

Linux on the RouterBoard

Michal Štramberský

Bakalářská práce
2015



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2014/2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Michal Štramberský**

Osobní číslo: **A11170**

Studijní program: **B3902 Inženýrská informatika**

Studijní obor: **Informační a řídicí technologie**

Forma studia: **prezenční**

Téma práce: **Linux na platformě RouterBoard**

Téma anglicky: **Linux on the RouterBoard Platform**

Zásady pro vypracování:

1. Seznamte se s platformou RouterBoard.
2. Provedte instalaci vhodné distribuce Linuxu.
3. V systému nakonfigurujte běžně využívané služby.
4. Popište zabezpečení systému.
5. Zprovozněte konfigurační rozhraní.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. Full Debian "Lenny" on an RB433AH board (Atheros AR7161). [online]. [cit. 2015-01-23]. Dostupné z: www.opensource.dyc.edu/mips-devel
2. NEMETH, Evi, Garth SNYDER a Trent R HEIN. Linux: kompletní příručka administrátora : 2. aktualizované vydání. Vyd. 1. Brno: Computer Press, 2008, 984 s. ISBN 978-80-251-2410-9.
3. KRČMÁŘ, Petr. Linux: postavte si počítačovou síť. 1. vyd. Praha: Grada, 2008, 182 s. ISBN 978-80-247-1290-1.
4. Návod k obsluze: Platforma RouterBoard s přeinstalovaným RouterOS Mikrotik. [online]. s. 5 [cit. 2015-01-23]. Dostupné z: www.i4wifi.cz/img.asp/?attid=261160
5. SOBEL, Mark G. Linux - praktický průvodce. Praha: Computer Press, 1999, 945 s. ISBN 80-7226-190-8

Vedoucí bakalářské práce:

doc. Ing. Martin Sysel, Ph.D.

Ústav počítačových a komunikačních systémů

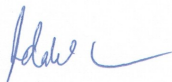
Datum zadání bakalářské práce:

6. března 2015

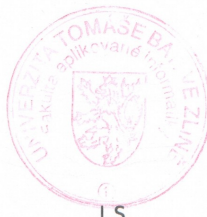
Termín odevzdání bakalářské práce:

22. května 2015

Ve Zlíně dne 6. března 2015



doc. Mgr. Milan Adámek, Ph.D.
děkan



L.S.



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

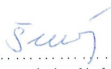
Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s tím, že tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považuji se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor;
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně 24. 5. 2015


.....
podpis diplomanta

ABSTRAKT

Práce se zabývá základním představením platformy RouterBoard, především hardwarovou konfigurací RB450G. Popisem licenčního systému Mikrotik RouterOS. Představením operačního systému GNU/Linux a popsání distribucí GNU/Linux určených pro směrovací zařízení a jiná embedded zařízení. V praktická část se věnuje instalaci a konfiguraci operačního systému OpenWRT na zařízení RB450G, instalací používaných služeb a zabezpečení systému proti útokům zvenčí.

Klíčová slova: Linux, OpenWRT, RouterBoard, Mikrotik, RouterBoot, LuCI

ABSTRACT

This thesis deals with basic presentation of RouterBoard platform, especially RB450G's hardware configuration, description of licensed system Mikrotik RouterOS, introduction of operating system of the GNU/Linux and description of the GNU/Linux distribution appointed for routing devices and other embedded devices. Practical part describes installation and configuration of operating system OpenWRT on model RB450G and its installations of services securing the system against external attacks.

Keywords: Linux, OpenWRT, RouterBoard, Mikrotik, RouterBoot, LuCI

Děkuji vedoucímu bakalářské práce doc. Ing. Martinu Syslovi, Ph.D. za užitečné rady a především trpělivost. Dále bych chtěl poděkovat všem, kteří mne podporovali a motivovali v mé práci.

OBSAH

ABSTRAKT.....	5
ABSTRACT.....	5
ÚVOD.....	10
I TEORETICKÁ ČÁST.....	11
1 PLATFORMA ROUTERBOARD.....	11
1.1 RB450G.....	12
1.1.1 Architektura Mips.....	12
1.2 MIKROTIK ROUTEROS.....	12
1.2.1 Licence.....	13
1.2.2 Konfigurace.....	14
2 LINUX.....	15
2.1 DISTRIBUCE LINUXU POUŽÍVANÉ NA ROUTOVACÍCH ZAŘÍZENÍCH.....	15
2.1.1 OpenWRT.....	15
2.1.2 DebWRT.....	16
2.1.3 Linux Voyage.....	16
2.2 LINUX FILESYSTEM HIERARCHY STANDART (FHS).....	16
2.3 BEZPEČNOST SYSTÉMU.....	18
2.3.1 Viry a červi.....	18
2.3.2 Trojské koně.....	18
II PRAKTICKÁ ČÁST.....	20
3 UPGRADE MIKROTIK ROUTEROS, ROUTERBOOTU A ZÁLOHA SYSTÉMU.....	20
3.1 PŘIPOJENÍ ROUTERBOARDU PŘES WINBOX.....	20
3.2 UPGRADE MIKROTIK ROUTEROS.....	21
3.2.1 Zjištění aktuální verze a nejvyššího možného upgradu.....	21
3.2.2 Samotná instalace.....	21
3.3 UPGRADE ROUTERBOOTU.....	22
3.4 STAŽENÍ LICENČNÍHO KLÍČE.....	22
4 INSTALACE LINUXU.....	23
4.1 NASTAVENÍ SÉRIOVÉ LINKY.....	23
4.2 NASTAVENÍ BOOTOVÁNÍ V ROUTERBOOTU.....	23
4.3 NASTAVENÍ DHCP SERVERU.....	23
4.4 NASTAVENÍ SSH SERVERU.....	24
4.5 ZÁLOHA DISKŮ S MIKROTIK ROUTEROS.....	24
4.6 ULOŽENÍ LINUXU DO ROUTERBOARDU (OPENWRT).....	25
4.7 INSTALACE DRIVERŮ PRO SD KARTU.....	26

4.7.1	mount/unmount SD karty.....	26
4.7.2	Mountování karty po startu systému.....	26
5	KONFIGURACE LINUXU.....	28
5.1	NASTAVENÍ HESLA PRO UŽIVATELE ROOT.....	28
5.2	NASTAVENÍ HOSTNAME.....	28
5.3	VYTVOŘENÍ NOVÉHO UŽIVATELSKÉHO ÚČTU.....	29
5.4	UPGRADE SOFTWARE.....	30
5.5	NASTAVENÍ SÍŤOVÝCH ROZHRAŇÍ.....	30
5.5.1	Popis řešení RouterBoardu 450G.....	30
5.6	EDITACE ÚVODNÍHO BANNERU.....	33
6	POPIS SLUŽEB, JEJICH INSTALACE A KONFIGURACE.....	33
6.1	BUSYBOX.....	33
6.1.1	init.....	34
6.1.2	ash.....	34
6.1.3	Programy určené pro práci s řetězcí (awk, grep, sed).....	34
6.1.4	programy pro práci se soubory (cat, cd, cp, ls, mkdir, mv, pwd, rm, rmdir, touch).....	35
6.1.5	pomocné programy (echo, sleep).....	35
6.1.6	programy pro připojení zařízení (df, mount, umount).....	35
6.1.7	programy určené pro archivaci a komprimování souborů (gzip, tar).....	36
6.1.8	programy určené pro správu procesů (kill, ps, pidof).....	36
6.1.9	chmod.....	38
6.1.10	date.....	38
6.1.11	dd.....	39
6.1.12	dmesg.....	39
6.1.13	ln.....	39
6.1.14	netstat.....	40
6.1.15	ping.....	40
6.1.16	vi.....	40
6.2	OPKG.....	41
6.2.1	Popis služby.....	41
6.3	SUDO.....	42
6.3.1	Popis služby.....	42
6.3.2	Instalace a konfigurace služby.....	42
6.4	CRON.....	43
6.4.1	Popis služby.....	43
6.5	NTP A NASTAVENÍ ČASU.....	43
6.5.1	Popis služby.....	43
6.5.2	Instalace a konfigurace.....	44
6.6	SSH.....	45
6.6.1	Popis služby.....	45
6.6.2	Instalace a konfigurace.....	46

6.6.3	Zabezpečení služby.....	46
6.7	SFTP.....	47
6.7.1	Popis služby.....	47
6.7.2	Instalace služby.....	47
6.8	DHCP.....	47
6.8.1	Popis služby.....	47
6.8.2	Konfigurace DHCP serveru.....	48
6.9	FIREWALL (IPTABLES).....	48
6.9.1	Popis služby.....	48
6.9.2	Konfigurace služby.....	49
6.9.3	Country.....	51
6.10	LuCI.....	52
6.10.1	Popis služby.....	52
6.10.2	Instalace a konfigurace služby.....	53
6.10.3	Zabezpečení LuCI.....	54
6.10.4	Změna designu.....	55
6.10.5	Změna jazyka.....	56
7	STAŽENÍ OBRAZŮ DISKŮ S OPENWRT.....	56
	ZÁVĚR.....	56
	SEZNAM POUŽITÉ LITERATURY.....	58
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	61
	SEZNAM OBRÁZKŮ.....	63
	SEZNAM TABULEK.....	64
	SEZNAM PŘÍLOH.....	64

ÚVOD

Cílem práce je předvedení operačního systému Linux na platformě RouterBoard, přesněji na RouterBoardu 450G.

V teoretické části práce popisuje platformu RouterBoard, operační systém Mikrotik RouterOS, jeho možnosti a omezení licence. Dále popisuje operační systém Linux a jeho dostupné distribuce zaměřené na tuto problematiku.

Praktická část je zaměřena na zálohování původního operačního systému Mikrotik RouterOS, instalaci operačního systému OpenWRT. Dále popisuje softwarové vybavení, jeho instalaci, konfiguraci a zabezpečení. V poslední části se zabývá grafickým konfiguračním rozhraním LuCI.

I. TEORETICKÁ ČÁST

1 PLATFORMA ROUTERBOARD

Firma Mikrotik se specializuje na síťové zařízení jako jsou routery, modemy, switche. Všechny produkty pod platformou RouterBoard běží pod operačním systémem Mikrotik RouterOS. [2]

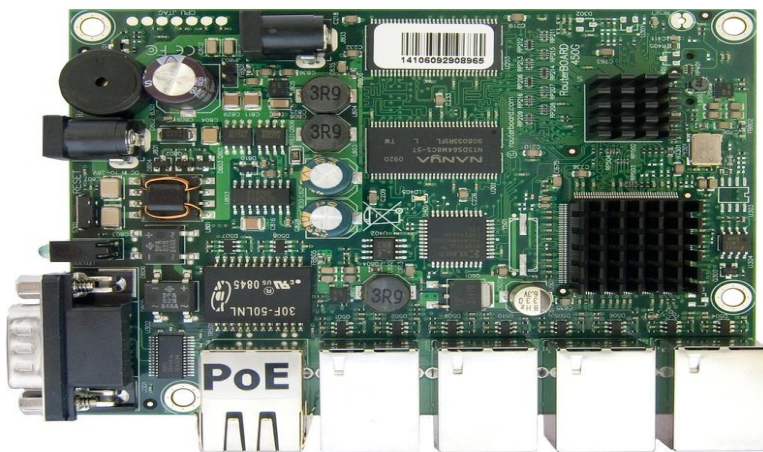
Jednotlivé verze se liší výkonem procesoru, velikostí flash paměti, počtem ethernetových portů (všechny ethernetové porty podporují automatické rozpoznání kříženého/standartního kabelu), počtem sériových portů RS-232C (pro komunikaci a konfiguraci přes Command Line Interface), počtem miniPCI slotů, počtem USB portů (do kterých je možnost zapojit flash disk nebo externí hardisk a používat RouterBoard jako FTP server) . Možností přidání MicroSD a Compact Flash karet. [1] [2] [3] [4]

RouterBoard lze napájet dvěma způsoby buď pomocí napájecího konektoru jacku (8-30V DC) nebo také pomocí PoE (Power over Ethernet 8-30V DC) dle standardu IEEE 802.3af a pasivním PoE, tedy napájení po vodiči ethernetového kabelu. Napájení přes PoE podporuje pouze tak označený konektor RJ45 na RouterBoardu. [2] [3]

Označení RouterBoardů se skládá ze tří číslic, kde první číslice určuje architekturu procesoru. Druhá číslice značí počet ethernetových portů (PoE je vždy pouze ethernet 1, pouze u série 4xx) a poslední číslice značí počet miniPCI. [1] [2] [3]

1.1 RB450G

Jak již název napovídá jedná se o RouterBoard, který obsahuje pouze 5 ethernetových portů. Má procesor Atheros 7161 (680 MHz), který je pod architekturou mips-be a jedná se o vylepšenou verzi RB450. RB450G má oproti RB450 navýšenou operační paměť z 32MB na 256MB. Dále RB450G nabízí možnost připojení MicroSD karty. [2] [5]



Obr. 1. RB450G [4]

1.1.1 Architektura Mips

Jedná se o RISC (tento typ procesorů má jednoduchou instrukční sadu) procesory bez automaticky organizované pipeline (zvládá udělat více operací za jeden takt procesoru). Tato architektura procesorů vznikla na počátku 80. let minulého století na Stanfordské univerzitě. [13]

1.2 Mikrotik RouterOS

Mikrotik RouterOS je operační systém dodávaný do produktů RouterBoard. Jedná se o operační systém na bázi Linuxu (Linux kernel 3.3.5+). Při koupi Mikrotik RouterOS získá zákazník určitou úroveň (level 0 odpovídá 24 hodin trial licence, level 1 odpovídá demo verzi, která je podmíněna registrací na webových stránkách mikrotik.com, levely 3-6 lze

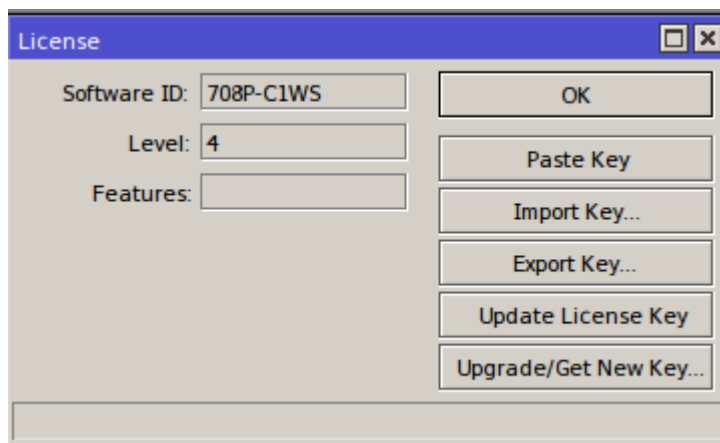
získat koupí produktu s předinstalovaným systémem nebo zakoupením licence v eshopu mikrotiku). [1] [3] [4] [6]

1.2.1 Licence

Licence Mikrotik RouterOS je vázaná na paměťové médium, kde je operační systém uložen (HDD, NAND). Informace o licenci lze získat z konzole `/system licence print` tento příkaz vypíše na konzoli id softwaru a úroveň licence. Dále lze tyto informace získat z programu WinBox, který je volně přístupný na stránkách www.mikrotik.com/download. Programy WinBox a NetInstall lze využít také k zálohování licenčního klíče v případě, že uživatel chce přeinstalovat systém. [1] [4]

Tab. 1. Licence Mikrotik RouterOS [4]

