UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2025 Karel Janoušek

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Virtuální laboratorní prostředí pro síťové služby

Bakalářská práce

2025 Karel Janoušek

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2024/2025

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Karel Janoušek**
Osobní číslo:	**I22235**
Studijní program:	**B0688A140009 Informační technologie**
Téma práce:	**Virtuální laboratorní prostředí pro síťové služby**
Zadávající katedra:	**Katedra informačních technologií**

Zásady pro vypracování

Cílem práce je vytvořit virtualizovaný server bez grafického uživatelského rozhraní, který bude poskytovat služby aplikační vrstvy modelu TCP/IP (DNS, HTTP, SMTP, FTP, TFTP, IRC, SSH). Server bude umožňovat pomocí FTP přístup k následujícím stahovaným programům: Wireshark, Nginx web server, SolarWinds TFTP, Putty. Součástí práce bude podrobná dokumentace vytvořeného prostředí a dvou sad laboratorních cvičení. První sada cvičení bude zaměřená na základní konfiguraci jednotlivých služeb a druhá sada na řešení běžných problémů a poruch. Klientská část bude mít grafické uživatelské rozhraní a bude sloužit k plnění připravených laboratorních cvičení.

Rozsah pracovní zprávy: **min. 30**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

RUEST, Danielle a Nelson RUEST. Virtualizace: podrobný průvodce. Vyd. 1. Brno: Computer Press, 2010, 408 s. ISBN 978-80-251-2676-9.
BRITTAIN, Jason a Ian DARWIN. Tomcat: The Definitive Guide. 2008. vyd. Sebastopol: O'Reilly, 2008. ISBN 978-0596-10106-0.
CHAPPELL, Laura. Wireshark Network Analysis. 2012. ISBN 978-189-3939-943.

Vedoucí bakalářské práce: **Ing. Soňa Neradová, Ph.D.**
Katedra informačních technologií

Datum zadání bakalářské práce: **15. prosince 2024**
Termín odevzdání bakalářské práce: **16. května 2025**

L.S.

prof. Ing. Petr Doležel, Ph.D. v.r.
děkan

Ing. Jan Panuš, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 28. února 2025

Prohlašuji:

Práci s názvem Virtuální laboratorní prostředí pro síťové služby jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 25. 08. 2025

Karel Janoušek v.r.

PODĚKOVÁNÍ

Rád bych vyjádřil upřímné poděkování své vedoucí práce, Ing. Soně Neradové, Ph.D., za její odborné vedení, vstřícný přístup, cenné podněty a cenné rady, které mi byly neocenitelnou oporou při zpracování této bakalářské práce. Mé poděkování patří rovněž mým rodičům za jejich trpělivost, podporu a porozumění během celého období mého studia.

ANOTACE

Tato bakalářská práce se zabývá návrhem a realizací virtuálního laboratorního prostředí pro síťové služby. Cílem je vytvořit virtualizovaný server s operačním systémem Ubuntu Server 24.04.3 LTS, který poskytuje služby aplikační vrstvy TCP/IP (DNS, HTTP, SMTP, FTP, TFTP, IRC a SSH), a klientské prostředí Ubuntu Desktop 24.04.3 LTS pro jejich využití. Součástí práce je rovněž příprava dvou sad laboratorních cvičení, zaměřených na základní konfiguraci služeb a řešení běžných problémů, s cílem podpořit praktickou výuku a porozumění principům síťových technologií.

KLÍČOVÁ SLOVA

Virtualizace, Ubuntu Server, Ubuntu Desktop, TCP/IP, DNS, HTTP, SMTP, FTP, TFTP, IRC, SSH, síťové služby, laboratorní cvičení, konfigurace, poruchy

TITLE

Virtual Lab Environment for Network Services

ANNOTATION

This bachelor thesis focuses on the design and implementation of a virtual laboratory environment for network services. The aim is to create a virtualized server running Ubuntu Server 24.04.3 LTS that provides application layer TCP/IP services (DNS, HTTP, SMTP, FTP, TFTP, IRC, and SSH), along with a client environment based on Ubuntu Desktop 24.04.3 LTS for their utilization. The thesis also includes the development of two sets of laboratory exercises, one focused on the basic configuration of services and the other on troubleshooting common issues, in order to support practical education and a deeper understanding of network technologies.

KEYWORDS

Virtualization, Ubuntu Server, Ubuntu Desktop, TCP/IP, DNS, HTTP, SMTP, FTP, TFTP, IRC, SSH, network services, laboratory exercises, configuration, troubleshooting

OBSAH

SEZNAM ILUSTRACÍ

SEZNAM ZKRATEK A ZNAČEK

DNS – Domain Name System

HTTP – HyperText Transfer Protocol

SMTP – Simple Mail Transfer Protocol

FTP – File Transfer Protocol

TFTP – Trivial File Transfer Protocol

IRC – Internet Relay Chat

SSH – Secure Shell

TCP/IP – Transmission Control Protocol / Internet Protocol

IP – Internet Protocol

OSI – Open Systems Interconnection

NIC – Network Interface Card

GUI – Graphical User Interface

LTS – Long Term Support

VM – Virtual Machine

TERMINOLOGIE

Virtualizace: Technologie umožňující provoz více operačních systémů současně na jednom fyzickém hardwaru prostřednictvím virtualizační vrstvy (hypervizoru).

Hypervizor: Software nebo firmware zajišťující vytvoření a správu virtuálních strojů. Rozlišujeme hypervizor typu 1 (běžící přímo na hardware) a typu 2 (běžící nad hostitelským operačním systémem).

Virtuální stroj (VM): Emulovaný počítačový systém provozovaný na fyzickém stroji pomocí hypervizoru.

Ubuntu Server: Linuxová distribuce bez grafického uživatelského rozhraní, určená pro provoz serverových aplikací a služeb.

Ubuntu Desktop: Linuxová distribuce s grafickým uživatelským rozhraním, určená pro běžné uživatele.

Síťová služba: Aplikační proces nebo aplikace poskytující funkcionalitu v síti (např. DNS, HTTP, SMTP).

DNS (Domain Name System): Systém pro překlad doménových jmen na IP adresy a naopak.

HTTP (HyperText Transfer Protocol): Protokol aplikační vrstvy používaný pro přenos webového obsahu.

SMTP (Simple Mail Transfer Protocol): Protokol sloužící pro přenos e-mailových zpráv.

FTP (File Transfer Protocol): Protokol umožňující přenos souborů mezi klientem a serverem.

TFTP (Trivial File Transfer Protocol): Jednoduchý protokol pro přenos souborů, využívaný zejména v prostředích s omezenými prostředky.

IRC (Internet Relay Chat): Protokol umožňující textovou komunikaci uživatelů v reálném čase.

SSH (Secure Shell): Protokol pro bezpečnou vzdálenou správu systémů a šifrovaný přenos dat.

TCP/IP (Transmission Control Protocol / Internet Protocol): Základní sada síťových protokolů používaná pro komunikaci v Internetu a počítačových sítích.

ÚVOD

Rozvoj informačních technologií a rostoucí požadavky na spolehlivost a bezpečnost počítačových sítí kladou důraz na kvalitní přípravu odborníků v této oblasti. Praktická výuka síťových technologií je nedílnou součástí vzdělávání, avšak často naráží na technické i organizační překážky, zejména z důvodu omezených hardwarových prostředků nebo rizika narušení funkčnosti produkčních sítí.

Jedním z efektivních řešení těchto problémů je využití virtualizace, která umožňuje vytvářet flexibilní a izolovaná prostředí pro simulaci a testování síťových služeb. Díky tomu je možné realizovat laboratorní cvičení bez potřeby fyzických zařízení a s vysokou mírou opakovatelnosti a bezpečnosti.

Cílem této bakalářské práce je návrh a implementace virtuálního laboratorního prostředí založeného na operačním systému Ubuntu Server 24.04.3 LTS, které bude poskytovat vybrané služby aplikační vrstvy TCP/IP (DNS, HTTP, SMTP, FTP, TFTP, IRC a SSH). Součástí řešení je také příprava klientského prostředí na systému Ubuntu Desktop 24.04.3 LTS a vytvoření dvou sad laboratorních cvičení. První sada se zaměřuje na základní konfiguraci služeb, zatímco druhá simuluje jejich běžné problémy a poruchy.

Struktura práce je rozdělena na část teoretickou, která popisuje principy virtualizace a vybrané síťové služby, a část praktickou, jež se věnuje návrhu a implementaci prostředí a tvorbě laboratorních cvičení.

1 TEORETICKÁ ČÁST

1.1 Virtualizace a její principy

Virtualizace představuje technologii, která umožňuje provoz více nezávislých operačních systémů na jednom fyzickém počítači. Základním prvkem je tzv. hypervizor, jenž odděluje hardwarové prostředky hostitelského systému a přiděluje je jednotlivým virtuálním strojům. Tyto virtuální stroje pak fungují jako samostatné počítače se svou vlastní konfigurací a operačním systémem.

Význam virtualizace v oblasti počítačových sítí spočívá především v efektivním využití hardwarových prostředků, snadné správě prostředí a možnosti rychle vytvářet izolované testovací a výukové scénáře. Virtualizace rovněž umožňuje centralizovanou správu serverů, vyšší flexibilitu při nasazování služeb a snadné zálohování či obnovu konfigurací.

Rozlišujeme dva základní typy hypervizorů:

- Hypervizor typu 1 (bare-metal) - běží přímo nad fyzickým hardwarem a umožňuje vysoký výkon a efektivní správu zdrojů. Příkladem jsou VMware ESXi, Microsoft Hyper-V nebo Proxmox.
- Hypervizor typu 2 (hostovaný) - běží nad hostitelským operačním systémem, což usnadňuje jeho použití na běžných pracovních stanicích. Typickým příkladem je VirtualBox nebo VMware Workstation.

Virtualizace se stala nedílnou součástí moderního IT prostředí a nachází uplatnění nejen v podnikových datových centrech, ale také ve vzdělávacích institucích, kde umožňuje studentům získat praktické zkušenosti bez nutnosti složité hardwarové infrastruktury.

1.2 Síťové služby aplikační vrstvy

Síťové služby aplikační vrstvy tvoří základní stavební kámen komunikace v počítačových sítích. Zajišťují funkce jako překlad doménových jmen, přenos webového obsahu nebo doručování e-mailů. Tato práce se podrobně věnuje službám DNS, HTTP, SMTP, FTP, TFTP, IRC a SSH, které jsou následně implementovány v praktické části.

1.2.1 DNS (Domain Name System)

Domain Name System (DNS) je hierarchicky uspořádaný a distribuovaný systém, který zajišťuje převod srozumitelných doménových jmen na IP adresy a naopak. Jeho vznik souvisí s potřebou nahradit původní způsob mapování názvů počítačů pomocí statického souboru

hosts.txt, který byl v počátcích internetu manuálně spravován. Jak počet zařízení na síti rostl, tento přístup se ukázal jako neudržitelný, což vedlo k vytvoření systému DNS.[2]

DNS je tvořen několika klíčovými komponentami:

- Resolver – klientská část, která odesílá dotazy DNS serverům.
- Autoritativní server – uchovává oficiální záznamy o konkrétní doméně a vrací odpovědi na dotazy týkající se této domény.
- Rekurzivní server – přijímá dotazy od klienta a postupně je přeposílá dalším DNS serverům, dokud nezíská požadovanou odpověď.

Celý systém funguje na základě hierarchické struktury, jejímž vrcholem jsou tzv. root servery. Ty odkazují na servery domén nejvyšší úrovně (TLD), jako jsou například .com, .org, nebo národní domény typu .cz.

Data v DNS jsou uložena ve formě tzv. resource records (RR). Mezi nejběžnější typy záznamů patří:

- A – přiřazuje doménové jméno k IPv4 adrese,
- AAAA – přiřazuje doménu k IPv6 adrese,
- MX – určuje poštovní server pro danou doménu,
- CNAME – vytváří alias jiného doménového jména,
- NS – určuje, který jmenný server je autoritativní pro danou doménu.

Bez funkčního systému DNS by běžné používání internetu nebylo prakticky možné – uživatelé by museli zadávat IP adresy ručně. Správná konfigurace a údržba DNS proto hraje klíčovou roli v chodu většiny síťových služeb.

