Grafické rozhraní pro nastavení iptables v GNU/Linux

Graphical interface for iptables setting in GNU/Linux

Bc. Martin Vičánek

Diplomová práce
2007

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav aplikované informatiky
akademický rok: 2006/2007

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Martin VIČÁNEK**
Studijní program:	**N 3902 Inženýrská informatika**
Studijní obor:	**Informační technologie**
Téma práce:	**Grafické rozhraní pro nastavení iptables v GNU/Linux**

Zásady pro vypracování:

1. Studium problematiky firewallů v OS GNU/Linux.
2. Literární rešerše na téma IPTABLES a grafických nadstaveb.
3. Praktická část bude obsahovat vytvoření grafického konfiguračního rozhraní pro IPtables. Konfigurace bude dostupná přes webové rozhranní (dynamické WWW stránky).
4. Ukázka praktického nasazení.

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

Deitel, H. M.: Operating Systems, Prentice Hall, 2004
Tanenbaum, A. S.: Modern operating systems, Prentice Hall, 2002
Linux – Dokumentační projekt, Computer Press, 2003
Sobell, M., G.: Linux–praktický průvodce, ComputerPress, 1999.
Nemeth, E., Snyder, G., Hein, T. R.: Linux – kompletní příručka administrátora. ComputerPress, 2004.

Vedoucí diplomové práce: **Ing. Martin Sysel, Ph.D.**
Ústav aplikované informatiky

Datum zadání diplomové práce: **13. února 2007**

Termín odevzdání diplomové práce: **28. května 2007**

Ve Zlíně dne 13. února 2007

L.S.

prof. Ing. Vladimír Vašek, CSc.
děkan

doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Tato práce se věnuje problematice netfilter/iptables v operačních systémech GNU/Linux – ve zkratce se jedná o nastavení firewallu na daném systému, který může zastávat velmi důležitou funkci, a tudíž je nutno jej co nejlépe chránit. V prvních několika kapitolách bude objasněn princip fungování sítí a systémů na nich závislých, dále pak jednoduchý úvod do problémů bezpečnosti. Na konec budou uvedeny příklady již fungujících aplikací podobných té, která bude vyvíjena v praktické části.

Vývoj a popis této aplikace je popsán v části druhé, kde je tak učiněno pomocí obrázkové dokumentace a návodů pro práci s tímto softwarem. Tento software splňuje všechny požadavky pro nastavení základních pravidel firewallu tak, aby bylo možno dosáhnout co největší bezpečnosti.

Klíčová slova: Linux, GNU/Linux, firewall, netfilter, iptables.

ABSTRACT

This study deals with the netfilter/iptables problems in operating system GNU/Linux - in brief how the setting of firewall on given system is concerned. It is able to serve crucial function and in consequence it is necessary to shield it preferably. In a number of first chapters it will be clarified the principle of network and systems functioning, which are dependent on them. Next there will be further simple introduction to the problems of safety and in the end there will be mentioned examples of already exist similar type applications, which will be developed in practical part.

Development and description of this application is described in the second part, where it is done through the use of pictorial documentation and instructions for the work with this software. This software fulfils all requirements for firewall's setting of basic principles so it would be possible to achieve the highest safety.

Keywords: Linux, GNU/Linux, firewall, netfilter, iptables.

Poděkování:

Chci poděkovat vedoucímu mé diplomové práce Ing. Martinu Syslovi, Ph.D. za provedení problematikou GNU/Linux, iptables, programování PHP a v neposlední řadě též zabezpečením. Za jeho odborné rady a připomínky týkající se dané látky děkuji.

Za jazykovou korekci této práce děkuji paní Ludmile Vičánkové.

Prohlašuji, že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve Zlíně …….

Podpis diplomanta

OBSAH

ÚVOD

Nyní v době snadno dostupného internetu je velké procento uživatelů osobních počítačů připojeno k tomuto celosvětově uznávanému médiu. Takovéto připojení je realizováno prostřednictvím nejrůznějších technologií, které jsou většinou určitým způsobem propojeny mezi sebou a tvoří celek (síť) a ten je nutno co nejlépe chránit a bránit.

Tato práce se zabývá problematikou nastavení takovéto efektivní ochrany pro operační systém GNU/Linux. Důvod, proč byl vybrán zrovna tento operační systém (dále jen OS), je zjevně ten, že se začíná používat v čím dál větším měřítku. Daný OS je nasazen v mnoha sítích jako velmi důležitý prvek a pro funkčnost tohoto celku je mnohdy nezbytný. Proto je nutno, aby byl dobře chráněn on samotný a pokud se používá jako server k oddělení internetu a intranetu, tak také aby chránil tyto dva celky mezi sebou. Prvek pro takovouto ochranu se nazývá firewall.

V OS GNU/Linux je firewall nastavován pomocí utility netfilter/iptables, jež je zapouzdřena v samotném jádru tohoto systému.

V první části této práce bude uvedena problematika síťové komunikace, Linux firewall iptables a jednotlivé typy. Jednotlivé části budou popsány a vysvětlen jejich princip a funkčnost. Protože hlavním cílem je problematika iptables, budou ostatní technologie popsány jen v takovém rozsahu, aby čtenář pochopil jejich princip a nutnost použití dané technologie v daném oboru. Dále budou uvedeny příklady, již vytvořených aplikací pro konfiguraci Linux iptables a jejich posouzení.

Praktická část bude zahrnovat postup vytvoření vlastní aplikace pro ovládání iptables, její podrobné vysvětlení a ukázku funkčnosti. Jelikož aplikace bude programována pro přístup s webového rozhraní, tzn. pomocí běžného prohlížeče, je nutno použít programovací jazyk, který takovou možnost podporuje. Pro danou aplikaci je zvolen jazyk PHP. Popis daného jazyka bude v této práci jen obecný, neboť to není hlavním cílem této práce. Hlavním cílem je ulehčení nastavování a manipulace s utilitou iptables, jež je v OS Limux realizována pomocí příkazového řádku. Diky grafickému rozhraní, které bude vytvořená aplikace obsahovat, se nastavení firewallu pod daným OS značně zjednoduší a práce při konfiguraci se tak podstatně urychlí.

I. TEORETICKÁ ČÁST

1 PROBLEMATIKA KOMUNIKACE A OS GNU/LINUX

V jednoduchém příkladu nastíníme princip pro použití firewallu. Vytvoříme síť o několika počítačových stanicích (takováto síť může mít i stovky klientských stanic), poté tyto stanice budeme chtít připojit do internetové sítě. Nejdříve je nutno vybrat PC který bude zabezpečovat funkci serveru a na tento server nainstalujeme OS LINUX. Operačních systémů tohoto typu je celá řada, k našim účelům můžeme vybrat např. *Slackware*[1]*, Debian, RedHat, CentOS, Suse* a spoustu dalších distribucí. Tento server připojíme pomocí síťových karet do internetu a do vnitřní sítě. K tomu, aby daný PC fungoval, je nutno nastavit IP adresy, směrování a v neposlední řadě též firewall. V OS LINUX to bude pravděpodobně firewall iptables. Později si ukážeme jak tento firewall nakonfigurovat.

1.1 Síťová komunikace

Síť je soubor hostitelů, kteří mezi sebou komunikují. Tato komunikace je možná pomocí speciálního jazyka, který se nazývá *protokol*. Tyto protokoly jsou pravidla, která definují, jakým způsobem se budou dané zprávy mezi hostiteli vyměňovat. Dnes nejpoužívanější protokol pro síťovou komunikaci se nazývá TCP/IP.

1.1.1 Sítě TCP/IP

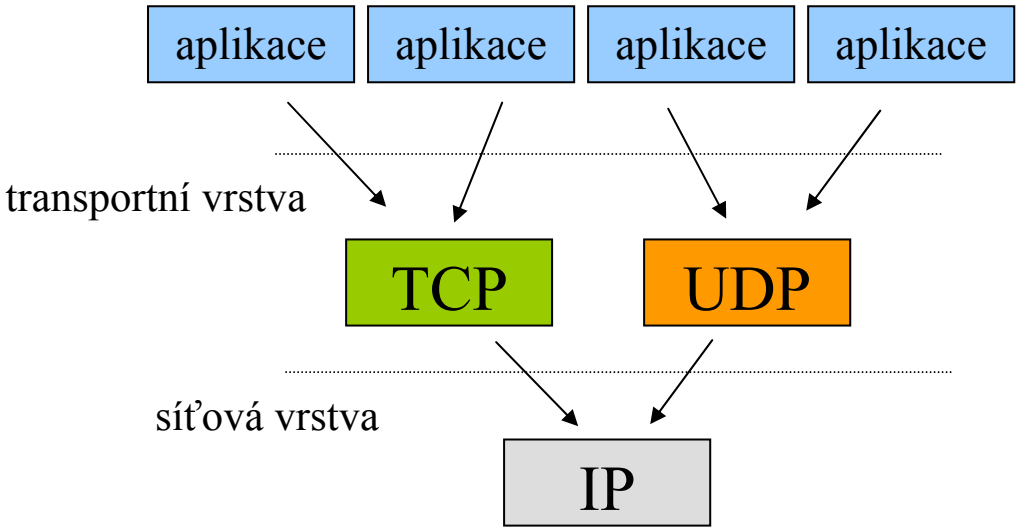
Komunikace v naší síti probíhá pomoci protokolu TCP/IP. Místo přenášení celých souborů dohromady jsou data rozdělena do malých jednotek, tzv. **paketů,** které cílový hostitel po přijetí opět spojí. Pakety mohou být typu:

- **TCP** – jejich doručení je zabezpečeno tzn., že se vždy dostanou k cílovému hostiteli. To zabezpečuje daný protokol
- **UDP –** Tento typ nemá žádné potvrzovací mechanizmy a tím může docházet ke ztrátě paketů.

[1] Distribucí Operačního systému UNIX je spousta a většina jich je volně k dispozici pod licencí GPL. Více informací o všech distribucích je možné najít na adrese URL: http://www.linuxsoft.cz/

- **ICMP** – Informují cílového hostitele o situacích, které mohou nastat. Využívají se tedy pro servisní účely.

Aplikace si mohou samy vybrat, který způsob budou používat, zda TCP nebo UDP.



Obr. 1. Spolehlivá a nespolehlivá komunikace

Každý paket má **hlavičku**, která mimo jiné obsahuje **zdrojovou a cílovou adresu**, tedy unikátní IP adresu počítače, odkud paket pochází a kam směřuje. TCP/UDP pakety dále v hlavičce obsahují **číslo portu**. Cílová adresa nám říká, kam paket směřuje. Port se používá k rozlišení aplikací, které mezi sebou komunikují. Analogicky to funguje i se zdrojovou adresou a portem.

Firewall funguje na principu filtrace těchto paketů pomocí několika aspektů. Nejčastěji to jsou zdrojová adresa, cílová adresa, čísla portů, protokol. Protože paket obsahuje veškeré informace o spojení a přenášených datech, je možno pomocí firewallu filtrovat téměř cokoliv.

Tolik ke zjednodušenému úvodu do počítačových sítí.

1.2 Operační systém GNU/Linux

Linux je jádrem několika počítačových operačních systémů. Je známým příkladem svobodného softwaru a Open source vývoje. Na rozdíl od proprietárních operačních systémů jako Windows či Mac OS, je celý jeho zdrojový kód volně k dispozici pro veřejnost a kdokoli jej může svobodně používat, upravovat a dále distribuovat [6].

Ačkoliv termín *Linux*[2] značí Linuxové jádro, často se používá pro označení celých unixových operačních systémů (známých jako **GNU/Linux**), které sestávají z Linuxového jádra a zároveň z knihoven a nástrojů z projektu GNU, ale i z dalších zdrojů. V nejširším významu GNU/Linuxová distribuce uceleně spojuje základní systém s velkým balíkem aplikačního softwaru, navíc často zajišťuje uživatelsky přívětivou instalaci a následné aktualizace [6].

1.2.1 Části operačního systému

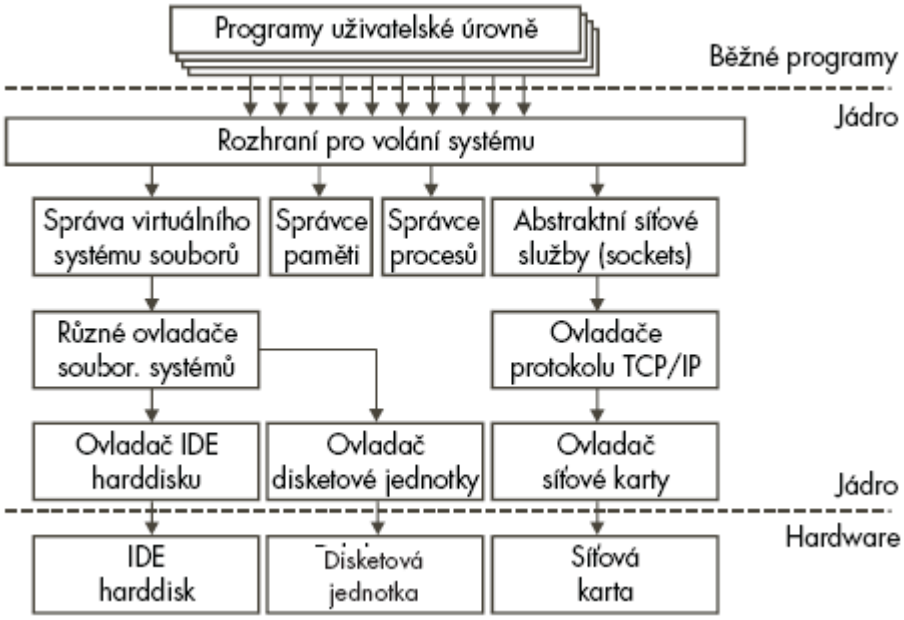
Operační systém GNU/Linux je složen z **jádra systému**[3], **systémových programů** a **aplikačních programů**. Uživatelé pracují především s aplikačními programy. Základním nástrojem je tedy jádro systému, které má za úkol spouštět programy, přidělovat systémové prostředky, spravovat záznamy na disku a také přijímá a odesílá pakety v naší síti. Vlastnost přijmout a odeslat pakety využívá v jádře zkompilovaný netfilter/iptabels, který je předmětem této práce.

Jádro systému se skládá z několika důležitých subsystémů. Které to jsou, a jaká je jejich návaznost ukazuje Obr. 2.

---

[2] Opravdu často se jádro systému chybně ztotožňuje se samotným operačním systémem. Operační systém poskytuje ve srovnání s jádrem mnohem více služeb.

[3] Jádro systému (Linux) začal původně psát finský studen Linus Thorvalds jako svůj koníček.



Obr. 2. Důležité části jádra systému

Operační systém UNIX je oblast velmi obsáhlá a popsat ji vydá na tlustou knihu, jakou je např. Dokumentační projekt, který by si měli vážní zájemci o práci s tímto systémem přečíst.

Výše popsaný text měl pouze přiblížit práci daného operačního systému, kde je patrné, že veškerou komunikaci která prochází přes protokol TCP/IP nám zabezpečuje jádro tohoto systému. Toto jádro má v sobě zapouzdřeno velké množství funkcí, které provádí, ale pro náš účel je důležitá pouze jedna z nich a to ta, která obstarává správu sítě a na ni také navázaného síťového netfilteru. Netfilter firewall tedy slouží pro filtraci paketů, jež prochází přes síťová rozhraní, a to jakýmkoliv směrem. Firewall vezme daný paket a rozhoduje, co s ním udělá. Navíc může změnit v hlavičce datagramu informace. Dále existují **stavové firewally** (vč. iptables). To znamená, že si firewall dokáže navíc pamatovat, k čemu který paket slouží. Později pak dokáže rozhodnout, co s příchozím paketem udělá v závislosti na předešlé komunikaci.

2 GNU/LINUX IPTABLES FIREWALL

Tato část pojednává o teorii IP filtru, co to vlastně je a jakým způsobem zpracovává požadavky a další základní věci.

Firewally obecně jsou většinou specializované systémy tvořící bezpečnostní a monitorovací hranici mezi několika sítěmi. Firewallů je mnoho druhů, některé se aktivně podílí na síťovém provozu, a to tím způsobem, že do něj přímo zasahují. Nejčastěji jsou nasazovány mezi LAN a Internetem, kde chrání tyto dvě vrstvy mezi sebou.

Druhy firewallů:

- Proxy brány: Je to firewall pracující na úrovni aplikační vrstvy. U tohoto typu je zakázán IP forwarding (předávání paketů z jedné sítě do druhé), místo toho je použit zprostředkovatel, který vyřizuje veškerou komunikaci. Pakety tudíž směřují pouze k němu. Například požadavek na www stránku z vnější sítě je poslán pouze proxy bráně, ta jej přijme a poté vygeneruje požadavek do vnější sítě jako svůj vlastní. Proxy brány se specializují na jednu konkrétní službu (www, FTP, email, …). Pracují na aplikační (7) vrstvě modelu ISO/OSI
- Paketové filtry: Pohyb na síti je sledován po jednotlivých paktech, které přes firewall prochází. Pracují na síťové (3) a transportní (4) vrstvě modelu ISO/OSI. Pakety jsou posuzovány podle atributů obsažených v hlavičce, jako je zdrojová/cílová adresa, rozhraní, ze kterého paket přišel, zdrojový/cílový port. Tento typ filtrů je nenáročný na systémové zdroje, a tudíž je i rychlý. Proces posuzování se odehrává v jádře systému.

