SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta chemickej a potravinárskej technológie

Evidenčné číslo: FCHPT-5414-39583

Implementácia proxy riešenia ústavu

Diplomová práca

2013 **Bc. Branislav Mitterpach**

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta chemickej a potravinárskej technológie

Evidenčné číslo: FCHPT-5414-39583

Implementácia proxy riešenia ústavu

Diplomová práca

Študijný program: automatizácia a informatizácia v chémii a potravinárstve
Číslo študijného odboru: 2621
Názov študijného odboru: 5.2.14 automatizácia
Školiace pracovisko: Ústav informatizácie, automatizácie a matematiky
Vedúci záverečnej práce: prof. Ing. Miroslav Fikar, DrSc

Bratislava 2013 **Bc. Branislav Mitterpach**

Slovenská technická univerzita v Bratislave
Ústav informatizácie, automatizácie a matematiky

Fakulta chemickej a potravinárskej technológie
2012/2013

ZADANIE DIPLOMOVEJ PRÁCE

Evidenčné číslo:	FCHPT-5414-39583
ID študenta:	39583
Autor práce:	Bc. Branislav Mitterpach (39583)
Študijný program:	automatizácia a informatizácia v chémii a potravinárstve
Študijný odbor:	5.2.14 automatizácia
Vedúci práce:	prof. Ing. Miroslav Fikar, DrSc.
Miesto vypracovania:	ÚIAM FCHPT STU v Bratislave

Názov témy: **Implementácia proxy riešenia ústavu**

Rozsah práce: 60

Špecifikácia zadania:

Implementácia proxy a VPN servera na sieti ústavu. Táto bude založená na existujúcej infraštruktúre routera s distribúciou pfSense. Diplomant vytvorí testujúcu infraštruktúru v analogickými vlastnosťami, ako skutočná sieť ústavu, na ktorej implementuje nasledovné úlohy:
- inštalácia a konfigurácia router pfSense
- konfigurácia oddelených VPN pre zamestnancov a študentov
- vypracovanie literárneho prehľadu existujúcich proxy riešení
- implementácia proxy riešenia nad pfSense, vytvorenie riešenia pre správu reportovania na ústave
- presun na skutočnú sieť ústavu
- overenie funkčnosti navrhnutého riešenia

Zoznam odbornej literatúry:

1. STREBE, M. -- PERKINS, C. *Firewally a proxy - servery : Praktický průvodce.* Praha: Computer Press, 2003. 450 s. ISBN 80-7226-983-6.
2. ZHOU, W. -- MA, Z. Semantic Filtering and Content Recurrence for Web Page Accessing Based on Proxy. *Computer Technology and Development Vol. 17 No. 4.* ISSN 1673-629X.

Dátum zadania diplomovej práce: **18. 02. 2013**
Termín odovzdania diplomovej práce: **25. 05. 2013**

Bc. Branislav Mitterpach
študent

prof. Ing. Miroslav Fikar, DrSc.
vedúci pracoviska

prof. Ing. Miroslav Fikar, DrSc.
garant študijného programu

Čestné prehlásenie

Čestne prehlasujem že diplomovú prácu som vypracoval samostatne, na základe zdrojov uvedených v literatúre, pod vedením vedúceho diplomovej práce a s pomocou vedomostí nadobudnutých počas štúdia.

.....

Branislav Mitterpach

Poďakovanie

Moje najväčšie poďakovanie patrí prof. Ing. Miroslavi Fikarovi, DrSc. za jeho jedinečný prístup ktorého sa mi dostalo počas celého vypracovávania tejto práce, za nasmerovanie v ťažkých okamihoch, za ústretový prístup, za mimoriadne férové jednanie a vôbec mimoriadne vysoký profesionálny štandard počas celej spolupráce. Ďalej moje poďakovanie patrí: Ing. Františkovi Benovičovi za vzájomné posilňovanie počas paralelnej práce na publikáciách a Bc Ľubošovi Mičencovi za motivujúce rozhovory a dobrý pocit že je na tom niekto momentálne horšie.

Zhrnutie

Diplomová práca sa zameriava na implementáciu proxy riešenia na sieti ústavu. Samotná implementácia zahŕňa inštaláciu VPN servera, proxy servera a vytvorenie systému reportovania. Pre inštaláciu VPN servera bola vybraná aplikácia OpenVPN, pre inštaláciu proxy servera aplikácia Squid. Obe aplikácie boli inštalované, konfigurované a testované na distribúcii pfSense, ktorá predstavovala činnosť hlavného bodu vo vytvorenom testovacom prostredí predstavujúcom časť skutočnej siete ústavu. Správa reportovania bola riešená vlastným programom v jazyku perl, ktorý používa ako zdroj dát informácie získané činnosťou proxy servera. Vytvorená sústava bola presunutá na skutočnú sieť ústavu, kde sa overila jej funkčnosť.

Kľúčové slová : Proxy server, OpenVPN, pfSense

Abstract

Diploma thesis deals with implementation of proxy solution on department network. Implementation itself includes installation of VPN server, proxy server, and creation of a reporting system. Project OpenVPN was chosen for VPN server, project Squid was selected for installation of proxy server. Both application were installed, configured, and tested on distribution pfSense which represents functionality of main server in created test environment which simulates a part of department network infrastructure. Administration of reporting system was handled with own program in perl language, which uses source data provided by proxy server. Created system was moved to real department network, where its functionality was tested.

Key words: Proxy server, OpenVPN, pfSense

Obsah

1 Úvod

Sieť a elektronická sieťová komunikácia sa v dnešnom období stala nutnou súčasťou existencie každej organizácie. Správa siete zahŕňa okrem vhodného návrhu, aj správu a neustály rozvoj v rámci prispôsobovania sa potrebám organizácie. S pribúdajúcim časom oblasť technologických aspektov komunikačného odvetvia prudko narastá a zažíva vývoj, ktorý bol silne badateľný ako v histórii uplynulých desiatok rokov, tak aj v súčastnosti. Pre prispôsobenie tomuto trendu je potrebné inovovať a nasadzovať nové metódy zabezpečenia a reportovania o činnosti siete. Tiež je potrebné rozširovať aktuálne služby a nasadzovať nové v súlade s potrebami organizácie a všeobecne zaužívanými službami v rámci globálnej úrovne. Takýmto prístupom možno predísť degradácii alebo stagnácii informačnej štruktúry organizácie a vyhnúť sa ohrozeniam, ktoré z takéhoto stavu plynú. Kritickým aspektom pri hodnotení informačnej infraštruktúry je bezpečnosť a služby s ňou súvisiace. V rámci vnútornej infraštruktúry je väčšinou bezpečnosť aplikovaná rôznymi spôsobmi, aby zabránila nežiadúcim prístupom na dôležité miesta, ktoré sú dostupné z vnútra organizácie. Tento druh politiky však tiež zabraňuje prístupu zvonka pre užívateľov, ktorý by ho potrebovali. Tento problém možno vyriešiť implementáciou súkromných virtuálnych sietí, kedy možno vhodnou voľbou súčastí zabezpečiť šifrované pripojenie z vonkajšieho priestoru, pri ktorom možno garantovať autorizáciu a autentifikáciu užívateľov. Vhodnou voľbou sa javí celosvetovo najrozšírenejší open-source projekt OpenVPN. Projekt sa svojou širokou používateľskou základňou teší veľkej podpore na mnohých platformách a systémoch ako napríklad pfSense, ktorá je už úspešne zavedená v produkčnom prostredí organizácie, ktorej sa táto práca týka. Ďalším kritickým aspektom informačnej infraštruktúry je sledovanie a vyhodnocovanie činnosti a sieťových prenosov koncových bodov, systémov alebo užívateľov v rámci spravovanej siete. Monitorovanie činnosti potrebuje každá organizácia či už z administratívnych, technických alebo právnych dôvodov. Dobre aplikovaným monitoringom činnosti a reportovaní o nej možno včas odhaliť škodlivú činnosť, infiltrácie alebo iné hrozby a včas proti nim aktívne zasiahnuť. Podľa zvolenej formy monitoringu sa javí najvhodnejšie implementovať proxy server a vyžadovaným spôsobom z neho spracovávať záznamy, pričom je cieľom vytvoriť report vhodný k analýze. Zvoleným proxy serverom pre implementáciu je Squid, ďalší open-source projekt, ktorý ponúka množstvo nastavení a spôsobov konfigurácie, čím plní podmienky zmiešaného reportovania.

2 Implementácia VPN servera

2.1 Prenos dátového toku a šifrovanie dát

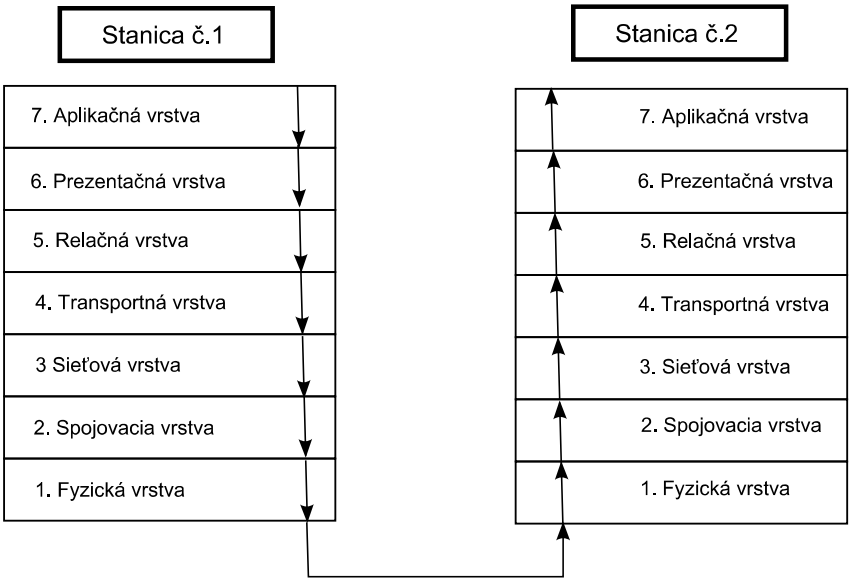
2.1.1 Princípy prenosu dát

2.1.1.1 Teória prenosu dát

Opis súčasnej technologickej úrovne siete je veľmi komplexná úloha vzhľadom na rôznorodú úroveň jej realizácie, zložitú a dlhú cestu, ktorú táto technológia prekonala. Princípom ostáva prenos informácie od jednej bodu v sieti k druhému. Prenos dátovej jednotky po tejto ceste tvorí mnoho medzi krokov a prechodov cez rôzne úrovne. Proces tejto cesty tvorí sám o sebe zložitú činnosť. Vhodnou pomôckou sa javí OSI model sieťovej komunikácie, na obrázku číslo 1, ktorý sa stal všeobecne používaným opisom pri prenose dát medzi dvomi stanicami.