1.2.2 HTTP (HyperText Transfer Protocol)

HyperText Transfer Protocol (HTTP) je základní protokol, který zajišťuje přenos dat v prostředí World Wide Webu. Slouží ke komunikaci mezi klientem – obvykle webovým prohlížečem – a serverem. Díky němu je možné přenášet nejen text, ale i obrázky, multimediální obsah a další typy souborů.

HTTP funguje na principu požadavku a odpovědi (request–response):

1. klient (např. prohlížeč jako Chrome, Firefox nebo Edge) odešle požadavek,
2. server tento požadavek zpracuje a vrátí odpověď, která obsahuje buď požadovaná data, nebo chybové hlášení.

Mezi základní vlastnosti HTTP patří:

- standardně běží na portu 80,
- data nejsou šifrována (zabezpečená varianta HTTPS využívá protokol TLS/SSL a port 443),
- podporuje různé metody požadavků – např. GET, POST, PUT, DELETE – které definují, co má server s daty udělat,
- odpovědi serveru jsou doplněny stavovými kódy, jako například:
    - 200 OK – požadavek byl úspěšně zpracován,
    - 404 Not Found – požadovaný obsah nebyl nalezen,
    - 500 Internal Server Error – došlo k chybě na straně serveru.

HTTP se od svého vzniku vyvíjel. Verze HTTP/1.1 zavedla trvalé spojení a další optimalizace. Modernější protokoly HTTP/2 a HTTP/3 [4] přinášejí efektivnější přenos dat pomocí multiplexování a dalších technologií, které snižují latenci a zvyšují spolehlivost.

Díky své univerzálnosti a jednoduchosti se HTTP stal základem webové komunikace. V současnosti je však jeho šifrovaná varianta HTTPS standardem pro většinu webového provozu, protože zajišťuje vyšší úroveň bezpečnosti při přenosu dat mezi klientem a serverem.

1.2.3 SMTP (Simple Mail Transfer Protocol)

Simple Mail Transfer Protocol (SMTP) je základní protokol určený pro přenos e-mailových zpráv v počítačových sítích. Byl poprvé definován v roce 1982 v dokumentu RFC 821 a dodnes tvoří základní kámen infrastruktury elektronické pošty. Jeho aktuální podoba je specifikována v RFC 5321.[6]

SMTP funguje na principu komunikace mezi klientem a serverem. Odesílající server, označovaný jako Mail Transfer Agent (MTA), předává e-mail cílovému serveru, který je zodpovědný za doručování zpráv pro danou doménu. K tomu se využívá DNS, konkrétně MX záznamy (Mail Exchange), které určují, na který server má být pošta směrována.

Mezi klíčové vlastnosti SMTP patří:

- používá standardně port 25 pro komunikaci mezi servery a port 587 pro odesílání zpráv z klientských zařízení,
- je určen výhradně pro přenos zpráv; jejich stahování zajišťují jiné protokoly, jako například POP3 nebo IMAP,
- původní verze nepodporovala šifrování, proto se dnes běžně používá rozšíření STARTTLS, které umožňuje zabezpečený přenos pomocí protokolu TLS,
- samotné zprávy se přenášejí ve formátu ASCII, zatímco přílohy a multimediální obsah jsou k e-mailu připojovány prostřednictvím standardu MIME (Multipurpose Internet Mail Extensions).

SMTP je dnes nepostradatelným nástrojem pro distribuci elektronické pošty. Jeho správná konfigurace, včetně zabezpečení a správného nastavení DNS záznamů, je klíčová pro spolehlivé a bezpečné doručování zpráv.

1.2.4 FTP (File Transfer Protocol)

File Transfer Protocol (FTP) patří mezi nejstarší protokoly aplikační vrstvy, určené k přenosu souborů mezi klientem a serverem. Poprvé byl definován už v roce 1971 a přestože má své limity, dodnes se používá tam, kde je požadován jednoduchý a spolehlivý přenos dat.

FTP funguje na základě modelu klient–server, kdy klient žádá o přístup k souborům a server je zpřístupňuje. Komunikace probíhá standardně přes port 21, zatímco samotný přenos dat využívá dynamicky přidělené porty, v závislosti na režimu přenosu.

Protokol podporuje dva způsoby navazování datového spojení:

- Active mode – datové spojení iniciuje server směrem ke klientovi,
- Passive mode – datové spojení navazuje klient, což je vhodnější ve firewallem chráněných sítích.[7]

Ověření uživatele probíhá zpravidla pomocí uživatelského jména a hesla, ale FTP nabízí i tzv. anonymous přístup, kdy je možné se připojit bez autentizace. Z hlediska bezpečnosti je však důležité zmínit, že standardní FTP nepřenáší data šifrovaně. Pro bezpečnější komunikaci se proto používají rozšíření FTPS (FTP přes TLS) nebo alternativní protokoly jako SFTP, který využívá šifrování prostřednictvím SSH.

Přestože moderní požadavky na bezpečnost vedly k poklesu jeho využití na veřejném internetu, FTP zůstává relevantní zejména v rámci interních sítí a při distribuci souborů, kde není kladen důraz na šifrování přenosu.

1.2.5 TFTP (Trivial File Transfer Protocol)

Trivial File Transfer Protocol (TFTP) je zjednodušená varianta protokolu FTP, navržená pro použití v prostředích s omezenými prostředky. Byl definován v dokumentu RFC 1350 a místo spolehlivého TCP využívá pro přenos transportní protokol UDP, což výrazně zjednodušuje jeho fungování.

Základní charakteristiky TFTP:

- komunikuje přes port 69,
- nepodporuje autentizaci ani žádné pokročilé funkce pro správu souborů,
- je určen především pro automatizovaný přenos malých souborů, typicky při zavádění (bootování) síťových zařízení nebo distribuci konfiguračních souborů,
- jeho jednoduchost je výhodná v uzavřených systémech, ale zároveň znamená bezpečnostní riziko, pokud je použit v otevřených nebo nezabezpečených sítích.

TFTP se i dnes uplatňuje především v oblasti síťové administrace a správy zařízení, kde umožňuje rychlý, bezobslužný přenos souborů bez potřeby složité konfigurace. Díky minimálním nárokům na prostředky se často používá v integrovaných systémech, například při síťovém bootování routerů, switchů nebo jiných zařízení.

1.2.6 IRC (Internet Relay Chat)

Internet Relay Chat (IRC) je komunikační protokol navržený pro přenos textových zpráv v reálném čase. Vznikl v roce 1988 a patří mezi první nástroje, které umožnily veřejnou i soukromou online komunikaci ve skupinách napříč internetem.

IRC funguje na základě modelu klient–server:

- uživatelé se připojují k IRC serveru prostřednictvím specializovaných klientských aplikací,
- komunikace probíhá buď v tzv. kanálech (chatovacích místnostech), nebo formou soukromých zpráv mezi uživateli,

- servery mohou být navzájem propojené do distribuované sítě, což umožňuje sdílenou komunikaci mezi různými komunitami po celém světě.

Charakteristické vlastnosti IRC:

- používá se standardně port 6667,
- samotný protokol je textový a nešifrovaný, proto se v moderních implementacích často doplňuje o šifrování pomocí TLS,
- podporuje uživatelskou hierarchii, včetně operátorů kanálů, kteří mohou řídit přístup, moderovat diskusi nebo spravovat práva ostatních účastníků,
- umožňuje také přenos jednoduchých souborů prostřednictvím protokolu DCC (Direct Client-to-Client).

Ačkoli IRC postupně ustoupilo novějším a pohodlnějším komunikačním platformám, stále si drží místo v některých technicky orientovaných komunitách. Díky své jednoduchosti, nízkým nárokům a nezávislosti na centralizovaných službách zůstává funkčním nástrojem zejména, kde je vyžadována rychlá a přímá textová komunikace bez zbytečných nadstaveb.

1.2.7 SSH (Secure Shell)

Secure Shell (SSH) je síťový protokol určený k bezpečné vzdálené správě zařízení a šifrovanému přenosu dat.[8] Vznikl v roce 1995 jako reakce na bezpečnostní slabiny starších nešifrovaných protokolů, jako byly Telnet nebo rlogin. Od té doby se SSH stal de facto standardem pro vzdálený přístup k serverům a síťovým zařízením.

Hlavní vlastnosti protokolu SSH:

- používá port 22,
- veškerá komunikace probíhá šifrovaně, což zajišťuje důvěrnost a integritu přenášených dat,
- autentizace může být založena na heslech, avšak běžnější a bezpečnější metodou je využití asymetrických kryptografických klíčů,
- kromě přístupu k příkazovému řádku umožňuje bezpečný přenos souborů prostřednictvím protokolů SCP a SFTP,

- podporuje pokročilé funkce jako je tunelování nebo přesměrování portů (port forwarding), které umožňují přístup k dalším službám skrze šifrovaný kanál.

Díky své vysoké úrovni zabezpečení a univerzálnosti je SSH široce nasazováno v oblasti správy serverů, síťových prvků i vestavěných zařízení. Jeho použití je dnes považováno za nezbytný standard v každém prostředí, kde je vyžadována bezpečná vzdálená správa.

1.3 Využití virtualizace v laboratorní výuce

Virtualizace je v současnosti jedním z nejpraktičtějších nástrojů pro výuku počítačových sítí a správy systémů. Umožňuje provozovat několik nezávislých systémů na jednom fyzickém zařízení, což výrazně usnadňuje jak přípravu, tak samotné vedení laboratorních cvičení. Studenti tak mohou bezpečně testovat a zkoušet různé scénáře bez obav, že ohrozí reálný provoz nebo potřebují drahý hardware.

Mezi hlavní výhody virtualizace ve vzdělávání patří:

- Izolace prostředí – každý student nebo každé cvičení může běžet v samostatném virtuálním stroji, takže i chybné nastavení nebo selhání systému neovlivní nikoho dalšího.
- Snadná opakovatelnost – připravené prostředí lze jednoduše naklonovat nebo obnovit do původního stavu, což usnadňuje opakování testů nebo samostatnou práci mimo výuku.
- Úspora nákladů – místo samostatného zařízení pro každého studenta stačí výkonnější server, na kterém lze spustit více virtuálních instancí najednou.
- Bezpečnost – experimentování neohrožuje produkční síť školy a studenti se tak mohou učit i z vlastních chyb.
- Škálovatelnost – počet virtuálních strojů lze přizpůsobit velikosti skupiny i konkrétní náplni výuky.

Ve výuce síťových služeb přináší virtualizace ještě větší přínos – umožňuje studentům prakticky konfigurovat a spravovat služby jako DNS, HTTP, SMTP nebo SSH, testovat jejich dostupnost a simulovat běžné chyby nebo výpadky. Tím si nejen upevní teoretické znalosti, ale také se naučí přemýšlet v souvislostech a hledat řešení reálných problémů.

Virtualizace tak představuje moderní, efektivní a prakticky orientovaný přístup, který odpovídá aktuálním požadavkům v oblasti IT a pomáhá studentům lépe se připravit na skutečnou praxi.

2 PRAKTICKÁ ČÁST

2.1 Návrh a architektura virtuálního prostředí

Cílem praktické části práce je vytvořit izolované, opakovatelné a snadno spravovatelné laboratorní prostředí postavené na virtualizační platformě Oracle VM VirtualBox. Toto prostředí umožňuje bezpečně testovat a konfigurovat síťové služby bez dopadu na fyzickou infrastrukturu hostitelského zařízení.

Základ prostředí tvoří dvojice virtuálních strojů – server a klient – propojených pomocí interní sítě s názvem labnet. Tato síť je zcela oddělena od fyzické sítě hostitele a slouží výhradně pro účely laboratorních úloh. Díky tomu je zajištěna bezpečnost, izolace i plná kontrola nad provozovaným prostředím.

2.1.1 Virtualizační platforma a topologie

Pro realizaci byl zvolen hypervizor typu 2 – Oracle VM VirtualBox (verze 7.1.12), který nabízí jednoduchou správu virtuálních strojů, možnost definovat interní sítě a integrovaný DHCP server pro automatické přidělování IP adres.

Navržená topologie je jednoduchá a přehledná: jeden server a jeden klient jsou propojeni v rámci jedné izolované podsítě. Tento model:

- minimalizuje riziko nechtěného ovlivnění produkčních sítí,
- umožňuje snadné škálování prostředí – například přidáním dalších klientů nebo serverů bez potřeby zásahu do základní architektury,
- poskytuje realistické podmínky pro testování síťových služeb i jejich poruchových stavů.

