VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

FILTROVANIE SIEŤOVEJ PREVÁDZKY

NETWORK TRAFFIC FILTERING

SAMOSTATNÝ PROJEKT
INDIVIDUAL PROJECT

AUTORI PRÁCE — **Bc. JOZEF URBANOVSKÝ**, **Bc. ADRIÁN TOMAŠOV**

AUTHORS

BRNO 2019

Obsah

Úvod

Táto práca je dokumentáciou k samostatnému projektu z predmetu *Aplikovaná kryptografie*. Samostatný projekt sa zaoberá filtrovaním sieťovej prevádzky v systéme GNU/Linux. Cieľom tohto projektu je programovo realizovať aplikáciu, ktorá má na starosť filtrovať zašifrovanú sieťovú prevádzku. Aplikácia má vytvárať štatistiky o type a množstve šifrovaných dát v sieti a ponúknuť možnosť si ich zobraziť v grafickej forme.

Kapitola 1

Analýza

V tejto kapitole je analyzovaná problematika a spôsob riešenia, ktorý je použitý pri tvorbe tejto programovej aplikácie. Dokumentácia predpokladá, že čitateľ pozná koncepty TCP/IP sieťového modelu a Linuxového jadra.

1.1 Firewall a filtrovanie

Firewall možno označiť ako ľubovoľné sieťové zariadenie, alebo sieťovú aplikáciu, ktorá slúži k riadeniu sieťovej prevádzky medzi logicky oddelenými sieťami. Podstata firewallu je v predefinovaných pravidlách pre jednotlivé pakety a toky na základe informácii z rôznych vrstiev ISO/OSI modelu. Firewally je možné rozdeliť do nasledujúcich kategórii na základe vývoja počítačových sietí [1].

- Paketový filter
- Aplikačný filter
- Stavový paketový filter
- Stavový paketový filter s kontrolou protokolov ľubovoľných vrstiev ISO/OSI modelu

1.2 Implementácia v Linuxovom jadre

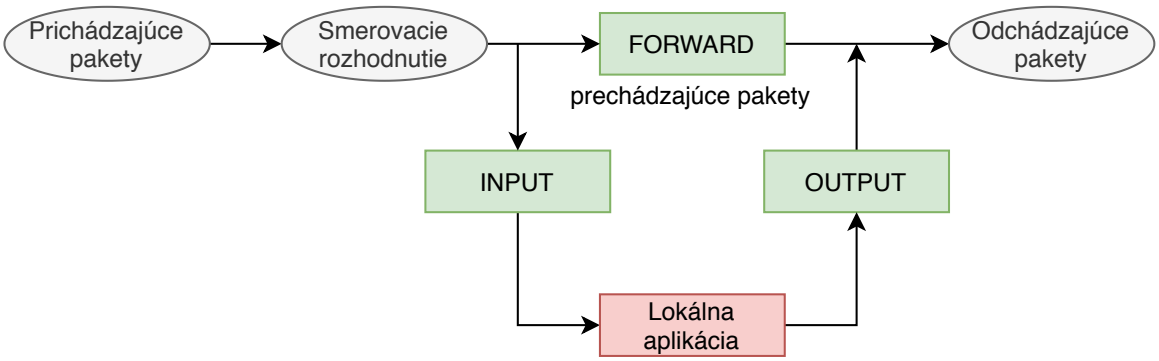
Moderné Linuxové jadro ponúka granulárnu kontrolu rôznych implementácii firewallu pre filtrovanie sieťových paketov. Práca rozoberá staršiu implementáciu filtrovania pomocou *iptables* a nástupcu vo forme *nftables* [2].