Obrázok č. 1: OSI model

OSI model vykresľuje priebeh informácie od jej vzniku, v aplikačnej vrstve na základe užívateľských vstupov, cez jednotlivé vrstvy. Na jednotlivých vrstvách sa postupne transformuje a naberá informácie kvôli celkovému prenosu, pre potreby jej výmeny medzi dvoma stanicami. Na poslednej fyzickej vrstve dochádza k prenosu informácií pomocou fyzického média. V druhej pracovnej stanici dochádza k spätnej transformácii dát cez jednotlivé úrovne na vyžadovanú informáciu [1]. Posledný krok sa realizuje na druhej stanici v aplikačnej úrovni, kde je informácia distribuovaná užívateľovi tak ako to zobrazuje obrázok č. 2:.



Obrázok č. 2: Priebeh transformácie dát medzi stanicami

OSI model (Open Systems Interconnection model) používa sedem vrstiev (Layers), pričom najvrchnejšia je označená ako vrstva č. 7. Význam jednotlivých vrstiev je nasledovný:

1. **Fyzická vrstva** - Opisujú ju elektrické, mechanické, procedurálne a funkcionálne špecifikácie pre aktiváciu, správu a deaktiváciu fyzického spojenia medzi komunikujúcimi stanicami. Charakteristikami, ktoré ju definujú, sú úrovne napätia, frekvencia, fyzická rýchlosť komunikácie, maximálna dĺžka prenosového média a druhy fyzických konektorov. Typickou jednotkou tejto vrstvy je bit.

2. **Linková vrstva** - (spojovacia) Táto vrstva zodpovedá za bezchybný prenos informácií medzi dvomi bezprostredne spojenými stanicami v sieti. Túto vrstvu definujú rôzne druhy sietí a protokolov, ktoré využíva, zahŕňajúc fyzické adresovanie,(pomocou MAC adries), kontrola chýb prenosu, kontrola prenosu dátových rámcov (jednotky prenosu na tejto úrovni, z angl. Frames), oznamovanie chýb prenosu vyšším vrstvám.

3. **Sieťová vrstva -** Táto vrstva zodpovedá za bezchybný prenos dát v rámci celej siete medzi dvoma stanicami, úloha tejto vrstvy je veľmi významná. Táto vrstva využíva na smerovanie dát sieťové adresy, ktoré sa líšia od fyzických adries, pretože nie sú viazané na konkrétny hardware. Táto vrstva je zodpovedná za výber optimálnej trasy, ktorou dáta budú putovať, zo všetkých možných prepojení medzi dvoma stanicami. (Mechanizmus práce na tejto

úrovni bude vysvetlený neskôr). Typickou jednotkou tejto vrstvy je paket alebo balíček (z angl. packet)

4. **Transportná vrstva** - Táto vrstva zodpovedá za bezchybné doručenie dát od relačnej vrstvy, za doručenie v správnom poradí, za správne rozdeľovanie toku dát do segmentov a väčšinou aj za kontrolu prietoku dát, čím zabezpečuje, aby nižším vrstvám neboli dodávané dáta rýchlejšie, než dokážu reálne spracovať. Táto vrstva zároveň spracováva dáta od rôznych aplikácií naraz a zodpovedá za ich posunutie na doručenie nižším vrstvám. Transportná vrstva používa množstvo algoritmov na overenie správnosti doručenia dát, najznámejšími protokolmi tejto úrovne sú TCP a UDP.

5. **Relačná vrstva** – Táto vrstva zodpovedá za vytvorenie, správu, udržiavanie a ukončenie komunikačnej relácie, relácia nastáva na základe požiadavky a odpovede aplikácie lokalizovanej na komunikujúcich staniciach. Za reláciu možno považovať sumu rôznych spojení rôznych protokolov, ktoré sú vyžiadané aplikačnou úrovňou.

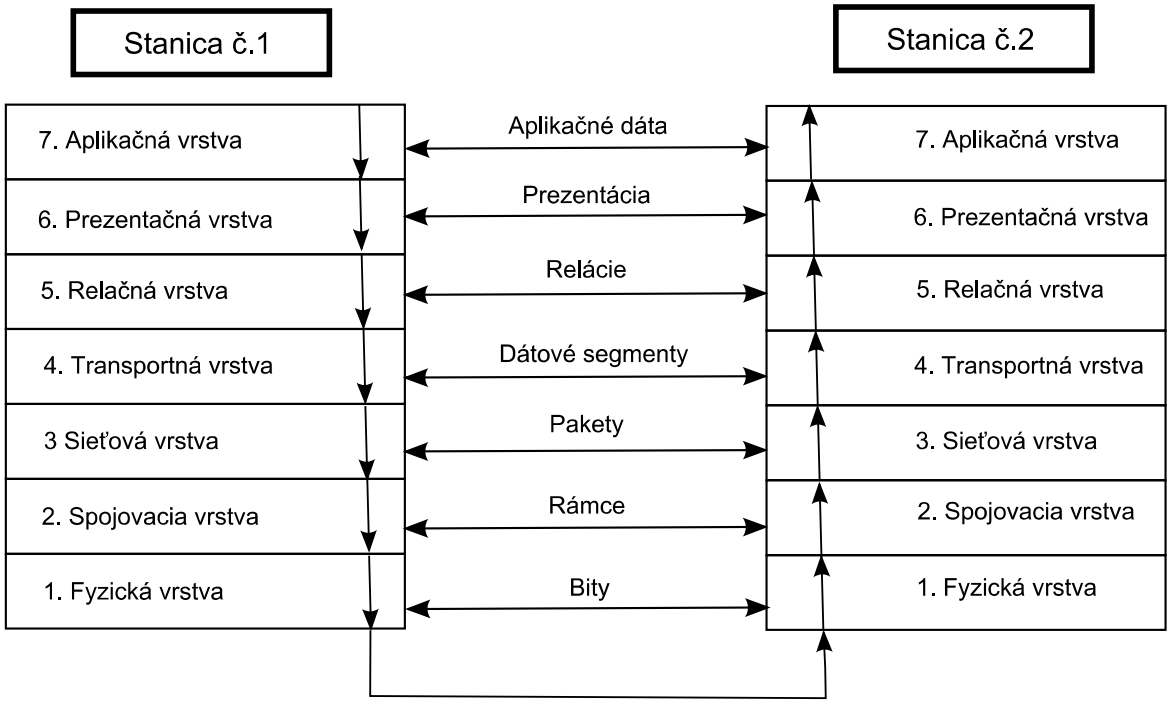
6. **Prezentačná vrstva** – Táto vrstva zodpovedá za korektnú úpravu dát tak, aby im rozumela druhá stanica. Stanice niekedy nepracujú na rovnakých platformách operačných systémov a spôsob, akým aplikácia spracováva dáta na jednej stanici, nemusí byť totožný ako na druhej stanici. Úlohou prezentačnej vrstvy je práve správna transformácia týchto dát.

7. **Aplikačná vrstva** – Táto vrstva zodpovedá za komunikáciu s aplikáciou, ktorá je zdrojom prenášaného komponentu, teda konkrétnych dát, ktoré sú dôvodom celého sieťového prenosu. Táto vrstva má najbližšie k užívateľovi, ktorý zadáva vstupy do aplikácie. Konkrétne má táto vrstva na starosti identifikáciu zdroja dát, synchronizáciu komunikácie a identifikáciu komunikačného partnera, pri identifikácii partnera táto vrstva zisťuje jeho dostupnosť mysliac tým maximálnu rýchlosť výmeny dát a rýchlosť ich spracovávania [1], [2],[3].

2.1.1.2 Zapuzdrenie dát

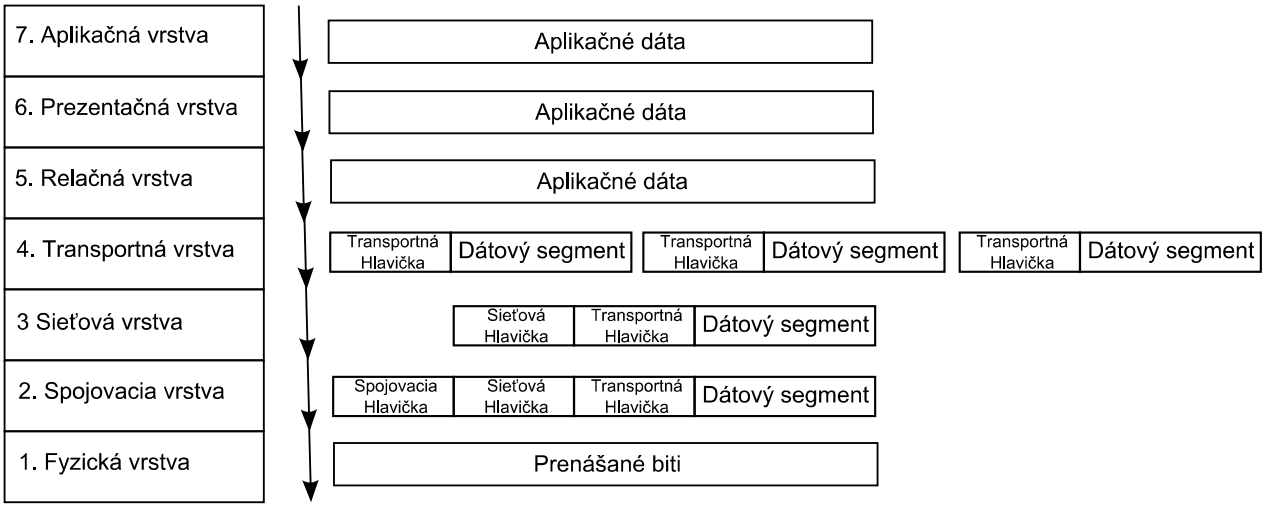
Počas prestupu jednotlivými vrstvami dáta menia svoju formu až po ich prenos do cieľovej stanice, kde sa transformujú naspäť. Pochopenie vrstiev OSI modelu dáva možnosť rozdeliť model

na aplikačnú a prenosovú vrstvu, zatiaľ čo posledné tri vrstvy sú často rôzne spájané. Činnosť prvých štyroch sa dá presne definovať, prvé tri vrstvy majú za cieľ vytvoriť dáta určené na prenos rôznymi transformáciami obsahu vzhľadom na aplikačné potreby. Akonáhle sa dáta dostanú na transportnú vrstvu, tvoria už nemenný celok. Dáta samotné sú usporiadaný reťazec bitov, ktorý v správnom poradí dáva zmysel a vytvára tak informáciu pre prenos napríklad textový reťazec, binárnu inštrukciu, mapu znakov. Po dosiahnutí transportnej vrstvy sa formát týchto dát už viac nemení a dospeli sme k tomu, čo možno nazvať platený náklad (z angl. PAYLOAD), pretože tieto dáta tvoria svojou veľkosťou skutočný prenášaný objem. Ak chce transportná vrstva poslať dáta ďalej, použije na to nižšie vrstvy, ešte predtým však dochádza k jemnej zmene dát, ktorá je základným princípom prenosu po vrstvách a nazýva sa zapuzdrenie. Pri opätovnom pohľade na OSI model je zapuzdrenie veľmi praktickým spôsobom, každá vrstva totiž komunikuje s rovnakou vrstvou na druhej strane pomocou vlastného formátu dát, zobrazeného na obrázku č.3 .