2.1 IP filtr

IP filtr byl původně navrhnut, aby prováděl různé reakce na požadavky, které byly na třetí vrstvě modelu ISO/OSI (síťové vrstvě). Iptables umí pracovat také se čtvrtou vrstvou ISO/OSI a mnoho dnešních filtrů tuto možnost využívá

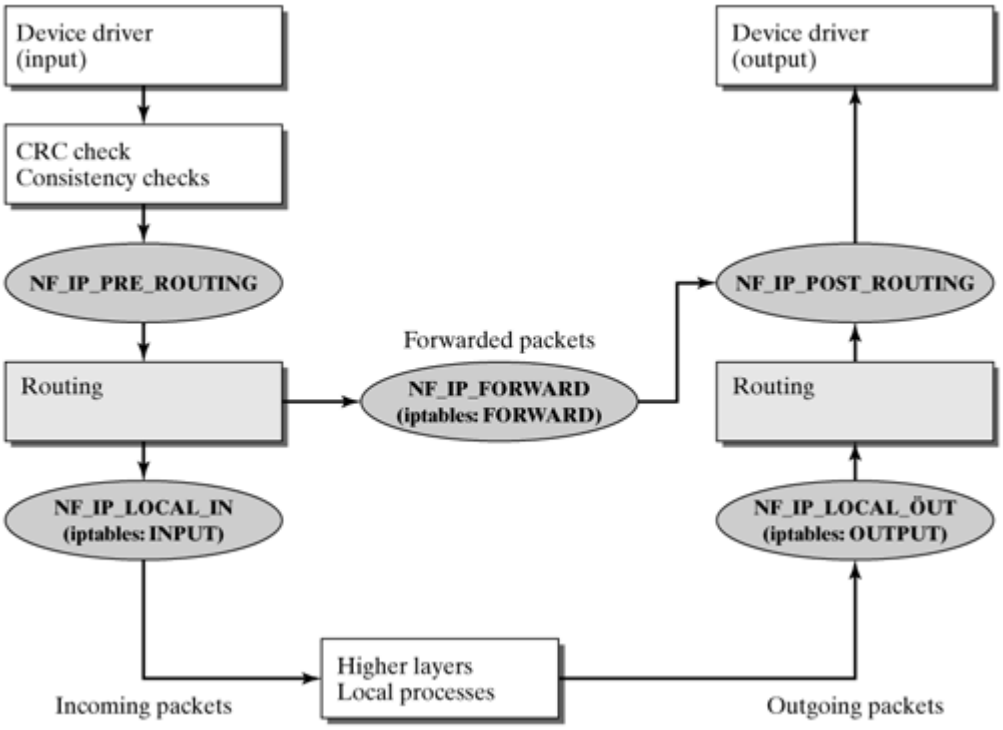
Stavový IP filtr sleduje hlavičky datagramů a v nich parametry, které chceme určitým způsobem filtrovat. V základním návrhu byly sledovány zdrojová a cílová adresa, TOS/DSCP/ECN, TTL, protokol, atd. Iptables umí i hlubší nahlédnutí do paketu a tím rozšiřuje možnosti filtrace, např. podle MAC adresy.

Iptables dokáže sledovat pakety z určitého proudu (zdroje, cíle) tak, jak přicházejí na dané zařízení, a to ve stejném čase – connection tracking. Tato vlastnost umožňuje překlad síťových adres (DNAT, SNAT), která je hojně používaná a sítě by bez ní byly jen velmi těžko realizovatelné, a to z důvodu omezeného počtu síťových adres.

Iptables má i své nevýhody. Nedokáže spojit data ve dvou paketech dohromady, a tudíž hrozí nebezpečí, že nenalezne řetězec, který je rozdělen ve více paketech. Tato možnost není ošetřena, neboť by vzrostly mnohonásobně nároky na výpočetní výkon a paměť daného stroje. Pro ošetření této možnosti je nutné použít Proxy server. Také je vyvíjen nový L7-filter, který je stále ještě ve fázi testování a vývoje.

Filtrování je možno provádět v několika fázích:

- Po přijetí datagramu ze sítě
- Během přesměrování datagramu)
- Před odesláním datagramu



Obr. 3. Architektura filtrování paketů (netfilter)

Na Obr. 3. je ukázáno několik záchytných bodů, které slouží pro definování pravidel v netfilteru. Vyobrazených pět bodů (hook - háků) je definováno ve specifickém

hlavičkovém souboru *Linuxu/netfilter_ipv4.h.* Firewall rozhoduje o osudu paketů na základě **pravidel**, která jsou sekvenčně uspořádaná v **řetězcích**. Iptables má prázdné řetězce pro každý směr.

- `NF_IP_PRE_ROUTING`: Pakety přicházející na firewall prochází tímto bodem dříve, než jsou zpracovány směrovacím kódem. Zde jsou zachycovány pakety, než se dostanou ke zpracování. Tento bod používá mechanismus NAT. Pro definici pravidel je použit řetězec PREROUTING
- `NF_IP_LOCAL_IN`: Na tento bod přicházejí ty pakety, které jsou adresovány místnímu stroji. V iptables jsou pravidla definovaná pomocí řetězce INPUT.
- `NF_IP_FORWARD`: Zde projdou všechny pakety, které nejsou směrovány do místního počítače a opouštějí počítač přes jiné síťové rozhraní. V iptables je definován řetězcem FORWARD.
- `NF_IP_LOCAL_OUT`: Tímto bodem odcházejí všechny pakety, které byly vytvořeny místním počítačem. OUTPUT se nazývá řetězec pro definici ochozích pravidel.
- `NF_IP_POST_ROUTING`: Zde je poslední možnost změny nebo zrušení paketu. Tento bod se používá pro monitorování toku dat a aplikování účtujících metod. Definování pravidel je možno v řetězci POSTROUTING.

Ipfiltr tedy hlídá všechny platné spojení a jejich porty, které ukládá do tabulky a to za pomocí těchto záchytných bodů, přes které paket prochází. Pokud se na daném bodě vyskytne paket, který odpovídá pravidlu na něm definovaném, uplatní se na něj politika, jež dané pravidlo určuje.

2.2 Podmínky a výrazy IP filtru

Všeobecné podmínky, kterým je nutno porozumět před uvedením možnosti nastavení.

- **Drop/Deny** : Paket je zamítnut neboli zrušen. Odesílatel není nijak informován o jeho zrušení. Paket je prostě z komunikace vyloučen a už se dále nikde nevyskytuje.
- **Reject** : Tato možnost je téměř stejná jako předchozí. Rozdíl je v tom, že zde je poslána informace o zrušení paketu zpět jeho hostiteli. Odpověď může být automatická nebo specifikovaná.

- **State**: Specifický stav balíčku, např. Syn paket v TCP spojení.
- **Chain**: Řetězce, které obsahují nastavení pravidel, jež jsou aplikovaná na paketech. Každý řetězec má svůj specifický význam.
- **Table**: Každá z tabulek má svůj význam. Existuji čtyři základní: raw, nat, mangle, filter. Např. nat tabulka je určena k tomu, aby překládala síťové adresy a filtr tabulka pro filtrování paketů.
- **Match**: Jedná se o rozšíření definovaných pravidel, kdy nestačí určit pouze IP část, ale je nutné je filtrovat podrobněji. Je možné definovat *match extensions* a *match target.*
- **Target:** Definuje cíl každého pravidla, tzn., jestli je pravidlo přijato, zamítnuto, atd. Říká tedy, co se bude dít s paketem, pokud pravidlo vyhovělo.
- **Jump:** Podobné jako Target, rozdíl je jen v tom, že paket je předán na další řetězec, na kterém je znovu posuzován.
- **Accept** : Paket, kterému vyhovuje dané pravidlo, je puštěn dále ke zpracování na dalších pravidlech.

Základem každého pravidla je jedna nebo více podmínek, které umožňují identifikovat, a tím pádem vybrat požadovaný paket. Pokud paket vyhovuje daným podmínkám, vykoná se akce, která je součástí definice pravidla. Není-li podmínka splněna, pokračuje se ve vyhodnocování dalších pravidel. Pokud se nesplní ani jedno pravidlo, je aplikována defaultní politika (default policy).

2.2.1 Tabulky iptables

Iptables ukládá řetězce v *tabulkách*. Ty jsou následujícího typu: ***raw, filter, nat a mangle***. Bez uvedení typu tabulky se defaultně používá filter. Zde jsou uvedeny řetězce input, forward a output. V nat tabulce jsou řetězce prerouting, postrouting a forward.

2.2.1.1 NAT tabulka

Jedná se o jednu ze základních tabulek a slouží pro překlad síťových adres. Jedna z důvodů, proč se tato možnost masově rozšířila, je nízká cena realizace oproti použití např. Cisco PIX firewall atd.

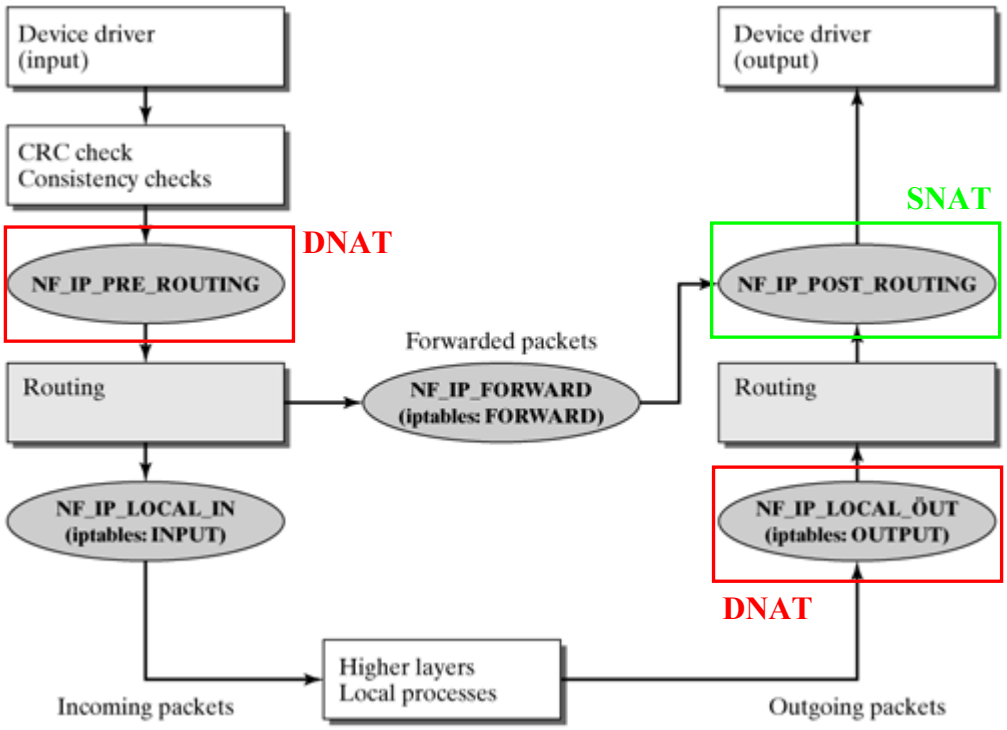
Díky nat tabulce je možné:

- Zvýšit bezpečnost vnitřní sítě jejím skrytím
- Převést jednu adresu veřejnou na neveřejné tak, aby bylo možno přistupovat i z vnitřních PC do vnější sítě.

Nat server tedy přijímá paket a přepisuje zdrojovou nebo cílovou adresu. Po této změně vytvoří nový kontrolní součet paketu. Adresa je změněna podle určitého vzoru, který definuje dané pravidlo. Překládáme li zdrojovou adresu, jedná se o SNAT (Source Network Address Translation). SNAT je tedy nejvíce používaný. Překládá všechny neveřejné adresy na jednu adresu veřejnou.

Aby bylo možno dostat se na daný stroj zvenku, je možno použít DNAT (Destination Network Address Translation), kdy je přeložena cílová (veřejná) adresa na adresu neveřejnou.

Kde jsou jednotlivé operace prováděny, je ukázáno na Obr. 4.



Obr. 4. Průchod datagramu nat tabulkou

Případ, kdy se DNAT používá na OUTPUT je ten, kdy „natované“ pakety vznikají na lokálním PC. V praxi je to jedna z nejméně používaných možností.

2.2.1.2 RAW tabulka

Používá se k označení paketů, které nemají být sledovány pomocí connection tracking. Na paketu je pomocí pravidla nastavena hodnota NOTRACK, díky čemuž nebude sledováno spojení daného paketu.

Tabulka používá pouze řetězy PREROUTNIG a OUTPUT. V jiných místech to není možné, neboť už by byly zahrnuty do sledování.

Pro použití této tabulky je nutno nahrát přídavný modul do iptables.

2.2.1.3 Filter tabulka

Jak již název říká, používá se tato tabulka k filtrování paketů. Jedná se o nejpoužívanější tabulku, a také je nastavena jako výchozí, pokud nebyla určena tabulka jiná. Filtrování je možné provádět na paketu procházejícím systémem v jakékoliv jeho části.

Přicházející pakety jsou podle definovaných pravidel sledovány a následně je na ně aplikováno ACCEPT nebo DROP.

Jedná se o základní navrženou tabulku, která byla určena pro veškeré filtrování. Postupem času samozřejmě přibyly další. Pořád ale platí, že hlavní filtrovací pravidla by měla být prováděna zde.

2.2.1.4 Mangle tabulka

Používá se k označování paketů pro jejich další použití.

Zde se používají následující úkoly:

- TOS : (typ služby). Mění typ požadovaných služeb v paketu. Používá se, má-li být změněn typ služby paketu, který se může určitým způsobem vymykat chování na síti.
- TTL : (doba života). Používá se ke změně hodnoty života paketu. Jedno z použití je např., sjednocení délky života paketů.

- MARK: Vytváří značky, které jsou rozpoznány programy iproute2, a ty můžou s paktem dělat nejrůznější úpravy a sledování. Např. omezení šířky pásma pomocí CBQ.

Př. Pomocí Mangle je možno označit pakety značkou P2P. Poté je možno tyto pakety zachytávat a dávat jim rozdílnou prioritu a šířku pásma (které má samozřejmě každá síť omezeně).

2.3 Základní nastavení iptables

Nastavení je prováděno pomocí série řetězců a přepínačů, které nám dané pravidlo definují. Celé takto vzniklé pravidlo má následující syntaxi:

```
iptables [tabulka] [akce] [chain] [ipčást] [match] [target] [target_info]
```

Tabulka

O nich již byla řeč výše a základní z nich zde byly představeny. Typ tabulky je definován v příkazu přepínačem *„-t“*. Pokud není nastavena žádná tabulka, jak již bylo řečeno, tak bude použita výchozí (tedy filter).

Akce

Co všechno se dá s pravidlem dělat, je uvedeno v manuálu k iptables. Zde jsou popsány jen některé základní možnosti:

-A	--append	- přidá nové pravidlo na konec řetězce
-D	--delete	- smaže pravidlo
-R	--replace	- nahradí číslo pravidla jiným pravidlem
-I	--insert	- vloží nové pravidlo na začátek řetězce
-L	--list	- vypsání všech pravidel v řetězci
-F	--flush	- vyprázdní všechna pravidla
-N	--new-chain	- vytvoření vlastního řetězce
-X	--delete-chain	- smazání vlastního řetězce
-P	--policy	- výchozí politiky řetězce
-E	--rename-chain	- přejmenování vlastního řetězce

Polici – výchozí politika je definována pro každou tabulku, tzn. co se má udělat s paketem, pokud nevyhovuje žádnému z pravidel. Defaultně je tato politika nastavena na ACCEPT – povolí tudíž všechny pakety, které nejsou nikde jinde definovány.

Při konstrukci firewallu je ale nutno řídit se pravidlem „co výslovně není povoleno, je zakázáno“. Pro nastavení výchozí politiky je použit ovladač *„-P“*, jak již bylo zmíněno, nastavení může být DROP nebo ACCEPT .

Pro zahazování všech nespecifikovaných paketů bude tedy nastavení

```
iptables –P INPUT DROP
```

IP část

V této části je možno určit rozhraní, IP adresy, protokoly.

Má-li být zajištěno, aby dané pravidlo bylo aplikováno pouze na pakety přicházející na určité síťové rozhraní, musí být použit přepínač *„-i“ (--in-interface)* nebo analogicky pro odcházející *„-o“ (--out-interface)*. Tato možnost nefunguje, jsou-li síťová rozhraní v mostu (bridge).

Další věcí je možnost definice protokolu, kterým je paket posílán. Standardně jsou to protokoly ICMP, UDP, TCP. Je možné definovat všechny protokoly, které jsou uvedeny v */etc/protocols.* Typ protokolu se určuje pomocí přepínače *„-p“ (protocol).*

Pro určení zdrojové IP adresy se používá přepínač *„-s“ (--src, --source)* a cílové *„–d“ (--dst, --destination)*. IP adresa a maska se zapisuje ve tvaru 192.168.1.254/24 nebo 192.168.1.254/255.255.255.0.

```
iptables -A INPUT -i eth1 -s 192.168.0.0/16 -j ACCEPT
```

Jump

Určuje akci, která se provede, pokud dané pravidlo splňuje všechny podmínky. Přepínač je *„-j“ (--jump)*. Základní akce jsou DROP, ACCEPT nebo REJECT.

Místo *jump* je možno použít GOTO *„-g“ (--goto)*. Tato volba je určena, k použití pravidla v jiných řetězcích.

Příklad nastavení:

Všechny pakety, které přijdou na rozhraní eth1 z jiného rozsahu než 192.168.0.0/16, budou zahozeny.

- Filtrování bude probíhat v tabulce *filter*, tudíž tuto tabulku není nutné definovat.
- Výchozí pravidlo bude DROP – všechno zahodit.
- Všechny adresy s daného rozsahu budou povoleny.

```
iptables –P INTUP DROP
iptables -A INPUT -i eth1 –s 192.168.0.0/16 -j ACCEPT
```

Pokud by byla výchozí politika ACCEPT je možno zápis změnit následovně:

```
iptables –P INTUP ACCEPT
iptables -A INPUT -i eth1 –s ! 192.168.0.0/16 -j DROP
```

Vykřičník značí negaci daného pravidla.

2.3.1 Match a target

Jedná se o rozšíření stávajících pravidel. Vždy totiž nestačí použít filtrování pomocí *IP adres* a *interface*. Pokud je tedy potřeba definovat pravidlo podrobněji, použije se přepínač *„-m“*.

Některé možnosti jsou následující:

- --sport (--source-port) – zdrojový port který má být filtrován.
- --dport (--destination-port) – cílový port.
- -m multiport `--destination-ports`, `--source-ports`, `--ports` – možnost uvedení více portů oddělených čárkou.
- `-m iprange --src-range`, `--dst-range` – rozsah adres

- `-m mac --mac-source -` definuje MAC adresu
- `-m limit --limit 1/s --limit-burst 100 -` určuje kolik paketů může přijít na daný stroj nebo jím projít. *`--limit`* určuje čas 1/s m h d, *`--limit-burst`* určuje počet paketů.
- `-m state --state -` Určuje stav spojení.
  - NEW – nové spojení
  - INVALID – chybné spojení – možnost tyto chybné pakety filtrovat
  - RELATED – možnost navázané komunikace na jiných portech i přesto, že jsou zakázány.
  - ESTABLISHED – komunikaci si řídí propojené strany.

Možností je samozřejmě spousta, zde byly uvedeny jen ty nejzákladnější.

