# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS


# NÁVRH GRAFICKÉHO ROZHRANÍ FIREWALLU S VYUŽITÍM QT4 FRAMEWORKU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

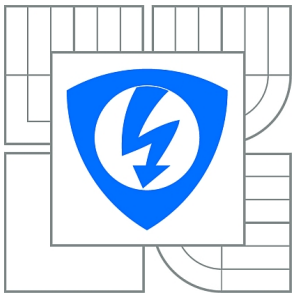

AUTOR PRÁCE MARTIN ŠTEFANY
AUTHOR



BRNO 2010

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

# NÁVRH GRAFICKÉHO ROZHRANÍ FIREWALLU S VYUŽITÍM QT4 FRAMEWORKU

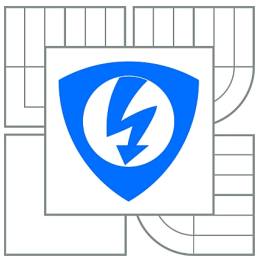
ESTABLISHMENT OF THE GRAPHIC INTERFACE FOR FIREWALL USING QT4 FRAMEWORK

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — MARTIN ŠTEFANY
AUTHOR

VEDOUCÍ PRÁCE — Ing. TOMÁŠ MATOCHA
SUPERVISOR

BRNO 2010



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky a komunikačních technologií**

**Ústav telekomunikací**

# Bakalářská práce

bakalářský studijní obor
**Teleinformatika**

| | | | |
|---|---|---|---|
| ***Student:*** | Martin Štefany | ***ID:*** | 106820 |
| ***Ročník:*** | 3 | ***Akademický rok:*** | 2009/2010 |

**NÁZEV TÉMATU:**

## Návrh grafického rozhraní firewallu s využitím Qt4 frameworku

**POKYNY PRO VYPRACOVÁNÍ:**

Navrhněte a realizujte kompletní GUI pro správu a konfiguraci firewallu. Tato aplikace musí využívat knihovnu Qt4. Popište možnosti linuxového firewallu netfilter a následně aplikaci do firewallu implementujte tak, aby spolupracovala a bylo možné pravidla upravovat. GUI musí být schopně stávající pravidla rozvněž zobrazit.

**DOPORUČENÁ LITERATURA:**

[1] NEMETH, E., SNYDER, G., HEIN T. Linux - Kompletní příručka administrátora. Computer Press, 2004. 880 s. ISBN: 80-722-6919-4.
[2] Qt Reference Documentation [online]. Dostupný z WWW:http://doc.trolltech.com/4.5/.

| | | | |
|---|---|---|---|
| ***Termín zadání:*** | 29.1.2010 | ***Termín odevzdání:*** | 2.6.2010 |

***Vedoucí práce:*** Ing. Tomáš Matocha

**prof. Ing. Kamil Vrba, CSc.**
*Předseda oborové rady*

**UPOZORNĚNÍ:**

## ABSTRAKT

Cieľom práce je návrh aplikácie, ktorá poslúži ako grafické rozhranie pre terminálový nástroj iptables, pričom ide o nástroj využívajúci framework Netfilter na ovládanie firewallu v jadre operačného systému GNU/Linux. Grafické rozhranie má zjednodušiť správu firewallu v Linux-e, pretože si užívateľ nemusí pamätať všetky príkazy a zároveň mu grafické rozhranie zobrazuje aktuálnu štruktúru a obsah firewallu. Práca popisuje nielen formát príkazov nástroja iptables a ich možnosti, ale aj samotnú štruktúru a funkciu firewallu v Linux-e. Navrhnutá aplikácia je napísaná v jazyku C++ s využitím aspektov objektovo orientovaného programovania a frameworku Qt4. Qt4 je výborný framework na tvorbu grafických rozhraní, prináša množstvo vlastných tried a metód, ktoré rozširujú možnosti programátora pri tvorbe, či už grafických, alebo terminálových aplikácií pre veľké množstvo platforiem. Súčasťou práce je aj manuál k navrhnutému a naprogramovanému grafickému rozhraniu, aplikácii qIPtables, ktorý má priblížiť užívateľovi prácu s touto aplikáciou a správy firewallu.

## KLÍČOVÁ SLOVA

GNU, Linux, Ubuntu, Netfilter, iptables, firewall, paketový filter, grafické rozhranie, GUI, GNOME, KDE, C++, OOP, Qt4 framework, Qt Designer, Qt Linguist, IDE, Qt Creator, qIPtables

## ABSTRACT

The aim of this thesis is to design an application, which will serve as a graphical interface to the terminal application iptables. iptables is an application which uses the Netfilter framework for managing firewall in operating system GNU/Linux. Graphical interface is a way how to raise a comfort of firewall configuration and management, because user doesn't have to remember all of the commands and graphical interface also shows him actual structure and contents of the firewall in a tree view. Thesis describes format and options of the commands and also the firewall structure and its function in Linux. Designed application is written in C++ language using aspects of object oriented programming and uses Qt4 framework. Qt4 is a great framework for creating graphical user interfaces, brings a lot of new classes and methods and extends programmer's possibilities during designing graphical or terminal applications for lots of platforms. Thesis also includes a manual to designed graphical interface, to the application qIPtables, which user can use to learn the basics of using this application and firewall management.

## KEYWORDS

GNU, Linux, Ubuntu, Netfilter, iptables, firewall, packet filter, graphic interface, GUI, GNOME, KDE, C++, OOP, Qt4 framework, Qt Designer, Qt Linguist, IDE, Qt Creator, qIPtables

ŠTEFANY, Martin *Návrh grafického rozhraní firewallu s využitím Qt4 frameworku*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2010. 45 s. Vedoucí práce byl Ing. Tomáš Matocha, PhD.

# PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Návrh grafického rozhraní firewallu s využitím Qt4 frameworku“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Brno ..... .....

(podpis autora)

# OBSAH

# ÚVOD

V súčasnosti každý počítač, pripojený k sieti Internet, vyžaduje ochranu pomocou firewallu. Jednou z možností, ako tento počítač chrániť firewallom, je nainštalovať aplikáciu plniacu funkciu firewallu na tento počítač a nastaviť pravidlá filtrácie, pričom takéto aplikácie môžu sledovať nielen sieťové spojenia danej aplikácie, ale aj jej správanie v operačnom systéme. Ďalšou z možností je inštalácia firewallu medzi sieť, cez ktorú sa tento počítač pripája na Internet a samotný Internet, čo zabezpečí ochranu nielen jedného počítača, ale aj ochranu celej siete. V tomto prípade plní často funkciu firewallu sieťový smerovač, ktorý môže byť zastúpený práve zariadením od niektorého zo známych výrobcov sieťových prvkov, alebo môže ísť aj o server postavený na niektorom z UNIXových operačných systémov, napr. operačný systém GNU/Linux (a jeho rôzne distribúcie), ktorý potom plní v sieti viacero funkcií, teda nielen funkciu prístupovej brány a firewallu (DHCP server, Web/FTP server a pod.).

V prípade využitia jednoduchšieho a lacnejšieho riešenia v podobe servera s nainštalovaným operačným systémom GNU/Linux, poskytuje samotné jadro Linuxu funkcionalitu filtrovania a modifikácie paketov. Práve od verzie 2.4 sa v tomto linuxovom jadre nachádza nástroj *iptables*, ktorý je súčasťou projektu Netfilter. Ide však o čisto terminálový nástroj, kde je nutné všetky príkazy zadávať ručne do terminálu. Tieto príkazy je tak nutné si všetky pamätať a vedieť ich správne používať, čo však začínajúcim správcom a hlavne domácim užívateľom znižuje komfort používania a neumožňuje im jednoduché nastavenie firewallu pre ich systém, čím môže vzniknúť určité bezpečnostné riziko.

Úlohou tejto práce je návrh aplikácie, ktorá poslúži ako grafické rozhranie, ktoré zvýši komfort používania nástroja *iptables*, poskytujúceho tak široké možnosti tvorby kvalitného firewallu. V teoretickej časti práce sú opísané základné teoretické informácie o operačnom systéme Linux, distribúcii Kubuntu a firewallu v Linuxe – nástroji *iptables*. Praktická časť popisuje vytvorenú aplikáciu, ktorá slúži ako grafická nadstavba pre nástroj *iptables*.

**Táto práca predpokladá u čitateľa isté skúsenosti s používaním operačného systému GNU/Linux. Na návrh a vývoj aplikácie bola použitá distribúcia Ubuntu 10.04 LTS a jej derivácia Kubuntu 10.04 LTS s doinštalovaným Qt4 frameworkom, vývojovým rozhraním Qt Creator a ďalšími pomocnými aplikáciami určenými na vývoj softvéru pomocou Qt4.**

# 1 GNU/LINUX VO FUNKCII FIREWALLU

## 1.1 Operačný systém GNU/Linux

GNU/Linux je operačný systém, ktorý patrí do rodiny UNIXových operačných systémov. GNU označuje slobodný softvér z projektu GNU, používaný práve v kombinácii s linuxovým kernelom (jadrom operačného systému). Autorom jeho prvého jadra z roku 1991 je Linus Thorvalds [7]. Najnovšou verziou jadra Linux je verzia 2.6, na ktorej sú postavené pravdepodobne všetky súčasné linuxové distribúcie. Verzia 2.6 je nástupcom jadra 2.4, ktoré prinieslo nástroj *iptables* a jeho funkcionalitu firewallu pre operačný systém GNU/Linux.

Podľa webových stránok [13] je Linux moderný operačný systém, ktorý podporuje beh v 32 aj 64 bitovom režime, zvláda viacprocesorové systémy a zároveň je užívateľsky prívetivejší než ktorýkoľvek iný operačný systém. Z hľadiska bezpečnosti je Linux systémom, ktorý nepotrebuje žiadne komplikované zabezpečovacie systémy alebo antivírusové systémy a odoláva mnohým známym bezpečnostným komplikáciám, aj keď určite nie úplne všetkým. Aktualizácie sú, rovnako ako celý systém, dostupné zdarma a väčšinou včas. Linux je možné výkonovo aj vzhľadovo prispôsobiť, obsahuje veľké množstvo aplikácií na rôzne účely, dostupných pomocou centrálneho nástroja na inštaláciu aplikácií z tzv. repozitárov a väčšina z nich má dostupnú kvalitnú dokumentáciu a zdrojové kódy. Linux bol primárne vyvinutý ako viacužívateľský operačný systém, to znamená, že na ňom môže súčasne pracovať viacero používateľov pod rôznymi prihlasovacími údajmi.

Tab. 1.1: Prehľad všeobecne zameraných linuxových distribúcií

| **Distribúcia** | **Poznámka** |
|---|---|
| Debian | Obľúbená nekomerčná distribúcia |
| Ubuntu | Populárna distribúcia, určená aj pre začiatočníkov |
| Red Hat Enterprise | Obchodná verzia systému Red Hat Linux |
| Fedora | Neobchodná verzia systému Red Hat Linux |
| CentOS | Voľne šíriteľná verzia Red Hat Enterprise Linux |
| SUSE Linux Enterprise | Obchodná verzia systému SUSE |
| openSUSE | Voľne šíriteľná verzia systému SUSE |
| Mandriva (Mandrake) | Jedna z užívateľsky najpriateľskejších distribúcií |
| Gentoo | Veľmi flexibilná a výkonná distribúcia |
| Slackware | Stabilná, základná a orezaná distribúcia |
| Slax | Prenosná, malá a rýchla distribúcia |

Operačný systém GNU/Linux je dostupný vo forme distribúcií od rôznych tvorcov. Distribúcia znamená, že k základnému jadru operačného systému, ktorého tvorbou sa zaoberá samotný linuxový projekt, tvorca pribalí príkazy, daemonov a ďalší softvér. Takto zdieľajú všetky distribúcie rovnaký rodokmeň linuxového jadra, ale v ostatných doplnkoch sa líšia. Tabuľka 1.1 [10] uvádza zoznam najrozšírenejších a zrejme aj najobľúbenejších všeobecne zameraných distribúcií.

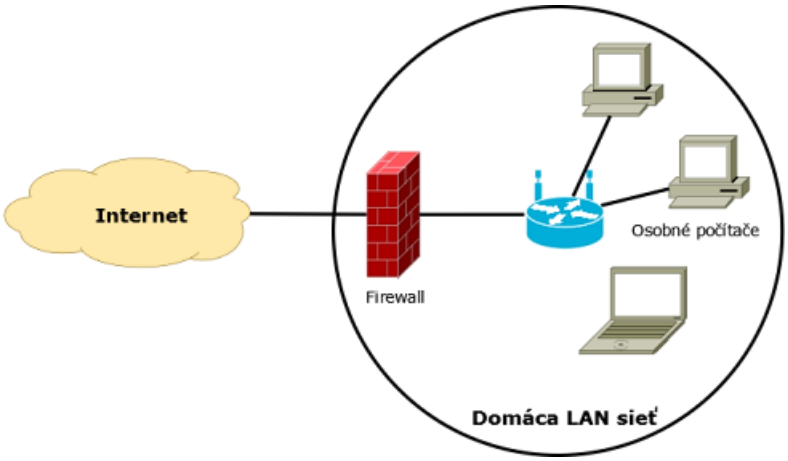
Distribúcia Ubuntu v súčasnosti zohráva najväčšiu rolu pri rozširovaní Linux-u, povedomia o jeho existencii a možnostiach, pretože jeho tvorca, spoločnosť Canonical, sa snaží Linux zjednodušiť a spríjemniť pre užívateľov a tým sprístupniť jeho používanie väčšiemu počtu používateľov, čím sa snaží konkurovať ostatným najrozšírenejším operačným systémom.