<i>level</i>	<i>0</i> <i>(Trial Mode)</i>	<i>1</i> <i>(Free Demo)</i>	<i>3</i> <i>(WISP CPE)</i>	<i>4</i> <i>(WISP)</i>	<i>5</i> <i>(WISP)</i>	<i>6</i> <i>(Controller)</i>
Prize	no key	registration required	-	\$45	\$95	\$250
Initial Config Support	-	-	-	15 days	30 days	30 days
Wireless AP	24h trial	-	-	yes	yes	yes
Wireless Client and Bridge	24h trial	-	yes	yes	yes	yes
RIP, OSPF, BGP protocols	24h trial	-	yes	yes	yes	yes
EoIP tunnels	24h trial	1	unlimited	unlimited	unlimited	unlimited
PPPoE tunnels	24h trial	1	200	200	500	unlimited
PPTP tunnels	24h trial	1	200	200	500	unlimited
L2TP tunnels	24h trial	1	200	200	500	unlimited
VPN tunnels	24h trial	1	200	200	unlimited	unlimited
VLAN interfaces	24h trial	1	unlimited	unlimited	unlimited	unlimited
HotSpot active users	24h trial	1	1	200	500	unlimited
RADIUS client	24h trial	-	yes	yes	yes	yes
Queues	24h trial	1	unlimited	unlimited	unlimited	unlimited
Web proxy	24h trial	-	yes	yes	yes	yes
User manage active sessions	24h trial	1	10	20	50	unlimited
Number of KVM guests	none	1	unlimited	unlimited	unlimited	Unlimited



Obr. 2. WinBox (licence)

1.2.2 Konfigurace

Nenastavený RouterBoard běžící pod operačním systémem Mikrotik RouterOS, který lze konfigurovat buď pomocí příkazové řádky připojením přes sériovou linku nebo pomocí programu WinBox, který umožňuje vyhledávat všechny aktivní RouterBoardy připojené skrz ethernet pomocí MAC adresy. [1] [3] [7]

V případě, že provedeme základní nastavení RouterBoardu jako je IP adresa, maska sítě, tak se nám nabízejí další možnosti konfigurace. Lze využít možnost konfigurace RouterBoardu pomocí webového rozhraní. Další možností je pomocí příkazové řádky přes SSH protokol. [7]

2 LINUX

Operační systém Linux je prací finského programátora Linuse Torvaldse, který ji v roce 1991 vytvořil na základě operačního systému UNIX (při vývoji bylo vycházeno ze standardů BSD (Berkeley Software Distribution) a UNIX system V, která je dodnes vyvíjena v Unixových laboratořích společnosti AT&T) a zveřejnil na Internetu pod licencí GNU/GPLv2 . [8] [10] [11]

Pojem Linux ve skutečnosti představuje jádro dnešních Linuxových operačních systémů, které zodpovídá za správu nad procesorem, pamětí, harddisky a jinými perifériemi. Zato obslužné programy, které vznikly modernizací původních programů vytvořených pro UNIX, byly většinou vytvořeny pod záštitou projektu GNU. Soubor Linuxového jádra a těchto malých programů se nazývá Linuxová distribuce, která se správně označuje jako GNU/Linux. Těchto distribucí je v současnosti nepřehledné množství (<http://lwn.net/Distributions/> existuje v současnosti 595 distribucí, které se liší v různých maličkostech). [8] [10] [11]

Společenství lidí okolo Linuxu stále pracuje na jeho vývoji. Podle [12] vychází nová verze jádra přibližně co 2,7 měsíce. Na jeho vývoji se podílí především společnosti Red Hat, Novell, IBM a Oracle. V případě, že se objeví nová periférie, tak má Linux dostatečnou základnu dobrovolníků ale i profesionálů (podle [12] je v současnosti 70-95% vývojářů pracujících na jádru Linuxu profesionálové, tudíž neplatí mýtus, že by byl Linux vyvíjen převážně dobrovolníky), kteří vytvoří pro dané zařízení ovladač pro Linux. [8] [12]

2.1 Distribuce Linuxu používané na routovacích zařízeních

2.1.1 OpenWRT

Jedná se o Linuxovou distribuci určenou především na routery na embedded zařízeních (jednouúčelová zařízení ve kterém je zabudován řídicí počítač, který zařízení spravuje), která vznikla v roce 2004, když firma Linksys zveřejnila zdrojové kódy firmware (jejich firmware byl založen na kódu pod licencí GNU, takže Linksys museli firmware zveřejnit pod stejnou licencí). [16] [14]

Původně byl OpenWRT určen pouze pro směrovače ze série WRT54G, ale po čase se rozšířila podpora okolo sta platform. OpenWRT lze konfigurovat buď pomocí webového rozhraní LuCI nebo pomocí CLI v BusyBoxu. [15] [16] [14]

OpenWRT vychází ve verzích, které jsou pojmenovány podle alkoholických koktejlů, na které je recept v baneru po spuštění systému. Pro zajímavost recept na poslední stabilní verzi, která vyšla v říjnu 2014 Barrier Breaker je 1,5 cl Galliana, 4,5 cl černého rumu, 1 cl kakaa a 12 cl kávy. [15] [16]

2.1.2 DebWRT

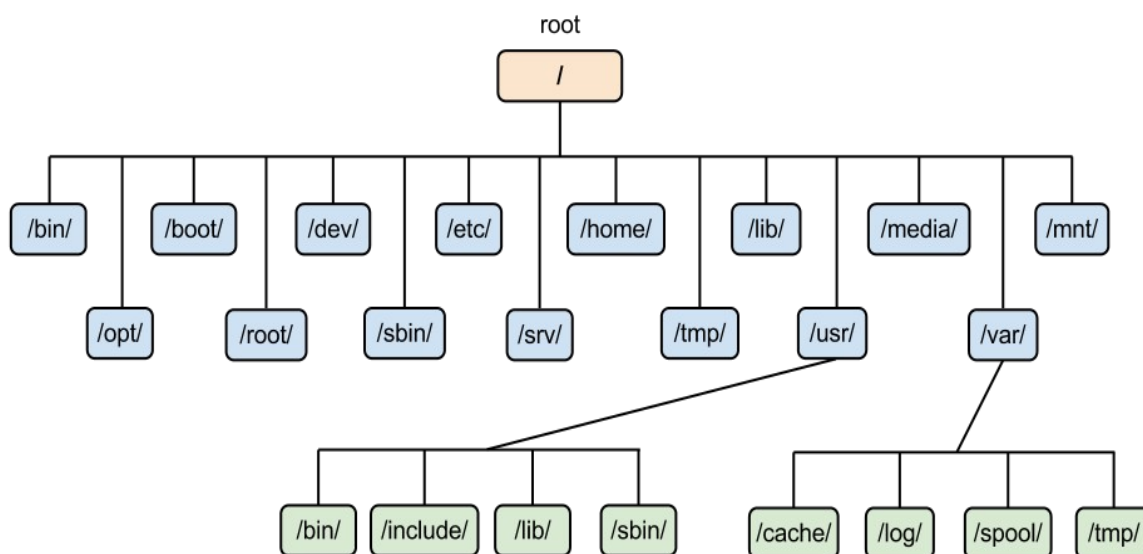
DebWRT je linuxová distribuce určená pro routovací zařízení. Tato distribuce je kombinací distribucí OpenWRT a GNU/Debian. DebWRT využívá firmware z OpenWRT a výhodu balíčkovacího systému Debianu dpkg. Díky tomuto balíčkovacímu systému má DebWRT přístup k rozsáhlým repositářům Debianu. Bohužel podpora DebWRT je takřka nulová, o čem svědčí fakt, že za poslední dva roky nepřišla ze strany vývojářů žádná změna. Tato distribuce lze využít především pro zařízení s architekturou procesoru mipsel a ve výjimečných případech také pro některé zařízení s architekturou procesoru mips. [17] [18]

2.1.3 Linux Voyage

Linux Voyage je derivát distribuce GNU/Debian. Stejně jako výše zmíněný DebWRT i linux Voyage využívá repositáře a balíčkový systém z Debianu. Tento systém je určený na desky s architekturou procesoru x86, především desky od firem PC Engines ALIX/WRAP a Soekris Engineering inc. [19]

2.2 Linux Filesystem Hierarchy Standard (FHS)

Vzhledem k tomu, že je Linux k dostání ve skoro šestistech distribucích, tak musejí vývojáři Linuxu sjednotit základní hierarchii adresářů Linux do standardu FHS. Díky tomuto standardu nemají vývojáři problémy s implementací programového vybavení do různých distribucí. [8] [11]



Obr. 3. Typická struktura adresářů v operačním systému Linux [9]

Nejvýše postaveným adresářem v hierarchickém systému Linuxu je kořenový adresář označovaný jako *root*, který se zapisuje pomocí symbolu `/`. [8] [11]

V adresáři **/bin** se nacházejí důležité spustitelné soubory, které zastupují základní příkazy nutné pro fungování systému. V adresáři **/boot** se nachází jádro systému označené názvem začínající na **vmlinuz**., dále se zde nacházejí soubory nutné pro zavedení systému. Adresář **/dev** slouží k uchování souborů zařízení. V **/etc** se nachází strojově závislá konfigurace. Administrativní, konfigurační a jiné systémové soubory jsou uloženy v tomto adresáři (např. `/etc/passwd` obsahuje seznam uživatelů s oprávněním do systému). Adresář **/home** slouží jako domovský adresář pro uživatele. Přesněji řečeno domovský adresář každého uživatele je adresář vytvořený v adresáři **/home**. V některých systémech může být označení **/home** zaměněno za **/inhouse** nebo **/clients**. Adresář **/lib** slouží pouze pro knihovny a součásti kompilátoru jazyka C. **/media** obsahují souborové systémy odstranitelných médií. Adresář **/mnt** slouží k připojení lokálních souborových systémů podle potřeby. Tento adresář není vhodné využívat pro instalaci programů. Ve starších distribucích **/mnt** zastupuje funkce výše zmíněného adresáře **/media**. **/opt** je nepovinný adresář určený pro doplňkové balíčky s aplikačním softwarem. V adresáři **/proc** se nachází virtuální souborový systém s informacemi o jádrech a procesech. Adresář **/root** slouží jako domovský adresář uživatele *root*. V adresáři **/sbin** se nacházejí obslužné programy určené pro start, opravu a zotavení systému. Adresář **/tmp** slouží programům pro dočasné a pracovní soubory. Tyto soubory se po restartu počítače likvidují. Adresář **/usr** tradičně obsahuje soubory, které

obsahují informace používané systémem v adresáři ***/usr/bin*** obsahuje standardní spustitelné programy operačního systému Linux, které nejsou potřebné v jednovýživatelském módu, podadresář ***/usr/lib*** obsahuje knihovny a podpůrné soubory pro obslužné programy, ***/usr/local*** obsahuje lokální software, tedy software, který uživatel nainstaluje a adresář ***/usr/sbin*** obsahuje soubory pro správu systému, které jsou méně důležité. A posledním důležitým adresářem je ***/var***, který obsahuje proměnlivé soubory (tedy data, které se za běhu operačního systému mění) a data určené specifické pro daný systém. Nacházejí se zde systémové logy (***/var/log***), další prostor pro přechodné soubory (***/var/tmp***), soubory elektronické pošty (***/var/mail***) atd. Některé Linuxové distribuce již nemusejí používat různé podadresáře. [8] [11] [20]

2.3 Bezpečnost systému

2.3.1 Viry a červi

Oproti operačnímu systému Windows je operační systém Linux proti virům povětšinou imunní. Což je do jisté míry tím, že v současnosti neexistuje mnoho známého škodlivého softwaru cíleného přímo na operační systém Linux což si lze vysvětlit tím, že systém Linux je od základu bezpečný díky omezení práv pro běžného uživatele. Druhé vysvětlení je mnohem snadnější a říká, že operační systém Linux má na trhu oproti operačnímu systému Microsoft Windows značně méně uživatelů (jedná se o uživatele desktopu, u serverů je tomu přesně naopak) a tím pádem není Linux pro tvůrce virů, tak atraktivní. [11]

Nejspíš nejzásadnějším důvodem je fakt, že řadový uživatel Linuxu nemá možnost zasahovat nikde mimo svůj adresář, který bývá ve většině Linuxových distribucí umístěn v ***/home/jmeno_úctu***. Uživatel nemá bez oprávnění root možnost zápisu a spouštění souborů mimo svůj přidělený adresář. Tím pádem pokud by se přes uživatele dostal do systému vir, tak by měl omezené pravomoce pouze na uživatelský adresář. Navíc každá služba má svůj vlastní uživatelský účet. [11] [21] [22]

2.3.2 Trojské koně

Trojské koně jsou části kódu nebo samostatné programy, které vykonávají činnost o kterou nemá uživatel zájem. Takovým příkladem byl program od firmy Sony na ochranu před kopírováním na mnoha audio CD v letech 2004-2005. Tento program měl za úkol poškodit uživatele sdílející hudbu na operačních systémech Windows, tím že do nich vyvrtal díry, které mohly být zneužity červy a viry. [11]

V případě Linuxu se jednou za čas objeví trojský kůň v některém z balíčků. V historii se jednalo například o sendmail, tcpDump, OpenSSH. Tyto škodlivé kódy umožňovaly vstupovat do systémů obětí, ale vzhledem k rozsáhlosti Linuxové komunity a faktu, že velká část těchto balíčků je pod licencí GPL, byly tyto kódy během několika dní nalezeny a odstraněny. [11]

II. PRAKTICKÁ ČÁST

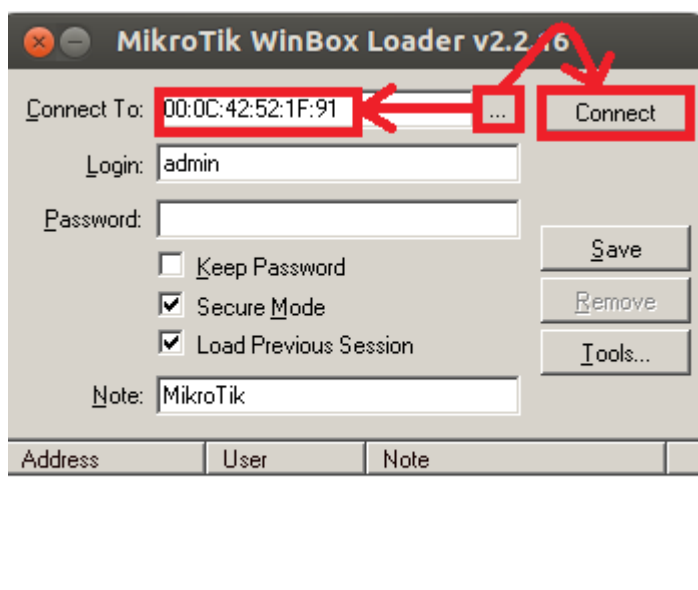
3 UPGRADE MIKROTIK ROUTEROS, ROUTERBOOTU A ZÁLOHA SYSTÉMU

Před vlastní prací byl zadán požadavek na aktualizaci Mikrotik RouterOS, RouterBootu a zálohování klíče pro případ, že by se Mikrotik RouterOS během práce poškodil, případně smazal.

3.1 Připojení RouterBoardu přes WinBox

Přes sériové rozhraní je nutné nastavit, aby se RouterBoard bootoval z NAND paměti, kde je uložen operační systém Mikrotik RouterOS. Poté spojit pomocí kabelu RJ-45 RouterBoard a počítač. [3] [4]

Kliknutím na tlačítko označené ... (červeně zvýrazněné) se nám zobrazí seznam MAC adres připojených aktivních RouterBoardů. Přihlášení do defaultního účtu pro Mikrotik RouterOS je username: admin a kolonka password: zůstává prázdná. Pomocí tlačítka Connect se uživatel přihlásí do operačního systému Mikrotik RouterOS, který se v těchto bodech bude využívat. [23]



Obr. 4. WinBox přihlášení

3.2 Upgrade Mikrotik RouterOS

Pro aktualizaci Mikrotik RouterOS je nutné nejprve zjistit na kterou verzi lze tento systém upgradovat.