Takto navržené prostředí vytváří stabilní základ pro následující laboratorní úlohy, které se zaměřují na implementaci a testování konkrétních síťových protokolů.

2.1.2 Logická a adresní struktura sítě

Propojení virtuálních strojů probíhá prostřednictvím interní sítě s názvem labnet, která je spravována integrovaným DHCP serverem VirtualBoxu. Tato síť je vytvořena pomocí nástroje VBoxManage z hostitelského zařízení následovně:

```
.\VBoxManage.exe dhcpserver add --netname labnet --ip 192.168.55.1 --netmask
255.255.255.0 --lowerip 192.168.55.10 --upperip 192.168.55.254 --enable
```

Tím je nastavena podsíť 192.168.55.0/24 s DHCP rozsahem 192.168.55.10–192.168.55.254. Adresní prostor je rozdělen koncepčně na dvě části:

- 192.168.55.1–9: vyhrazeno pro servery (statická IP adresace nebo adresní rezerva),
- 192.168.55.10–254: určeno pro klientské stanice (dynamické přidělování přes DHCP).

Toto rozdělení zajišťuje přehlednou strukturu sítě, umožňuje snadnější správu a zároveň připravuje prostředí na případné rozšíření v budoucnu.

2.1.3 Parametry virtuálních strojů

Oba virtuální stroje – server a klient – byly vytvořeny se shodnými hardwarovými parametry, které zajistí plynulý běh laboratorních úloh a zároveň ponechávají dostatečnou rezervu pro další experimenty.

Server – Ubuntu Server 24.04.3 LTS

- CPU: 4 jádra
- RAM: 4 GB
- Disk: 50 GB (reálně využito přibližně 10 GB)
- Síť:
    - 1× síťový adaptér v režimu NAT (pro přístup k internetu),
    - 1× adaptér v režimu Vnitřní síť (labnet).

Klient – Ubuntu Desktop 24.04.3 LTS

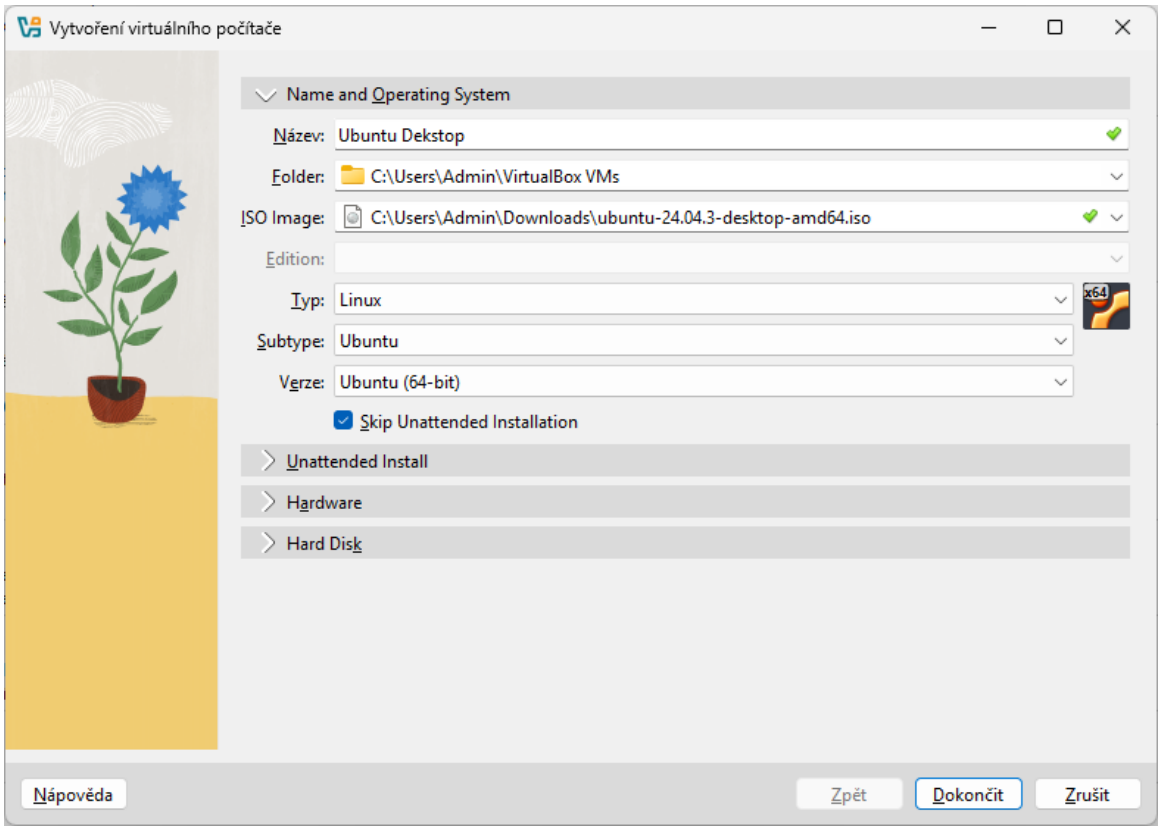
- CPU: 4 jádra
- RAM: 4 GB
- Disk: 50 GB (reálně využito přibližně 15 GB)
- Displej:
    - povolená 3D akcelerace,
    - video paměť 256 MB
- Síť:

- 1× adaptér v režimu NAT,
- 1× adaptér v režimu Vnitřní síť (labnet).

Tato konfigurace poskytuje dostatečný výkon i flexibilitu pro praktické úlohy zaměřené na správu a testování síťových služeb v rámci laboratorního prostředí.

2.2 Založení virtuálních strojů a základní konfigurace ve VirtualBoxu

Pro laboratorní prostředí byly vytvořeny dva virtuální stroje – Ubuntu Desktop a Ubuntu Server.

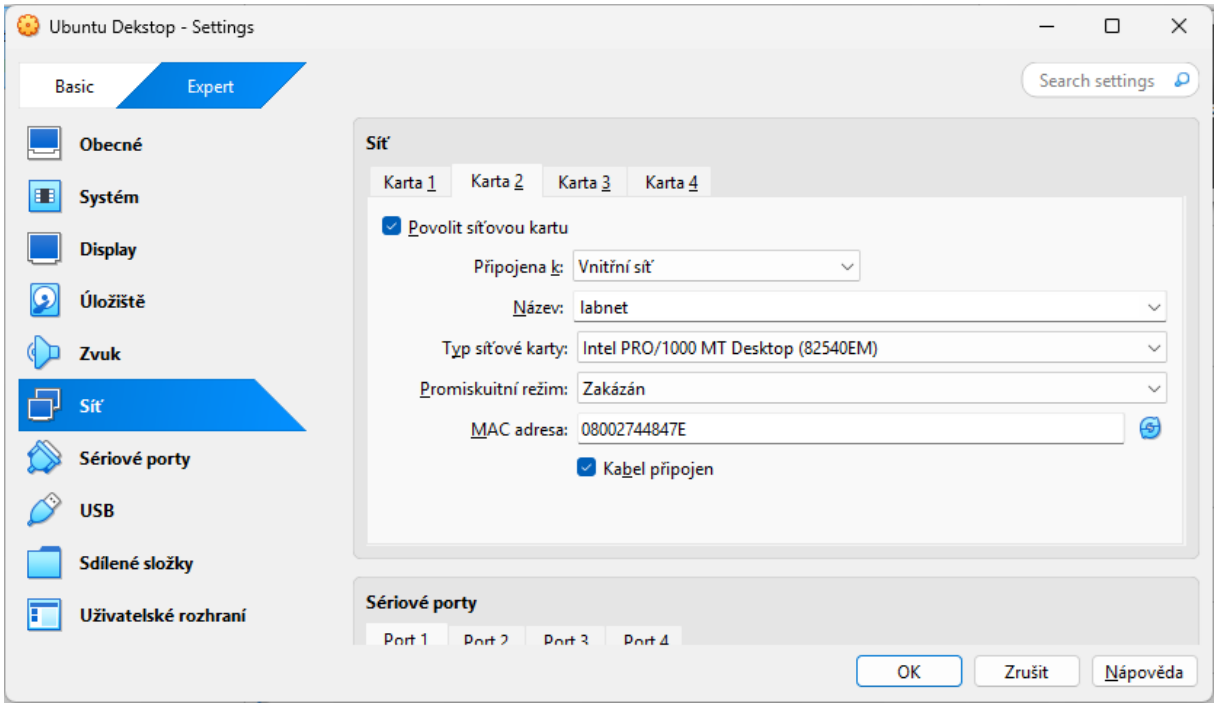


Obrázek 1: Vytváření virtuálního stroje v aplikaci VirtualBox

Ubuntu Desktop

1. Vytvoření nového VM s názvem Ubuntu Desktop.
2. Připojení instalačního ISO obrazu: ubuntu-24.04.3-desktop-amd64.iso.
3. Před zahájením instalace je vypnuta bezobslužná instalace, aby bylo možné nastavit vše ručně.

4. Po vytvoření virtuálního stroje je potřeba přejít do Nastavení -> Režim Expert a provést následující úpravy:
    - Zobrazování (Display): povolit 3D akceleraci a nastavit video paměť na 256 MB.
    - Síť -> Karta 2: povolit druhý síťový adaptér, zvolit režim Vnitřní síť a zadat název sítě labnet.[1]



Obrázek 2: Přidání síťové karty a její nastavení k vnitřní síti labnet

Ubuntu Server

Postup vytvoření serverového VM je obdobný:

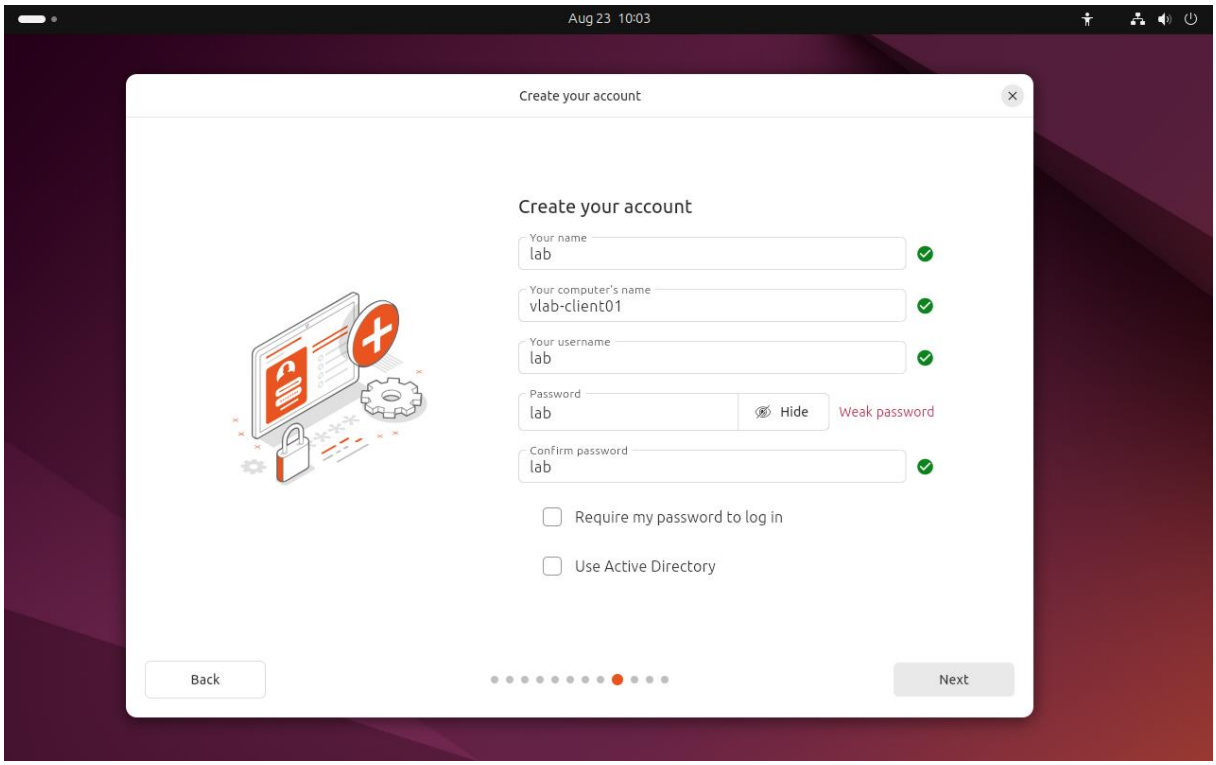
1. Název virtuálního stroje: Ubuntu Server.
2. Připojen instalační ISO obraz: ubuntu-24.04.3-live-server-amd64.iso.
3. V síťových nastaveních je přidán druhý adaptér v režimu Vnitřní síť (labnet), stejně jako u klienta.