Pomocí *target extensions* je možné provádět další operace jako je NAT, MARK, TOS, TTL a další modifikace paketů.

2.4 Grafické nadstavby pro netfilter/iptables

Aby bylo možné pracovat rychleji, přehledněji a efektivněji, bylo nutno přijít s grafickým uživatelským rozhraním pro netfilter. Jelikož má netfilter velmi flexibilní nastavení a je vysoce komplexní, není tato úloha nikterak jednoduchá. Několik osob a organizací se snažilo vytvořit grafické rozhraní, ale mnoho neuspělo. Těch co uspělo, není příliš a opravdu kvalitních nadstaveb je jen hrstka.

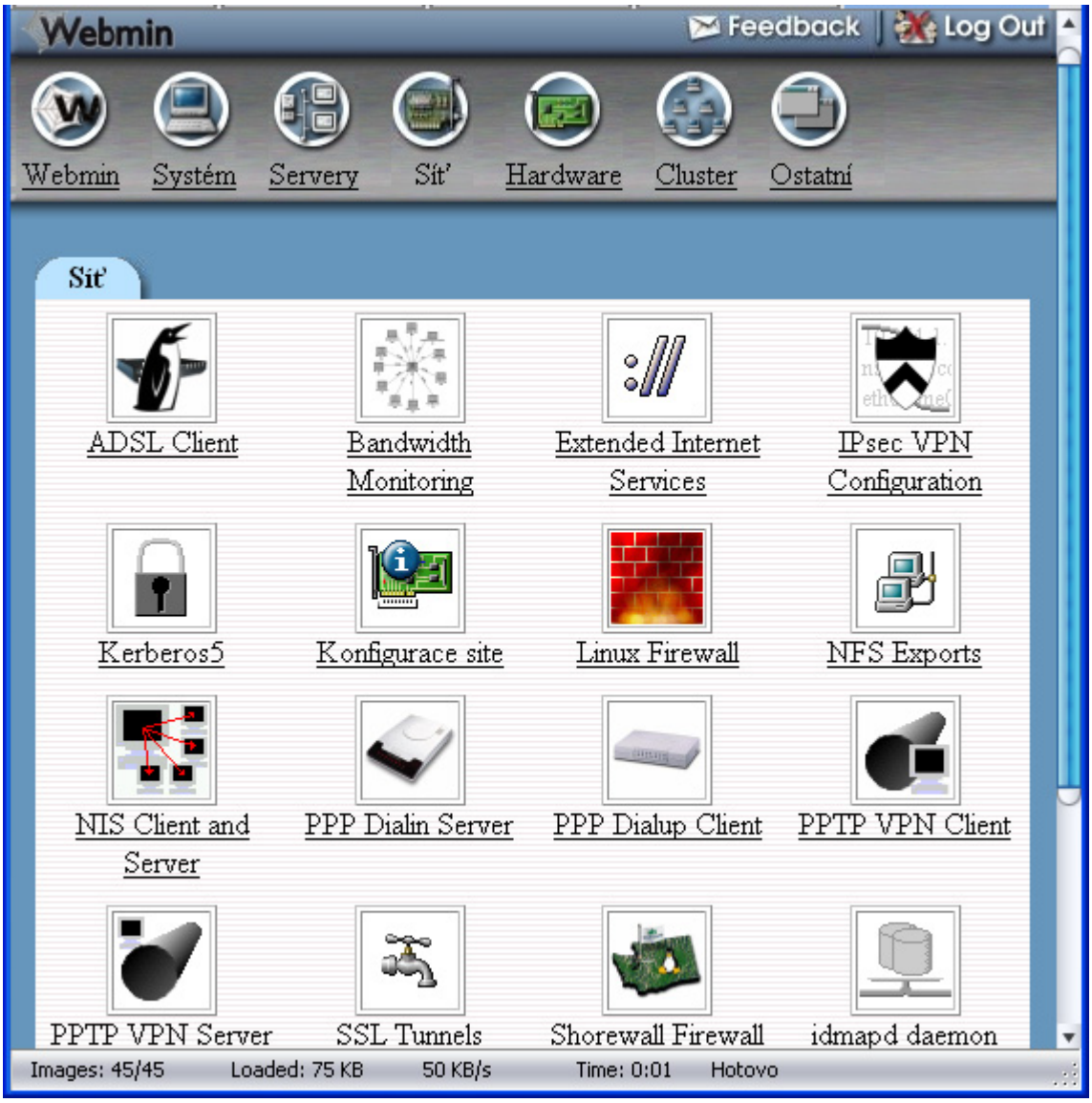
2.4.1 Webmin

Obecně se jedená o jeden z nejlepších nástrojů pro natavení firewall, a ne jen jeho. Toto grafické uživatelské rozhraní umožňuje kompletní správu operačního systému, a to přes webové rozhraní – služba http nebo https. Je možno nastavit služby jako např. DNS, sdílení souborů, uživatelské účty, webový server apache a spoustu dalších.

Instalace probíhá pomocí již předkompilovaných balíčků, kterých je velké množství. Jsou připraveny na všechny více používané operační systémy Unix, BSD nebo GNU/Linux. Webmin obsahuje jednoduchý webový server a CGI programy napsané v PERL 5.

Samotné nastavení probíhá v grafickém rozhraní, kde je nutno si vybrat, co konkrétně chceme konfigurovat. I menší chyba může znamenat velké ztráty, tudíž je nutno zde pracovat důsledně.

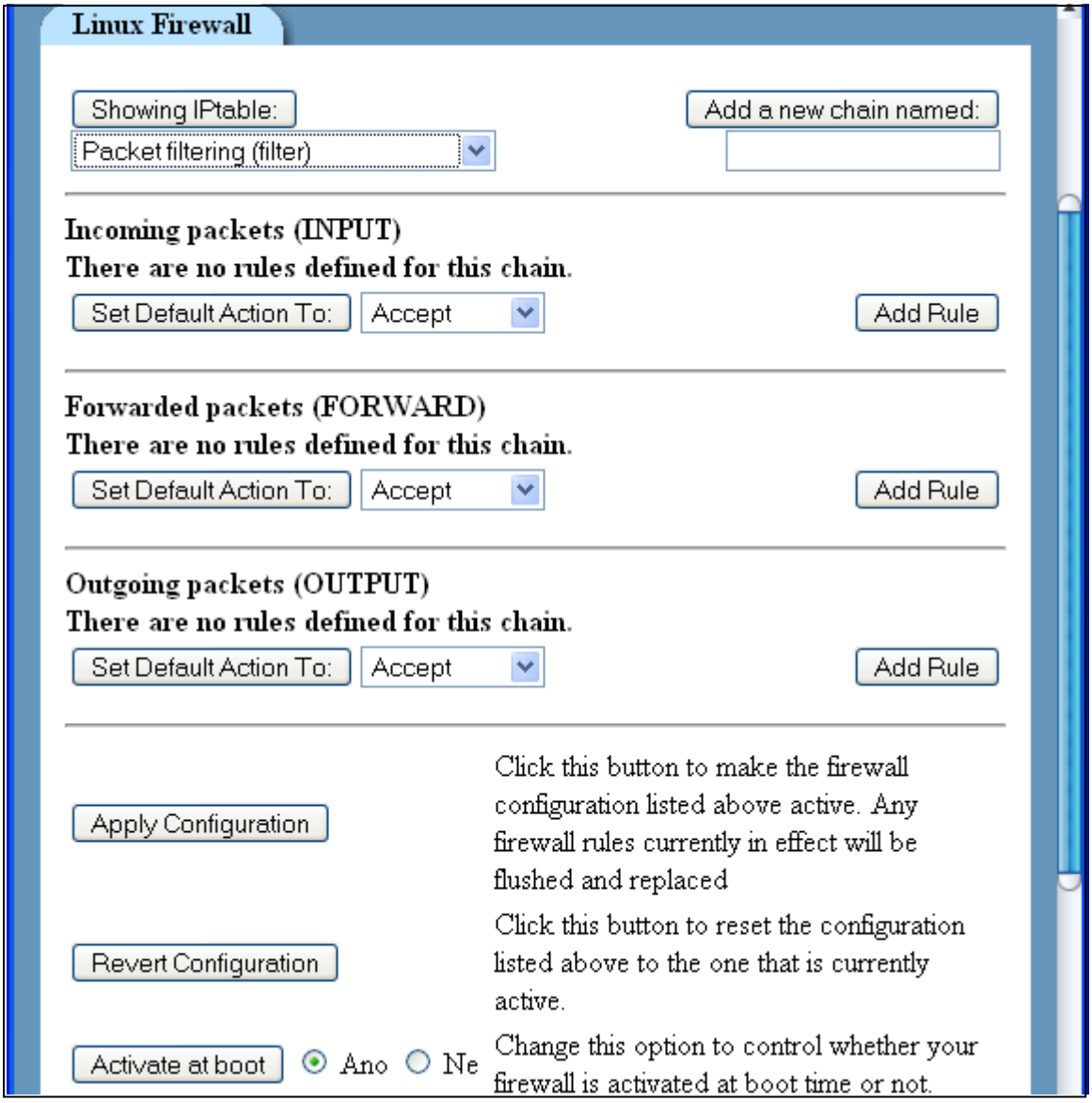
Pro nastavení iptables je nutno zvolit záložku *síť* a zde *Linux Firewall* (Obr. 5.). Jak je vidět na obrázku, je zde mnoho dalších možností ke konfiguraci sítě na našem stroji. Pro naše účely je důležitá pouze ikona Linux Firewall, je zde vidět také *Shorewall Firewall*, ale ta slouží k nastavení jiné nadstavby netfilteru.



Obr. 5. Webmin – rozložení ikon v záložce síť

Po rozkliknutí požadované ikony se zobrazí možnosti nastavení iptables.

Na Obr. 6. je ukázána možnost nastavení v tabulce filter (v rozbalovacím menu je možnost vybrat si ostatní tabulky nat a mangle). Po výběru částí pro umístění našeho pravidla, je nutno vybrat tlačítko *add rule* (přidat pravidlo). Ukáže se formulář s výběrem výše zmíněných možností filtrování. Po vyplnění formuláře je vše nutno potvrdit a pravidlo uložit. Tento postup se opakuje, dokud nejsou přidána všechna potřebná pravidla. Volbou *Apply Configuration* se pravidla aplikují do systému a začínají pracovat na blokování, resp. povolování námi požadovaných paketů.



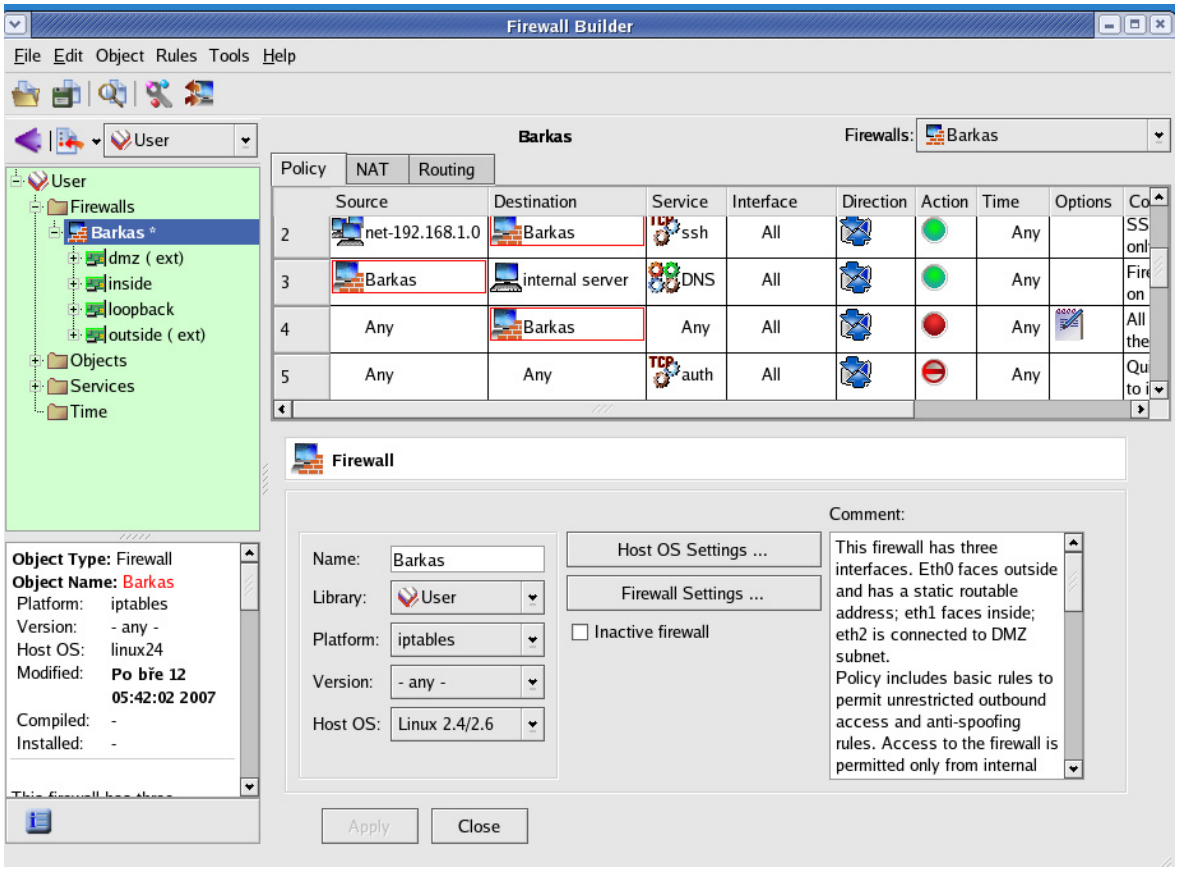
Obr. 6. Webmin – ukázka tabulky filter

2.4.2 Firewall Builder

Jedná se o multi-platformní nástroj pro konfiguraci firewallu. Je zde možno definovat různé typy firewallů a spravovat různorodé systémy, např. Cisco PIX, OpenBSD PF a také iptables.

Nastavení je prováděno elegantně pomocí drag & drop v GUI rozhraní, kdy je možné objekty přetahovat myší, a tím nastavovat celou politiku firewallu. Užitečná se též jeví možnost migrace nastavení firewallu mezi již zmíněnými platformami.

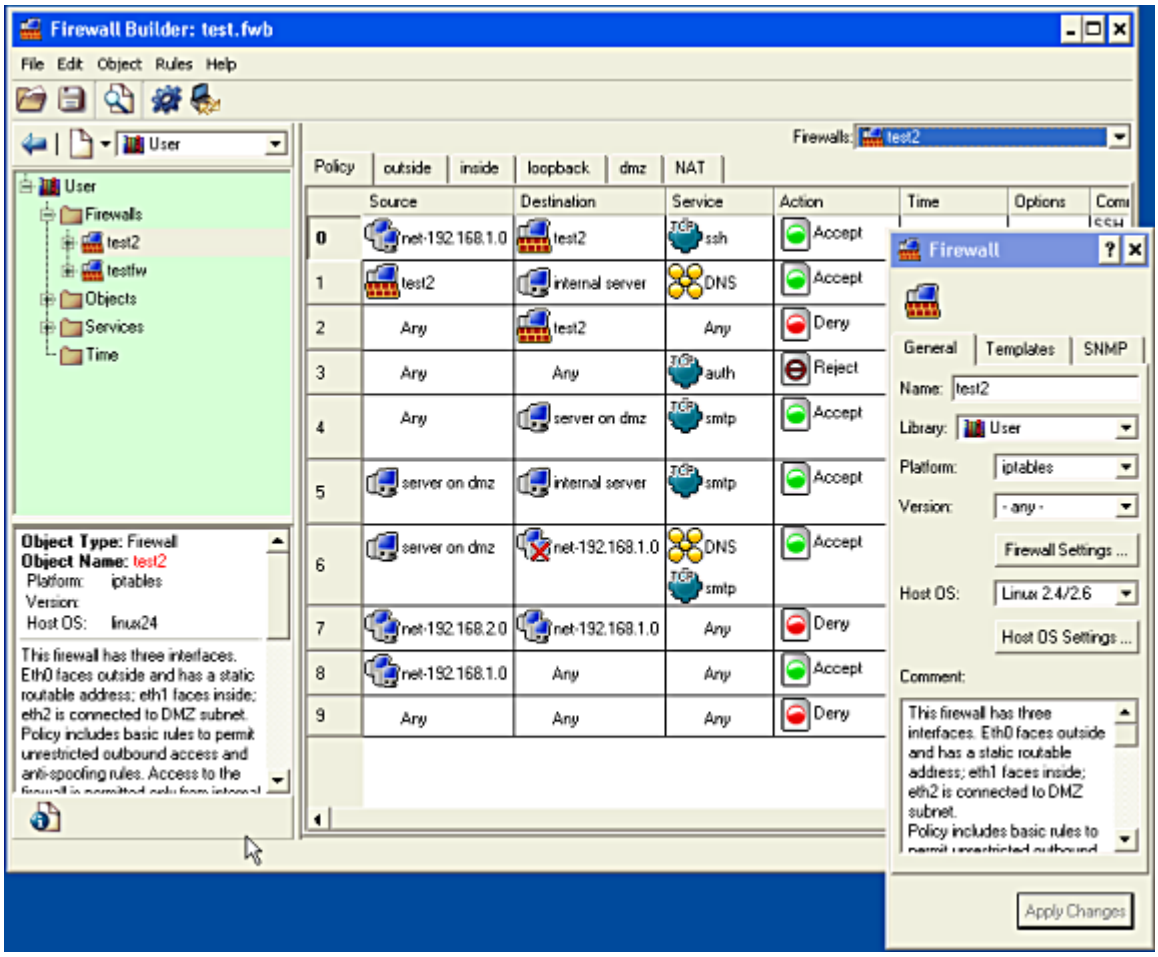
Nástroj je konstruován pro řízení více systémů pomocí síťové objektové databáze, kdy zásah na jednom „centrálním“ stroji se odráží na strojích ostatních.



Obr. 7. Automaticky vygenerovaný firewall pomocí Fwbuilder.

Po jednoduché instalaci z balíčku rpm, byl vytvořen firewall pomocí daného grafického rozhraní. K nastavení a automatickému vygenerování stačilo pouze zadat jméno firewallu, hostujícího stroje a platformu filtru (v našem případě) iptables. Po zadání těchto údajů je možno vytvořit zabezpečení manuálně nebo použít některou z předpřipravených možností.