3.2.1 Zjištění aktuální verze a nejvyššího možného upgradu

V připojeném WinBoxu na RouterBoard se nejprve zjistí informace o licenci, které se nacházejí v *System => License*, kde uživatel zjistí Software ID, maximální verzi na kterou lze systém aktualizovat a úroveň licence. [23] [4]

3.2.2 Samotná instalace

Na stránkách <http://www.mikrotik.com/download> si uživatel podle série RouterBoardu (série znamená první číslo v označení) vybere architekturu a poté si vybere verzi ke stažení, podle kapitoly 3.2.1. [4]

Přes Winbox zkopíruje rozbalené instalační soubory do modulu *Files*, který slouží jako uložisko pro upgrade Mikrotik RouterOS, RouterBootu, zálohy konfigurace RouterBoardu a k ukládání routovacích tabulek. [23]

Po zkopírování do RouterBoardu stačí restartovat RouterBoard modulem *System => Reboot*. Po restartu se stačí znova přihlásit do systému přes WinBox a zkontrolovat

Licenci. Pokud lze Mikrotik RouterOS ještě aktualizovat, tak celý postup stačí opakovat. [4]

3.3 Upgrade RouterBootu

Stejně jako při aktualizaci Mikrotik RouterOS, tak i zde je nutno nejprve zjistit informace o biosu RouterBoardu, tedy RouterBootu. Tyto informace lze získat z CLI, ke které se dá dostat přes modul ve WinBoxu *Telnet*. Zde stačí napsat příkaz: */system RouterBoard print* a uživatel se dovídá aktuální verzi RouterBootu a o nejvyšší verzi, která je dostupná. [24] [25]

Ze stránek *www.RouterBoard.com*, kde si stačí vybrat váš RouterBoard, stačí stáhnout aktuální RouterBoot. Ten se přepokopíruje přes WinBox do modulu *Files*. Poté stačí napsat v CLI příkaz: */system RouterBoard upgrade* a RouterBoard se restartuje a aktualizuje.

[25]

3.4 Stažení licenčního klíče

Klíč lze získat a stáhnout do počítače například ve WinBoxu v modulu *System => License*, kde je tlačítko *Export Key...* V případě, že by uživatel Mikrotik RouterOS smazal a opět nainstaloval, tak pomocí tlačítka *Import Key...* vrátí licenční klíč do systému. [4]

4 INSTALACE LINUXU

Vzhledem k tomu, že RouterBoard 450G nedisponuje žádným USB slotem a ani slotem pro CompactFlash kartu a jediné přenosné médium pro přenos dat je SD karta, kterou ale nedovede zavést RouterBoot, tudíž je jedinou možností jak nabootovat vlastní operační systém pomocí ethernetu.

4.1 Nastavení sériové linky

Byla nastavena sériová komunikace v programu hyperterminal podle těchto parametrů: [3]

Název:	mikrotik
Připojit pomocí:	COM7
Bity za sekundu:	115 200
Datové bity:	8
Parita:	žádná
Stop-bity:	1
Řízení toku:	Hardware

4.2 Nastavení bootování v RouterBootu

Pro změnu bootování se musí uživatel připojit k RouterBoardu před zaváděním Mikrotik RouterOS přes sériové rozhraní stiskem libovolné klávesy.

V jednoduchém menu uživatel zvolí *boot device*, který bývá pod klávesou *o*. Zde je potřeba nastavit, aby RouterBoot zaváděl operační systém ze sítě, tedy volbu *e - boot over Ethernet*. Po nastavení bootovacího média je ještě potřeba nastavit protokol bootování, který se ukrývá pod volbou *boot protocol*, zde nastavíme bootování pomocí protokolu DHCP, tedy *2 - DHCP protocol*. [26]

4.3 Nastavení DHCP serveru

Na počítač ze kterého byl zaváděn prozatímní operační systém do RouterBoardu byl nainstalován balík obsahující DHCP server. Tomuto serveru byl vytvořen adresář v kořenovém adresáři a byl nastaven konfigurační soubor `/etc/dhcp/dhcp.conf`. Pro přenos zaváděcího obrazu se použil protokol TFTP.

```
sudo apt-get install isc-dhcp-server
mkdir /tftpboot
cat /etc/dhcp/dhcpd.conf
...
default-lease-time 21600;
max-lease-time 2100;
allow booting;
subnet 10.0.0.0 netmask 255.255.255.0 {
    range 10.0.0.200 10.0.0.250;
    option broadcast-address 10.0.0.255;
    option routers 10.0.0.138;
    option domain-name-servers 10.0.0.138;
    filename "vmlinux.elf";
    group {
        next-server 10.0.0.1;
        host tftpclient {
            filename "vmlinux.elf";
        }
    }
    ...
sudo gedit /etc/default/atftpd
USE_INETD=false
OPTIONS="--tftpd-timeout 300 --retry-timeout 5 --mcast-port 1758 --mcast-addr
239.239.239.0-255 --mcast-ttl 1 --maxthread 100 --verbose=5 /tftpboot"
```

4.4 Nastavení SSH serveru

Posílání souborů mezi RouterBoardem a počítačem probíhá pomocí protokolu SSH, který není ve většině Linuxových distribucích nainstalovaný. [26] [27]

Pomocí byl nainstalován ssh server na počítači:

```
sudo apt-get install openssh-server openssh-client
sudo etc/init.d/ssh start
```

Pro vysvětlení v dalších bodech může uživatel posílat soubory na RouterBoard a zpět pomocí příkazu:

```
scp <adresa souboru> <cílový adresář>
```

Kde adresář na vzdáleném serveru se označuje syntaxí `účet_na_počítači@ip_adresa:/cesta`

4.5 Záloha disků s Mikrotik RouterOS

Ač bylo v této práci předvedeno jak zálohovat klíč na licenci Mikrotik RouterOS, tak podle doporučení [27] se v této kapitole byly zkopírovány oddíly v RouterBoardu.

NVRAM RouterBoardu je rozdělena na 7 oddílů, kde pro tuto práci jsou důležité pouze oddíly pro jádro systému a systémové soubory.

```
# cat /proc/mtd
dev:          size          erasesize      name
mtd0:         0000b000      00001000      routerboot
mtd1:         00001000      00001000      hard_config
mtd2:         00002000      00001000      bios
mtd3:         00001000      00001000      soft_config
mtd4:         00040000      00040000      booter
mtd5:         003c0000      00040000      kernel
mtd6:         3fc00000      00040000      rootfs
```

Tyto sekce lze zálohovat příkazy v nabootovaném prozatímním systému OpenWRT, z předchozího kapitoly.

```
# cd /tmp
# dd if=/dev/mtd5 | gzip > routeros_kernel.img.gz
# dd if=/dev/mtd6 | gzip > routeros_rootfs.img.gz
# scp routeros_kernel.img.gz user@10.0.0.1:/tftpboot
# scp routeros_rootfs.img.gz user@10.0.0.1:/tftpboot
```

Pokud by chtěl uživatel pomocí této zálohy vrátit systém musel by napsat tuto skupinu příkazů (jedná se vlastně o obrácený postup jako při zálohování): [27]

```
# cd /tmp
#scp ubuntu@10.0.0.1:/tftpboot/routeros_kernel.img.gz .
#scp ubuntu@10.0.0.1:/tftpboot/routeros_rootfs.img.gz .
#zcat routeros_kernel.img.gz | dd of=/dev/mtd5
#zcat routeros_rootfs.img.gz | dd of=/dev/mtd6
```

4.6 Uložení Linuxu do RouterBoardu (OpenWRT)

Do adresáře určeného pro tftp server (desktop připojený k RouterBoardu přes ethernet) bylo překopírováno jádro systému (kernel) a zabalené systémové soubory, které byly získány ze stránek distribuce [<https://OpenWRT.org/>]. Tyto dva soubory (vmlinux.efi a rootfs.tar.gz) byly do paměti RouterBoardu nataženy pomocí příkazu *scp* (viz. Níže), tyto soubory byly vloženy do příslušných diskových oddílů v NVRAM, konkrétně do bloku 5 (blok určený pro jádro systému) a bloku 6 (blok určený pro systémové soubory). Poté byl RouterBoard restartován příkazem *reboot*. [26] [27]

Pomocí této sady příkazů byl uložen operační systém OpenWRT do RouterBoardu na příslušné diskové oddíly:

```
# cd /tmp
# scp ubuntu@10.0.0.1:/tftpboot/OpenWRT/vmlinux.efi .
# scp ubuntu@10.0.0.1:/tftpboot/OpenWRT/rootfs.tar.gz .
# mtd erase kernel
# mount -t yaffs /dev/mtdblock5 /mnt
# cp vmlinux.elf /mnt/kernel
# umount /mnt
# mtd erase rootfs
# mount -t yaffs /dev/mtdblock6 /mnt
# cd /mnt
# tar xpf /tmp/rootfs.tar.gz .
# cd /tmp
# umount /mnt
# reboot
```

Po restartu bylo změněno v RouterBootu bootování zpět na NVRAM a nový operační systém OpenWRT byl spuštěn.

4.7 Instalace driverů pro SD kartu

Základní balíky neobsahovaly drivery pro čtečku SD karet. Proto byly nainstalovány drivery z repositářů příkazem *opkg install*. [4]

```
root@OpenWRT:/#opkg update
root@OpenWRT:/#opkg install kmod-fs-ext4 e2fsprogs cfdisk kmod-nls-base kmod-nls-
cp437 kmod-nls-iso8859-1
root@OpenWRT:/#reboot
```

4.7.1 mount/unmount SD karty

Mountování karty probíhá příkazem:

```
root@OpenWRT:/#mount /dev/mmcblk0p1 /mnt/
```

Odmountování příkazem:

```
root@OpenWRT:/#umount /mnt/
```

4.7.2 Mountování karty po startu systému

Podle [4] byl aplikován postup na mountování karty pomocí programu block-mount. Byl stažen balíček z repositářů OpenWRT s názvem mount-block. Tento program umí nalézt všechny diskové zařízení a namountovat je. Vzhledem k tomu, že block-mount vyžaduje pro svoji funkčnost soubor *fstab* v adresáři */etc/config/* na rozdíl od jiných programů, které pracují se stejným souborem v adresáři */etc/*, proto byl smazán soubor */etc/fstab* a vytvořen

symbolický odkaz v tomto adresáři se souborem *fstab* v adresáři */etc/config/*. Dále byly do *fstabu* přidány zařízení, které vyhledal jeden z programů obsažených v balíku *mount-block*. Tento program vyhledal pouze dva oddíly na SD kartě. V adresáři */etc/init.d/* byl aktivován obslužný script nainstalovaný s *block-mountem* s názvem *fstab*. Pomocí parametru *enable* byl vytvořen symbolický odkaz na tento script v adresáři */etc/rc.d* podle scriptu (pokud skript obsahuje *START* znamená to, že se tento script bude spouštět při zapnutí systému, pokud obsahuje *STOP* znamená to, že se příkaz aktivuje při ukončování systému. Číslo u těchto dvou parametrů znamená prioritu scriptu, čím nižší, tím je priorita vyšší).

```
root@OpenWRT:/#opkg update
root@OpenWRT:/#opkg install block-mount
root@OpenWRT:/#rm /etc/fstab
root@OpenWRT:/#ln -s /etc/config/fstab /etc/fstab
root@OpenWRT:/#block detect > /etc/fstab
root@OpenWRT:/#/etc/init.d/fstab enable
root@OpenWRT:/#reboot
```

Po restartu systém naběhl a dokonce byla i namountovaná SD karta v příslušném adresáři nastaveném v konfiguračním souboru */etc/fstab*, ale program, který má na starost mountování po startu systému se snažil namountovat i zařízení, které v konfiguračním souboru */etc/fstab* nebyly obsaženy (diskové oddíly v NVRAM). Pokusy *block mountu* namountovat oddíly, které nebyly evidované v souboru */etc/fstab* zapříčinily, že při startu systému jádro nahlásilo přes čtyřicet chybových hlášení o neúspěšný pokus tyto oddíly namountovat. Navíc se start systému zpomalil o 5 vteřin, proto za účelem předejít mountu diskových odílů uložených v NVRAM byly do souboru */etc/fstab* tyto oddíly přidány ručně pomocí textového editoru vi a přidán k nim parametr *enabled=0*.

```
root@OpenWRT:/#vi /etc/fstab
root@OpenWRT:/#reboot
```

Po restartu systému se vůbec nic nezměnilo, protože ač si program *block mount* bere data konfigurační data z nakonfigurovaného adresáře, tak chybu nevypisuje přímo *mount block*, ale další program z balíčku *mount list*, který slouží k vyhledání všech diskových zařízení na desce a získání informací o nich, které předá *block mountu*. S tohoto důvodu byl celý balíček *block-mount* odstraněn a jako náhrada byl vytvořen vlastní bash script na základu kapitoly 5.6.1, takže pokud chce uživatel automaticky mountovat svoji SD kartu, tak stačí aby v souboru */etc/init.d/sd_boot* přidat parametr *enable*, v opačném případě parametr *disable*

```
root@OpenWRT:/#cp /etc/init.d/fstab /etc/init.d/sd_boot
root@OpenWRT:/#opkg remove block-mount
```

```
root@OpenWRT:/#vi /etc/init.d/sd_boot
root@OpenWRT:/#cat /etc/init.d/sd_boot
#!/bin/bash /etc/rc.common

START=40

start(){
    mount /dev/mmcblk0p1 /mnt/mmc
}
stop(){
    /sbin/block umount
}
root@OpenWRT:/#/etc/init.d/sd_boot enable
root@OpenWRT:/#reboot
```

5 KONFIGURACE LINUXU

5.1 Nastavení hesla pro uživatele root

Bylo nastaveno heslo pro uživatele *root* pomocí příkazu *passwd*: [4]

```
root@OpenWRT:/#passwd root
Changing password for root
New password:
Bad password: too weak
Retype password:
Password for root changed by root
```

Jedním z bezpečnostních prvků operačního systému GNU/Linux je to, že na konzoli nelze při zadávání hesla vidět počet napsaných znaků (viz. řádek 3 a 5). Heslo uživatele *root*, lze změnit i bez napsání parametru *root*.

```
root@OpenWRT:/#passwd
Changing password for root
New password:
Bad password: too short
Retype password:
Password for root changed by root
```

Pokud je heslo menší jak 6 znaků program uživatele informuje, že je heslo krátké (heslo musí obsahovat alespň jeden znak jinak je neplatné).

5.2 Nastavení hostname

Bylo změněno hostname z *OpenWRT* na *RB450G* pomocí příkazů: [4]

```
root@OpenWRT:/#cat /etc/config/system
config system
    option hostname OpenWRT
    option timezone UTC
...
root@OpenWRT:/#vi /etc/config/system
root@OpenWRT:/#reboot
...
root@RB450G:/#
```

V textovém editoru *vi* bylo změněno hostname RouterBoardu z *OpenWRT* na *RB450G* a uloženo. Tato změna se projevila až po restartování systému příkazem *reboot*.