## 1.2 Definícia pojmu firewall

Slovo **firewall** (angl. „ohnivá stena“) označuje zväčša sieťový prvok, ktorý slúži na riadenie a zabezpečenie sieťovej prevádzky medzi dvoma sieťami, pričom tieto siete majú väčšinou rôzne úrovne zabezpečenia a dôveryhodnosti. Typickým príkladom je použitie firewallu na oddelenie domácej (lokálnej) počítačovej siete od Internetu, v prípade firemného riešenia ide o oddelenie firemnej siete od Internetu, prípadne o oddelenie jednotlivých častí siete od seba. Samotný firewall kontroluje dátové toky na základe definovaných pravidiel, určujúcich ktorá sieťová prevádzka má byť prepustená a ktorá zablokovaná.[4][9]

Firewally existujú na viacerých úrovniach a vo viacerých formách. Najnižšou úrovňou je paketový filter a jeho rozšírením je stavový paketový filter, ktorý berie do úvahy aj stavy spojení (ďalšia teória je podrobne popísaná v ďalšej časti). Medzi firewally s vyššou úrovňou zabezpečenia môžeme zaradiť aplikačné brány alebo tzv. proxy firewally a tiež najmodernejšie systémy s detekciou prieniku do systému IDS (Intrusion Detection System) [4]. Medzi jednoduchšie a menej nákladné riešenie patrí využitie jedného systému zabezpečujúceho funkcie prístupovej brány, firewallu, NAT, prípadne aj WWW servera alebo poštovej brány. Takéto riešenie je vhodné aj v domácnosti, kde úlohu prístupovej brány, DHCP servera, NAT a firewallu plní domáci smerovač.

Nákladnejším, ale zároveň bezpečnejším riešením je využitie niekoľkých oddelených systémov, ktoré rozdeľujú sieť na rôzne zabezpečené úrovne. Použitím dvoch takýchto systémov vytvoríme medzi nimi demilitarizovanú zónu (DMZ), v ktorej umiestnené aplikačné servery majú zvýšenú úroveň zabezpečenia a zároveň tieto systémy riadia nezávisle sieťovú prevádzku medzi jednotlivými sieťami.




Obr. 1.1: Firewall v domácej počítačovej sieti

## 1.3 Práca s paketmi, (stavový) paketový filter

Paketový filter je softvér (často súčasťou jadra operačného systému alebo firmvéru hardvérového zariadenia) kontrolujúci hlavičky paketov, ktoré ním prechádzajú a rozhoduje o ich ďalšom osude. Paketový filter môže rozhodnúť o zahodení paketu, ako keby nebol nikdy prijatý, o zamietnutí paketu, pričom je odosielateľovi zaslaná správa o jeho nedoručení, o jeho prijatí na ďalšie spracovanie alebo môže ísť o zložitejšiu akciu, ako je napríklad preklad sieťových adries alebo zaznamenanie informácií o spracovávanom IP pakete.[8]

Na rozhodovanie, ktoré pakety sa majú ďalej odovzdať do systému na spracovanie a ktoré, naopak, zahodiť alebo odmietnuť, je možné použiť rad kritérií:

- **typ protokolu** (TCP, UDP, ICMP a pod.),
- **číslo portu** (iba pre protokoly TCP/UDP),
- **typ paketu** (TCP SYN/ACK, ICMP echo-request, echo-reply a pod.),
- **zdrojové** alebo **cieľové sieťové rozhranie** firewalu,
- **zdrojová** alebo **cieľová adresa** počítača alebo siete.

Je dôležité si uvedomiť, že filtrovanie paketovým filtrom prebieha na tretej (štvrtej) vrstve sieťového modelu OSI, tzv. sieťovej (transportnej) vrstve. Takýto filter nepozná počas kontroly aplikácie, ktoré spojenie používajú, stará sa iba o samotné spojenie. Špeciálnym prípadom je proxy firewall, ktorý dokáže filtrovať až na aplikačnej vrstve. Takýto firewall je už schopný skontrolovať obsah paketu a zamedziť tak napríklad prístup k nebezpečným a nevhodným webovým stránkam, prípadne kontrolovať, či na port určený pre prijímanie požiadaviek HTTP serverom, prichádzajú legitímne HTTP požiadavky.[9]

Nevýhodou paketového filtra je fakt, že nerozpozná súvislosti medzi spojeniami, každý paket tak spracúva samostatne, a preto pri službách, ako napr. FTP prenos

súborov, kde sa vytvára spojenie na náhodný port, je nutné povoliť celý rozsah portov alebo takéto služby nevyužívať a zakázať. V prvom prípade to môže byť nebezpečné, v tom druhom zase nerealizovateľné.

Špeciálnym prípadom paketového filtra je **stavový paketový filter**, ktorý už rozoznáva súvislosti medzi jednotlivými paketmi, rozlišuje vzťahy medzi spojeniami a uchováva si ich jednotlivé stavy. Jedno nadviazané spojenie a s ním všetky pakety, ktoré k nemu patria, sa nazýva relácia (session). Tento stav spojenia, prípadne vzťahy medzi spojeniami, je možné využiť priamo v pravidlách, a tým povoliť napr. odpovede na odoslané webové požiadavky, prípadne takto vyriešiť problém s FTP prenosmi.[10]

## 1.4 Netfilter a *iptables*

Linuxové jadrá obsahovali paketový filter od verzie jadra 1.1. V roku 1994 sa objavila prvá generácia založená na nástroji *ipfw* zo systému BSD. Tá bola vylepšená pre verziu jadra 2.0 a ponúkala užívateľský nástroj *ipfwadm* na zadávanie filtrovacích pravidiel. V roku 1998, s príchodom novej verzie jadra 2.2, bol predstavený nástroj *ipchains*. Následne, spolu s ďalším prepisom jadra na verziu 2.4, bola v roku 1999 predstavená štvrtá generácia, nástroje *iptables* a *ip6tables*.[8]

Rozdiel medzi výrazmi *iptables* a Netfilter môže pôsobiť mierne zmätočne. Oficiálny názov projektu zaoberajúci sa vývojom paketového filtra pre jadrá Linux-u je Netfilter. Tento termín však označuje aj framework v rámci tohto jadra, ktorý potom nástroj *iptables* využíva na vykonávanie rôznych operácií s paketmi (napr. filtráciu). Netfilter je teda framework a *iptables* je nástroj na spracovanie príkazov z príkazového riadka definujúcich pravidlá a ich nastavenie do jadra Linux-u.[14]

### 1.4.1 Nástroje *iptables* a *ip6tables*

#### *iptables* v distribúcii Ubuntu

Balíček s nástrojom *iptables* je v distribúcii Ubuntu už predinštalovaný spolu s balíčkom *ufw* (Uncomplicated Firewall). Samotný balíček s nástrojom *iptables* obsahuje príkazy pre IP verziu 4 (`iptables`, `iptables-save` a `iptables-restore`) a tiež pre IP verziu 6 (príkazy `ip6tables`, `ip6tables-save` a `ip6tables-restore`) a taktiež manuálovú dokumentáciu, dostupnú cez príkaz `man iptables` a pod.

#### *iptables* verzus *ip6tables*

Nástroje *iptables* a *ip6tables* držia v jadre oddelené štruktúry pre filtrovanie paketov IP protokolov verzie 4 a 6. Preto existujú aj odlišné príkazy, ktorými je možné tieto

pravidlá v jadre spravovať. Na nastavenie pravidiel pre protokol IPv4 sa používa príkaz `iptables`, pre IPv6 príkaz `ip6tables`, za ktoré je nutné pridať ďalšie parametre. Je však dôležité pripomenúť, že tieto príkazy je nutné v systéme spúšťať s právami superužívateľa. To znamená byť v systéme prihlásený ako superužívateľ (čo môže byť nebezpečné) alebo použiť príkaz `sudo`, ktorý umožňuje bežným užívateľom spúšťať príkazy s právami superužívateľa.

Aktuálne nastavenie pravidiel je uložené v jadre systému a stráca sa pri vypnutí alebo reštartovaní systému. Nástroj *iptables* preto obsahuje príkazy `iptables-save` a `iptables-restore` (pre protokol IPv4) a podobne príkazy `ip6tables-save` a `ip6tables-restore` (pre protokol IPv6), pomocou ktorých je možné uložiť aktuálne pravidlá z jadra do súboru alebo ich zase zo súboru vložiť do jadra.

### 1.4.2 Štruktúra *iptables*

#### Tabuľky (tables)

Tabuľka (table) je konštrukcia, ktorá definuje hlavné kategórie funkcií, ako je filtrovanie paketov alebo preklad sieťových adries (NAT). *iptables* obsahuje štyri tabuľky:

- **filter**;
- **nat**;
- **mangle**;
- **raw**.

Filtrovacie pravidlá sa pridávajú do tabuľky **filter**, pravidlá pre preklad sieťových adries do tabuľky **nat**, tabuľka **mangle** obsahuje špeciálne pravidlá na úpravu hlavičiek v IP paketoch a tabuľka **raw** má jediný účel, ktorým je označovanie paketov, ktoré nemajú byť sledované stavovým firewallom.[14]

#### Reťaze (chains)

Každá tabuľka má svoju vlastnú sadu vstavaných (built-in) reťazí, ale užívateľ môže nadefinovať ďalšie. Všetky vstavané reťaze jednotlivých tabuliek sú kvôli prehľadnosti uvedené v tabuľke 1.2 [6]. Zoradenie reťazí *iptables*, a teda aj poradie v akom všetky pakety prechádzajú týmito reťazami, je znázornené v prílohe A.

Nadefinovať je možné ľubovoľný počet vlastných reťazí v ktorejkoľvek zo štyroch tabuliek. Tieto vlastné reťaze sa následne využívajú na sprehľadnenie štruktúry firewallu alebo na vytvorenie naozaj zložitého systému filtrovania paketov. Užívateľsky nadefinované reťaze sa potom môžu používať ako **akcie (targets)** v jednotlivých pravidlách, vždy však iba v rámci danej tabuľky.

Tab. 1.2: Prehľad jednotlivých tabuliek *iptables* a reťazí, ktoré obsahujú.

| **Tabuľka** | **Reťaz** | **Popis** |
|---|---|---|
| **filter** | FORWARD | Filtrovanie paketov medzi rozhraniami |
| | INPUT | Filtrovanie paketov pre lokálny systém |
| | OUTPUT | Filtrovanie paketov na výstupe systému |
| **nat** | PREROUTING | Preklad adries pred smerovaním (DNAT) |
| | POSTROUTING | Preklad adries po smerovaní (SNAT) |
| | OUTPUT | Preklad adries paketov na výstupe systému |
| **mangle** | PREROUTING<br>POSTROUTING<br>OUTPUT<br>INPUT<br>FORWARD | Modifikácia hlavičiek v IP paketoch - zmena TTL (Time-To-Live) hodnoty, zmena TOS (Type-Of-Service) hodnoty pre využitie QoS (Quality of Service) atď. |
| **raw** | PREROUTING<br>OUTPUT | Označenie paketov, ktoré nemajú byť sledované modulmi *iptables* |

### 1.4.3 Formát *iptables* príkazov

Štruktúra firewallu je už objasnená, a preto je potrebné naznačiť spôsob, ktorým sa dá táto štruktúra firewallu naplniť pravidlami. Zadávanie príkazov v jednoduchých príkladoch bude popísané v ďalších častiach. V tejto časti budú popísané všetky teoretické možnosti príkazov definujúcich pravidlá. Súpis všetkých možností vyzerá podľa manuálových stránok [3] nasledovne:

**iptables [-t tabuľka] -[AD]** reťaz špecifikácia_pravidla [nastavenia]
**iptables [-t tabuľka] -I** reťaz [číslo_pravidla] špecifikácia_pravidla [nastavenia]
**iptables [-t tabuľka] -R** reťaz číslo_pravidla špecifikácia_pravidla [nastavenia]
**iptables [-t tabuľka] -D** reťaz číslo_pravidla [nastavenia]
**iptables [-t tabuľka] -[LFZ]** [reťaz] [nastavenia]
**iptables [-t tabuľka] -N** reťaz
**iptables [-t tabuľka] -X** [reťaz]
**iptables [-t tabuľka] -P** reťaz target [nastavenia]
**iptables [-t tabuľka] -E** starý_názov_reťaze nový_názov_reťaze

Pozn.: hranaté zátvorky v príkazoch znamenajú, že daný parameter je nepovinný. Jednotlivé príkazy je však nutné kvôli zrozumiteľnosti rozobrať podrobnejšie.

### Definícia pravidla

Definíciu pravidla v príkazoch môžu tvoriť **podmienky**, **rozšírenia podmienok**, **akcia** a **nastavenia akcie**. Prakticky bude tvorba pravidiel naznačená v časti 1.5.