Na rozdíl od předchozího nástroje zde není možnost přístupu přes webové rozhraní. Vše se tudíž musí konfigurovat buď na stanici, vzdáleně pomocí SSH nebo pomocí vzdálené plochy. V případě konfigurace přes SSH je výstup pouze textový, tudíž GUI zde ztrácí na své váze.



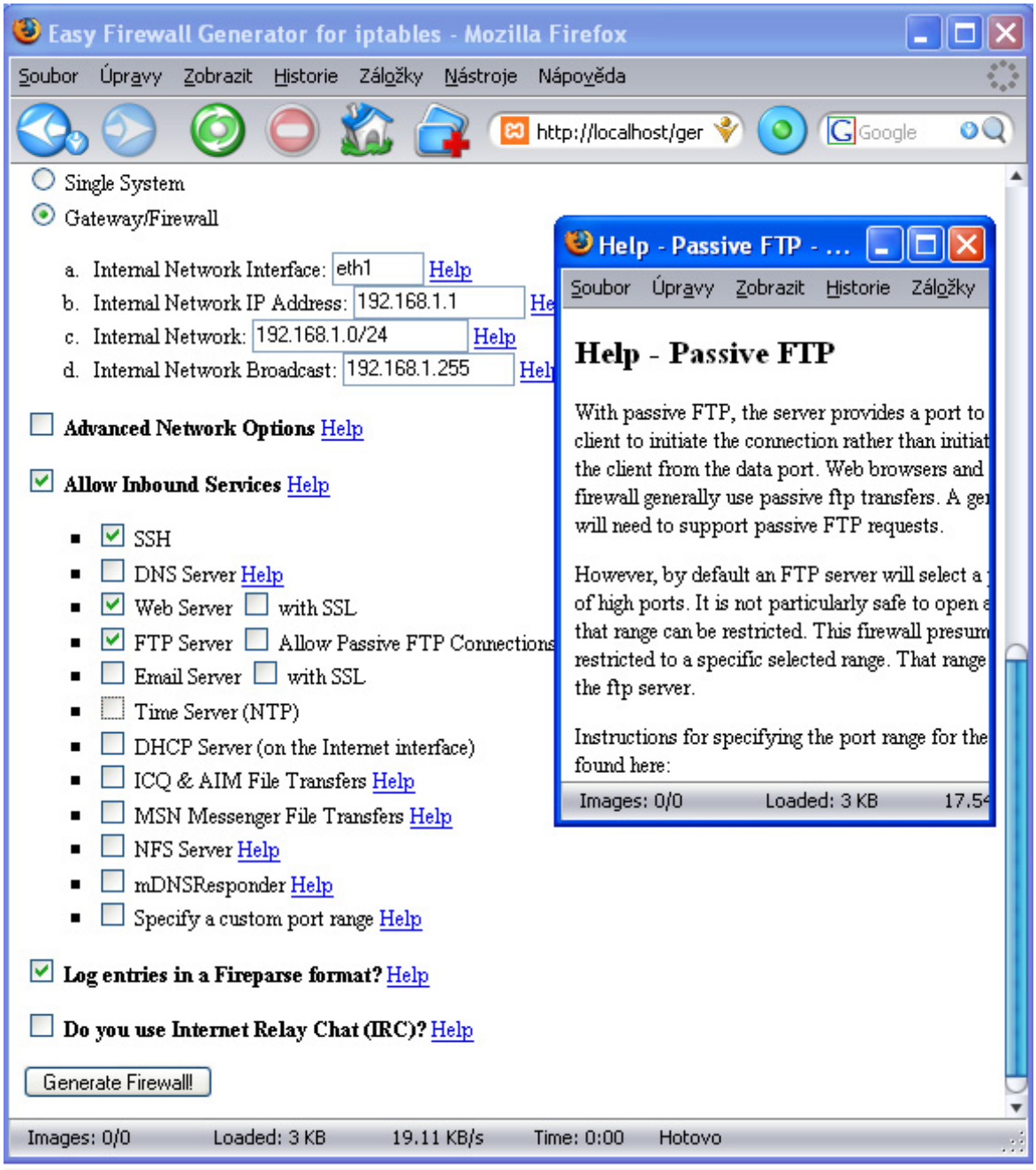
Obr. 8. Firewall vytvořený manuálně pomoci Fwbuilderu

Nastavení firewall může být uloženo pomocí skriptu spustitelného na daném systému nebo v XML souboru.

2.4.3 Easy Firewall Generátor

Program, který funguje rozdílným způsobem než předcházející. Jedná se o skript napsaný v jazyce PHP. Jako vstup je použit formulář, do kterého je nutno zadat všechny údaje pro konfiguraci firewallu. Následně je vygenerován skript pro inicializaci v iptables.

Instalace programu není nutná, stačí ho jen spustit ve webovém prohlížeči. Je zde možnost přečíst si krátký tutoriál o funkci daného programu.



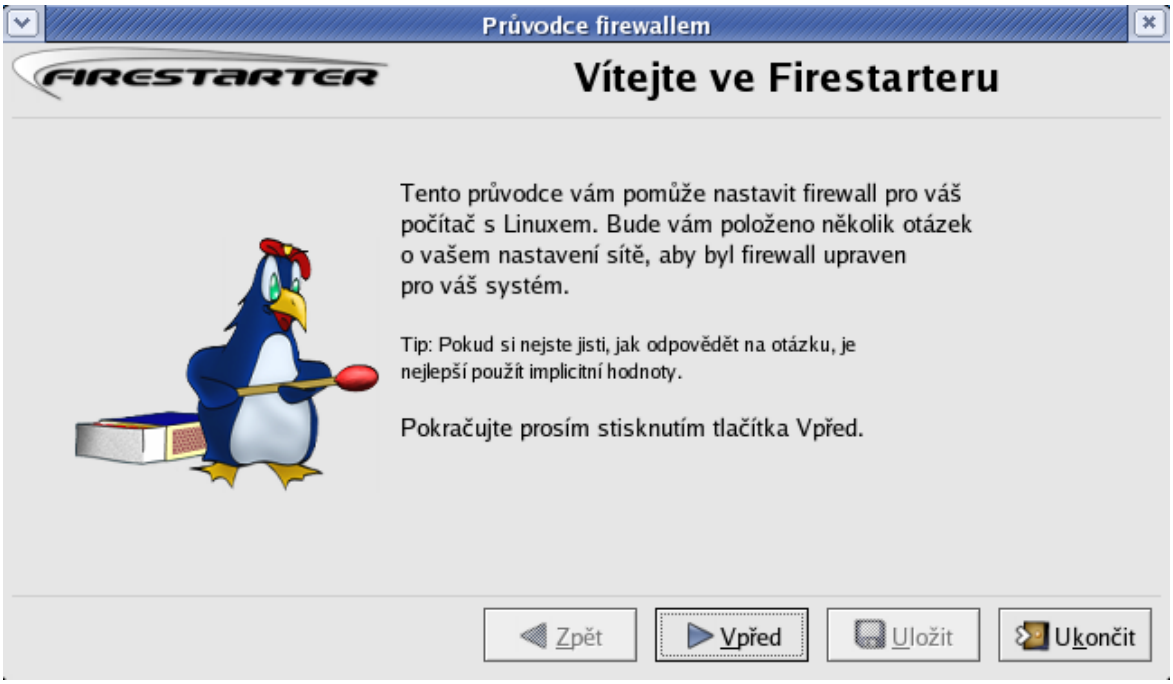
Obr. 9. Easy Firewall Generátor

U každé možnosti je popsáno k čemu slouží, což dělá program přehledným. Nastavení zde není příliš mnoho a jedná se jen o základní možnosti.

Pro vygenerování skriptu slouží tlačítko „Generate Firewall“.

2.4.4 Firestarter

Hned po spuštění programu se objeví velmi pěkně a přehledně zpracovaný průvodce, který umožní začátečníkovi vytvořit velice jednoduchou a přitom účinnou konfiguraci. Tato konfigurace není ani tak dána možností volby, ale spíš již přednastavená programem. V průvodci je pouze možnost vybrat interface připojený k internetu a popř. interface do lokální sítě s možností použití NATu. Po dokončení nastavení průvodce se vygeneruje skript pro *bash*, poté se ještě průvodce zeptá, zda má firewall spustit a nabídne důvody, proč by tak nemělo být učiněno. Při výběru spuštění se skript zavede do iptables a nastavená pravidla začnou fungovat. Skript je možno si tedy ručně doladit podle svých potřeb.



Obr. 10. Firestarter – průvodce

Při nastavováni je možné dále firewall editovat. Možnosti nejsou nikterak veliké, ale pravidla jsou docela složitá. Firestarter dělá velkou část úkonů automaticky a proto je výsledný skript vcelku obsáhlý a jeho účinnost je obrovská.

Při změně pravidel nebo nabíhání programu je veškerý přenos zakázán, čímž se zvyšuje zabezpečení systému.

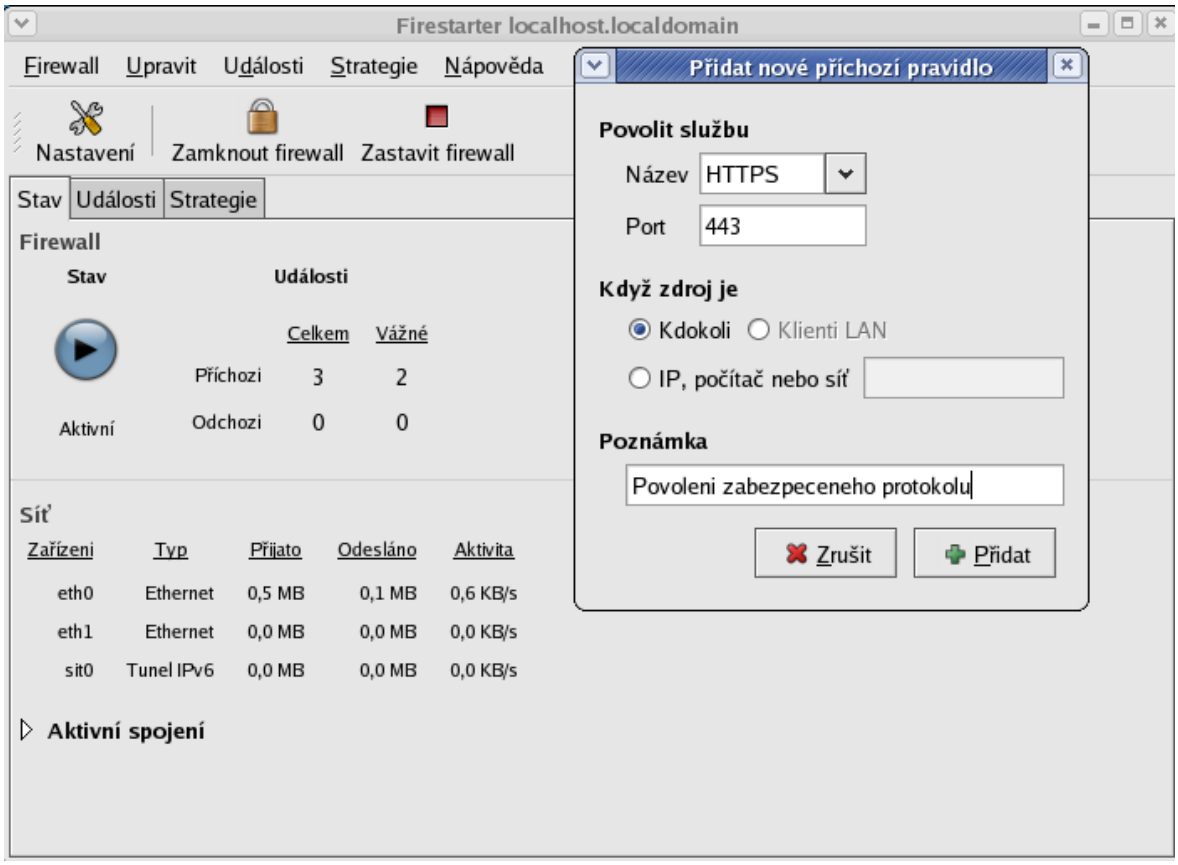
Firestarter zaznamenává události a jako první možnost v nastavení je přidaní pravidla určujícího, od kterých lokálních stanic se nemají tyto události zaznamenávat.

Dále je zde možnost filtrování příjmu a tvorby ICMP paketů. Ochrana před TCP floodem. Nechybí ani možnost nastavení flagů ToS pro lepší směrování – je možno priorizovat pracovní stanice, servery, X Windows systém, a to pomocí propustnosti, spolehlivosti nebo interaktivity. Také je dostupná filtrace portů s popisem blokované služby.

Program je lokalizován do bezmála 30 jazyků a nechybí ani čeština.

Tento program dokáže sledovat síťový provoz, a to v reálném čase. Po konfiguraci je tudíž možno vidět, jaké pakety procházejí přes firewall. Při neoprávněném provozu je možné ho zastavit a povolit až při opravě pravidel tak, aby daný provoz vyhovoval požadavkům uživatele.

K pokročilejším nastavením patří třeba NAT.



Obr. 11. Firestarter – přidání pravidla

2.4.5 Shrnutí grafických aplikací pro iptables

Bylo zde nastíněno několik různorodých aplikací pro pohodlné zkonfigurování firewallu. Díky těmto aplikacím je možno vyhnout se konfiguraci iptables přes příkazový řádek. Programy jsou pro různě pokročilé uživatele – od plně grafického rozhraní až po jednoduchý php skript.

I když jsou všechny programy v grafické formě, je nutné, aby uživatel měl minimální znalost, jak má postupovat. Na rozdíl od nastavení iptables v textovém rozhraní, které je spíše pro odborníky, nabízí grafické rozhraní velkou oporu i pro ty nejmíň zkušené, kteří mají jen základní znalosti portů, adres, funkce DHCP, popř. NAT.

V praktické části bude ukázáno, jak se taková aplikace vytváří. Ale poněvadž to není úkol jednoduchý, nedá se přesně předpokládat výsledný efekt ani do které z těchto skupin bude daná aplikace zapadat.

II. PRAKTICKÁ ČÁST

3 APLIKACE PRO NASTAVENÍ NETFILTER/IPTABLES

Cílem této práce bylo vytvořit aplikaci, která bude umožňovat základní nastavení modulu iptables v operačních systémech GNU/Linux. Důvodem je, aby uživatelé pracující na těchto systémech mohli snadněji nastavit firewall, a tím se chránit proti případným pokusům o průnik do daného systému nebo v případě síťového routeru o průnik do části sítě, do které směruje svůj provoz.

3.1 Podpora aplikace

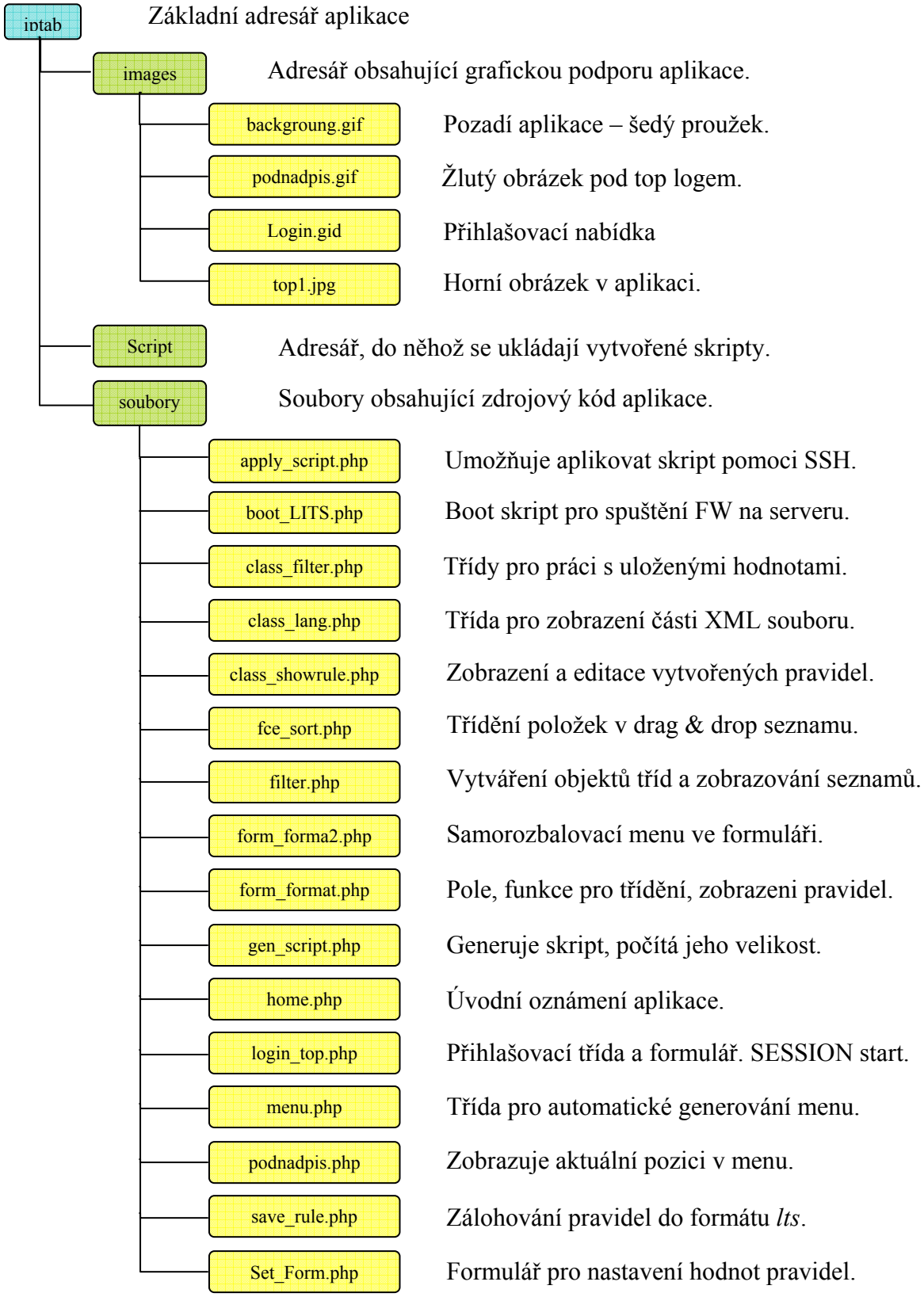
Ze zadání plyne, že celý program bude možno spustit z webového prohlížeče. Z toho důvodu byl zvolen programovací jazyk PHP verze 5. V tomto jazyce za pomocí programu Zend Studio[4] byla celá aplikace naprogramována. Dále je v programu použit repositář PEAR, což je zkratka termínu PHP Extensions and Application Repository. Jedná se o systém balíčků rozšiřující jádro PHP.