5.3 Vytvoření nového uživatelského účtu

Byl vytvořen nový uživatelský účet, pro možnost připojení do systému, protože z důvodu bezpečnosti bylo uživatelské jméno *root* zakázáno pro připojení. Pro vytvoření dalších účtů byla stažena z repositářů aplikace *shadow-useradd*. [4]

```
root@RB450G:/#opkg update
root@RB450G:/#opkg install shadow-useadd
root@RB450G:/#useradd login
root@RB450G:/#passwd login
Changing password for login
New password:
Bad password: too week
Retype password:
Password for login changed by root
root@RB450G:/#mkdir home
root@RB450G:/#cd /home
root@RB450G:/#mkdir login
root@RB450G:/#vi /etc/passwd
root@RB450G:/#cat /etc/passwd
...
login:x:1000:1000:login:/home/login:/bin/ash
```

Z důvodu aby mohl neprivilegovaný účet provádět změny v systému s pověřením uživatele *root*, byl stažen z repositářů balíček *sudo* (viz kapitola 6.3), který umožňuje komukoli v systému s tímto příkazem provádět operace *roota*.

```
root@RB450G:/#opkg install sudo
```

Pro příklad uživatel bude moci provádět operace superuživatele pomocí napsání *sudo* před příkaz. Před vyplněním příkazu se systém zeptá uživatele na heslo superuživatele (*roota*) a v případě shody se požadavek vyplní.

```
root@RB450G:/#sudo opkg install ...
```

5.4 Upgrade software

Byl pomocí příkazu *opkg* zjištěn seznam nainstalovaného software, který je možné upgradovat. Posléze byl tento software upgradován. Příkaz *opkg upgrade* lze využít pro zkrácení tak, že se do jednoho řádku napíší všechny programy pro aktualizaci.

```
root@RB450G:/#opkg list-upgradable
...
root@RB450G:/#opkg upgrade wpad-mini
root@RB450G:/#opkg upgrade odhcpd6c
root@RB450G:/#opkg upgrade wpad-mini
root@RB450G:/#opkg upgrade hostapd-common
```

5.5 Nastavení síťových rozhraní

Byly nastaveny dvě síťové rozhraní, jedno pro eth0 (wan) a druhé pro eth1 (lan). Rozhraní lan má statickou IP adresu 192.168.1.1 s maskou 255.255.255.0 a rozhraní wan má nastaveno získávání IP adresy ze serveru (modemu) pomocí DHCP klienta. [4]

```
root@RB450G:~#vi /etc/config/network
root@RB450G:~#cat /etc/config/network
```

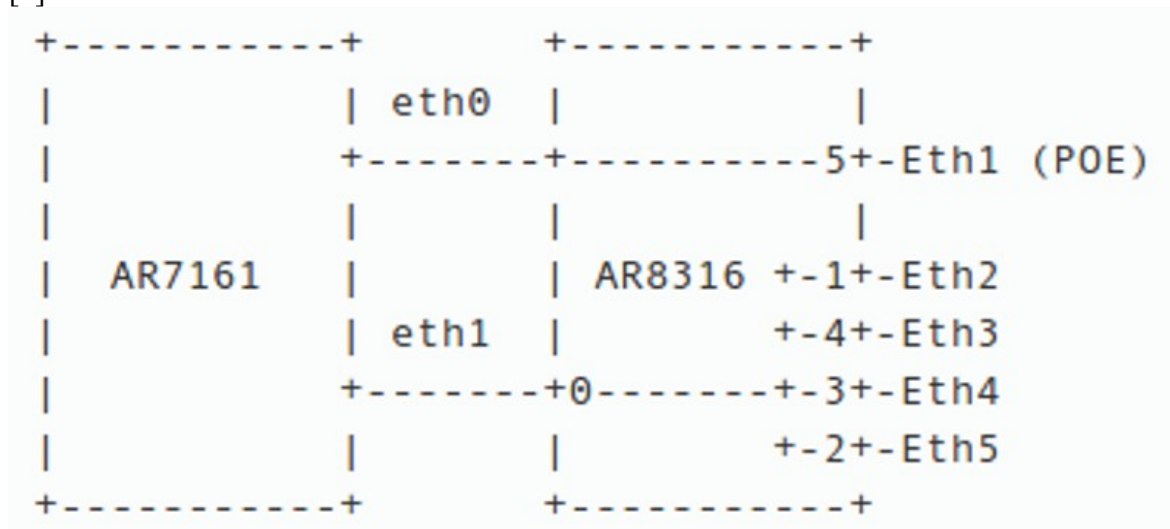
...

```
config interface 'lan'
    option ifname 'eth1'
    option force_link '1'
    option type 'bridge'
    option proto 'static'
    option ipaddr '192.168.1.1'
    option netmask '255.255.255.0'
```

```
config interface 'wan'
    option ifname 'eth0'
    option proto 'dhcp'
```

5.5.1 Popis řešení RouterBoardu 450G

AR7161 má připojeny pouze dva řadiče ethernetu, což vysvětluje proč byly vytvořeny v předchozím bodu pouze dvě síťové rozhraní pro 5 ethernetových portů. Rozhraní eth0 slouží pro připojení do sítě wan a rozhraní eth1 slouží pro připojení do switchu viz. obrázek. [4]



Obr. 5. Popis síťové konfigurace RouterBoardu [4]

Podle [4] [27] není dostupný firmware, pomocí kterého by bylo možné nastavovat vlastní síťové rozhraní pro ethernetové porty na AR8316. Tyto porty lze částečně konfigurovat tím, že se nastaví rozhraní pod eth1 jako switch (zde lze využít alespoň možnosti konfigurace přes protokol 802.1q). Pro možnost konfigurace switchu přes LuCI byla přidána část konfigurace získána z [4].

```
root@RB450G:~#vi /etc/config/network
root@RB450G:~#cat /etc/config/network
...
config switch
    option reset 1
    option enable_vlan 1
...
```

Z [52] bylo zkopírováno řešení směrování pomocí protokolu 802.1q. Zde se nachází 6 síťových rozhraní. Síťové rozhraní wan využívá protokolu PPPoE a zbylé rozhraní protokol VLAN. U switch vlan možnost option ports znamená, které porty daný VLAN využívá (port 0 je v tomto případě jádro systému, znak t za číslem portu symbolizuje taggování). Taggování slouží k rozšíření rámce o informaci, že se jedná o protokol 802.1q a informace důležité pro routování. Vzhledem k tomu, že se změnil rámec, je potřeba aby byl změněn také kontrolní součet. [52] [53]

```
”
config 'interface' 'loopback'
    option 'ifname' 'lo'
    option 'proto' 'static'
    option 'ipaddr' '127.0.0.1'
    option 'netmask' '255.0.0.0'

config 'interface' 'wan'
    option 'ifname' 'eth0'
    option 'proto' 'pppoe'
    option 'username' 'xxxxxxx'
    option 'password' 'xxxxxxx'

config 'interface' 'admin'
    option 'proto' 'static'
    option 'ifname' 'eth1.101'
    option 'ipaddr' '10.1.1.1'
    option 'netmask' '255.255.255.0'

config 'interface' 'user'
    option 'proto' 'static'
    option 'ifname' 'eth1.102'
    option 'ipaddr' '10.1.2.1'
    option 'netmask' '255.255.255.0'
```



```
config 'interface' 'voip'  
    option 'proto' 'static'  
    option 'ifname' 'eth1.103'  
    option 'ipaddr' '10.1.3.1'  
    option 'netmask' '255.255.255.0'
```

```
config 'interface' 'guest'  
    option 'proto' 'static'  
    option 'ifname' 'eth1.104'  
    option 'ipaddr' '10.1.4.1'  
    option 'netmask' '255.255.255.0'
```

```
config 'interface' 'dmz'  
    option 'proto' 'static'  
    option 'ifname' 'eth1.105'  
    option 'ipaddr' '10.1.5.1'  
    option 'netmask' '255.255.255.0'
```

```
config 'switch'  
    option 'name' 'eth1'  
    option 'reset' '1'  
    option 'enable_vlan' '1'
```

```
config 'switch_vlan' 'vlan_admin'  
    option 'device' 'eth1'  
    option 'vlan' '101'  
    option 'ports' '0t 1t 2'
```

```
config 'switch_vlan' 'vlan_user'  
    option 'device' 'eth1'  
    option 'vlan' '102'  
    option 'ports' '0t 1t'
```

```
config 'switch_vlan' 'vlan_voip'  
    option 'device' 'eth1'  
    option 'vlan' '103'  
    option 'ports' '0t 1t 3'
```

```
config 'switch_vlan' 'vlan_guest'  
    option 'device' 'eth1'  
    option 'vlan' '104'  
    option 'ports' '0t 1t'
```

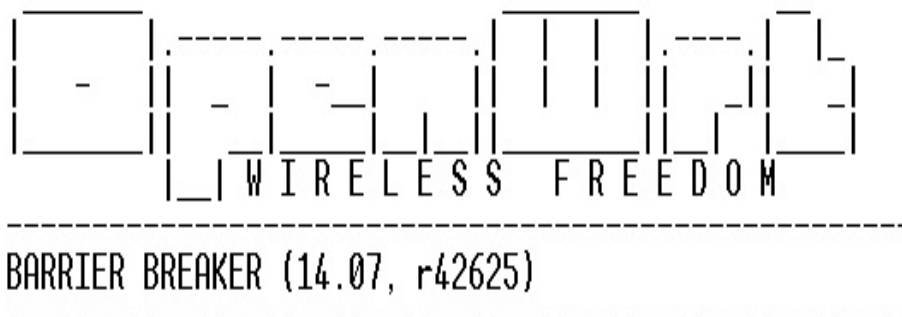
```
config 'switch_vlan' 'vlan_dmz'  
    option 'device' 'eth1'  
    option 'vlan' '105'  
    option 'ports' '0t 1t 4'
```

„ [52]

5.6 Editace úvodního banneru

Byl zkrácen banner operačního systému OpenWRT, který se nachází v souboru `/etc/banner`, pouze na název systému a jeho verzi v textovém editoru *vi*. Tento banner se zobrazuje uživatelům po přihlášení do systému.[4]

```
root@RB450G:/#vi /etc/banner
root@RB450G:/#cat /etc/banner
```



Obr. 6. Upravený Baner

6 POPIS SLUŽEB, JEJICH INSTALACE A KONFIGURACE

6.1 BusyBox

Jedná se o projekt Bruce Perense, který se pokoušel vytvořit bootovatelný Linuxový systém o velikosti jedné diskety (tento projekt vycházel z faktu, že jádro Linuxu se skládá z velkého množství malých binárních souborů, kde každý obsahuje až kilobajty nadbytečných dat). BusyBox je nyní plnohodnotný příkazový procesor sloužící pro operační systémy UN*X. Nevýhodou BusyBoxu je absence oblíbeného příkazu `man` (manuálové stránky pro obsluhu podprogramů jenž BusyBox obsahuje). Tento příkazový interpret má v sobě implementovány základní programy pro chod linuxu: [28] [29]

6.1.1 `init`

Jedná se o systémového daemona a vlastně i hlavní proces celého systému s PID 1 (Proces ID). Tento proces má na starost spouštění všech procesů, které si mohou poté samy spouštět své vlastní podprocesy. Když podproces „zahyne“ musí jeho mateřský proces zajistit jeho správné ukončení (v opačném případě se proces změní v zombie). Dále tento proces dohlíží nad tím, který proces se při startu systému spustí dříve a který později. V OpenWRT je `init` skript součástí procesu `procd`. [30] [31]

6.1.2 `ash`

Ash (Almquist SHell označovaný jako A Shell nebo také `sh`) je označení pro textové rozhraní používané v Unixu a Unix-like systémech. Toto rozhraní je spuštěno ihned po startu systému a vytvoří CLI pomocí které uživatel zadává příkazy. Deriváty Ashe jsou instalovány jako výchozí shell například v distribucích FreeBSD, MINIX, Android... [31] [32] [33]

6.1.3 Programy určené pro práci s řetězcí (`awk`, `grep`, `sed`)

BusyBox obsahuje programovací jazyk `awk`, program pro filtraci řetězců `grep` a program určený pro nahrazování textu `sed`. Ve většině případů je vhodnější využívat program `grep`, který nemá tolik možností jako program vytvořený v `awk`, ale má mnohem snadnější a přehlednější syntaxi. Program `grep` má několik alternativních názvů (`egrep`, `fgrep` ...), kde písmeno před názvem programu určuje první parametr programu. `Awk` je programovací jazyk pojmenovaný podle jeho autorů (Alfred V. Aho, Peter J. Wienberger a Brian W. Kernighan). Jeho neznámější implementace je vyvíjená projektem GNU a jmenuje se

gawk. V Unixových systémech je programovací jazyk *awk* používán pro automatickou konstrukci příkazů. Poslední ze skupiny je program *sed*, určený pro náhradu textu. [31] [34] [35]

6.1.4 programy pro práci se soubory (cat, cd, cp, ls, mkdir, mv, pwd, rm, rmdir, touch)

Jedná se o jednoduché programy pomocí kterých se pracuje se soubory a adresáři. Jedná se opravdu o triviální operace jako je výpis souborů v aktuálním adresáři, vytvoření adresáře, odstranění adresáře atd.

- *cat* – tento program slouží k výpisu souboru do okna terminálu, nebo do souboru (technicky vzato se pak jedná o kopírování)
- *cd* – slouží k procházení adresářů
- *cp* – slouží ke kopírování
- *ls* – vypíše soubory a složky v aktuálním adresáři
- *mkdir* – vytvoří v aktuálním adresáři nový adresář
- *mv* – přesouvá souborem
- *pwd* – vypíše na konzoli aktuální adresář ve kterém se nachází uživatel
- *rm* – slouží k odstranění souboru (pomocí parametru *-rf* lze mazat i adresáře)
- *rmdir* – odstraní adresář
- *touch* – slouží v první řadě ke změně data poslední úpravy souboru, ale lze také využít k vytváření prázdných nových souborů (pokud uživatel zadá soubor, jenž neexistuje, tak se soubor vytvoří)

Tyto jednoduché příkazy/programy mají desítky parametrů a možností využití, které lze najít například v manuálových stránkách. [31] [35]

6.1.5 pomocné programy (echo, sleep)

Tyto programy samy o sobě nemají žádné smysluplné využití, a proto je jejich využití pouze v bash scriptech.

- echo – tento program vypíše uživateli textový řetězec
- sleep – pozastaví aktuální proces na dobu ve vteřinách jenž má ve svém parametru

[31] [35]

6.1.6 programy pro připojení zařízení (df, mount, umount)

Program mount slouží k mountování (připojení nového diskového oddílu do systému). Při mountování je potřeba vytvořit adresář ve kterém bude připojený daný diskový oddíl (nejčastěji se pro tuto situaci využívá adresáře FHS /mount nebo /media). Příkaz df uživateli sdělí výpis právě připojených diskových oddílů jejich kapacitu, využití a přípojně místo. Příkazem umount se zařízení odpojí od systému. [31] [35]

6.1.7 programy určené pro archivaci a komprimování souborů (gzip, tar)

Linuxové programy pro komprimování souborů slouží k bezztrátové kompresi souborů, tato komprese je založena na principu Huffmanova kódování, které si podle četnosti opakování znaků vytváří vlastní kód s tím, že frekventovanější znaky mají kratší bitový zápis (nejfrekventovanější znak může mít délku až 1 bit). V případě dekomprese probíhá inverzní postup, kde se podle slovníků dosadí originální kód. [31] [35] [36]

Oproti komprimování archivace nesloží ke snížení velikosti výstupního souboru, ale pouze ke sloučení více souborů či adresářů do jediného souboru. V tomto souboru má samozřejmě každý soubor uloženy své parametry. K archivaci se využívá program tar, který uživateli umožní rovnou při archivaci i komprimovat archiv některým z nainstalovaných komprimačních programů (gzip) [31] [35]

6.1.8 programy určené pro správu procesů (kill, ps, pidof)

Tyto programy umožňují uživateli sledovat procesy v systému a ovládat je. V Linuxu má každý proces své ID, které mu přiřadí systém. Jediný proces, který má vždy identické PID je proces init viz bod 6.1.1. [31] [35]