**Podmienky (matches)**: Každé pravidlo v *iptables* môže mať určité podmienky, hovoriace, ktoré informácie z hlavičky paketu má firewall pri kontrole brať do úvahy a v prípade, že bude daný paket vyhovovať, akú **akciu** vykonať [14]:

- `-s`, `--source`: zdrojová IP adresa alebo adresa siete;
- `-d`, `--destination`: cieľová IP adresa alebo adresa siete;
- `-p`, `--protocol`: typ protokolu, aký paket obsahuje;
- `-i`, `--in-interface`: vstupné rozhranie, na ktorom bol paket prijatý;
- `-o`, `--out-interface`: výstupné rozhranie.

**Rozšírenia podmienok (match extensions)**: Rozširujú pôvodné možnosti **podmienok** a tak môžu byť na nich závislé. To znamená, že napríklad rozšírenie pre TCP protokol môže byť v pravidle použité, iba ak sa v pravidle nachádza podmienka pre protokol TCP (`-p tcp`). Týchto rozšírení podmienok obsahuje *iptables* podobne ako rozšírení akcií, veľké množstvo a nižšie je uvedených iba niekoľko, pričom kompletný zoznam a popis je možné nájsť v manuálových stránkach *iptables* [3]:

- `-m tcp`: rozšírenie pre protokol TCP,
- `-m udp`: rozšírenie pre protokol UDP,
- `-m icmp`: rozšírenie pre protokol ICMP,
- `-m state`: rozšírenie pre stavový firewall,
- `-m mac`: rozšírenie pre kontrolu MAC adries.

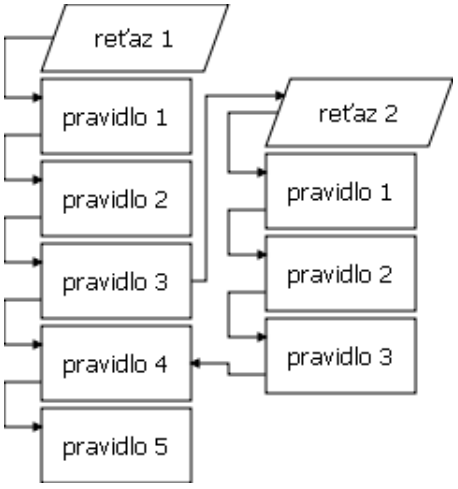
**Akcie (targets)**: V prípade, že dané pravidlo vyhovuje podľa podmienok paketu, ktorý je práve kontrolovaný, vykoná *iptables* akciu definovanú v pravidle za podmienkami. Akciou môže však byť nielen prijatie alebo odmietnutie paketu, či iná definitívna akcia, ale zároveň aj užívateľsky definovaná reťaz. Takto bude ďalej paket kontrolovaný podľa pravidiel tejto reťaze. V pravidle je možné akciu definovať jedným z dvoch parametrov:

- `-j`, `--jump akcia` alebo
- `-g`, `--goto názov_reťaze`.

Rozdiel je však v tom, že ak je za parametrom `--jump` ako akcia užívateľsky definovaná reťaz, a v tejto reťazi paket nevyhovie žiadnemu pravidlu, bude sa v kontrole pokračovať v pôvodnej reťazi, z ktorej bol paket presmerovaný do užívateľskej reťaze (obrázok 1.2). Pri použití parametra `--goto` sa už v spracovávaní v pôvodnej reťazi nepokračuje.

Akcií, ktoré môžu byť zapísané v pravidle za parametrom `-j` je opäť mnoho a ich kompletný zoznam je možné nájsť v manuálových stránkach *iptables* [3]. Ďalej sú uvedené len tie najdôležitejšie:

- `ACCEPT` – paket je prepustený cez firewall
- `DROP` – paket je zahodený, ako keby nebol nikdy prijatý
- `QUEUE` – paket je odoslaný užívateľskému procesu v systéme
- `RETURN` – paket je vrátený na ďalšie spracovanie volajúcej reťazi v prípade, že bol touto reťazou paket nasmerovaný na spracovanie do inej reťaze
- `meno_reťaze` – paket sa bude ďalej spracovávať v reťazi „meno_reťaze“
- `meno_rozšírenia` - paket sa odovzdá na spracovanie rozšíreniu (modulu):
  - `LOG` – informácie o pakete sú zaznamenané do syslogu operačného systému
  - `REJECT` – paket je zahodený, ale jeho odosielateľovi je odoslaná ICMP správa
  - `MASQUERADE` – modifikuje zdrojovú adresu odchádzajúceho paketu na adresu sieťového rozhrania, ktorým paket firewall opustí
  - `SNAT` – modifikuje zdrojovú IP adresu odchádzajúceho paketu
  - `DNAT` – modifikuje cieľovú IP adresu prichádzajúceho paketu
  - `REDIRECT` – presmeruje paket samotnému systému (firewallu), čo sa používa napr. pri transparentnom proxy firewalle




Obr. 1.2: Presmerovanie spracovania paketu akciou **--jump** medzi reťazami

**Rozšírenia akcií:** V prípade, že je ako akcia uvedené niektoré rozšírenie zabezpečujúce, napr. preklad adries, logovanie paketov do systému, úpravu TTL hodnoty v hlavičke IP paketu a pod., tak je nutné nastaviť toto rozšírenie ďalšími parametrami, ktorými sa určí, napr. na akú zdrojovú (cieľovú) adresu sa má pôvodná adresa prekladať, s akým prefixom sa má informácia o pakete zapísať do logov systému alebo na akú hodnotu sa má nastaviť TTL.

Tieto nastavenia sú však rozdielne pre každé rozšírenie a nie je predmetom tejto práce ich všetky popísať. Ako príklad je uvedená zmena zdrojovej IP adresy pri preklade adries: `-j SNAT --to-source ip_adresa[-ip_adresa][:port-port]`.

### 1.4.4 Štruktúra *ip6tables* a formát príkazov

Rozdiel v štruktúre medzi *iptables* a *ip6tables* je v absencii tabuľky **nat** u *ip6tables*, a tiež v rozšíreniach. Formát príkazov je však totožný, pričom namiesto *iptables* je nutné použiť *ip6tables*.

## 1.5 Nastavenie firewallu nástrojom *iptables*

Nastavenie firewallu znamená naplnenie štruktúry *iptables* tak, aby podľa potreby pravidlá filtrovali nežiadúce pakety, ochránili firewall, sieť a počítače k nej pripojené. To znamená nielen vytvorenie týchto pravidiel v jednotlivých reťaziach, ale tiež nastavenie predvolenej akcie tým vstavaným, a prípadne tvorbu vlastných reťazí.

### 1.5.1 Nastavenie štandardnej politiky

Štandardná politika sa môže nadefinovať pre všetky vstavané reťaze tabuliek. V základnom nastavení, pokiaľ užívateľ nenadefinuje inak, je štandardná politika všetkých reťazí nastavená na **ACCEPT**. Toto nastavenie sa však z bezpečnostného hľadiska vôbec neodporúča. Najbezpečnejšou politikou je politika hovoriaca: „Čo nie je povolené, to je zakázané“. A to platí hlavne o reťaziach tabuľky **filter**, ktorá má pakety filtrovať. Na nastavenie štandardnej politiky sa používa príkaz s parametrom `-P` alebo `--policy` vo formáte:

```
iptables [-t tabuľka] -P reťaz akcia;
```

(`iptables [-t tabuľka] --policy reťaz akcia;`)

pričom parameter `-t tabuľka` je nepovinný a v prípade, že chýba, systém príkaz automaticky doplní parametrom `-t filter`. Ako **akcia** môže byť uvedené **ACCEPT** alebo **DROP**.

Ako príklad poslúži príkaz, ktorý nastaví predvolenú politiku reťaze **INPUT** tabuľky filter na hodnotu **DROP**, takže všetky pakety smerujúce na tento počítač sa budú štandardne zahadzovať:

```
iptables -P INPUT DROP;
```

Pre najzákladnejšie nastavenie štandardnej politiky je potrebné určiť akciu pre všetky tri reťaze tabuľky **filter**:

```
iptables -P INPUT DROP;
iptables -P OUTPUT ACCEPT;
iptables -P FORWARD DROP;
```

Takéto nastavenie zahadzuje pakety smerujúce na a cez tento počítač a povoľuje všetky pakety odchádzajúce z tohto počítača. Užívateľ sa však môže rozhodnúť aj pre prísnejšie nastavenie zakazujúce všetky pakety.

### 1.5.2 Tvorba a mazanie užívateľských reťazí

Nová užívateľsky definovaná reťaz sa vytvorí buď príkazom s parametrom `-N` alebo `--new` a názvom novej reťaze:
`iptables [-t tabuľka] -N priklad;`
Pozn.: Pri tvorbe reťazí platí to, čo všeobecne platí v UNIXových systémoch, a to, že názvy reťazí sú citlivé na veľké/malé písmená, a preto sa berú názvy „Test“ a „TEST“ ako dva rozdielne. To umožňuje, napr. aj vytvorenie reťazí s názvami podobnými vstavaným reťaziam ako „inPUT“, „OUTput“ a pod.

Zmazať je možné iba užívateľsky vytvorenú reťaz a na tento účel slúži kombinácia príkazov s parametrami `-F (--flush)` a `-X (--delete-chain)`. Pretože pred zmazaním reťaze musí byť mazaná reťaz prázdna, musí byť ako prvý zadaný príkaz, ktorý zmaže všetky pravidlá z reťaze a až následne príkaz, ktorý zmaže samotnú reťaz:

```
iptables -F priklad;
iptables -X priklad;
```

### 1.5.3 Premenovanie užívateľskej reťaze

Na premenovanie reťaze slúži príkaz s parametrom `-E (--rename-chain)` vo forme:
`iptables [-t tabuľka] -E starý_názov_reťaze nový_názov_reťaze;`
ale premenovať je však opäť možné iba užívateľsky definovanú reťaz, kde však ide skôr o kozmetickú úpravu.

### 1.5.4 Pridávanie pravidiel do reťazí

Všeobecný tvar príkazu, ktorý pridá filtrovacie pravidlo do reťaze vyzerá nasledovne:

```
iptables [-t tabuľka] -A reťaz špecifikácia_pravidla [nastavenia];
iptables [-t tabuľka] -I reťaz [číslo_pravidla] \
         špecifikácia_pravidla [nastavenia];
```

Rozdiel medzi parametrami `-A (--append)` a `-I (--insert)` je v tom, že parameter `-A` pridá podmienku na koniec reťaze a parameter `-I` pridá pravidlo na zadané poradie v reťazi, pričom pravidlá sa v reťazi číslujú od 1. Ak nebude poradie zadané, tak sa pravidlo pridá na začiatok reťaze.

Za všetkými podmienkami nasleduje parameter akcie `-j`, definujúci osud paketu, ktorý vyhovuje tomuto pravidlu. V prípade, že je pravidlo príliš dlhé, pretože obsahuje mnoho podmienok alebo parametrov rozšírenia, je možné použiť znak „\“, ktorý zabezpečí, že pravidlo bude (napr. zo súboru) prečítané ako celok. Znak „`!`“ slúži na negáciu (invertovanie) podmienky.

### Pravidlo pre protokol

Parameter `-p názov_protokolu` definuje podmienku, ktorá bude kontrolovať aký protokol IP paket prenáša. Ako `názov_protokolu` je možné uviesť hodnotu `all` (pre všetky protokoly), prípadne samotný názov jedného protokolu `tcp`, `udp` alebo `icmp`. Tento parameter prijíma aj názov alebo číslo ďalších protokolov, ale jeho názov musí byť uvedený v súbore `/etc/protocols`:

```
iptables -A INPUT -p icmp -j DROP;
```

Uvedené pravidlo spôsobí, že všetky prichádzajúce pakety patriace protokolu ICMP budú zahodené.

### Pravidlo pre zdrojovú/cieľovú adresu

Zdrojová adresa sa do podmienky definuje parametrom `-s`, cieľová adresa zase parametrom `-d`. Za tento parameter je nutné uviesť adresu s maskou siete (vo formáte napr. /255.255.255.0 alebo /24), prípadne iba adresu:

```
iptables -A INPUT -s 192.168.16.0/255.255.255.0 -j ACCEPT;
```

Pri IP adrese hosta nie je nutné masku uvádzať, použije sa maska 255.255.255.255:

```
iptables -A INPUT -d 10.0.0.1 -j DROP;
```

### Pravidlo pre vstupné/výstupné sieťové rozhranie

Na povolenie, prípadne zakázanie prijímania/odosielania paketov z určitého/na určité sieťové rozhranie, slúži parameter `-i` pre vstupné rozhranie a parameter `-o` pre výstupné rozhranie. Za tento parameter sa uvádza názov sieťového rozhrania:

```
iptables -A OUTPUT -o lo -j ACCEPT;
```

Toto pravidlo povoľuje odosielanie paketov na pseudozariadení, nazvanom rozhranie spätnej slučky (loopback). Spätná slučka bráni prieniku datagramov, ktoré systém posiela sám sebe na sieť. Namiesto toho sa transformujú priamo z výstupnej fronty do vstupnej fronty v jadre.[10]

Pravidlo pre bežné rozhranie, ktoré zahodí všetky IP pakety prichádzajúce rozhraním **eth0**, vyzerá nasledovne:

```
iptables -A INPUT -i eth0 -j DROP;
```

### Pravidlo pre zdrojový/cieľový port

Pravidlo s podmienkami pre zdrojový/cieľový port už využíva rozšírenia podmienok zhody a vyžaduje najprv použitie parametra pre protokol TCP alebo UDP. Parametrom `--source-port` definujeme podmienku pre zdrojový port a parametrom `--destination-port` podmienku pre cieľový port. Pre tieto parametre existujú aj aliasy `--sport` a `--dport`.