2.3 Instalace operačních systémů

Ubuntu Desktop 24.04.3 LTS

Při instalaci klientského operačního systému byly zvoleny následující kroky:

- Neinstalovat third-party software (např. grafické ovladače, kodeky apod.).
- Vybrat možnost Smazat disk a nainstalovat Ubuntu (automatické rozdělení diskového prostoru).
- Vytvořit uživatelský účet:
    - Uživatel: lab
    - Heslo: lab
    - Název počítače: vlab-client01



Obrázek 3: Vytváření nového uživatele v Ubuntu Desktop

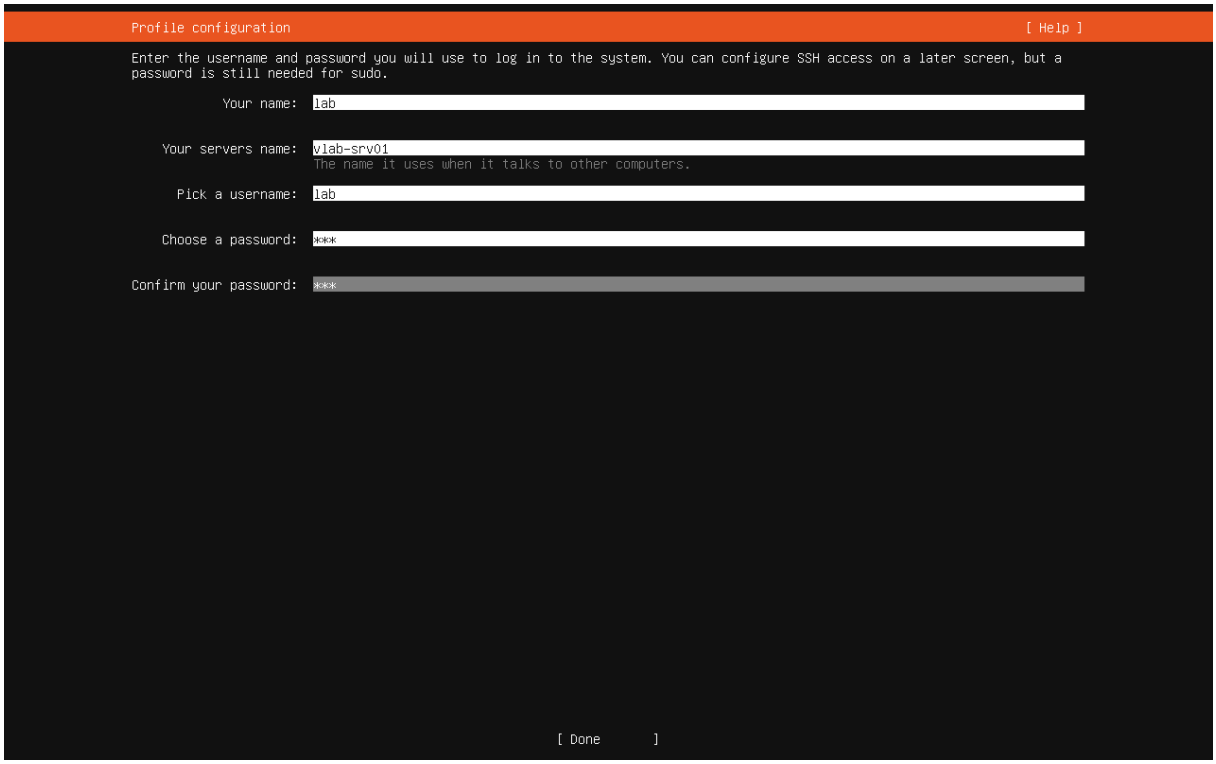
- Zbytek instalace byl dokončen s výchozími volbami.

Ubuntu Server 24.04.3 LTS

Instalace serverového systému probíhala s těmito volbami:

- Vybrána byla plnohodnotná instalace (ne varianta "minimal").
- Nebyl instalován žádný third-party software ani ovladače.

- V části pro nastavení disku bylo potřeba ručně vypnout použití LVM – výchozí nastavení totiž vytváří LVM skupinu, která v laboratorním prostředí není potřebná a může komplikovat další správu disku.
- Vytvořen byl uživatelský účet:
    - Uživatel: lab
    - Heslo: lab
    - Hostname: vlab-srv01



Obrázek 4: Vytváření nového uživatele v Ubuntu Server

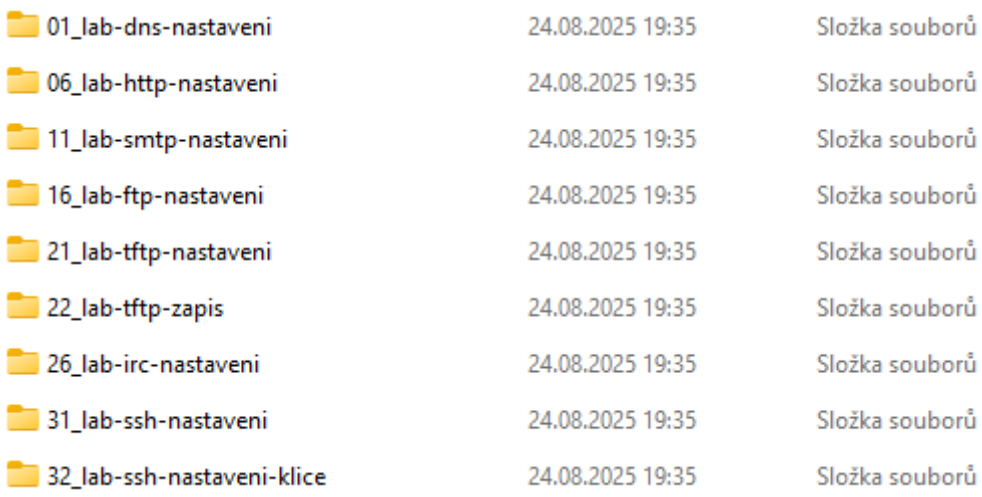
- Volitelně bylo možné při instalaci zaškrtnout instalaci OpenSSH serveru – pokud nebyl přidán, bude SSH doinstalováno ručně v rámci samostatného laboratorního cvičení.
- Instalace byla dokončena bez přidávání dodatečných snap balíčků.

3 LABORATORNÍ CVIČENÍ

Následující textová část uvádí specifikace pro každý úkol. Podrobný popis je přiložený ve složce Labs.zip v digitálním archívu na Stag portalu.

3.1 Sada I – Základní konfigurace služeb

Tato sada laboratorních cvičení je zaměřena na praktické uvedení do provozu základních síťových služeb aplikační vrstvy – konkrétně DNS (BIND9), HTTP (Nginx), SMTP (Postfix), FTP (vsftpd), TFTP (tftpd-hpa), IRC (InspIRCd) a SSH (OpenSSH).



Obrázek 5: Seznam úloh Sada I

3.1.1 LAB 01 – DNS (BIND9) – Základní konfigurace

Úkolem tohoto laboratorního cvičení je nasadit autoritativní DNS server BIND9 na virtuálním stroji vlab-srv01, který bude obsluhovat:

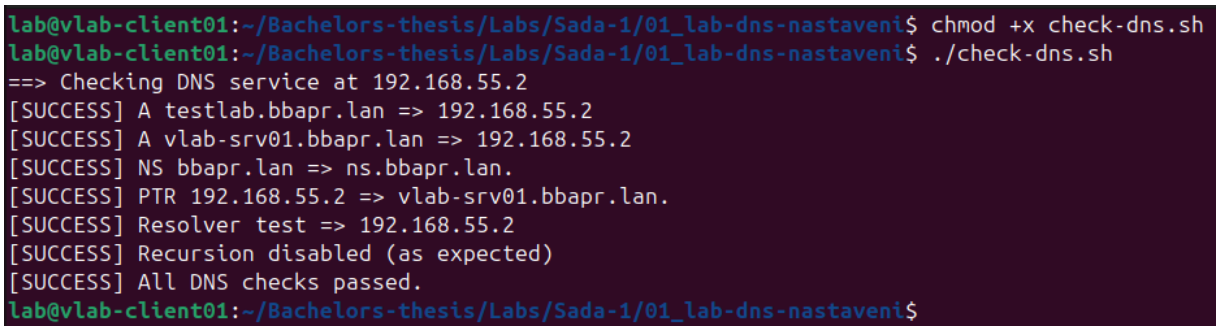
- dopřednou zónu bbapr.lan,
- reverzní zónu 55.168.192.in-addr.arpa.

Součástí úkolu je vypnutí rekurze, nastavení přístupu pouze z lokální sítě a zajištění toho, aby klientský stroj vlab-client01 používal tento DNS server jako svůj primární resolver.[2]

Struktura poskytnutých materiálů

Ve složce 01_lab-dns-nastaveni/ jsou k dispozici tyto podpůrné materiály:

- zadani.txt – stručné zadání cvičení pro studenta,
- check-dns.sh – skript pro automatizovanou kontrolu konfigurace (spouští se na klientovi).



Obrázek 6: Ukázka kontroly pro Laboratorní cvičení 1

Adresář solution/ (pro vyučující):

- server/ – referenční konfigurace BIND9:[3]
    - named.conf.local – definice zón bbapr.lan a 55.168.192.in-addr.arpa,
    - named.conf.options – naslouchání na 127.0.0.1 a 192.168.55.2, povolené dotazy pouze z localhost a sítě 192.168.55.0/24, rekurze zakázána (recursion no;),
    - db.bbapr.lan a db.192.168.55 – vzorové zónové soubory s definicemi SOA, NS, A a PTR záznamů,
    - apply-solution-server.sh – skript pro nasazení referenční konfigurace a statické IP.
- client/ – podpora pro konfiguraci klienta:
    - apply-solution-client.sh + šablona client-60-labnet.yaml.tpl – přesměrování resolveru na 192.168.55.2 a instalace nástroje dnsutils.

Zadání pro studenta

Server (vlab-srv01)

1. Nastavit statickou IP adresu 192.168.55.2/24 na rozhraní připojeném do sítě labnet.
2. Nainstalovat a nakonfigurovat BIND9 jako autoritativní server pro:
    - zónu bbapr.lan, která obsahuje alespoň:
        - NS záznam pro ns.bbapr.lan,
        - A záznamy pro ns, vlab-srv01, testlab -> všechny na 192.168.55.2,

- reverzní zónu 55.168.192.in-addr.arpa, která obsahuje:
    - PTR záznam: 192.168.55.2 -> vlab-srv01.bbapr.lan.

3. V souboru named.conf.options nastavit:
    - naslouchání na 127.0.0.1 a 192.168.55.2,
    - allow-query pouze pro localhost a 192.168.55.0/24,
    - recursion no; (rekurze vypnuta).[3]
4. V obou zónách definovat správnou SOA hlavičku se Serial ve formátu YYYYMMDDnn.
5. Ověřit funkčnost služby a případně upravit nastavení firewallu (UFW) pro porty 53/UDP a 53/TCP.

Klient (vlab-client01)

- Přesměrovat DNS resolver na 192.168.55.2 (pomocí Netplanu nebo systemd-resolved).
- Ověřit funkčnost dotazů pomocí nástroje dig.

Automatická kontrola – check-dns.sh

Skript check-dns.sh umožňuje automatickou kontrolu správnosti konfigurace DNS. Spouští se na klientovi např. takto:

```
chmod +x ./check-dns.sh && sudo ./check-dns.sh
```

Skript provádí následující testy:

- ověření A záznamu: testlab.bbapr.lan -> 192.168.55.2,
- ověření PTR záznamu: 192.168.55.2 -> vlab-srv01.bbapr.lan,
- ověření, že rekurze je zakázána (např. dotaz na example.com nevrátí odpověď),
- kontrola, zda DNS služba běží a odpovídá správně zadaným klientům.

