktorý obmedzuje jeho priepustnosť, implementácia OpenVPN prevažne v používateľskom priestore (user-space) operačného systému. To znamená, že sieťové pakety musia byť neustále kopírované medzi jadrom (kde je sieťová karta) a používateľským priestorom (kde

beží OpenVPN proces pre šifrovanie a dešifrovanie). Toto neustále prepínanie kontextu medzi jadrom a používateľskou aplikáciou generuje dodatočnú réžiu procesora, ktorá limituje celkovú priepustnosť pri vysokých rýchlostiach, aj keď samotné kryptografické operácie (AES-256-GCM) sú hardvérovo akcelerované. Zložitosť protokolu a jeho rozsiahle funkcie môžu tiež prispieť k vyššej réžii v porovnaní s minimalistickými protokolmi.

- **SSTP** dosiahol najnižšiu priepustnosť **184 Mbit/s**. Tento protokol je založený na **TCP protokole** (konkrétne na porte 443, rovnako ako HTTPS) a využíva šifrovacie sady TLS/SSL, v našom prípade **TLS_AES_256_GCM_SHA384**. Hoci toto uľahčuje prechádzanie cez firewally, jeho výkon je ovplyvnený inherentnou réžiou tunelovania TCP cez TCP, ako aj komplexnosťou vrstvy SSL/TLS. Táto architektúra protokolu môže pridávať dodatočnú výpočtovú a prenosovú réžiu, ktorá prirodzene znižuje dosiahnuteľnú priepustnosť. Dôležitým faktorom v tomto testovacom prostredí je aj skutočnosť, že **SSTP server bol implementovaný na operačnom systéme Windows Server**, zatiaľ čo ostatné testované protokoly (WireGuard, OpenVPN, IKEv2, L2TP/IPsec) boli v tomto testovacom prostredí umiestnené na Linuxovom serveri. Rozdiely v optimalizácii TCP/IP stacku, sieťových ovládačov a celkovej architektúre VPN služieb medzi týmito dvoma operačnými systémami mohli taktiež prispieť k nižšej dosiahnutej priepustnosti pre SSTP.

Okrem samotnej priepustnosti bolo považované za dôležité zvážiť aj širší vplyv konfigurácie a zvoleného protokolu na celkový výkon siete.

- **Vplyv šifrovacích algoritmov a réžie:** Ako ukázali výsledky, použitie moderných a efektívnych AEAD (Authenticated Encryption with Associated Data) algoritmov ako **ChaCha20-Poly1305** a **AES-GCM** je kľúčové pre vysoký výkon, pretože minimalizujú réžiu kryptografických operácií. Hardvérová akcelerácia týchto algoritmov má zásadný význam pre protokoly ako L2TP/IPsec a IKEv2.
- **Vplyv transportného protokolu (UDP vs. TCP):** Rozdiely vo výkone medzi protokolmi používajúcimi **UDP** (WireGuard, IKEv2, L2TP/IPsec) a protokolmi používajúcimi **TCP** (SSTP, OpenVPN) sú zrejmé. UDP, ako bezspojový protokol, vo všeobecnosti ponúka potenciál pre vyššiu priepustnosť pre VPN, pretože nemá vlastné mechanizmy pre kontrolu toku a retransmisie. To znamená, že sa vyhýba réžii spôsobenej vnáraním jednej vrstvy TCP do druhej. Naopak, TCP zabezpečuje spoľahlivé doručenie, ale jeho mechanizmy (ako riadenie toku a kontrola zahltenia)

môžu byť pri tunelovaní iného TCP spojenia cez neho neefektívne a viesť k výraznému poklesu výkonu.

- **Optimalizácia implementácie a operačný systém:** Optimalizácia kódu a umiestnenie protokolu (v jadre vs. v používateľskom priestore) majú zásadný vplyv na výkon. WireGuard v tomto smere profituje zo svojej in-kernel implementácie a minimalistického dizajnu, čo mu umožňuje dosahovať takmer natívnu sieťovú rýchlosť. OpenVPN, napriek svojej flexibilitě, je obmedzený režou spojenou s behom v používateľskom priestore. Navyše, ako bolo pozorované u SSTP, **rozdielna platforma servera (Windows Server oproti Linuxu pre ostatné protokoly) môže ovplyvniť celkový výkon.** Výkonnosť sieťových služieb na rôznych operačných systémoch sa líši v závislosti od ich architektúry, implementácie TCP/IP stacku, ovládačov a optimalizácií.