Obrázok č. 3: Jednotky komunikácie vrstiev OSI modelu

V praxi každá vrstva, počínajúc štvrtou, obohatí aplikačné dáta vzniknuté na základe užívateľských vstupov, o vlastnú hlavičku a niekedy aj záhlavie ako je tomu na obrázku č.4. Nižšia vrstva, ktorej sú takéto obohatené dáta zaslané, prijme celú túto množinu dát a vníma ju ako vlastný platený obsah a tiež pridá svoju hlavičku a pätu, viď obrázok č.4.



Obrázok č 4: Zapuzdrenie priebehom vrstiev OSI modelu

Takto to nasleduje až po poslednú vrstvu a následný prenos: Na druhej strane, spätným stúpaním dát od najnižšej vrstvy po najvyššiu, každá vrstva zoberie prijaté dáta, na základe hlavičky a päty vykoná operácie potrebné pre korekciu a overenie integrity dát, hlavičku a pätu aktuálnej sieťovej vrstvy zahodí a posunie zvyšok vyššej úrovni, ktorá v takto upravených dátach operáciu opakuje a to znovu a znovu až po doručenie dát aplikačnej vrstve a ich konečné použitie [1],[2].

2.1.1.3 Smerovania dát

Pri aplikácií uvedených informácií vyplýva, že prenosovou stanicou v sieti môže byť každá stanica a komunikácia sa zdá veľmi jednoduchá. Pri uvedených princípoch technológie boli zanedbané reálne problémy ako limitovaná dĺžka prenosového média, (elektrický alebo optický kábel postupne tlmí signál). V sieti sa preto musia vyskytovať viaceré zariadenia, aj na trase bez potreby vetvenia. Nie všetky zariadenia v sieti určené na prenos sú užívateľské stanice, väčšinu tvoria zariadenia operujúce na druhej alebo tretej vrstve OSI modelu, a to sú:

1. **Opakovač** – (z angl. Repeater), toto zariadenie má za účel len zopakovať prijatý signál zo vstupu na výstup, opakovačmi sa riešia problémy pri veľkých dĺžkach káblov.
2. **Rozbočovač** – (z angl. Hub), zariadenie ktoré má rovnaký význam ako opakovač, ale má viacero vstupov, pričom vždy keď prijme dáta z jedného vstupno-výstupného portu, tak ich pošle na všetky ostatné a nestará sa o smerovanie
3. **Brána** – (z angl. Bridge), ide o pomerne inteligentné zariadenie, ktoré spravuje určitú jemu

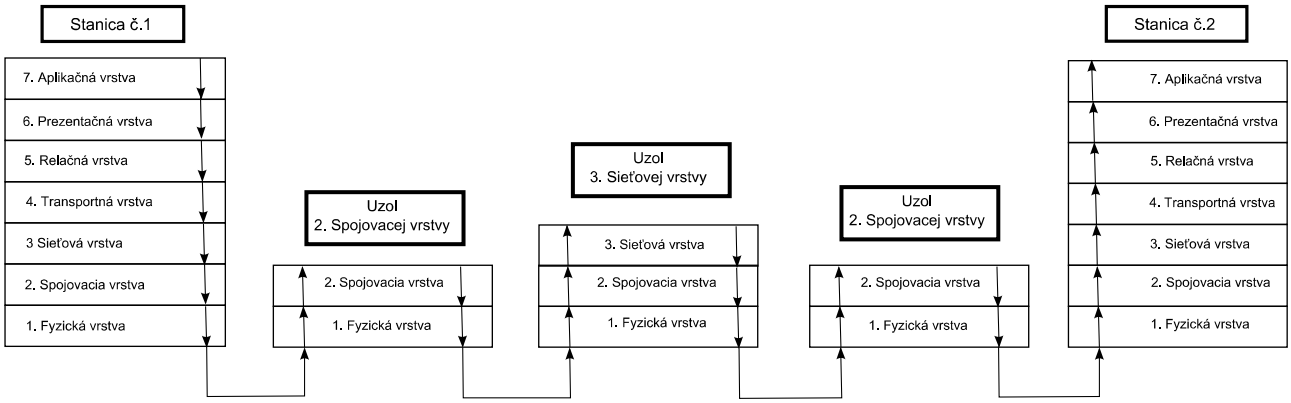
definovanú časť siete na druhej, teda spojovacej, vrstve. Brána je zariadenie, ktoré má tiež veľa fyzických portov. Pokiaľ obdrží dátový rámec tak na základe adresovania v druhej vrstve, pomocou MAC adries, zistí či ho môže poslať niekam vo svojej časti podsiete alebo nie. V prípade nenájdenia adresáta vo svojej časti siete ho pošle na predvolenú bránu, čo je výstup, pri ktorom brána, ako zariadenie, predpokladá že všetko, čo nenájde vo svojej časti, sa nachádza za ňou. Brána predpokladá, že ďalšie zariadenia za ňou sa nachádzajúce, si s dátovým rámcom budú vedieť poradiť.

4. **Smerovač** - (z angl. Router), ide o zariadenie ktoré svojou podstatou, funkciou a mechanizmom tvorí základný kameň v procese transferu dát cez sieť. Smerovač funguje na tretej vrstve OSI modelu pracuje na základe adresovania IP adries (z ang. internet protokol). Princíp smerovača tkvie v tom, že pozná všetky siete priamo naň pripojené. Všetky údaje má zapísané v smerovacej tabuľke, pomocou IP adresy a masky podsiete. Tým má jasne definované, aký rozsah IP adries sa môže nachádzať za každým jeho portom, na ktorý odosiela prijaté informácie.

Pre vysvetlenie konceptu smerovania treba poznať pojmy IP adresu, maska podsiete, predvolená brána, broadcast (vysielanie) a adresa siete :

- **IP adresa** – (protokolu Ipv4) je 32 bitov dlhý reťazec bitov, ktorý v každej sieti jednoznačne identifikuje jednu stanicu a nemôže byť duplicitný, väčšinou sa označuje v dekadickom formáte ako číslo ktoré tvoria 4 trojčíslia - 8 bitové skupiny, oddelené bodkou. (Napr 192.169.45.234)
- **Maska podsiete** – Ide tiež o 32 bitov dlhý reťazec, ktorý sa uvádza v rovnakej forme ako IP adresa, je zložený z reťazca samých jednotiek, ktorý sa v určitej časti mení na samé nuly. Maska podsiete udáva, ktorá čas siete je hostiteľská a ktorá sieťová, teda ktoré stanice v sieti môžu v sieti navzájom komunikovať bez toho, aby prekročili bránu siete.
- **Predvolená brána** – Ide o (väčšinou prvú) IP adresu v podsieti, na túto IP adresu sa odvolávajú stanice v sieti pokiaľ chcú komunikovať mimo podsiete
- **Broadcast** – (Vysielanie) Ide o najvyššiu možnú adresu s adresného rozsahu podsiete, jej špeciálnym významom je kontaktovať všetky stanice naraz, používa sa na vysielanie sieťových systémových správ, správy od nej počúvajú všetky stanice v sieti.
- **Adresa siete** – IP adresa ktorá končí nulou alebo viacerými nulami, napríklad 192.168.1.0, adresa siete udáva vetvu siete, na ktorej sa nachádzajú stanice.

Komunikácia zariadení v sieti v rámci OSI modelu sa dá zobraziť nasledovne na obrázku č. 5:



Obrázok č.5: Tok dáta sieťovými uzlami nižších vrstiev

Smerovacie a spojovacie zariadenia v sieti dáta spracúvajú len v rámci prvých troch vrstiev alebo dvoch vrstiev. Pri prechode každou stanicou sú zahodené dáta ako hlavička a záhlavie prichádzajúceho rámca, vyhodnotí sa kam sa zo stanice chce dostať, dostáva novú hlavičku a záhlavie a je zaslaný na ďalší bod. Hlavička a záhlavie danej vrstvy sa menia počas prechodu po celej sieti, nakoľko potrebné dáta nie sú pozmenené, len sa neustále prebaľujú podľa potreby smerovania. Na zariadeniach spojovacej vrstvy druhej vrstvy OSI modelu sa dáta smerujú na základe MAC adresy, takže adresovanie podlieha algoritmu podobnému pri zariadení brána. Na tretej vrstve sú dáta smerované pomerne zložitejšie, pre tento koncept treba najprv poznať základy adresného priestoru v tretej vrstve, teda IP adries (protokolu IPv4).
Teoretický rozsah IP adries je teda od 0.0.0.0 do 255.255.255.255, pre vhodnejšiu identifikáciu však boli rozdelené do skupín:

1. **Trieda A 1.x.x.x až 126.x.x.x**
2. **Trieda B 128.0.x.x až 191.254.x.x**
3. **Trieda C 192.0.0.x až 223.254.254.x**
4. **Trieda D 224.0.0.x až 239.254.254.254**
5. **Trieda E 240.0.0.0 až 254.255.255.254**

V tomto rozdelení je známa časť adresy sieť a zvyšná neznáma časť adresa stanice. Trieda A teda môže mať 127 sietí a 16 miliónov staníc na každej, trieda B môže mať 65 000 staníc na každej zo

