VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

SYSTÉM PRO ŘÍZENÍ MODULŮ OPTICKÝCH VLÁKNOVÝCH ZESILOVAČŮ
SYSTEM FOR CONTROLLING OPTICAL FIBER AMPLIFIER MODULES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Jan Matoušek**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Petr Münster, Ph.D.**
SUPERVISOR

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Jan Matoušek ***ID:*** 186137

Ročník: 2 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Systém pro řízení modulů optických vláknových zesilovačů

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je návrh a realizace systému pro komunikaci a ovládání modulů používaných v optických telekomunikačních sítích. Práce bude dále řešit komunikaci jednodeskového počítače s periferiemi (LCD, tlačítka, moduly optických vláknových zesilovačů) a zabezpečení OS Linux proti útokům přes komunikační rozhraní Ethernet. Výsledkem diplomové práce bude funkční zařízení včetně ovládání přes LCD s tlačítky a univerzálním ovládacím rozhraním pro komunikaci s moduly.

DOPORUČENÁ LITERATURA:

[1] NEGUS, Chris. Linux bible. Ninth edition. Indianapolis, Indiana: John Wiley & Sons, 2015. ISBN 978-1118999875.

[2] BARRETT, Daniel J., Richard E. SILVERMAN a Robert G. BYRNES. Linux security cookbook. Sebastopol, CA: O'Reilly, 2003. ISBN 978-0596003913.

Termín zadání: 3.2.2020 ***Termín odevzdání:*** 1.6.2020

Vedoucí práce: Ing. Petr Münster, Ph.D.

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Tato diplomová práce se zabývá návrhem telekomunikačního zařízení pro optické zesilovače a realizací univerzálního softwaru pro komunikaci s optickým zesilovačem. Bylo vytvořeno univerzální ovládací rozhraní pro ovládání modulů vzdáleně pomocí vzdáleného přístupu pomocí protokolu SSH nebo pomocí LCD displeje a tlačítek na zařízení. Dále je uvedeno zabezpečení systému Linux. Byl proveden návrh zařízení do 1U skříně, kdy byl navržen přední a zadní panel včetně rozmístění prvků uvnitř zařízení. Následně byla vyzkoušena funkčnost zařízení, komunikace s LCD displejem a optickým modulem.

KLÍČOVÁ SLOVA

Gentoo, Linux, optika, Raspberry, zabezpečení

ABSTRACT

This diploma thesis deals with the design of the telecommunication device for optical amplifiers and the implementation of the universal software for thecommunication with an optical amplifier. An universal control interface has been created to control modules remotely via the SSH remote access or via the LCD display and buttons on the device. The Linux security is listed in this thesis. The device was designed in the 1U case, where the front and rear panels and the arrangement of elements inside were designed. Subsequently, the functionality of the device and the communication with the LCD display and the optical module were tested.

KEYWORDS

Gentoo, Linux, optics, Raspberry, security

MATOUŠEK, Jan. *Systém pro řízení modulů optických vlákonových zesilovačů*. Brno, 2020, 67 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Petr Münster, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Systém pro řízení modulů optických vlákonových zesilovačů“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Petru Münsterovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod

Tato diplomová práce se zabývá propojením jednodeskového počítače s optickým modulem. V zadání je požadováno vytvoření konfiguračního rozhraní pro konfiguraci modulu a také propojení s LCD displejem a tlačítky. Dále je požadováno zabezpečení systému OS Linux.

První kapitola se zabývá rozdělením sítí z několika pohledů například dle rozlohy, účelu v síti atd. Dále jsou zde zmíněny typy optických zesilovačů dle umístění na optické trase, Ramanovském a EDFA zesilovači.

Druhá kapitola obsahuje obecné schéma telekomunikačního zařízení, informace o napájecích zdrojích, komunikačních rozhraních. Dále jsou uvedeny informace o systému GNU/Linux, jeho distribucích a minimálních požadavcích na jednotlivé distribuce. Také jsou zde uvedeny možnosti zabezpečení systému proti útoku a možné útoky na systém GNU/Linux.

Poslední kapitola je věnována praktické realizaci práce. Je zde uveden návrh předního a zadního panelu 1U, informace o jednotlivých částech zařízení a propojení modulu, LCD displeje a tlačítek s jednodeskovým počítačem. Následně je uvedena instalace a konfigurace základního systému Gentoo Linux. Dále je zde popsán program pro ovládání modulu pomocí vzdáleného přístupu, zobrazování hodnot pomocí LCD displeje a ovládání tlačítky. Na závěr kapitoly je uvedeno zabezpečení systému proti útokům.

1 Optické telekomunikační sítě

1.1 Rozdělení sítí

Telekomunikační sítě lze rozdělit podle několika hledisek. Jedním z možných dělení je rozdělení podle rozlohy, která je pokryta.

Nejmenší rozlohu pokrývají sítě PAN (Personal Area Network), propojují zařízení na několik metrů. Propojení na tuto vzdálenost umožní například technologie Bluetooth. Větší síť je LAN (Local Area Network), která může propojovat počítače v rámci podniku, školního kampusu a podobně. Tato síť má obvykle maximální dosah několik kilometrů a standardně bývá realizována optickými nebo metalickými propoji.

Síť MAN (Metropolitan Area Network) je obvykle používána na propojení menších sítí LAN, čtvrtí v rámci města. Zde je již žádoucí vysoká rychlost přenosu. Největší sítí v tomto rozdělení je síť WAN (Wide Area Network), která propojuje předchozí sítě a překračuje území států. Jde běžně o páteřní propoje mezi státy nebo kontinenty. Vyžadována je vysoká propustnost a rychlost přenosu.[1]

Další možností je rozdělení dle účelu, který část sítě plní, na několik vrstev.[2]

Přístupová vrstva Jde o nejzákladnější vrstvu umožňující uživatelům přístup do sítě. Jsou zde připojeny koncové stanice, servery, tiskárny IP (Internet Protocol) telefony a podobně. Propojení těchto zařízení je zpravidla pomocí přepínačů případně bezdrátových přístupových bodů. Tato vrstva je připojena k distribuční. V této části sítě je vhodné řešit omezení broadcast provozu případně označování paketů pro následné použití QoS (Quality of Service) politiky v síti.

Distribuční vrstva Tato vrstva slouží k propojení přístupové a páteřní vrstvy. Je zde obvykle provedena filtrace provozu dle zdrojových adres, rozdělení všesměrových zpráv a sítí například na virtuální sítě. Připojení do páteřní vrstvy je obvykle několika linkami z důvodu zvýšení dostupnosti v případě poruchy. Propojení je realizováno směrovači.

Páteřní vrstva Tato vrstva sítě propojuje distribuční síť s dalšími nebo páteřní sítí poskytovatele. Jde o nejvyšší vrstvu v tomto dělení. Redundantní linky by měly být samozřejmostí. Síť vyžaduje vysokou propustnost, jelikož obhospodařuje velký objem dat. Používají se vysokorychlostní přepínače a směrovače.

Pokud se nevyplatí mít oddělenou páteřní a distribuční vrstvu, je možné sloučit tyto vrstvy dohromady a vytvořit tak jednu vrstvu.

Další možné dělení je podle topologie, kterou síť tvoří. Nejzákladnější typy jsou uvedeny níže: [3]

Kruhová topologie Každý prvek sítě je propojen s dalšími dvěma a všechny systémy tvoří kruh (poslední je připojen k prvnímu). Existují i další varianty této topologie například Dvojitý kruh, který řeší výpadek jedné linky mezi prvky v síti.

Sběrnicová topologie Jedná se o jednu z nejstarších topologii. Prvky sítě jsou připojeny na společné médium, které spolu sdílejí pro komunikaci.

Hvězdicová topologie Tento typ topologie je dnes vidění velice často v sítích LAN. Existuje zde jeden prvek (obvykle přepínač), který propojuje ostatní prvky v síti.

A další možností rozdělení je například na fyzickou a logickou topologii, kde logická topologie bývá realizována pomocí tzv. VLAN (Virtual Local Area Network)

Optické sítě se obvykle používají na úrovni sítě MAN a WAN případně na úrovni páteřní a distribuční sítě.

1.1.1 Typy optických zesilovačů

V této části bude popsáno několik typů optických zesilovačů z pohledu zesilování optického signálu.

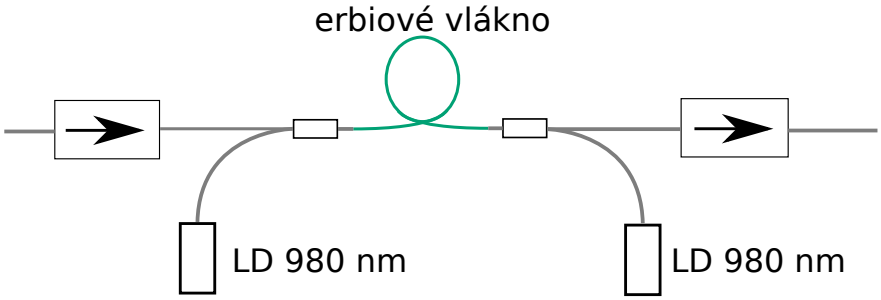
V optických telekomunikačních sítích se používá několik typů zesilovačů, které lze rozdělit podle jejich umístění na optické trase.

Na výstupu ze zařízení může být umístěn výkonový zesilovač tzv. power boost, který umožňuje výrazně navýšit výstupní výkon a zvýšit tak dosah. Další typ zesilovače se označuje jako linkový zesilovač (in-line). Je umístěn na trase a zesiluje přijatý signál. Těchto zesilovačů může být i několik za sebou, dokud nedochází k omezení vlivem např. disperze. Posledním zesilovačem, který je tentokrát umístěn před přijímačem, se nazývá předzesilovač. Umožňuje zesílit slabý vstupní signál pro další zpracování.[4]

Kromě níže nastíněných zesilovačů existují například i tzv. SOA (Semiconductor Optical Amplifier) zesilovače založené na polovodičové technologii.

Zesilovače s vlákny dopovanými vzácnými zeminami

Na obrázku 1.1 je schéma tzv. EDFA (Erbium doped fiber amplifier) zesilovače. Princip fungování bude znázorněn na tomto zesilovači. Podobně fungují i ostatní zesilovače s dopováním vzácnými zeminami. Tento zesilovač pomocí dopování energie do erbiového vlákna zesiluje signál. Dopování probíhá pomocí dvou diod s vlnovou délkou 980 nm, kdy dojde k excitaci iontů erbia a při návratu iontů na původní energetickou hladinu dojde k zesílení procházejícího signálu.[5]



Obr. 1.1: Schéma jednoduchého EDFA zesilovače [5]

Ramanovské zesilovače

Tyto zesilovače využívají ke své funkci tzv. Ramanův rozptyl. Podstatou je interakce mezi světlem procházejícím materiálem a tímto prostředím. Následkem toho dojde k frekvenčnímu posuvu světla.

Samotný princip fungování spočívá ve stimulovaném Ramanově rozptylu v optickém vlákně. Pokud jsou vhodně zvoleny frekvence budících prvků, lze tohoto rozptylu dosáhnout a díky přesunu energie dojde ke generování tzv. postranního vidu.

Tento rozptyl existuje v každém optickém vlákně a jeho širokopásmovost je v tomto výhodou.[6]

Rozeznáváme dva typy práce Ramanovského zesilovače:

Rozprostřený zesilovač Zde se na zesílení signálu podílí celá optická trasa. Je buzen z opačného konce trasy, a vyrovnává tak ztráty na koncové části šíření. Výhodou tohoto režimu je nižší šum.

Diskrétní zesilovač Tento režim je konstruován jako jeden blok, který je umístěn trase. Má definovanou šířku pásma a zisk na určitých vlnových délkách.

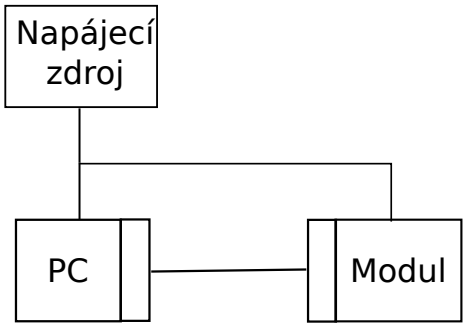
2 Obecná struktura telekomunikačního zařízení

Obecně je složení telekomunikačního zařízení uvedeno na obrázku 2.1. Skládá se z napájecího zdroje, počítače a například optického modulu.

Jako zdroj napájení je vhodné volit redundantní zdroj s co největší účinností. Pokud by jedna napájecí větev selhala, je možné zařízení plně napájet z druhé větve.

Počítač je obvykle základní deska s libovolným procesorem a tento blok řídí například připojený optický modul, umožní nastavení parametru na modulu a podobně. Nemusí být příliš výkonný, neboť zde obvykle neprobíhají náročné operace. Rozhraní na počítači umožňuje zprostředkovat komunikaci s modulem nebo například připojení k internetu.

Modul je v tomto případě část, která je za použití některého komunikačního rozhraní připojena k počítači a ten jej řídí. Rozhraní je obvykle RS232, RS485, případně i další.



Obr. 2.1: Blokové schéma telekomunikačního zařízení

2.1 Napájecí zdroje

Napájecích zdrojů existuje na trhu spousta. Obvykle se rozdělují na dva základní typy, a to lineární a spínané zdroje. Jejich rozdíl spočívá hlavně ve způsobu zatěžování regulačního členu. U spínacích zdrojů je tento prvek zatěžován nikoliv kontinuálně, ale impulsně, což umožňuje odebírat vyšší výkon z takto zatěžovaného prvku než při kontinuální zátěži.

Výhodou spínaných zdrojů je vyšší účinnost a malé rozměry. Jsou ovšem vhodné pro velký rozdíl mezi vstupním a výstupním napětím s použitím transformátoru. Dále jsou vhodné spíše pro napájení digitálních obvodů, jelikož mohou generovat vysokofrekvenční rušení, které není jednoduché odfiltrovat a mohlo by působit problémy v analogových obvodech.

Spínané zdroje lze konstruovat jako zvyšující (step-up) nebo snižující (step down). U těchto zdrojů je také nutné dbát na riziko rušení sítě a zajistit minimalizaci pronikání vyšších harmonických složek do sítě, která zdroj napájí.

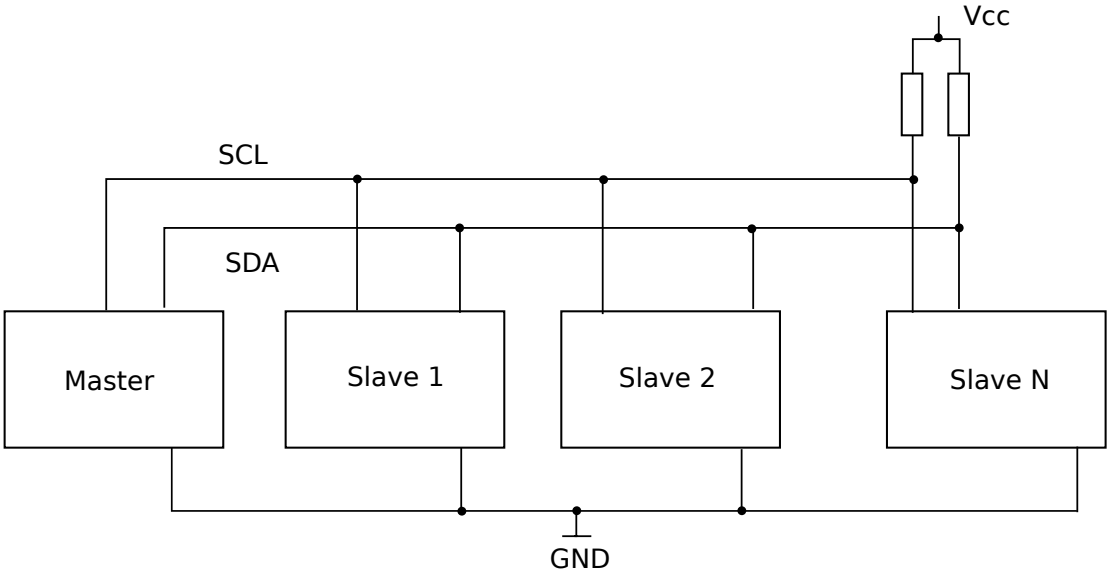
Druhým zástupcem jsou lineární zdroje, které jsou jednodušší na konstrukci. Mají ovšem nižší účinnost a pro větší výkony bývají i rozměrné. Tyto zdroje lze konstruovat pouze jako snižující. Tyto zdroje jsou vhodné pro napájení analogových částí obvodů, které by mohly být ovlivněny spínáním u spínacích zdrojů.[7]

2.2 Rozhraní

2.2.1 I2C

Sběrnice I2C (Internal-Integrated Circuit) je určena pro komunikaci mezi integrovanými obvody. V dnešní době lze připojovat pomocí této sběrnice například LCD, paměti, mikrokontroléry a další digitálně řízené obvody.

Jedná se o sběrnici, která využívá pro komunikaci pouze dva vodiče. Vodič SDA (Serial Data) je používán pro přenos dat a vodič SCL (Serial Clock), po němž se přenáší signál pro synchronizaci komunikujících stran (synchronní přenos dat). Toto provedení optimalizuje počet vstupně výstupních pinů v obvodech, a lze tak zjednodušit zapojení.[8, 9]



Obr. 2.2: Připojení zařízení na sběrnici I2C [9]

Na sběrnici se využívá rozlišení, kdy jedno zařízení je nastaveno jako master a ostatní jako slave. Toto umožňuje připojení více zařízení na jednu I2C sběrnici. Zařízení se následně rozeznávají podle nastavené adresy, která může mít 7 bitů nebo

10 bitů. Teoreticky lze využít 128 nebo 1024 adres. V praxi jsou některé adresy rezervovány, například pro oslovení všech zařízení na sběrnici. Problematická bývají i zařízení, u kterých nelze změnit adresu. Z důvodu jednoznačné adresace musí být adresa na sběrnici unikátní tzn. toto zařízení může být na sběrnici pouze jednou.