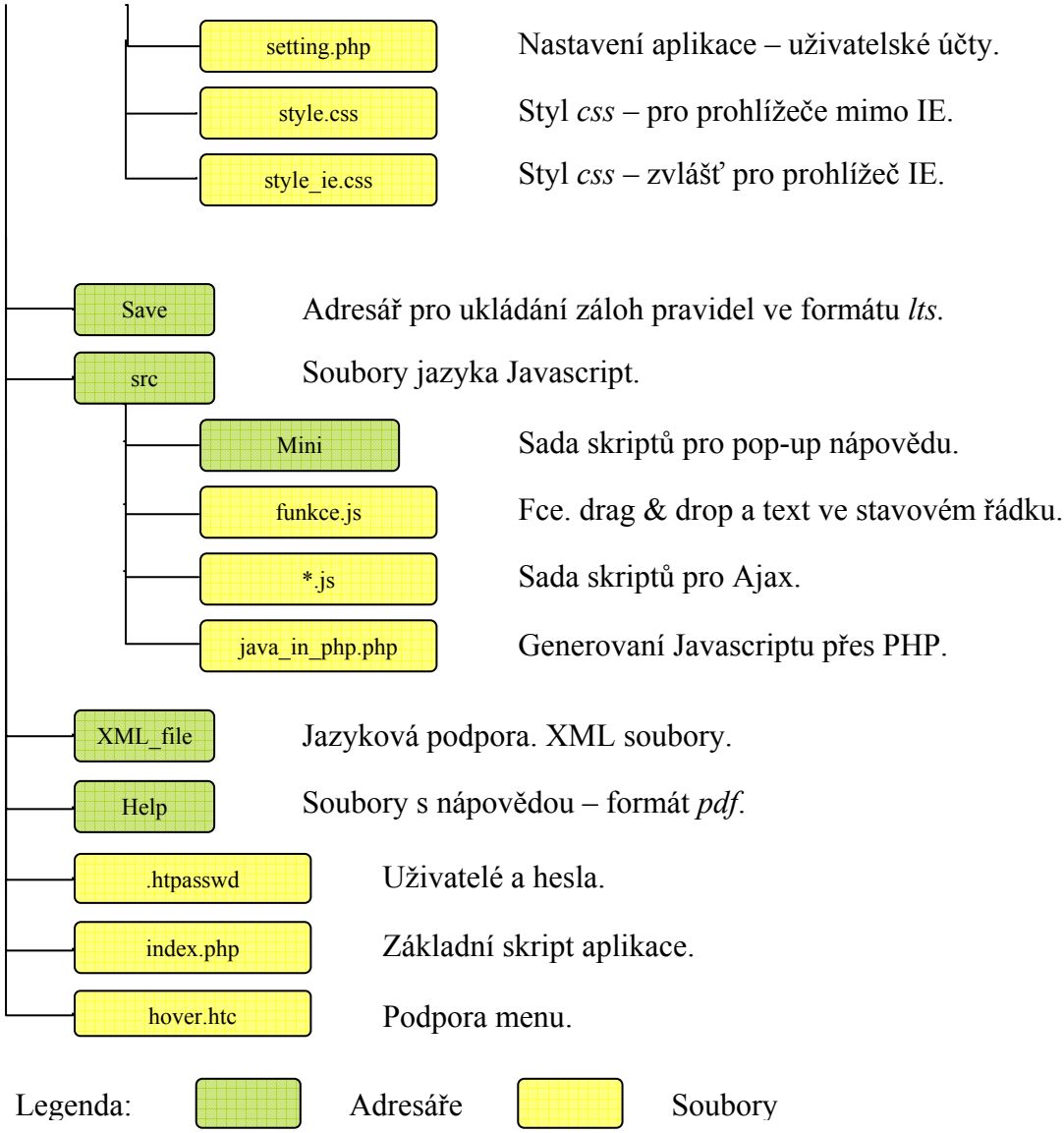
Balíčky PEAR se od jiných běžných formátů balíčků (jako jsou balíčky RPM v systému Linux, balíčky systému Debian nebo balíčky ve formátu PKG používané v systému System V UNIX) příliš neliší. Jedním z hlavních rozdílů je fakt, že balíčky PEAR jsou navrženy tak, aby byly nezávislé na hostitelské platformě. Nejsou tedy omezeny určitou rodinou operačních systémů. Většina balíčků PEAR je na platformě nezávislá. Takovéto balíčky lze instalovat na libovolné platformě, na níž lze instalovat distribuci PHP. [7]

Z důvodu použití PHP5 a repositáře PEAR je nutné, aby webový server, na kterém bude tato sada skriptů spouštěna (jedná se o skripty vytvářené aplikace), podporovala skriptovací jazyk této verze (nebo vyšší) a to i s daným rozšířením PEAR. Stejně tak jako další rozšíření, která nejsou v základním nastavení, a o nichž bude řeč později. Pozn.: kompletní shrnutí bude uvedeno na konci v části „Nastavení webového serveru pro spuštění aplikace“.

[4] Programováno v demoverzi dostupné z URL: <http://www.zend.com/store/download_list.php> [2007 – 4 – 28]

3.1.1 Struktura souboru aplikace LITS





Uvedená struktura zobrazuje, jak jsou jednotlivé soubory seřazeny ve struktuře aplikace. U jednotlivých souborů je popsáno, k čemu daný skript slouží. Popis je uveden jen heslovitě. Jelikož zdrojový kód ani vysvětlení nebudou v práci uvedeny, je pro důkladnější pochopení aplikace nutno nahlédnout přímo do zdrojového kódu, kde jsou podrobné popisy u každé funkce a třídy.

Níže bude uveden pouze popis a grafický vzhled aplikace. Bude zde popsáno, jak jednotlivé části aplikace pracují, a také zde budou pro objasnění uvedeny jen velmi krátké části zdrojového kódu.

3.2 Popis a grafický vzhled aplikace

Jak už bylo řečeno, aplikace má za úkol nastavovat iptables pro systém Linux a od toho se také odvíjí její název, který tvoří počáteční písmena ze slov výstižné věty v anglickém jazyce. Tato věta zní „*L*inux *I*p*T*ables *S*etting“. Zkratka tvořící efektivní název aplikace je potom LITS.

Obr. 12. Grafické logo aplikace LITS

Stejně jako tento obrázek (Obr. 12) tak i ostatní vznikly pomocí velmi kvalitní sady programu CorelDRAW Graphics Suite X3[5].

3.2.1 Přihlášení do aplikace

Při zadání URL adresy s cílem na danou aplikaci, se jako první ukáže vstupní formulář, který požaduje po uživateli zadat přihlašovací jméno a heslo (Obr. 13).

Část kódu starající se o obsluhu přihlašování, je z repositáře PEAR zmíněného výše. Pokud tento nebude nainstalován, nepodaří se do aplikace přihlásit a ani ji spustit. Na obrazovce se objeví pouze podklad (šedobílé proužky), který je načten v bufferu.

Kód, starající se o přihlašování, je v souboru: *soubory/login_top.php*. Jako první jsou zde vloženy soubory z repositáře PEAR definované touto částí kódu:

[5] Použita demoverze, kterou je možno získat z URL: <http://apps.corel.com/int/cz/cgsx3.html> [2007-03-16]

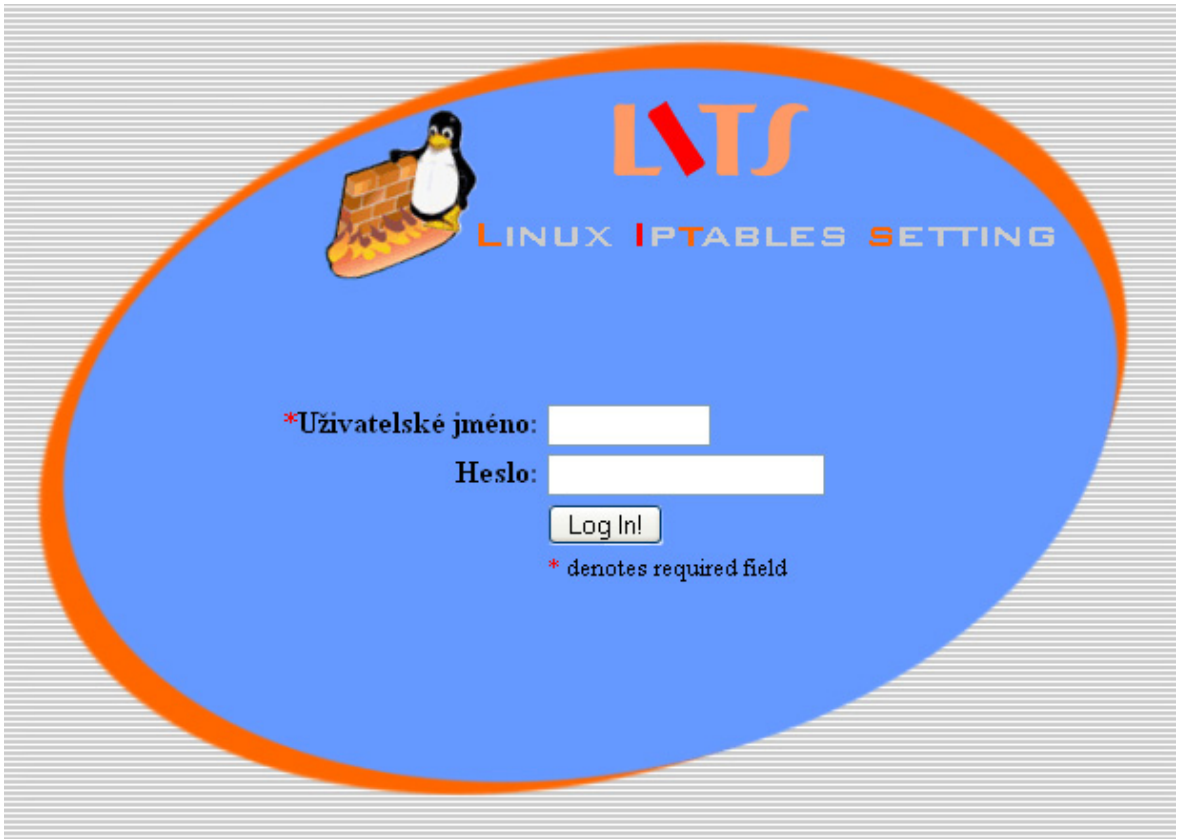
```
require_once "Auth.php";

require_once 'HTML/QuickForm.php';
```

Aplikace tedy využívá PEAR balíčky:

- Auth
- QuickForm

Výše uvedené balíčky jsou nutné k tomu, aby celá aplikace fungovala.

Přihlašování probíhá vytvořením objektu třídy Auth, kterému jsou následně předána data získaná z formuláře, dále pak soubor s uživateli a hesly. Pokud třída najde v souboru *.htpasswd* uživatelské jméno a heslo shodné s údaji získanými z přihlašovacího formuláře, bude žadateli umožněn přístup. Budou-li data chybná, celá procedura se opakuje.



Obr. 13. Přihlašovací formulář do aplikace LITS

Nebude-li zadáno jméno, aplikace na to upozorní. Heslo se uvádět nemusí, to záleží na správci aplikace.

Standardní uživatel je administrátor, který má uživatelské jméno: admin, heslo: admin[6]. Pouze tento uživatel může měnit uživatelské účty libovolného uživatele. Všichni ostatní můžou manipulovat pouze se svým účtem.

3.2.2 Vícejazyčná podpora

Protože se jedná o aplikaci, kterou bude možno šířit pod licencí GPL[7], existuje určitá pravděpodobnost, že bude používána i v jiných zemích. Z toho důvodu byla do aplikace vložena vícejazyčná podpora. Tato podpora je v samotném kódu realizována pomocí souborů XML. Data z toho souboru jsou načítána pomocí nativní podpory XML, která je realizována prostřednictvím rozhraní *SimpleXML*. SimpleXML je extenze, není tedy pevnou součástí PHP 5. Aby byla k dispozici, je třeba konfigurovat PHP s příznakem *--enable-simplexml* (je možné, že určité balíky určené k instalaci PHP5 již mají tuto volbu aktivovanou). Základním stavebním kamenem datového modelu extenze SimpleXML je objekt *SimpleXMLElement*. Dokument XML je reprezentován objektem kořenového elementu (root element). Dokument a jeho kořenový element jsou v SimpleXML jedna a táž entita.

Kód ukazující jak je daný objekt použit v programu:

```
$xml = new SimpleXMLElement('XML_file/'. $soubor[$i][0] .'.xml', NULL, TRUE);
```

Tento řádek vytváří v programu nový objekt třídy SimpleXMLElement. Třída předá nově vytvořenému objektu odkaz na soubor (*$soubor[$i][0]*) XML, kde se do proměnné *$soubor* ukládá aktuálně vybraná jazyková sada.

---

[6] Po prvním přihlášení se doporučuje toto heslo změnit.

[7] Pokud se programátor rozhodne šířit svůj program pod licencí GNU GPL, zavazuje se příjemci tohoto programu poskytnout stejná práva, jaká má on sám nebo jaká mu byla prostřednictvím GPL poskytnuta. Tak je zajištěno, že svobodný kód nemůže být využit v kódu proprietárním. Celá licence je dostupná na URL: <http://www.gnu.org/copyleft/gpl.html> [2007-5-4]

Všechny jazykové sady jsou uloženy v adresáři *XML_file*. Z tohoto adresáře program automaticky vybírá všechny soubory a dává možnost výběru jazykové sady. Pokud tedy bude chtít uživatel používat jiný jazyk, než češtinu a angličtinu[8], stačí, když do daného adresáře vloží soubor s názvem podle jazyka (např. czech.xml) a program mu již nabídne tuto možnost automaticky, bez jakéhokoliv zásahu do kódu programu. Všechny soubory jsou kódovány v jazykové sadě UTF-8, a proto je nutné, aby i nově vytvořené XML dokumenty obsahující jazykové struktury, byly v tomto formátu kódovány.

Struktura XML souboru:

```
<?xml version="1.0" encoding="UTF-8"?>
<CZECH>

      <LANG>
            <TEXT>cz</TEXT>
      </LANG>

      <FORMPOPIS>
            <POPIS>Co s pravidlem? :</POPIS>
            <POPIS>IP adresy a porty pro DNAT :</POPIS>
            <POPIS>IP adresy a porty pro SNAT :</POPIS>
            <POPIS>Zdrojová síť nebo adresa :</POPIS>
            <POPIS>Cílová síť nebo adresa :</POPIS>
            <POPIS>Vstupní Interface :</POPIS>
            <POPIS>Výstupní Interface :</POPIS>
            <POPIS>Síťový protokol :</POPIS>
            <POPIS>Zdrojový port(y) :</POPIS>
            <POPIS>Cílový port(y) :</POPIS>
            <POPIS>Porovnání src a dst portů :</POPIS>
```

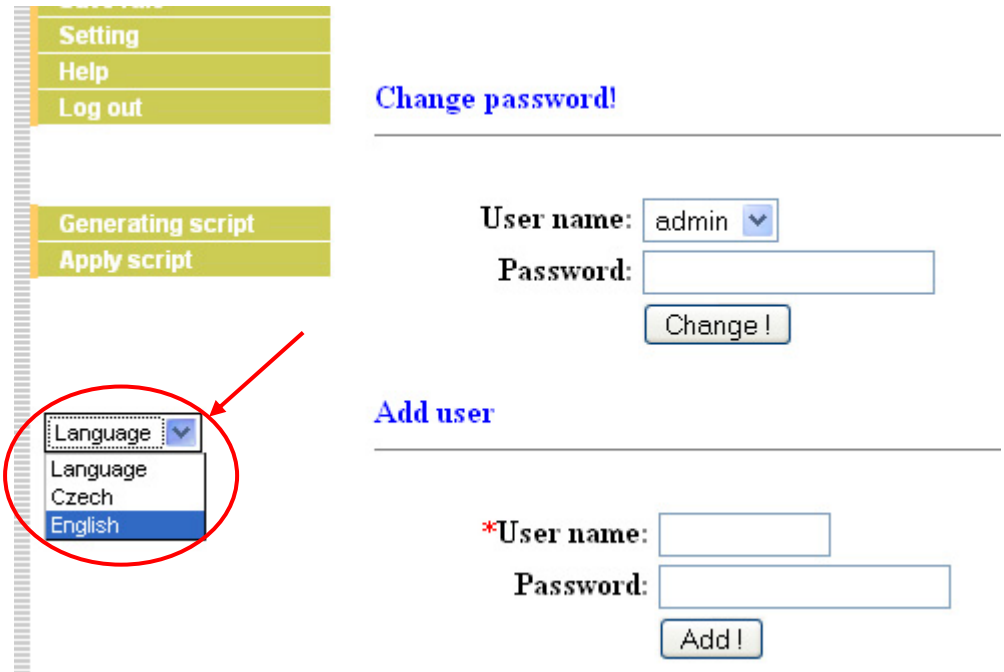
[8] Zde uvedené jazyky jsou do aplikace vloženy jako standardní nabídka.

```
            <POPIS>TCP flags :</POPIS>
            <POPIS>Nastavení TCP :</POPIS>
            <POPIS>Typ ICMP :</POPIS>
            <POPIS>Limitní poměr :</POPIS>
            <POPIS>Stav spojení :</POPIS>
            <POPIS>Typ Služby :</POPIS>
      </FORMPOPIS>
```

V aplikaci jsou jednotlivé položky vybírány pomocí výše uvedeného objektu $xml a to následujícím způsobem:

```
            $xml->FORMPOPIS->POPIS[0];
```