Výstup skriptu obsahuje přehledné značky [SUCCESS] / [FAILED] a návratový kód 0 v případě plného úspěchu (jinak 1). To umožňuje plně automatizované vyhodnocení výsledků cvičení.

3.1.2 LAB 06 – HTTP (Nginx) – Základní nasazení

V tomto cvičení se studenti seznámí se základy nasazení webového serveru Nginx. Cílem je zpřístupnit jednoduchou testovací stránku na adrese http://vlab-srv01.bbapr.lan/ a zároveň zprovoznit adresářovou listaci souborů dostupnou pod cestou /files/.

Poskytnuté materiály

Složka 06_lab-http-nastaveni/ obsahuje vše potřebné k provedení úlohy:

- zadani.txt – specifikace požadavků pro studenta,
- client/check-http.sh – automatizační skript pro ověření funkčnosti (spouští se na klientském stroji),
- solution/ – referenční řešení určené pro vyučující:
    - lab-http.conf – ukázkový konfigurační soubor pro Nginx, naslouchající na IP adrese serveru a s aktivní listací v /files/,
    - index.html – výchozí úvodní stránka, která signalizuje správný běh služby (obsahuje text „BBAPR – Nginx je v provozu“),
    - srv_www_lab_http/files/sample.txt – ukázkový soubor, který by měl být viditelný v adresářové listaci,
    - apply-solution.sh – skript pro automatizované nasazení celého řešení na straně serveru.

Automatická kontrola – check-http.sh

Správnost řešení lze ověřit pomocí připraveného skriptu check-http.sh, který se spouští na klientském systému s oprávněním root [5]. V případě potřeby si sám doinstaluje potřebný nástroj curl.

Skript testuje:

- že server vrací HTTP odpověď s kódem 200,[4]
- že úvodní stránka obsahuje text „BBAPR – Nginx je v provozu“,
- že adresář /files/ je dostupný a obsahuje soubor sample.txt.

Při splnění všech podmínek skript vypisuje výstup ve formátu [SUCCESS] / [FAILED] a vrací návratový kód 0, což je možné využít pro automatizované vyhodnocení výsledků.

3.1.3 LAB 11 – SMTP (Postfix) – Základní relaying v lokální síti

Tato laboratorní úloha se zaměřuje na nasazení poštovního serveru Postfix pro potřeby lokálního doručování zpráv v rámci vnitřní sítě. Cílem je umožnit odeslání e-mailu z klientského stroje na adresu lab@bbapr.lan tak, aby byla zpráva doručena do poštovní schránky uživatele lab na serveru vlab-srv01. Odesílání nebo příjem zpráv mimo lokální síť (např. na veřejný internet) není předmětem tohoto cvičení.[6]

Poskytnuté materiály

V adresáři 11_lab-smtp-nastaveni/ jsou připraveny následující podpůrné soubory:

- zadani.txt – přehled úkolu určený studentovi,
- client/check-smtp.sh – skript, který na klientovi odesílá testovací zprávu a vyhodnocuje její úspěšné přijetí,
- server/check-delivery.sh – pomocný nástroj pro ověření doručení zprávy do souboru /var/mail/lab na serveru,
- solution/ – referenční řešení pro vyučující, obsahující:
    - main.cf – ukázkovou konfiguraci Postfixu s následujícími klíčovými parametry:
        - myhostname = vlab-srv01.bbapr.lan
        - mydomain = bbapr.lan
        - myorigin = $mydomain
        - mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain[6]
        - inet_interfaces = 192.168.55.2, 127.0.0.1
        - inet_protocols = ipv4
        - mynetworks = 127.0.0.0/8, 192.168.55.0/24

        - smtpd_recipient_restrictions = permit_mynetworks, reject_unauth_destination

- apply-solution.sh – skript, který instaluje potřebné balíčky (postfix, mailutils), aplikuje konfiguraci, otevře port 25 ve firewallu (UFW) na síti labnet a spustí službu.

Automatická kontrola – check-smtp.sh

Skript check-smtp.sh se spouští na klientském zařízení a k odesílání e-mailu používá nástroj swaks. Při spuštění vygeneruje jedinečný identifikátor X-BBAPR-ID, odešle zprávu na SMTP server a vyhodnotí úspěšnost na základě odpovědi.

Základní parametry (lze změnit pomocí proměnných prostředí):

- HOST – cílový SMTP server (výchozí: vlab-srv01.bbapr.lan),
- TO – příjemce (výchozí: lab@bbapr.lan),
- FROM – odesílatel (výchozí: student@bbapr.lan),
- TIMEOUT, ATTEMPTS, SLEEP_BETWEEN – nastavení pro pokusy o doručení.

3.1.4 LAB 16 – FTP (vsftpd) – Anonymní read-only server

Tato laboratorní úloha se zaměřuje na zprovoznění anonymního FTP serveru pomocí služby vsftpd. Server běží ve read-only režimu, nevyužívá TLS a zpřístupňuje veřejný repozitář umístěný v adresáři /srv/ftp/pub. Přístup je umožněn výhradně anonymním uživatelům bez možnosti zápisu.

Server je zároveň nakonfigurován pro PASV režim s vymezeným rozsahem portů 30000–30009/TCP a statickou IP adresou 192.168.55.2, na které naslouchá.

Poskytnuté materiály

Ve složce 16_lab-ftp-nastaveni/ jsou připraveny následující soubory:

- zadani.txt – přehled požadavků pro studenta,
- client/check-ftp.sh – skript, který z klienta ověřuje přístup na server a testuje základní funkčnost (listování adresářů, stahování souboru),
- solution/ – referenční řešení pro vyučující, obsahuje:
    - vsftpd.conf – ukázková konfigurace vsftpd s důrazem na klíčové direktivy:

        - anonymous_enable=YES
        - local_enable=NO
        - write_enable=NO
        - anon_root=/srv/ftp
        - ssl_enable=NO
        - pasv_enable=YES
        - pasv_min_port=30000
        - pasv_max_port=30009
        - pasv_address=192.168.55.2[7]
        - listen=YES
        - listen_ipv6=NO
    - apply-solution.sh – skript, který:
        - vytvoří adresářovou strukturu /srv/ftp/pub/ a podadresáře wireshark, nginx, solarwinds-tftp, putty s ukázkovými soubory,
        - povolí potřebné porty ve firewallu (21 a 30000–30009/TCP),
        - spustí a nakonfiguruje službu vsftpd.

Automatická kontrola – check-ftp.sh

Skript check-ftp.sh slouží ke kontrole funkčnosti FTP serveru z klientské stanice. Připojení probíhá pomocí nástroje curl přes anonymní FTP.

Spouští se následovně:

- sudo ./check-ftp.sh.

Testuje se následující:

- že na adrese ftp://$HOST/pub/ existují složky wireshark, nginx, solarwinds-tftp, putty,
- že je možné stáhnout soubor putty.txt z adresáře pub/putty/ do dočasného umístění,

- po stažení je vypočítán a vypsán SHA256 hash souboru pro manuální porovnání s originálem umístěným na serveru (/srv/ftp/pub/putty/putty.txt).

3.1.5 LAB 21 TFTP (tftpd-hpa) – Základní konfigurace TFTP

V tomto cvičení je cílem nasadit jednoduchý TFTP server pomocí balíčku tftpd-hpa na serveru vlab-srv01. Server bude fungovat v režimu read-only a zpřístupní soubory uložené v adresáři /srv/tftp. Ověření funkčnosti probíhá z klientské stanice stažením testovacího souboru a kontrolou jeho obsahu.

Poskytnuté materiály

Složka 21_lab-tftp-nastaveni/ obsahuje:

- zadani.txt – specifikace úkolu pro studenta,
- client/check-tftp.sh – skript pro automatizované ověření funkčnosti ze strany klienta,
- solution/ – referenční řešení určené pro vyučující, zahrnující:
    - apply-solution.sh – skript, který:
        - vytvoří adresářovou strukturu /srv/tftp/test/ s odpovídajícími právy a vlastníkem tftp:tftp,
        - připraví testovací soubor hello.txt s obsahem "Hello World",
        - upraví konfigurační soubor /etc/default/tftpd-hpa s typickým nastavením služby:
            - TFTP_USERNAME="tftp"
            - TFTP_DIRECTORY="/srv/tftp"
            - TFTP_ADDRESS="192.168.55.2:69"
            - TFTP_OPTIONS="--secure"
        - aktivuje službu tftpd-hpa a případně upraví firewall (UFW) pro UDP port 69 v rámci interní sítě.

Automatická kontrola – check-tftp.sh

Skript check-tftp.sh slouží k ověření dostupnosti a správné konfigurace TFTP serveru. Spouští se z klienta následovně:

sudo ./check-tftp.sh

Funkce skriptu:

- Pokusí se stáhnout soubor test/hello.txt z TFTP serveru,
- Pokud je dostupný TFTP klient, použije jej; jinak provede fallback přes curl s protokolem tftp://,
- Výchozí cesta pro uložení staženého souboru je /tmp/hello.txt,
- Skript následně přečte první řádek souboru a ověří, že začíná textem "Hello World".

3.1.6 LAB 22 TFTP (tftpd-hpa) – Povolení zápisu (PUT)

Tato laboratorní úloha navazuje na předchozí konfiguraci TFTP serveru a rozšiřuje ji o podporu zápisu souborů (PUT). Cílem je umožnit, aby bylo možné nahrávat soubory pouze do vyhrazené složky /srv/tftp/upload, zatímco zbytek TFTP kořenového adresáře zůstane v režimu read-only.

Poskytnuté materiály

Ve složce 22_lab-tftp-zapis/ se nachází:

- zadani.txt – specifikace úkolu pro studenta,
- client/check-tftp-upload.sh – skript, který z klienta provádí kompletní test nahrání a následného stažení souboru,
- solution/ – referenční řešení pro vyučující:
    - apply-solution.sh – automatizační skript, který:
    - vytvoří složku /srv/tftp/upload s vlastníkem tftp:tftp a právy pro zápis,
    - ponechá kořen /srv/tftp s oprávněními 755 (pouze pro čtení),
    - upraví konfigurační soubor /etc/default/tftpd-hpa následovně:
        - TFTP_USERNAME="tftp"
        - TFTP_DIRECTORY="/srv/tftp"
        - TFTP_ADDRESS="192.168.55.2:69"
        - TFTP_OPTIONS="--secure --create"

- aktivuje službu tftpd-hpa a případně upraví firewall (UFW) pro port 69/UDP v rámci interní sítě.

Automatická kontrola – check-tftp-upload.sh

Skript check-tftp-upload.sh ověřuje, zda je možné soubor do serveru nejen nahrát, ale i zpětně stáhnout, a zkontrolovat tak zachování jeho obsahu. Spouští se z klienta:

sudo ./check-tftp-upload.sh

Průběh testu:

1. Skript vytvoří dočasný soubor /tmp/BBAPR22-<timestamp>.txt s jedinečným obsahem.
2. Provede nahrání (PUT) tohoto souboru do umístění upload/ na serveru:
    - preferovaně pomocí klienta tftp,
    - pokud není dostupný, použije curl s protokolem tftp://.
3. Pokusí se soubor stáhnout zpět (GET) do /tmp/<soubor>.dl.txt.

3.1.7 LAB 26 IRC (InspIRCd) – Základní nasazení IRC serveru

Tato laboratorní úloha je zaměřena na základní nasazení IRC serveru pomocí služby InspIRCd. Cílem je umožnit připojení klienta ke kanálu #lab, odeslání zprávy a ověření, že server správně reaguje a umožňuje základní komunikaci podle IRC protokolu.

Po úspěšném připojení na port 6667/TCP by měl klient obdržet kód 001 (úspěšné přihlášení), vstoupit do kanálu #lab a odeslat zprávu.

Poskytnuté materiály

Ve složce 26_lab-irc-nastaveni/ jsou k dispozici tyto soubory:

- zadani.txt – zadání pro studenta obsahující popis cíle a akceptační podmínky,
- client/check-irc.sh – skript pro automatizovaný test, který naváže spojení s IRC serverem, ověří odpověď kódem 001, provede vstup do kanálu a odešle zprávu,
- solution/ – referenční konfigurace pro vyučující:
    - apply-solution.sh – skript, který:

- nainstaluje InspIRCd,
- vytvoří minimální konfigurační soubor /etc/inspircd/inspircd.conf,
- povolí port 6667/TCP ve firewallu pro síť labnet,
- restartuje službu.