1.2.1 netfilter

netfilter je možné reprezentovať ako framework Linuxového jadra, slúžiacia pre filtrovanie paketov, preklad sieťových adries alebo preklad sieťových portov. Jeho hlavnú úlohu v jadre plnia hooky, ktoré dovolujú meniť správanie jadra ostatným modulom. Každý paket prechádzajúci kernel prejde cez sadu hookov, ktoré môže zaregistrovať predurčený modul jadra cez callback a následne zareagovať spustením obslužnej procedúry. Netfilter hooking systém obsahuje moduly jadra ako napríklad `ip_tables`, `ip6_tables`, `arp_tables` a `ebtables`, ktoré možno reprezentovať ako tabuľky pre definíciu pravidiel firewallu [3, 2]. Zjedodušené fungovanie smerovania v module *netfilter* je možné vidieť na obrázku 1.1.

1.2.2 iptables

Modul jadra `ip_tables` spolu s userspace programom *iptables* slúži na manipuláciu s tabuľkami *Xtables*, ktoré umožnujú združovať sady pravidiel do reťazcov. Reťazce následne definujú jednotlivé pravidlá pre pakety a sú spracovávané sekvenčne. Pravidlá umožnujú ovyplvňovať priechod sieťovým zásobníkom, kde každý paket musí prejsť aspoň jednou tabuľkou. [4, 3, 2]



Obr. 1.1: Základné reťazce *iptables* obsiahnuté v *netfilteri* v tabuľke filter

iptables ponúka konfiguráciu *netfilteru* ako stavového paketového filtra možného filtrovať sieťovú prevádzku na základe typu protokolu, zdrojovej a cieľovej adresy, zdrojového a cieľového portu, znalostí protokolu a ich stavoch.

Hlavným problémom *iptables* a dôvodom zlej reputácie, je primárne vysoká duplicita kódu, nakoľko existuje samostatná tabuľka pre každý sieťový protokol, problémy so škálovaním, rýchlosť spracovávania a mnohé iné.

1.2.3 nftables

Náhrada *iptables* vo forme *nftables* je podsystém v Linuxovom jadre, ktorý mení a nahradzuje určité časti samotného *netfilteru*. Základným blokom tohto podsystému je pridanie virtuálneho stroja do Linuxového jadra, ktorý je schopný spúšťať binárny kód určený na prezeranie sieťových paketov a rozhodovanie podľa pravidiel [2, 3].

nftables neobsaujú žiaden špecifický kód naviazaný na protokol a umožnujú analyzovať aj neznáme pakety, ktorých spracovanie je definované užívateľom cez userspace program *nft*. V prípade nutnosti rozšírenia samotného firewallu je teda nutné len vytvoriť nový binárny kód, ktorý je následne vložený do virtuálneho stroja na vykonávanie a mimo toho nie je potreba meniť žiadnu časť jadra.

Kapitola 2

Návrh aplikácie

V tejto kapitole sú popísané jednotlivé prvky firewall aplikácie, hlavne teda grafické prostredie pre správu a analýzu, ale aj jadro samotného firewallu. Celý projekt je vyvinutý v jazyku `Python3.6`, ktorý ponúka univerzálnosť, jednoduchosť a veľa možností, ktoré uľahčujú a zrýchľujú vývoj projektov.

2.1 Virtuálne prostredie

Celá aplikácia je vyvíjaná vo virtuálnom prostredí `pipenv`, aby bol zaručená zhoda verzií jednotlivých externých modulov na rôznych systémoch, hlavne teda počas vývoja a nasadenia aplikácie na strane servera. Všetky potrebné balíky su definované aj s konkrétnymi verziami v súbore `Pipfile`.

2.2 Grafické rozhranie

Táto aplikácia je navrhnutá, aby bežala ako služba na servery, ktorý slúži ako firewall. Jedným z cieľov tohto projektu je, aby aplikácia poskytovala štatistiky vo forme grafov. Aby sme sa vyhli inštalácií grafického prostredia na firewalle, ktoré je závislé na množstve rôznych balíkov, rozhodli sme sa vytvoriť užívateľské rozhranie pomocou webového frameworku `Django` (popisaný v sekcií 2.2.1). Cez tento web bude možné upravovať konfiguráciu firewallu a sledovať aj dynamicky generované grafy pomocou `HighCharts` (popísaný v sekcií 2.2.2).