Uživatel může sledovat běžící procesy pomocí příkazu ps. Samotný příkaz ps bez parametru zobrazí pouze procesy spuštěné z konzole ve které byl příkaz napsán, pro zobrazení

všech procesů v systému je potřeba přidat parametr ax (toto platí na většině distribucí Linuxu, protože je BusyBox odlehčen, tak příkaz ps nemá žádné parametry a vypíše se tak, jako by uživatel napsal na jiné distribuci ps ax). [31] [35]

```
login@RB450G:~$ ps
  PID USER      VSZ STAT COMMAND
    1 root        1388 S    /sbin/procd
    2 root          0 SW    [kthreadd]
    3 root          0 RW    [ksoftirqd/0]
    4 root          0 SW    [kworker/0:0]
    5 root          0 SW<  [kworker/0:0H]
    6 root          0 SW    [kworker/u2:0]
    7 root          0 SW<  [khelper]
    8 root          0 SW    [kworker/u2:1]
   65 root          0 SW<  [writeback]
   68 root          0 SW<  [bioset]
   70 root          0 SW<  [kblockd]
  101 root          0 SW    [kworker/0:1]
  106 root          0 SW    [kswapd0]
  153 root          0 SW    [fsnotify_mark]
  270 root          0 SW<  [deferwq]
  273 root          0 SW    [kworker/0:2]
  425 root         884 S    /sbin/ubusd
  426 root        1364 S    /bin/ash --login
  723 root          0 SW<  [cfg80211]
  798 root        1036 S    /sbin/logd -S 16
  799 root        1180 S    /sbin/logread -f -F /var/log/messages -p /var/run/
  832 root        1544 S    /sbin/netifd
  850 root        1152 S    /usr/sbin/odhcpd
  878 root        1364 S    /usr/sbin/crond -f -c /etc/crontabs -l 8
  893 root        3296 S    /usr/sbin/sshd -D
  960 root        1360 S    /usr/sbin/ntpd -n -l -p ntp.nic.cz
  977 root          0 SW    [kworker/0:3]
 1037 root        2096 S    /usr/sbin/uhttpd -f -h /www -r RB450G -x /cgi-bin
 1044 root        1360 S    udhcpd -p /var/run/udhcpd-eth0.pid -s /lib/netifd/
 1127 nobody       928 S    /usr/sbin/dnsmasq -C /var/etc/dnsmasq.conf -k
 1202 root          0 SW    [kworker/0:4]
 1210 root        5904 S    sshd: login [priv]
 1212 login        5904 R    sshd: login@pts/0
 1213 login        1364 S    -ash
 1218 login        1360 R    ps
```

Obr. 7. Výpis procesů programem ps

První sloupec je PID (již zmíněné ID procesu, pomocí této hodnoty se na procesy ukazuje). Ve druhém sloupci se nachází odkud je proces spuštěn, třetí sloupec zobrazuje využití operační paměti procesem, čtvrtý stav procesu a pátý je příkaz, který proces spustil.[31] [35]

Příkaz kill slouží k řízení procesu (většinou k jeho rychlému ukončení, příkaz kill má své jméno podle signálu 9, který má za úkol proces odstranit). Seznam signálů jenž operační systém podporuje lze zjistit příkazem kill -l. [31] [35] [37]

Tab. 2. Seznam nejznámějších signálů [37]

Číslo	Název	Popis
1	HUP (Hangup)	Tento signál proces obdrží tehdy, když je uzavřen jeho řídicí terminál.
2	INT (Interrupt)	Toto je signál, který proces obdrží, když běží v terminálu a uživatel stiskne Ctrl+C. Obvykle ukončí proces.
3	QUIT	Ukončí proces a запиše stav paměti, se kterou program pracoval (tzv. core dump).
4	ILL (Illegal instruction)	Tento signál posílá operační systém, když proces vyvolá neznámou instrukci.
8	FPE (Floating point exception)	Tímto signálem jádro trestá programy, které se snaží dělit nulou, atp.
9	KILL (Kill)	„Zabije“ proces (okamžitě). Nelze obejít.
10	USR1 (User-defined)	Uživatelsky definovaný signál.
11	SEGV (Segmentation fault)	Obvykle posílá operační systém programům, které chybně pracují s pamětí.
15	TERM (Terminate)	Ukončí proces.
18	CONT (Continue)	Obnoví běh procesu po obdržení některého ze dvou předchozích signálů.
19	STOP	Zastaví proces. Nelze obejít.
20	TSTP (Terminal stop)	Zastaví proces, ale lze obejít. Tento signál proces obdrží, když běží interaktivně v shellu a uživatel stiskne Ctrl+Z.

Dále existuje příkaz `pidof`, který uživateli vrátí PID na základu jména procesu, ale vzhledem k tomu, že název procesu musí být napsán přesně je vhodnější vyhledávat PID procesu kombinací příkazů `ps` a `grep` viz příklad: [31] [35] [37]

```
#ps ax | grep ini
#pidof init
```

6.1.9 chmod

Tento program slouží ke změně práv souboru a složky, které může změnit její majitel nebo uživatel root. Chmod se dá využít buď symbolicky nebo číselně. Při symbolickém zápisu se používá označení kategorie, které se změna práv týká (**u** – user, **g** – group, **o** – other), operace (+ – přidat oprávnění, - – odebrat oprávnění, = – nastaví aktuální parametry) a parametry (**r** – čtení, **w** – psaní, **x** – spuštění u souboru; **r** – čtení názvu obsažených položek, **w** – vytváření nových souborů a adresářů, **x** – vstup do adresáře). V případě číselného zápisu se zápis skládá ze 4 číslic (pokud je zapsáno méně číslic jsou zleva postupně psány nuly aby dorovnály počet číslic), kde první číslo symbolizuje speciální práva, druhá uživatele, třetí skupinu a čtvrtá ostatní. Numerická hodnota je součet práv (4 – čtení, 2 – zápis, 1 – spuštění) viz příklad. [8] [31] [35]

```
#chmod 700 test.sh
#chmod u=rwx, g=, o= test.sh
```

6.1.10 date

Je program sloužící k zobrazení času a data v požadovaném formátu. Dále umožňuje uživateli měnit manuálně čas v systému pomocí parametru -s. [31] [35]

6.1.11 dd

O názvu tohoto programu kolují na internetu desítky fám, zde byly vybrány ty nejčastější a nejpravděpodobnější. Obrázek si může udělat každý sám:

Říká se, že netradiční syntaxe tohoto programu (ve stylu *parametr=hodnota*) je parodie na IBM System / 360 JCL, která měla komplikovaný zápis známý jako “*Dataset Definition*”. [38]

Další alternativou je to, že když uživatel použije špatný parametr, tak může ztratit všechna data na disku s tohoto důvodu se programu *dd* přezdívá “*Disk Dump*”. [38]

„*dd* je program, jehož hlavním účelem je nízkoúrovňové kopírování a konverze surových dat. Původ tohoto nástroje je zvláštní. Název *dd* je „zkratkou“ convert and copy, přičemž někde se lze dočíst, že to není „cc“ (jak by se dalo očekávat) právě proto, že takto se jmenuje kompilátor.“ [38]

Program se ovládá pomocí dvou parametrů (*if* a *of*), kde *if* slouží pro nízkoúrovňové kopírování (input file) a *of* pro zápis (output file). Použití toho příkazu lze vidět v kapitole 4.5. [31] [35]

6.1.12 dmesg

Tento program slouží k výpisu hlášení Linuxového jádra do okna terminálu. Jádro Linuxu zaznamenává výpisy do kruhového bufferu, odtud jsou postupně vyčítány daemoneklogd nebo syslogd a ukládány do souboru */var/log/syslog*. [39]

6.1.13 ln

Vytvoří odkaz na soubor. V Linuxu existují dva typy odkazů pevné a symbolické. Pevné odkazy lze vytvářet pouze v rámci jednoho diskového oddílu jsou pevně svázány s originálem. To znamená, že se data po smazání originálu neztratí, ztratí se až tehdy jsou-li odstraněny všechny pevné odkazy a originál. Zato symbolické odkazy lze vytvářet kdekoli, ale nejsou vázány na originál, takže pokud uživatel originál odstraní, symbolický odkaz odkaz bude stále v systému (logicky bude odkazovat na neexistující soubor, podobné jako

je zástupce v operačním systému Windows). [31] [35] [40]

6.1.14 netstat

Je příkaz, pomocí kterého může uživatel vypsát na konzoli routovací tabulku, všechna tcp spojení a seznam rozhraní podle parametrů přidané k příkazu. [31] [35]

```
login@RB450G:~$ netstat -r
Kernel IP routing table
Destination      Gateway         Genmask         Flags        MSS Window  irtt Iface
default          10.0.0.138     0.0.0.0         UG           0 0        0 eth0
10.0.0.0         *              255.255.255.0   U            0 0        0 eth0
192.168.1.0      *              255.255.255.0   U            0 0        0 br-lan
```

Obr. 8. Výpis routovací tabulky programem netstat

6.1.15 ping

Je program využíváný pro získání odezvy ze serveru. Tento příkaz bez parametrů získává do přerušení odezvu každou vteřinu, proto je vhodné využívat k tomuto příkazu parametr `-c` a hodnotu určující počet měření. Dále je možné nastavit periodu parametrem `-i` (minimální perioda je 0.2 vteřin). [31] [35]

```
#ping -i 0.2 -c 2 seznam.cz
PING seznam.cz (77.75.76.3) 56(84) bytes of data.
64 bytes from www.seznam.cz (77.75.76.3): icmp_seq=1 ttl=247 time=18.8 ms
64 bytes from www.seznam.cz (77.75.76.3): icmp_seq=2 ttl=247 time=19.6 ms
```

```
- - - seznam.cz ping statistics - - -
2 packets transmitted, 2 received, 0% packet loss, time 200 ms
rtt min/avg/max/mdev = 18.831/19.255/19.680/0.446 ms
```

6.1.16 vi

Jedná se o textový editor určený pro operační systémy GNU/Linux. V tomto programu uživatel přechází pomocí různých klávesových módů. Viz níže předvedeno v tabulkách:

Tab. 3. Vi ukončení programu[50]

Příkaz	Popis
:q	Ukončí program bez uložení
:q!	Ukončí program bez uložení (je nutný použít, pokud byla provedena nějaká změna)
:x	Ukončí program s uložení
:wq název	Ukončí program a uloží data do souboru s názvem odpovídající parametru název

Tab. 4. *Vi hlavní příkazy[50]*

Příkaz	Popis
i	Následující text bude vložen před pozici kurzoru
a	Následující text bude vložen za pozici kurzoru
I	Následující text bude vložen na začátek řádku
A	Následující text bude vložen na konec řádku
o	Následující text bude vložen na novou řádku vytvořenou pod kurzorem
R	Následující text bude přepisovat starý text od pozice kurzoru
S	Následující text bude nahrazovat celou řádku
x	Smaže jeden znak na pozici kurzoru
dd	Smaže aktuální řádek
7dd	Smaže následujících 7 řádků
y\$	Uloží do schránky vše od pozice kurzoru do konce řádky
p	Vloží text ze schránky na konec řádku

Tab. 5. *Vi nahrazování slovních spojení[50]*

Příkaz	Popis
s/wikipedia/Wikipedie/	nahradí řetězec „wikipedia“ řetězcem „Wikipedie“ - pouze první následující výskyt
s/wikipedia/Wikipedie/g	nahradí řetězec „wikipedia“ řetězcem „Wikipedie“ - všechny výskyty na aktuální řádce
%s/wikipedia/Wikipedie/g	nahradí řetězec „wikipedia“ řetězcem „Wikipedie“ - všechny výskyty v souboru
1,\$s/wikipedia/Wikipedie/g	nahradí řetězec „wikipedia“ řetězcem „Wikipedie“ - všechny výskyty v souboru
.,158s/wikipedia/Wikipedie/g	nahradí řetězec „wikipedia“ řetězcem „Wikipedie“ - všechny výskyty od aktuální řádky do řádku číslo 158

[31] [35]

6.2 Opkg

6.2.1 Popis služby

Jedná se o správce softwaru sloužící k vyhledání vhodného software v repositářích a jeho nainstalováním. V jiných distribucích Linuxu známe obdobné programy jako jsou *apt-get*, *aptitude*, *pacman*, *yum* atd. Program *opkg* se slangově někdy označuje jako Entware, protože převážně odkazuje na uložště Entware určené emdebed zařízením. [4]

Obsluha programu *opkg* je velice jednoduchá a skládá se z příkazu *opkg* a podpříkazu určující, co má program *opkg* dělat. Pokud uživatel chce stahovat a instalovat je potřeba nejdříve použít příkaz *opkg update*, čímž se stáhne do paměti zařízení seznam dostupných balíčků k instalaci, poté stačí napsat *opkg install <jméno balíčku>* a software se se všemi sounáležitostmi stáhne a nainstaluje. Odstranění software probíhá pomocí podpříkazu *remove <jméno balíčku>*. Aktualizace software probíhá příkazem *opkg upgrade*, který automaticky upgraduje všechny dostupné balíčky (pokud by chtěl uživatel aktualizovat pouze konkrétní balíček, tak použije tento příkaz *opkg upgrade <jméno balíčku>*). Program *opkg* má ještě několik dalších možností, o kterých si uživatel může přečíst na wiki.openwrt.org. [4]

6.3 Sudo

6.3.1 Popis služby

Sudo je bezpečnostní program vyvíjený programátorem Toddem Millerem. Pomocí tohoto programu mohou získat ostatní uživatelé přístup k celým nebo částečnému oprávnění superuživatele root. Tento program využívá konfiguračního souboru */etc/sudoers*, kde může superuživatel nastavit logování příkazů sudo, nastavovat, kteří uživatelé získají která práva a podobně. Soubor */etc/sudoers* se nesmí editovat jinak než pomocí příkazu *visudo*, protože tento program kontroluje správnost syntaxe (při špatné syntaxi by program Sudo mohl být nefunkční a uživatelé by nedovedli konfigurační soubor otevřít, aby jej opravili) a navíc hlídá aby daný soubor needitovalo najednou více uživatelů. [31] [35]

6.3.2 Instalace a konfigurace služby

Z repositářů OpenWRT byla nainstalována služba *sudo* a poté byla nakonfigurována příkazem *visudo*, tak aby mohli všichni uživatelé příkazem *sudo* získat práva superuživatele (z důvodu, že v SSH je nedovoleno přihlášení pod uživatelským jménem *root* a uživatelé

by nemohli dostatečně spravovat RouterBoard bez tohoto oprávnění). [4]

```
root@RB450G:~#opkg update
root@RB450G:~#opkg install sudo
root@RB450G:~#visudo
root@RB450G:~#cat /etc/sudoers
...
Defaults targetpw # Ask for the password of the target user
ALL ALL=(ALL) ALL # WARNING: only use this together with 'Defaults targetpw'
...
```

6.4 Cron

6.4.1 Popis služby

Jedná se o obslužnou aplikaci, která se zapíná ihned při startu systému, kde si načte všechny své konfigurační soubory a uloží je do operační paměti a poté zůstává uspaný. Cron se periodicky probouzí každou minutu, aby zkontroloval modifikace časů u naplánovaných úloh, načte znovu změněné soubory, provede všechny naplánované operace pro danou minutu a nakonec se zase uspí. V operačním systému OpenWRT se konfigurační soubory nacházejí v adresáři `/etc/crontabs` a v souboru `/etc/crontab`. Pro editaci crontabu se používá příkaz: `crontab -e`. [8]

Každý záznam v crontabu se skládá z šesti parametrů, kde prvních pět parametrů, jsou časové hodnoty při kterých se má spustit požadovaný příkaz. Jedná se o minutu, hodinu, den v měsíci, den v týdnu a měsíc. Šestá parametr je příkaz, který se má vyplnit. Pokud nelze určit hodnota, jako například v parametru den v týdnu, protože uživatel potřebuje zapínat zadaný příkaz třeba prvním dnem v měsíci, tak jako parametr den v týdnu se zapíše hvězdička. [8]

6.5 NTP a nastavení času

6.5.1 Popis služby

Jedná se o systémového daemona (dříve byl tento daemon znám pod názvem `xntpd`), který implementuje NTP (Network Time Protocol), tento protokol slouží pro synchronizaci hodin s přesností milisekund. `Ntpd` implementuje jak klientskou, tak i serverovou část protokolu NTP. V konfiguračním souboru `/etc/ntp.conf`, který tento daemon čte, při zavádění systému se dá specifikovat klientská síť a časové servery. [11]

Pro nastavení systémového času lze použít také utilita *npddate*, ale není to lepší řešení než *ntpd*, protože se může stát, že tok času se bude jevit jako nesouvislý, a to hlavně když se pomocí *ntpdate* vrátí čas do minulosti. Oproti tomu *ntpd* využívá méně násilné volání *adjtimex*, které předchází velikým časovým skokům tím, že mění pouze rychlost systémových hodin. [11]

6.5.2 Instalace a konfigurace

Byl nainstalován z repositářů OpenWRT systémový daemon *ntpd*, dále byly nastaveny jako jeho zdroje pro synchronizaci náhodně vybrané České servery. Pro ruční konfiguraci času byl stažen z repositářů program využívající ntp protokol *ntpdate*, který nastaví datum a čas v počítači na datum získané z ntp serveru. Tento program by měl používat pouze uživatel obeznámený s nastavením programů na RouterBoardu pracujícími s časem (například *cron*), protože velkou změnou času by tento software mohl některé operace opakovat (při změně času do minulosti), nebo naopak přeskočit (při změně času do budoucnosti). Dále bylo v systémovém konfiguračním souboru */etc/config/system* nastaveno časové pásmo určené pro Českou republiku a prvotně nastaven čas přes příkaz *ntpdate* ze serveru *nic.cz*, který tuto službu provozuje. Na závěr byl spuštěn daemon *ntpd*, který bude při startu systému synchronizovat čas z přednastavených serverů a v průběhu čas regulovat pomocí programu *adjtimex*. [4]

```
root@RB450G:/#opkg update
root@RB450G:/#opkg install ntpd ntpdate
root@RB450G:/#/etc/init.d/sysntpd disable
root@RB450G:/#vi /etc/config/system
root@RB450G:/#cat /etc/config/system
config system
    option 'hostname' 'RB450G'
    option 'timezone' 'CET-1CEST,M3.5.0,M10.5.0/3'
...
root@RB450G:/#ntpdate ntp.nic.cz
root@RB450G:/#vi /etc/ntp.conf
root@RB450G:/#cat /etc/ntp.conf
...
driftfile /var/lib/ntp/ntp.drift

server 147.32.127.254 iburst #ntp2.sh.cvut.cz
server 195.113.144.238 iburst #tak.cesnet.cz
server 147.228.52.11 iburst #clock1.zcu.cz

restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap
restrict default ignore
...
```

```
root@RB450G:/#/etc/init.d/ntpd enable
root@RB450G:/#reboot
...
root@RB450G:/#date
```

Po restartu systému odpovídalo datum aktuálnímu času. V případě, že by měl uživatel méně výkonné zařízení, tak by jako alternativu daemona ntpd mohl použít pouze příkaz ntpdate, který by byl aktivován při startu systému a poté nastaven do cronu, aby se čas co nějaký časový úsek synchronizovalo (Zde by samozřejmě nebyl samotný router jako další zařízení poskytující korigování času přes NTP a navíc by ve výjimečných případech mohl nastat výše zmíněný problém s ntpdate například se zmíněným cronem.).