Na základe vykonaných testov a detailnej analýzy je možné konštatovať, že:

- **WireGuard** sa javí ako najefektívnejší a najrýchlejší VPN protokol pre dosahovanie maximálnej priepustnosti v testovacom prostredí, a to predovšetkým vďaka jeho modernému kryptografickému dizajnu a in-kernel implementácii. Je ideálny pre scenáre, kde je rýchlosť kritická.
- **L2TP/IPsec a IKEv2** sú považované za veľmi silné kandidáty pre aplikácie vyžadujúce vysokú priepustnosť a robustnú bezpečnosť. Ich výkon je silne závislý od dostupnosti hardvérovej akcelerácie pre IPsec operácie.
- **OpenVPN** zostáva flexibilnou a bezpečnou voľbou, avšak je potrebné zvážiť potenciálne obmedzenia priepustnosti, ktoré vyplývajú predovšetkým z režie spojenej s jeho implementáciou v používateľskom priestore operačného systému. Táto architektonická voľba môže spôsobovať vyššiu spotrebu CPU a nižšiu efektívnu priepustnosť v porovnaní s jadrovými protokolmi.
- **SSTP** sa ukázal ako najpomalší z testovaných protokolov, primárne kvôli jeho závislosti na TCP protokole a s tým spojenou inherentnou vrstvenou réžiou. Jeho výkon mohol byť tiež ovplyvnený špecifikami implementácie na platforme Windows Server. Jeho hlavnou výhodou zostáva schopnosť prechádzať cez väčšinu firewallov vďaka použitiu štandardného HTTPS portu.

Táto analýza poskytuje cenné poznatky pre výber vhodného VPN protokolu v závislosti od konkrétnych požiadaviek na priepustnosť, bezpečnosť, toleranciu k režii a sieťové podmienky, vrátane zváženia operačného systému servera.

ZÁVĚR

Táto práca analyzovala výkonnostné charakteristiky vybraných virtuálnych súkromných sietí, ktoré sú kľúčové pre zabezpečenie komunikácie. Cieľom bolo experimentálne porovnať ich priepustnosť a efektivitu. Očakávalo sa, že práca poskytne IT odborníkovi praktické odporúčania pre nasadenie VPN s dôrazom na priepustnosť a bezpečnosť.

Boli úspešne nakonfigurované a testované protokoly OpenVPN, IKEv2, L2TP/IPsec, SSTP a WireGuard. Merania priepustnosti odhalili výrazné rozdiely. WireGuard dosiahol najvyšší výkon vďaka in-kernel implementácii a minimalistickému dizajnu. L2TP/IPsec a IKEv2 vykázali taktiež vysokú priepustnosť, pripísanú efektívnej hardvérovej akcelerácii IPsec. OpenVPN dosiahol strednú priepustnosť, obmedzenú implementáciou v používateľskom priestore. Najnižšiu priepustnosť vykazoval SSTP, čo je spojené s réžiou TCP tunelovania a špecifikami Windows Server implementácie.

Na základe výsledkov bol vyhodnotený vplyv konfigurácií a protokolov na celkový výkon. Bola zdôraznená dôležitosť hardvérovej akcelerácie a vplyv architektúry protokolu (jadro vs. používateľský priestor). Potvrdila sa výhoda UDP pre VPN z hľadiska minimalizácie režie.

Získané poznatky poskytujú cenné informácie pre výber VPN protokolov, poukazujúc na potenciál WireGuardu pre vysokovýkonné nasadenia. Testovanie bolo vykonané v špecifickom virtuálnom prostredí, a výsledky sa preto môžu líšiť.

Pre ďalší výskum by bolo prínosné rozšíriť merania o vplyv sieťovej záťaže, latencie a paketovej straty, ako aj porovnať spotrebu systémových prostriedkov a vplyv implementácií na výkon v rôznych OS.