2.2.1 Django

Webový framework `Django` je napísaný v jazyku `Python` a slúži na rýchly vývoj a návrh aplikácií. Hlavnou myšlienkou je nevyvíjať už existujúce veci a riešenia, ale zamerať sa hlavne na novú aplikáciu. V základe každého `Django` projektu nájdeme autentifikáciu užívateľov a abstrakciu nad rôznymi databázovými systémami. Tento framework používa poupravený `Jinja2` šablónovací systém pre generovanie html stránok [5].

2.2.2 HighCharts

Tento JavaScriptový modul slúži na generovanie veľkého množstva rôznych interaktívnych webových grafov ako napríklad čiarové, koláčové či pruhový graf. Práca s týmto modulom

je veľmi jednoduchá a spočíva vo vytvorení reťazca dát vo formáte `JSON`, ktorý obsahuje všetky potrebné informácie. O zvyšok sa uz modul postará sám [6].

2.3 Firewall

Tento firewall používa `nftables` na filtrovanie sieťovej trafiky. Konfigurácia môže byť zadaná priamo pomocou programu `nft` alebo pomocou webového grafického rozhrania, kde je textové pole s aktuálnou konfiguráciou, ktorú možno upravovať.

2.4 Štatistika

Ako zdroj dát pre generovanie grafov firewall používa čítače, ktoré sú pridané priamo v `nftables` konfigurácií. Používa aj niektoré systémové nástroje, ktoré túto štatistiku zbierajú. Tieto dáta sa načítajú a spracujú pre každú požiadavku na webový server, ktorý vygeneruje príslušné `HighCharts` grafy.

Kapitola 3

Implementácia aplikácie

Realizovaná aplikácia implementačne zodpovedá jej predloženému návrhu. Grafické rozhranie vo forme webovej aplikácie, ktoré beží ako služba na serveri, je využité na reprezentáciu dát filtrovanej prevázdky a nastavenie samotných pravidiel firewallu.

Firewall je naimplementovaný pomocou podsystému jadra známeho pod menom *Netfilter*. Jeho súčasťou je aj nová implementácia filtrovania prevádzky vo forme *nftables*, ktoré sú ovládané cez userspace program `nft`. Webové rozhranie ponúka užívateľovi možnosť upravovať konfiguračný súbor pravidiel pre firewall, zálohovať ho a opätovne nahrať.

Samotný základný konfiguračný súbor detekuje šifrovanú prevádzku na základe hlavičiek tretej a štvrtej vrstvy ISO OSI modelu, prípadne cieľových a zdrojových portov štvrtej vrstvy. Vdaka tomu, že užívateľ má plnú kontrolu nad konfiguračným súborom *nftables*, je možné aby filtroval čokoľvek podľa pravidiel userspace programu [7].