Oba vodiče jsou v klidu na logické úrovni 1, což zajišťují pull-up rezistory na těchto vodičích. Hodnota těchto odporů je určena rychlostí sběrnice a pohybuje se v řádech kiloohmů. Rychlost přenosu je určena dle nejpomalejšího připojeného zařízení. Přenosová rychlost může nabývat hodnot 10, 100, 400 kbps, 1 Mbps a 3,4 Mbps.

Hodinový signál pro synchronizaci generuje zařízení nastavené jako master na vodiči SCL. Data jsou přijímána všemi zařízeními na sběrnici a podle adresy si určují, zda je zpráva určena jim, či nikoliv (broadcast). Adresa je vysílána před daty.[8, 9]

Přenos je složen z následujících částí:

Klidový stav Nejsou vysílána žádná data na sběrnici a negeneruje se hodinový signál.

Začátek přenosu nebo další části Vznikne přechodem logické úrovně na vodiči SDA z 1 na 0. Hodnota na SCL zůstává v logické jedničce.

Přenos Samotný přenos dat probíhá po vodiči SDA v sérii 8 bitů (1 Byte) od nejvyššího bitu po nejnižší. Data se přenáší pouze, pokud je SCL v logické 0, jen tehdy lze změnit hodnotu na vodiči SDA.

Potvrzující bit Provede potvrzení přijmu dat a jde také o informaci protistraně, že je možné pokračovat v přenosu. Logická 0 odpovídá přijetí v pořádku, logická 1 značí selhání. Pokud není odesláno nic, znamená to konec přenosu. Protistrana následně zašle potvrzující bit a stop bit.

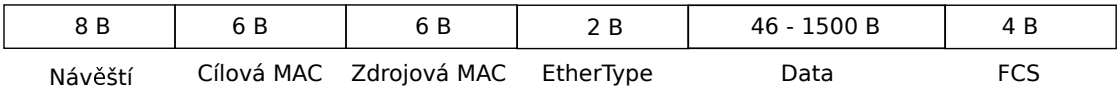
2.2.2 Ethernet

Ethernet je definován ve standardu IEEE (Institute of Electrical and Electronics Engineering) 802.3.[10] Dokument je velmi rozsáhlý a obsahuje většinu standardizovaných variant pro různé rychlosti a přenosová média. Přenos může probíhat po koaxiálním kabelu, kroucené dvoulince nebo optickém kabelu.[11]

Ethernet má vlastní adresaci a funguje na druhé vrstvě modelu ISO/OSI. Adresace probíhá pomocí MAC (Media Access Control) adres, které jsou napevno zakódovány do síťové karty a jejíž délka je 48 bitů. MAC adresa je rozdělena na dvě části, kde 24 bitů identifikuje výrobce příslušné karty a druhých 24 bitů by mělo být jedinečných pro každou vyrobenou kartu.[12] MAC adresu v zařízení není možné změnit, ale lze používat jinou MAC za použití softwaru.

Existuje několik typů rámců pro Ethernet. Dnes asi nejpoužívanějším typem je rámec Ethernet verze 2. Jeho velikost se pohybuje od 64 do 1518 Bytů. Jeho

struktura je na obrázku 2.3.[13]



Obr. 2.3: Rámec Ethernet verze 2 [13]

Rámec, kromě výše uvedených MAC adres, obsahuje ještě další pole:[13]

Návěští definuje začátek rámce.

EtherType toto pole udává protokol vyšší vrstvy u rámce Ethernet verze 2. Pokud je přenášen Ethernetový rámec dle 802.3, je v tomto poli délka datového pole. Pokud jde o rámec dle 802.3, pak je hodnota v tomto poli menší než 1536. Pokud je vyšší nebo rovna, jedná se o rámec Ethernet verze 2.

Data obsahuje přenášená data. délka 46 – 1500 Bytů

FCS toto čtyřbytové pole obsahuje kontrolu bezchybného přenosu rámce pomocí CRC kontrolního součtu nad daty.

V dnešní době se lze nejčastěji setkat s kroucenou dvoulinkou nebo optickým kabelem.[10, 11] Kroucená dvoulinka musí splňovat různé parametry a byl definován konektor 8P8C označovaný RJ45, který má stanovené zapojení dle dvou barevných standardů 568A a 568B a záleží na každém, který použije. Zapojení konektoru je v tabulce 2.1. Pokud je kabel na jedné straně nakrimpován podle 568A a na druhé 568B, pak vzniká tzv. křížený kabel.[14]

Tab. 2.1: Zapojení konektoru dle barevných standardů 568A a 568B[14]

Pin	**568A**	**568B**
1	bílá/zelená	bílá/oranžová
2	zelená	oranžová
3	bílá/oranžová	bílá/zelená
4	modrá	modrá
5	bílá/modrá	bílá/modrá
6	oranžová	zelená
7	bílá/hnědá	bílá/hnědá
8	hnědá	hnědá

Ethernet je dnes definován v mnoha rychlostech a stále se vyvíjí nové.

2.2.3 RS 232

Toto standardizované rozhraní je používáno pro komunikaci s počítači a zařízeními. Lze takto propojit dvě tato zařízení a například provést z PC konfiguraci připojeného

zařízení.

Komunikace na tomto rozhraní probíhá za použití jediného vodiče pro každý směr komunikace. V základu jsou používány vodiče pro vysílání TxD, příjem RxD a uzemnění značené GND. Avšak mohou být doplněny ještě dalšími vodiči, například vodiči pro řízení přenosu RTS, CTS.[15] Zakončení je obvykle pomocí konektorů D-Sub typu CANON-9 nebo CANON-25. Zapojení konektoru CANON-9 male je v tabulce 2.2.

Tab. 2.2: Zapojení konektoru CANON-9 male [15]

Pin	**Označení**	**Popis**
1	CD	Carrier Detect
2	RxD	Receive
3	TxD	Transmit
4	DTR	Data Terminal Ready
5	GND	Ground
6	DSR	Data Set Ready
7	RTS	Request to Send
8	CTS	Clear to Send
9	RI	Ring Indicator

Při přenosu je nutné, aby vysílač a přijímač v případě asynchronního přenášení dat pracovaly se stejnou hodnotou frekvence. Každá sekvence dat je předcházena start bitem, který linku přepne do patřičného stavu. Poté se přenesou data zabezpečená paritou a jeden nebo několik stop bitů. Následně je linka opět v klidovém stavu až do příchodu dalšího start bitu.

Pro korektní příjem je třeba nastavit:

- Baudrate – jde o rychlost přenosu dat po lince,
- počet datových bitů – kolik dat bude přeneseno,
- počet stop bitů – obvykle jeden nebo dva bity, které ukončují přenos,
- kontrolu toku dat.

Logické stavy mohou být reprezentovány ±5 V, ±10 Vnebo ±15 V. V dnešní době se lze setkat i s nižšími hodnotami.[15]

2.3 GNU/Linux

GNU/Linux je svobodný operační systém se svobodnou licencí GNU GPL, jehož autorem je Linus Torvalds a mnoho dalších programátorů díky licencování. Systém se skládá z jádra a aplikací obvykle z rodiny GNU. Označení pouze slovem Linux

ovšem není přesné, jelikož se může jednat o celý systém nebo pouze o jádro. Formálně správně je tedy právě GNU/Linux, ovšem v práci bude pro zjednodušení používáno pouze označení Linux.[16]

Linux není pouze jeden. Existují různé systémy označované jako distribuce Linuxu, kde každá má svoji politiku pro licence programů, které obsahuje, nebo se liší v distribuci programů apod.[16]

2.3.1 Jádro

Jde o základní program v počítači, který se spustí jako první a zajišťuje inicializaci hardwaru. Následně jsou spouštěny další aplikace.

V Linuxu se používá tzv. monolitické jádro, které vznikne kompilací všech zdrojových kódů procedur a všeho, co je dále potřeba do jednoho spustitelného souboru. Každá funkce je tedy viditelná pro ostatní a je možné je takto i přímo volat. Výhodou tohoto modelu je rychlost. Pokud ovšem někde v jádře dojde k chybě (například na ovladači síťové karty), může dojít k pádu celého systému. V případě přidání nějaké další funkce nebo ovladače je nutné celé jádro znovu zkompilovat.

Variantou je vytvoření modulárního jádra, které má v základu jen to nejnutnější pro běh. Ostatní části zvané moduly jsou následně nahrávány za běhu systému dynamicky do paměti, pokud jsou potřeba. V tomto případě není nutné pro přidání funkce kompilovat celé jádro, ale lze tuto funkci zkompilovat jako modul, který se bude načítat. Nejčastěji jsou tímto způsobem načítány moduly pro další souborové systémy.[17]

2.3.2 Příklad distribucí GNU/Linuxu

Zde je uvedeno několik distribucí pro ukázku náročnosti na systémové prostředky a některých dalších rozdílů mezi nimi.

Debian

Tuto distribuci tvoří sdružení jednotlivců, kteří si dali za cíl vytvořit svobodný operační systém.

V této distribuci lze nalézt přes 50 000 předkompilovaných balíčků, které lze jednoduše instalovat z repositáře. Balíčky mají příponu .deb. Debian udržuje několik verzí (stable, testing, unstable).

Distribuce podporuje celkem deset architektur procesoru, mezi nimiž lze nalézt například architekturu pro 32 a 64 bitové PC (AMD a Intel) nebo architekturu ARM (32 i 64 bit). Pro systém ve verzi stable jsou doporučené minimální požadavky rozděleny na systémy bez desktopového prostředí (512 MB RAM a 2 GB místa na

disku) a s desktopovým prostředím (1 GB RAM a 10 GB místa na disku). Minimální procesor pro instalaci na 64 bitové PC je Pentium 4 s taktovacím kmitočtem 1 GHz.[18]

Ubuntu

Za touto distribucí stojí společnost Canonical a je odvozena od Debianu. Jde o kompletní desktopový operační systém, který má i serverovou verzi. Je k dispozici jak podpora komunity, tak profesionální podpora. Používá stejný systém pro balíčky softwaru a má stejné minimální požadavky na systém. Je vydávána každé dva roky ve verzi LTS (Long Term Support) s pětiletou podporou, nebo jsou vydávány verze každých šest měsíců s podporou 18 měsíců.[19]

Gentoo Linux

Je také vyvíjena komunitou podobně jako Debian, ale je založena na kompilaci softwaru ze zdrojových kódů, což umožňuje správci vytvořit si systém na míru svým potřebám. Distribuce nemá verze jako například Debian nebo Ubuntu, ale jde o systém tzv. rolling-release (průběžné aktualizace). Dále je zde možné pomocí tzv. USE flagů určit například závislosti, a tím snížit množství závislostí v systému.

Minimální doporučené požadavky jsou 2,75 GB místa na disku pro základní instalaci, 512 MB RAM. Pro instalaci bez grafického prostředí je uváděno 10 GB místa na disku nebo méně (podle velikosti programů). Obsahuje podporu pro 11 architektur procesoru.[20]

2.3.3 Firewall

Systémy v dnešní době komunikují pomocí počítačových sítí. V této komunikaci se používají datové bloky označované jako pakety. Paket obsahuje záhlaví, kde jsou uvedeny například údaje, kdo jej vytvořil a komu má být doručen. A také přenášená data. Pakety mohou mít proměnnou velikost oproti například buňkám.

Komunikaci mezi jednotlivými sítěmi je nutné monitorovat a případně provádět filtraci provozu za účelem ochrany před útokem nebo ochranou dat uvnitř sítě nebo jednotlivých strojů.

Zařízení, která toto umožňují, jsou označována jako firewally. Ty mohou být hardwarové nebo softwarové. Softwarové jsou obvykle používány k ochraně jednotlivých koncových strojů, hardwarové jsou nejčastěji použity v síti na trase mezi koncovými stroji a slouží k ochraně několika strojů zároveň.[21, 22]

Detekci a opatření je možné provádět na kterékoliv vrstvě ISO/OSI modelu, avšak nejčastěji jsou v praxi používány následující firewally [21, 22]:

- paketový firewall je definován na sítové vrstvě,
- paketový firewall se stavovou inspekcí na transportní vrstvě,
- proxy firewall nacházející se na aplikační vrstvě.

Paketový filtr provádí filtraci na sítové vrstvě na základě zdrojové a cílové adresy a portů. Tyto adresy jsou součástí záhlaví IPv4 (Internet Protocol version 4) a IPv6 (Internet Protocol version 6) paketu, porty jsou pak součástí záhlaví transportní vrstvy.

Filtry se stavovou inspekcí používají navíc i informaci o stavu spojení, které vytváří například protokol TCP (Transmission Control Protocol). Spojení se může nacházet ve stavech [23]:

NEW paket nepatří k žádnému z dosavadních spojení, což může být paket s TCP příznakem SYN, který značí začátek nového spojení.

ESTABLISHED umožní komunikovat v již navázaném spojení.

RELATED umožňuje vytvoření dalších spojení za použití konexe pro již navázané spojení.

INVALID paket nemůže být identifikován, je vadný, případně není známa konexe, ke které patří.

Tento typ firewallu filtruje pakety také podle informací v záhlaví a navíc umožňuje kontrolovat, v jakém stavu je spojení, avšak ignoruje data, která nese.

Kontrola dat, která jsou nesena paketem, je prováděna tzv. proxy firewallem umístěném na aplikační vrstvě. Data mohou obsahovat například nevalidní data, která mohou způsobit pád aplikace. Proxy firewall tato data analyzuje a dokáže posoudit, zda neobsahují chyby nebo škodlivý kód.

Každý z uvedených způsobů má své výhody a nevýhody a obvyklé umístění v síti. Paketové a proxy firewally disponují velkou propustností, ale lze provést útok pomocí vyšších vrstev. Obvykle se umisťují na hranici sítě. Toto lze eliminovat použitím proxy firewallu, který nedisponuje velkou propustností, neboť provádí kontrolu dat. Umístění je až za předchozími a obvykle je těchto firewallů několik pro různé služby z důvodu zajištění propustnosti.

Iptables

Tento nástroj je spjat s firewallem v jádrech Linux ve verzi 2.4.x a novějších. Mezi hlavní vlastnosti patří [24]:

- Filtrování provozu na IPv4 a IPv6,
- překlad adres a portů,
- nastavení QoS pro odchozí provoz a značkování paketů pro další zpracování.

Jde o strukturu tabulek obsahující pravidla pro zacházení s pakety, které dorazí. Tyto tabulky jsou umístěny na různých místech, kterými paket prochází. Každá

tabulka se prochází sekvenčně od shora dolů. Pokud paket vyhovuje pravidlu v tabulce, provede se příslušná akce, jež je uvedena u splněného pravidla. Co například může nastat, pokud paket vyhovuje některému pravidlu? Jedná se o tzv. targety.

ACCEPT přijme paket.

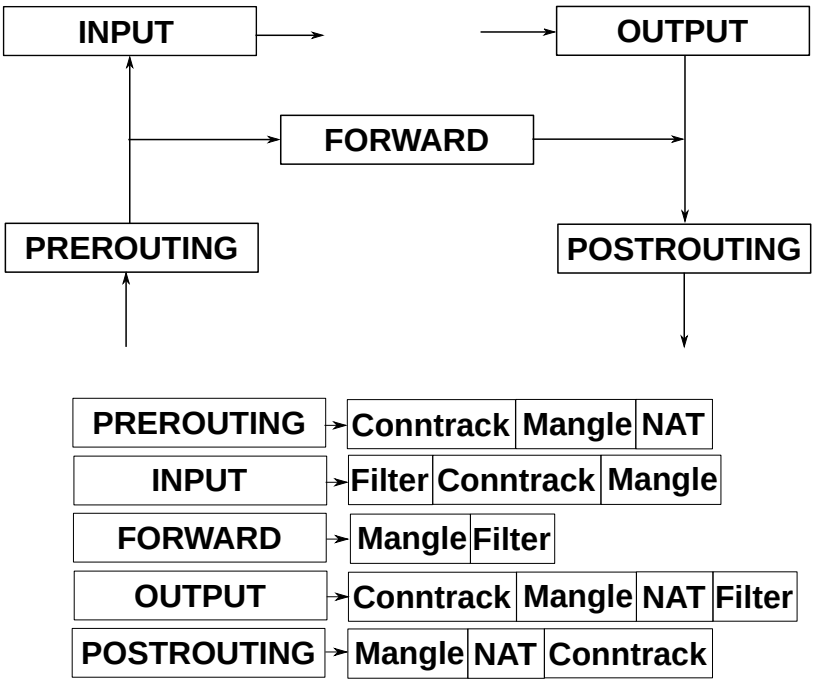
DROP paket zahodí.

RETURN vrátí paket do předchozího řetězce, odkud byl vložen.

QUEUE propustí paket z kernelu do uživatelského prostoru.

LOG provede záznam informací u paketu do logu. (Nemusí být vždy k dispozici.)

REJECT odmítne paket a zašle zpět zprávu ICMP (Internet Control Message Protocol), kterou definujeme. (Nemusí být vždy dostupné.)



Obr. 2.4: Tabulky v iptables [24]

Názvy jednotlivých souborů tabulek určené pro filtraci a význam:

- INPUT – přicházející pakety určené pro tento stroj,
- OUTPUT – pakety, které stroj odesílá,
- FORWARD – pakety, které mají být směrovány do jiné sítě.

Příkazy pro konfiguraci firewallu

Konfigurace firewallu se provádí pomocí příkazů ve formátu (ne všechny části jsou nutné vždy):

```
iptables <tabulka> <akce> <definice pravidla> <target>
<informace doplnující pro target>
```

Tento příkaz existuje i ve verzi pro IPv6 síť, jde o `ip6tables` Níže je uveden příklad akcí, které lze provést. Nejedná se o kompletní výpis.[25]

-A přidá nové pravidlo na konec tabulky.

-I vloží pravidlo na začátek tabulky.