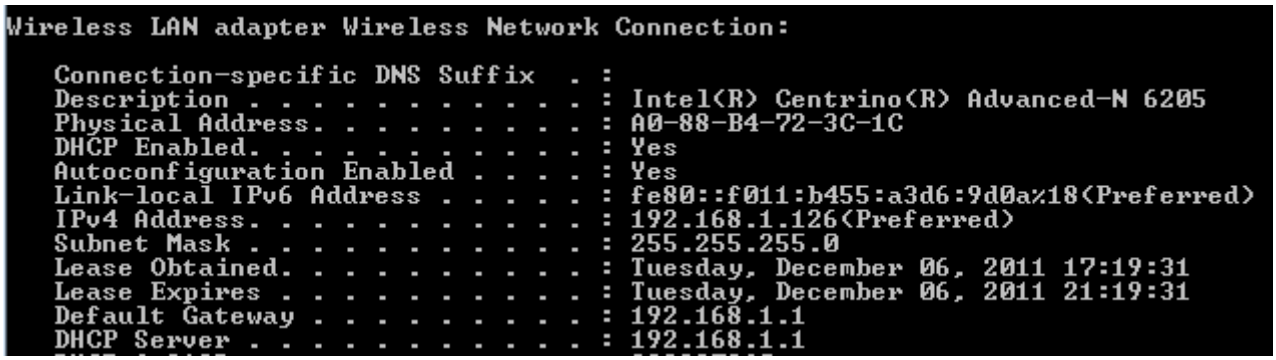
16 000 sietí, trieda C 254 staníc na dvoch miliónoch sietí. Trieda D sa používa na rôzne multicastové vysielania, trieda E ostala rezervovaná pre budúce použitie. Napriek tom tu sú ešte štyri špeciálne rozsahy sietí a dve špeciálne IP adresy:

- **127.x.x.x** – Tento rozsah sa používa na diagnostické účely a väčšinou celý rozsah patrí slučke lokálnej stanice, všetky pokusy o naviazania spojenia stanice s touto IP skončia opäť na stanici, je to IP sieťového adaptéru.
- **10.x.x.x** – Rezervovaný rozsah pre vnútornú sieť
- **192.168.x.x** – Rezervovaný rozsah pre vnútornú sieť
- **172.16.x.x** – Rezervovaný rozsah pre vnútornú sieť

Adresy vnútorných sietí nie sú unikátne a na internetovej sieti sa nevyskytujú, používajú sa vo vnútorných sieťach, ktoré sú oddelené od internetovej siete.

- 255.255.255.255 – Všetky dáta odoslané na túto IP adresu sú vysielané všetkým PC v aktuálnom rozsahu.
- **0.0.0.0** – Táto IP adresa sa používa v smerovacej tabuľke a väčšinou je na konci, znamená všetky ostatné siete, ktoré nie sú inak definované v sieti.

Konfigurácia sieťového adaptéra stanice môže vyzerať podobne ako na obrázku č.6:



Obrázok č. 6: Konfigurácia sieťového adaptéru

Mechanizmus interakcie údajov sa dá opísať sústavou matematických vzťahov. Ako prvú sa bude analyzovať maska podsiete. Jej binárny doplnok určí, koľko staníc sa nachádza v rovnakej podsieti,

Dekadický formát:	255 . 255 . 255 . 0
Binárny formát:	11111111.11111111.11111111.00000000
Binárny doplnok:	00000000.00000000.00000000.11111111
Dekadický doplnok	000 . 000 . 000 . 255

Spolu so stanicou môže byť v podsieti 255 staníc maximálne, ak sa binárne vynásobí maskou podsiete IP adresa získa sa adresa siete.

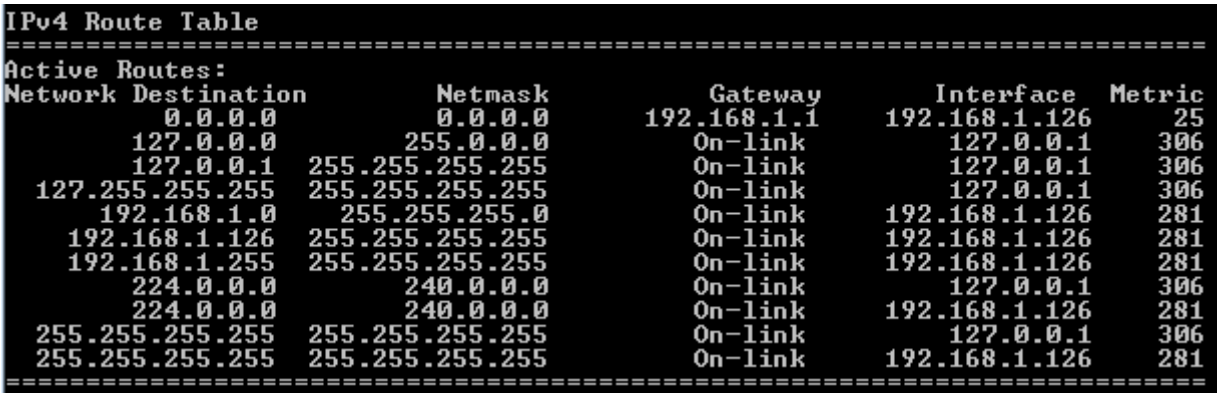
Maska podsiete:	11111111.11111111.11111111.00000000
IP adresa:	11000000.10101000.00000001.01111110
Binárny súčin:	11000000.10101000.00000001.00000000
Dekadický formát binárneho súčinu :	192 . 168 . 1 . 0

Následne sa už iba určí predvolená brána (Väčšinou prvá IP adresa z rozsahu) a broadcast (vždy posledná IP z rozsahu)

Predvolená brána :	192.168.1.1
Broadcast:	192.168.1.255

Ak by bola maska podsiete iná, získali by sa aj tieto sieťové údaje iné., Z príkladu vyplýva fakt, že vždy, keď sa v maske podsiete zmení posledný bit, nasledujúci po poslednej jednotke, rozdelí dostupnú podsieť na dve podsiete. Význam Masky podsiete je aj v rozdeľovaní siete na podsiete.

Väzba masky podsiete a IP adresy sa využíva hlavne v smerovacej tabuľke .



IPv4 Route Table

Active Routes:

Network Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.1.1	192.168.1.126	25
127.0.0.0	255.0.0.0	On-link	127.0.0.1	306
127.0.0.1	255.255.255.255	On-link	127.0.0.1	306
127.255.255.255	255.255.255.255	On-link	127.0.0.1	306
192.168.1.0	255.255.255.0	On-link	192.168.1.126	281
192.168.1.126	255.255.255.255	On-link	192.168.1.126	281
192.168.1.255	255.255.255.255	On-link	192.168.1.126	281
224.0.0.0	240.0.0.0	On-link	127.0.0.1	306
224.0.0.0	240.0.0.0	On-link	192.168.1.126	281
255.255.255.255	255.255.255.255	On-link	127.0.0.1	306
255.255.255.255	255.255.255.255	On-link	192.168.1.126	281

Obrázok č. 7: Jednoduchá smerovacia tabuľka

Na obrázku č.7 vidno jednoduchý príklad smerovacej tabuľky pre jednu stanicu. Prvý stĺpec označuje IP adresu, kam sa chcú dáta doručiť. Druhý stĺpec označuje masku podsiete pre danú lokalitu. Tretí predvolenú bránu, štvrtý fyzický port a piaty metriku. Vynásobením prvého a druhého stĺpca sa získa adresa siete, predvolená brána znamená ďalšiu stanicu na ktorú budú dáta, v prípade potreby odoslať dáta na danú lokalitu, odoslané. Fyzický port znamená, na ktorý sieťový adaptér budú dáta odoslané. Metrika je váha danej cesty: čím je menšia tým je viac uprednostňovaná, je využívaná najmä, keď môže smerovač doručiť dáta viacerými cestami (veľmi častý prípad). Pomocou tohto systému smerovač vytvára bod medzi viacerými sieťami, vonkajšou i vnútornými. Celá internetová sieť je hlavne sústava smerovačov. Pri smerovaní sa používa tiež CIDR notácia, kde sa za adresou siete uvedie, za lomítkom, neprerušený počet binárnych jednotiek masky podsiete, týmto sa získa predstava o divokej karte a to údaj, koľko staníc môže byť v danej adrese sieti. Napríklad formát 192.168.138.0/26 udáva, že v tej istej sieti môže byť umiestnených na poslednom trojčíslí 255 teoretických staníc [1],[2].

2.1.1.4 Smerovanie na vyšších vrstvách

Pri komunikácii aplikácie nestačí iba systém smerovania na prvých troch úrovniach OSI modelu, celý spôsob komunikácie sa rozšíri aj o porty. Port má formát čísla od jednotky po 65535. Port je číslo kanálu, ktorým spolu komunikujú stanice, zatiaľ čo všetky dáta sú zasielané na rovnakú IP, dáta s portom 80 sú určené pre webový prehliadač, dáta s portom 22 sú určené pre vzdialené prihlásenia a tak ďalej. Port sa zapisuje ako číslo, ktoré sa píše po IP adrese a tieto dva údaje rozdeľuje dvojbodka, napríklad notácia 192.168.1.55:80

Pri smerovaní na aplikačnej vrstve stojí za zmienku aj protokol DNS (Domain Name Server). DNS protokol je najzákladnejšia služba pri prezeraní webových stránok, jeho jedinou

úlohou je meniť názvy stránok z tvarov písmenových znakov, teda doménových anotácií, na IP adresy. Potom sú dáta určené prehliadaču skutočne smerované na pravú stanicu, samozrejme táto činnosť sa dá uskutočniť aj opačne. Vzhľadom na decentralizovať všetkých domén a webového priestoru celkovo, neexistuje hlavná databáza. DNS protokol vytvára hlavne sústavu pravidiel pre vzájomnú komunikáciu čiastkových databáz po celom svete a vyhodnocovanie odpovedí [2].