SEZNAM POUŽITÉ LITERATURY

- [1] FERGUSON, Paul a Geoff HUSTON. *What is a VPN?* Online. Cisco, Telstra. Apríl 1998, Revízia 1. Dostupné z: <http://sol.te.net.ua/www/nanog/vpn.pdf>. [cit. 2025-03-06].
- [2] CRIST, Eric F. a Jan Just KEIJSER. *Mastering OpenVPN*. Birmingham: Packt Publishing, 2015. 288 s. ISBN 978-1-78355-313-6.
- [3] THE KNOWLEDGE ACADEMY. *Top 12 Disadvantages of VPN That You Should Know About*. Online. 2025. The Knowledge Academy. Dostupné z: <https://www.theknowledgeacademy.com/blog/disadvantages-of-vpn/>. [cit. 2025-03-07].
- [4] VPN.com. *Best VPNs For 443 TCP Port: Secure Your Connection With Ease*. Online. VPN.com. Dostupné z: <https://www.vpn.com/protocol/tcp-on-port-443/>. [cit. 2025-03-07].
- [5] Cloudflare. *What happens in a TLS handshake?* Online. Cloudflare. Dostupné z: <https://www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake/>. [cit. 2025-03-07].
- [6] *Srovnání VPN Protokolů: PPTP vs. L2TP vs. OpenVPN vs. SSTP vs. IKEv2*. Online. VPNMentor. Dostupné z: <https://cs.vpnmentor.com/blog/srovnani-vpn-protokolu-pptp-vs-l2tp-vs-openvpn-vs-sstp-vs-ikev2/>. [cit. 2024-10-04].
- [7] KAUFMAN, C. *RFC 4306 – Internet Key Exchange (IKEv2) Protocol*. Microsoft, 2005. Dostupné z: <https://tools.ietf.org/html/rfc4306>. [cit. 2024-10-04].
- [8] DATAHACKER. *Demystifying IKE/IKEv2*. Online. Datahacker. Dostupné z: <https://datahacker.blog/industry/technology-menu/information-security/demystifying-ike-ikev2#2-phase>. [cit. 2025-04-14].
- [9] PALO ALTO NETWORKS. *What is IKEv2?* Online. Palo Alto Networks. Dostupné z: <https://www.paloaltonetworks.com/cyberpedia/what-is-ikev2>. [cit. 2025-04-14].
- [10] HAMZEH, J., PALL, G., VERTHEIN, W., TAARUD, J. a LITTLE, W. *Layer Two Tunneling Protocol "L2TP"*. Online. RFC 2661. Internet Engineering Task Force. August 1999. Dostupné z: <https://datatracker.ietf.org/doc/html/rfc2661>. [cit. 2025-04-14].
- [11] CISCO. *LAC and LNS*. Online. Cisco. Január 2022. Dostupné z: https://www.cisco.com/c/en/us/td/docs/routers/cnBNG/cnBNG-CP/2022-01-x/Config-Guide/b_cnbnng_cp_config_guide-2022-01/m_lac_and_lns.pdf. [cit. 2025-04-14].

- [12] HUAWEI. *Basic Concepts of L2TP*. Online. Huawei. Dostupné z: <https://support.huawei.com/enterprise/en/doc/EDOC1100212572/1d8182dc/basic-concepts-of-l2tp>. [cit. 2025-04-14].
- [13] PALO ALTO NETWORKS. *What Is L2TP (Layer 2 Tunnel Protocol)?* Online. Palo Alto Networks. Dostupné z: <https://www.paloaltonetworks.com/cyberpedia/what-is-l2tp>. [cit. 2025-04-14].
- [14] PALO ALTO NETWORKS. *What is SSTP?* Online. Palo Alto Networks. Dostupné z: <https://www.paloaltonetworks.com/cyberpedia/what-is-sstp>. [cit. 2025-04-16].
- [15] SHAH, M.H. a A. MAHMOOD. *SSTP: a Scalable and Secure Transport Protocol for Smart Grid Data Collection*. Online. 2012. Dostupné z: https://www.researchgate.net/publication/224266283_SSTP_a_Scalable_and_Secure_Transport_Protocol_for_Smart_Grid_Data_Collection. [cit. 2025-04-16].
- [16] DONENFELD, Jason A. *WireGuard: Modern, Secure, Fast VPN Tunnel*. Online. WireGuard. Dostupné z: <https://www.wireguard.com/papers/wireguard.pdf>. [cit. 2025-04-16].
- [17] PALO ALTO NETWORKS. *What is WireGuard?* Online. Palo Alto Networks. Dostupné z: <https://www.paloaltonetworks.com/cyberpedia/what-is-wireguard>. [cit. 2025-04-16].
- [18] ZIMMERMANN, Philip R. *The Official PGP User's Guide*. Cambridge, Mass.: MIT Press, 1995.
- [19] KUROSE, James F. a Keith W. ROSS. *Computer Networking: A Top-Down Approach*. 7th ed. Boston: Pearson Education, 2017.
- [20] MAN7.ORG. *wg-quick(8) - Linux manual page*. Online. MAN7.ORG. Dostupné z: <https://www.man7.org/linux/man-pages/man8/wg-quick.8.html>. [cit. 2025-05-11].
- [21] EASY-RSA. *Easy-RSA: Home*. Online. Easy-RSA. Dostupné z: <https://easy-rsa.readthedocs.io/en/latest/>. [cit. 2025-05-11].
- [22] OPENVPN. *Reference Manual For OpenVPN 2.6*. Online. OpenVPN. Dostupné z: <https://openvpn.net/community-resources/reference-manual-for-openvpn-2-6/>. [cit. 2025-05-12].