Nasledujúce pravidlo povoľuje pakety prichádzajúce na port 80, na ktorom počúvajú webové servery:

```
iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT;
```

Cez tento parameter je, samozrejme, možné definovať aj rozsah portov:

```
iptables -A OUTPUT -p udp -m udp --sport 12000:12099 -j DROP;
```

#### Pravidlo využívajúce sledovanie stavov spojení firewallom

Takýmto pravidlom je možné povoliť pakety, ktoré patria k už existujúcim spojeniam alebo naopak, filtrovať pakety, ktoré nepatria k žiadnym spojeniam. Pravidlá pre stavový firewall využívajú, podobne ako pravidlá pre porty, rozšírenia a definujeme ich parametrom `-m state --state zoznam_stavov`. Rozlišované stavy sú **ESTABLISHED**, **INVALID**, **NEW** a **RELATED**, pričom do príkazu je ich možné zadať viac a musia byť oddelené čiarkami bez medzier. Nasledujúce pravidlo povoľuje všetky IP pakety obsahujúce TCP datagramy patriace k už vytvoreným TCP spojeniam alebo tie, ktoré s nimi súvisia (riešenie problémov s FTP spojeniami a pod.):

```
iptables -I INPUT -p tcp -d 10.0.0.1 -m state \
--state ESTABLISHED,RELATED -j ACCEPT;
```

### 1.5.5 Mazanie pravidiel z reťazí a výmena pravidiel

Tak, ako bolo spomenuté už pri mazaní reťazí, pravidlá sa z reťaze mažú hromadne príkazom s parametrami `-F (--flush) názov_reťaze` alebo po jednom s parametrom `-D (--delete)`, pričom existujú dve verzie:

```
iptables -D reťaz špecifikácia_pravidla;
iptables -D reťaz poradové_číslo_pravidla;
```

Pravidlá sa v reťaziach číslujú zhora dole od čísla 1. Podobne je možné pravidlo vymeniť príkazom s parametrom `-R (--replace)`:

```
iptables -R reťaz poradové_číslo_pravidla nová_špecifikácia_pravidla;
```

### 1.5.6 Výpis pravidiel a práca s počítadlami

Pravidlá je možné zobraziť pomocou parametra `-L (--list)`, pridanie názvu reťaze za tento parameter spôsobí, že sa vypíšu len pravidlá zadanej reťaze:

```
iptables [-t tabuľka] -L [reťaz]
```

Odporúča sa pripojiť aj parameter `-n`, ktorý zabráni prekladu IP adries na ich doménové názvy:

```
iptables -t nat -n -L PREROUTING;
```

Vynulovanie počítadiel paketov a bajtov jednotlivých reťazí vykoná príkaz s parametrom `-Z (--zero)`, pričom opäť je možné pripojiť názov reťaze:

```
iptables [-t tabuľka] -Z [reťaz]
```

V prípade, že nebude zadaný za parameter `-Z` názov reťaze, vynulované budú počítadlá všetkých reťazí v danej tabuľke.

# 2 NÁVRH GRAFICKÉHO ROZHRANIA

## 2.1 Qt4 framework a vývojové nástroje

Qt je komplexný C++ framework určený na vývoj cross-platformných aplikácií, využívajúci prístup „napíš raz, skompiluj kdekoľvek“. Používa jeden strom zdrojových kódov a jednoduchú rekompiláciu, čo umožňuje písanie aplikácií pre Windows, Mac OS X, Linux, Solaris, HP-UX a ďalšie verzie Unixu používajúce X11. Knižnice a nástroje Qt sú taktiež súčasťou Qt Utopia Core, produktu, ktorý poskytuje vlastný systém okien nad platformou Embedded Linux. [1]

Qt má podporu pre multimédiá a 3D grafiku, multijazyčnosť, SQL, XML, unit testing a taktiež poskytuje špecifické rozšírenia viazané na platformu pre špecializované aplikácie. Framework Qt je základ, na ktorom bolo postavené grafické rozhranie KDE (K Desktop Environment) a mnoho ďalších open source aplikácií. [1][11]


Obr. 2.1: Logo Qt frameworku

### Licencie

Qt je dostupné pod rôznymi licenciami. V prípade komerčného použitia je potrebné zakúpiť komerčnú Qt licenciu. Naopak, na vývoj open source aplikácií je Qt dostupné pod GPL licenciou.

### Qt Designer

Qt Designer je návrhový nástroj na návrh a tvorbu grafických rozhraní s využitím Qt komponentov (widgetov). Aplikácie totiž môžu byť napísané kompletne ako zdrojový kód alebo je možné použiť práve Qt Designer.

Tento nástroj eliminuje časovo náročný cyklus „kompilácie, linkovania a spustenia“ pri návrhu grafických rozhraní. Toto umožňuje vývojárovi jednoducho skontrolovať a opraviť návrh rozhrania. Náhľadová funkcia zase umožňuje vývojárovi, napríklad na systéme Mac OS X vidieť formuláre v inom štýle, v štýle systému Windows alebo Linux a podobne. [11]

### Qt Linguist

Qt poskytuje výbornú podporu na preklad aplikácií do rôznych jazykov. Preklad aplikácie je možné rozdeliť až medzi tri osoby – manažéra vývoja, programátora a prekladateľa. Väčšina textu, ktorú je potrebné preložiť sú slová alebo krátke frázy, objavujúce sa ako názvy okien, položky v menu, vyskakovacie pomocné texty, názvy tlačidiel atď.

V prvom kroku píše programátor všetky texty v pôvodnom jazyku, ale pomocou jednoduchej syntaxe označí všetky tieto texty na ďalší preklad. Manažér vývoja následne vygeneruje sadu prekladateľských súborov a tieto odovzdá prekladateľovi. Ten všetky tieto sady súborov s textami preloží a vráti manažérovi vývoja. Ten potom vygeneruje ďalšiu kompaktnú verziu prekladu, ktorú už následne využije samotná aplikácia na zmenu jazyka. Tieto kroky je možné vykonávať v niekoľkonásobných cykloch, pri vývoji a zmene aplikácie, pričom sú vždy zachované už preložené texty a texty, ktoré ešte neboli preložené, sú zase jednoducho identifikovateľné.

### Qt Assistant

Qt Assistant je nástroj, ktorý pracuje podobne ako webový prehliadač a obsahuje nielen kompletnú Qt dokumentáciu (Qt reference documentation), ale aj manuály ostatných nástrojov ako je Qt Designer alebo Qt Linguist.

Samotná dokumentácia obsahuje krížové odkazy, vďaka ktorým je čítanie dokumentácie veľmi pohodlné a jednoduché, bez ďalšieho dohľadávania. Qt Assistant indexuje každý súbor dokumentácie fulltextovým vyhľadávacím enginom a následne poskytuje možnosť vyhľadávania slov, resp. fráz v celej dokumentácii. [11]

### Qt Creator

Qt Creator je kompletné integrované vývojové prostredie (IDE) na tvorbu aplikácií pomocou Qt. Qt Creator beží na každej platforme podporovanej samotným Qt frameworkom. Predkompilovaná aplikácia je dostupná pre Microsoft Windows, Mac OS X a Linux. Toto vývojové prostredie je možné spustiť aj na iných platformách, ale predtým je nutná jeho kompilácia z verejne dostupných zdrojových kódov.

Kompletný Qt SDK (Software Development Kit) vývojový kit obsahuje Qt Creator, najnovšiu verziu Qt frameworku a zároveň aj pomocné nástroje Qt Assistant, Qt Designer a Qt Linguist. Nástroje Qt Assistant a Qt Designer sú priamo integrované do rozhrania Qt Creator, prípadne je možné všetky tieto nástroje spustiť samostatne. [11]

## 2.2 *qIPtables* – grafické rozhranie pre *iptables*

### 2.2.1 Postup pri návrhu aplikácie *qIPtables*

Základnými predpokladmi pre vytvorenie grafického rozhrania boli znalosti objektovo orientovaného programovania v jazyku C++, s ktorým sme sa stretli už počas štúdia. Ďalej bolo potrebné naštudovať si manuálové stránky nástroja *iptables* [3], a tiež niekoľko tutoriálov, prípadne návodov tvorby firewallu v Linux-e pomocou tohto nástroja. V neposlednom rade bolo nutné naštudovanie dokumentácie ku Qt4 frameworku [11] a tiež literatúry [1], ktorá výborným a veľmi názorným spôsobom nielen popisuje tvorbu grafických rozhraní v C++ pomocou Qt, ale zároveň tiež prináša vzorové zdrojové kódy, ktoré veľmi uľahčujú prvé kroky pri tvorbe grafických rozhraní pomocou nástroja Qt Designer.

Pri návrhu a programovaní grafického rozhrania sme použili vývojové prostredie Qt Creator 1.2.1 a Qt4 framework vo verzii 4.6.2. Všetko nainštalované, samozrejme, pod operačným systémom Linux, na distribúcii Kubuntu 10.04 LTS. Ako odrazový mostík sme použili vzorovú ukážku tvorby grafickej aplikácie z literatúry [1]. Týmto zdrojovým kódom sme následne doprogramovali potrebnú funkcionalitu, vytvorili pomocou nástroja Qt Designer centrálnu časť rozhrania, dialógy na zmenu predvolenej politiky reťaze, pridanie reťaze, premenovanie reťaze, pridanie pravidla a dialóg na zmenu nastavení aplikácie. Hlavné okno grafického rozhrania však nebolo vytvorené pomocou nástroja Qt Designer. To znamená, že obsah hlavného okna, ktorý okrem centrálnej časti tvorí panel menu a stavový riadok bol naprogramovaný ako bežný zdrojový kód, pričom stredná časť (vytvorená v Qt Designer) je do hlavného okna v tomto zdrojovom kóde iba vložená. Výsledkom je grafické rozhranie, ktoré dostalo názov *qIPtables*.

### 2.2.2 Popis *qIPtables*

Výsledkom tejto práce je aplikácia (grafické rozhranie), ktorá slúži ako grafická nadstavba pre nástroj *iptables* tak, ako to vyžaduje zadanie práce. Presnejšie povedané, táto aplikácia dokáže vizuálne zobraziť štruktúru *iptables*, ktorá obsahuje 4 spomínané tabuľky, z ktorých každá zase obsahuje svoje vstavané reťaze a tie obsahujú jednotlivé pravidlá. Zároveň umožňuje plne ovládať celý nástroj *iptables* tým, že na základe akcií užívateľa v grafickom rozhraní generuje príkazy pre *iptables*, tie zadáva do systému akoby ich zadal sám užívateľ a následne zobrazí ich výsledok. Zobrazenie výsledku znamená, že ak pomocou grafického rozhrania pridáme pravidlo do niektorej z reťazí, tak sa toto pravidlo takmer okamžite zobrazí vo vizuálnej podobe štruktúry firewallu v aplikácii *qIPtables*. Tým všetkým sa však budeme zaoberať v časti, ktorá slúži ako manuál k aplikácii.

### 2.2.3 Inštalácia závislostí a kompilácia zdrojových kódov

Pred používaním aplikácie je nutné nainštalovať kompilátor zdrojových kódov g++, doinštalovať závislosti, ktoré *qIPtables* potrebuje k behu (ak ešte v systéme nie sú nainštalované) a následne aplikáciu zo zdrojových kódov skompilovať.

#### Inštalácia kompilátora g++ a Qt4 frameworku do Kubuntu

Na inštaláciu kompilátora g++ a Qt4 frameworku je možné použiť repozitáre distribúcie Ubuntu, ktoré sú zároveň aj repozitármi distribúcie Kubuntu. Do otvoreného terminálu zadáme nasledujúce príkazy, ktoré doinštalujú GNU C++ kompilátor g++ a vývojové prostredie Qt Creator. Qt Creator automaticky nainštaluje aj svoje závislosti a teda celý Qt4 framework:

```
user@kubuntu:~$ sudo apt-get install g++ qtcreator
```

V prípade, že nechceme inštalovať do systému Qt Creator a ostatné nástroje, ale iba kompilátor g++ a základné knižnice Qt4, zadáme do terminálu príkaz:

```
user@kubuntu:~$ sudo apt-get install g++ libqt4-dev
```

Po ukončení inštalácie v prvom alebo v druhom prípade by sme mali mať v systéme všetko potrebné na kompiláciu zdrojových kódov aplikácie *qIPtables*.