3.1 Testovanie a demo

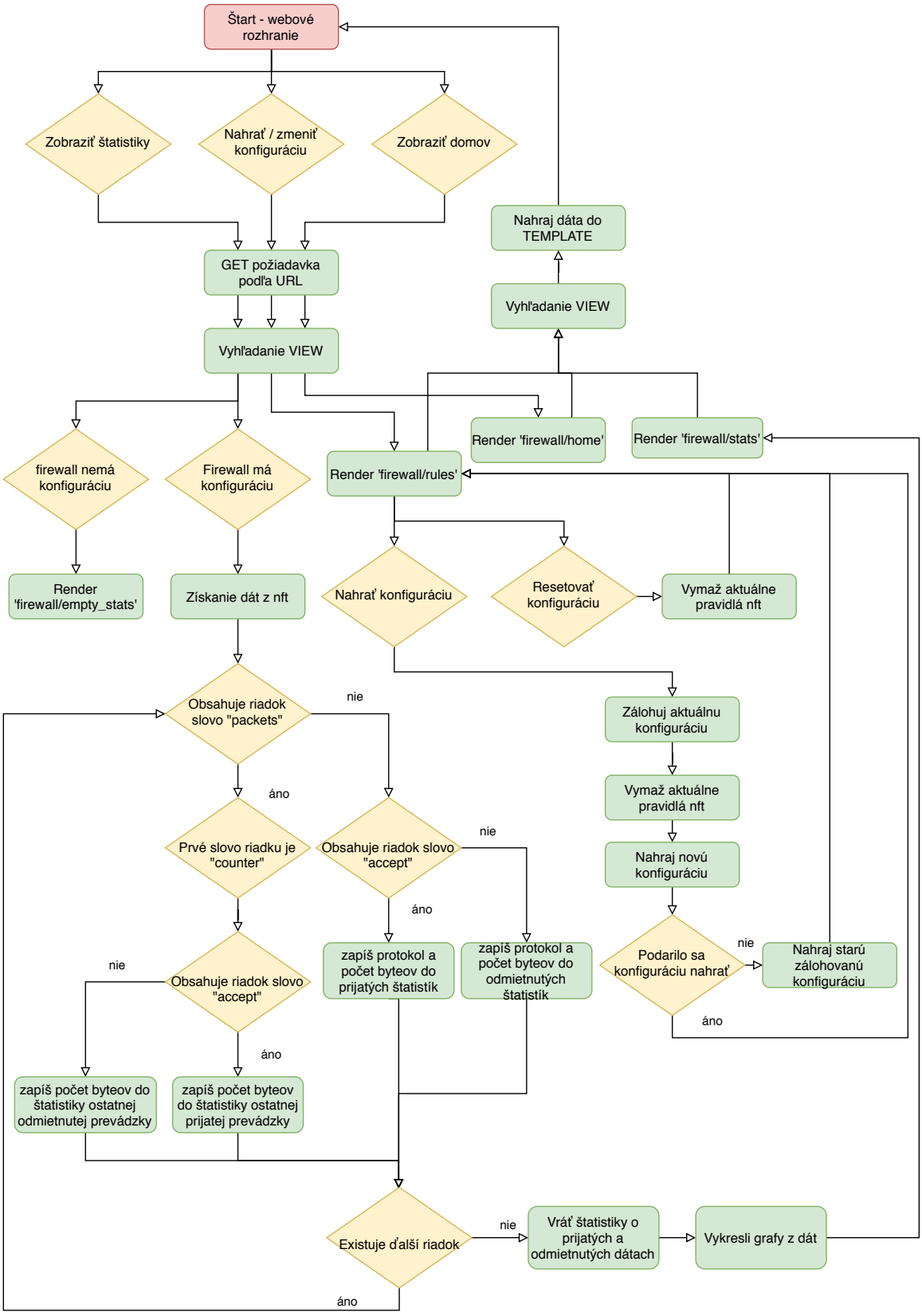
K aplikácií bolo aj kvôli unifikácii testovacieho prostredia a možnosti prezentácie dopísané aj virtuálne prostredie s plnou automatizáciou predvedenia aplikácie. Tento cieľ bol dosiahnutý za pomoci virtuálnych prostredí v *Libvirte*, ktoré sú nakonfigurované cez *Vagrant*.