Pro účely laboratorního cvičení se používá pouze nešifrované spojení bez TLS.

Automatická kontrola – check-irc.sh

Skript check-irc.sh umožňuje ověřit funkčnost IRC serveru z klientského zařízení. Test je interaktivní a využívá nástroj nc (netcat). Spouští se následovně:

sudo ./check-irc.sh

Co skript kontroluje:

- naváže spojení s IRC serverem na adrese 192.168.55.2:6667,
- očekává přijetí kódu 001, který potvrzuje úspěšné přihlášení,
- odešle standardní sekvenci IRC příkazů:
  - NICK, USER,
  - JOIN #lab,
  - PRIVMSG #lab :ahoj z klienta,
  - QUIT,

veškerý průběh ukládá do souboru /tmp/irc26.out.

3.1.8 LAB 31 SSH (OpenSSH) – Základní konfigurace SSH

V rámci tohoto cvičení je cílem zprovoznit službu OpenSSH na serveru vlab-srv01, a to tak, aby se uživatel lab mohl připojit prostřednictvím hesla přes síť labnet na standardním portu 22/TCP. Konfigurace musí být nastavena tak, aby:

- server explicitně naslouchal pouze na adrese 192.168.55.2,
- bylo zakázáno přihlášení uživatele root,

- bylo povoleno přihlášení pomocí hesla.

Poskytnuté materiály

Ve složce 31_lab-ssh-nastaveni/ se nachází:

- zadani.txt – stručný popis požadavků a cílů úlohy,
- client/check-ssh.sh – skript pro automatizované ověření, zda se klient dokáže přihlásit na server přes SSH pomocí hesla,
- solution/ – referenční řešení pro vyučující:
    - apply-solution.sh – skript, který:
    - doinstaluje balíček openssh-server,
    - vytvoří konfigurační soubor /etc/ssh/sshd_config.d/bbapr.conf s následujícím minimem:
        - Port 22
        - ListenAddress 192.168.55.2[8]
        - PermitRootLogin no
        - PasswordAuthentication yes
    - restartuje službu ssh,
    - povolí port 22/TCP ve firewallu (UFW) pro síť labnet.

Automatická kontrola – check-ssh.sh

Skript check-ssh.sh slouží k ověření, zda je možné se z klienta připojit na server přes SSH a úspěšně se přihlásit pomocí hesla. Spouští se jednoduše: ./check-ssh.sh

Co skript ověřuje:

- dostupnost nástrojů ssh a sshpass; pokud chybí, skript vypíše instalační pokyny,
- provede přihlášení na zadaného uživatele (SSH_USER) na cílový server (HOST),
- po úspěšném přihlášení spustí příkaz whoami.

3.1.9 LAB 32 SSH (OpenSSH) – Autentizace pomocí klíčů

Tato úloha navazuje na předchozí cvičení a rozšiřuje konfiguraci SSH o autentizaci pomocí kryptografických klíčů. Cílem je umožnit uživateli lab z klientského stroje vlab-client01 přihlášení na vlab-srv01 bez nutnosti zadávání hesla, a to pomocí klíče typu ED25519. V závěru je možné volitelně zakázat přihlášení heslem a ponechat pouze přihlašování přes klíč.

Poskytnuté materiály

Ve složce 32_lab-ssh-nastaveni-klice/ se nachází:

- zadani.txt – popis cíle a postupu pro studenta,
- client/ – nástroje pro klientskou stanici:
    - gen-and-install-key.sh – skript pro vygenerování ED25519 klíče (výchozí cesta ~/.ssh/lab_ed25519) a nahrání veřejné části na server pomocí ssh-copy-id. Pro první připojení využívá heslo lab,
    - check-ssh-keyonly.sh – skript pro ověření, že je možné přihlášení pouze pomocí klíče, bez nutnosti zadávání hesla.
- solution/ – referenční konfigurace pro vyučující:
    - apply-solution.sh – povolí autentizaci pomocí klíčů (PubkeyAuthentication yes) a dočasně ponechá heslové přihlášení (PasswordAuthentication yes) kvůli ssh-copy-id. Také otevře port 22/TCP ve firewallu a restartuje službu ssh,
    - harden-disable-password.sh – volitelný krok, který zakáže přihlášení heslem a ponechá pouze přístup klíčem (PasswordAuthentication no).

Automatická kontrola – generování a nasazení klíče

Skript gen-and-install-key.sh se spouští z klienta a automatizuje celý proces vygenerování klíče a jeho nahrání na server. Výchozí použití:

./client/gen-and-install-key.sh

Volitelné proměnné:

- HOST – cílový server (výchozí: vlab-srv01.bbapr.lan),
- USER – přihlašovací jméno (výchozí: lab),

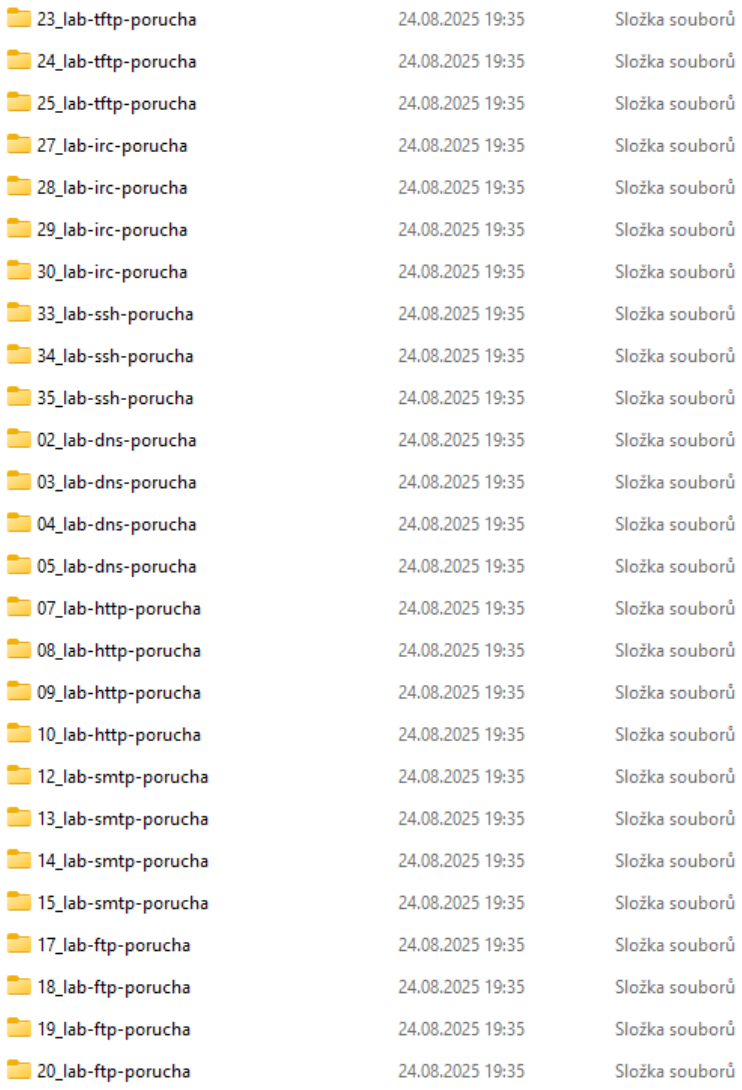
- PASS – heslo uživatele (výchozí: lab),
- KEY – cesta ke generovanému klíči (výchozí: ~/.ssh/lab_ed25519).

Skript:

- zkontroluje dostupnost nástrojů ssh, sshpass a ssh-copy-id,
- vygeneruje klíč bez passphrase,
- nahraje veřejnou část na server.

3.2 Sada II – Poruchy a trouble shooting

Tato sada se zaměřuje na simulaci a řešení běžných chyb v konfiguraci služeb. Cílem je naučit se rozpoznat příčinu problému, provést opravu a ověřit správnou funkčnost.



Obrázek 7: Seznam úloh Sada II

3.2.1 LAB 02-05 – DNS (BIND9) – Shrnutí poruch

V této části byly simulovány čtyři vybrané poruchy v konfiguraci autoritativního DNS serveru BIND9, které odpovídají reálným chybám, s nimiž se může správce sítě setkat. Každý scénář je připraven ve formě skriptu na straně serveru (apply-broken.sh), který poruchu aktivuje, a kontrolního skriptu na klientovi (check-porucha.sh), který ověřuje, zda je konfigurace opravena.

Kontrola vždy ověřuje tři základní podmínky:

- A záznam testlab.bbapr.lan správně ukazuje na 192.168.55.2,
- PTR záznam pro 192.168.55.2 odpovídá vlab-srv01.bbapr.lan,
- DNS server má zakázanou rekurzi (externí dotaz je odmítnut).

LAB 02 – Dotazy z klienta jsou odmítány

- Symptom: Klient neobdrží odpověď – REFUSED nebo prázdná odpověď.
- Příčina: V souboru named.conf.options je povoleno dotazování pouze z localhost:
    - allow-query { localhost; };
- Oprava: Rozšíření povolených klientů o celou laboratorní síť:
    - allow-query { localhost; 192.168.55.0/24; };
- Následně je potřeba reloadovat konfiguraci:
    - sudo rndc reload

(V případě aktivního UFW zkontrolovat pravidla pro porty 53/TCP a 53/UDP.)

LAB 03 – Dotazy končí na timeout

- Symptom: Dotazy z klienta (dig) vyprší, nedochází k odpovědi.
- Příčina: DNS server naslouchá pouze na rozhraní 127.0.0.1:
    - listen-on { 127.0.0.1; };
- Oprava: Přidání IP adresy serveru v síti labnet do seznamu rozhraní:
    - listen-on { 127.0.0.1; 192.168.55.2; };

- Poté reload:
    - sudo rndc reload.

LAB 04 – Neplatný SOA serial (zóna se nenačte)

- Symptom: DNS server vrací SERVFAIL, zóna se nenačítá, odpovědi chybí.
- Příčina: Chybný formát serial čísla v záznamu SOA – například obsahuje písmeno:
    - 2025A81302; neplatný seriál.
- Oprava: Opravit serial na čistě číselný formát YYYYMMDDnn:
    - 2025081302; validní seriál.
- Ověřit syntaxi:
    - sudo named-checkzone bbapr.lan /etc/bind/db.bbapr.lan,
    - sudo rndc reload.

LAB 05 – Syntaktická chyba v SOA (chybí závorka)

- Symptom: Dotazy selhávají (SERVFAIL), zóna se nenačítá.
- Příčina: V záznamu SOA chybí uzavírací závorka):
    - @ IN SOA ns.bbapr.lan. admin.bbapr.lan. (
    - 2025081303
    - 604800 86400 2419200 604800
    - ;) zde chybí závorka
- Oprava: Dopsat závorku a pro přehlednost okomentovat jednotlivé hodnoty:
    - @ IN SOA ns.bbapr.lan. admin.bbapr.lan. (
    - 2025081303; Serial (YYYYMMDDnn)
    - 604800; Refresh
    - 86400; Retry
    - 2419200; Expire

    - 604800); Negative Cache TTL

- Poté ověřit konfiguraci a reloadovat službu:
    - sudo named-checkzone bbapr.lan /etc/bind/db.bbapr.lan
    - sudo rndc reload

3.2.2 LAB 07-10 – HTTP (Nginx) – Shrnutí poruch

Tato část shrnuje čtyři vybrané poruchy související s provozem webového serveru Nginx, které byly simulovány v rámci laboratorního prostředí. Každá porucha reprezentuje běžné konfigurační chyby, s nimiž se lze při nasazování webového serveru setkat.[5]

Ověření správné funkce probíhá pomocí skriptu check-porucha.sh spouštěného na klientovi. Ten kontroluje zejména:

- dostupnost služby na adrese http://vlab-srv01.bbapr.lan/,
- návratový kód HTTP 200 (OK),
- přítomnost specifického textu „BBAPR – Nginx je v provozu“ v těle odpovědi.