-D smaže pravidlo. Lze jej definovat buď zadáním celého pravidla, nebo pomocí indexu v tabulce.

-P definuje výchozí politiku tabulky.

-L provede výpis pravidel v tabulce. Pokud se tabulka nezadá, vypíší se všechny včetně pravidel.

Pro definici pravidel slouží následující parametry, které definují podmínky, které musí paket při průchodu splnit, aby se pravidlo uplatnilo. Opět se nejedná o všechny. Další možnosti lze nalézt v dokumentaci k iptables/ip6tables. Jsou uvedeny zkrácené verze parametrů, ale je možné používat i dlouhé názvy. Negace se provede přidáním `!` před argument parametru.[25]

Pro definici protokolu je používáno `-p`. Definice zdrojové resp. cílové adresy se provede pomocí `-s` resp. `-d`. Pro omezení provozu dle vstupního nebo výstupního rozhraní je použito `-i`. Parametr `-j` určuje, co se má s paketem stát. Pro názornost je uvedeno jedno pravidlo níže.

```
iptables -A INPUT -p tcp -s !192.168.1.0/24  -j DROP
```

Příkaz přidá do tabulky INPUT záznam s pravidlem, které definuje, že paket protokolu TCP z jiného adresního rozsahu než je uvedeno bude zahozen.

2.3.4 Uživatelé a skupiny

Každý linuxový systém musí obsahovat minimálně jednoho uživatele zvaného root s UID (User Identifier). Tento uživatel má možnost přímo ovlivnit nastavení systému a má neomezená práva. Je proto vhodné vytvářet další uživatele například pro běžnou práci.[26]

Každý uživatel je součástí nějaké výchozí skupiny, která mu byla na začátku přiřazena. Uživatel však může být součástí i dalších skupin. Informace o skupinách jsou uloženy v souboru `/etc/group` a každá skupina má unikátní identifikátor označovaný GID (Group Identifier).

Údaje o uživatelích a skupinách následně umožňují nastavení práv pro přístup k souborům (práva pro vlastníka, skupinu a ostatní) v souborovém systému. To může být vhodné například pro skrytí obsahu souboru s hesly uživatelů, resp. pouze uživatel root bude moci do souboru zapsat a číst apod.[26]

Informace o všech lokálních uživatelích jsou uloženy v souboru `/etc/passwd`, jejich hesla jsou pak obvykle v hashované podobě v `/etc/shadow`. Pro každého uživatele v souboru `passwd` je definováno několik údajů oddělených dvojtečkou. Jedná se o [26]:

- Uživatelské jméno,
- heslo (případně písmeno x signalizující, že heslo je uloženo v `/etc/shadow`),
- UID,
- GID,
- komentář sloužící pro popis účtu,
- cesta k domovské složce,
- výchozí uživatelský shell.

Příklad struktury pro jednoho uživatele ze souboru `/etc/passwd`.

```
root:x:0:0:root:/root:/bin/bash
```

Podobné rozložení se nachází i v souboru skupin. Samozřejmě s jinými údaji. Konkrétně jde o název skupiny, heslo, GID a uživatelé přiřazené do skupiny.

Uživatele i skupiny jde přidávat pomocí příkazů `useradd`, `groupadd` a přidání uživatele do skupiny lze provést příkazem `gpasswd`.[26]

Hesla

Z důvodu zvýšení bezpečnosti jsou prováděna opatření, která mají za cíl bránit používání jednoduchých hesel typu „1234“ a podobně. Jde například o vyžadování minimální délky hesla, minimální počet jednotlivých skupin znaků v hesle (malá a velká písmena, číslice, speciální znaky).

K tomuto lze využít autentizační knihovnu v Linuxu označovanou PAM (Pluggable Authentication Modules). Zde se v souboru `/etc/pam.d/passwd` definují minimální požadavky na hesla (vždy jde o `proměnná=hodnota`). Jde o [27]:

minlen udává minimální délku hesla. Neumožní tedy heslo, které je kratší.

dcredit označuje minimální počet číslic.

lcredit nejméně udaný počet malých písmen (obdobně ucredit pro velká písmena).

ocredit označuje speciální znaky.

difok udává, kolik znaků musí být jiných než v původním hesle.

Je také vhodné, aby uživatel změnil své heslo po určité době. Toho lze dosáhnout nastavením expirace hesla, což lze provést v souboru `/etc/login.defs` pomocí následujících voleb. [28]

PASS_MAX_DAYS udává počet dní pro platnost hesla. Po uplynutí této doby bude vyžadována jeho změna.

PASS_MIN_DAYS udává minimální interval ve dnech mezi změnami hesla.

PASS_WARN_AGE označuje počet dní před ukončením platnosti hesla, kdy bude zobrazováno upozornění na změnu hesla.

Je též vhodné provést omezení na počet přihlášení pro uživatele. To je možné provést použitím příkazu `faillog` například takto:

```
faillog -m 3 -u <uživatel>
```

Dojde k omezení počtu chybných přihlášení na tři pro zvoleného uživatele.[29]

2.3.5 SSH

SSH (Sechure Shell) je protokol pro vzdálené připojení k serveru na TCP portu 22. Během připojení je veškerá komunikace šifrována, což neplatí například u protokolu Telnet.

Pomocí Diffie–Hellmannova protokolu jsou dohodnuty klíče mezi oběma stranami. Následně je ověřena totožnost serveru pomocí zaslaného otisku klientem a poté je i klient ověřen serverem (dle zvolené metody).[30] Pokud vše proběhne v pořádku, je možné na serveru pracovat.

SSH se v Linuxu nejčastěji objevuje v implementaci OpenSSH. Některé možnosti jsou uvedeny v následujícím seznamu.[31]

- Silná kryptografie v podobě šifrování pomocí například AES, RSA, ECDSA.
- Přesměrování portů. Umožní například zabezpečení protokolů, které nemají šifrování.
- Použití asymetrické kryptografie pro autentizaci.
- Připojení souborového systému na vzdáleném stroji.

Diffie-Hellman

Jedná se o protokol, který umožňuje výměnu soukromého klíče po nezabezpečené síti. Je založen na principu asymetrického šifrování.

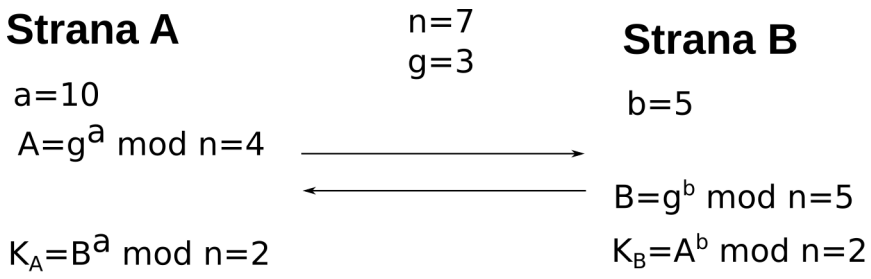
Vytvořen byl v roce 1976 a je pojmenován po svých autorech, kterými jsou W. Diffie a M. Hellman.

Bezpečnost tohoto protokolu je dána využitím problému diskrétního logaritmu, kdy jeho výpočet je relativně jednoduchý, avšak inverzní operace je složitá a časově náročná. Diskrétní logaritmus je založen na cyklických grupách.[32]

Protokol lze použít pro vyjednání bezpečného klíče mezi komunikujícími stranami, ale nelze použít k šifrování a dešifrování zpráv.

Pro generování klíče je znám modul (prvočíslo) a generátor grupy nebo podgrupy (přirozené číslo). Každá komunikující strana si zvolí velké náhodné přirozené číslo a pomocí těchto hodnot vypočítá číslo, které zašle protistraně. Ta po diskrétním

umocnění tohoto čísla svým přirozeným číslem získá tajný klíč, který bude používat k šifrování při komunikaci s protistranou.[32] Znázornění je na obrázku 2.5.



Obr. 2.5: Dohodnutí klíče pomocí Diffie-Hellmanova protokolu [32]

Případný útočník nezná zvolená čísla, jelikož tato čísla neopustí počítač. Pokud by ovšem dokázal zjistit toto číslo pro jednu ze stran, mohl by následně komunikaci odposlouchávat.

Protokol není odolný proti útoku mužem uprostřed, kdy se útočník vydává za protistranu. V tomto případě si útočník vyjedná klíč s oběma stranami a poté probíhá komunikace přes něj.

V SSH je problém muže uprostřed vyřešen pomocí ověření protistrany, kdy je nutné potvrdit klientem, že udaný otisk serveru odpovídá serveru, na který se požaduje klient připojit. A klient se k serveru hlásí za použití nějaké autentizační metody.

Zabezpečení serveru SSH

SSH server v Linuxu obsahuje mnoho voleb pro zvýšení bezpečnosti, určení, kdo a jak se smí připojit k serveru a podobně. Tato konfigurace je uložena v souboru `/etc/ssh/sshd_config`. Některé konfigurační volby jsou uvedeny níže.[28]

PermitEmptyPasswords udává, zda je možné přihlášení pro uživatele bez hesla (yes) nebo nikoliv (no). Výchozí hodnota neumožní přihlášení bez hesla.

PubkeyAuthentication umožní použití kombinace uživatelské jméno/klíč.

PasswordAuthentication definuje, zda je možné se přihlásit heslem nebo nikoliv.

PermitRootLogin povolí nebo zakáže vzdálené přihlášení na uživatele root. Existuje i hodnota `prohbit-password`, která umožní přihlášení na účet root jen za použití klíče a `force-command-only`, která umožní přístup na účet root za použití klíče, ale jen při specifikaci příkazu, který se má provést.

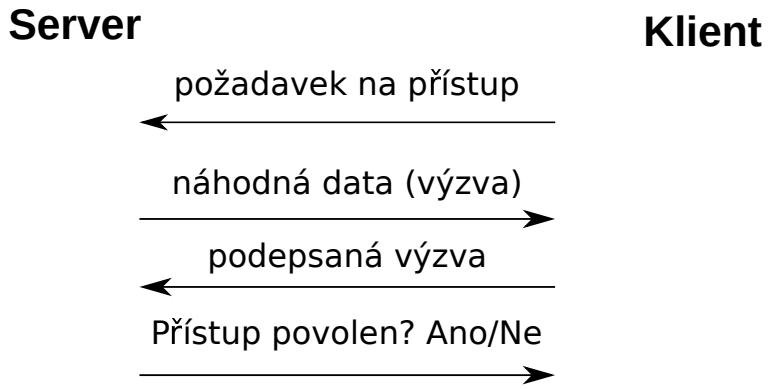
AllowUsers, DenyUsers tyto volby povolí nebo zabrání přihlášení zadaným uživatelům. Podobné volby lze použít i na skupinu `AllowGroups`, `DenyGroups`.

Server v Linuxu obsahuje mnoho možností, které umožní omezit uživatele i po přihlášení na server, například jen na použití určitých příkazu atd.

Použití klíčů

Použití klíčů je založeno na použití asymetrické kryptografie, kdy existují dva klíče. Jeden je veřejný a slouží k šifrování dat nebo ověření podpisu a druhý je privátní (soukromý) a umožňuje dešifrovat data nebo vytvořit podpis. Soukromý klíč je nutné střežit, aby nedošlo k jeho odcizení a případnému zneužití. Veřejný klíč se poté nahraje na server, kam se budeme připojovat.

Při ověřování identity se používá podepisování náhodné sekvence dat (výzvy), kterou zašle server, soukromým klíčem na disku klienta. Server po přijetí sekvence a podpisu tento podpis ověří na základě uloženého veřejného klíče a povolí přístup nebo jej zamítne.[33] Znázornění je uvedeno na obrázku 2.6.



Obr. 2.6: SSH autentizace klíčem [33]

Soukromý klíč se nepřenáší přes síť, a není tak možné jej například kompromitovaným serverem zjistit. Veřejný klíč je možné nahrát na několik serverů a následně používat jeden soukromý klíč k přístupu na tyto servery.[33]

V systému Linux existuje snadný způsob, jak vygenerovat tento pár klíčů. Pro OpenSSH existuje následující příkaz:

```
ssh-keygen -t rsa
```

Tento příkaz vygeneruje dvojici klíčů s asymetrickou šifrou RSA (zvoleno parametrem `-t rsa`). Dojde k dotazu na heslo k soukromému klíči. Pokud nedošlo ke změně cesty pro uložení klíčů, jsou soubory uloženy v domovském adresáři uživatele v `.ssh/id_rsa` (soukromý klíč) a `.ssh/id_rsa.pub` (veřejný klíč). Nyní je nutné vložit veřejný klíč na server a následně jej umístit do `ssh/authorized_keys`, což lze provést příkazem:[33]

```
cat id_rsa.pub >> .ssh/authorized_keys
```

2.3.6 Soubory a adresáře v GNU/Linux

V operačním systému Linux je vše reprezentováno souborem. Existuje jeden adresářový strom, který začíná tzv. kořenovým adresářem (`/`) a následně se větví do dalších složek obsahujících konfigurace služeb, dostupná zařízení, dočasné soubory, uživatelská data atd. Veškeré diskové oddíly se připojují do této struktury. Pokud název adresáře nebo souboru začíná tečkou, pak se jedná o tzv. skrytý adresář nebo soubor, který se nezobrazí ve výpisech, pokud není zadán parametr, který tyto skryté položky zobrazí.

Strukturu adresářů definuje FHS (Filesystem Hierarchy Standard)[34]. Tento standard udává, kde se nachází jaká data. Co lze nalézt ve vybraných adresářích, je uvedeno níže.

/bin soubory, které lze spustit všemi uživateli v systému.
/boot jádro systému, soubory zavaděče a další pomocné soubory.
/dev soubory definující fyzická a logická zařízení.
/etc konfigurace služeb (globální).
/home domovské adresáře uživatelů.
/lib sdílené knihovny a moduly jádra.
/media prostor pro připojení například USB flash disku.
/proc stav systému a procesů.
/root domovský adresář uživatele root.
/sbin systémové programy.
/tmp dočasné soubory.
/usr sekundární adresářová struktura. Zde lze nalézt například zdrojové kódy.

Práva

Každý soubor a adresář nacházející se ve stromové struktuře musí mít definovaného vlastníka a skupinu. Každý soubor a adresář má uvedeny tři kategorie práv: pro vlastníka, skupinu a ostatní. Tyto tři kategorie se obvykle označují pomocí písmen: user (`u`), group (`g`) a other (`o`).

Základní práva pro přístup k souborům jsou čtení označované `r`, povolení k zápisu `w` a spuštění `x`. Pokud je soubor možné spustit, pak je obvykle spuštěn s právy uživatele, který tak učinil. Existují dvě výjimky, které toto chování mění. Pokud je nastaven setuid bit (pouze pro vlastníka), pak je soubor spuštěn s právy uživatele, který soubor vlastní. Obdobně se chová i nastavení bitu setgid, který spustí soubor s právy skupiny, která je u souboru uvedena.[35, 36]

Práva pro adresář jsou nezávislá na právech pro soubory a adresáře pod ním. Právo číst `r` povoluje zobrazit obsah adresáře. Právo zapsat `w` umožní provádět změny v adresáři a oprávnění pro přístup `x` znamená, že lze procházet adresářem.

Pokud je nastaveno právo pro přístup, ale není zároveň nastaveno právo pro čtení, musí uživatel znát jméno souboru, který chce zobrazit, jelikož zobrazení obsahu adresáře není povoleno.[35, 36]

2.3.7 Sudo

Tento nástroj slouží administrátorovi k přidělení určitých oprávnění jinému neprivilegovanému uživateli nebo skupině. V závislosti na konfiguraci je následně možné spouštět příkazy jako jiný uživatel. Každé úspěšné i neúspěšné použití se zaznamená.

Sudo je alternativou k nástroji `su` pro spouštění příkazů jako jiný uživatel (například root). V tomto případě však nedojde ke spuštění root shellu (v případě uživatele root), ale pouze k dočasné eskalaci práv. To může být výhodné v případě chyby a možnosti zničit systém. Použití tohoto příkazu je následovné:

```
sudo prikaz
```

Pokud má být tento příkaz použit, musí být nejdříve pro daného uživatele povolen. To lze provést v souboru `/etc/sudoers`. Uvedený soubor lze přímo editovat běžným textovým editorem. Je ovšem doporučeno použití následujícího příkazu pod uživatelem root.

```
visudo
```

Toto zabrání v editaci několika osobám současně a zároveň provede po uložení základní kontrolu syntaxe, aby nedošlo k zablokování přístupu. Pokud je požadavek na změnu editoru na jiný, použije se místo předchozího.[37, 38, 39]

```
EDITOR=<nazev-editoru> visudo
```

V souboru lze použít velké množství konfiguračních voleb. Základní syntaxe je ve tvaru:

```
login-user host = (user-as-run-command) command
```

Pokud má být příkaz spuštěn pouze jako uživatel root, pak lze závorku vynechat. Definice příkazu musí být definována plnou cestou, nestačí například `bash`, ale je nutné uvést `/bin/bash`. Pokud má být uvedeno několik příkazů, je to možné provést použitím desetinné čárky jako oddělovače příkazů.

V části `login-user` lze použít i název skupiny. Tu je nutné definovat s použitím znaku `%`. Pro definování skupiny `wheel` se tedy uvede `%wheel`.[37, 38, 39]

V konfiguraci lze také použít aliasy, které ji mohou zpřehlednit. Je také vhodné používat velká písmena pro aliasy. Jde o následující volby:

User_Alias slouží k nastavení aliasů pro několik uživatelů.
Cmnd_Alias uvozuje alias pro příkazy.
Host_Alias určuje alias pro hosty.