#### Doplnenie ostatných závislostí pre *qIPtables* a oprava IBUS

Funkčnosť *qIPtables* závisí na grafickom nástroji, ktorý umožňuje spúšťať v systéme aplikácie s právami superužívateľa. Medzi takéto nástroje patrí *gksu*, *gksudo* alebo *kdesudo*. Nástroje *gksu* a *gksudo* sa nachádzajú vo východzej inštalácii distribúcie Ubuntu a nástroj *kdesudo* zase vo východzej inštalácii distribúcie Kubuntu. Preto je nutné potom *qIPtables* správne nakonfigurovať, ktorý z týchto nástrojov má *qIPtables* používať na zadávanie príkazov do systému. V predvolenom nastavení používa aplikácia nástroj *gksu*. Prípadne do distribúcie Kubuntu nástroj *gksu* doinštalovať:

```
user@kubuntu:~$ sudo apt-get install gksu
```

Mezdi ďalšie závislosti *qIPtables* patria príkazy `update-rc.d`, `mv`, `chown`, a `chmod`, ktoré slúžia na prácu so súbormi a pod., pričom bývajú v zdravej inštalácii distribúcie vždy prítomné.

V súčasnej distribúcii Kubuntu 10.04 LTS dochádza k malému problému, ktorý spôsobuje chybové hlásenie o vstupnej metóde IBus. Odporúčame preto IBus doništalovať a nastaviť ako ho vstupnú metódu príkazmi:

```
user@kubuntu:~$ sudo apt-get install ibus ibus-pinyin
user@kubuntu:~$ im-switch -s ibus
```

a potom systém zreštartovať, čím by sa chybové hlásenie viac zobrazovať nemalo.

#### Kompilácia zdrojových kódov

Kompilácia zdrojových kódov nie je náročná, ale podľa výkonu počítača zaberie viac alebo menej času. Tieto zdrojové kódy je možné nájsť na DVD disku, ktorý je súčasťou tejto práce. V termináli prejdeme do adresára so skopírovanými zdrojovými kódmi a postupne zadáme tieto tri príkazy:

```
user@kubuntu:~/qIPtables$ qmake -project
user@kubuntu:~/qIPtables$ qmake
user@kubuntu:~/qIPtables$ make
```

Prvý príkaz `qmake -project` vytvorí podľa obsahu adresára so zdrojovými kódmi Qt projekt – súbor s príponou `.pro`. V našom prípade to bude súbor `qIPtables.pro`. Podľa tohto projektu následne príkaz `qmake` vytvorí pre g++ kompilátor súbor `Makefile` a nakoniec príkaz `make` podľa súboru `Makefile` skompiluje zdrojové kódy.

Po kompilácii zdrojových kódov vznikne binárny spustiteľný súbor `qIPtables`, ktorý je samotným grafickým rozhraním pre nástroj *iptables*.

## 2.3 Manuál k aplikácii *qIPtables*

V predchádzajúcej časti sme nainštalovali do systému všetky potrebné závilosti pre aplikáciu *qIPtables*, skompilovali jej zdrojové kódy, čím vznikol spustiteľný súbor, ktorý už slúži ako ktorákoľvek iná aplikácia v systéme. V tejto časti popíšeme používanie tejto aplikácie, jej možnosti, funkcie a nastavenie. Aplikácia je lokalizovaná v anglickom, slovenskom a českom jazyku, pričom jazyková mutácia sa volí automaticky podľa aktuálneho nastavenia jazyka a regiónu operačného systému.

### 2.3.1 Prvé spustenie

Aplikáciu *qIPtables* spustíme bežným spôsobom – dvojklikom z adresára, v ktorom sme skompilovali zdrojové kódy. Počkáme, kým sa aplikácia spustí a zobrazí sa hlavné okno programu. Ak používame pracovné prostredie KDE (Kubuntu 10.04 LTS), bude *qIPtables* po spustení vyzerať ako v prílohe B.1. V pracovnom prostredí GNOME a iných bude *qIPtables* vyzerať podobne, rozdiel bude iba v dekoráci samotného okna, čo závisí nielen na používanom pracovnom prostredí, ale aj na tom, ktorý manažér a dekorátor okien toto pracovné prostredie využíva.

*qIPtables* si počas štartu kontroluje v systéme prítomnosť súboru s nastaveniami, ktorý hľadá v domácom adresári užívateľa v podadresári `.qiptables/`. Ak je tento súbor nenájdený, automaticky sa tento adresár vytvorí v domovskom adresári užívateľa, ktorý aplikáciu spustil.

V ňom sa potom vytvorí ešte súbor s predvolenými nastaveniami. Predvolené nastavenia znamenajú, že počas ďalšej kontroly prítomnosti závislostí sa bude hľadať binárny súbor shellu `/bin/bash`, súbor grafického sudo `/usr/bin/gksu` a súbory nástroja *iptables* – `iptables`, `iptables-save`, `iptables-restore` v adresári `/sbin`. Zároveň budú parametre `[IP-FORWARDING]` a `[SPOOF-PROTECTION]` nastavené na `no`. Ukážka súboru s predvolenými nastaveniami, ktorý vygeneruje *qIPtables* sa nachádza v prílohe C.

Či už bude súbor s nastaveniami nájdený alebo ho aplikácia vytvorí, v ďalšom kroku dôjde ku kontrole, či všetky tieto súbory existujú. Ak sa niektorý z nich nenájde, aplikácia zobrazí chybové hlásenie, že nie sú všetky závislosti splnené a že ich je nutné doinštalovať, prípadne len skontrolovať nastavenia a opraviť cesty k jednotlivým súborom. V tomto prípade sa síce hlavné okno aplikácie zobrazí, ale niektoré jeho časti budú nefunkčné – tlačidlá budú sivé a nebude ne možné kliknúť a nezobrazí sa stromová štruktúra firewallu. Až po opravení konfigurácie (doinštalovaní závislostí) bude hlavné okno plne funkčné a bude vyzerať ako v prílohe B.1.

### 2.3.2 Hlavné okno

Hlavné okno popíšeme v pôvodnej anglickej verzii, pričom stále platí, že aplikácia sa na systémoch s nastavením regiónu a jazyka na slovenský alebo český zobrazí preložená. K tomu sa využívajú pomocné súbory `qIPtables_sk.qm` a `qIPtables_cs.qm`, ktoré obsahujú preklad celej aplikácie. Tieto preklady boli vytvorené v nástroji Qt Linguist.

Vo vrchnej časti okna sa nachádza panel, ktorý obsahuje menu **qIPtables** a menu **Help**. Menu **Help** obsahuje položky **iptables manpage**, **About** a **About Qt**.

Položka **About** zobrazí jednoduché dialógové okno s informáciami o aplikácii a položka **About Qt** zase dialógové okno s informáciami o použitej verzii knižníc Qt4. Súčasťou aplikácie sú aj manuálové stránky *iptables*, ktoré sú prevzané z webu [3] a tieto manuálové stránky sú prístupné práve cez položku **iptables manpage** (príloha B.9).

Prvé dve položky menu **qIPtables** umožňujú uloženie súčasných pravidiel *iptables* do súboru (**iptables-save to file**) a tiež opätovné načítanie týchto pravidiel zo súboru do jadra (**iptables-restore from file**) rovnako, ako keby sme použili príkazy `iptables-save` a `iptables-restore` v bežnom termináli systému. Toto môže byť praktické, ak už máme nejakú sadu pravidiel uloženú v súbore a nechceme všetky tieto pravidlá zadávať znova od prvého do systému cez aplikáciu *qIPtables*. Takto jednoducho pravidlá zo súboru načítame do jadra a prípadne potom doplníme ďalšie cez grafické rozhranie. Uloženie pravidiel do súboru môže zase poslúžiť ako záloha alebo ako súbor, pomocou ktorého zadáme pravidlá do ďalších počítačov.

Pomocou položky **(Re)Install Firewall Startup Script** je možné nechať aplikáciu vygenerovať súbor so skriptom, ktorý bude umiestnený do `/etc/init.d/` a príkazom `update-rc.d` zaregistrovaný tak, aby pri štarte operačného systému zapísal do jadra všetky pravidlá, ktoré tam boli v momente, keď sme nechali tento skript vygenerovať. Tento skript rieši problém, že pravidlá firewallu sa stratia pri vypnutí alebo reštartovaní. Táto funkcia bola otestovaná v distribúciach Ubuntu a Debian, na ostatných distribúciach fungovať nebude.

Tento aplikáciou generovaný skript je vytvorený podľa skriptov zo stránok [12]. Skript bol mierne upravený, aby ho bolo možné z *qIPtables* vygenerovať a tiež boli doplnené LSB informácie [2], ktoré sú odporúčané na registráciu startup skriptov príkazom `update-rc.d`.

Položka **Preferences** slúži na prístup k nastaveniam aplikácie (príloha B.8) a položka **Exit** aplikáciu zatvorí bežným spôsobom, rovnako ako krížik v pravom hornom rohu hlavného okna (ľavom hornom v prípade Ubuntu 10.04 LTS).

V strednej časti hlavného okna aplikácie nájdeme stromovú štruktúru zobrazujúcu aktuálne pravidlá a reťaze firewallu a pod týmto stromom usporiadané tlačidlá, ktoré pridajú a zmažú reťaz, pridajú a zmažú pravidlo, vyčistia reťaz, vynulujú počítadlá a pod. Túto stromovú štruktúru je možné jednoducho rozbaliť a zbaliť tlačidlami **Expand tree view** a **Collapse tree view**.

V spodnej časti okrem stavového riadka nájdeme aj tzv. vstavanú konzolu, ktorá oznamuje užívateľovi všetky vykonávané akcie. To znamená, že ako sa počas štartu vykonávajú jednotlivé akcie načítania nastavení (prípadne vytvorenie tej predvolenej), kontroly potrebných súborov a načítanie firewallu, tak všetky tieto akcie aplikácia oznamuje výpisom do tejto konzoly. Ako už bolo spomenuté skôr, aplikácia generuje na základe akcií užívateľa v hlavnom okne príkazy, ktoré následne zadáva do systému a toto, samozrejme, tiež vypisuje do konzoly. V prípade chyby vykonania príkazu aplikácia zobrazí chybové hlásenie. Obsah tejto konzoly je možné vymazať kliknutím na tlačidlo **Clear console**.

Vedľa tohto tlačidla sa nachádza ešte tlačidlo **Reload firewall**, ktoré z jadra operačného systému znovu načíta pravidlá firewallu a obnoví stromové zobrazenie.

### 2.3.3 Nastavenie *qIPtables*

Okno s nastaveniami aplikácie vyvoláme kliknutím na položku **Preferences** v menu **qIPtables**. Nastaviť je možné systémové premenné, ktoré povoľujú preposielanie paketov medzi rozhraniami a ochranu proti spoofingu, a cesty k jednotlivým súborom (príkazom), ktoré *qIPtables* využíva na správu firewallu. Vzhľad dialógového okna s predvolenými nastaveniami sa nachádza v prílohe B.8.

### 2.3.4 Správa firewallu pomocou *qIPtables*

#### Nastavenie štandardnej politiky

Začneme zmenou predvolenej politiky vstavaných reťazí. Každá vstavaná reťaz má nastavenú predvolenú politiku, ktorá určuje, čo sa s paketom stane, ak nevyhovie ani jednému z pravidiel tejto reťaze. Táto politika môže byť nastavená na hodnotu **ACCEPT** alebo **DROP**. V grafickom rozhraní *qIPtables* sa predvolená politika nastavuje tlačidlom **Change default policy**, pričom je nutné predtým v stromovej štruktúre označiť, ktorej reťazi chceme politiku zmeniť. *iptables* neumožňuje nastaviť politiku užívateľsky vytvoreným reťaziam a preto v prípade, že bude označená užívateľsky vytvorená reťaz, *qIPtables* zobrazí chybové hlásenie. Toto chybové hlásenie sa zobrazí aj v prípade, že bude označená tabuľka alebo pravidlo.

Pri správnom výbere po kliknutí na tlačidlo **Change default policy** sa zobrazí dialógové okno ako v prílohe B.2. Tu môžeme zvoliť akú politiku chceme zvolenej reťazi nastaviť. Z možností **DROP** alebo **ACCEPT** vyberieme požadovanú politiku a následne klikneme na tlačidlo **Change default policy**. Teraz aplikácia zavolá príkaz *iptables*, ktorý pre vybranú reťaz nastaví zvolenú politiku a zobrazí zmenu v stromovej štruktúre.

#### Vynulovanie počítadiel paketov a bajtov

Ak chceme vynulovať počítadlá paketov a bajtov nejakej reťaze, tak túto reťaz označíme a klikneme na tlačidlo **Reset counters**. Zobrazí sa okno s otázkou, či naozaj chceme vynulovať počítadlá pre vybranú reťaz z danej tabuľky a po kliknutí na tlačidlo **Yes** dôjde opäť k vykonaniu príkazu a zobrazeniu zmien. Podobne je možné vynulovať počítadlá pre úplne všetky reťaze kliknutím na tlačidlo **Reset all counters**, pričom je túto akciu nutné opäť potvrdiť tlačidlom **Yes**.