Vzhledem k možnosti konfigurace NTP pomocí LuCI (webové grafické konfigurační rozhraní pro RouterBoard viz. Bod 6.10) byl místo ntpd zvolen program ntpclient pro, který využívá také službu adjtimex a navíc má pro LuCI dostupný modul pro konfiguraci, v tomto modulu byl také nakonfigurován.

6.6 SSH

6.6.1 Popis služby

SSH tedy (Secure Shell) je protokol, který slouží pro bezpečnou komunikaci na vzdálených serverech. Byl vytvořen na Finské univerzitě poté, co byla celá univerzitní síť napadena hackerskou metodou man-in-the-middle (tedy útočník odposlouchává tok dat po síti, kudy projdou i velmi citlivé informace jako jsou hesla). Protože v tu dobu celá síť komunikovala pouze pomocí telnetu, tak si útočník snadno získal přístupy do systému. Rok od zmíněného útoku byla vytvořena Tatuem Ylönemem první verze SSH (SSH-1), o rok později vyšla aktuální verze SSH-2. [41] [42]

Hlavní odlišností mezi SSH a Telnetem je ten, že SSH má šifrovanou veškerou komunikaci. Komunikace je šifrována pomocí asymetrického šifrování, kde server má dva klíče. Klíč pro šifrování zpráv se nazývá veřejný klíč, klíč pro dešifrování se nazývá privátní klíč. Uživatel se připojí na port SSH (standartně 22), server pošle uživateli veřejný klíč a klient ověří, zda takový klíč zná. Klient vygeneruje klíč, kterým bude šifrována veškerá komunikace a zašifruje ho veřejným klíčem, který získal od serveru. Tento zašifrovaný klíč pošle serveru a ten ji pomocí privátního klíče dešifruje. Postup pro přenos zprávy je jednoduchý server pošle zprávu pomocí veřejného klíče uživatele a uživatel ji pomocí privátního klíče dešifruje. [41] [42]

Při prvním spojení by si měl uživatel ověřit správnost klíčů. Protože klíče bývají velice dlouhé řetězce znaků a proto by byla kontrola pro uživatele poměrně náročná. Pro tento účel existuje pojem fingerprint (v překladu otisk prstu). SSH dovede dlouhé klíče proměnit v jedinečný matematický otisk, který pak předloží uživateli. [41] [42]

6.6.2 Instalace a konfigurace

V nainstalovaném OpenWRT byl předinstalovaný SSH server *Dropbear*. Tento program byl vytvořený s ohledem na nízké hardwarové možnosti některých embedded zařízení, což není problém u RouterBoardu 450G, navíc pro další konfiguraci SSH se *Dropbear* silně nevyhovující. Proto byl odinstalován a byly smazány všechny konfigurační soubory, které správce balíků nesmazal. [4]

```
root@RB450G:~/etc/init.d/dropbear disable
root@RB450G:~/#opkg update
root@RB450G:~/#opkg remove dropbear
root@RB450G:~/#rm -rf /etc/dropbear
root@RB450G:~/#opkg install openssh-server openssh-sftp-server
root@RB450G:~/etc/init.d/sshd enable
```

Po instalaci byla nastavena IP adresa na eth1 192.168.1.1 s maskou sítě 255.255.255.0. Na PC byla staticky nastavena IP adresa odpovídající dané síti (tedy 192.168.1.2 s maskou sítě 255.255.255.0) a bránou na rozhraní eth1 v RouterBoardu. Poté pomocí SSH klienta Putty proběhlo první připojení do RouterBoardu přes protokol SSH. Jako parametr pro určení cílové destinace lze použít i hostname RouterBoardu nastavené v kapitole 5.2.

```
root@RB450G:~/#ifconfig eth1 192.168.1.1 netmask 255.255.255.0 up
```

6.6.3 Zabezpečení služby

V konfiguračním souboru SSH serveru */etc/ssh/sshd_config* byla nastavena verze protokolu SSHv2, protože SSHv1 má řadu bezpečnostních chyb. Dále bylo zakázáno, aby se uživatelé připojovali jako root, protože pokud útočník najde na serveru povolený port pro SSH (zde je využíván defaultní port 22), tak může zkusit hesla s tím, že zná alespoň uživatelské jméno na rozdíl od jiných účtů, kde se uživatelská jména liší. Dále bylo povoleno připojení k SSH pouze z eth1 (192.168.1.1) a bylo zakázáno přihlášení uživatele bez hesla. [4] [43]

```
root@RB450G:/#vi /etc/ssh/sshd_config
root@RB450G:/#cat /etc/ssh/sshd_config
#port pro SSH (default 22)
Port 22
#Zákaz logování bez hesla
PermitEmptyPasswords no
#Blokování uživatele root
PermitRootLogin no
#Povolení IP adresáře
ListenAddress 192.168.1.1
#Protokol SSH
Protocol 2
AuthorizedKeyFile ./ssh/authorized_keys
Subsystem sftp /usr/lib/sftp-server
```

Další možností zabezpečení SSH je například změna defaultního portu 22 na některý jiný, který není na serveru využíván. Tato změna zabezpečí server pouze proti útočníkům procházejícím sítí a hledající pouze port 22, v případě nalezení se pokoušejí o bruteforce útok nebo útok pomocí slovníků. Ale vzhledem k tomu, že tito uživatelé nejsou schopni zkontrolovat jiné porty až na port 22, tak je nepravděpodobné, aby zkoušeli prolomit jiný účet než root, který byl stejně zablokován. [43]

6.7 SFTP

6.7.1 Popis služby

Jedná se o bezpečnější verzi protokolu FTP. Tento protokol využívá služeb protokolu SSHv2 (lze využít i na starší verzi SSHv1, zde je ale narušena nezávislost na architektuře počítače) k šifrovanému přenosu. Na rozdíl od protokolu SCP má SFTP rozsáhlejší možnosti pro vedlejší operace obdobně jako protokol. [42]

6.7.2 Instalace služby

Byl nainstalován SFTP server z repositářů OpenWRT pomocí příkazu `opkg`. [4]

```
root@RB450G:/#opkg update
root@RB450G:/#opkg install openssh-sftp-server
```


6.8 DHCP

6.8.1 Popis služby

DHCP (Dynamic Host Konfiguration Protocol) je protokol pracující na portech 67 a 68. Tento protokol je nástupce protokolu BOOTP a slouží k automatickému rozdělování IP adres pro všechny uživatele v síti. [4] [44] [45]

Uživatel (protože ještě nemá přidělenou IP adresu) využívá broadcasting (adresa příjemce je 255.255.255.255) pomocí kterého vyšle pakety (tyto pakety se nazývají DHCPDISCOVER) do všech zařízení v síti. Vyslané pakety si přečtou všechny DHCP servery v síti a reagují na něj posláním odpovědi v podobě paketu s názvem DHCPOFFER. Klient si vybere ze všech dostupných nabídek pouze nejvhodnější IP adresu a odpovídá zpátky serveru, který uživateli tuto nabídku zaslal paketem DHCPREQUEST. Poté server odpoví klientovi paketem DHCPACK a od této chvíle může klient využívat jemu přidělenou IP adresu. [4] [44] [45]

Samotná konfigurace DHCP serveru na Linuxových operačních systémech probíhá v konfiguračním souboru `/etc/dhcpd.conf` (v OpenWRT je tento soubor umístěn v adresáři `/etc/config/`), kde se v horní části nastavují informace o síti (jméno sítě, maska sítě, adresa brány a další volitelné údaje). Ve druhé části se konfiguruje vlastnosti jednotlivých podsítí a jednotlivých zařízení (například nastavit, že zařízení s určitou MAC adresou bude mít vždy určitou IP adresu). [4] [44] [45]

6.8.2 Konfigurace DHCP serveru

Byl nakonfigurován konfigurační soubor `/etc/config/dhcp` pro rozhraní lan v konfiguračním rozhraní LuCI. Server přiděluje IP adresy v rozsahu od 10 do 254 po dobu 12 hodin. [4]

```
root@RB450G:~#cat /etc/config/dhcp
```

```
...
config dhcp 'lan'
    option interface 'lan'
    option leasetime '12h'
    option start '10'
    option limit '224'
    ra_management '1'
```

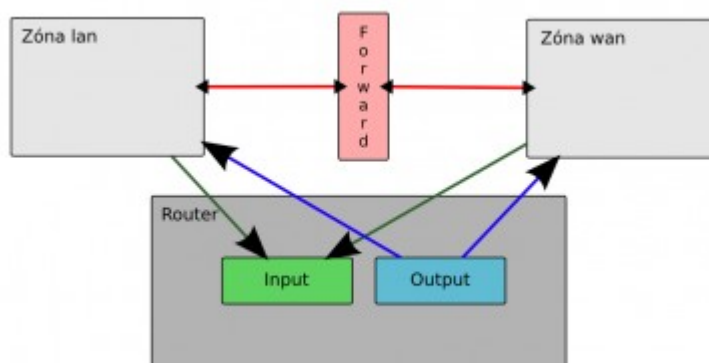
```
config dhcp 'wan'
    option interface 'wan'
    option ignore '1'
```

6.9 Firewall (Iptables)

6.9.1 Popis služby

Jedná se o část jádra Linuxu, která zvládá libovolnou činnost na úrovni paketu a je to taková alternativa aplikací známých jako Firewall. Každý paket, který doputuje do Linuxového jádra musí projít jedním z řetězců jenž jsou uloženy v tabulkách. Ty poté rozhodnou, jak systém s paketem naloží. [41] [46]

OpenWRT má vlastní aplikaci nadřazenou aplikaci firewallu, která lze konfigurovat pomocí LuCI. Tento firewall funguje na povolování pravidel pro určité zóny (zóny symbolizují síťová rozhraní na daném zařízení, kde jedna zóna může zastupovat i více zařízení), které si uživatel definuje v konfiguračním souboru pro firewall `/etc/config/firewall`. Druhá alternativa je využívat konfiguraci iptables ze scriptu `/etc/firewall.user`, která se spouští při startu firewallu. [41] [46]



Obr. 9. Schéma OpenWRT firewallu [47]

Iptables má tři základní tabulky *filter* (standartní tabulka určená k filtrování paketů), *nat* (tabulka sloužící pro překlad adres) a *mangle* (umožňuje zpracovávat hlavičky paketů, které umí značkovat, manipulovat s některými parametry a podobně). V každé ze zmíněných tabulek se nachází několik řetězců jimiž pakety procházejí. V tabulce *filter* jsou tři základní řetězce *INPUT* (příchozí), *OUTPUT* (odchozí) a *FORWARD* (přůchozí). Každý z těchto řetězců má nastaveno své hlavní pravidlo, které se nazývá *policy* a říká co udělat s paketem pokud nevyhoví (defaultně je toto pravidlo nastavováno systémem na *ACCEPT*, ale z hlediska zabezpečení a konfigurace firewallu je vhodnější používat opačný postoj *DROP*). [41] [47]

6.9.2 Konfigurace služby

V konfiguračním souboru pro firewall byly vytvořeny zóny pro síťové rozhraní eth1 (lan) a eth0 (wan). Tuto konfiguraci byla vytvořena v konfiguračním rozhraní LuCI v modulu *Network=>Firewall*. [4] [47]

```
root@RB450G:/#cat /etc/config/firewall
```

```
...
```

```
config zone
```

```
    option name 'lan'
    option network 'lan'
    option output 'ACCEPT'
    option forward 'ACCEPT'
    option input 'ACCEPT'
```

```
config zone
```

```
    option name 'wan'
    option network 'wan'
    option network 'wan6'
    option output 'ACCEPT'
    option forward 'ACCEPT'
    option input 'ACCEPT'
```

```
config forwardin
```

```
    option src 'lan'
    option dest 'wan'
```

```
...
```

Bylo zakázáno jakkoli na zařízení využívat protokol Telnet, dále byl přidán zákaz přístupu ze zóny wan do zařízení (protokol 22 jenž je vyhrazen na zařízení pro službu SSH a port 443 jenž je využíván na serveru pro službu HTTPS, kterou využívá grafické rozhraní LuCI). Konfigurace byla opět vytvořena v konfiguračním rozhraní LuCI. [4] [47]

```
root@RB450G:/#cat /etc/config/firewall
```

```
...
```

```
config rule
```

```
    option src 'wan'
    option proto 'tcp'
    option dest_port '22'
    option name 'SSH'
    option dest_ip '10.0.0.6'
    option target 'DROP'
```

```
config rule
```

```
    option dest_port '23'
    option name 'Telnet'
    option proto 'tcp udp'
    option src '*'
    option dest_ip '192.168.1.1'
    option dest_ip '10.0.0.6'
```

```
option target 'DROP'
```

```
config rule
```

```
option src 'wan'
option proto 'tcp'
option dest_port '443'
option dest_ip '10.0.0.6'
option name 'HTTPS'
option target 'DROP'
```

```
...
```

Byla nastavena ochrana SSH proti metodě brutal force (metoda při níž útočník zkouší pomocí slovníku, případně všechny kombinace hesel), při níž firewall hlídá počet přihlášení uživatele v určitém časovém úseku (SSH dovoluje 5 špatných přihlášení, poté se komunikace ukončí). Pokud ano, tak firewall paket upustí. Pro tuto práci bylo nutné stáhnout a nainstalovat z repositáře modul pro iptables, který si ve své paměti pamatuje neúspěšné pokusy o přihlášení. [48]

```
root@RB450G:~#opkg update
```

```
root@RB450G:~#opkg install iptables-mod-contrack-extra
```

Do konfiguračního souboru firewallu byl navrácen odkaz na skript `/etc/firewall.user`, který se vykoná po konfiguraci OpenWRT firewallu. Skript `/etc/firewall.user` je určený pro konfiguraci firewallu pomocí příkazů iptables. Zde byla napsána doplňující konfigurace pro port 22 (určený pro službu SSH). [4] [48]

```
root@RB450G:~#vi /etc/config/firewall
```

```
root@RB450G:~#cat /etc/config/firewall
```

```
...
```

```
config include
```

```
option path '/etc/firewall.user'
```

```
root@RB450G:~#vi /etc/firewall.user
```

```
cat /etc/firewall.user
```

```
iptables -N ssh
```

```
iptables -A ssh -i eth0 -j DROP
```

```
iptables -A ssh -m recent - --update - --second 150 - --hitcount 1 - --name SSH -j DROP
```

```
iptables -A ssh -m recent - --set - --name SSH -j ACCEPT
```

```
iptables -A ssh -j ACCEPT
```

```
#iptables -I zone_lan_input -p tcp - --dport 22 -m state - --state NEW -j ssh
```

```
iptables -I INPUT -p tcp - --dport 22 -m state - --state NEW -j ssh
```

V prvním řádku byl vytvořen vlastní řetězec ssh, ve kterém probíhá výše zmíněný postup.