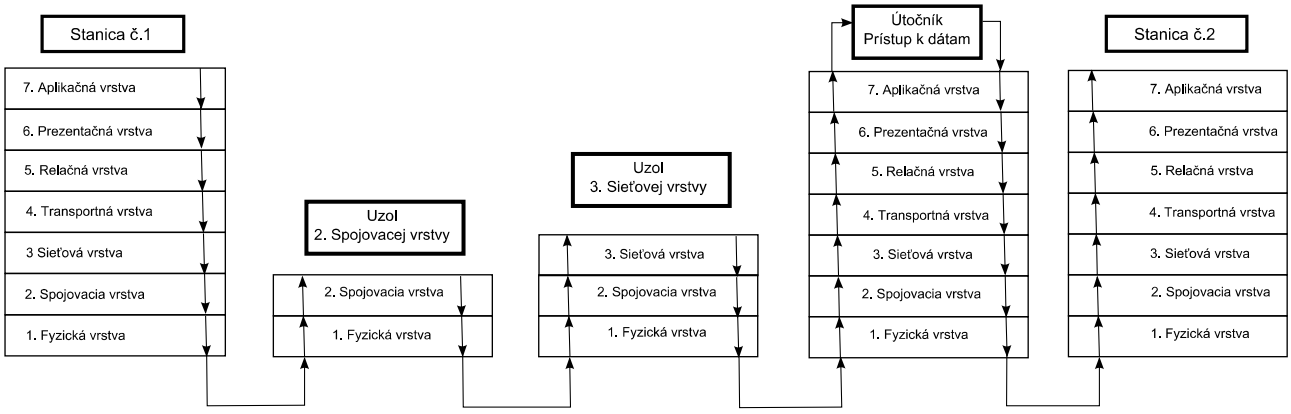
2.1.1.5 Prenos informácií všeobecne

Celý systém komunikácie možno opísať príkladom cestujúceho človeka. Cestovateľ zobrazuje dáta, ktoré hľadajú svoj cieľ, pritom sa pohybuje ulicami (fyzická vrstva). Pozerá na čísla domov, ktoré sú pre každú ulicu jedinečné, ale na rôznych uliciach sa môžu opakovať a dá sa podľa nich orientovať len pre každú ulicu zvlášť (spojovacia vrstva). Cestovateľ sa pohybuje v meste , v ktorom sú ulice, ktoré sa môžu opakovať aj v inom meste, ale mestá sa v štáte neopakujú rovnako ako sa štát neopakuje na svete. Tým je každá adresa jednoznačne určená (smerovacia vrstva). Cestovateľ nájde adresu, príde k domu a chce zazvoniť na zvonček, aby bol vpustený. Ak zazvoní na nesprávny, nikto mu neotvori (smerovacia až aplikačná vrstva, v prípade že je port blokovaný, dáta sú stanicou odmietnuté)

2.1.2 Ochrana dát a šifrovanie

2.1.2.1 Potencionálne útoky

Po pochopení konceptu sieťovej komunikácie v prechádzajúcej sekcii sa dajú opísať aj útoky, ktoré nastávajú pri prenose dát. Ak sa vynechajú útoky v aplikačnej vrstve komunikujúcej stanice, útok môže nastať na ktoromkoľvek bode, ktorý sa nachádza v prenosovej sústave medzi dvoma stanicami.



Obrázok č. 8: Možná poloha útočníka na trase prenosu dát

Útočník môže byť akákoľvek stanica, ktorá je pripojená na akýkoľvek prenosové zariadenie, ktoré sa využíva na komunikáciu medzi stanicami, či už druhej alebo tretej vrstvy. Útočník presvedčí daný bod, aby všetka komunikácia medzi dvomi stanicami prechádzala aj cez jeho stanicu

Tento druh útoku sa označuje ako MiM (z angl. Man in the Middle attack), v preklade muž v strede, cielene odpočúva prenášané dáta a chce využiť ich obsah a uskutočniť tak iný druh útoku, alebo ich pozmeniť a nútiť jednu zo strán vydať údaje, ktoré môžu byť cenné. Väčšinou je cieľom útoku získať zasielané prihlasovacie údaje k rôznym prístupom.

V dnešnej dobe nato môže útočník použiť skutočne širokú paletu techník, väčšinou sú využívané plne automatizované softvérové nástroje, ktoré sú voľne dostupné na internete. Žiaľ takto dávajú moc jedincom bez potrebných znalostí podnikať efektívne útoky, čo je veľmi nebezpečné, z tohto dôvodu masového rozšírenia je potreba chrániť obsah dát nutná. Po technickej stránke útočník postupuje tak že zachytená jednotka dát prejde spätným procesom zapuzdrenia. Prenášaný obsah dát podstúpi analýzu, vykonajú sa prípadné zmeny a dátová jednotka je opäť procesom zapuzdrenia zaslaná k skutočne adresovanej stanici, postup zobrazuje obrázok č.8 [6],[8],[9].

2.1.2.2 Princíp znemožnenia útokov

Zakročiť voči útočníkovi v tomto prípade nie je vôbec jednoduché. Stovky dôverných dát, hesiel, prístupov, súkromných správ a iných cenných údajov môžu byť prenesené a úspešne napadnuté v rámci niekoľkých sekúnd. V najlepšom prípade bude daný sieťový uzol potrebovať na vyhodnotenie útoku a odpojenie útočníka len pár sekúnd ale tie mu budú stačiť. Pri stovkách sieťových uzlov, ktorými prechádza dátová komunikácia je nemožné si byť istý, či niekto dáta neodpočúva. Jedinou funkčnou stratégiou obrany voči odpočúvaniu dátového prenosu sa javí znehodnotenie dát, ktoré sú cieľom útočníka. Možností ako to uskutočniť je veľa, najčastejšie sa prenášané dáta pri aplikovaní vhodnej bezpečnostnej politiky zašifrujú, komunikujúce stanice si tak začnú informácie vymieňať v jazyku, ktorému rozumejú iba oni.

Vytvorenie šifrovaného prenosu medzi dvomi účastníkmi má za predpoklad vytvorenie vlastného virtuálneho spojenia, v rámci ktorého sú všetky údaje šifrované. Takéto spojenie je označené ako virtuálna privátna sieť alebo tiež VPN (z angl. virtual private network). Šifrovanie sa vykonáva na aplikačnej časti OSI modelu a zašifrované dáta sa posúvajú časti OSI modelu zodpovednému za dátový prenos. Zašifrované dáta pomocou procesu zapuzdrenia prepraví až k druhej strane. Takto vznikne medzi zariadeniami akýsi súkromný tunel, ktorého obsah môžu čítať iba oni. Za predpokladu že iba stanice, komunikujúce medzi sebou, dokážu dešifrovať prenášané dáta, možno tento prenos označiť za bezpečný [6],[8],[9].

2.1.2.3 Stratégia implementácie súkromných virtuálnych sietí

Pri prenášaní šifrovaných dát je vždy možné ich dešifrovanie kýmkoľvek, dôležitou premennou je však čas potrebný na dešifrovanie. Princípom je zvoliť dostatočne silné šifrovanie nato, aby sa útočníkovi nepodarilo pri bežne dostupnej výpočtovej kapacite dešifrovať obsah dát v reálnom čase. Určený čas môže byť zhruba 100 rokov, teda čas, za ktorý informácie stratia akýkoľvek význam v reálnom živote. Bezpečný prenos musí spolu s obsahom nečitateľným treťou stranou plniť viacero podmienok:

- **Šifrovaný obsah dát** – prenášané dáta nesmú byť v čitateľnom formáte
- **Integrita** – prenášané dáta nesmú byť pozmenené alebo vymenené za iné
- **Autentifikácia** – obidve strany si musia byť isté identitou druhej strany

Existuje veľa konkrétnych protokolov a riešení pre vytvorenie VPN okruhu, zahŕňajúc tým nadviazanie a samotnú komunikáciu medzi komunikujúcimi stanicami. . Pri snahe o kategorizáciu týchto protokolov, alebo nájdenie ich spoločného menovateľa, sa dá vychádzať z prístupu, akým dochádza k šifrovaniu dát. Existujú dva základné prístupy k šifrovaniu dát, symetrické a asymetrické šifrovanie [8],[9].

2.1.2.4 Symetrické šifrovanie dát

Od slova symetrické teda súmerné šifrovanie, toto šifrovanie využíva princíp, pri ktorom sa využije kľúč na zašifrovanie dát a rovnaký kľúč je potrebný na spätné dešifrovanie. Kľúč aj šifrované dáta sú vo svojej podstate reťazcom bitov (ktoré sa môžu zobrazovať ako reťazce znakov), samotné šifrovanie je otázkou voľby konkrétneho protokolu alebo matematickej metódy. V oboch prípadoch je pri symetrickom šifrovaní k dispozícii široká paleta možností, takto vzniknutý kľúč sa stáva zdieľaným, pretože ho majú obe strany a potrebujú ho ku komunikácii, označuje sa aj ako zdieľané tajomstvo (z angl. Shared secret). Hlavnými výhodami symetrického šifrovania je malá záťaž výpočtových prostriedkov, z čoho plynie, že šifrovanie nie je limitujúcim prvkom rýchlosti komunikácie. Symetrické šifrovanie je prvým a historicky najstarším, počiatky siahajú k roku 1976, prístupom k ochrane dát pri prenose. Počas uplynulého času bolo na tento druh šifrovania úspešne použitých množstvo techník útokov, hlavnými nevýhodami sú :

- Vzhľadom na symetrickosť existujú efektívne metódy na lineárnu analýzu a následné

získanie kľúča, existuje tiež množstvo rovnako efektívnych neanalytických metód

- Pre úspešnú komunikáciu potrebujú obe stanice rovnaký kľúč, problémom sa stáva distribúcia, nakoľko je tajný, nemal by byť prenášaný nešifrovaným prenosom, ale k šifrovanej komunikácii musia strany kľúč už mať, vzniká tak bludný kruh nekonečného cyklu.

Symetrické šifrovanie rieši prvý problém zdieľaným kľúčom, ktorý je platný len pre jednu reláciu, prípadne menením kľúča a matematickej podstaty šifrovania počas komunikácie. Druhý menovaný problém rieši rôznymi metódami výmeny kľúča pri iniciácii komunikácie, k dispozícii sú rôzne metódy podania rúk. Podanie rúk je jednoduchý algoritmus, ktorý nastáva pri iniciácii spojenia a má za úlohu bezpečnú výmenu zdieľaného kľúča výmenou iných parametrov. Ani jedno riešenie však nie je úplne účinné, pri výmene dát na začiatku komunikácie môže útočník umiestnený na trase medzi stanicami, takzvaný útok muža v strede MiM, obdržať tieto dáta a druhej stanici podsunúť vlastné. Následne po obdržaní odpovede z druhej stanice podsunie pôvodnej stanici tiež vlastné dáta, takže započne dve šifrované spojenia, pričom mu je k dispozícii všetka komunikácia staníc, ktoré si myslia že komunikujú iba medzi sebou a navyše bezpečne. Z uvedeného vyplýva, že neexistuje spôsob ako zabrániť útoku MiM. Moderné implementácie symetrického šifrovania skôr vytvárajú algoritmy na praktické potlačenie tohto útoku a pravidelne dodávajú rôzne záplaty na novovzniknuté spôsoby prekonania týchto bezpečnostných algoritmov. Vo svojej podstate ho úplne nevylučujú a nedokážu mu zabrániť. Teoretickým zabránením by bolo doručiť zdieľaný kľúč cieľovej stanici inou cestou, pri dominancii sieťovej komunikácie pomocou siete internet sa iné spôsoby premietajú do nákladov a o ich nasadení rozhoduje už konkrétny dôvod a cieľ využívania.

Napriek uvedeným nevýhodám sa dnes symetrické šifrovanie rozšírene využíva práve pre jeho výhodu rýchleho prenosu, pri jeho implementácii treba zvážiť množstvo otázok bezpečnosti. Implementácii sa dnes venujú skôr veľké firmy, ktoré majú k dispozícii kapacity na riešenie rizík bezpečnosti a dodávajú hotové riešenia spolu zo spôsobmi dodania zdieľaných kľúčov [6],[8],[9].