- [23] STRONGSWAN. *pki Tool :: strongSwan Documentation*. Online. strongSwan. Dostupné z: <https://docs.strongswan.org/docs/latest/pki/pki.html>. [cit. 2025-05-14].
- [24] STRONGSWAN. *ipsec.secrets Reference*. Online. strongSwan. Dostupné z: <https://wiki.strongswan.org/projects/strongswan/wiki/Ipsecsecrets>. [cit. 2025-05-14].
- [25] LINUX.DIE.NET. *ipsec.conf(5) - Linux manual page*. Online. LINUX.DIE.NET. Dostupné z: <https://linux.die.net/man/5/ipsec.conf>. [cit. 2025-05-14].
- [26] LINUX.DIE.NET. *xl2tpd.conf(5) - Linux manual page*. Online. LINUX.DIE.NET. Dostupné z: <https://linux.die.net/man/5/xl2tpd.conf>. [cit. 2025-05-16].
- [27] LINUX.DIE.NET. *pppd(8) - Linux manual page*. Online. LINUX.DIE.NET. Dostupné z: <https://linux.die.net/man/8/pppd>. [cit. 2025-05-16].

SEZNAM OBRÁZKŮ

Obrázok 1 TLS Handshake	22
Obrázok 2 IKEv2 Process flow.....	32
Obrázok 3 Ukážka L2TP.....	36
Obrázok 4 Topológia testovacej siete	47
Obrázok 5 Topológia siete s aktívnym tunelom	48
Obrázok 6 Spustenie inštalátora.....	51
Obrázok 7 Licenčná dohoda	52
Obrázok 8 Cesta inštalácie.....	53
Obrázok 9 Aktualizácie produktu	53
Obrázok 10 Vytvorenie zástupcov	54
Obrázok 11 Typ licencie	55
Obrázok 12 Spustenie sprievodcu.....	56
Obrázok 13 Výber inštalačného média	57
Obrázok 14 Výber inštalačného média pokračovanie.....	57
Obrázok 15 Špecifikácia virtuálneho stroja	58
Obrázok 16 Konfigurácia diskového priestoru	59
Obrázok 17 Prispôsobenie hardvéru	60
Obrázok 18 Výber virtuálneho stroja.....	61
Obrázok 19 Priradenie k routeru	62
Obrázok 20 Príklad výpisu príkazu ip a.....	63
Obrázok 21 Príklad výpisu ipconfig	63
Obrázok 22 Overenie konektivity VM pomocou ping.....	64
Obrázok 23 Dokončená inštalácia.....	65
Obrázok 24 Výber jazyku Windows server	67
Obrázok 25 Volba edície operačného systému Windows.....	68
Obrázok 26 Výzva na zadanie hesla Windows	69
Obrázok 27 Ukážka spustenia iperf3 Windows	71
Obrázok 28 Výpis privátneho a verejného kľúča	73
Obrázok 29 Spustenie tunelu wireguard na servery.....	75
Obrázok 30 wg show server.....	78
Obrázok 31 wg show klient	78
Obrázok 32 Spustenie OpenVPN server	84
Obrázok 33 Spustenie OpenVPN klient.....	86
Obrázok 34 ping serveru OpenVPN	87