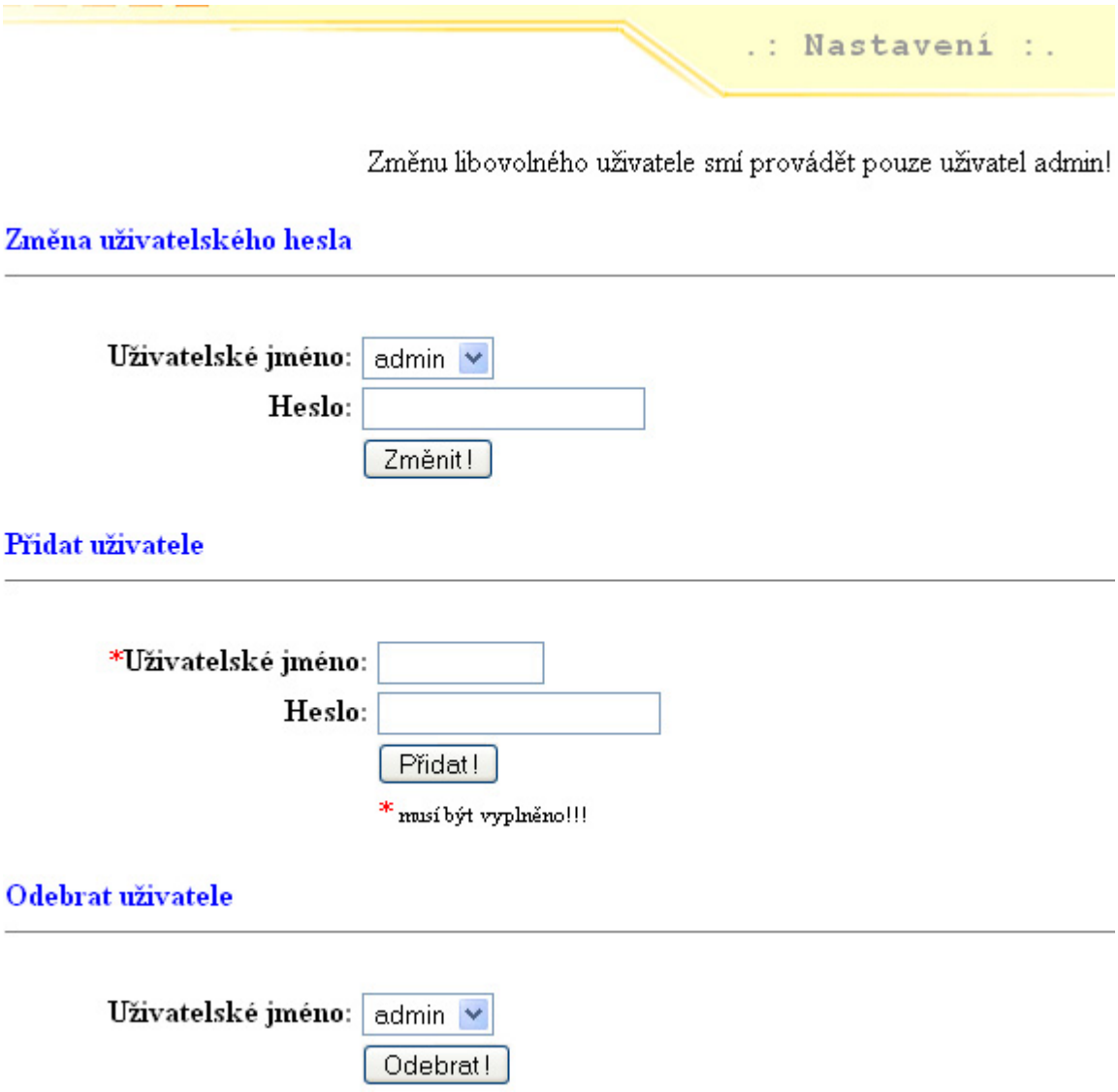
V programu je možno vybrat jazyk pomocí rozbalovacího seznamu, který je umístěn pod menu (viz. Obr. 14). Tato nabídka je tvořena pomocí javascriptu.



Obr. 14. Možnost výběru jazykové sady

3.2.3 Nastavení

Nastaveni aplikace je jen základní. Umožňuje uživateli měnit pouze uživatelské účty. V případě uživatele „neadmina" pouze účet svého vlastního profilu. Uživatel administrátor[9] nemůže být z aplikace odstraněn.



Obr. 15. Nastavení aplikace – správa uživatelů

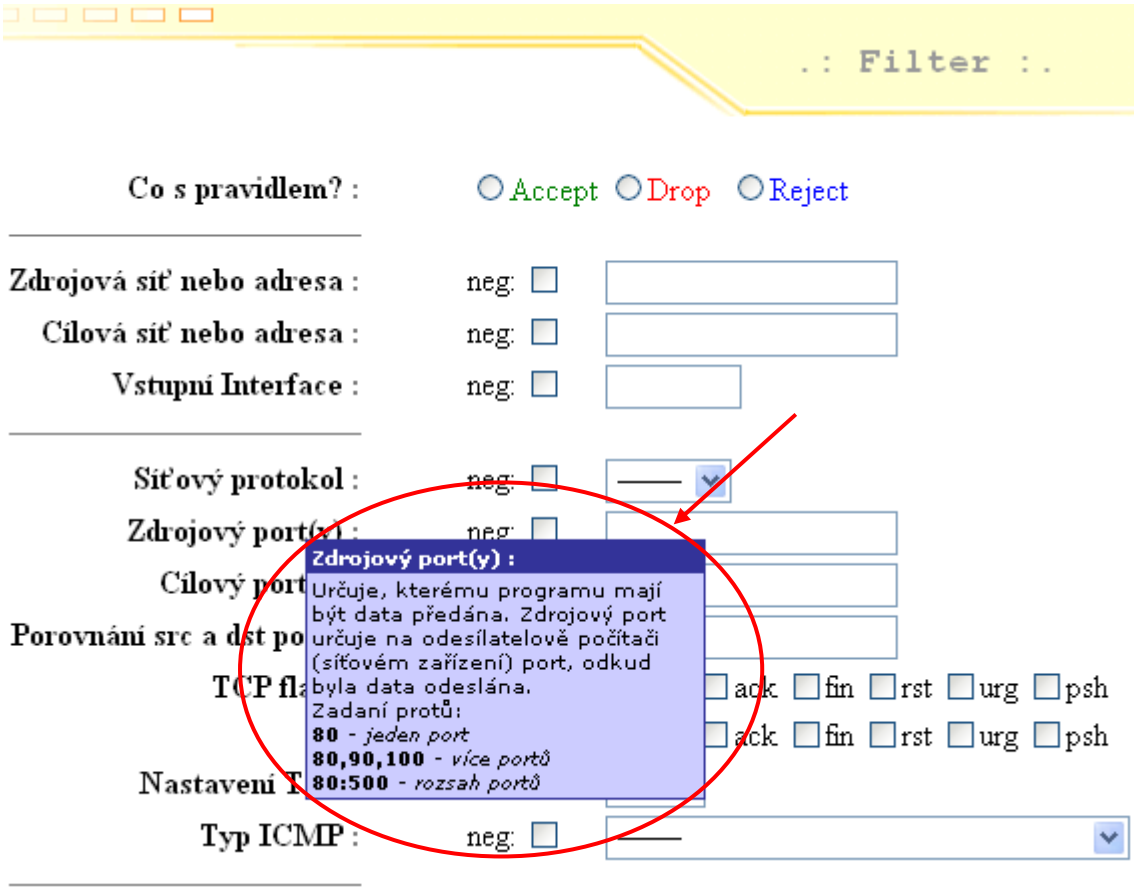
[9] Administrátor má standardně nastavené přihlašovací jméno: admin, heslo: admin. Heslo se dá pomocí nastavení změnit a to je také po přihlášení doporučeno.

Nastavení aplikace spolupracuje s repositářem PEAR, konkrétně s již vytvořenou třídou pro přihlašování do aplikace, kde za pomocí tohoto objektu zjišťuje všechny potřebné informace. Nastavení aplikace je prováděno pomocí kódu umístěného v souboru: */soubory/setting.php*.

3.2.4 Nápověda v LITS

Jako většina aplikací má i LITS nápovědu. Nápověda je realizována pomocí souboru, který je možno zobrazit v záložce *Nápověda*. Tento soubor je ve formátu *pdf* a je v něm obsažena textová a obrázková prezentace o aplikaci LITS. Do této nápovědy je možno nahlédnout v příloze této práce.

Do aplikace byl vložen ještě jeden druh nápovědy, tzv. rychlá nápověda. Tuto nápovědu je možno zobrazit v aplikaci po najetí myší na určitý text.

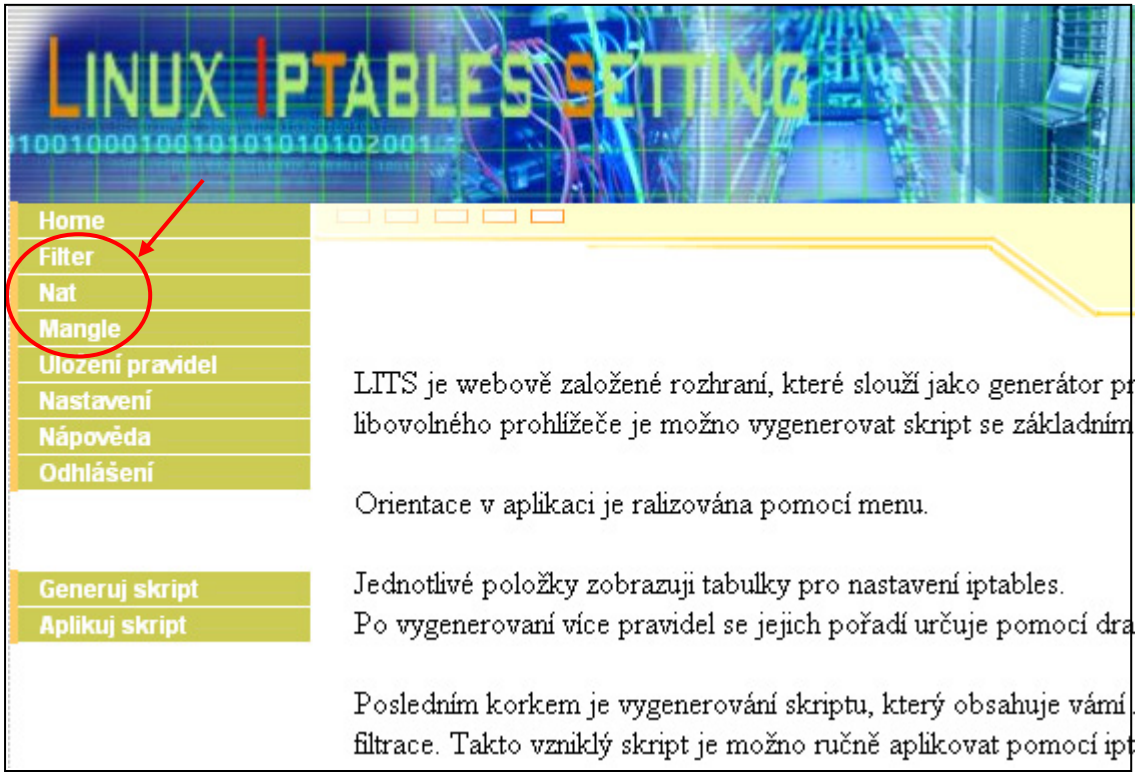


Obr. 16. Rychlá nápověda

Rychlá nápověda je realizována pomocí souboru *class_lang.php*. Zde je definována třída *lang*, která se stará o vyskakování oken. Třída načítá data z aktuální jazykové sady xml, a ty zobrazuje. Vyskakování oken je realizováno pomocí sady javascriptu overLIB[10]. Pro použití byl použit balík „Mini“, jenž je umístěn v adresáři *src/*.

3.2.5 Nastaveni pravidel iptables v aplikaci

Základem celé aplikace je nastavení pravidel pro iptables. Tyto pravidla jsou nastavována podle standardu frameworku netfilter/iptables. Možnost nastavení se dělí do tří oblastí podle druhu použití. Tyto oblasti jsou definovány v iptables pomocí tzv. tabulek, které obsahují námi vytvořená pravidla. Pro přístup k nastavení těchto jednotlivých tabulek bylo nutno vytvořit ovládací strukturu, jež daný přístup umožní. Tato struktura je tvořena pomocí menu aplikace, kde jsou přímo uvedeny názvy těchto tabulek (filter, nat, mangle). Ukázka je vidět na Obr. 17.



Obr. 17. Struktura menu pro výběr tabulky

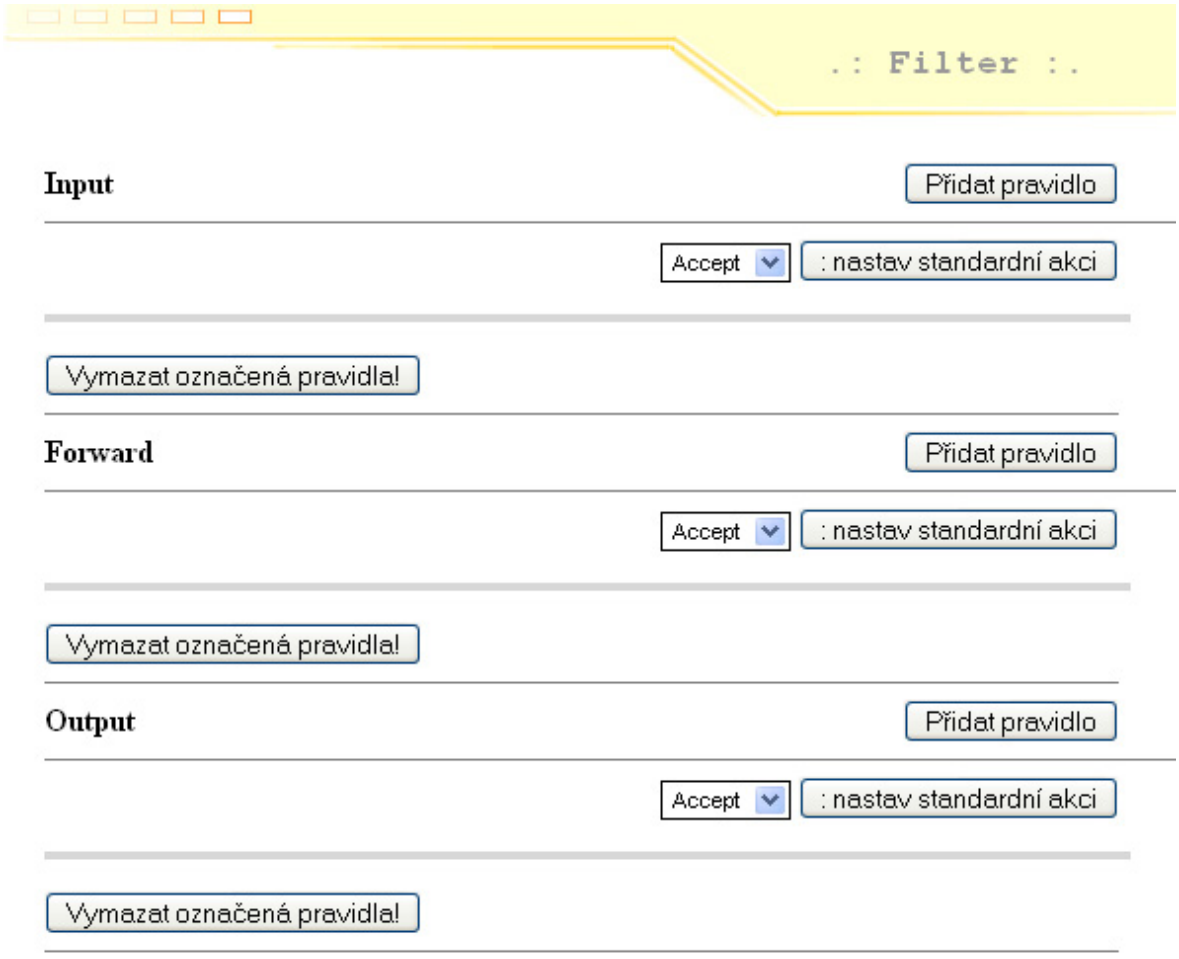
[10] Sada a manuál je dostupný na URL: < http://www.bosrup.com/web/overlib/> [2007-3-13]

Pro výběr nastavení požadované tabulky klikne uživatel na požadovanou položku. Zde se objeví seznam chain (řetězců), které je možno aplikovat v rámci dané tabulky. Pro jednotlivé hodnoty se toto nastavení odlišuje.

3.2.5.1 Přidání pravidla

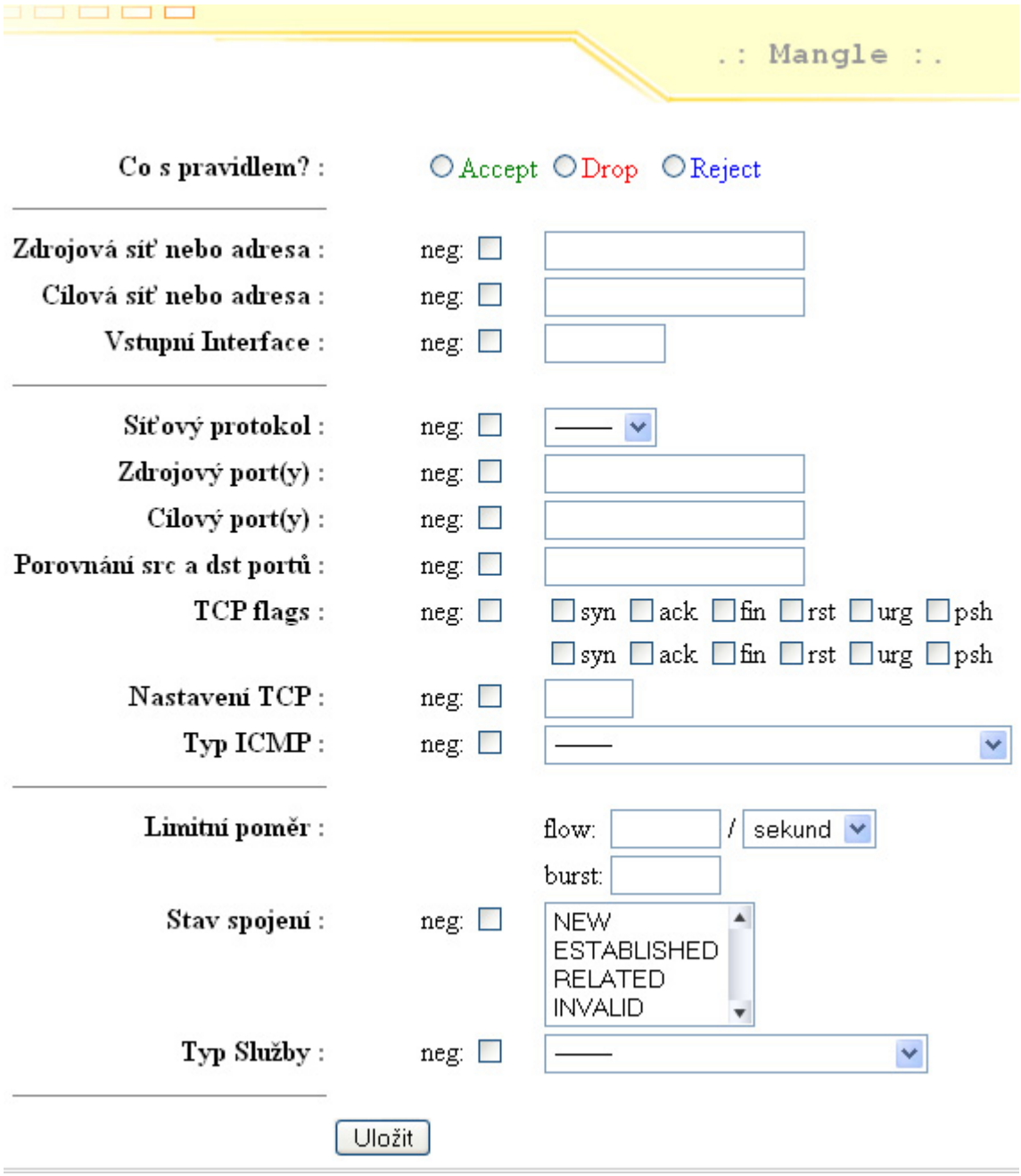
Při vybrání položky tabulky je zobrazena obrazovka s danými řetězci (viz. Obr. 18). Tyto položky jsou v aplikaci realizovány pomocí třídy *c_showrule*, která je obsažena v souboru *class_showrule.php*. Tato třída se stará pouze o zobrazení vybraných pravidel a později o jejich editaci.