V posledním řádku je vytvoření odkazu na řetězec ssh v řetězci INPUT (ideální by bylo využít řetězec `zone_lan_input` ten ale slouží pouze pro OpenWRT firewall, takže předchozí zablokování portu 22 pro síťové rozhraní wan nebude fungovat, proto byla tato ochrana také přidána zde). V tomto řetězci probíhá kontrola na pravidla vytvořená v konfiguračním souboru firewallu pro zónu lan. [51]

6.9.3 Country

Pro zvýšení efektivity firewallu slouží country, které uživateli ukáží, která podmínka nastavená v iptables nejčastěji vyhověla (administrátor by se měl snažit, aby byly nejčastější podmínky upřednostňovány, protože pokud podmínka vyhoví, ukončí se procházení seznamu a tím se zrychlí chod). Tyto country lze nalézt v LuCI v modulu Status=>Firewall (druhý sloupec) a restartovat je pomocí kliknutí na odkaz Reset Counters. [49]

6.10 LuCI

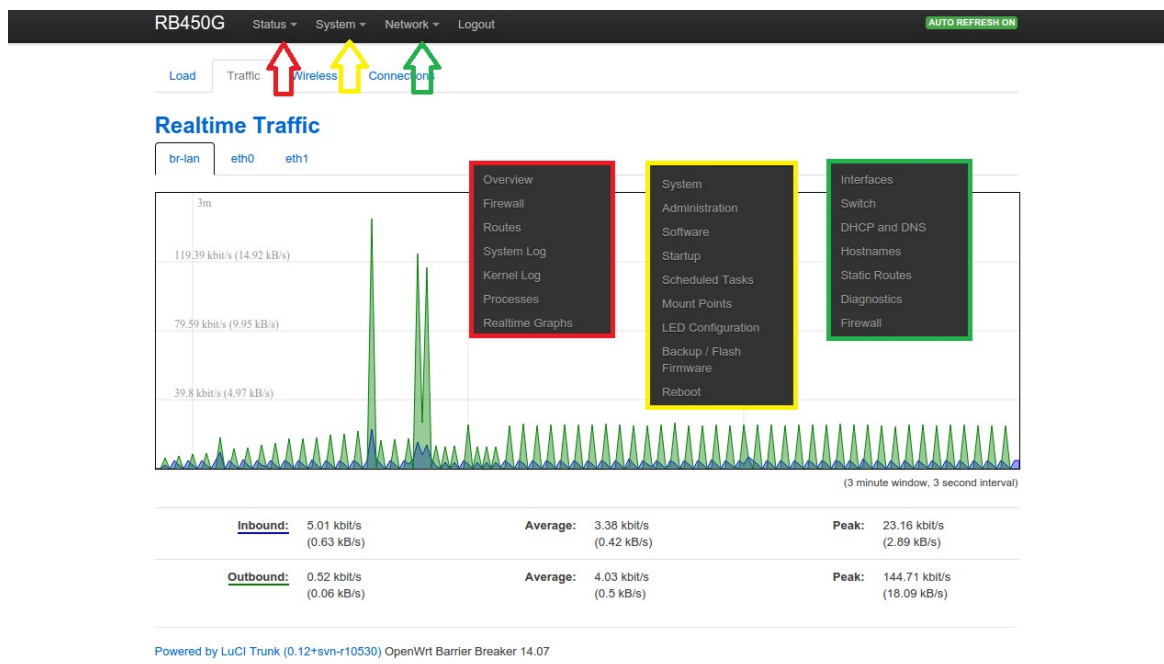
6.10.1 Popis služby

LuCI je interaktivní webové rozhraní využívající technologii asynchronního JavaScriptu a je určené pro konfiguraci zařízení s operačním systémem OpenWRT. Toto zařízení je vyvíjeno od verze OpenWRT 8.09-Kamikaze, což odpovídá roku 2008. [49]

Uživatelské rozhraní je velice jednoduché a skládá se tří záložek, Status, System, Network a tlačítka Logout. Záložka Status se skládá z informací o stavu síťových rozhraní, firewallu, dají se zde najít systémové logy, dokonce lze pomocí LuCI pozastavovat a ukončovat procesy. Dále se zde nacházejí realtime statistiky s vzorkovací frekvencí 0,33Hz. [49]

V záložce System umožňuje uživateli spravovat softwarovou výbavu svého zařízení, tento modul pracuje se správcem repositářů v operačním systému opkg a uživateli předává informace o nainstalovaném software, o dostupném software a dává uživateli možnost odinstalovat software přímo z webu. Dále je zde možnost zálohovat systém, který se v případě našeho zařízení vytvoří archiv konfiguračních souborů z adresáře */etc*, bohužel se jedná pouze o konfigurační soubory, takže zde uživatel nenalezne například zaváděcí skripty uložené v adresáři */etc/init.d*. V případě nahrání zálohy se daná konfigurace přepíše a systém se restartuje. Dále je zde možnost editace konfiguračního souboru crontab (viz. Kapitola 3.1). A nakonec se v této záložce nachází modul pro konfiguraci LED diod, přesněji řečeno uživatel si zde může vybrat LED diodu a přidat k ní podmínku rozsvícení a zhasnutí. [49]

V poslední záložce se nachází vše co souvisí se sítí. To znamená nastavování adres, firewallu, DHCP a jiných služeb ke kterým si uživatel stáhne potřebné moduly pro LuCI. Seznam všech dostupných modulů pro LuCI je k dispozici příkazem *opkg list | grep luci*. [49]



Obr. 10. Struktura modulů LuCI

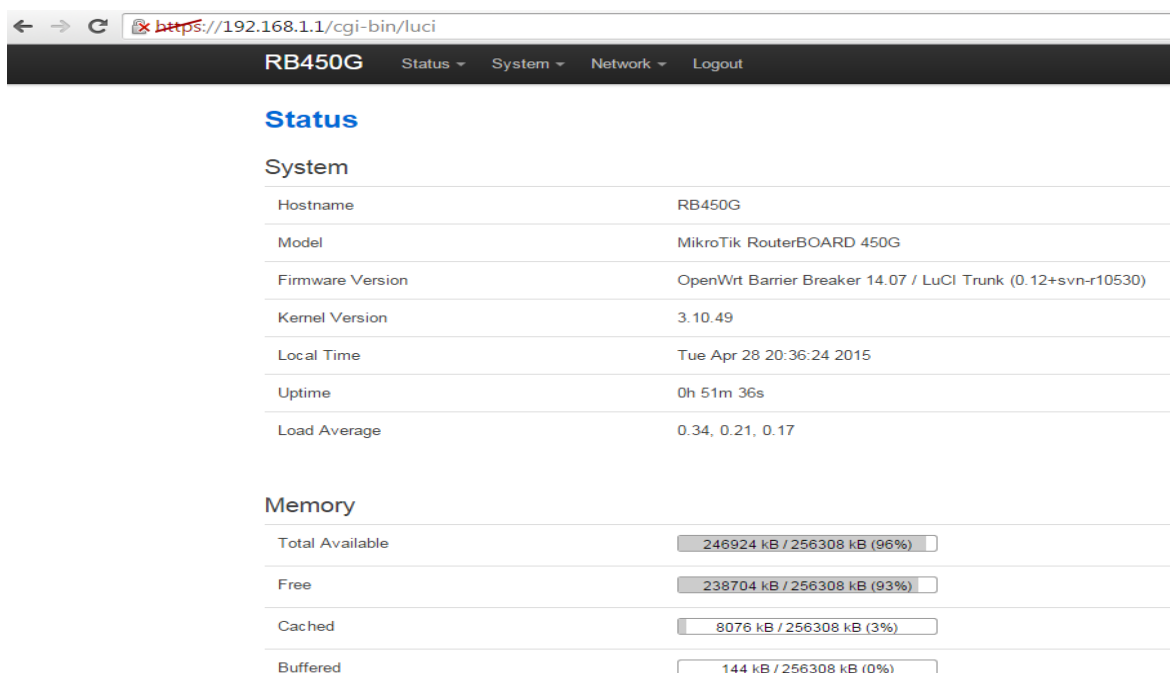
Na obrázku 10 je vidět seznam modulů, které se ukrývají pod výše zmíněnými záložkami. V pozadí je vidět graf závislosti přenosu dat na čase. Počáteční výchylky zobrazují posílání po síti webovou stránku. Zbylé menší výchylky zobrazují cyklický sběr dat prohlížeče co tři vteřiny.

6.10.2 Instalace a konfigurace služby

Do systému byla nainstalována webová aplikace LuCI z repositářů OpenWRT. Příkazem `opkg list | grep luci-luci-i18n-` byl vypsán seznam balíčků pro Luci pro podporu jazyků, ale vzhledem k absenci češtiny bylo od této instalace upuštěno a byl ponechán defaultní nativní jazyk, tedy angličtina. [4]

```
root@RB450G:~# opkg update
root@RB450G:~# opkg install luci
root@RB450G:~# opkg list | grep luci-i18n
...
root@RB450G:~# /etc/init.d/uhttpd enable
root@RB450G:~# /etc/init.d/uhttpd start
```

Po instalaci po instalaci proběhlo připojení na Luci napsáním do prohlížeče ip adresu RouterBoardu, která byla nastavena v bodu 5.5. a připojil jsem se do systému jako root s heslem nastaveným v bodu 5.1.



System	
Hostname	RB450G
Model	MikroTik RouterBOARD 450G
Firmware Version	OpenWrt Barrier Breaker 14.07 / LuCI Trunk (0.12+svn-r10530)
Kernel Version	3.10.49
Local Time	Tue Apr 28 20:36:24 2015
Uptime	0h 51m 36s
Load Average	0.34, 0.21, 0.17

Memory	
Total Available	246924 kB / 256308 kB (96%)
Free	238704 kB / 256308 kB (93%)
Cached	8076 kB / 256308 kB (3%)
Buffered	144 kB / 256308 kB (0%)

Obr. 11. LuCI první přihlášení

6.10.3 Zabezpečení LuCI

Bylo zakázáno připojení k LuCI pomocí protokolu HTTP, místo tohoto protokolu byl nastaven protokol HTTPS, který je podobně jako protokol SSH šifrovaný, takže útočník nebude moci odposlouchávat přenos. [4]

```
root@RB450G:~# opkg update
root@RB450G:~# opkg install px5g uhttpd-mod-tls
root@RB450G:~# uci delete uhttpd.main.listen_http
root@RB450G:~# commit
```

Bylo povoleno připojení k LuCI pouze z lokální sítě (eth1).

```
root@RB450G:~# vi /etc/config/uhttpd
root@RB450G:~# cat /etc/config/uhttpd
...
list listen_https '192.168.1.1:443'
...
```

V kódu LuCI bylo manuálně nastaveno, že se do systému smí připojit pouze účet vytvořený v kapitole 5.3., tento účet zde bude mít plnou moc, proto je důležité, aby přístup do LuCI mělo co nejméně uživatelů. Z důvodu ochrany bylo vypnuto předepisování uživatelského jména a zakázán přístup pro uživatele root (díky tomu musí útočník navíc zjistit název uživatelského účtu).

```
root@RB450G:/#vi /usr/lib/lua/luci/controller/admin/index.lua
root@RB450G:/#cat /usr/lib/lua/luci/controller/admin/index.lua
```

```
...
    page.sysauth = {"login"}
```

```
...
root@RB450G:/#reboot
```

Pokud by uživatel chtěl přidat například účet root mezi účty, které se mohou připojit do LuCI, tak by postupoval podle tohoto postupu:

```
root@RB450G:/#vi /usr/lib/lua/luci/controller/admin/index.lua
root@RB450G:/#cat /usr/lib/lua/luci/controller/admin/index.lua
```

```
...
    page.sysauth = {"login", "root"}
```

```
...
root@RB450G:/#reboot
```

Dále bylo z přihlašovacího formuláře odstraněno tlačítko reset. Toto tlačítko slouží k restartování zařízení do defaultního nastavení.

```
root@RB450G:/#cat /usr/lib/lua/luci/view/sysauth.htm
```

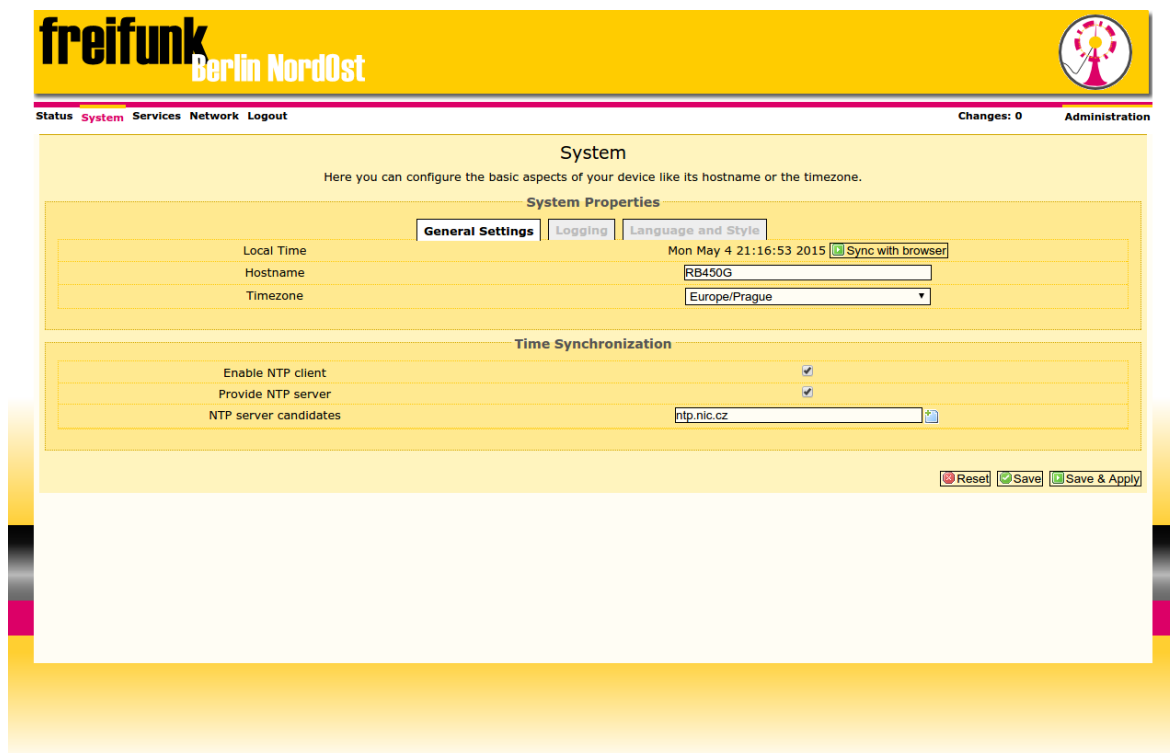
```
...
<div>
<input type="submit" value="<%.Login%>" class="cbi-button cbi-button-apply" />
<input type="reset" value="<%.Reset%>" class="cbi-button cbi-button-reset" />
</div>
```

```
...
root@RB450G:/#vi /usr/lib/lua/luci/view/sysauth.htm
root@RB450G:/#cat /usr/lib/lua/luci/view/sysauth.htm
```

```
...
<div>
<input type="submit" value="<%.Login%>" class="cbi-button cbi-button-apply" />
</div>
```

6.10.4 Změna designu

Byl z repositářů stažen balíček s jiným designem pro LuCI, tento balík byl vyhledán pomocí LuCI v záložce *System => Software*, kde byl prvně updatován správce balíků a poté nalezen požadovaný balíček. Po stažení a instalaci balíku *luci-theme-freifunk-bno* byl v LuCI změněn design v modulu *System => System => Language and Style*, kde si nyní lze vybrat mezi dvěma styly.