Nastavení dle výše uvedeného je pro uživatele následující. Pro ostatní volby je analogické.

```
User_Alias <nazev-aliasu> = uživatel1, uživatel2 ...
```

Při použití `sudo` se vyžaduje standardně zadání hesla aktuálně přihlášeného uživatele. Pokud je požadováno spouštění příkazu bez požadavku na heslo. Je to možné použitím `NOPASSWD` v konfiguračním souboru následujícím způsobem.

```
login-user host = (user-as-run-command) NOPASSWD: command
```

Dále lze pro nastavení výchozích voleb použít klauzuli `Defaults`. Lze takto nastavit například požadavek na jiné heslo než heslo aktuálně přihlášeného uživatele. Některé volby jsou uvedeny níže.
timestamp_timeout udává dobu, po kterou nebude znovu vyžadováno heslo.
runaspw použitím této možnosti nebude vyžadováno heslo aktuálně přihlášeného uživatele, ale heslo uživatele pod kterým se má příkaz spustit.
passwd_tries určuje počet pokusů pro zadání hesla.
Dále lze měnit například proměnné prostředí a podobně.[37, 38, 39]

Pro spuštění příkazu jako jiný uživatel, vyjma uživatele root, se použije:

```
sudo -u <uzivatel> příkaz
```

2.3.8 Cron

Tato utilita umožňuje spouštění programů v určený čas. Existuje mnoho provedení, avšak v tomto textu bude popsána služba označovaná `cronie`.

Tabulka cronu je zvlášť pro každého uživatele. Syntaxe souboru s příkazy a definicí času spouštění je následující.[40, 41]

```
# Minuty Hodiny Den-v-měsíci měsíc den-v-týdnu příkaz
# (0 - 59) (0 - 23) (1 - 31) (1 - 12) (0 - 6, neděle - sobota)
```

Následující definice bude spouštět příkaz každou minutu, druhý příkaz se spustí každou desátou minutu.

```
* * * * * příkaz
*/10 * * * * příkaz
```

Pro manipulaci s tabulkou se používá příkaz `crontab` s parametry.[40, 41]

-u definuje pro kterého uživatele je požadována tabulka cronu.

-l zobrazí záznamy v tabulce.

-e umožňuje editaci záznamů.

2.3.9 Portage

Portage označuje oficiální balíčkovací a distribuční systém softwaru v Gentoo Linuxu. Obsahuje tzv. ebuildy, které popisují, jak bude software konfigurován, odkud se bude stahovat zdrojový kód, jak bude přeložen a nainstalován. Samozřejmě je součástí i popis licence, závislosti atd. Lze jej považovat za jakýsi základ Gentoo Linuxu.

Portage tvoří tzv. portage strom, kde jsou v určité hierarchii uloženy ebuildy, které obsahuje. Obsahuje i další soubory, například se jedná o jednotlivé profily, ze kterých si lze vybrat, bezpečnostní oznámení nebo popis tzv. USE flagy.

USE flagy mohou být velmi zajímavé, jelikož umožňují definovat některé závislosti programů. Lze tak vytvořit systém více na míru zařízení, kde bude systém používán. Výchozí USE flagy jsou závislé na zvoleném profilu a lze globálně povolit další nebo některé zakázat v souboru `/etc/portage/make.conf`.[42]

Konfigurační soubory jsou uloženy v `/etc/portage/` a některé důležité budou uvedeny v tomto textu s vybranými proměnnými. Další lze dohledat v dokumentaci k Gentoo Linuxu. V `make.conf` lze najít například.[43]

USE definuje USE flagy, které se mají použít nebo zakázat oproti zvolenému profilu. Pokud je uvedeno například `USE="alsa -arts"`, pak se aktivuje flag `alsa` a naopak se vypne `arts`, jelikož je uvozen pomlčkou.

ACCEPT_KEYWORDS označuje, které architektury jsou požadovány k instalaci. Pokud se uvede pouze architektura, znamená to, že budou instalovány pouze programy ve verzích označených za stabilní. Pokud se architektuře předřadí `~`, pak se povolí i instalace testing verzí.

MAKEOPTS definuje, kolik vláken může běžet zároveň při překladu programu. Například `MAKEOPTS="-j4"` udává, že lze spustit čtyři vlákna, a tím využít i více jader procesoru.

Mezi další možnosti patří například nastavení parametrů pro kompilátor jazyka C nebo C++ a další.

Soubory nebo adresáře `package.mask` a `package.unmask` slouží k tzv. zamaskování nebo odmaskování některé verze programů, například z důvodu chyby v programu může být zamaskována tato verze a nedojde tak k její instalaci. Nebo dojde k její změně, pokud už byla tato verze instalována. Jedná se o soubor nebo adresář se soubory, kde jsou tyto informace uvedeny.

Soubor nebo adresář `package.use` umožňuje definovat USE flagy pro jednotlivé programy nebo jejich konkrétní verze. K podobnému účelu slouží i soubor nebo adresář `package.keywords` v případě, že je požadován některý program například v testovacích verzích.[42]

Portage obsahuje také spoustu příkazů pro práci s ním. Hlavním z nich je příkaz `emerge`, který slouží k instalaci, odinstalaci, aktualizaci apod.

Syntaxe a některé volby tohoto příkazu budou uvedeny dále. Volby lze samozřejmě kombinovat. Základní syntaxe je následující: [44]

```
emerge <volby> <balicky>
```

-u provede aktualizace zvoleného balíčku.

-f proběhne pouze stažení zdrojových kódů a jejich kontrola pomocí kontrolního součtu.

-p provede výpočty závislostí a ukáže, co vše se bude provádět.

-v vypíše podrobnější informace o prováděných operacích. Zobrazí například USE flagy, které budou u balíčků použity.

-a provede dotaz na pokračování po zobrazení informací o následně prováděných operacích.

Synchronizace portage stromu lze provést několika způsoby. Například pomocí:

```
emerge-webrsync
```

Tento příkaz provede aktualizaci stromu pomocí portu 80. Aktualizace je také důležitá pro správnou funkci kontroly bezpečnostních problémů pomocí v textu dále uvedeného nástroje `glsa-check`.

Portage obsahuje více příkazů, jako je například aktualizace konfiguračních souborů pomocí `etc-update` a další.

2.3.10 Glsa-check

GLSA (Gentoo Linux Security Advisory) jsou oznámení o bezpečnostních problémech a jejich možné opravě v Gentoo Linuxu (obvykle aktualizací na novější verzi).

Program `glsa-check` tyto záznamy parsuje a porovnává s nainstalovanými programy, což umožní zjistit, zda v systému existuje program s bezpečnostním problémem. Tyto informace získává z portage stromu, který je nutné aktualizovat.

Program má následující parametry.[45]

-t tento parametr zobrazí nalezené problémy v systému.

-d umožní prohlédnout zadané oznámení o bezpečnostním problému.

-p zobrazí možnost, kterou je možné zadanou chybu opravit.

-f umožní opravit zadanou chybu, která je zadána za tímto parametrem. Pokud je zadáno `all`, opraví se všechny nalezené chyby.

Tento program ocení zejména správci serverů, kde se používají pouze bezpečnostní aktualizace a nikoliv aktualizace celého systému.[45]

2.3.11 Aide

AIDE (Advance Intrusion Detection Enviroment) je systém detekce průniků tzv. Host-based Intrusion Detection System. Aide skenuje soubory a další zdroje a zjištěné informace ukládá do své databáze. Při následné kontrole dochází k porovnávání záznamů s aktuálními daty v systému.

Hlavní konfigurační soubor programu je se nachází v `/etc/aide/aide.conf`. V tomto souboru je uvedeno, které soubory se mají kontrolovat a co vše bude kontrolováno. Je používána tzv. krátká notace. Význam některých zkrácených voleb je uveden dále.[46]

p vyžaduje kontrolu oprávnění nad soubory (čtení, zápis, spuštění).

u kontroluje vlastníka souborů.

g provede kontrolu skupiny.

s kontroluje velikost. Pokud se uvede velké S, pak dojde ke kontrole velikosti souboru, ale poplach se spustí pouze v případě, že je soubor menší, než je uvedeno v databázi programu. Toto je vhodné například pro logovací soubory.

m provede kontrolu času změny souboru.

md5 použije se hash MD5.

sha1 pro použití hash funkce SHA1.

Hashe, které je možné standardně používat, jsou MD5 a SHA1. Co vše má být kontrolováno, lze uvést v konfiguračním souboru takto a následně provést specifikaci u souboru nebo adresáře viz dále.

```
Logs=p+u+g+S
```

Pokud adresář obsahuje i složky nebo soubory, které se nemají zahrnout do databáze programu. lze to provést pomocí `!` před označením adresáře.

```
/var/log Logs
!/var/log/cups
```

Výše uvedené v konfiguračním souboru provede kontrolu složky `/var/log` podle nastavení kontrol Logs, ale vynechá složku `cups`, jelikož je uvozena vykřičníkem.

Před prvním použitím je nutné databázi Aide inicializovat prvním příkazem, následně vytvořenou databázi nakopírovat na uvedené místo pomocí `cp` a následná kontrola se provede pomocí druhého příkazu.

```
aide --init --config=/etc/aide/aide.conf

aide --check --config=/etc/aide/aide.conf
```

Kontrola by v ideálním případě měla být provedena offline a databáze programu by měla být uložena na bezpečném místě pro případ kompromitace systému.[46]

2.3.12 Úprava parametrů jádra pro TCP/IP

Linuxové jádro obsahuje parametry, kterými lze ovlivnit chování TCP/IP stacku. Tyto parametry lze změnit, pokud je povolena možnost změny v konfiguraci jádra, editací příslušných souborů v `/proc` nebo pomocí nástroje `sysctl`. Pro trvalé nastavení je nutné uložit tato nastavení, jelikož při restartu dojde k vymazání nastavených údajů v `/proc`. Konfigurační soubor pro `sysctl` je v `/etc/sysctl.conf`, kde lze nastavit i další volby.

net.ipv4.ip_forward pokud je nastaveno na `0`, nedochází k přeposílání paketů.

net.ipv4.icmp_echo_ignore_broadcasts při nastavení této volby na `1` nebude docházet k odpovědím na ICMP paket, který bude mít uveden v cílové adrese broadcastovou IP adresu. Zamezí se tím použití stroje pro zesílení útoku Smurf.

net.ipv4.tcp_syncookies omezí dopady SYN flood útoku.

net.ipv4.conf.all.accept_redirects deaktivací nebude směrovací tabulka umožňovat změnu pomocí ICMP paketů.

net.ipv4.icmp_echo_ignore_all aktivováním této volby přestane stroj odpovídat na veškeré ICMP zprávy.

Voleb existuje více, zde byly uvedeny jen některé. Pokud by bylo požadováno změnit soubory v `/proc/sys`, pak jsou tyto soubory k nalezení v této složce a cesta k nim je například `/proc/sys/net/ipv4/tcp_syncookies`. Pro volbu nastavení `syncookies`.[47, 48]

Fail2ban

V tomto případě jde o službu, která sleduje logy na serveru, které se týkají pokusů o přihlášení například pomocí SSH. Tento software hledá v logu neúspěšné pokusy o přihlášení. Pokud dojde k určitému počtu chybných pokusů v závislosti na nastavení, pak dojde k zablokování přístupu z dané IP adresy na dobu stanovenou v konfiguraci. Blokace probíhá na síťové vrstvě, což je výhodné v případě větších botnetů, které by se mohly pokoušet prolomit heslo.

Jedná se o software, který se neomezuje jen na některé služby na serveru. Je tedy možné do něj doplnit prohlížení dalších logů nebo provádění detekce u dalších služeb serveru.

Podobný tomuto softwaru je DenyHosts, který je ale omezen pouze na SSH a nedochází zde k blokaci na úrovni síťové vrstvy.[49, 50]

Konfigurace je umístěna v `/etc/fail2ban/`. V tomto adresáři se nachází další soubory a složky. Pro základní konfiguraci slouží soubor `fail2ban.conf`. Pro nastavení blokace přístupu se používá soubor `jail.conf`, který slouží k zapínání a vypínání filtrů uložených ve složce `filter.d`. Akce, které se mají provádět pro zablokování adres, jsou uloženy ve složce `action.d` V tomto souboru existuje několik voleb pro nastavení chování a monitorovaných služeb, některé jsou uvedeny níže.

filter definuje název použitého filtru ze složky `filter.d`. Dle tohoto filtru je následně počítán počet pokusů o přístup. Každá shoda zvýší počítadlo.

logpath ukazuje na logovací soubor, který se má prohlížet pomocí filtru.

maxretry Maximální počet pokusů, než dojde k zablokování přístupu z dané IP adresy.

findtime udává časový interval v sekundách, za který se vyhodnocuje, zda k blokaci dojde či nikoliv. Pokud v uvedené době nedojde k nalezení shody, nastaví se tento údaj na nulu.

bantime definuje čas v sekundách, po který bude IP adresa, která splnila podmínky, zablokovaná.

enabled Uuává, zda je uvedená položka povolena nebo nikoliv.

Pro kontrolu nastavení, aktivních blokací a interakcí s programem existuje příkaz:

```
fail2ban-client status        # Vypíše aktivní služby, které hlídá.
fail2ban-client status <jail> # Vypíše detailní informace
                              # o blokovaných IP atd.
```

V případě problémů s některými filtry lze použít nástroj, který použije zadaný filtr a zpracuje log v příkazovém řádku. Místo cesty k filtru je možné jeho definici vložit přímo mezi uvozovky.[49, 50]

```
fail2ban-regex <cesta-k-logu> <cesta-k-filtru>
```

2.4 Možné útoky na systém Linux a jeho služby

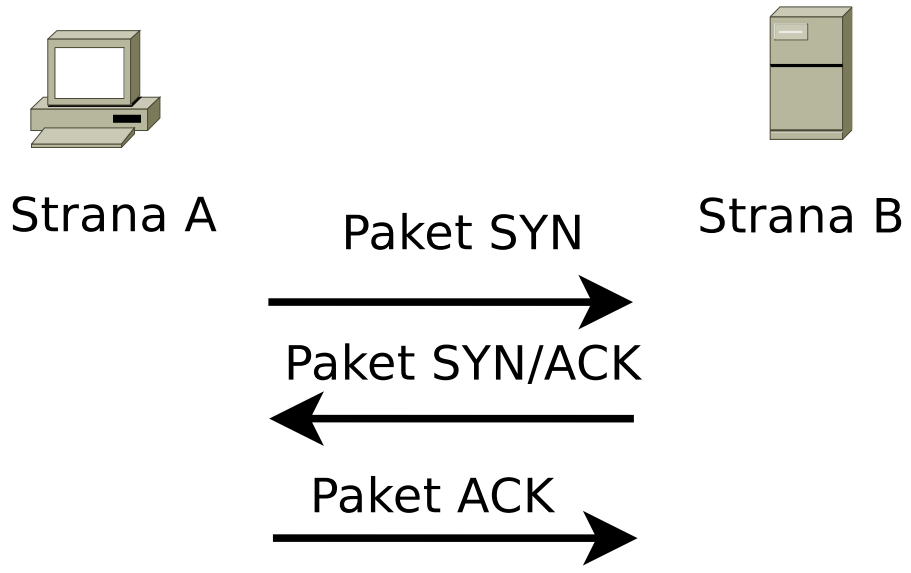
Tato část se bude zabývat některými typy útoků na systém Linux. Útoky lze rozdělit na:

- Útoky na dostupnost – jde o o útoky, které se snaží vyčerpat prostředky (paměť, procesorový čas, kapacitu linky atd.) stroje, a zamezit tak dostupnosti jeho služeb. Útoky bývají většinou označovány jako DoS (Denial of Service), případně DDoS (Distributed Denial of Service).
- Útoky na získání přístupu – v tomto případě se útočník snaží například odposlechnout nebo získat přihlašovací údaje jiným způsobem, využít chyby aplikace apod.

Jak bude útok vypadat a jakým způsobem bude veden je závislé na motivaci útočníka, jeho znalostech a prostředcích. Málo motivovaný útočník nebude pravděpodobně používat pokročilé techniky. Naopak, pokud je síť zabezpečena a nedaří se do ní dostat, může být použit útok na dostupnost služby, aby došlo alespoň k nějaké komplikaci pro správce.[51]

2.4.1 SYN Flood

Při tomto útoku jde o vyčerpání systémových prostředků. Je zde použit protokol TCP a jeho příznaky pro navazování spojení. Standardně je spojení navazováno tzv. three-way handshakem, kdy je nejdříve stranou A (klientem) zaslán paket s příznakem SYN (na straně B přejde port do stavu SYN_RECV). Na tento paket je stranou B (serverem) odpovězeno paketem s příznaky SYN/ACK a další ACK paket od A dokončí navázání spojení (u B dojde k přechodu do stavu ESTABLISHED), jak je uvedeno na obrázku 2.7.



Obr. 2.7: Standardní spojení pomocí SYN [51]

Během útoku jsou však zaslány jen první dva pakety (SYN a SYN/ACK). Tímto způsobem dojde k vyhrazení prostředků na serveru a tyto prostředky jsou vyhrazeny

do doby vypršení časovače. Tyto prostředky jsou vyhrazeny pro každé nekompletně navázané spojení a může dojít k vyčerpání prostředků na straně B. Pokud dojde k vyčerpání prostředků, pak již nelze navázat další spojení a dojde k odepření služby nebo zablokování systému.

Aby nedocházelo k navázání spojení, je obvykle v úvodním paketu podvržena IP adresa odesílatele, která není používána. Pokud by na této adrese existoval systém schopný odpovědět, pak by odeslal paket s příznakem RST, neboť spojení neinicioval a spojení by bylo zrušeno.

Tento typ útoku generuje relativně malý datový tok na lince, avšak dokáže vyřadit i velké servery. Zároveň není snadné dohledat útočníka podle IP adres, jelikož tyto jsou podvržené.

Vypsáním všech spojení lze rozeznat útok v systému. Pokud je zde mnoho nekompletně navázaných spojení, je možné, že server čelí tomuto útoku.[51]

Obrana je zde možná několika způsoby:

Zvětšení velikosti fronty pro navazovaná spojení Zvětšením fronty lze dosáhnout většího počtu navazovaných spojení a je tak možné čelit útoku. Tato možnost vyžaduje dodatečné prostředky v systému, což může mít vliv na výkon serveru.