LAB 07 – Webová stránka není dostupná z klienta

- Symptom: Klientská stanice nedokáže navázat spojení – stránka se nenačte, dojde k timeout nebo connection refused.
- Příčina: Server Nginx naslouchá pouze na 127.0.0.1 (loopback) a není dostupný z interní sítě labnet.
    - listen 127.0.0.1:80;
- Oprava: Změnit konfiguraci na explicitní naslouchání na IP adrese serveru:
    - listen 192.168.55.2:80;
- Poté ověřit syntaxi:
    - sudo nginx -t
    - sudo systemctl reload nginx

LAB 08 – Nesprávně nastavený dokumentový kořen

- Symptom: Po načtení stránky se zobrazí chybová hláška 404 nebo jiný obsah než očekávaný.
- Příčina: Direktiva root v server blocku směřuje na výchozí adresář /var/www/html namísto požadovaného /srv/www/lab-http.
- Oprava: Aktualizovat cestu k dokumentovému kořeni a zajistit funkční adresářovou listaci ve /files/:

```
server {
  listen 192.168.55.2:80;
  server_name vlab-srv01.bbapr.lan;
  access_log /var/log/nginx/lab_http_access.log;
  error_log /var/log/nginx/lab_http_error.log;
  root /srv/www/lab-http;
  index index.html;
  location /files/ { autoindex on; }
}[5]
```

- Následně ověřit konfiguraci a reloadovat službu.

LAB 09 – Chyba v syntaxi konfigurace

- Symptom: Služba Nginx není spuštěna, nebo při reloadu hlásí chybu. Klient nedostává žádnou odpověď.
- Příčina: V konfiguračním souboru se nachází syntaktická chyba – např. chybějící středník, neuzavřený blok nebo neplatná direktiva.
- Oprava: Opravit celý server block do validní podoby:

```
server {
  listen 192.168.55.2:80;
  server_name vlab-srv01.bbapr.lan;
  root /srv/www/lab-http;
  index index.html;
  location /files/ { autoindex on; }
}[5]
```

- Ověřit pomocí:

- sudo nginx -t
- sudo systemctl start nginx

LAB 10 – HTTP port blokován firewallem

- Symptom: Konfigurace serveru je správná, ale připojení z klienta stále selhává.
- Příčina: UFW (Uncomplicated Firewall) blokuje příchozí připojení na port 80/TCP, případně není povoleno směrem z adaptéru připojeného k síti labnet.
- Oprava: Povolování provozu v rámci rozhraní labnet (např. enp0s8):
  - sudo ufw delete deny in on enp0s8 to any port 80 proto tcp || true
  - sudo ufw allow in on enp0s8 to any port 80 proto tcp
  - sudo ufw reload

3.2.3 LAB 12-15 – SMTP (Postfix) – Shrnutí poruch

Následující přehled se zaměřuje na čtyři typické chyby spojené s provozem poštovního serveru Postfix v rámci interní laboratorní sítě. Tyto poruchy simulují běžné konfigurační nedostatky nebo bezpečnostní omezení, která mohou zabránit správnému odeslání nebo doručení e-mailové zprávy.

Správná funkce služby je ověřována nástrojem swaks, který provede odeslání zprávy na adresu lab@bbapr.lan přes SMTP server vlab-srv01.bbapr.lan. Na cílovém serveru se následně kontroluje přítomnost unikátní hlavičky X-BBAPR-ID ve schránce uživatele lab (/var/mail/lab).

LAB 12 – SMTP port nedostupný z klienta

- Symptom: Nástroj swaks hlásí chybu připojení – spojení není navázáno (timeout, connection refused).
- Příčina: Firewall UFW blokuje příchozí spojení na port 25/TCP v rámci rozhraní labnet.
- Oprava: Umožnit příchozí připojení na port 25 v rámci rozhraní enp0s8:
  - sudo ufw delete deny in on enp0s8 to any port 25 proto tcp || true
  - sudo ufw allow in on enp0s8 to any port 25 proto tcp

    - sudo ufw reload
- Následně lze ověřit naslouchání službou:
    - ss -ltnp | grep :25
    - postfix check

LAB 13 – Relay access denied

- Symptom: Pokus o odeslání zprávy končí chybou „Relay access denied“.
- Příčina: Server neakceptuje doménu bbapr.lan jako lokální, protože není uvedena v konfiguraci mydestination v souboru main.cf.
- Oprava: Ujistit se, že doména je správně zahrnuta:
    - myhostname = vlab-srv01.bbapr.lan
    - mydomain = bbapr.lan
    - myorigin = $mydomain
    - mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
    - smtpd_recipient_restrictions = permit_mynetworks, reject_unauth_destination
- Restart služby:
    - sudo systemctl restart postfix

LAB 14 – Zpráva není doručena do očekávané schránky

- Symptom: Odeslání proběhne bez chyby, ale zpráva není nalezena v /var/mail/lab.
- Příčina: Pošta je doručována jiným způsobem – například do adresáře ~/Maildir/ místo systémového souboru mbox. To nastává při nastavení:
    - home_mailbox = Maildir/
- Oprava: Obnovit výchozí způsob doručování (systémový mbox):
    - # home_mailbox = Maildir/
    - home_mailbox =
- Poté je nutné znovu odeslat testovací zprávu a ověřit obsah souboru /var/mail/lab.

LAB 15 – Firewall opět blokuje SMTP

- Symptom: Chování odpovídá chybě z LAB 12 – swaks se nemůže připojit na port 25.
- Příčina: UFW stále obsahuje pravidlo blokující port 25 pro rozhraní labnet.
- Oprava: Stejný postup jako u LAB 12:
    - sudo ufw delete deny in on enp0s8 to any port 25 proto tcp || true
    - sudo ufw allow in on enp0s8 to any port 25 proto tcp
    - sudo ufw reload
- Diagnostiku lze podpořit těmito příkazy:
    - ss -ltnp | grep :25
    - postconf -n | grep '^inet_interfaces'

3.2.4 LAB 17-20 – FTP (vsftpd) – Shrnutí poruch

Tato sada laboratorních úloh se zaměřuje na diagnostiku a řešení běžných problémů spojených s provozem anonymního a uživatelského FTP serveru vsftpd. Testování probíhá prostřednictvím skriptu client/check-porucha.sh, který ověřuje úspěšné připojení, výpis adresářů a přenos souborů, především v pasivním režimu (PASV), a to jak anonymně, tak prostřednictvím lokálního uživatelského účtu ftpuser.[7]

LAB 17 – Blokace PASV režimu firewallovým pravidlem

- Symptom: Řídicí spojení na port 21/TCP funguje, ale přenos dat v PASV režimu (např. výpis obsahu, stažení souboru) selhává s chybou nebo časovým limitem.
- Příčina: Firewall (UFW) blokuje rozsah portů 30000–30009/TCP, který je použit pro datové spojení v pasivním režimu.
- Oprava: Povolení pasivních portů na rozhraní interní sítě labnet:
    - sudo ufw delete deny in on enp0s8 to any port 30000:30009 proto tcp || true
    - sudo ufw allow in on enp0s8 to any port 30000:30009 proto tcp
    - sudo ufw reload

LAB 18 – Nesprávná adresa pro PASV připojení

- Symptom: Připojení k serveru proběhne, ale datový přenos selhává – výpis nebo přenos souboru neproběhne.
- Příčina: Direktiva pasv_address je nastavena na 127.0.0.1, tedy loopback, což není dostupné z klienta v síti. Navíc může být zakázán aktivní režim.
- Oprava: Opravit adresu a povolit aktivní přenosy:
    - pasv_address=192.168.55.2[7]
    - port_enable=YES
- Tyto hodnoty musí být uvedeny v souboru /etc/vsftpd.conf.

LAB 19 – Neúspěšné přihlášení lokálním uživatelem

- Symptom: Uživatelské přihlašovací údaje ftpuser/labftp nejsou akceptovány a klient se nemůže přihlásit.
- Příčina: Konfigurace FTP serveru zakazuje lokální uživatele (local_enable=NO).
- Oprava: Povolit přihlášení lokálních uživatelů a případně nastavit odpovídající prostředí:
    - local_enable=YES
    - chroot_local_user=YES
    - local_root=/srv/ftp/private
- Poté restartovat službu vsftpd a ověřit funkčnost.

LAB 20 – Uživatelský seznam blokuje přihlášení

- Symptom: Přihlašovací pokus uživatele ftpuser selže, přestože heslo je správné.
- Příčina: Seznam /etc/vsftpd.user_list funguje jako seznam zakázaných uživatelů (userlist_deny=YES) a přítomnost ftpuser vede k jeho zablokování.
- Oprava: Přepnout logiku seznamu tak, aby fungoval jako whitelist:
    - userlist_enable=YES

- userlist_file=/etc/vsftpd.user_list
- userlist_deny=NO

- Obsah souboru vsftpd.user_list musí obsahovat řádek ftpuser.

3.2.5 LAB 23-25 – FTP (vsftpd) – Shrnutí poruch

Laboratorní cvičení zaměřená na protokol TFTP (Trivial File Transfer Protocol) se věnují běžným problémům, které mohou nastat při konfiguraci služby tftpd-hpa. Kontrolní skript client/check-porucha.sh testuje schopnost serveru obsloužit požadavky na stažení (GET) a v případě LAB 25 také nahrání (PUT) souborů v rámci sítě labnet.

LAB 23 – Blokovaný port UDP 69

- Symptom: Klientský nástroj hlásí timeout, chybu spojení nebo že se server neozývá.
- Příčina: Firewall na serveru blokuje přístup na port 69/UDP, který TFTP využívá pro přenos.
- Oprava: Povolit provoz na portu 69/UDP pro rozhraní v síti labnet:
  - IFACE=enp0s8
  - sudo ufw delete deny in on "$IFACE" to any port 69 proto udp || true
  - sudo ufw allow in on "$IFACE" to any port 69 proto udp
  - sudo ufw reload

LAB 24 – Nesprávně nastavený TFTP_DIRECTORY

- Symptom: Klient se připojí, ale hlásí chybu „File not found“ i u souborů, které by měly být dostupné (např. test/hello.txt).
- Příčina: Chybná cesta ke kořenovému adresáři TFTP serveru v souboru /etc/default/tftpd-hpa.
- Oprava: Ujistit se, že konfigurace odpovídá očekávanému umístění:
  - TFTP_USERNAME="tftp"
  - TFTP_DIRECTORY="/srv/tftp"
  - TFTP_ADDRESS="192.168.55.2:69"

- TFTP_OPTIONS="--secure"

- Poté restartovat službu:
  - sudo systemctl restart tftpd-hpa

LAB 25 – Selhání uploadu kvůli právům

- Symptom: Klient se pokusí nahrát soubor (PUT) do složky upload/, ale operace selže. Následné stažení (GET) logicky selhává, protože soubor neexistuje.
- Příčina: Proces tftpd-hpa nemá oprávnění zapisovat do adresáře /srv/tftp/upload.
- Oprava: Ujistit se, že adresář má správného vlastníka a dostatečná oprávnění:
  - sudo chown tftp:tftp /srv/tftp/upload
  - sudo chmod 777 /srv/tftp/upload
  - sudo systemctl restart tftpd-hpa
- Dále je třeba zkontrolovat, že v konfiguračním souboru je povoleno vytváření nových souborů:
  - TFTP_OPTIONS="--secure --create"

3.2.6 LAB 27-30 – IRC (InsprIRCd) – Shrnutí poruch

Služba IRC představuje jednoduchý, ale specifický protokol pro textovou komunikaci. V laboratorním prostředí je testována pomocí skriptu client/check-porucha.sh, který ověřuje základní připojení na port 6667/TCP, proběhnutí IRC handshake (včetně přijetí kódu 001), vstup do kanálu #lab a úspěšné odeslání zprávy. Níže jsou popsány typické poruchy a jejich řešení.