Obr. 12. LuCI jiný webdesign

6.10.5 Změna jazyka

Jazykový balík lze stáhnout z repositářů OpenWRT v CLI jak bylo ukázáno v kapitole 6.10.2 nebo obdobně jako se stahoval design pro LuCI v kapitole 6.10.4. Změna jazyka lze změnit v modulu *System => System => Langue and Style*. [4]

7 STAŽENÍ OBRAZŮ DISKŮ S OPENWRT

Byly vytvořeny pomocí programu dd obrazy disků s jádrem a systémovými soubory nakonfigurovaného operačního systému OpenWRT. Tyto obrazy byly přesunuty na microSD kartu.

```
# cd /tmp
# dd if=/dev/mtd5 | gzip > openwrt_kernel.img.gz
# dd if=/dev/mtd6 | gzip > openwrt_rootfs.img.gz
# scp openwrt_kernel.img.gz user@10.0.0.1:/tftpboot
# scp openwrt_rootfs.img.gz user@10.0.0.1:/tftpboot
```

ZÁVĚR

V této práci byla popsána platforma RouterBoard, operační systém Mikrotik RouterOS a popsán RouterBoard 450G.

Byl vysvětlen pojem GNU/Linux a popsány vybrané distribuce určené pro směrovací techniku. Byl vysvětlen souborový systém GNU/Linux a vysvětleno, proč je GNU/Linux odolnější proti virům a jiné havěti.

V praktické části byl zálohovaný původní licenční klíč operačního systému Mikrotik RouterOS, dále se pomocí zavedeného dočasného operačního systému GNU/Linux vytvořila kopie diskových oddílů určených pro jádro operačního systému a systémové soubory.

Také práce obsahuje popis instalace operačního systému OpenWRT, zprovoznění microSD karty, vytvoření uživatelských účtů a nastavení síťových rozhraní. Potom obsahuje popis instalace a konfigurace žádaných služeb jako je Cron, NTP, Firewall, SSH, SFTP, DHCP a webového grafického rozhraní LuCI.

Dále je popsáno zabezpečení LuCI, kde je povolen pouze protokol HTTPS a je zakázáno se přihlásit pomocí uživatele root. Také bylo popsáno zabezpečení SSH, kde je zakázáno přihlášení ze sítě wan a také je použita ochrana SSH proto Brutal Force útoku, zákaz přihlášení uživatele root a zákaz přihlášení účtů bez hesla.

V posledním bodě je podrobněji představeno ovládání webového grafického rozhraní LuCI.

Praktické využití této práce naleznou vlastníci zařízení RouterBoardu, kterým pro práci nedostačuje operační systém Mikrotik RouterOS ať už z důvodu licenčního omezení nebo z důvodu nemožnosti doplnění systému o vlastní programy.

SEZNAM POUŽITÉ LITERATURY

- [1] BLAHA, Libor. *Konfigurace a nastavení platformy RouterBoard*. UTB ve Zlíně, 2009. bakalářská práce.
- [2] *Mikrotik routerboard product catalog Q2 2014* [online]. [cit. 2015-05-17]. Dostupné z: download2.mikrotik.com/2014-Q2.pdf
- [3] I4WIFI A. S. *Návod k obsluze: Platforma RouterBoard s předinstalovaným RouterOS Mikrotik* [online]. [cit. 2015-05-17]. Dostupné z: files.i4wifi.cz/inc/_doc/attach/StoItem/3308/RouterBoard_manual_CZE.pdf
- [4] MIKROTIK. *MikroTik Wiki* [online]. [cit. 2015-05-17]. Dostupné z: wiki.mikrotik.com/wiki
- [5] *I4wifi* [online]. [cit. 2015-05-17]. Dostupné z: i4wifi.cz
- [6] *MUM* [online]. [cit. 2015-05-17]. Dostupné z: www.mum.mikrotik.com
- [7] MIKROTIK. *MikroTik RouterOS* [online]. [cit. 2015-05-17]. Dostupné z: www.mikrotik.com/pdf/what_is_routeros.pdf
- [8] SOBELL, Mark G. *Linux: praktický průvodce*. Praha: Computer Press, 1999. 80-7226-190-8
- [9] *FHS* [online]. [cit. 2015-04-20]. Dostupné z: jplindquist.com/wp-content/uploads/2013/03/
- [10] *Wikipedie: Linux* [online]. [cit. 2015-05-17]. Dostupné z: cs.wikipedia.org/wiki/Linux
- [11] NEMETH, Evi, Garth SNYDER a Trent R HEIN. 2008. *Linux: kompletní příručka administrátora : 2. aktualizované vydání*. Vyd. 1. Brno: Computer Press, 984 s. ISBN 978-80-251-2410-9.
- [12] *Wikipedie: Linux jádro* [online]. [cit. 2015-05-17]. Dostupné z: cs.wikipedia.org/wiki/Linux_%28j%C3%A1dro%29
- [13] *Wikipedie: Architektura MIPS* [online]. [cit. 2015-05-17]. Dostupné z: http://cs.wikipedia.org/wiki/Architektura_MIPS
- [14] OpenWRT. *OpenWRT Wiki* [online]. [cit. 2015-05-17]. Dostupné z: wiki.openwrt.org/
- [15] OpenWRT. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikime-

- dia Foundation, 2001- [cit. 2015-05-17]. Dostupné z:
cs.wikipedia.org/wiki/OpenWRT
- [16] OpenWRT. *OpenWRT* [online]. [cit. 2015-05-17]. Dostupné z: openwrt.org/
- [17] DebWRT. *DebWRT* [online]. [cit. 2015-05-17]. Dostupné z:
<http://www.DebWRT.net/>
- [18] *Wikipedia: DebWRT* [online]. [cit. 2015-05-17]. Dostupné z:
en.wikipedia.org/wiki/DebWRT
- [19] *Filesystem Hierarchy Standard* [online]. [cit. 2015-05-17]. Dostupné z:
<http://www.pathname.com/fhs/pub/fhs-2.3.html>
- [20] HÝBL, Alois. *Průvodce Linuxem 8: zabezpečení Linuxu* [online]. [cit. 2015-05-17].
Dostupné z: linuxexpres.cz/praxe/pruvodce-linuxem-8-zabezpeceni-linuxu
- [21] OHNESORG, Dan. *Linux a viry* [online]. [cit. 2015-05-17]. Dostupné z: [linux.cz/vi-
ry.html](http://linux.cz/viry.html)
- [22] *Aktualizace RouterOS* [online]. [cit. 2015-05-17]. Dostupné z: [www.mikrotik.tlupa.-
com/?p=278](http://www.mikrotik.tlupa.-com/?p=278)
- [23] JANOŠEC, Josef. *Jak aktualizovat bootloader v RouterBoardu* [online]. [cit. 2015-05-17]. Dostupné z: josef.janosec.net/jak-aktualizovat-bootloader-v-RouterBoardu
- [24] Upgrade biosu. *Ispforum* [online]. [cit. 2015-05-17]. Dostupné z: [ispforum.cz/view-
topic.php?f=5&t=3859](http://ispforum.cz/view-topic.php?f=5&t=3859)
- [25] *Mips Development Environments: Full Debian "Lenny" on an RB433AH board (Atheros AR7161)* [online]. [cit. 2015-05-17]. Dostupné z: [http://opensource.-
dyc.edu/mips-devel](http://opensource.-dyc.edu/mips-devel)
- [26] VOYAGE LINUX. *Voyage Linux* [online]. [cit. 2015-05-17]. Dostupné z: [http://li-
nux.voyage.hk/](http://linux.voyage.hk/)
- [27] *Mikrotik RouterBoard 450G* [online]. [cit. 2015-05-17]. Dostupné z:
wiki.hwmn.org/w/Mikrotik_RouterBoard_450G
- [28] BusyBox. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikime-
dia Foundation, 2001- [cit. 2015-05-17]. Dostupné z:
<http://cs.wikipedia.org/wiki/BusyBox>
- [29] BUSYBOX. *BusyBox* [online]. [cit. 2015-05-17]. Dostupné z: busybox.net
- [30] HORÁK, Jan. *Co je to proces init a jaká je jeho historie* [online]. In: . [cit. 2015-05-

- 17]. Dostupné z: www.wild-web.eu/blog/co-je-to-proces-init-a-jaka-je-jeho-historie/
- [31] *Linux Documentation* [online]. [cit. 2015-05-17]. Dostupné z: linux.die.net
- [32] Almquist shell. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-05-17]. Dostupné z: http://cs.wikipedia.org/wiki/Almquist_shell
- [33] Unixový shell. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-05-17]. Dostupné z: cs.wikipedia.org/wiki/Unixov%C3%BD_shell
- [34] BRANDEJS, Michal. *Programovací jazyk textových manipulací: awk (1)* [online]. [cit. 2015-05-17]. Dostupné z: ics.muni.cz/bulletin/articles/33.html
- [35] KERRISK, Michael. *Linux man page* [online]. [cit. 2015-05-17]. Dostupné z: www.man7.org
- [36] Huffmanovo kódování. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-05-17]. Dostupné z: cs.wikipedia.org/wiki/Huffmanovo_k%C3%B3dov%C3%A1n%C3%AD
- [37] WATZKE, David. *Unixové nástroje: 13 (ps, kill a signály)* [online]. In: . [cit. 2015-05-17]. Dostupné z: abclinuxu.cz/clanky/unixove-nastroje-13-ps-kill-a-signaly#kill-a-signaly
- [38] WATZKE, David. *Unixové nástroje: 11 (split, dd)* [online]. [cit. 2015-05-17]. Dostupné z: abclinuxu.cz/clanky/navody/unixove-nastroje-11-split-dd#dd
- [39] *Výpisy z jádra* [online]. [cit. 2015-05-17]. Dostupné z: ucsimply.cz/elnx/uvod-do-vyvoje-ovladacu/vypisy-z-jadra/
- [40] WATZKE, David. *Unixové nástroje: 2 (ls, ln)* [online]. [cit. 2015-05-17]. Dostupné z: abclinuxu.cz/clanky/navody/unixove-nastroje-2-ls-ln#ln
- [41] KRČMÁŘ, Petr. *Linux: postavte si počítačovou síť*. Praha: Grada, 2008. 978-80-247-1290-1
- [42] SSH file transfer protocol. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-05-17]. Dostupné z: cs.wikipedia.org/wiki/SSH_file_transfer_protocol
- [43] DOČEKAL, Michal. *Správa linuxového serveru: Praktické rady pro zabezpečení SSH* [online]. [cit. 2015-05-17]. Dostupné z: linuxexpres.cz/praxe/sprava-linu-

xoveho-serveru-prakticke-rady-pro-zabezpeceni-ssh

- [44] Dynamic Host Configuration Protocol. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-05-17]. Dostupné z: http://cs.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol
- [45] MICROSOFT. *Microsoft Support* [online]. [cit. 2015-05-17]. Dostupné z: support.microsoft.com/en-us
- [46] ŠTRAUCH, Adam. *OpenWRT:: DHCP, firewall a webové rozhraní* [online]. [cit. 2015-05-17]. Dostupné z: root.cz/clanky/openwrt-dhcp-firewall-a-webove-rozhrani/
- [47] ŠTRAUCH, Adam. *OpenWRT: naklikáváme firewall* [online]. [cit. 2015-05-17]. Dostupné z: root.cz/clanky/openwrt-naklikavame-firewall/
- [48] DOČEKAL, Michal. *Správa linuxového serveru: Praktické rady pro zabezpečení SSH II* [online]. [cit. 2015-05-17]. Dostupné z: <http://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru-prakticke-rady-pro-zabezpeceni-ssh-1>
- [49] ŠTRAUCH, Adam. *LuCI: webové rozhraní pro OpenWRT* [online]. [cit. 2015-05-17]. Dostupné z: root.cz/clanky/luci-webove-rozhrani-pro-openwrt/
- [50] Vi (editor). *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2015-05-19]. Dostupné z: http://cs.wikipedia.org/wiki/Vi_%28editor%29#Ovl.C3.A1d.C3.A1n.C3.AD
- [51] DOČEKAL, Michal. *Správa linuxového serveru: Praktické rady pro zabezpečení (nejen) SSH IV* [online]. [cit. 2015-05-17]. Dostupné z: <http://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru-prakticke-rady-pro-zabezpeceni-ssh-4>
- [52] , csoto. OpenWRT: forum. In: *AR8316 Swich Support (page 8)* [online]. [cit. 2015-05-19]. Dostupné z: <https://forum.openwrt.org/viewtopic.php?id=21837&p=8>
- [53] BOUŠKA, Petr. *VLAN: Virtual Local Area Network* [online]. [cit. 2015-05-19]. Dostupné z: <http://www.samuraj-cz.com/clanek/vlan-virtual-local-area-network/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

BSD	Berkeley Software Distribution
CD	Compact Disc
CLI	Command Line Interface
DHCP	Dynamic Host Configuration Protocol
DC	Direct Current
FHS	Filesystem Hierarchy Standart
FTP	File Transfer Protocol
GNU	Gnu is Not UNIX
GPL	General Public License
HDD	Hard Disk Drive
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
ID	IDentification
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
LED	Light Emitting Diode
MAC	Media Access Control
MIPS	Million Instructions Per Second
NTP	Network Time Protocol
PID	Process ID
PoE	Power over Ethernet
PPPoE	Point to Point Protocol over Ethernet
RB	RouterBoard
RISC	Reduced Instruction Set Computer
SCP	Short-Curcuit Protection
SD	Secure Digital
SFTP	SSH File Transfer Protocol
SSH	Secure Shell
TFTP	Trivial File Transfer Protocol
USB	Universal Serial Bus
V	Volt
VLAN	Virtual Local Area Network

SEZNAM OBRÁZKŮ

Obr. 1. RB450G [4].....	12
Obr. 2. WinBox (licence).....	14
Obr. 3. Typická struktura adresářů v operačním systému Linux [9].....	17
Obr. 4. WinBox přihlášení.....	21
Obr. 5. Popis síťové konfigurace RouterBoardu [0].....	30
Obr. 6. Upravený Baner.....	31
Obr. 7. Výpis procesů programem ps.....	35
Obr. 8. Výpis routovací tabulky programem netstat.....	37
Obr. 9. Schéma OpenWRT firewallu [47].....	46
Obr. 10. Struktura modulů LuCI.....	49
Obr. 11. LuCI první přihlášení.....	50
Obr. 12. LuCI jiný webdesign.....	52

SEZNAM TABULEK

Tab. 1. Licence Mikrotik RouterOS [4].....	13
Tab. 2. Seznam nejznámějších signálů [33].....	34

SEZNAM PŘÍLOH

P I: Prozatimní Linux určený k zavádění ze sítě [27]	
P II: Obraz oddílu kernel operačního systému OpenWRT	
P III: Obraz oddílu rootfs operačního systému OpenWRT	
P IV: Obraz oddílu kernel operačního systému Mikrotik RouterOS	
P V: Obraz oddílu rootfs operačního systému Mikrotik RouterOS	
P VI: Klíč k systému Mikrotik RouterOS	
P VII: Textový dokument s názvy účtů a hesly pro operační systém OpenWRT	