2.1.2.5 Asymetrické šifrovanie

Princípom asymetrického šifrovania je existencia dvoch kľúčov (kľúčového páru), jeden s kľúčov sa použije na zašifrovanie dát a druhý na odšifrovanie, označujú sa ako súkromný a verejný kľúč (verejný kľúč sa často nazýva aj certifikát). Následnou podmienkou je, že verejný kľúč musí byť dostupný druhej stanici, ktorá ho potrebuje k šifrovaniu dát, jeho distribúcia je bezpečná, zatiaľ čo súkromný kľúč nikdy neopustí pôvodnú stanicu a používa sa na dešifrovanie

obsahu dát zašifrovaného verejným kľúčom.

Matematickou podmienkou tohto mechanizmu sa stáva neschopnosť zistiť súkromný kľúč na základe analýzy verejného kľúča, takéto metódy zatiaľ nie sú známe. Pochopiteľne, na základe analýzy dostatočného množstva zašifrovaných dát a verejného kľúča, je možné súkromný kľúč získať. Exaktná metóda zatiaľ nie je známa, pomocou rôznych metód hrubej sily je to možné v čase niekoľkých rokov v závislosti od veľkosti kľúča, v asymetrickom šifrovaní ostáva veľkosť kľúča často voľbou užívateľa. Kľúčový pár užívateľa sa musí meniť kvôli zachovaniu bezpečnosti šifrovania, dĺžku používania určuje momentálna úroveň výkonu výpočtovej technológie, z nej sa určí približný čas, ktorý potrebuje útočník na získanie súkromného kľúča hrubou silou.

Pri iniciácii prenosu a podania rúk sú vymenené verejné kľúče, algoritmus výmeny kľúčov tiež používa metódy, ktoré prakticky zamedzujú získaniu kľúča treťou stranou. Tento prístup sťažuje získanie verejného kľúča, avšak pri jeho prípadnom získaní treťou stranou, narozdiel od symetrického šifrovania, nehrozí žiadne bezpečnostné riziko a pre uchovanie bezpečnosti stačí dodržiavať obmenu kľúčového páru v určitých časových rozostupoch. Podmienka šifrovaného prenosu je týmto skvele splnená.

Nevýhodou asymetrického šifrovanie, v kontraste so symetrickým šifrovaním, sa stáva jeho matematická náročnosť, ktorá má zvýšené nároky na procesný čas a pri silnej miere šifrovania sa môže stať limitujúcim faktorom rýchlosti komunikácie. Výhodou tohto šifrovania je, napriek matematickej náročnosti priebehu šifrovania, jednoduchosť implementácie pre bežného užívateľa, ktorého činnosť sa pri použití niektorých jednoduchých nástrojov zmení iba na generovanie kľúčov [6],[8],[9].

2.1.2.6 Autentifikácia a integrita asymetrického šifrovania

Používanie verejného a súkromného kľúča skvele rieši otázku bezpečnosti prenosu, ale otázku integrity dát rieši iba čiastočne, pričom nerieši otázku autentifikácie. Útočník môže pri zistení pokusu o výmeny verejných kľúčov tieto nahradiť svojimi a opäť môže odpočúvať dátový prenos. Zvyšné dve otázky integrity a autenticity plne efektívne rieši digitálne podpisovanie.

Každý kľúčový pár obsahuje okrem matematického vzťahu zašifrovania verejným kľúčom a dešifrovania súkromným kľúčom, ešte jeden brilantný matematický vzťah. Čo sa zašifruje jedným kľúčom je možné dešifrovať iba druhým a naopak, definovanie súkromného a verejného sa určuje iba konvenciou pri ich vytváraní. Vďaka tomuto poznatku existuje veľmi logický spôsob ako zaručiť autenticitu dát – digitálny podpis. Algoritmus digitálneho podpisu vyžaduje použitie hash funkcií, tento druh funkcií jednosmerne šifruje dáta, možnosť dešifrovať ich naspäť neexistuje.

Hash funkcia vytvára z ľubovoľne dlhého reťazca určitý reťazec o konkrétnej dĺžke, tá závisí od voľby konkrétnej funkcie, hash funkcia vytvori z rovnakého reťazca vždy rovnaký zašifrovaný reťazec, stačí aby sa zmenil jeden bit v pôvodnom reťazci a zašifrovaný reťazec sa zmení. Zaslanie hash reťazca spolu s dátami zaručí, že druhá strana si môže dáta spracovať rovnakou hash funkciou a výsledky porovnať zo zaslaným hash reťazcom, ak súhlasia obsah dát nebol zmenený. Samozrejme ak útočník chce meniť obsah dát, jednoducho zmení aj priložený hash reťazec, aj tento problém integrity rieši digitálny podpis.

Algoritmus digitálneho podpisu: najprv sa aplikuje hash funkcia na dáta určené na prenos, získaný hash reťazec sa zašifruje súkromným kľúčom, zašifrovanie súkromným kľúčom sa označuje ako podpísanie, následne dáta určené na prenos zašifrujú verejným kľúčom druhej strany. Druhej strane sú zaslané tri súčasti:

- Hash reťazec prenášaných dáta zašifrovaný súkromným kľúčom (podpísaný)
- Verejný kľúč odosielateľa (certifikát)
- Samotné dáta zašifrované verejným kľúčom druhej strany

Druhá strana obdrží tieto súčasti, verejným kľúčom odosielateľa dešifruje hash reťazec odosielajúcej strany, čím je potvrdené že ho mohol zašifrovať len a len držiteľ verejného kľúča (certifikátu). Nasleduje dešifrovanie prijatých dát vlastným súkromným kľúčom, pomocou hash funkcie získa hash reťazec, ten porovná s prijatým hash reťazcom. V prípade zhody je zaručená integrita dát, teda pomocou hash funkcií, a autenticita, pretože hash mohol zaslať len niekto kto disponuje celým kľúčovým párom [8],[9].

2.1.2.7 Certifikačná autorita

Aj pri akceptácii princípov digitálneho podpisu je tu šanca, že útočník útokom typu MiM pri výmene certifikátov podstrčí svoje certifikáty a celú komunikáciu bude odpočúvať. Táto možnosť je však mizivá, pretože verejné kľúče by mali byť široko verejne známe a mala by tu byť možnosť získať si ich z iných zdrojov než len pri iniciácii komunikácie. Túto možnosť poskytuje princíp certifikačnej autority. Certifikačná autorita plní význam tretej strany, ktorej sa komunikujúce strany môžu kedykoľvek opýtať, či je druhá strana dôveryhodná a jej identita je pravá.

Po technickej strane certifikačná autorita nerobí nič iné než svojím súkromným kľúčom zašifruje verejný kľúč jednej strany. Druhá strana si následne verejným kľúčom certifikát dešifruje

a získa verejný kľúč pôvodnej strany, ktorý je ešte zašifrovaný súkromným kľúčom druhej strany, je teda dvakrát digitálne podpísaný a komunikujúca strana si môže byť istá autenticitou druhej strany.

Význam certifikačnej autority je v tom, že jej verejný certifikát musí byť všade a vždy dostupný. Certifikačnou autoritou sú väčšinou obrovské spoločnosti alebo organizácie pôsobiace na poli dátovej komunikácie alebo bezpečnosti. Ich verejné kľúče (certifikáty) sú väčšinou priamo zakomponované do jadra internetového prehliadača alebo iným spôsobom verejne distribuované vždy z viacerých dôveryhodných zdrojov. Získať podpis certifikačnej autority znamená zaslať jej svoj verejný kľúč spolu s požiadavkou na podpis CSR (z angl. Certificate signing request), v ktorom sú uvedené informácie o strane, ktorá žiada digitálny podpis ako názov spoločnosti, email, doména a iné identifikátory bežne používané v elektronickom svete ale aj reálnom živote, obidva súbory digtálne podpíše a zašle certifikačnej autorite. Prácou certifikačnej autority je overiť si skutočnú vernosť požiadavky, uistiť sa že žiadateľ o podpis nesnaží vystupovať pod cudzím menom spoločnosti alebo už existujúcej spoločnosti, tento podpis je samozrejme platený. Pri podpisovaní certifikačnou autoritou nemá útočník pomocou útoku MiM žiadnu šancu. V prípade výmeny svojho certifikátu za certifikát z CSR oprávneného žiadateľa, nemohol by spätne doručiť žiadateľovi jeho certifikát podpísaný certifikačnou autoritou pretože oprávnený žiadateľ by to ihneď zistil pri kontrole z verejne dostupným CA certifikátom. Tento model poskytuje veľmi dobre ošetrené všetky medzery v komunikácii a zaručuje splnenie všetkých podmienok bezpečného prenosu [7],[8],[9].

2.1.3 Dostupné riešenia

2.1.3.1 TLS/SSL

Väčšina protokolov, zaoberajúcich sa bezpečným prenosom dát na báze asymetrického šifrovania, sa pohybuje v rozmedzí aplikačnej až transportnej vrstve OSI modelu. Symetrické šifrovanie sa snaží pôsobiť na nižších úrovniach, a to hlavne na sieťovej, symetrický princíp sa často dotýka priamo systémového jadra. Komunikujúce strany sa dajú definovať vzhľadom na aplikačnú vrstvu, najrozšírenejším víťazom sa stáva webový prehliadač a všetky aplikácie, ktoré pomocou neho komunikujú. V tomto prípade, tiež mnohých iných, pôsobí rodina SSL protokolov (z angl. Secure Sockets Layer) a ich nasledovníci rodina TLS protokolov. (z angl. Transport Layer Security). Obidva protokoly pôsobia nad transportnou vrstvou, ktorej posúvajú na transport už zašifrované časti dát, tieto bezpečnostné protokoly tiež využíva veľmi veľa iných štandartných komunikačných protokolov ako HTTP, FTP, SMTP, NNTP, XMPP, SMTP a samozrejme VPN. V dnešnom prostredí nadobúda tiež využitie pri telefonických službách a protokole VoIP. Rodina

protokolov TLS a SSL komunikuje na základe princípov asymetrického šifrovania, zahŕňa tiež vlastné upravené algoritmy, počiatočného podania rúk a veľmi rozšírenú diagnostiku prípadných chýb v komunikácii protokolu. TLS/SSL protokoly oplývajú bohatým rozsahom súčastí a prešli dlhým vývojom do súčasnej podoby, pričom značne napredovali, čoho dôsledkom sú dnes najrozšírenejšou a najuznávanejšou platformou pri bezpečnej komunikácii. S podobnými princípmi pracuje aj rodina PGP protokolov rovnako zameraná na bezpečný prenos a hlavne digitálne podpisovanie, využívané prevažne v emailovom styku [4],[5].