LAB 27 – Blokace portu 6667 v UFW

- Symptom: Klient se nedokáže připojit – pokus o spojení vyprší nebo je okamžitě odmítnut.
- Příčina: Firewall (UFW) blokuje příchozí připojení na port 6667/TCP z interní sítě labnet.
- Oprava: Odstranit případné blokování a výslovně povolit komunikaci:
  - IFACE=enp0s8

- sudo ufw delete deny in on "$IFACE" to any port 6667 proto tcp || true
- sudo ufw allow in on "$IFACE" to any port 6667 proto tcp
- sudo ufw reload

LAB 28 – Naslouchání pouze na loopbacku

- Symptom: Server sice běží a port 6667 je otevřen, ale dostupný jen z localhostu.
- Příčina: Konfigurace <bind> směřuje pouze na adresu 127.0.0.1.
- Oprava: Upravit bind direktivu tak, aby zahrnovala IP v síti labnet:
  - <bind address="192.168.55.2" port="6667" type="clients">
- Poté restartovat službu:
  - sudo systemctl restart inspircd

LAB 29 – Restriktivní přístupová politika (allow)

- Symptom: Připojení po TCP proběhne, ale server během přihlašování klienta odmítne.
- Příčina: Direktiva <connect> povoluje přístup pouze z 127.0.0.1/32.
- Oprava: Upravit pravidlo tak, aby byla povolena celá laboratorní podsíť:
  - <connect allow="192.168.55.0/24" maxchans="20" timeout="60" pingfreq="120" sendq="262144" recvq="8192">
- A opět restartovat službu:
  - sudo systemctl restart inspircd

LAB 30 – Vyžadované přihlašovací heslo

- Symptom: Klient pošle handshake, ale spojení je serverem odmítnuto kvůli absenci hesla (např. chybové hlášení „bad password“).
- Příčina: V konfiguraci <connect> je zadáno pole password="...", což vyžaduje, aby se klient ověřil heslem pomocí IRC příkazu PASS.
- Oprava: Odebrat atribut password a ponechat pouze povolenou síť:

- <connect allow="192.168.55.0/24" maxchans="20" timeout="60" pingfreq="120" sendq="262144" recvq="8192">

- Potvrdit změnu restartem serveru:
  - sudo systemctl restart inspircd

3.2.7 LAB 33-35 – SSH (OpenSSH) – Shrnutí poruch

Služba SSH tvoří základ pro vzdálenou správu unixových systémů. V laboratorním prostředí je testována jak varianta přihlášení pomocí hesla, tak i autentizace pomocí asymetrických klíčů. Kontrolní skript client/check-porucha.sh ověřuje, zda je port 22/TCP dostupný, zda funguje přihlášení uživatele lab pomocí hesla a zda je funkční přihlášení pomocí předem nainstalovaného ED25519 klíče.[8]

LAB 33 – Blokace SSH portu 22 v UFW

- Symptom: Klient není schopen navázat TCP spojení na port 22 – spojení skončí chybou timeout nebo connection refused.
- Příčina: Firewall (UFW) blokuje port 22/TCP na rozhraní labnet, ačkoliv služba běží správně.
- Oprava: Odblokovat port 22/TCP na síťovém rozhraní serveru:
  - IFACE=enp0s8
  - sudo ufw delete deny in on "$IFACE" to any port 22 proto tcp || true
  - sudo ufw allow in on "$IFACE" to any port 22 proto tcp
  - sudo ufw reload

LAB 34 – Služba naslouchá na nestandardním portu

- Symptom: SSH služba není dosažitelná na portu 22, přestože je spuštěna a funkční.
- Příčina: Konfigurace SSH serveru nastavuje jiný port, typicky 2222/TCP, který není v souladu s očekáváním skriptu a klienta.
- Oprava: Vrátit konfiguraci na standardní port 22 a upravit odpovídající pravidla firewallu:
  - Soubor /etc/ssh/sshd_config.d/bbapr-port.conf:

        - ListenAddress 192.168.55.2[8]
        - Port 22
        - PasswordAuthentication yes
        - PermitRootLogin no
- Následně restartovat službu a aktualizovat UFW:
    - sudo systemctl restart ssh
    - sudo ufw delete allow in on enp0s8 to any port 2222 proto tcp || true
    - sudo ufw allow in on enp0s8 to any port 22 proto tcp
    - sudo ufw reload

LAB 35 – Selhání autentizace klíčem (StrictModes)

- Symptom: SSH spojení je navázáno, ale přihlášení pomocí klíče skončí neúspěchem (zpravidla bez výzvy k zadání hesla, přesto odmítnuto).
- Příčina: Při aktivovaném režimu StrictModes yes server kontroluje přísně oprávnění a vlastnictví klíčových souborů a adresářů. Pokud jsou oprávnění příliš volná, spojení je odmítnuto z bezpečnostních důvodů.
- Oprava: Opravit vlastnictví a oprávnění pro adresář ~/.ssh a soubor authorized_keys, následně restartovat SSH:
    - USER=lab HOME=/home/lab
    - sudo chown -R "$USER:$USER" "$HOME/.ssh"
    - sudo chmod 700 "$HOME/.ssh"
    - sudo chmod 600 "$HOME/.ssh/authorized_keys"
    - sudo systemctl restart ssh

4. BUDOUCÍ ROZVOJ A ROZŠÍŘENÍ

Tato kapitola navrhuje, jak by bylo možné dále rozvíjet vytvořené virtuální laboratorní prostředí. Cílem je ukázat možnosti, jak stávající topologii (virtuální server a klient ve VirtualBoxu, interní síť labnet a sada laboratorních úloh) rozšířit směrem k dalším síťovým službám, vyšší bezpečnosti, automatizaci a také k didaktickým scénářům. Základní myšlenkou je zachovat jednoduchost, opakovatelnost a izolovanost prostředí, včetně podpory automatického testování pomocí skriptů typu check-*.sh. To vše s možností škálování na více uživatelů a komplexnější úlohy.

4.1 Možnosti rozšíření služeb a bezpečnosti

Tato část shrnuje vybrané možnosti rozšíření služeb se zaměřením na bezpečnost, které lze implementovat a testovat v laboratorním prostředí.

4.1.1 HTTPS a správa certifikátů

Pomocí Nginx je možné zavést podporu šifrované komunikace přes HTTPS (TLS). V testovacím prostředí lze využít tyto varianty:

- Vlastní certifikační autorita (CA) a instalace důvěryhodného kořenového certifikátu na klienta.
- Self-signed certifikáty pro základní demonstraci TLS.
- Bezpečnostní rozšíření:
    - OCSP stapling.
    - HTTP Strict Transport Security (HSTS).
    - Bezpečnostní hlavičky (např. CSP, X-Frame-Options).

4.1.2 DNSSEC pro autoritativní DNS server

Zónu bbapr.lan lze zabezpečit pomocí DNSSEC:

- Podepisování zóny pomocí nástrojů dnssec-keygen a dnssec-signzone.
- Práce s klíči KSK/ZSK a jejich rotace.
- Možnost použití lokálního trust anchoru i bez nadřazené domény.

4.1.3 Příjem a odesílání pošty (IMAP/POP3 + SASL)

Pomocí Dovecotu lze přidat možnost:

- Přístupu k doručené poště (IMAP/POP3).

- Autentizace pomocí SASL při odesílání přes port 587/TCP.
- Šifrování spojení pomocí STARTTLS.

4.1.4 Bezpečný přenos souborů (FTPS)

- FTPS – např. pomocí vsftpd s podporou TLS.

4.1.5 Reverzní proxy

Nginx může plnit roli reverzní proxy pro vícero backend serverů. Možnosti testování:

- Simulace výpadků backendu.
- Přepínání požadavků (max_fails, retry).

ZÁVĚR

Cílem této bakalářské práce bylo navrhnout a realizovat virtuální laboratorní prostředí zaměřené na vybrané síťové služby aplikační vrstvy modelu TCP/IP a připravit k němu dvě sady laboratorních úloh – jednu pro základní konfiguraci služeb a druhou pro řešení typických poruch. Celé řešení bylo postaveno na virtualizační platformě Oracle VM VirtualBox s využitím interní sítě labnet a dvou virtuálních strojů: vlab-srv01 (Ubuntu Server 24.04.3 LTS) a vlab-client01 (Ubuntu Desktop 24.04.3 LTS). Server poskytuje služby DNS, HTTP, SMTP, FTP, TFTP, IRC a SSH; FTP slouží i jako úložiště nástrojů (např. Wireshark, Nginx, SolarWinds TFTP, PuTTY). Celé prostředí je doplněno jednotně strukturovanou dokumentací a sadou ověřovacích skriptů.

Výsledkem je funkční, izolované a opakovatelně použitelné laboratorní prostředí, které umožňuje studentům bezpečně testovat a konfigurovat síťové služby bez rizika zásahu do produkční infrastruktury. První sada cvičení se zaměřuje na správné nasazení jednotlivých služeb (např. autoritativní DNS v BIND9, webový server Nginx, mail server Postfix pro lokální doručování, FTP a TFTP v různých režimech, IRC server InspIRCd a SSH s podporou autentizace klíčem). Druhá sada se soustředí na běžné poruchy – od chybné syntaxe v DNS až po firewall blokující služby. Každé cvičení je doplněno automatickým testovacím skriptem (check-*.sh), který umožňuje rychlé ověření funkčnosti a podporuje spravedlivé hodnocení.

Původním záměrem bylo odevzdat již předpřipravené virtuální stroje ve formátu Open Virtualization Format (OVF). Kvůli omezením dostupného úložiště na straně Univerzitní knihovny ale nebylo možné nahrát tak objemné soubory, a proto jsem zvolil alternativní přístup – dodání prostředí ve formě konfiguračních skriptů.

Zároveň jsem kvůli možným licenčním omezením nepřistoupil k distribuci instalačních souborů nástrojů jako Wireshark, Nginx, SolarWinds TFTP nebo PuTTY pomocí FTP. Místo toho jsou ve cvičeních použity „placeholder“ soubory, které tyto nástroje nahrazují bez porušení licenčních podmínek.

Práce ukazuje, že virtualizace je praktickým a efektivním nástrojem pro výuku síťových technologií: umožňuje rychlou obnovu výchozího stavu, snižuje nároky na hardware a usnadňuje škálování i samostatnou práci studentů. Z pohledu správy je prostředí přehledné, dobře dokumentované a připravené na další rozvoj.

Některé oblasti – jako HTTPS, DNSSEC, IMAP/POP3, SFTP/FTPS, reverzní proxy, podpora IPv6 nebo infrastruktura jako kód – zůstaly mimo rozsah této práce. Jsou však uvedeny v kapitole o možném rozšíření jako doporučené směry budoucího vývoje. Těmi mohou být například bezpečnostní rozšíření, další síťové služby, automatizace pomocí nástrojů typu Ansible nebo zavedení kontinuální integrace pro ověřování konzistence úloh.

POUŽITÁ LITERATURA

[1] ORACLE. Oracle VM VirtualBox User Manual. Verze 7.2. Online. Dostupné z: https://www.virtualbox.org/manual/ [cit. 2025-08-24].

[2] MOCKAPETRIS, Paul V. Domain names – concepts and facilities. RFC 1034. Online. Fremont (CA): RFC Editor, 1987. Dostupné z: https://www.rfc-editor.org/rfc/rfc1034 [cit. 2025-08-17].

[3] INTERNET SYSTEMS CONSORTIUM. BIND 9 Administrator Reference Manual (ARM). Verze 9.21.11. Online. Dostupné z: https://bind9.readthedocs.io/en/v9.21.11/ [cit. 2025-08-18].

[4] FIELDING, Roy T.; NOTTINGHAM, Mark; RESCHKE, Julian (eds.). HTTP Semantics. RFC 9110. Online. RFC Editor, 2022. Dostupné z: https://www.rfc-editor.org/rfc/rfc9110 [cit. 2025-08-18].

[5] NGINX, Inc. Beginner's Guide. Online. Dostupné z: https://nginx.org/en/docs/beginners_guide.html [cit. 2025-08-18].

[6] KLENSIN, John C. Simple Mail Transfer Protocol. RFC 5321. Online. RFC Editor, 2008. Dostupné z: https://www.rfc-editor.org/rfc/rfc5321 [cit. 2025-08-18].

[7] POSTEL, Jon; REYNOLDS, Joyce. File Transfer Protocol. RFC 959. Online. RFC Editor, 1985. Dostupné z: https://www.rfc-editor.org/rfc/rfc959 [cit. 2025-08-19].

[8] YLONEN, Tatu; LONVICK, Chris. The Secure Shell (SSH) Protocol Architecture. RFC 4251. Online. RFC Editor, 2006. Dostupné z: https://www.rfc-editor.org/rfc/rfc4251 [cit. 2025-08-24].

SEZNAM PŘÍLOH

Příloha A: Archív „Labs“ (sady laboratorních cvičení)

- Digitální archív obsahuje dvě části: Sadu I a Sadu II, které slouží jako podpora k laboratorní části práce.
- Archív je přiložena v digitální podobě v knihovně Univerzity Pardubice jako součást této bakalářské práce.