#### Pridanie, premenovanie a zmazanie užívateľsky definovanej reťaze

Ak chceme pridať (vytvoriť) novú reťaz, musíme najprv označiť tabuľku, do ktorej chceme novú reťaz pridať a následne kliknúť na tlačidlo **Create user–defined chain**. Následne sa zobrazí dialógové okno, do ktorého napíšeme názov novej reťaze a klikneme na tlačidlo **Create user–defined chain** (príloha B.3). Tým sa vytvorí nová reťaz, ktorú môžeme potom tlačidlom **Rename user–defined chain** premenovať (príloha B.4) alebo tlačidlom **Delete user–defined chain** zmazať (príloha B.5). Stále však platí, že premenovať alebo zmazať je možné iba užívateľsky vytvorenú reťaz, nikdy nie reťaz vstavanú a tiež nie je možné zmazať reťaz, ktorá obsahuje nejaké pravidlo. Pred zmazaním je teda z takejto reťaze nutné zmazať všetky pravidlá.

### Hromadné zmazanie pravidiel a reťazí

Pravidlá je, samozrejme, možné mazať postupne, jedno po druhom, ale v prípade, že chceme zmazať užívateľsky vytvorenú reťaz a tá obsahuje niekoľko desiatok pravidiel, bolo by postupné mazanie zbytočne zdĺhavé a pomalé. Na hromadné zmazanie pravidiel z nejakej reťaze (vstavanej aj užívateľskej) slúži tlačidlo **Flush chain**, ktoré zobrazí potvrdzovacie okno a po potvrdení zmaže z vybranej reťaze všetky pravidlá. Podobne funguje tlačidlo **Flush and delete everything**, ktoré zmaže všetky pravidlá zo všetkých reťazí a tiež všetky užívateľsky vytvorené reťaze vo všetkých tabuľkách.

### Vytvorenie a zmazanie pravidla

Pridávanie pravidiel je okrem grafického zobrazenia stromovej štruktúry tabuliek, reťazí a pravidiel firewallu asi najdôležitejšou funkciou celej aplikácie. Pravidlo môže totiž obsahovať niekoľko podmienok, ich rozšírení a tiež akciu a nastavenie jej rozšírenia, čo v prípade tvorby príkazov v termináli vyžaduje určitú znalosť správnej syntaxe a tiež parametrov, ktoré samotné pravidlo vytvoria. Aplikácia *qIPtables* pridanie pravidla zjednodušuje tým, že kliknutím na tlačidlo **Add rule** zobrazí dialógové okno, kde je možné vybrať jednotlivé podmienky a nastaviť akciu aj s rozšíreniami, pričom samotné pravidlo je generované okamžite a postupne podľa akcií užívateľa a po potvrdení tlačidlom **Add rule** je toto pravidlo pridané do firewallu.

Rozdiel je však v tom, či označíme nejakú reťaz a klikneme na tlačidlo **Add rule** alebo označíme niektoré pravidlo a klikneme na toto tlačidlo. Pri označení reťaze bude potom nové pravidlo pridané na koniec reťaze, ale pri označení niektorého pravidla bude nové pravidlo vložené na miesto označeného pravidla. To je výhodné ak chceme pravidlo pridať, napríklad, do stredu alebo na začiatok reťaze.

Dialógové okno, v ktorom je možné nové pravidlo vytvoriť je rozdelené na lišty **Match and target** a **Match extensions (optional)** a tieto lišty sú zobrazené v prílohe B.6. Funkčnosť položiek v týchto lištách však závisí na tom, do ktorej reťaze sa snažíme pravidlo pridať. Napríklad, pravidlo s podmienkou pre vstupné rozhranie (alebo s podmienkou pre MAC adresu) je možné pridať iba do reťazí **PREROUTING**, **INPUT** a **FORWARD**, a pravidlo s podmienkou pre výstupné rozhranie zase iba do reťazí **FORWARD**, **OUTPUT** a **POSTROUTING**. Zároveň sú však jednotlivé rozšírenia podmienok protokolov aktívne podľa toho, či je aktívna podmienka pre daný protokol (viď. prílohy B.6 a B.7).

Podmienku do pravidla pridáme tak, že zaškrtneme dané políčko (napr. **Protocol**, **Source Address**, **Destination Interface** a pod.), čím sa aktivujú ďalšie zaškrtávacie a vyplňovanie políčka, ktoré vytvoria jednu podmienku. Pri vyplňovacom políčku protokolu sa nachádza aj rolovacie menu, v ktorom je možné vybrať niektorý

z protokolov a ten sa následne vloží vedľa do vyplňovacieho políčka. Takto je možné vybrať napríklad protokol TCP alebo UDP. Toto rolovacie menu obsahuje aj položku <**other**>, ktorá indikuje, že do vyplňovacieho políčka je možné napísať ktorýkoľvek iný názov alebo číslo protokolu, ak sa nachádza v súbore `/etc/protocols`.

Manuálové stránky obsahujú popis veľkého množstva rozšírení podmienok, a nebolo možné všetky tieto rozšírenia zahrnúť do dialógového okna, takže ako náhradu je možné zaškrtnúť políčko **Other Match Extension** a do vyplňovacieho políčka toto rozšírenie zapísať. Je nutné si však uvedomiť, že tieto rozšírenia sú závislé na momentálne používanej verzii linuxového jadra a používanej verzii nástroja *iptables*. Niektoré z nich sú však dokonca iba experimentálne a niektoré nefungujú na systémoch SMP.[3]

Obsah rolovacieho menu s výberom akcie pravidla taktiež závisí na tom do ktorej reťaze ktorej tabuľky pravidlo pridávame. Keďže ako akcia môže slúžiť aj užívateľská reťaz, tak aj ako akciu je možné nastaviť niektorú z užívateľských reťazí danej tabuľky. Nie je však možné vytvoriť nekonečnú slučku, kedy pravidlo v reťazi **test** presmeruje spracovanie paketu opäť do reťaze **test**. Nástroj *iptables* zadanie príkazu odmietne s chybovým hlásením a preto *qIPtables* ani pridanie takéhoto pravidla neumožňuje – v rolovacom menu sa daná užívateľská reťaz neobjaví.

Akciu **NOTRACK** je možné zadať iba v pravidle, ktoré bude v niektorej z reťazí tabuľky **raw**, akciu **DNAT** zase iba v reťazi **PREROUTING** tabuľky **nat**, akcie **SNAT** a **MASQUERADE** iba v reťaziach **OUTPUT** a **POSTROUTING** tabuľky **nat** a podobne. [3]

Niektoré akcie umožňujú ešte ich dodatočné nastavenie. Akcia **REJECT**, ktorá zamietne paket a jeho odosielateľovi to oznámi ICMP správou, umožňuje nastaviť typ ICMP správy, ktorá sa odošle v prípade, ak bude nejaký paket zamietnutý. Preto sa pri výbere akcie **REJECT** aktivuje aj ďalšie rolovacie menu v ktorom je možné vybrať položku **--reject-with**, ktorá aktivuje vyplňovanie políčko, do ktorého je možné zapísať typ správy, ktorá sa má pri zamietnutí paketu danou podmienkou odoslať späť odosielateľovi. Typy správ je možné nájsť v manuálových stránkach *iptables* a tiež sa zobrazia ak necháme kurzor nad týmto políčkom – tzv. tooltip.

Takýto pomocník (tooltip) sa zobrazuje nad väčšinou prvkov dialógového okna na pridávanie pravidla a umožňuje tak lepšie porozumieť danej podmienke a hlavne správne vyplniť potrebné údaje. Pomocou pre užívateľa by mali byť aj masky, ktoré umožnia iba správne vyplnenie políčok pre IP a MAC adresy. Zaškrtávacie políčko **Invert** slúži vždy ako negácia (invertovanie) daného parametra.

A nakoniec, zmazanie pravidla vykonáme označením pravidla, kliknutím na tlačidlo **Delete rule** a potvrdením potvrdzovacieho okna.

# 3 ZÁVER

Výsledkom tejto práce je aplikácia *qIPtables* – plnohodnotné grafické rozhranie pre nástroj *iptables*. Táto aplikácia umožňuje ovládať firewall v jadre operačného systému Linux rovnako ako samotný nástroj *iptables*, no používanie grafického rozhrania, a teda aplikácie *qIPtables* prináša vyšší komfort pre užívateľov, ktorí majú radšej grafické ako terminálové aplikácie.

Grafické rozhranie je napísané v jazyku C++ s využitím prvkov objektovo orientovaného programovania a s využitím Qt4 frameworku. Qt4 je výborný framework na tvorbu grafických rozhraní, a nielen to. Prináša množstvo vlastných tried a metód, ktoré rozširujú možnosti programátora pri tvorbe, či už grafických, alebo terminálových aplikácií pre veľké množstvo platforiem, pričom samotný zdrojový kód je možné skompilovať na ktorejkoľvek platforme. Tvorbu takýchto aplikácií uľahčujú aj vývojové nástroje, ktoré sú dodávané s Qt4 a ide hlavne o integrované vývojové prostredie Qt Creator, ktoré v sebe zahŕňa okrem editora zdrojového kódu, ktorý zvýrazňuje syntax typickú pre Qt4, Debuggera a množstva príkladov aj pomocné nástroje Qt Designer na tvorbu grafických okien a dialógov, bez napísania jediného riadku zdrojového kódu, a tiež nástroj Qt Assistant, ktorý slúži ako výborný pomocník pri hľadaní dokumentácie ku Qt4 frameworku.

Práca zahŕňa aj postup na inštaláciu potrebných súčastí na kompiláciu a beh aplikácie *qIPtables* ako je GNU C++ kompilátor g++ a, samozrejme, Qt4 Framework alebo prípadne iba jeho knižnice. Asi najdôležitejšou časťou práce je manuál k samotnej aplikácii *qIPtables*, ktorý toto grafické rozhranie nielen popisuje, ale prináša aj postup, ako je možné firewall v operačnom systéme Linux aplikáciou *qIPtables* spravovať.

# LITERATURA


[1] BLANCHETTE, Jasmin; SUMMERFIELD, Mark. *C++ GUI Programming with Qt 4* [online]. Stoughton (Massachusetts) : Courier, 2006 [cit. 2010–05–17]. Dostupné z WWW: <http://www.scribd.com/doc/12993777/>. ISBN 0-13-187249-4.

[2] *DebianWiki* [online]. 2005–08–24, last edited 2010-05-06 [cit. 2010-05-31]. How to LSBize an Init Script. Dostupné z WWW: <http://wiki.debian.org/LSBInitScripts>.

[3] *Die.net* [online]. 1996 [cit. 2010–05–12]. Iptables(8) – Linux man page. Dostupné z WWW: <http://linux.die.net/man/8/iptables>.

[4] Firewall In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, [cit. 2010–05–03]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Firewall>.

[5] *Firewalls + Security* [online]. 2005 [cit. 2009–12–16]. Dostupné z WWW: <http://linuxzoo.net/page/firewall.html>.

[6] HARRISON, Peter. *Linux Home Networking* [online]. 2007 [cit. 2009–12–05]. Dostupné z WWW: <http://www.linuxhomenetworking.com/wiki>.

[7] History of Linux In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, [cit. 2010–05–12]. Dostupné z WWW: <http://en.wikipedia.org/wiki/History_of_Linux>.

[8] *Linux 2.4 Packet Filtering HOWTO* [online]. 2002 [cit. 2009–11–17]. Dostupné z WWW: <http://www.iptables.org/documentation/index.html>.

[9] *Linux dokumentační projekt.* 3. aktualiz. vyd. Brno : Computer Press, 2003. 1001 s. ISBN 80–7226–761–2

[10] NEMETH, E., SNYDER, G., HEIN T. *Linux – Kompletní příručka administrátora.* Computer Press, 2004. 880 s. ISBN: 80–722–6919–4.

[11] Nokia Corporation and/or its subsidiary(–ies). *Qt 4.6: Qt Reference Documentation* [online]. [cit. 2009–12–12]. Dostupné z WWW: <http://doc.trolltech.com/4.6/>.

[12] NORIS, I. *Príručka systémového administrátora* [online]. 2002–2007, Posledná zmena: 26/3/2007 [cit. 2010-05-17]. Firewall. Dostupné z WWW: <http://deja-vix.sk/sysadmin/firewall.html>.

[13] PŘIBYL, Adam. *Proč používat Linux* [online]. 2007 [cit. 2010–05–12]. Výhody operačního systému Linux.
Dostupné z WWW: <http://proc.linux.cz/vyhody.html>.