Snížení timeoutu pro čekání na RST/ACK Tato možnost není optimální, ale může snížit dopad útoku.

Použití SYN cookie Tato volba umožňuje detekovat a zaznamenat možné SYN-flood útoky. Pokud probíhá útok, je tato volba aktivována a umožní snížení požadavků na systémové prostředky.

Omezení počtu spojení pomocí firewallu Lze také definovat maximální počet paketů s příznakem SYN za určitý časový interval.

2.4.2 Smurf

Tento útok cílí na dostupnost služeb. Je využito zranitelnosti ICMP protokolu. Zasláním zprávy ICMP echo ve zranitelné síti na všesměrovou (broadcast) adresu nebo adresu sítě a následně všechny stanice v této síti, které jsou náchylné na tento útok, odpoví na ICMP zprávu.

Tento útok, vzhledem ke koeficientu zesílení, umožňuje s celkem malým provozem na straně útočníka vygenerovat vysoký provoz směrem k oběti. Například pokud bude v zesilovací síti 50 náchylných systémů, pak jeden ICMP echo paket vyvolá 50 ICMP echo reply paketů.

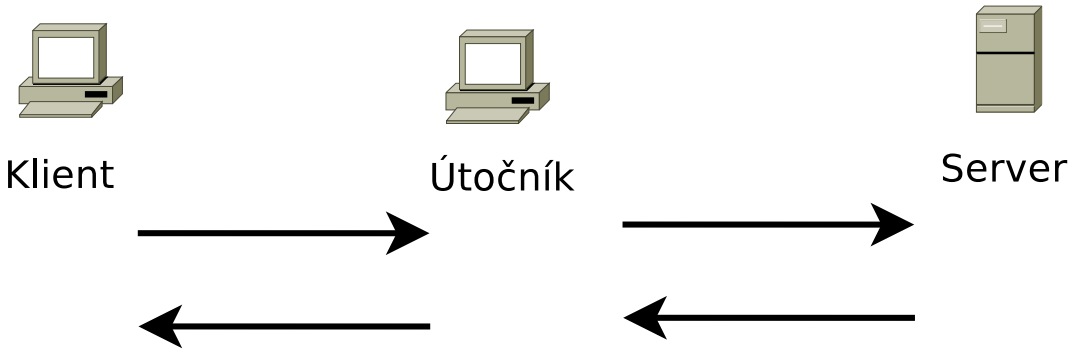
Při útoku zašle útočník ICMP echo pakety do zesilovací sítě (náchylné na tento útok) s podvrženou IP adresou odesílatele. Tu nastaví na adresu oběti útoku. Jelikož ICMP pakety směřují na broadcast adresu sítě a mají jako zdroj uvedenu adresu

oběti, vypadají tyto pakety v pořádku a všechny náchylné systémy v této síti odpoví na adresu oběti.

Obrana je realizována v případě Linuxového systému za pomoci parametru jádra, který zakáže odpovědi na ICMP pakety zaslané na všesměrovou adresu sítě. V případě směrovačů lze zakázat používání broadcastových přímých pingů. Tato ochrana nechrání přímo server, pouze znemožní použití sítě jako zesilovací.[51]

2.4.3 Man in the Middle

Tento útok necílí na dostupnost služeb, ale snaží se získat citlivé údaje. Například přihlašovací údaje k serveru. Útok se skládá ze dvou částí. První část vyžaduje přesměrování komunikace serveru a oběti přes systém ovládaný útočníkem viz obrázek 2.8.



Obr. 2.8: Princip Man in the Middle útoku

Toto je možné provést v lokální síti například pomocí techniky ARP spoofingu, kdy útočník využije možnost změny ARP (Address Resolution Protokol) tabulky zasíláním podvržených kombinací IP a MAC adres. Tímto dojde k přesměrování komunikace přes jeho systém. V internetu lze realizovat podvržení IP adresy například pomocí DNS (Domain Name System) spoofingu, kdy útočník nejprve kompromituje DNS server nebo lokální DNS cache. Zde podvrhne DNS záznamy, které neodkazují na legitimní server, ale na server útočníka.

V druhé části útočník například odposlouchává komunikaci nebo ji přímo mění. Tento útokje nejvíce nebezpečný pro nešifrovanou komunikaci, kde jsou data přímo viditelná. V případě šifrovaného přenosu dat je útočníkem zachycen jen náhodný řetězec. Pokud útočník požaduje data, musí zlomit příslušnou šifru.

Obrana proti tomuto útoku je v podobě asymetrické kryptografie a bezpečné výměny klíčů případně certifikátů, což umožní ověření protistrany například v podobě certifikátu a podpisu a tím použití šifrovaného spojení pro přenos citlivých dat.

Riziko podvržení IP adresy lze snížit správným nastavením přepínače v případě ARP spoofingu a ověřením podpisu DNS odpovědi pomocí DNSsec techniky.

2.4.4 Odposlech komunikace

Pro odposlech komunikace lze použít výše popsaný postup pro přesměrování komunikace a následné zachycení přenášených dat, případně přístup na některý síťový prvek. K zachycení provozu lze použít například program Wireshark.

Například hesla mohou být přenášena jako hash, pak je nutné pro útočníka ještě prolomit tento hash. V tomto případě je obrana podobná obraně před útokem Man in the Middle.

2.4.5 Hádání hesla ke službě

Prolomení hesla pro přístup do systému může být velký problém, pokud účet, který byl takto kompromitován, má velká oprávnění.

Pro lámání hesel lze použít:

- Slovníkový útok – Útočník disponuje souborem s možnými hesly a tato hesla jsou zkoušena pro přístup ke službě nebo systému.
- Útok hrubou silou – V tomto případě útočník nedisponuje seznamem hesel nebo slovníkový útok selhal. Je zde nutné vyzkoušet všechny možné kombinace znaků a čísel, dokud se nenajde shoda a přihlášení neproběhne úspěšně.
- Použití hash tabulek – V tomto případě útočník vlastní heslo ze systému, ale v hashované podobě, nebo jej například získal při přenosu hashe hesla po síti. Pomocí hashe hesel na seznamu lze vytvořit stejné hashe a porovnávat je se získaným hashem. Stejný hash odpovídá heslu v seznamu nebo jeho kolizi.

V tomto případě by se z pohledu zabezpečení měl vzít do úvahy i čas potřebný k prolomení hesla. Čím složitější heslo, tím déle bude trvat jeho prolomení. V rámci ochrany systému by neměla být volena slabá nebo snadno odhadnutelná hesla. Případně použít asymetrickou kryptografii pro ověření identity, pokud to lze.

2.4.6 Zranitelnost nultého dne

Tento typ útoku využívá chyby v softwaru, která není obecně známá a neexistuje tedy její oprava. Tzv. nultý den označuje čas od objevení chyby (i útočníkem) do její opravy v podobě záplaty nebo aktualizace. Útočník může v případě nalezení chyby oskenovat sledované systémy a pokud najde zranitelný software na některém serveru, může zkusit využít této chyby pro průnik do systému.

V případě obrany lze obecně doporučit oznamovat do sítě minimum informací o používaném softwaru. Například pro vzdálený přístup není nutné v úvodním banneru uvádět verzi tohoto softwaru.

2.4.7 Rootkity

Pod tímto označením je často myšlena sada nástrojů, která umožní útočníkovi mnoho různých věcí. Nejčastěji dochází ke skrytí útočníka a tohoto programu, který je v systému před zrakem administrátora. Toto se děje modifikací systémových nástrojů, které jsou běžně používány. Tyto nástroje po modifikaci neukazují například procesy, soubory nebo síťové spojení rootkitu nebo útočníka.

Rootkit může také obsahovat software, který bude zaznamenávat data uživatelů a odesílat je na server útočníka. Případně může dojít k vytvoření zadních vrátek do systému. Útočník se poté bude moci kdykoliv se do kompromitovaného systému. Rootkit může dále upravovat logy, aby zakryl svoji přítomnost nebo akce, které provádí, nebo provádět útoky na další systémy.

Rootkit může také využít možnosti nahrát se jako modul do jádra. Poté není nutné upravovat jednotlivé nástroje, ale tento rootkit může změnit přímo systémová volání.

Obrana v tomto případě spočívá hlavně v prevenci, aby se útočník do systému nedostal. Pokud k tomu dojde, může v odhalení kompromitace pomoci například logování událostí na jiný server, což běžně dostupní logovací démoni v Linuxu umožňují.

Lze využít nástroje pro sledování změn v souborech v podobě nástroje Aide nebo podobných.

Dále lze využít detektory rootkitů, které mohou rootkit detekovat. Jde o specializované nástroje, které využívají různé techniky pro odhalení rootkitů. Tyto nástroje dokáží teoreticky detekovat i dosud neznámé rootkity. Je vhodné je kombinovat s nástroji pro sledování změn v souborech.[52]

3 Praktická část

Cílem práce je připojení modulu optického zesilovače k počítači s operačním systémem Linux. Tato kapitola obsahuje informace o použitém hardware a jeho specifikacích. Dále je uvedeno nastavení a rozchození komunikace desky s modulem pomocí sériového rozhraní. Kapitola dále obsahuje informace o realizovaném programu pro zobrazování hodnot pomocí LCD displeje a ovládání tlačítky. Také byl vytvořen software pro vzdálenou konfiguraci.

3.1 Hardware

3.1.1 Raspberry Pi

Instalovaný systém má malé nároky v základu na hardware viz kapitola Gentoo Linux v úvodu práce. Proto byla zvolena deska Raspberry Pi. Deska je osazena 1 GB RAM paměti, čtyř jádrovým procesorem Cortex-A7 s taktovací frekvencí 900 MHz a architekturou ARM verze 7, LAN konektorem pro 10/100 Mbit/s Ethernet, headerem se 40 piny (vyvedena například sériová linka), USB 2.0 porty, HDMI portem a slotem pro microSD kartu. Dále je k dispozici například konektor pro připojení kamery, zvukový výstup atd. Z desky je vyvedena i sběrnice I2C na piny umístěné na headeru. Tato sběrnice bude nutná pro následné připojení LCD displeje. Deska je napájena stejnosměrným napětím 5 V pomocí micro USB portu.

Procesory ARM verze 7 umožňují spuštění celé škály Linuxu a také systému Microsoft Windows IoT. Velikost microSD karty závisí na použitém systému a jeho požadavcích na minimální prostor.

Tato deska byla zvolena z důvodu nízkých nároků na napájení a rozměry. Zároveň má vyvedeny sběrnice (sériová linka, I2C) na piny.

3.1.2 Optický zesilovací modul

Pro splnění práce byl použit optický zesilovací modul GOA-S000AC od Photonics , který je osazen 30 piny pro připojení napájení, sériové linky. Dále je na pinech vyvedena například signalizace poruchy nebo pin pro restart modulu. Modul je možné ovládat za použití sériové linky, po které se zasílají příkazy na změnu zesílení, zapnutí výstupu atd.

Modul, dále dle dokumentace, pracuje na stejnosměrném napětí 3,3 V a jeho spotřebovaný výkon je maximálně 5 W. Modul má integrovanou kontrolu teploty a výstupního optického výkonu s nastaveným automatickým vypnutím.

3.1.3 LCD displej

Při výběru LCD displeje byl požadavek instalace do 1U case, což znamenalo limitaci výškou této case. Dalším požadavkem byla délka zobrazovaného textu. Byl proto vybrán dvouřádkový LCD displej s 16 znaky na řádku. Displej komunikuje s Raspberry Pi pomocí I2C sběrnice za použití převodníku z paralelního portu a je napájen pomocí 3,3 V.

3.1.4 Tlačítka

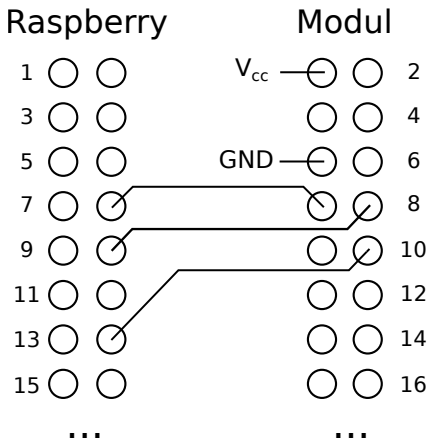
Jsou použita dvě stříbrná tlačítka pro změnu hodnot na displeji a případnou editaci. Stříbrná barva byla zvolena z důvodu následné instalace do hliníkového case. Tato tlačítka jsou připojena k desce Raspberry Pi pomocí tří vodičů.

3.1.5 Propojení

Připojení Raspberry Pi k modulu je provedeno pomocí sériové linky, která se nachází na obou zařízeních.

Raspberry Pi má tuto linku definovanou na pinech 8 a 10. Pro vysílání je použit pin 8 (TxD) a pro příjem pin 10 (RxD). Na optickém zesilovači jsou to piny 7 (RxD) a 8 (TxD). Dále je nutné propojit piny uzemnění GND například na Raspberry Pi pin 14 a pin 10 na modulu.

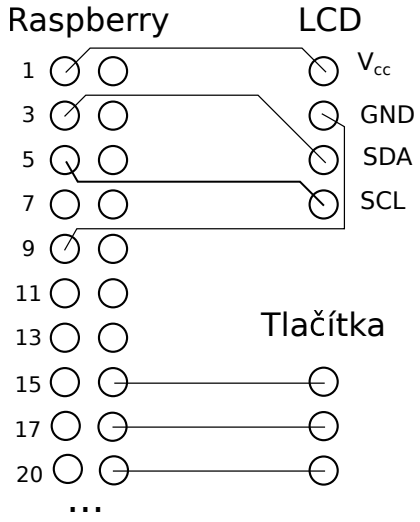
Dále je nutné připojit na optickém modulu napájení na jeden z pinů (1, 2, 29, 30) pro kladnou svorku a zápornou svorku například na pin 5. Konkrétní zapojení je znázorněno na obrázku 3.1.



Obr. 3.1: Propojení pinů mezi Raspberry Pi a modulem

Na Raspberry Pi je za použití jaderného modulu následně zřízena ještě jedna sériová linka na pinech 11 a 13. Informace o zprovoznění a podobně budou uvedeny později.

LCD displej, jak bylo uvedeno, komunikuje s Raspberry Pi pomocí I2C sběrnice. Na desce Raspberry Pi lze tuto sběrnici nalézt na pinech 3 (SDA) a 5 (SCL). Dále je nutné přivést na LCD displej napájení. Pro použití 3,3 V je na desce vyveden například pin 1 a uzemnění lze připojit například na pin 9. Zapojení je uvedeno na obrázku 3.2.

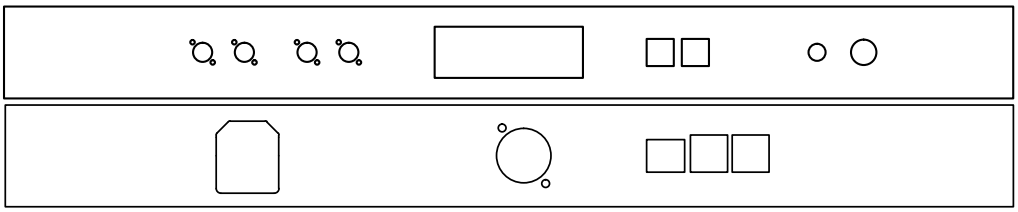


Obr. 3.2: Propojení pinů pro LCD a tlačítka

Pro připojení tlačítek jsou využity tři vodiče. Na desce s tlačítky je jeden společný vodič, který je připojen na GND pin 20 na Raspberry Pi. Ostatní vodiče jsou připojeny na GPIO (General Purpose Input Output) piny 16 a 18. Zapojení opět zobrazuje obrázek 3.2.

3.2 Návrh a realizace 1U case

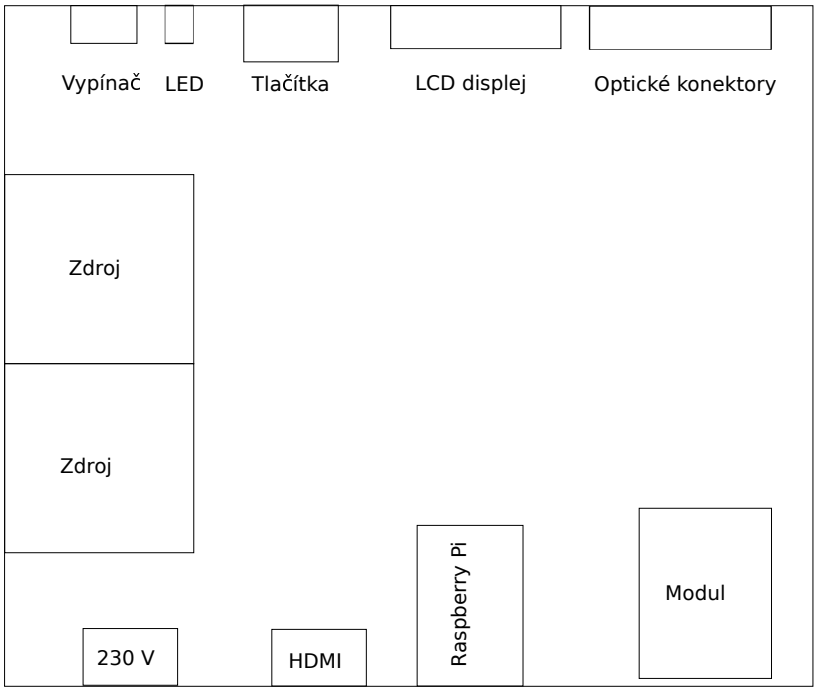
Pro uložení tohoto zařízení byl zvolen 1U case Fischer Vesa 1 360. Bylo nutné navrhnout přední a zadní panel a rozložení jednotlivých částí zesilovače v case. Návrh předního a zadního panelu je na obrázku 3.3.