2.1.3.2 OpenVPN

Veľmi rozšírenou platformou pre vytvorenie VPN okruhu je OpenVPN. Ako napovedá názov, jedná sa o otvorené riešenie, teda jeho zdrojový kód je k dispozícii všetkým užívateľom a každý ho môže upraviť podľa svojich potrieb, slobodne a ľubovoľne používať (v prípade nekomerčného použitia a dodržania licencií GPU a GPL všetkých súčastí).

Po technickej stránke ide o klient-server aplikáciu, pričom klient sa pripája len pomocou tenkého klienta. OpenVPN je pokročilý nástroj pre tvorbu VPN okruhov, obsahuje veľké množstvo nastaviteľných súčastí, väčšinou pracuje na báze asymetrického šifrovania, využívajúc pritom TLS/SSL protokoly, dokáže pracovať aj zo zdieľaným heslom, je tu možnosť voľby emulácie sieťového adaptéra, kompresie, spôsobu smerovania a mnohých iných, hlavnými výhodami sú [5], [8],[9]:

- Podpora širokého spektra operačných systémov a platforiem (unix, Linux, BSD, Windows, OS X)
- Jednoduchá konfigurácia
- Široká nastaviteľnosť a možnosť dôkladnej konfigurácie na veľmi nízkej úrovni nastavení

2.1.4 Zhrnutie cieľov VPN

Zriadenie VPN okruhu má za cieľ zabezpečiť prenos vzhľadom na ochranu súkromných alebo inak citlivých dát. Významy uplatnenia:

- Zabezpečiť prístup do demilitarizovanej zóny konkrétnej siete (firemná, súkromná), v ktorej sa je možné bezpečne pohybovať.
- Zabezpečiť prístup do verejnej siete z nedôveryhodného bodu pripojením na dôveryhodný bod, z ktorého je možné ďalej bezpečne pristupovať do verejnej siete.

- Možnosť pripojiť sa z ktoréhokoľvek bodu siete a pohybovať sa tak bez nutnosti konkrétnej stanice alebo adresy.

Za spomenutie určite stojí, že v dnešnej sieti tvoria najväčšie riziko verejne prístupné bezdrôtové siete, vzhľadom na šírenie signálu útočník nemusí mať žiadne špeciálne pripojenie a dokáže uskutočňovať útoky z bežným vybavením a bez zložitejšej softvérovej prípravy [6],[8],[9],[10].

2.2 Inštalácia VPN okruhu

2.2.1 Vyžadovaná konfigurácia

Cieľom integrácie VPN servera do už existujúcej informačnej štruktúry ústavu je umožniť študentom a zamestnancom Ústavu automatizácie informatizácie a riadenia procesov, Fakulty chemickej a potravinárskej technológie, Slovenskej technickej univerzity v Bratislave (KIRP FCHPT STU BA) pripojiť sa do vnútornej siete ústavu z vonkajšieho prostredia. Zabezpečené pripojenie z domu alebo iného umiestnenia je z viacerých dôvodov potrebné ako pre študentov tak pre zamestnancov. Preto implementovať takúto službu vhodne vyjadruje potrebu rozvoja organizačnej infraštruktúry ako dôsledok rozšírenia služby v okolitom prostredí a overenie jej pozitívnych dôsledkov. Časť informačnej štruktúry, ktorá sa bude konfigurovať, predstavuje už existujúci a v produkčnom prostredí činný, hlavný smerovač (router), ktorý funguje na báze distribúcie pfSense. Hlavný smerovač pôsobí ako jediný prístupový bod do vnútornej siete a okrem smerovania plní mnoho iných funkcií typických pre prístupový bod. Samotné nasadenie musí spĺňať okrem funkčnosti, taktiež mnoho politík prístupov k zdrojom vo vnútornej sieti, rozlične definovaných zvlášť pre študentov a zvlášť pre zamestnancov. Voľnými prvkami pri dosiahnutí vyžadovanej konfigurácie ostáva voľba konkrétnych nastavení algoritmov prenosu virtuálnej súkromnej siete. Tie musia splniť predpokladané bezpečnostné štandardy pri dodržaní vhodnej úrovne prenosového výkonu. Ďalšími podmienkami sú kompatibilita súčasnými službami a distribúciou pfSense.

2.2.2 Dostupné testovacie prostredie

Teoretický opis komunikácie napovedá iba o princípoch, reálne použitie lepšie ilustruje využitie. Cieľom bude zostrojiť jednoduchý VPN okruh za použitia OpenVPN, simulovať sa bude hypotetický univerzitný systém, ktorý tvorí jeden univerzitný server, priamo pripojený do verejnej siete. Na tento server budú pripojené dve oddelené siete, ktoré budú reprezentovať zamestnaneckú

a študentskú sieť. Potreba VPN okruhu vzniká zo strany zamestnancov vstupovať na svoje pracovné stanice z iných sietí a zároveň potreby mať prístupový bod, z ktorého je možné pristupovať do akademickej siete s akademickou IP adresou. Zo strany študentov je tu len potreba pristupovať na zdieľaný študentský server, ktorý zahŕňa užívateľské kontá zo súbormi, ktoré vytvorili v škole, prípadne môžu pristupovať na stanice umiestnené v tejto sieti za iným účelom. K dispozícii sú nasledovné súčasti:

- PC z viacerými sieťovými kartami, vystupujúce ako server priamo pripojený do verejnej siete
- Dve ďalšie stanice, z ktorých každá bude reprezentovať jednu stanicu v jednej sieti
- Prístup do verejnej siete a rozsah IP adries k práci (147.175.79.140 – 160)

Realizácia sa uskutoční zapojením dvoch testovacích staníc do dvoch odlišných sieťových kariet serveru, pričom ten sa sám pripojí tretím rozhraním do verejnej siete, toto jednoduché simulačné prostredie je viac než postačujúce

2.2.3 Rozvrhnutie podsietí

Celý adresný priestor sa skladá z už prideleného adresného priestoru 147.175.79.0/24 ústavu automatizácie informatizácie a riadenia procesov fakulty chemickej a potravinárskej technológie Slovenskej technickej univerzity v Bratislave (KIRP FCHTP STU BA). Tento adresný rozsah je tvorí 147.175.79.0/24 adries vyjadrených v CIDR notácii (z angl. Classless Inter-Domain Rouiting), táto notácia sa dá vyjadriť aj pomocou masky podsiete 255.255.255.0, ústav má k dispozícii celý adresný rozsah. Tento rozsah je však už pre potreby rozdelený na 4 podsiete a to alikvotne, maska podsiete sa rozšíri o toľko bytov, aby ich kombinácie dali počet sietí, ktoré chceme vytvoriť, a teda o dva (pri alikvotnom delení každý pridaný bit zdvojnásobí počet možných sietí), výpočet pre náčrt siete by mohol vyzerať zhruba nasledovne

Binárny formát:

IP adresa	10010011.10101111.01001111.00 000000
Maska podsiete:	11111111.11111111.11111111.11 000000
Divoká karta:	00000000.00000000.00000000.00 111111

Dekadický formát:

IP adresa	147.175.79.0

Maska podsiete: 255.255.255.192
Divoká karta: 0.0.0.63

Divoká karta označuje, koľko staníc bude obsahovať každá podsieť, pri alikvotnom rozdelení, samozrejme zmenšených o dve adresy pre predvolenú bránu a broadcast. Ak sa uvažuje alikvótne rozdelenie a fakt že predvolená brána bude prvá možná adresa v danej podsieti, budú informácie o sieťach vyzerať nasledovne.

Sieť 1:

Network ID: 147.175.79.0/26
Maska podsiete 255.255.255.192
Default gateway: 147.175.79.1
Broadcast: 147.175.79.63

Sieť 2:

Network ID: 147.175.79.64/26
Maska podsiete 255.255.255.192
Default gateway: 147.175.79.65
Broadcast: 147.175.79.127

Sieť 3:

Network ID: 147.175.79.128/26
Maska podsiete 255.255.255.192
Default gateway: 147.175.79.129
Broadcast: 147.175.79.191

Sieť 4:

Network ID: 147.175.79.192/26
Maska podsiete 255.255.255.192
Default gateway: 147.175.79.193
Broadcast: 147.175.79.255

Pridelený rozsah sa nachádza v tretej podsieti, zo systému delenia plynie fakt že akýmsi polovičným delením každú sieť možno rozdeliť na vopred známe množstvo podsietí.

V tomto prípade sa uvedený rozsah musí rozdeliť čo najvhodnejšie na dve podsiete. IP adresa smerovača (routera) sa môže zvoliť napríklad prvá IP adresu a to 147.175.79.140, ak sa chcú z nasledovného rozsahu urobiť čo najvhodnejšie dve podsiete z rovnakou maskou podsiete stačí uvažovať k akým intervalom, alebo častiam rozsahov, sa dostane polovičným delením celej časti tretej siete. Najprijateľnejšie sa javí rozdeliť celkový rozsah tretej siete na osem podsietí z použitím

rozsahu 147.175.79.128/29, alebo sieťovej masky 255.255.255.248, výpočtom sa zistí, ako by vyzeralo celé rozdelenie, výsledné siete budú mať nasledujúce atribúty:

Podsieť 1 (Zamestnanci) :

Network ID:	147.175.79.144/29
Maska podsiete:	255.255.255.248
Predvolená brána:	147.175.79.145
Broadcast:	147.175.79.150

Podsieť (Študenti):

Network ID:	147.175.79.152/29
Maska podsiete:	255.255.255.248
Predvolená brána:	147.175.79.153
Broadcast:	147.175.79.150

Z rozsahov podsietí využijeme prvú možnú IP adresu pre každú stanicu, vystupujúcu v danej podsieti, sieťovým kartám, ktoré vystupujú ako vstupy z podsietí sa nastavia IP adresy predvolených brán. Vstupnému portu smerovača, ktorý slúži ako prístup do vonkajšej siete pre obidve podsiete a smerovač samotný sa nastaví prvá IP z celkového rozsahu, (tento port sa často označuje WAN z angl. Wide area network) všetky adresné rozsahy budú vyzerať:

Smerovač (Router):

Port WAN

IP adresa:	147.175.79.140
Maska podsiete:	255.255.255.192
Predvolená brána:	147.175.79.129