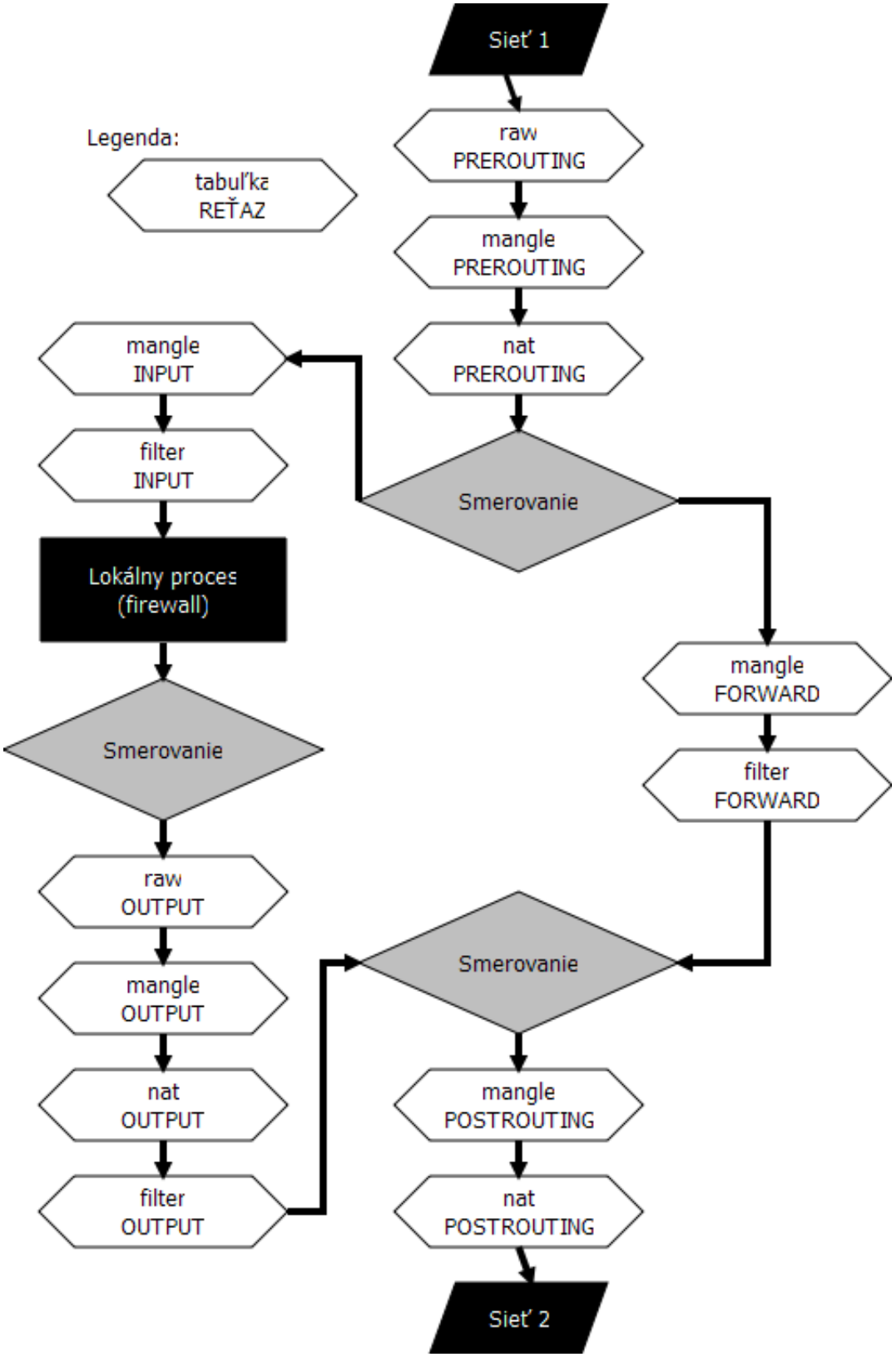
[14] RASH, Michael. *Linux firewalls : Attack Detection and Response with iptables, psad, and fwsnort.* 1st edition. San Francisco : No Starch Press, Inc., 2007. 311 s. ISBN 978–1–59327–141–1.

# ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

| | |
|---|---|
| BSD | Berkeley Software Distribution – jedna z derivácií pôvodného UNIXu |
| DMZ | Demilitarized Zone – miesto za firewallom alebo medzi firewallmi so zvýšenou bezpečnosťou |
| DNAT | Destination Network Address Translation – preklad cieľovej sieťovej adresy |
| GNOME | GNU Network Object Model Environment – pracovné rozhranie pre UNIXové operačné systémy |
| GNU | rekurzivná skratka GNU's Not Unix! označujúca slobodný softvér |
| IBus | Intelligent Input Bus – framework multijazyčnej vstupnej metódy pre UNIXové operačné systémy |
| IDE | Intergrated Development Environment – vývojové prostredie na tvorbu softvéru |
| IDS | Intrusion Detection System – systém na detekciu prienikov do systému |
| KDE | K Desktop Environment – pracovné rozhranie pre UNIXové operačné systémy |
| LTS | Long Time Support – označenie distribúcií Ubuntu s predĺženou podporou až na 5 rokov |
| MAC | Media Access Control address – fyzická adresa pre sieťové rozhrania |
| NAT | Network Address Translation – preklad sieťových adries, napr. pri nedostatku voľných IP adries |
| OSI | Open System Interconnection – model sieťovej komunikácie od organizácie ISO |
| SDK | Software Development Kit – vývojový balík na tvorbu softvéru obsahujúci potrebné knižnice a nástroje |
| SNAT | Source Network Address Translation – preklad zdrojovej sieťovej adriesy |
| UNIX | operačný systém vyvinutý v roku 1969, z ktorého neskôr vznikli ďalšie operačné systémy |

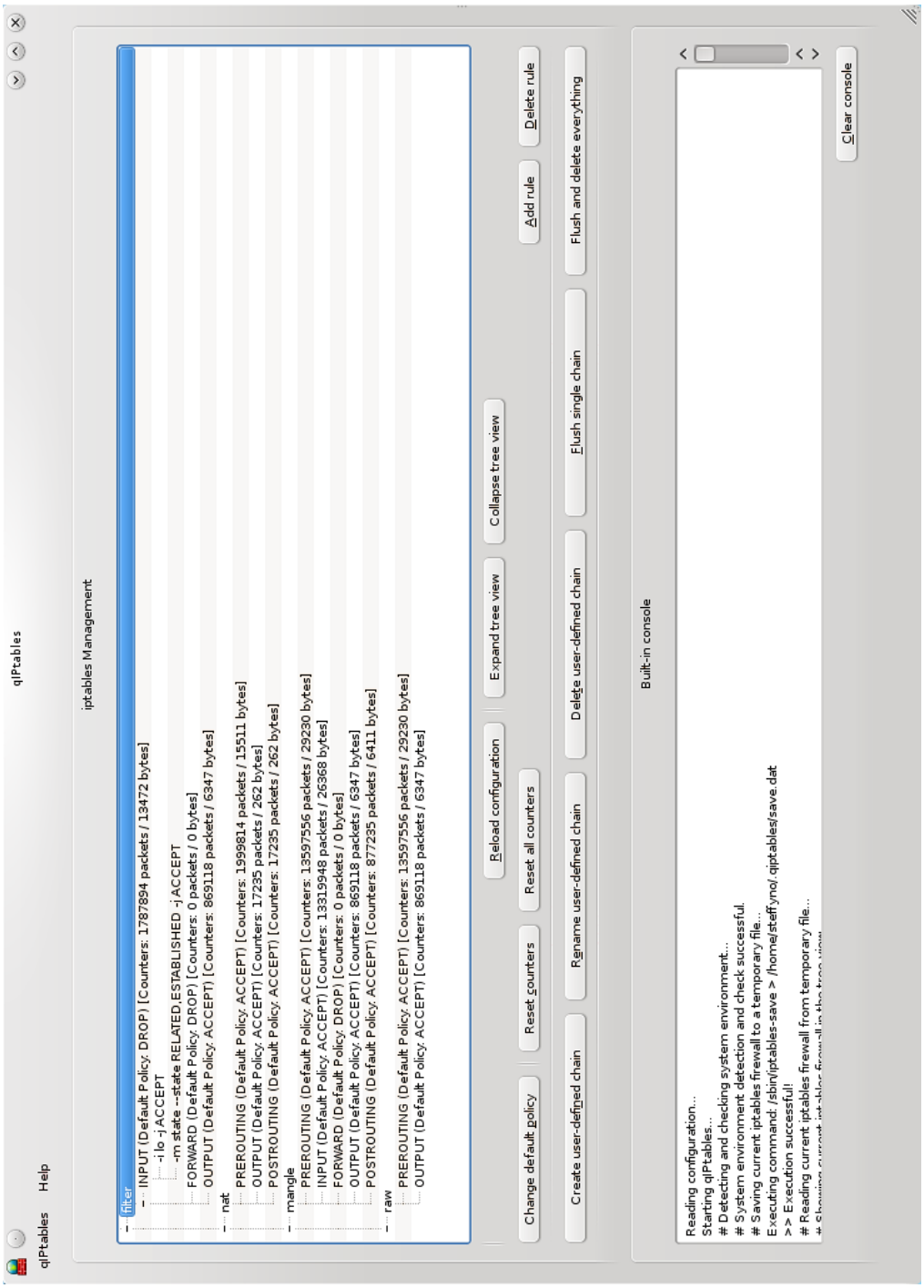
# ZOZNAM PRÍLOH

# A ŠTRUKTÚRA REŤAZÍ FIREWALLU

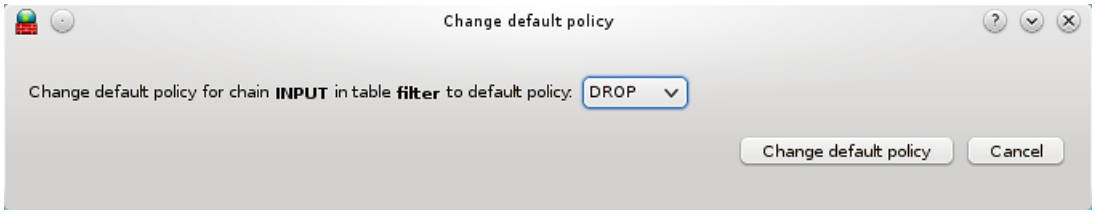


# B NÁHĽADY GRAFICKÉHO ROZHRANIA

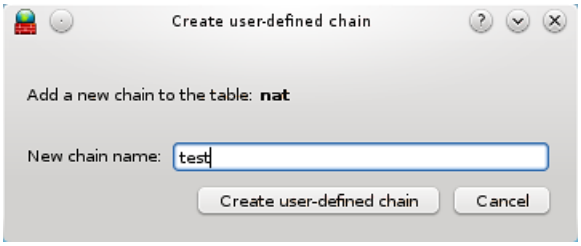
## B.1 Hlavné okno *qIPtables*



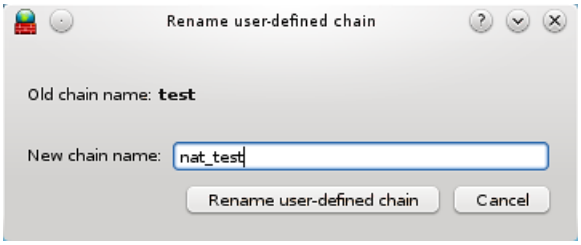
## B.2 Zmena predvolenej politiky vstavanej reťaze