Obrázok 35 Zobrazenie certifikátu ca-cert.pem	93
Obrázok 36 Výstup ipsec statusall.....	95
Obrázok 37 Ukážka rozhrania s pokročilými nastaveniami L2TP	102
Obrázok 38 Nastavenia PPP	102
Obrázok 39 Ukážka výstupu ip a L2TP	103
Obrázok 40 Ukážka Add Roles and Features	104
Obrázok 41 Výsledok konfigurácie AD CS	106
Obrázok 42 Dokončený snap-in	107
Obrázok 43 Strom MMC	108
Obrázok 44 Vlastnosti užívateľa	109
Obrázok 45 Allow access Dial-in	110
Obrázok 46 Role services Remote Access	110
Obrázok 47 Potvrdenie a inštalácia Remote Access.....	111
Obrázok 48 Žiadosť o nový certifikát.....	112
Obrázok 49 Pridaná hodnota sstp.server.....	113
Obrázok 50 Výsledok inštalácie certifikátu.....	113
Obrázok 51 Výber konfigurácie Routing a access.....	114
Obrázok 52 Výber vlastnej konfigurácie	115
Obrázok 53 Výber služby vlastnej konfigurácie.....	115
Obrázok 54 Výstup cat certnew.crt.....	118
Obrázok 55 Výsledok merania Wireguard	120
Obrázok 56 Výsledok merania OpenVPN.....	121
Obrázok 57 Výsledok merania IKEv2.....	121
Obrázok 58 Výsledok merania L2TP	121
Obrázok 59 Výsledok merania SSTP	122

SEZNAM TABULEK

Tabuľka 1 Súhrn nameraných priepustností pre jednotlivé VPN protokoly.....	122
Tabuľka 2 Kryptografické konfigurácie testovaných VPN protokolov	123

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

1-RTT 1 Round Trip Time

AD CS Active Directory Certificate Services

ADUC Active Directory Users and Computers

AEAD Authenticated Encryption with Associated Data

AES Advanced Encryption Standard

APT Advanced Package Tool

CA Certification Authority

CHAP Challenge Handshake Authentication Protocol

CN Common Name

CPU Central Processing Unit

CRL Certificate Revocation List

D-H Diffie-Hellman

DHCP Dynamic Host Configuration Protocol

DNS Domain Name System

DNSSEC Domain Name System Security Extensions

DoS Denial of Service

DPD Dead Peer Detection

DSL Digital Subscriber Line

EAP Extensible Authentication Protocol

ESP Encapsulating Security Payload

GB Gigabyte

GCM Galois/Counter Mode

GHz Gigahertz

HKDF HMAC-based Extract-and-Expand Key Derivation Function

HTTP HyperText Transfer Protocol

HTTPS	HyperText Transfer Protocol Secure
ID	Identity
IKEv1	Internet Key Exchange Version 1
IKEv2	Internet Key Exchange Version 2
IP	Internet Protocol
IPSec	Internet Protocol Security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISAKMP	Internet Security Association and Key Management Protocol
L2F	Layer 2 Forwarding
L2TP	Layer Two Tunneling Protocol
LAC	L2TP Access Concentrator
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LTS	Long Term Support
LNS	L2TP Network Server
Mbps	Megabits per second
MFA	Multi-Factor Authentication
MitM	Man-in-the-Middle
MMC	Microsoft Management Console
MODP	Modular Exponentiation
MS-CHAP	Microsoft Challenge-Handshake Authentication Protocol
MS-CHAPv2	Microsoft Challenge Handshake Authentication Protocol version 2
NAT	Network Address Translation
NAT-T	Network Address Translation – Traversal
OAKLEY	Oakley Key Determination Protocol

PAM	Pluggable Authentication Modules
PAP	Password Authentication Protocol
PDN	Public Data Networks
PEM	Privacy-Enhanced Mail
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PSK	Pre-Shared Key
PPTP	Point-to-Point Tunneling Protocol
QoS	Quality of Service
RADIUS	Remote Authentication Dial-In User Service
RAM	Random-access memory
RSA	Rivest–Shamir–Adleman
SA	Security Association
SAML	Security Assertion Markup Language
SAN	Subject Alternative Names
SHA	Secured Hash Algorithm
SKEME	Secure Key Exchange Mechanism for Internet
SSL/TLS	Secure Socket Layer and Transport Layer Security
SSTP	Secure Socket Tunneling Protocol
TCP	Transmission Control Protocol
TOTP	Time-based One-time Password
UDP	User Datagram Protocol
UFW	Uncomplicated Firewall
URL	Uniform Resource Locator
VM	Virtual Machine
VPN	Virtual Private Network

WAN Wide Area Network