Obr. 3.3: Návrh předního (nahoře) a zadního panelu (dole)

Na předním panelu je počítáno se čtyřmi optickými konektory, LCD displejem, tlačítky, LED diodou a vypínačem. Na zadním panelu jsou umístěny konektory

Raspberry Pi (USB, RJ45), dále se zde nachází konektor pro HDMI a zásuvka pro vstup 230 V. Návrh rozložení v case je na obrázku 3.4.



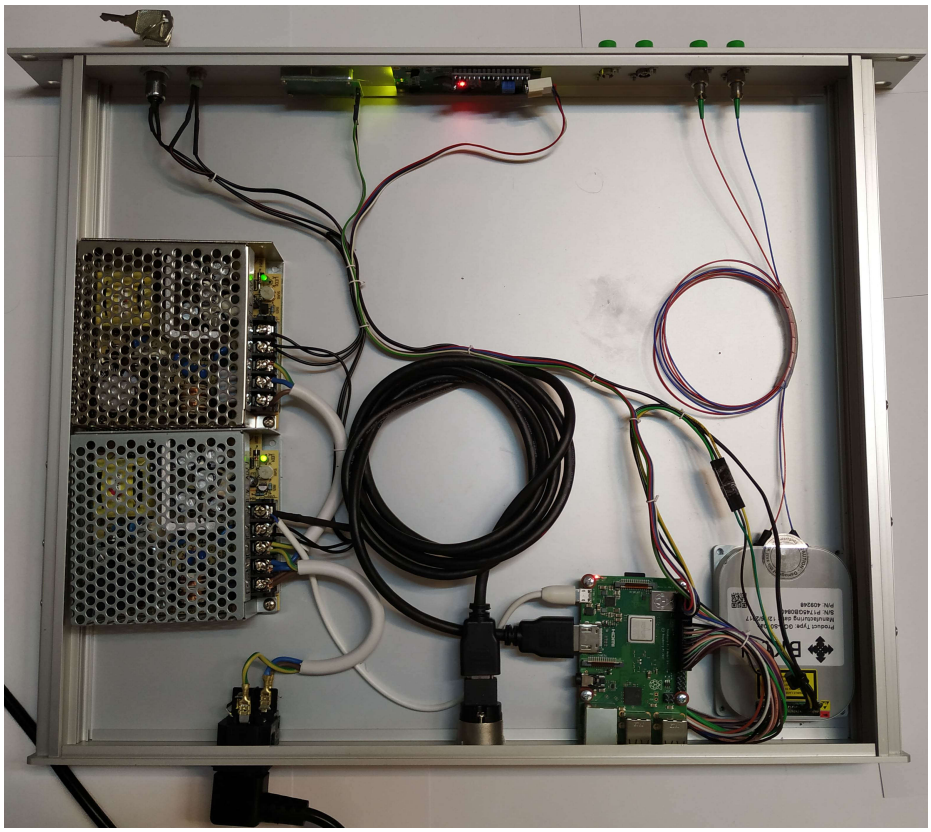
Obr. 3.4: Návrh rozložení

Následně proběhlo nakreslení panelů v programu Cad a jejich vyřezání pomocí vodního paprsku. Otvory byly vyřezány s rezervou, aby nedošlo k problémům při kompletaci zařízení.

Zařízení obsahuje dva napájecí zdroje od firmy Meanwell. Jeden pro napětí 3,3 V a druhý s napětím 5 V pro napájení desky Raspberry Pi pomocí micro USB a skrze desku i LCD displej.

Rozložení na panelech odpovídá jejich původnímu návrhu. Jako vypínač byl zvolen zámek na klíč, který je připojen mezi zdroj 3,3 V a optický zesilovací modul. Na zadním panelu se nachází spolu se zásuvkou 230 V i integrovaný vypínač přívodního napájení. Konektory RJ45 a USB byly použity přímo z desky. Fotky funkčního řešení jsou uvedeny na obrázcích 3.5 a 3.6.

Obr. 3.5: Fotka předního (nahoře) a zadního panelu (dole)



Obr. 3.6: Pohled shora na rozložení prvků

3.3 Instalace a konfigurace Gentoo Linux

Jelikož instalace systému Gentoo Linux na Raspbery Pi není jen nahrání obrazu disku na microSD kartu, jsou zde uvedeny jen důležité kroky instalace. Případné příklady a příkazy jsou uvedeny pro Linux.

Instalace systému se nachází na microSD kartě o velikosti 8 GB. Tuto kartu je nutné připravit na instalaci systému jejím naformátováním.

To lze provést například programem `fdisk`. Jako tabulka oddílů je použit Master Boot Record. Velikost prvního oddílu je 256 MB pro jádro a zavaděč. Zbytek volného prostoru je využit pro systém Gentoo Linux.

Vytvořené oddíly je nutné naformátovat. První oddíl je naformátován na souborový systém FAT32 a oddíl pro systém používá ext4 jako souborový systém. Takto naformátované oddíly je možné připojit a nahrávat do nich data.

Dalším krokem je stažení stage3 obrazu ze stránek Gentoo Linuxu. Pro Raspberry Pi 2 je určen obraz s `armv7` v názvu. Tento obraz je nutné rozbalit do budoucího kořenového oddílu. Jedná se o tar archiv s kompresí bzip2.

Poté je nutno stáhnout poslední verzi portage stromu (opět ze stránek Gentoo Linuxu) s informacemi o softwaru, který lze následně instalovat ze zdrojových

kódů. Opět jde o tar archiv s bzip2, který je třeba rozbalit. ale tentokrát do nového kořenového oddílu do složky `usr`.

Poslední stažený archiv, ale tentokrát zip, obsahuje informace pro zavedení systému, jádro a moduly jádra. Jeho umístění je na githubu Raspberry Pi a jde o stažení celého repositáře firmware. Následně je třeba jej rozbalit a nakopírovat soubory ve složce boot do prvního oddílu na kartě. Moduly jádra (adresář modules z archivu) je pak nutné nakopírovat do adresáře `lib`.

Nyní stačí editovat soubory na kartě. Všechny cesty jsou vztaženy k budoucímu kořenovému oddílu. V souboru `/etc/fstab` je nutné uvést oddíly, které jsou označeny `/dev/mmcblk0pX`, kde X je číslo oddílu, a jejich přípojné body. V souboru `/etc/shadow` je nutné přepsat řádek pro uživatele root na:

```
root::::::::
```

Tímto způsobem se vynuluje heslo pro roota. V souboru `/boot/cmdline.txt` je uvedena informace pro start systému (vše jeden řádek):

```
console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline
 rootwait
```

Nyní je možné kartu vyjmout z počítače a vložit do Raspberry Pi. Připojte také obrazový výstup a klávesnici, aby bylo možné dále konfigurovat systém na zařízení. Po připojení napájení systém naběhne a po zadání uživatelského jména root se lze dostat do systému.

Nyní je vhodné zvolit heslo pro uživatele root a vytvořit obyčejného uživatele. Heslo lze změnit pomocí příkazu `passwd` a uživatele pak pomocí `useradd`.

Připojení k síti je nakonfigurováno na statické adresy. Nejdříve je nutné vytvořit symbolický odkaz dle příkazu (uvažováno rozhraní eth0)

```
cd /etc/init.d
ln -s net.lo net.eth0
```

a následně do souboru `/etc/conf.d/net` vložit konfiguraci rozhraní například takto:

```
config_eth0="10.0.0.100/24"
routes_eth0="default via 10.0.0.1"
dns_servers_eth0="10.0.0.1 8.8.8.8"
```

Po přidání mezi služby spouštěné při startu dojde k automatickému připojení k síti a startu `sshd` pro vzdálený přístup.

```
/etc/init.d/net.eth0 start     # aktivuje konfiguraci
rc-update add net.eth0 default # přídá start eth0 mezi služby
                               # spouštěné po startu
rc-update add sshd default     # spuštení sshd po startu
```

3.3.1 Povolení sériové linky a I2C

Pro použití integrované sériové linky pro komunikaci s modulem je nutné provést dvě úpravy. První z nich je vytvoření souboru `/boot/config.txt` s obsahem

```
enable_uart=1
```

a v souboru `/etc/inittab` je nutné přidat před všechny sériové linky (řádky začínající s0, s1 atd.) znak `#`. Po restartování je možné použít integrovanou sériovou linku.

Pro případné připojení dalšího optického modulu byla za použití jaderného modulu soft_uart.ko zpřístupněna ještě jedna sériová linka na pinech 11 a 13. Bylo nutné stáhnout zdrojové kódy linuxového jádra pro Raspberry Pi ze služby GitHub a zkompilování jádra dle těchto kódů. Kompilaci jde v Gentoo Linuxu provést pomocí skriptu `genkernel`. Tyto skripty lze nainstalovat pomocí `emerge`.

```
emerge -a genkernel
```

Tento příkaz bude vyžadovat změnu licencí pro balík linux-firmware. Pokud bude zadáno `yes`, pak pomocí následujícího příkazu lze provést update konfiguračního souboru, který se má změnit. Dle zvolené volby bude původní soubor například přepsán novým.

```
etc-update
```

Skript očekává při kompilaci jádra zdrojové kódy v `/usr/src/linux`. Toto lze zajistit například vytvořením symbolického odkazu v tomto umístění.

Pokud je požadavek na získání aktuální konfigurace jádra, lze jej zjistit po nahrání modulu `configs` do jádra v souboru `/proc/config.gz`.

Pro editaci nastavení z aktuálního a následné uložení ke zdrojovým kódům lze použít následující příkaz.

```
genkernel --config=/proc/config.gz --menuconfig kernel
```

Dojde k otevření menu, kde lze změnit konfiguraci jádra, pokud je potřeba. Pokud není nutné provádět změny, lze parametr `--menuconfig` vynechat. Po potvrzení změn dojde k přeložení jádra a modulů a k následnému uložení jádra v `/boot/`. Při následné aktualizaci je možné místo výše uvedeného provést překopírování konfigurace z adresáře s původními zdrojovými kódy do adresáře s novými kódy.

Po překladu je nutné provést změnu v `/boot/config.txt`, kde je nutné uvést název nového jádra.

```
kernel=<nazev-souboru-s-jadrem>
```

Po restartu dojde k načtení nového jádra a je možné přeložit modul soft_uart.ko. Ve složce s modulem je nutné modul nejprve zkompilovat a následně nainstalovat.

```
make
make install
```

Poté je již možné zavést modul do jádra pomocí následujícího příkazu, nebo při startu obdobně jako níže uvedený modul pro I2C.

```
modprobe soft_uart
```

Po zavedení modulu se v adresáři `/dev` objeví nové zařízení s označením `ttySOFT0`, které označuje tuto linku.

Pro povolení I2C sběrnice je nutné uvést následující volbu

```
dtparam=i2c_arm=on
```

do `/boot/config.txt`. Následně je ještě nutné zavést ještě jeden modul pro I2C sběrnici. Jedná se o modul `i2c-dev`. Pro nahrání při startu systému se modul uvede do konfiguračního souboru například `/etc/modules-load/i2c.conf`.

Také je nutné provést instalaci obslužných programů `i2c-tools`, které umožní ověřit fungování I2C sběrnice a zároveň za použití USE flagu `python` doinstalují podporu pro jazyk Python. Tento USE flag je nutné definovat v konfiguraci portage například v souboru nebo adresáři `/etc/portage/package.use`.

```
emerge -a sys-apps/i2c-tools
```

Po instalaci lze zadat následující příkaz, který vypíše mimo jiné adresu LCD displeje, pokud je LCD displej správně zapojen a je napájen.

```
i2c-detect -y 1
```

3.4 Program pro komunikaci s modulem

Rozhraní pro komunikaci s moduly bylo vytvořeno v jazyce Python a je rozděleno na dvě části. A to na část pro LCD displej a na část pro vzdálené ovládání po přístupu pomocí SSH. Rozhraní počítá s dvěma moduly.

Program je v systému Gentoo Linux uložen v adresáři `/opt/conf_tool`, kde je vytvořeno několik souborů, odkud se následně načítají jednotlivá data. Ve všech případech jde o textové soubory, které lze editovat například pomocí programu `vim`. Základní soubory jsou následující.

current_module Zde jsou definovány moduly, které jsou používány dle adresáře s uloženou konfigurací pro modul.

lcd_conf.py Program pro zobrazení a ovládání modulu pomocí tlačítek a LCD displeje.

configuration_tool.py Program pro vzdálenou konfiguraci modulu pomocí vzdáleného přístupu pomocí SSH.

Každý modul je tvořen adresářem, který jej reprezentuje. V každém adresáři jsou umístěny soubory, které definují konfiguraci pro daný modul. Každý adresář obsahuje tyto soubory.

lcd_commands.data V tomto souboru se nachází příkazy pro práci pomocí LCD displeje a tlačítek.

module.conf V tomto souboru jsou definovány parametry pro komunikaci s modulem.

commands.data Zde jsou definovány příkazy rozhraní a příkazy modulu. Je zde také uvedena nápověda pro jednotlivé příkazy.

Dále je uvedena syntaxe jednotlivých konfiguračních souborů. Soubor obsahující číselný identifikátor pořadí a název složky s konfigurací modulu nese název `current_module`. Tyto údaje jsou odděleny dvojtečkou a každý řádek reprezentuje jeden modul.

```
<číselná identifikace>:<název složky s konfigurací modulu>
```

V souboru `commands.data` jsou uvedeny překlady příkazů a nápověda k nim. Jednotlivé položky v tomto souboru jsou odděleny středníkem. Nejprve je uveden příkaz používaný v ovládacím rozhraní. Druhý příkaz je pro modul a na třetí pozici je umístěna nápověda. Každý řádek definuje jeden příkaz. Pro lepší znázornění je syntaxe opět uvedena níže.

```
<příkaz v rozhraní>;<příkaz pro modul>;<nápověda>
```

Definice v souboru `module.conf` obsahuje definici sériové linky a její parametry. Syntaxe v souboru je `proměnná=hodnota`. Jednotlivé proměnné jsou uvedeny níže.

port Definuje sériové rozhraní na Raspberry Pi. Na toto rozhraní je modul připojen.
baudrate Definuje rychlost přenosu po sériové lince.
parity Zde je definováno použití paritních bitů. Hodnota 0 reprezentuje nepoužití parity. Hodnota 1 reprezentuje lichou a hodnota 2 sudou paritu.
dataBits Udává počet datových bitů.
stopBits Definuje počet ukončovacích bitů. Obvykle jeden nebo dva.

V souboru `lcd_commands.data` jsou uvedeny údaje pro zobrazení na LCD displeji a informace pro případné nastavení číselných hodnot. Oddělení jednotlivých částí je provedeno pomocí středníku. Nejdříve je v souboru umístěn příkaz pro načtení hodnoty z modulu včetně parametrů (příkaz použitý v rozhraní), následuje příkaz pro zápis hodnoty. Pokud nemá být povolen zápis, je tato položka prázdná. Následuje číselná hodnota definující krok, o který se zvýší/sníží hodnota pro zápis při stisku tlačítka. Posledním údajem je zde pozice hodnoty ze čtecího příkazu, která se má změnit. Číslování je prováděno od nuly. Opět každý řádek definuje jeden příkaz.

```
<příkaz pro vyčtení>;<příkaz pro zápis>;<krok>;<pozice hodnoty>
```

Pro komunikaci po sériové lince byla použita knihovna pro Python označovaná `pyserial`. Do Gentoo ji lze doinstalovat za použití příkazu `emerge`.

```
emerge -a dev-python/pyserial
```

Dále bylo nutné umožnit z Pythonu přístup na piny, kde jsou umístěna tlačítka. K tomuto bylo použito knihovny `RPi.GPIO`, kterou lze do systému instalovat pomocí následujícího příkazu.

```
pip install --user RPi.GPIO
```

Uvedený příkaz nainstaluje uvedenou knihovnu pod přihlášeným uživatelem. Toto je důležité, neboť i program `emerge` je napsán v jazyce Python a mohlo by dojít k problémům při instalaci bez parametru `--user`.

Dále byla použita volně šiřitelná knihovna pro komunikaci s LCD displejem, která je uložena v adresáři s programem. A další součásti Pythonu, například knihovna time.

3.4.1 Komunikace LCD s moduly

Jak již bylo uvedeno, program je napsán v jazyce Python. Vývojový diagram tohoto programu je uveden na obrázku 3.7. Nejprve byly v programu zpřístupněny piny na Raspberry Pi pomocí knihovny RPi.GPIO, což lze provést pomocí následujícího kódu.

```
import RPi.GPIO
buttonUP = 16
buttonDOWN = 18
GPIO.setmode(GPIO.BOARD)
GPIO.setup(buttonUP,GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(buttonDOWN,GPIO.IN,pull_up_down=GPIO.PUD_UP)
```

Bylo zvolena číslovací schéma pro piny. Piny se v tomto případě číslují podle fyzického umístění na Raspberry Pi. Byly definována čísla pinů tlačítek a pomocí interních pull-up rezistorů nastavena do logické 1. Při stisku tlačítka je na pinu detekována logická 0.

Následně pomocí funkce `configureLoad` dojde k načtení všech potřebných souborů pro konfiguraci sériové linky, příkazů atd. Návratovou hodnotou této funkce je definice sériového rozhraní.

Následně je definována funkce pro čtení dat z modulu `commandRead` a pro změnu hodnot v modulu `commandWrite`.

Dále byly přidány interní volby pro změnu zobrazeného modulu a pro přepínání IP adresy mezi statickou a získanou pomocí protokolu DHCP (Dynamic Host Control Protokol). Toto bylo provedeno z důvodu přítomnosti pouze jednoho portu pro připojení k síti. Na displeji je možné také zobrazit IP adresu.

Následně je v nekonečném cyklu volána funkce `commandRead`, resp. vybraný příkaz modulu nebo interní volba dle zobrazených údajů. V cyklu se také sleduje, zda nebylo stisknuto tlačítko.