## B.3 Pridanie užívateľsky definovanej reťaze

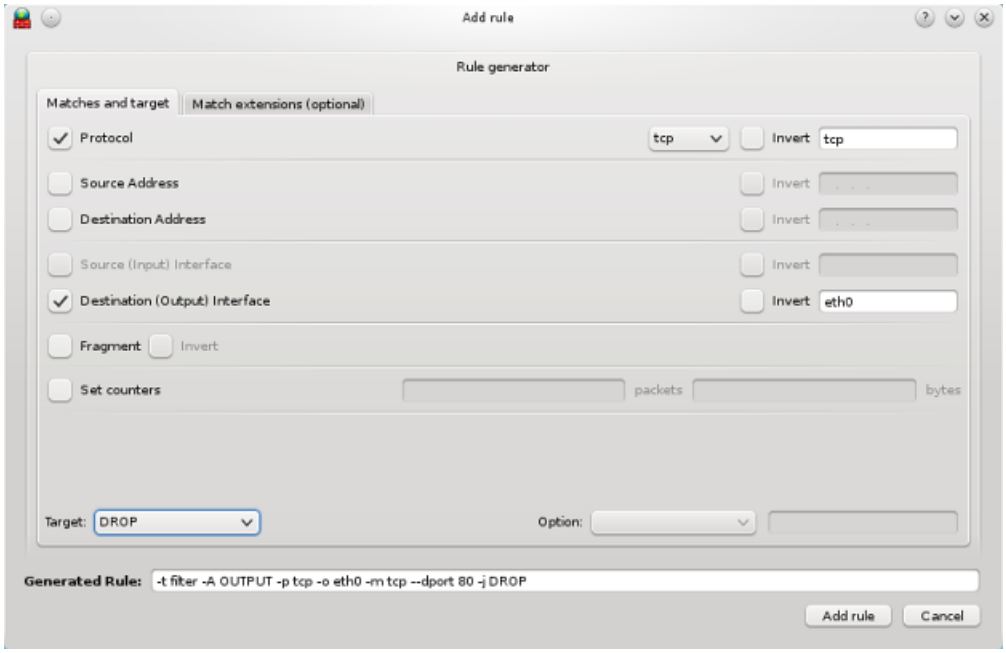


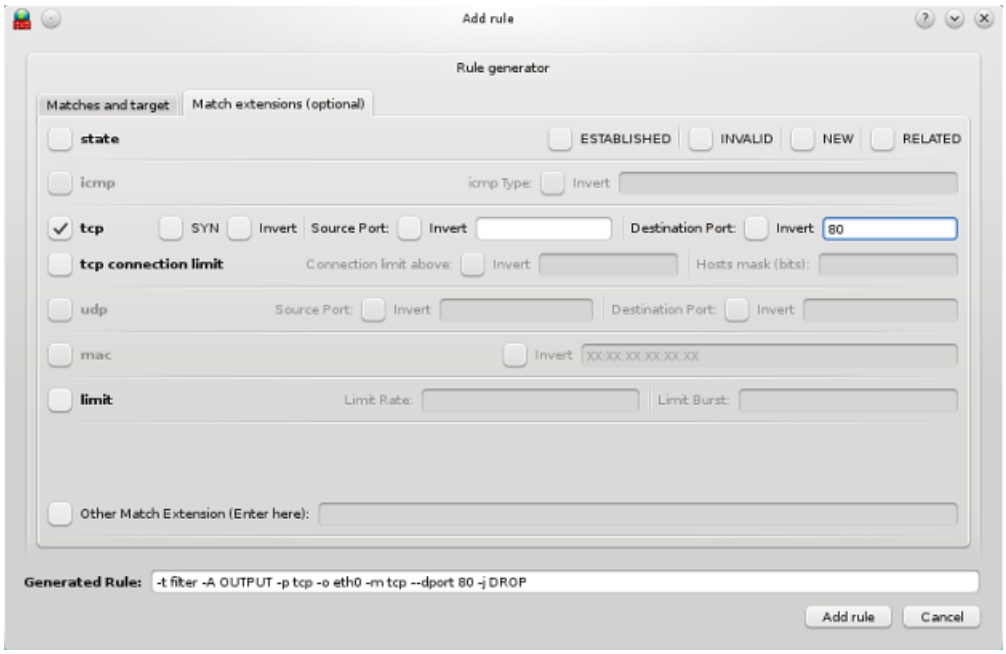

## B.4 Premenovanie užívateľskej reťaze



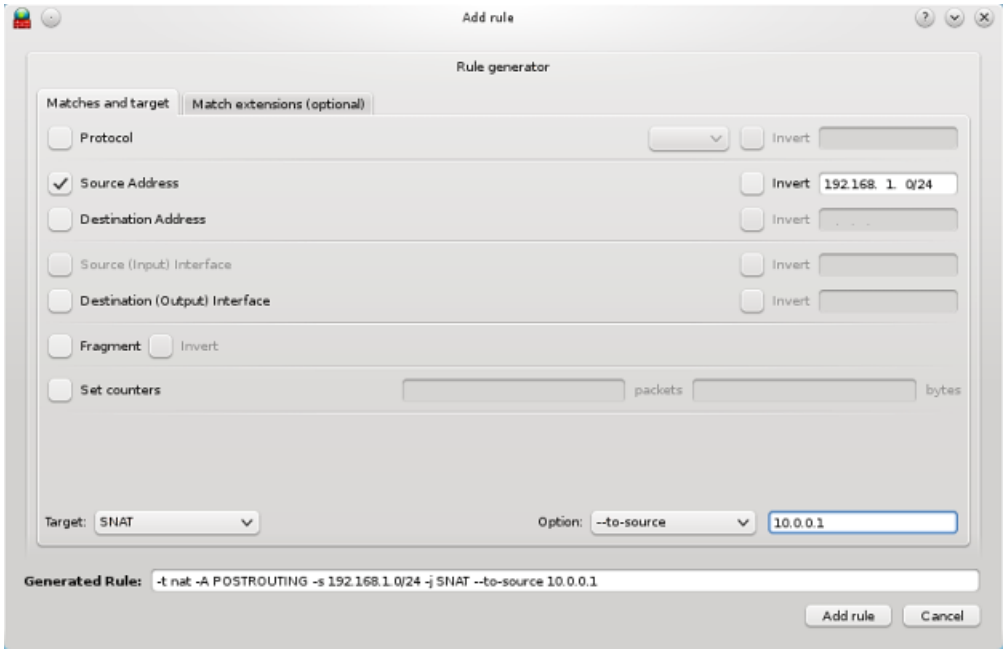

## B.5 Zmazanie užívateľskej reťaze

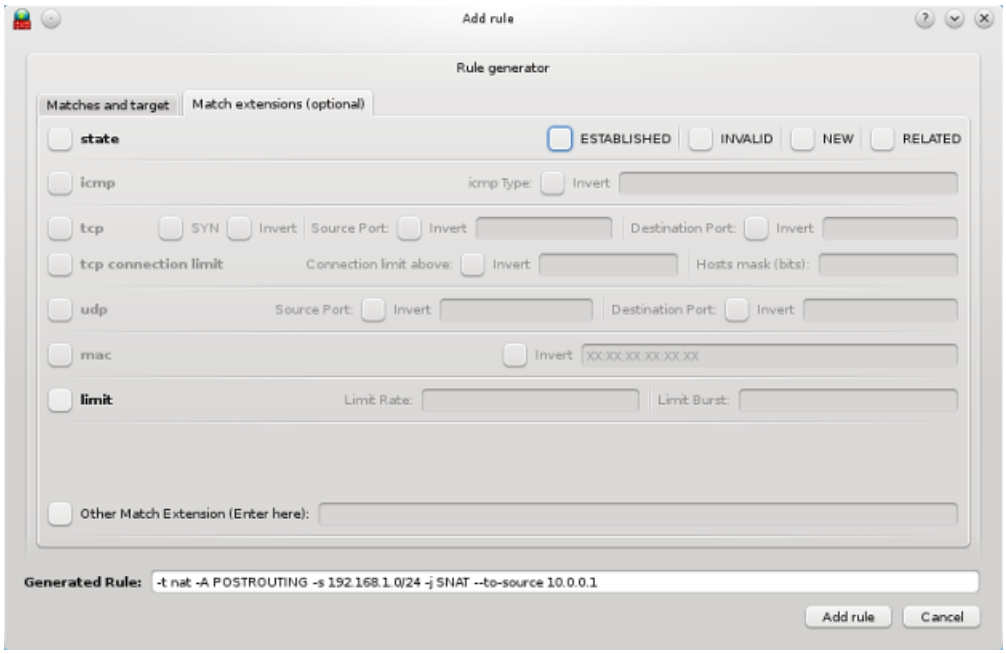
## B.6 Pridanie nového pravidla pre protokol TCP



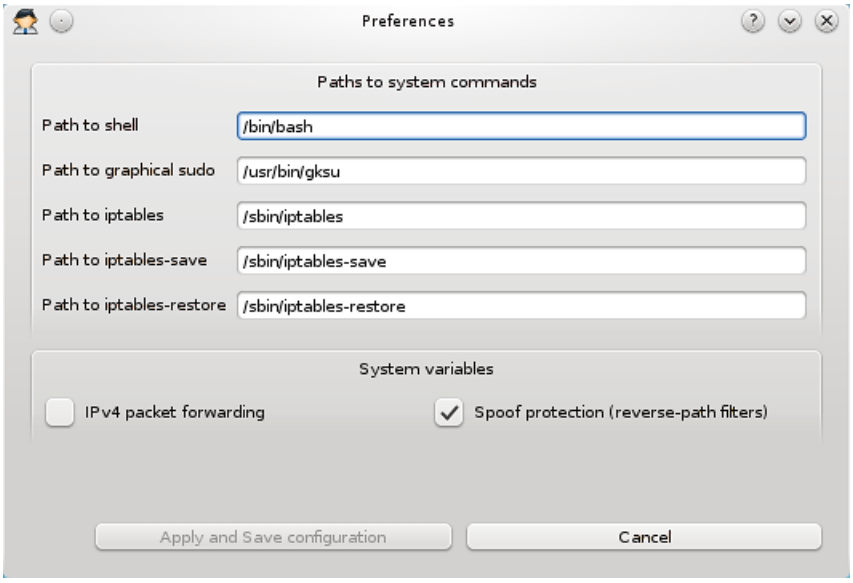



## B.7 Pridanie nového pravidla pre NAT

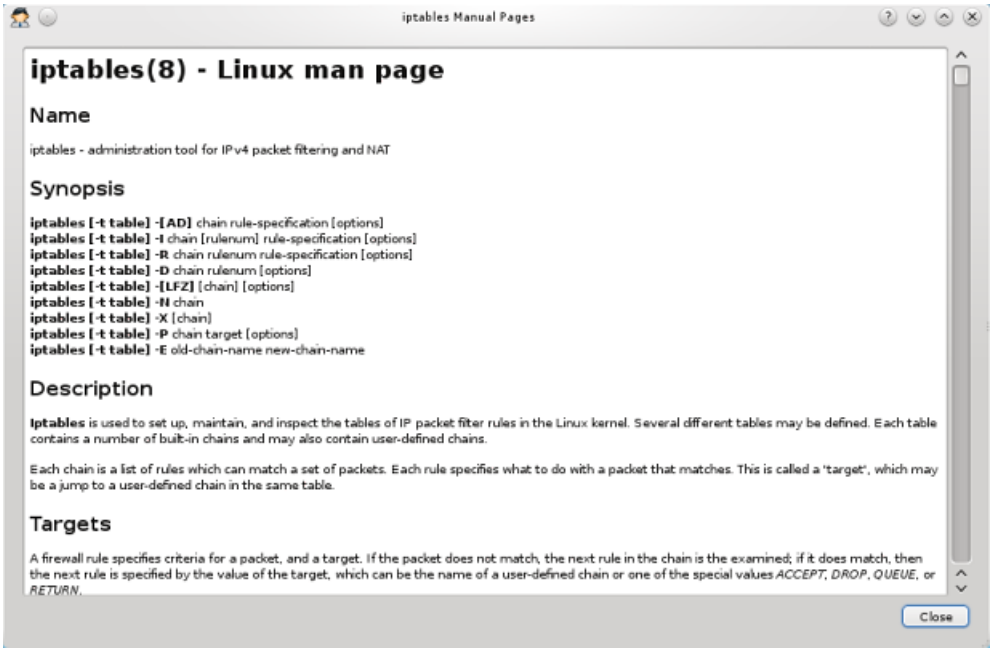


## B.8 Nastavenia aplikácie




## B.9 Manuálové stránky *iptables*



# C UKÁŽKA KONFIGURAČNÉHO SÚBORU

Obsah automaticky vygenerovaného súboru `config.dat` s nastaveniami pre aplikáciu *qIPtables* v domácom adresári používateľa, v podadresári `.qiptables/`:

```
[SHELL]
/bin/bash

[SUDO]
/usr/bin/gksu

[IPTABLES]
/sbin/iptables

[IPTABLES-SAVE]
/sbin/iptables-save

[IPTABLES-RESTORE]
/sbin/iptables-restore

[IP-FORWARDING]
no

[SPOOF-PROTECTION]
no
```

# D AUTOMATICKY GENEROVANÝ SKRIPT

Obsah automaticky vygenerovaného súboru `qiptables-firewall`, ktorý pri štarte operačného systému zapíše do jadra pravidlá firewallu (skrátený a upravený):

```
#!/bin/bash
### BEGIN INIT INFO
# Provides:          qiptables-firewall
# Required-Start:    $network
# Required-Stop:     $network
# Default-Start:     2 3 4 5
# Default-Stop:
# Short-Description: Apply firewall rules at the boot time
# Description:       qIPtables generated firewall startup script
### END INIT INFO

start() {
  trap "" 2 3
  echo "Starting firewall..."
  echo "Setting ip-spoof kernel protection..."
  sysctl -w net.ipv4.conf.default.rp_filter=1 ;
  sysctl -w net.ipv4.conf.all.rp_filter=1 ;

  echo "Setting default policies..."
  iptables -t filter -P INPUT DROP ;
  iptables -t filter -P FORWARD DROP ;
  iptables -t filter -P OUTPUT ACCEPT ;

  echo "Applying rules..."

  iptables -t filter -A INPUT -i lo -j ACCEPT ;
  iptables -t filter -A INPUT -m state \
             --state RELATED,ESTABLISHED -j ACCEPT ;

  echo "Firewall rules applied!"
  trap - 2 3
}
```

```
stop() {
  trap "" 2 3
  echo "Stopping firewall..."
  echo "Deleting all rules and custom chains..."
  iptables -t filter -F ;
  iptables -t filter -X ;

  echo "Setting default policies to ACCEPT..."
  iptables -t filter -P INPUT ACCEPT ;
  iptables -t filter -P FORWARD ACCEPT ;
  iptables -t filter -P OUTPUT ACCEPT ;

  trap - 2 3
}

restart() {
  stop
  start
}

case "$1" in
  start)
start
;;
  stop)
stop
;;
  restart)
restart
;;
  *)
echo $"Usage: $0 {start|stop|restart}"
exit 1
esac

exit $?
```

# E OBSAH DVD

- Elektronická verzia práce v PDF formáte;
- Súbory tejto práce vypracované podľa šablóny pre BP/DP v2.41, z ktorých bol PDF formát vytvorený;
- Zdrojové kódy aplikácie *qIPtables* s českým a slovenským lokalizačným súborom a ikonami;