Pokud má být přidáno další pravidlo, je nutno kliknout na tlačítko Přidat pravidlo. Zobrazí se formulářové pole s požadovanými hodnotami. Uživatel si vybere ty, které chce nastavit a dané pravidlo uloží. Po uložení je vrácen zpět na stránku, odkud zaslal požadavek na přidání pravidla.



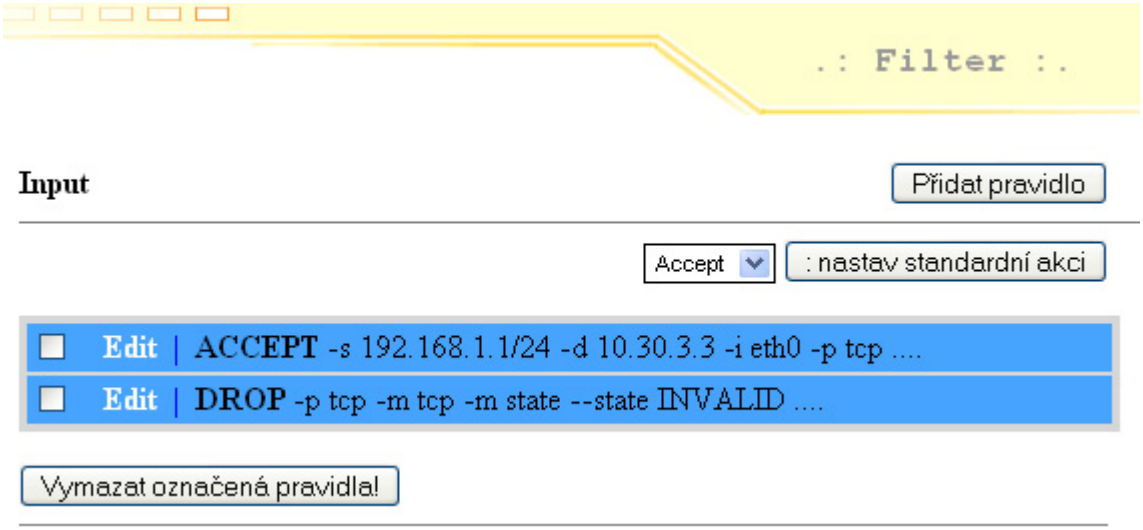
Obr. 18. Ukázka tabulky filter

Nově vzniklé pravidlo předá prostřednictvím odeslání formuláře (Obr. 19) svoje hodnoty třídě *c_netfilter,* jež je v souboru *class_filter.php*. Tato třída se stará o ukládání pravidel a jejich vrácení hodnot zpět aplikaci. Z nově vzniklého objektu této třídy si již zmíněná třída *c_showrulle* vezme hodnoty a stará se o jejich zobrazení na obrazovce. Vyobrazení je tvořeno pomocí výkonných javascriptů obsažených v balíku *Ajax*. Pomocí tohoto balíku jsou zobrazeny jednotlivé pravidla v blocích podle řetězce.

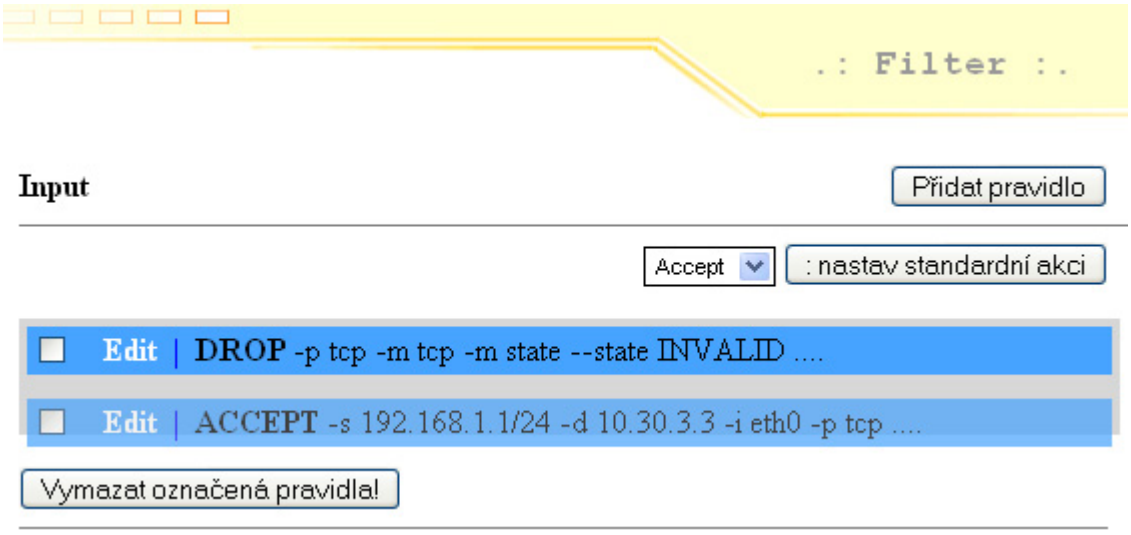


Obr. 19. Formulář pro vytvoření pravidla

Ajax vytvoří nabídky, které je možno přetahovat myší, a tím měnit pořadí jednotlivých pravidel. Jednotlivé položky tohoto seznamu jsou tvořeny pomocí html tagu (*ul, li*). Každý takto dynamicky vzniklý seznam má svoje id. Toto id je pro každý seznam jedinečný. A to z toho důvodu, aby ajax rozpoznal, o který konkrétní seznam se jedná a aplikoval na něj svoje metody. Z každého pravila je načteno prvních šedesát znaků a ty jsou v jednotlivých položkách seznamu zobrazeny.



Obr. 20. Ukázka vytvořených pravidel



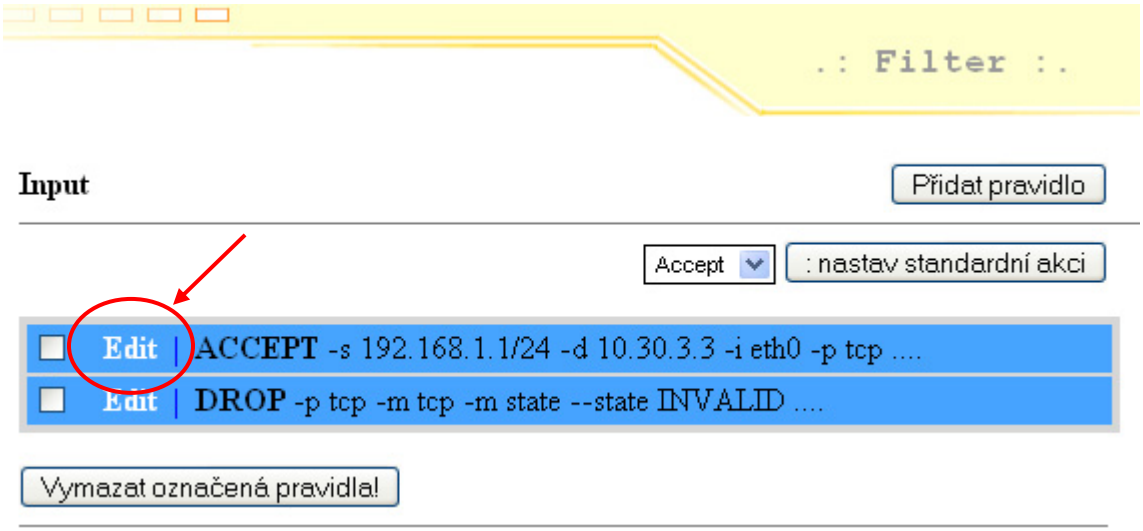
Obr. 21. Ajax – seznam drag & drop

Na předchozím obrázku je vidět, jak ajax pracuje. Ze široké škály možností byla v aplikaci využita pouze funkce drag & drop. Zde je vidět, jak je položka **ACCEPT** přetahována myší pod položku **DROP**.

Změnu pořadí provádí samostatná funkce *f_sort()*, které jsou předány nové pozice pomocí funkce *ajaxReqest.*

3.2.5.2 Změna pravidla

Změna pravidla se provádí po kliknutí na položku „*Edit*“ u příslušného pravidla, které má být právě upravováno (viz. Obr. 22).



Obr. 22. Editace již vytvořeného pravidla

Po kliknutí na tento odkaz se ukáže stejný formulář, jako pro přidání nového pravidla. Jediná změna je v tom, že položky, které byly nastaveny, jsou ve formuláři předvyplněny. Pro ukončení editace a uložení změn je nutno opět kliknout na tlačítko uložit, které je strategicky umístěné dole pod formulářem.

Při kliknutí na tlačítko editace je místo čísla následujícího pořadí (standardně se předává další pořadí pravidla po předchozím), předáno pořadí editovaného pravidla v dané tabulce resp. řetězci. Editace je tedy udělaná elegantním způsobem, a to tak, že pravidlo není ve skutečnosti editováno, ale je vytvořeno zcela nové, které nahrazuje pravidlo na pozici editovaného pravidla. Díky tomu je možno použít již vzniklé principy (třídy, funkce) pro ukládání těchto hodnot. Bylo nutno pouze vytvořit funkci pro získání čísla pravidla pro

editaci. Editaci pravidel obstarává třída *c_showrulle* a *c_netfilter*. První ze jmenovaných pouze vytváří odkaz „*Edit*“ a stará se o správné přiřazení hodnot z objektu třídy *c_netfilter*. Všechny hodnoty jsou předávány pomocí SESSION.

3.2.6 Generování skriptu

Cílem celé aplikace je vygenerování skriptu s pravidly pro iptables. Po nastavení všech požadovaných pravidel je možno skript vygenerovat. Generování skriptu probíhá pomocí skriptu *setting.php*. V tomto souboru se zjišťuje, zdali je nějaké pravidlo vytvořeno. Pokud nejméně jedno pravidlo existuje, je možno vygenerovat daná pravidla do souboru. Výsledný skript se ukládá do souboru *Script/script*.

Script byl úspěšně vygenerován!

Byl vygenerován následující script o velikosti 1,0 kB:

```
#!/bin/bash
#
#
# Script byl vygenerovan aplikaci LITS
# Autor aplikace: Barkas
#
#

#message function
function f_message() {
        echo -e `/bin/date +%d.%m.%Y-%H:%M:%S` $1: $2
}
echo ""

f_message "!!! Spoustim skript generovany aplikaci LITS " ""
f_message "START" "Zavadim pravidla IPtables"

# Cesta k programu iptables
  IPTABLES="/sbin/iptables"

#
```

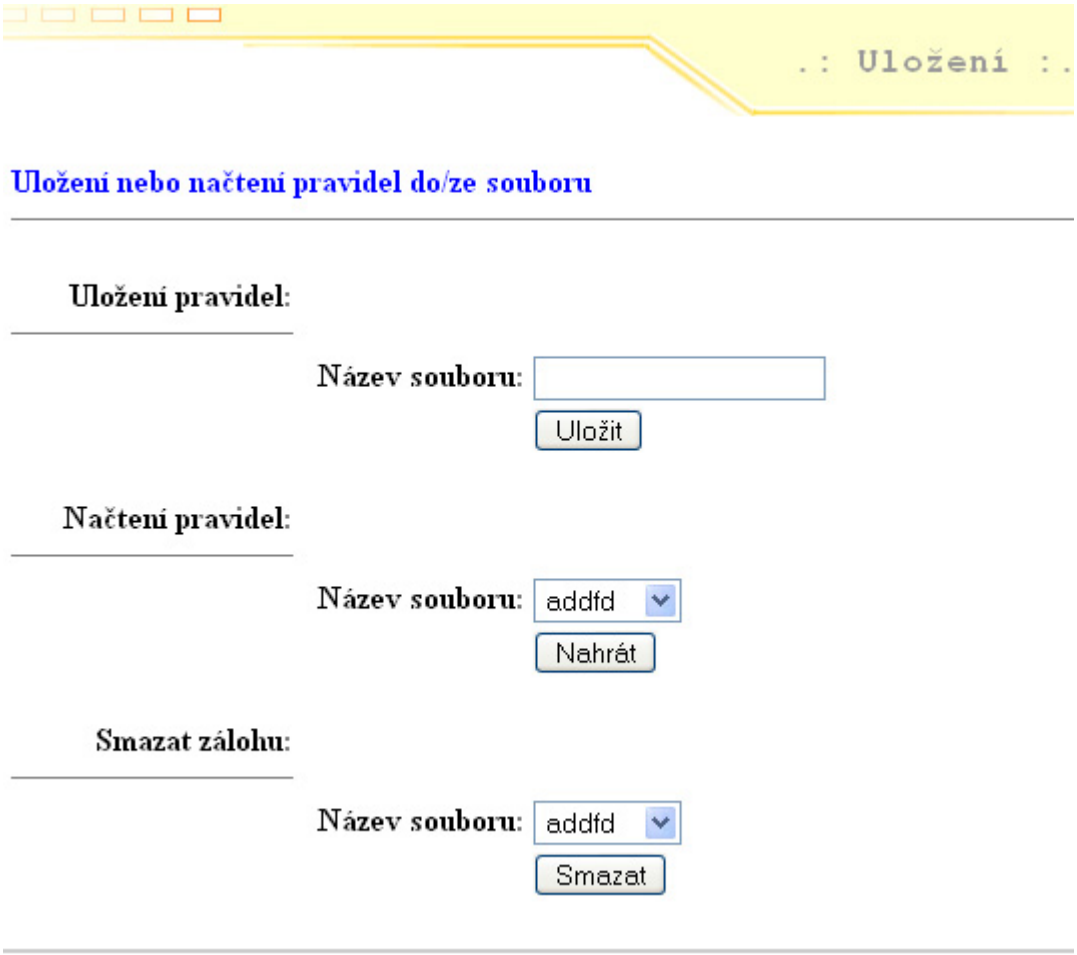
Obr. 23. Ukázka vygenerovaného skriptu.

Po vygenerování je na obrazovce zobrazeno, zdali byl skript vygenerován a jeho velikost. Také se zobrazuje podoba výsledného skriptu.

Výsledný skript je tvořen několika funkcemi, které se starají o výpis zpráv v případě ručního zavedení skriptu, nebo při zavedení po startu systému. Skript je možno spustit na jakémkoliv GNU/Linux, který podporuje netfilter/iptables. Pro bližší pochopení vygenerovaného skriptu se doporučuje prozkoumat jej.

3.2.7 Uložení vygenerovaných pravidel

Pokud vytvoříte pomocí aplikace LITS rozsáhlé nastavení firewall, bylo by po vygenerování skriptu a jeho případné aplikaci užitečné uložit si toto nastavení pro pozdější použití.



Obr. 24. Uložení nebo obnova pravidel do/ze souboru

Z obrázku (Obr. 24) je vidět, které úkony je možno s pravidly vykonávat. Všechny funkce k tomu určené jsou v souboru *soubory/save_rule.php.* Chce-li uživatel vytvořená pravidla uložit, zadá název souboru pro uložení a aplikace provede zálohou pravidel do adresáře *Save/.* Výsledný soubor bude mít název např. *záloha.lts*.

Formát *lts* je tvořen uložením serializovaných pravidel. Tato pravidla jsou oddělena pomocí oddělovačů, aby bylo později možné, je znovu obnovit. Oddělovače tabulek jsou TABLETEBLE a oddělovače jednotlivých pravidel v tabulkách tvoří název RULERULE. Tyto oddělovací části jsou při obnovení v souboru vyhledávány a pravidla tak tříděna zpět na své pozice.

Uložení souboru se projeví v rozbalovacích seznamech, kde bude daná záloha načtena k dalšímu použití. Bude možno ji obnovit nebo odstranit. Při obnovení se uložená pravidla načtou ze souboru a bude s nimi možno dále pracovat (editovat, mazat, měnit pořadí). Pozor, předchozí pravidla budou smazána. Před obnovením je proto vhodné provést zálohu nově vytvořených pravidel.

3.2.8 Aplikace pravidel na server

Základem celé aplikace LITS je, aby bylo možno na serveru, popř. osobním počítači nastavit firewall. Jelikož tento operační systém používá k nastavení FW aplikaci netfilter/iptables, která je v základním režimu konfigurovatelná přes shell[11] pomocí příkazového řádku, není tato úloha nikterak jednoduchá. Aby bylo možno tuto úlohu podstatně ulehčit, je možno použít grafickou nadstavbu jakou je např. LITS.
Z toho důvodu byla do aplikace LITS tato možnost naprogramována. Komunikace mezi serverem a aplikaci probíhá pomocí zabezpečeného přenosu SSH2. Po najetí na odkaz „*aplikuj skript“* se zobrazí přihlašovací formulář, do kterého je nutno zadat adresu serveru, na který má být skript aplikován, jméno a heslo uživatele, který má právo přistupovat k systémovým souborům na serveru, popř. právo konfigurace iptables. Pokud je přihlášení k serveru (osobnímu počítači) úspěšné, zobrazí se informace o serveru a sada tlačítek, pomocí nichž je možno pracovat s naposledy vygenerovaným skriptem.