Pokud dojde ke stisku jednoho tlačítka, změní se zobrazovaný údaj na LCD displeji. Pokud dojde ke stisknutí obou tlačítek zároveň, dojde k přechodu do editačního režimu, pokud je povolen. Tento režim je indikován znakem `E` na displeji vpravo. Následně je možné pomocí tlačítek změnit hodnotu. Potvrzení se provede opět stiskem obou tlačítek současně a zároveň dojde k ukončení editačního režimu. Pokud se nestiskne žádné tlačítko v editačním režimu, dojde po krátké době k opuštění tohoto režimu. V případě přepínacích voleb dojde při stisku obou tlačítek k přepnutí hodnoty. Jedná se například o změnu IP adresy mezi statickou a získanou pomocí DHCP.

LCD displej umožňuje také zobrazení hodnot ze dvou modulů. Hodnoty odpovídají vybranému modulu a jeho konfiguraci. Přepnutí se provede zobrazením momentálně vybraného modulu a stiskem obou tlačítek.

Pro spouštění tohoto programu byl vytvořen skript, který zajistí jeho spuštění při startu systému. Skript je uložen v `/etc/init.d/lcd_conf`.

3.4.2 Vzdálená konfigurace modulů

Toto rozhraní lze spustit pomocí následujícího příkazu pod běžným uživatelem. V systému je vytvořen účet `rpi`. Na tento účet se lze připojit pomocí SSH a znalosti hesla.

```
sudo /opt/conf_tool/configuration_tool.py
```

Vývojový diagram této části je uveden na obrázku 3.8. Na začátku dojde k výzvě k vybrání modulu, který má být konfigurován a následně jsou načtena data ze souborů modulu, která jsou nutná k fungování rozhraní. Funkce nese název `configureLoad`, avšak je mírně odlišná od stejnojmenné funkce pro LCD displej. V tomto případě nedochází k načítání dat ze souboru `lcd_commands.data`, jelikož tato data nejsou nutná. Návratovou hodnotou je opět definice sériového rozhraní.

V dalším kroku dojde k vypnutí ovládání pomocí LCD displeje, aby nedocházelo ke vzájemnému přepisování hodnot.

Následně je spuštěn cyklus, který zobrazí řádek pro zadání příkazu a vyčkává na zadání příkazu do rozhraní. Příkazy jsou definovány v konfiguraci modulu v souboru `commands.data` nebo jde o interní příkazy rozhraní, kdy lze například pomocí `changeserial` vybrat jiný modul pro konfiguraci, `quit` ukončí konfiguraci a umožní opět ovládání pomocí LCD displeje nebo `help`, který vypíše nápovědu k příkazům. Ukázka části nápovědy je uvedena níže. Znaky `>>>` označují zadaný příkaz.

```
>>>help
ldtemp_r --  Parametrem je číslo diody. Vypíše teplotu zvolené diody
 ve stupních Celsia.
inttemp_r --  Bez parametru. Zobrazí interní teplotu modulu ve
stupních Celsia.
mode_r --  Bez parametrů. Zobrazí zvolený režim PC - výkonový,
GC - ziskový, CC - proudový, OFF - vypnutí.
mode_s --  Parametrem je režim. Slouží pro zvolení režimu
PC - výkonový, GC - ziskový, CC - proudový, OFF - vypnutí.
voltage_r --  Zobrazí po zadání napájecí napětí ve voltech.
help -- Zobrazí tento výstup.
quit -- Ukončí konfigurační rozhraní.
```

Pokud dojde k zadání příkazu, který odpovídá příkazu modulu, dojde k jeho překladu na příkaz, kterému rozumí modul, a zašle se mu po sériové lince. Pokud není příkaz nalezen mezi interními příkazy ani příkazy definovanými pro modul, pak dojde k vypsání chybové hlášky a opět k výzvě pro zadání příkazu.

Požadavek na informace o modulu lze modulu zaslat pomocí příkazu `info`, jak je ukázáno v následujícím výpisu (znaky `>>>` uvozují možnost zadávání).

```
>>>info
RI
Vendor= Photonics
Module= GOA-S000AC
HW Ver= 0
HW Rev= A
SW Ver= 2.00
FW Ver= 0.00
Part Num= 409248
Ser. Num= P1745GB08406
Prod. Date= 120617
```

Pokud by vznikl požadavek na zjištění interní teploty, lze to provést pomocí `intemp_r`, případně na napájecí napětí `voltage_r`

```
>>>intemp_r
IT 31.76 C
>>>voltage_r
V 3.26 V
```

Proud na diodě a teplotu lze získat zadáním příkazu a čísla reprezentujícího diodu.

```
>>>ldtemp_r 1
LT 1 24.99 C
>>>ldcurrent_r 1
LC 1 0.60 mA
```

V rámci rozhraní lze i měnit parametry modulu, například režim, ve kterém pracuje modul, proud na diodě, zesílení nebo výkon. Dle možností modulu.

3.4.3 Vytvoření nové konfigurace modulu

Vytvoření konfigurace nového modulu je velmi jednoduché. Je vhodné zkopírovat funkční modul a následně změnit hodnoty v konfiguračních souborech modulu. Pokud je modul připojen, je nutné změnit i soubor `current_module` a uvést jej zde. Pozor na definici sériového rozhraní v `module.conf`, aby nedošlo k definici stejného rozhraní na více aktivních modulech.

3.5 Zabezpečení systému

V této části bude popsáno provedené zabezpečení systému na Raspberry Pi proti možným útokům po síti.

V systému Gentoo Linux se podobně jako v jiných systémech GNU/Linux využívá nástroj zvaný `iptables`. Firewall byl nastaven následujícími příkazy.

```
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
```

První řádek povoluje spojení, která byla navázána nebo budou navazována v souvislosti s jiným spojením. Druhý řádek povoluje všechna spojení na lokální smyčce (rozhraní lo). Třetí řádek povoluje ping na zařízení zprávou typu echo request a čtvrtý řádek definuje povolení pro přístup na SSH na portu 22. Poslední tři řádky definují výchozí politiky, které se uplatní, pokud paket nesplní žádné pravidlo. Pro `INPUT` a `FORWARD` bude vše zahazováno a pro odchozí provoz bude vše povoleno. Paket prochází cestou uvedenou na obrázku 2.4. Firewall je nutné přidat mezi služby aktivované při startu systému.

Ochranou proti MitM útoku poskytuje SSH tím, že při připojení zobrazuje otisk stroje, který může uživatel zkontrolovat při prvním připojení s otiskem, který mu byl dodán bezpečnou cestou. A následně při potvrzení dojde k uložení k IP adrese. V Linuxovém klientovi je zobrazena následující hláška.

```
The authenticity of host '192.168.2.2 (192.168.2.2)' can't be
established. ED25519 key fingerprint is
SHA256:Pt2SaDrOgkUHssWsxpRHZNcVjRqa2v0VAWHHDagt43E.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Otisk se může přegenerovat na serveru použitím následujícího příkazu. Otisků na serveru existuje více, pokaždé je použita jiná šifra. Šifra je specifikována pomocí parametru `-t`.

```
ssh-keygen -f /etc/ssh/ssh_host_ecdsa_key -t ecdsa
```

Pokud by se v budoucnu tento otisk lišil, například v Linuxovém klientovi nedojde k povolení přístupu k serveru a zobrazí se co je nutné provést, aby bylo možné se připojit. Tímto je zabráněno tomuto útoku.

Samotný server SSH je konfigurován v souboru `sshd_config`. Zde byl vybrán uživatel, který může přistupovat k SSH. Bylo zakázáno přesměrování grafického výstupu X11. Uživatel root se smí přihlásit, ale pouze certifikátem asymetrické kryptografie. Tento certifikát byl vygenerován na klientském stroji (v tomto případě Linux) pomocí následujícího příkazu.

```
ssh-keygen -t rsa
```

Délka klíče byla zvolena 2048 bitů a byla použita šifra RSA.

Následně byla veřejná část certifikátu přenesena na Raspberry Pi za použití sítě. Bylo nutné na chvíli povolit přihlášení uživatele root heslem, nahrát certifikát a opět pro uživatele root zakázat přístup heslem. V konfiguračním souboru se jedná o volbu `PermitRootLogin`. Certifikát byl přenesen následujícím příkazem.

```
ssh-copy-id root@192.168.2.2
```

Pro ochranu proti útoku SYN flood a Smurf byly změněny pomocí `sysctl` parametry jádra, které tomuto brání a následně i některé další následujícím způsobem v `sysctl.conf`. Jelikož je na zařízení pouze jedno rozhraní, není nutné, aby docházelo k přeposílání paketů.

```
net.ipv4.ip_forward = 0
net.ipv4.tcp_syncookies = 1
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.default.secure_redirects = 0
net.ipv4.icmp_echo_ignore_broadcasts = 1
```

Touto změnou došlo k omezení odpovědí na ping, který je směřován na všesměrovou adresu a změnu směrovacích tabulek pomocí ICMP. Byla také provedena aktivace syncookies, která může snížit nebo úplně zabránit útoku SYN flood.

Jelikož se uživatelé přihlašují pomocí hesla, je vhodné snížit počet pokusů o přihlášení a pokud dojde k jejich vyčerpání, bude IP adressa na určitou dobu zablokována. Důvodem je například pokus o brute-force útok pro prolomení hesla. Tímto se čas výrazně prodlouží. K tomuto účelu byl použit program `fail2ban`. Je však nutné nejdříve spustit logovací službu (daemona), aby mohl tento program správně pracovat. Instalace logovacího daemona a fail2ban se provede následovně.

```
emerge -av app-admin/sysklogd net-analyzer/fail2ban
```

Do `/etc/fail2ban/jail.conf` je nutné pro sledování přístupu na SSH přidat následující konfigurační volby.

```
[ssh]
enabled=true
filter=sshd
action=iptables[name=SSH, port=ssh, protokol=tcp]
logpath=/var/log/auth.log
maxretry=2
bantime=600
```

Oba výše uvedené programy je nutné spustit při startu systému. To lze provést pomocí následujících příkazů.

```
rc-update add sysklogd default
rc-update add fail2ban default
```

Na základě výše uvedeného se aktivuje sledování přístupů na SSH a pokud bude počet chybných pokusů o přístup vyšší než dva, pak dojde k zablokování IP adresy na deset minut.

Zablokované adresy lze zkontrolovat pomocí příkazu níže.

```
fail2ban-client status ssh
```

Adresa je zablokována pomocí `iptables` pomocí nově vytvořeného chainu, kde jsou uvedeny blokované IP adresy. Kontrola zablokovaných adres je možná dle výše uvedeného příkazu. Výstup je uveden níže.

```
Status for the jail: ssh
|- Filter
|  |- Currently failed: 0
|  |- Total failed:     3
|  `- File list:        /var/log/auth.log
`- Actions
|- Currently banned:    1
|- Total banned:        1
`- Banned IP list:      192.168.2.3
```

Pomocí programu `sudo` je možné pro některé příkazy zvýšit oprávnění běžnému uživateli, jak bylo uvedeno. Pro povolení spuštění příkazu pro konfigurační rozhraní je nutné pomocí `visudo` editovat konfigurační soubor a přidat následující.

```
rpi ALL=(root) NOPASSWD:/opt/conf_tool/configuration_tool.py
```

Pokud je konfigurace provedena správně, pak po uživateli nebude po zadání příkazu

```
sudo /opt/conf_tool/configuration_tool.py
```

vyžadováno heslo, standardně je požadováno heslo uživatele, který příkaz spustil, a dojde k otevření konfiguračního rozhraní pod uživatelem root. Ostatní příkazy, kde jsou vyžadována zvýšená práva, nebudou dotčeny. Při pokusu změnit adresu se vygeneruje následující výpis a provede se záznam do logu systému.

```
Sorry, user rpi is not allowed to execute '/bin/ls' as root
on computer.
```

Pro detekci průniků byl použit detekční software `aide`, který dokáže detekovat, že došlo ke změně v nějakém sledovaném souboru. Níže jsou uvedeny dvě proměnné, ve kterých je uvedeno umístění databáze. První uvádí umístění aktivní databáze a druhá místo, kam se vygeneruje nová databáze. Dále jsou v tomto souboru uvedeny údaje, které se mají sledovat na definovaných souborech. Konfigurace se nachází v souboru `/etc/aide/aide.conf`.

```
database=file:/var/lib/aide/aide.db
database_out=file:/var/lib/aide/aide.db.new
```

Po změně konfigurace je nutné vygenerovat novou databázi pro tento software. A následně provést kontrolu například v pravidelných intervalech pomocí definice v cronu. Informace o shodě dat s vygenerovanou databází signalizuje program pomocí výstupu, kde je mimo jiné uvedeno:

```
AIDE found NO differences between database and filesystem.
Looks okay!!
```

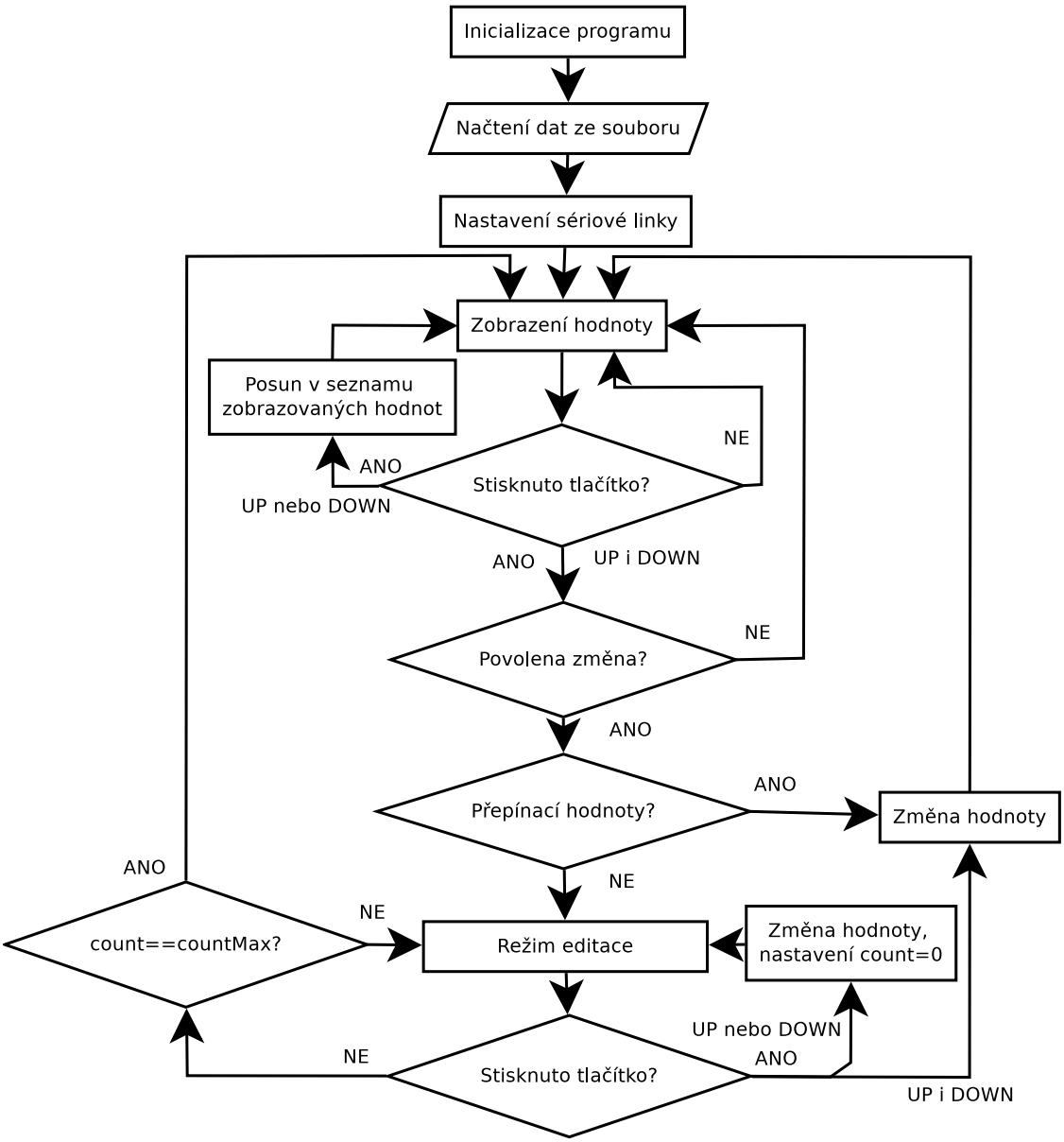
Je zároveň důležité, aby všechny bezpečnostní záplaty byly instalované. V Gentoo Linuxu toto umožní nástroj `glsa-check`, který z portage stromu parsuje informace o bezpečnostních problémech. Dokáže je zobrazit a také provést navrženou nápravu, například aktualizaci softwaru.