Pre virtuálne systémy bola použitá distribúcia *CentOS 8*, ktorá bola dostupná ako najbližšia alternatíva k Fedore 30 a 31, kde bola aplikácia vyvíjaná. Prvý virtuálny systém, nazvaný ako *generator*, slúži primárne pre naviazanie IPsec tunelu a generáciu prevádzky, ktorá ukazuje funkcionalitu aplikácie. Druhý virtuálny systém s názvom *firewall*, obsahuje už samotnú instanciu aplikácie v celej forme s webovým serverom. Tento systém taktiež obsahuje druhú časť IPsec tunelu a lokálny webový server presmerovaný na port 8000. Oba systémy obsahujú ich funkcionalitu v službách a inicializácia samotného firewallu v nft je zabezpečená nahraním základnej konfigurácie, ktorá obsahuje povolenie všetkej prevádzky, ktorá je sledovaná a rozdelená podľa typu známej použitej enkrypcie a jej typu. Zjednodušený vývojový diagram s istou mierou abstrakcie je možné vidieť na obrázku 3.1.

3.1.1 Inštalácia

Pre jednoduchosť overenia funkcionality stačí k spusteniu aplikácie so automatickým demom nainštalovať *Vagrant* a spustiť jeho `Vagrantfile`. Zbytok je zautomatizovaná inštalácia a pre overenie implementácie a funkčnosti stačí navštíviť lokálny webový server na adrese `localhost:8000`.

1. Nainštalovať Vagrant [8], možné použiť systémové "repo"v ktorom sa zvyčajne nachádza
2. Spustiť a nainštalovať predpripravené virtuálne systémy Vagrantu - `vagrant up`
3. V lokálnom internetovom prehliadači zadať adresu `localhost:8000`



Obr. 3.1: Zjednodušený vývojový diagram na základné použitie aplikácie.

Záver

Návrh aplikácie pre konfiguráciu firewallu a jej implementácia v systéme Linux je vyriešená a vďaka virtuálnemu prostrediu je schopná fungovať na rôznych Linuxových distribúciach.

Aplikácia bola naimplementovaná za pomoci userspace programov Linuxového systému za použitia webové rozhrania na ovládanie. Grafické webové rozhranie spĺňa zadanie pre konfiguráciu firewallu a filtráciu zašifrovanej prevádzky. Umožnuje užívateľovi ľubovoľne konfigurovať Linuxový firewall vo forme frameworku novšej časti *Netfilteru* - *nftables*. Aplikácia v reálnom čase sleduje prevádzku na sieti a dynamicky zobrazuje tieto údaje do grafu.

3.2 Rozdelenie práce

- Bc. Jozef Urbanovský <xurban66>
  - Frontend - Bootstrap, HTML
  - Backend - firewall, iptables, nftables
  - Automatizácia - Shell skripty
  - Dokumentácia
- Bc. Adrián Tomašov <xtomas32>
  - Frontend - Django, HighCharts, Bootstrap, HTML
  - Backend - systémové programy, nftables
  - Automatizácia - Vagrant, Shell skripty
  - Dokumentácia

Literatúra

[1] Oppliger, R.: Internet security: firewalls and beyond. *Communications of the ACM*, ročník 40, č. 5, Květen 1997: s. 92–102, doi:10.1145/253769.253802.
URL `https://doi.org/10.1145/253769.253802`

[2] Linux manual pages. [Online; navštívené 11.3.2020].
URL `https://www.die.net/`

[3] Harald Welte, P. N. A.: The netfilter.org project. [Online; navštívené 11.3.2020].
URL `https://netfilter.org`

[4] Dočekal, M.: Správa linuxového serveru: Linuxový firewall, základy iptables. [Online; navštívené 11.3.2020].
URL `https://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru-linuxovy-firewall-zaklady-iptables`

[5] Foundation, T. D. S.: Django project. [Online; navštívené 11.3.2020].
URL `https://www.djangoproject.com/`

[6] Highsoft: HighCharts. [Online; navštívené 11.3.2020].
URL `https://www.highcharts.com/`

[7] Netfilter: nftables wiki. [Online; navštívené 17.4.2020].
URL `https://wiki.nftables.org`

[8] HashiCorp: Vagrant. [Online; navštívené 17.4.2020].
URL `https://vagrantup.com/docs/installation`