[11] Shell je program, který čte příkazy z terminálu nebo ze souboru (tzv. skriptu) a spouští je.

Možnosti po přihlášení na server:

- **Pošli skript na server** – odešle skript uložený v adresáři *Script* pod názvem *script* a ten uloží na straně serveru do adresáře *etc/LITS*. Dále program nastaví tomuto souboru práva s atributy 755 (vlastník – čtení, zápis, spouštění, skupina – čtení, spouštění, ostatní – čtení, spouštění). Dále vygeneruje soubor pomocí skriptu *boot_LITS.php* a ten uloží opět na serveru se stejnými právy, a to do adresáře *etc/init.d/* pod názvem *boot_LITS*. Tento skript je vytvořen pro případ, že uživatel využije v aplikaci LITS možnost spustit skript při startu systému, a tím automatické načtení pravidel do iptables.
- **Aplikuj pravidla** – spustí nahraný skript na serveru. Po spuštění jsou všechny pravidla obsažená v daném skriptu aplikována do iptables a začnou být plně funkční. Při ručním spuštění skriptu (ne v aplikaci LITS) je možno sledovat informace o průběhu integrace nových pravidel.
- **Spuštění při bootování** – při restartu, nebo jakémkoliv pádu systému by pravidla aplikovaná v iptables nebyla po spuštění opět aktivní. Proto je nutné zajistit tuto možnost. Na serveru již nahraný skript *boot_LITS* obsahuje funkce pro spuštění skriptu po startu. Aplikace LITS zjistí ze serveru číslo runlevelu (tj. z jakého módu startuje počítač: runlevel 5 - grafický mód, runlevel 3 - jedná se o textový mód, atp.) a do tohoto runlevelu přidá spouštěcí záznam pro již zmíněný *boot_LITS,* který se už postará o spuštění skriptu s pravidly.

  Pro přidání skriptu *boot_LITS* do spouštěcího záznamu je využita utilita *chkconfig*. Z toho plyne, že aplikovat spuštění při startu je možno jen na systémech podporujících danou utilitu. Přímo ze serveru je možno přidat spouštěcí záznam ručně, a to pomocí následujícího příkazu:

```
chkconfig --add boot_LITS
```

  Po přidání je pravidlo aplikováno do již zjištěného runlevelu, který je obsažen ve spouštěcím skriptu.

- **Odpojení od serveru –** aplikace zruší spojení se serverem. Veškeré údaje o spojení budou smazány. Při další práci s aplikací skriptu je nutno se znovu přihlásit.

3.3 Nastavení webového serveru pro spuštění aplikace

Aby bylo možné aplikaci spustit, je nutno vědět, jaké jsou k tomu nutné nástroje a moduly.

Celá aplikace je nezávislá na platformě, na které běží. Nutností je, aby daný server podporoval následující moduly a rozšíření:

1. **PHP 5 –** aplikace byla v tomto jazyku naprogramována a využívá přímo funkce, které starší verze nepodporovaly.
2. **PEAR –** PHP 5 musí být zkompilován za pomocí tohoto repositáře. Problematika repositáře PEAR byla v práci uvedena výše.
3. **SSH 2 –** pro možnost komunikace se serverem je nutno, aby bylo toto rozšíření podporováno. Aplikace bude pracovat i bez něj, ale nebude možno aplikovat skript přímo na server prostřednictvím aplikace LITS.

3.3.1 Postup zprovoznění při odzkoušení

Jelikož konfigurace serveru nebyla předmětem dané práce a možnosti konfigurace serveru se na jednotlivých operačních systémech liší, bude zde popsán způsob, jakým byla aplikace odzkoušena.

Celá aplikace byla programována a zkoušena na systému Microsoft Windows XP, kde také běžel webový server pro tuto aplikaci. Webový server byl realizován pomocí základního balíku XAMPP Windows 1.6.1[12].

XAMPP 1.6.1 obsahuje:

- Apache 2.2.4
- MySQL 5.0.37
- PHP 5.2.1 + PHP 4.4.6 + PEAR
- PHP-Switch win32 1.0 (please use the "php-switch.bat")

[12] Balíky jak pro Windows, Linux, Mac OS a Solaris je možno stáhnout z URL: <http://www.apachefriends.org/en/xampp.html> [2007-5-5]

- XAMPP Control Version 2.4 from www.nat32.com
- XAMPP Security 1.0
- SQLite 2.8.15
- OpenSSL 0.9.8e
- phpMyAdmin 2.10.02
- ADOdb 4.94
- Mercury Mail Transport System v4.01b
- FileZilla FTP Server 0.9.23
- Webalizer 2.01-10
- Zend Optimizer 3.2.4
- eAccelerator 0.9.5 für PHP 5.2.1 (comment out in the php.ini)

Celá instalace je jednoduchá a spočívá ve spuštění instalačního průvodce. Ten je možno spustit pomocí dávkového souboru *setup_xampp.bat* nacházejícího se v základním adresáři.

Tento balík již obsahuje všechny potřebné rozšíření pro spuštění aplikace. Pouze pro nastavení SSH 2 je nutno v konfiguračním souboru *php.ini*, nacházejícího se v adresáři *xampp/apache/bin,* odkomentovat řádek:

```
extension=php_ssh2.dll
```

Potom stačí pouze rozbalit celou aplikaci do adresáře *htdocs/* a do webového prohlížeče zadat následující URL: *http://127.0.0.1/LITS*. Poté je možno začít pracovat s aplikací LITS (Linux iptables settings) a nastavovat pravidla podle toho, jak nejefektivněji je to možné.

Jako webový prohlížeč byl použit Mozilla Firefox 2.0.0.3 a pro tento prohlížeč byl také celý program programován. V ostatních prohlížečích není zaručena správná funkčnost aplikace.

ZÁVĚR

Závěrem se dá říct, že webové technologie se neustále vyvíjejí. K dispozici jsou programátorům stále lepší a komplexnější řešení a bezpečnější jazyky. Aplikace, které jsou dnes realizovány složitým způsobem, mohou být v blízké budoucnosti realizovány mnohem jednodušeji. Napomáhají tomu nové standardy, nápady a řešení rodící se v hlavách webových vývojářů a tvůrců webových technologií. I přesto, že při psaní aplikace byly použity moderní technologie a principy, je jen otázka času, kdy i tato nová aplikace bude zastaralá.

Díky této práci vznikla další možnost nastavení zabezpečení pro operační systémy GNU/Linux, která dává uživatelům na výběr další z různorodých aplikací. Jelikož vytvořená a v praktické části popsaná aplikace LITS byla programována pro možnost co nejjednoduššího ovládání, je možné, že ji využijí zejména začínající správci pro nastavení svých nově nainstalovaných systému a zmenší tím procento šíření virů mezi jednotlivými počítači.

Jelikož jsou dnes útoky proti informačním technologiím stále důmyslnější a efektivnější, je nutno aby se stejným tempem vyvíjela i pravidla a postupy pro zabezpečení systému a jejich subsystémů. Jelikož byla vytvořená a plně fungující aplikace LITS programována jako nadstavba pro iptables, není zaručeno, že během vývoje nezmění tato aplikace standardy pro načítání a ukládání nově vytvořených pravidel a aplikace se tím pádem stane nepoužitelná. Díky objektově vytvořenému kódu je samozřejmě možné tuto situaci rychle napravit.

Pojmout celou problematiku síťových technologií, GNU/Linux, či teorie zabezpečení by zabralo přinejmenším na tlustou knihu pro každé z těchto témat. Záměrem však bylo seznámit vás v této práci s těmito oblastmi na základní úrovni. Kromě seznámení s danými problematikami bylo hlavní prioritou vytvořit grafickou nadstavbu pro framework iptables. Vznikla tak velmi rozsáhlá a plně fungující aplikace nazvaná LITS (Linux Iptables Settings), kterou je možno zdarma šířit podle uvážení jednotlivých uživatelů.

Pro podrobnější pochopení látky je nutné neustále sledovat odbornou literaturu. Neboť jak půjde vývoj kupředu, bude i tato literatura modifikována a při nalezení chyb také opravována.

ZÁVĚR V ANGLIČTINĚ

At the conclusion, we can say that web technologies are constantly developing. Programmers have at their disposal better and more complex solutions and safer languages. The applications, which are realized in a more complicated way nowadays, can be realized much easier in the near future. This is supported by new standards, ideas and solutions originating in heads of web programmers and creators of web technologies. Even though modern technologies and principles were used during creation of applications, it is only a matter of time when such a new application becomes obsolete.

Due to this dissertation, there is a new possibility of security setting for operating systems GNU/Linux which lets users choose from further various applications. Since created and in the practical part described application LITS was programmed for the easiest possible operating, it is possible that it will be used by server administrators – beginners for set-up of their newly installed systems and it will reduce the percentage of virus spread between computers.

Since the attacks against information technologies are still more and more sophisticated and efficient today, it is necessary to keep the growth rate of rules and procedures of system and subsystem security at the same level. As the created and fully functional application LITS was programmed as an iptables upgrade, it is not guaranteed that this application shall not change standards of loading and saving of newly created rules during its development, and therefore becomes useless. Due to the created object-oriented code, we can fix this situation very soon.

To include the whole problematics of network technologies, GNU/Linux, or theory of security would require at least a thick book for each of these topics. However, the intention of this dissertation was to introduce these areas at the base level. In addition to acquaint with given problems was the main priority to create grafic upgrade for framework iptables. That way arose a very extensive and fully functional application called LITS (Linux Iptables Settings), that can be extend for free according to discretion of single users.

For in more detail understanding of this problem it is necessary to follow special literature all the time. Because how the development will go forward, then also this literature will be modified and corrected while finding of mistakes.

SEZNAM POUŽITÉ LITERATURY

[1] DEITEL, H. M.: *Operating Systems*, Prentice Hall, 2004.

[2] TANENBAUM, A. S.: *Modern operating systems*, Prentice Hall, 2002.

[3] Kol. Autorů: *Linux - Dokumentační projekt*, Praha: Computer Press, 2003. Dostupné na URL: <http://knihy.cpress.cz/DataFiles/Book/00000675/Download/K0819.pdf > [cit 2007-2-15].

[4] SOBELL, M., G.: *Linux-praktický průvodce*, Praha: Computer Press, 1999.

[5] NEMETH, E., SNYDER, G., HEIN, T. R.: *Linux - kompletní příručka administrátora*, Praha: Computer Press, 2004.

[6] *Linux*, Wikipedia.org [online], Dostupný z URL: <http://cs.wikipedia.org/wiki/Linux> [cit 2007-2-15].

[7] GUTMANS, A., BAKKEN, S. S., RETHANS, D.: *Mistrovství v PHP 5*, Praha: Computer Press, 2005.

[8] CASTAGNETTO, J., RAWAT, H., SCHUMANN, S., SCOLLO, CH., VELIATH, D.: *Programujeme PHP profesionálně*, Brno: Computer Press, 2004.

[9] YOUNG, M. J.: *XML krok za krokem*, Praha: Mobil Media, 2002.

[10] KOSTRHNOUT, A.: *Stavíme si malou síť*, Praha: Computer Press, 2001.

[11] DOSTALEK, L. a kol.: *Velký průvodce protokoly TCP/IP: Bezpečnost*, Praha: Computer Press, 2003

[12] *PHP,* The PHP Group [online], Dostupný na URL: < http://www.php.net/> [cit 2007-4-11]

[13] *PEAR,* The PHP Group [online], Dostupný na URL: < http://pear.php.net/> [cit 2007-4-11]

[14] *CentOS,* The Community ENTerprise Operating Systém [online], Dostupný na URL: < http://www.centos.org/> [cit 2007-4-28]

[15] *The World Wide Web Consortium (w3c),* Massachusetts Institute of Technology, Europen Research Consortium for Informatics and Mathematics, Keio University [online], Doustpný na URL: < http://www.w3.org/> [cit 2007-5-4]

[16] *Netfilter/iptables,* The netfilter webmaster [online], Dostupný na URL: <http://www.netfilter.org/> [cit 2007-5-5]

[17] *Iptables Firewall*, LinuxGuruz [online], Dostupný na URL: <http://www.linuxguruz.com/iptables/> [cit 2007-5-7]

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

BSD (*Berkeley Software Distribution*) – odvozenina UNIXU distribuovaná kalifornskou univerzitou.

CBQ (*Class-Based Queueing*) – jedná se o skript usnadňující nastavení šířky pásma.

CGI (*Common Gateway Interface*) – předpis pro program spouštěný na serveru, jak má číst data od www serveru a jak mu má data posílat.

CSS (*Cascading Style Sheets*) – jazyk pro formátování internetových stránek.

DHCP (*Dynamic Host Configuration Protocol*) - aplikační protokol z rodiny TCP/IP. Používá se pro automatické přidělování IP adres koncovým stanicím v síti.

DNAT (*Destination NAT*) – změna IP adres paketů procházejících zařízením, kdy se cílová IP adresa převádí mezi různými rozsahy.

DSCP (*Differentiated Services Code Point*) je pole v paketu IP umožňující přiřazení různých úrovní služeb síťovému provozu. Je to umožněno označením každého paketu v síti kódem DSCP a přiřazením odpovídající úrovně služby danému paketu.

FTP (*File Transfer Protocol*) je protokol aplikační vrstvy z rodiny TCP/IP, je určen pro přenos souborů mezi počítači, na kterých mohou běžet velmi rozdílné operační systémy.

FW (*Firewall*) - síťové zařízení, které slouží k řízení a zabezpečování síťového provozu mezi sítěmi s různou úrovní důvěryhodnosti a/nebo zabezpečení.

GNU (*GNU's Not Unix)* - projekt zaměřený na svobodný software. Původní cíl byl vyvinout operační systém se svobodnou licencí, který však neobsahuje žádný kód původního UNIXu.

GUI (*Graphical User Interface*) - druh komunikace s počítačem mající podobu interaktivních grafických prvků

ICMP *(Internet Control Message Protocol*) - jeden z jádrových protokolů ze sady protokolů internetu. Používají ho operační systémy počítačů v síti pro odesílání chybových zpráv – např. pro oznámení, že požadovaná služba není dostupná.

IDE (*Integrated Drive Electronics*) - obchodní název pro rozhraní *Advanced Technology Attachment* (ATA) pevných disků počítačů

IP *(Internet Protocol*) - datový protokol používaný pro přenos dat přes paketové sítě. Tvoří základní protokol dnešního Internetu.

ISO/OSI (*International Organization for Standardization* / *Open Systems Interconnection*) – jedná se o referenční model, jehož úlohou je poskytnout základnu pro vypracování norem pro účely propojování systémů.

LAN (*Local Area Network*) – ve volném překladu znamená 'místní síť", často se používá také termín 'lokální síť". Síť LAN lze vytvořit mezi minimálně dvěma počítači.

LITS (*Linux Iptables Settings)* – aplikace pro nastavení firewallu pro linux.

MAC (*media access control*) - jedinečný identifikátor síťového zařízení, který používají různé protokoly druhé (linkové) vrstvy OSI.

NAT (*Network address translation*) - je funkce síťového routeru pro změnu IP adres paketů procházejících zařízením, kdy se zdrojová nebo cílová IP adresa převádí mezi různými rozsahy.

OS (*Operační systém*) - je sada programů (software) umožňujících co nejefektivnější využití hardware počítače.

P2P (*Peer-to-peer*) -

PC (*Personal computer*) - je označení pro typ počítače určený pro použití jednotlivcem.

PEAR (*PHP Extension and Application Repositury*) – framework pro PHP.

PHP (*PHP Hypertext Preprocessor*) - skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek.

RPM (*RPM Package Manager*) - balíčkovací systém pro Linux původně vyvinutý firmou Red Hat pro Red Hat Linux, avšak v současné době se kromě mnoha distribucí linuxu využívá i v dalších operačních systémech.

SNAT (*Source NAT*) – změna IP adres paketů procházejících zařízením, kdy se zdrojová IP adresa převádí mezi různými rozsahy.

SSH (*Secure Shell*) - klient/server protokol v síti TCP/IP, který umožňuje bezpečnou komunikaci mezi dvěma počítači pomocí transparentního šifrování přenášených dat.

TCP (*Transmission Control Protocol*) - jedním ze základních protokolů sady protokolů Internetu, konkrétně představuje transportní vrstvu.

TOS (*Type Of Service*) - sada čtyř-bitových návěští (flagů) v IP hlavičce. Pokud je některý z těchto bitů nastaven, routery mohou zpracovávat tyto datagramy rozdílně oproti datagramům bez nastavených TOS bitů.

TTL (*Time to live*) - je parametr paketu protokolu IP, který zajišťuje nepřetížení sítě bloudícími pakety. Obsahuje číslo, které se při přechodu přes jednotlivé části sítě snižuje o 1. Pokud se číslo sníží až na 0, je paket smazán.

UDP (*User Datagram Protocol*) - tzv. *nespolehlivý* protokol ze sady protokolů internetu. UDP protokol přenáší datagramy mezi počítači v síti, ale na rozdíl od TCP nezaručuje, zda přenášený paket neztratí, nezmění pořadí paketů, nebo zda některý paket nedoručí vícekrát.

UNIX (*Unary Information and Computing Service*) - je víceúlohový a víceuživatelský operační systém, který je implementován na mnoha hardwarových platformách.

URL (*Uniform Resource Locator*) - řetězec znaků s definovanou strukturou a slouží k přesné specifikaci umístění zdrojů informací (ve smyslu dokument nebo služba) na Internetu.

UTF-8 (*UCS Transformation Format*) - způsob kódování řetězců znaků Unicode/UCS do sekvencí bajtů.

WWW (*World Wide Web*) - ve volném překladu „Celosvětová pavučina“, je označení pro aplikace internetového protokolu HTTP. Je tím myšlena soustava propojených hypertextových dokumentů.

XML (*eXtensible Markup Language*) - obecný značkovací jazyk, umožňuje snadné vytváření konkrétních značkovacích jazyků pro různé účely a typy dat.

SEZNAM OBRÁZKŮ

SEZNAM PŘÍLOH

PŘÍLOHA P I: NÁPOVĚDA A MANUÁL K APLIKACI LITS