```
emerge-webrsync
glsa-check -t all
glsa-check -f <ID_problému>
glsa-check -f all
```

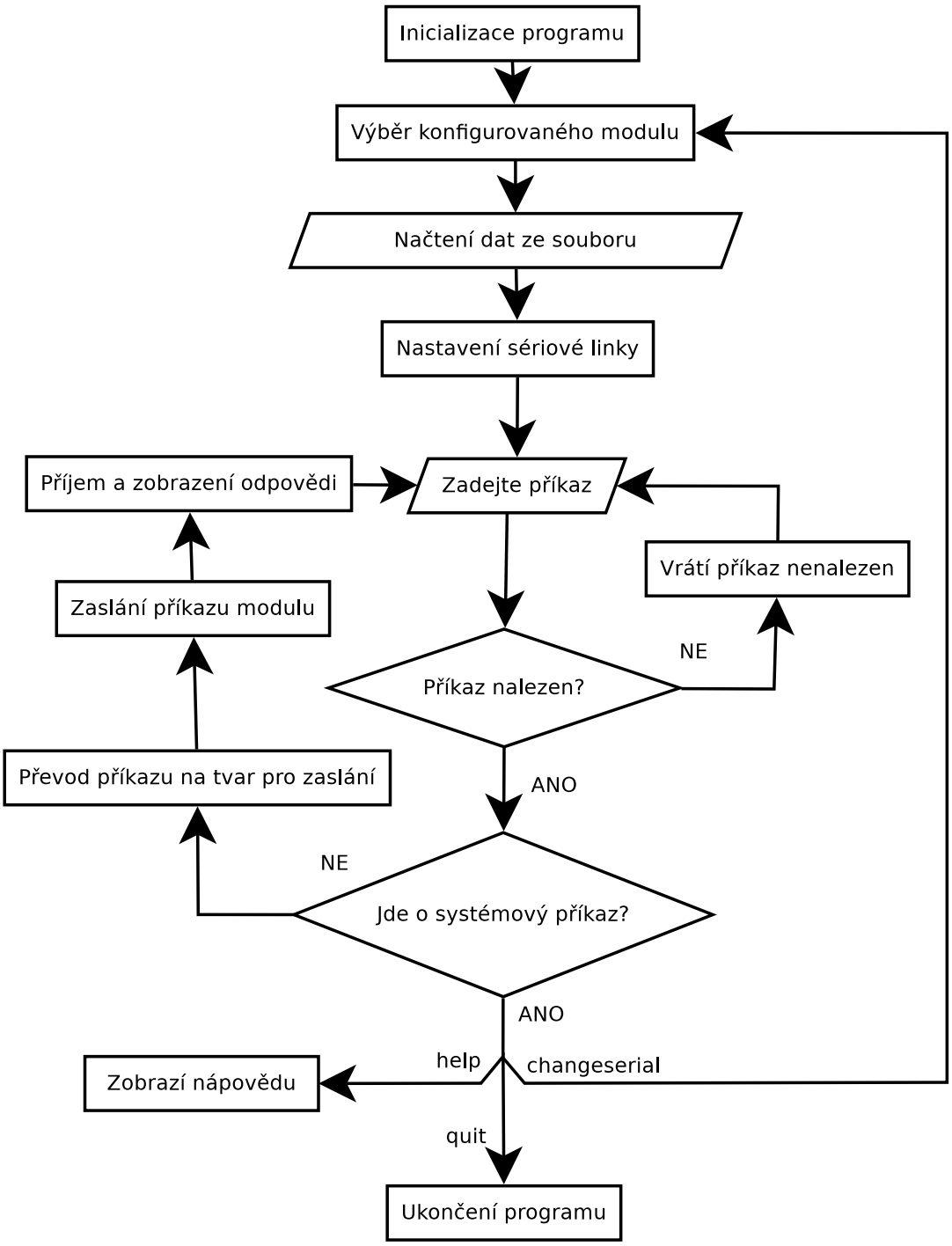
První řádek aktualizuje portage strom, druhý příkaz provede kontrolu, zda se v systému nenachází zranitelné balíčky, třetí následně aplikuje opravu zadané zranitelnosti a poslední příkaz opraví všechny nalezené problémy. Správce se může zároveň přihlásit do mailing listu, kde jsou tyto problémy také zveřejňovány. Takto lze získat informace o bezpečnostních problémech a reagovat na jejich výskyt doporučenou opravou.

V instalovaném systému `glsa-check` neukázal žádný balíček s bezpečnostním problémem. Což značí hláška uvedená níže.

```
This system is not affected by any of the listed GLSAs
```



Obr. 3.7: Vývojový diagram rozhraní LCD



Obr. 3.8: Vývojový diagram navrženého rozhraní pro vzdálenou konfiguraci

Závěr

V první části této diplomové práce bylo uvedeno rozdělení sítí dle rozlohy, účelu v síti, topologie a také optických zesilovačů podle umístění na optické trase. Dále jsou zde uvedeny základní informace o Ramanovských a EDFA zesilovačích.

V následující části byly zmíněny informace o napájecích zdrojích a komunikačních rozhraních a také o systému GNU/Linux, jeho distribucích a následně informace o firewallech a uživatelských účtech a některých programech, například pro omezení počtu pokusů o přihlášení. Také byl uveden protokol SSH a použití asymetrické kryptografie. V této kapitole jsou také prezentovány možnosti omezení lokálního i vzdáleného přístupu k systému, možné útoky a také možná obrana.

V poslední části byl propojen optický modul, tlačítka a LCD displej s jednodeskovým počítačem Raspberry Pi. Modul byl připojen pomocí sériové linky, LCD displej pomocí I2C sběrnice a tlačítka pomocí GPIO pinů. Byl proveden návrh předního a zadního panelu 1U case a také bylo navrženo rozmístění prvků uvnitř skříně. Dále byla provedena instalace operačního systému Gentoo Linux a bylo vytvořeno komunikační rozhraní pro konfiguraci modulů v jazyce Python. Také byl vytvořen program pro výpis informací na LCD displej a případně změnu některých hodnot modulu pomocí tlačítek. Také bylo provedeno zabezpečení proti útokům na systém Linux.

Vývojové diagramy, syntaxe souborů, které jsou používány rozhraním, návrh panelů a popis zapojení byly uvedeny v práci. Funkčnost zabezpečení a komunikace byla ověřena v reálném zapojení a výstupy jsou uvedeny v práci včetně fotografií zařízení. Součástí práce je elektronická příloha obsahující program popsaný v práci (konfigurační rozhraní).

Literatura

[1] DVOŘÁK, Tomáš. *Optické zesilovače a jejich aplikace* [online]. Brno, 2016 [cit. 2019-12-18]. Dostupné z: https://www.vutbr.cz/studenti/zav-prace/detail/93754. Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce Ing Petr Münster, Ph.D.

[2] KŘIŽANOVSKÝ, Pavel. *Návrh vysoce dostupných campus sítí* [online]. 2012 [cit. 25. 4. 2020]. Dostupné z: https://www.cisco.com/c/dam/global/cs_cz/assets/expo2012/pdf/

ARCH1_Campus_Design_Pavel_Krizanovsky.pdf

[3] BOUŠKA, Petr. *Počítačové sítě - základní topologie < články -> SAMURAJ-cz.com* [online]. 30.07.2007, 07.04.2009 [cit. 25. 4. 2020]. Dostupné z: https://www.samuraj-cz.com/clanek/pocitacove-site-zakladni-topologie/

[4] BRANČ, Martin. *Media konvertory a optické switche* [online]. Brno, 2011 [cit. 2019-12-19]. Dostupné z: https://www.vutbr.cz/studenti/zav-prace/detail/39820. Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce Ing Petr Münster, Ph. D.

[5] MARTIN-RAMOS, Pablo, Pedro CHAMORRO-POSADA a Jesus MARTIN-GIL. *Novel Erbium(III) and Ytterbium(III)-based materials for Optoelectronic and Telecommunication applications*. 2014. ISBN 9788469701058. DOI: 10.13140/2.1.4293.7282.

[6] BURČÍK, J. *Ramanovské zesilovače pro telekomunikace* [online]. 29. 03. 2006 [cit. 30. 4. 2020]. Dostupné z: http://access.feld.cvut.cz/view.php?nazevclanku=ramanovske-zesilovace-pro-telekomunikace&cisloclanku=2006032401

[7] BABČANÍK, jan. *Spínané zdroje | Vývoj.HW.cz* [online]. 2. Květen 2007 [cit. 23. 4. 2020]. Dostupné z: https://vyvoj.hw.cz//teorie-a-praxe/spinane-zdroje.html

[8] TIŠNOVSKÝ, Pavel. *Komunikace po sériové sběrnici I2C - Root.cz* [online]. 8. 1. 2009 [cit. 20. 4. 2020]. Dostupné z: https://www.root.cz/clanky/komunikace-po-seriove-sbernici-isup2supc/

[9] *Stručný popis sběrnice I2C a její praktické využití k připojení externí eeprom 24LC256 k mikrokontroléru PIC16F877 | Vývoj.HW.cz* [online]. 20. Květen 2000 [cit. 20. 4. 2020]. Dostupné z: https://vyvoj.hw.cz/navrh-obvodu/strucny-popis-sbernice-i2c-a-jeji-prakticke-vyuziti-k-pripojeni-externi-eeprom-24lc256

[10] *IEEE 802.3 ETHERNET* [online]. 9. 12. 2019 [cit. 28. 11. 2019]. Dostupné z: http://www.ieee802.org/3/

[11] *Jiří Peterka: Báječný svět počítačových sítí, část XX: Příběh Ethernetu* [online]. 2006 [cit. 29. 11. 2019]. Dostupné z: https://www.earchiv.cz/b06/b1200001.php3

[12] *IEEE 802.3 MAC Frame and Address Format* [online]. 1. 11. 2001 [cit. 28. 11. 2019]. Dostupné z: http://www.informit.com/articles/article.aspx?p=131216&seqNum=5

[13] PETERKA, Jiří. *Lekce 4: Ethernet II* [online]. ©2015 [cit. 22. 4. 2020]. Dostupné z: https://www.earchiv.cz/l226/gifs1/NSWI021v4_4.pdf

[14] *How are the Color Codes 568A and 568B different?* [online]. 11 June 2019 [cit. 28. 11. 2019]. Dostupné z: https://www.apc.com/us/en/faqs/FA158594/

[15] *Data Communications Basics / A Reference Guide* [online]. [cit. 20. 11. 2019]. Dostupné z: https://www.camiresearch.com/Data_Com_Basics/data_com_tutorial.html

[16] *Co je to Linux* [online]. 17.1.2006 [cit. 15. 11. 2019]. Dostupné z: https://www.abclinuxu.cz/ucebnice/uvod/co-je-to-linux

[17] KOMOSNÝ, Dan a kol. *Síťové operační systémy*. FEKT VUT, Brno, 2018. 131 s.

[18] *Debian GNU/Linux - instalační příručka* [online]. 2019 [cit. 15. 11. 2019]. Dostupné z: https://www.debian.org/releases/stable/amd64/index.cs.html

[19] *Domovská stránka | Ubuntu CZ/SK* [online]. 2019 [cit. 15. 11. 2019]. Dostupné z: https://www.ubuntu.cz/

[20] *Gentoo Wiki* [online]. 2019 [cit. 15. 11. 2019]. Dostupné z: https://wiki.gentoo.org/

[21] *What Is a Firewall?* [online]. [cit. 30. 11. 2019]. Dostupné z: https://www.cisco.com/c/en/us/products/security/firewalls/what-is-a-firewall.html

[22] SCHULTZ, E. Eugene. *Types of Firewall* [online]. , 7 stran [cit. 30. 11. 2019]. Dostupné z: http://ittoday.info/AIMS/DSM/83-10-41.pdf

[23] *Správa linuxového serveru: Linuxový firewall, základy iptables II - Linux E X P R E S* [online]. 20. červenec 2010 [cit. 3. 12. 2019]. Dostupné z: https://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru-linuxovy-firewall-zaklady-iptables-2

[24] *Netfilter/iptables project homepage - The netfilter.org project* [online]. ©1999-2014 [cit. 3. 12. 2019]. Dostupné z: https://netfilter.org/

[25] *Správa linuxového serveru: Linuxový firewall, základy iptables - Linux E X P R E S* [online]. 1. červenec 2010 [cit. 3. 12. 2019]. Dostupné z: https://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru-linuxovy-firewall-zaklady-iptables

[26] *Basic Security and User Types in Linux : Linux Essentials* [online]. August 1, 2013 [cit. 6. 12. 2019]. Dostupné z: http://web.theurbanpenguin.com/51-basic-security-and-identifying-user-types/

[27] *Security - ArchWiki* [online]. 10 December 2019 [cit. 10. 12. 2019]. Dostupné z: https://wiki.archlinux.org/index.php/Security

[28] *34 Linux Server Security Tips & Checklists for Sysadmins | Process Street | Checklist, Workflow and SOP Software* [online]. March 21, 2018 [cit. 8. 12. 2019]. Dostupné z: https://www.process.st/server-security/

[29] *Faillog(8) [linux man page]* [online]. 06/24/2011 [cit. 8. 12. 2019]. Dostupné z: https://www.unix.com/man-page/linux/8/faillog/

[30] *SSH — bezpečné používání vzdáleného počítače a kopírování dat* [online]. © 2018 [cit. 6. 12. 2019]. Dostupné z: http://www.dsl.cz/jak-na-to/jak-na-ssh

[31] *OpenSSH Features* [online]. [October 8, 2019] [cit. 6. 12. 2019]. Dostupné z: https://www.openssh.com/features.html

[32] *Implementation of Diffie-Hellman Algorithm - GeeksforGeeks* [online]. [18 July 2018] [cit. 7. 12. 2019]. Dostupné z: https://www.geeksforgeeks.org/implementation-diffie-hellman-algorithm/

[33] *Jak se přihlašovat na SSH bez zadávání hesla - Root.cz* [online]. 15. 4. 2010 [cit. 8. 12. 2019]. Dostupné z: https://www.root.cz/clanky/jak-se-prihlasovat-na-ssh-bez-zadavani-hesla/

[34] *Filesystem Hierarchy Standard.* 2015. Dostupné také z: https://refspecs.linuxfoundation.org/FHS_3.0/fhs-3.0.pdf

[35] KALSI, Tajinder. *Practical Linux Security Cookbook: Secure your Linux machines and keep them secured with the help of exciting recipes*. 1. Birmingham: Packt Publishing, 2016. ISBN 9781785286421.

[36] *Understanding Linux File Permissions - Linux.com* [online]. May 18, 2010 [cit. 10. 12. 2019]. Dostupné z: https://www.linux.com/tutorials/understanding-linux-file-permissions/

[37] *Sudo in a Nutshell* [online]. ©2020 [cit. 8. 5. 2020]. Dostupné z: https://www.sudo.ws/intro.html

[38] *Sudo - Archwiki* [online]. 2020, 6 May 2020 [cit. 8. 5. 2020]. Dostupné z: https://wiki.archlinux.org/index.php/Sudo

[39] *Sudo - Gentoo Wiki* [online]. 2019 [cit. 8. 5. 2020]. Dostupné z: https://wiki.gentoo.org/wiki/Sudo

[40] *Cron - ArchWiki* [online]. 2020, 2 May 2020 [cit. 7. 5. 2020]. Dostupné z: https://wiki.archlinux.org/index.php/cron

[41] *Cron - Gentoo Wiki* [online]. 2020, 7 January 2020 [cit. 7. 5. 2020]. Dostupné z: https://wiki.gentoo.org/wiki/Cron

[42] *Poznejte své Gentoo (1) - Root.cz* [online]. 6. 12. 2004 [cit. 12. 5. 2020]. Dostupné z: https://www.root.cz/clanky/poznejte-sve-gentoo/

[43] */etc/portage/make.conf - Gentoo Wiki* [online]. 5 May 2020 [cit. 12. 5. 2020]. Dostupné z: https://wiki.gentoo.org/wiki//etc/portage/make.conf

[44] *Balíčkovací systém Gentoo Linuxu - II* [online]. 12. 3. 2004 [cit. 12. 5. 2020]. Dostupné z: https://www.abclinuxu.cz/clanky/navody/balickovaci-system-gentoo-linuxu-ii

[45] *Poznejte své Gentoo (2) - Root.cz* [online]. 13. 12. 2004 [cit. 5. 5. 2020]. Dostupné z: https://www.root.cz/clanky/poznejte-sve-gentoo-2/

[46] *AIDE - Gemtoo Wiki* [online]. 28 June 2019 [cit. 17. 5. 2020]. Dostupné z: https://wiki.gentoo.org/wiki/AIDE

[47] *Security Handbook/Network security - Gentoo Wiki* [online]. 7 August 2017 [cit. 13. 5. 2020]. Dostupné z: https://wiki.gentoo.org/wiki/Security_Handbook/Network_security

[48] *Sysctl - ArchWiki* [online]. 15 May 2020 [cit. 13. 5. 2020]. Dostupné z: https://wiki.archlinux.org/index.php/sysctl

[49] *MANUAL 0 8 - Fail2ban* [online]. 25 May 2013 [cit. 14. 5. 2020]. Dostupné z: https://www.fail2ban.org/wiki/index.php/MANUAL_0_8

[50] KRČMÁŘ, Petr. *Fail2ban: konec hádání hesel na serveru - Root.cz* [online]. 24. 6. 2013 [cit. 14. 5. 2020]. Dostupné z: https://www.root.cz/clanky/fail2ban-konec-hadani-hesel-na-serveru/

[51] SCAMBRAY, Joel, George KURTZ a Stuart MCCLURE. *Hacking bez tajemství*. 2. aktualiz. vyd. Praha: Computer Press, 2002. Komunikace a sítě. ISBN 80-722-6644-6.

[52] DOČEKAL, Michal. *Správa linuxového serveru: Rootkity - Linux E X P R E S* [online]. 14. July 2011 [cit. 14. 5. 2020]. Dostupné z: https://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru-rootkity

Seznam symbolů, veličin a zkratek

AIDE	Advance Intrusion Detection Enviroment
ARP	Address Resolution Protokol
DHCP	Dynamic Host Control Protokol
DDoS	Distributed Denial of Service
DNS	Domain Name System
DoS	Denial of Service
EDFA	Erbium doped fiber amplifier
FHS	Filesystem Hierarchy Standard
GID	Group Identifier
GLSA	Gentoo Linux Security Advisory
GPIO	General Purpose Input Output
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineering
I2C	Internal-Integrated Circuit
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
LAN	Local Area Network
LTS	Long Term Support
MAC	Media Access Control
MAN	Metropolitan Area Network
PAM	Pluggable Authentication Modules
PAN	Personal Area Network
QoS	Quality of Service
SDA	Serial Data
SCL	Serial Clock
SOA	Semiconductor Optical Amplifier
SSH	Sechure Shell
TCP	Transmission Control Protocol
UID	User Identifier
VLAN	Virtual Local Area Network
WAN	Wide Area Network

Seznam příloh

A Obsah CD

Na přiloženém CD je umístěna elektronická verze diplomové práce včetně zdrojových souborů. Také jsou zde umístěny zdrojové kódy vytvořených programů a konfigurace pro používaný modul.