Port podsiete Zamestnanci

IP adresa:	147.175.79.145
Maska podsiete	255.255.255.248

Port podsiete Zamestnanci

IP adresa:	147.175.79.153
Maska podsiete	255.255.255.248

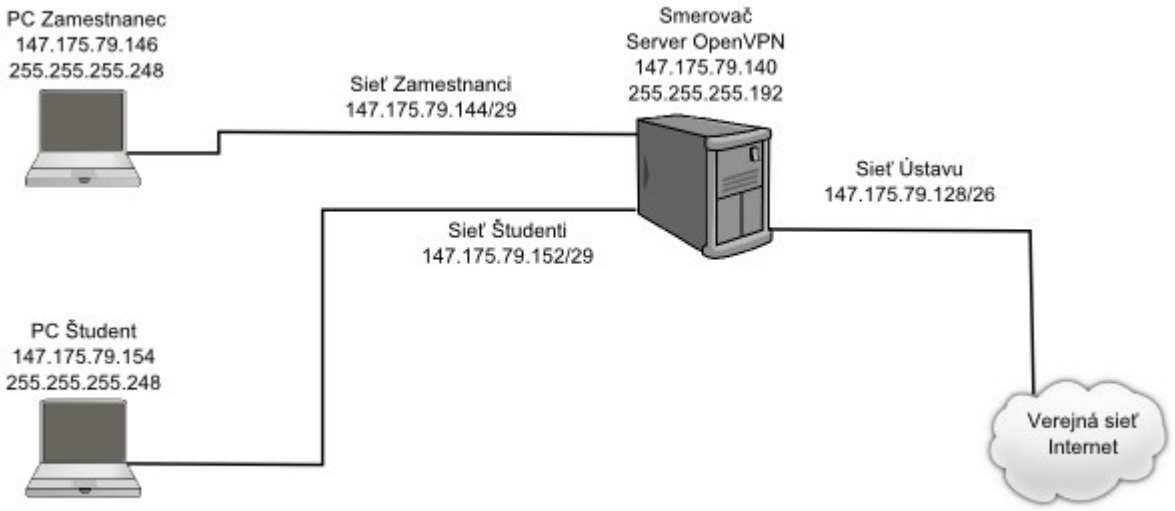
Stanica Zamestnanec:

IP adresa:	147.175.79.146
Maska podsiete:	255.255.255.248
Predvolená brána:	147.175.79.145

Stanica Študent:

IP adresa:	147.175.79.154
Maska podsiete:	255.255.255.248
Predvolená brána:	147.175.79.153

Fyzickú topológiu spolu zo sieťovými údajmi približuje obrázok č. 9 :



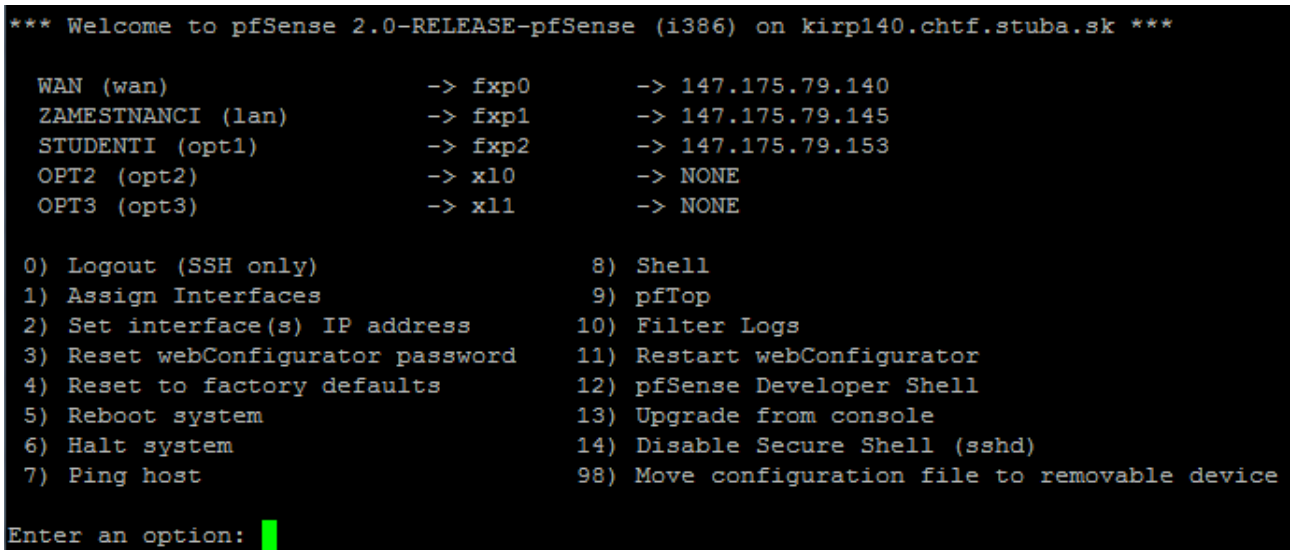
Obrázok č. 9: Topológia zapojenia

2.2.4 Konfigurácia routeru a sieťových adaptérov

PfSense je open-source projekt fungujúci na báze platforme FreeBSD, jedná sa o upravenú distribúciu tohto jadra, ktorej snahou je vystupovať ako router a firewall, zároveň obsahuje rozmanitý prierez open-source programov a súčastí, ktoré sa dajú pri tejto činnosti využiť. To všetko pripravené na použitie vo web GUI (z angl. Graphical User Interface), ktoré je veľmi priateľské a jednoduché na ovládanie aj pre ľudí, ktorý nemajú hlbšie vedomosti o sieti, prípadne nepoznajú všetky možnosti, pfSense samozrejme obsahuje aj implementáciu OpenVPN [11].

V praktickej časti bude PfSense nainštalované na počítač z viacerými sieťovými kartami, ktorý bude plniť funkciu smerovača, firewallu a serverovej inštalácie OpenVPN, pfSense sa dá zdarma stiahnuť z internetu zo stránky projektu. Po stiahnutí stačí obraz pohodlne vypáliť na kompaktný disk a inštalácia môže začať. Samotný priebeh inštalácie nie je predmetom záujmu, počas nej však treba zadať niekoľko dôležitých súčastí, ktorými sú hlavne sieťové rozsahy. Sieťové údaje pre smerovač zadáme priamo pri inštalácii až nato príde čas, dôležitou voľbou je zakázať

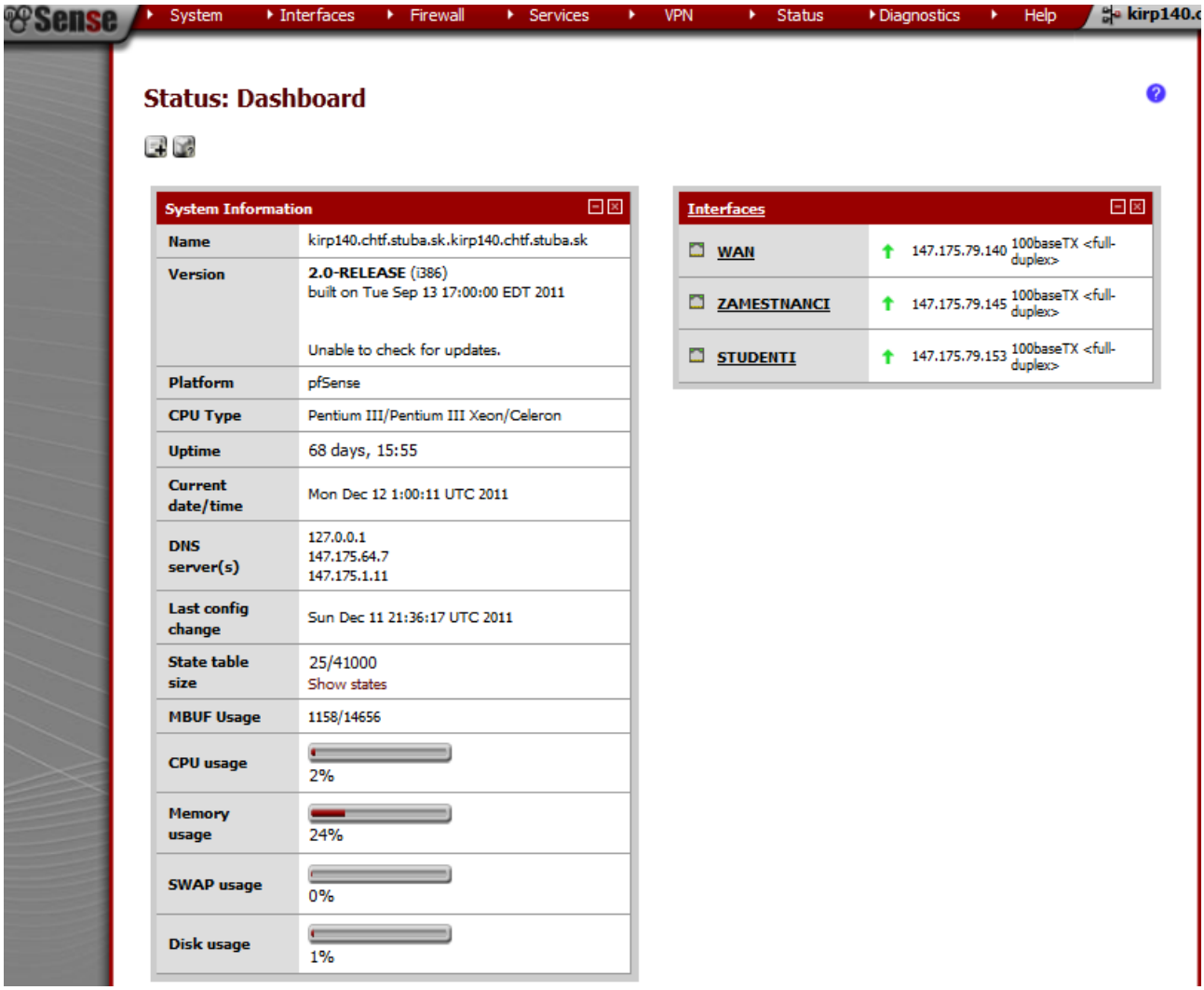
DHCP (z angl. Dynamic Host Configuration Protocol) protokol. DHCP má za úlohu priraďovať staniciam v sieti IP adresy automaticky bez nutnosti ich ručnej konfigurácie, pre demonštračné účely a konkrétne sieťové rozsahy sa od tejto voľby upúšťa, hlavné je zadať adresu pre WAN port, ostatné údaje sa dajú nastaviť aj cez web GUI. Ak sa všetky sieťové nastavenia správne zadali, tak po prihlásení na server v móde príkazového riadku ovládanie prechádza na voľbu možností podľa obrazovky



Obrázok č.10: Možnosti nastavenia pfSense

Názvy volieb majú dostatočne vysvetľujúci charakter pre pochopenie, pfSense je možné plne konfigurovať aj z príkazového riadku po zvolení možnosti číslo 8.

Po prihlásení sa do WEB GUI, jednoducho sa zadá IP adresa smerovača do okna prehliadača, vyzerá základné konfiguračné rozhranie ako na obrázku č. 11:



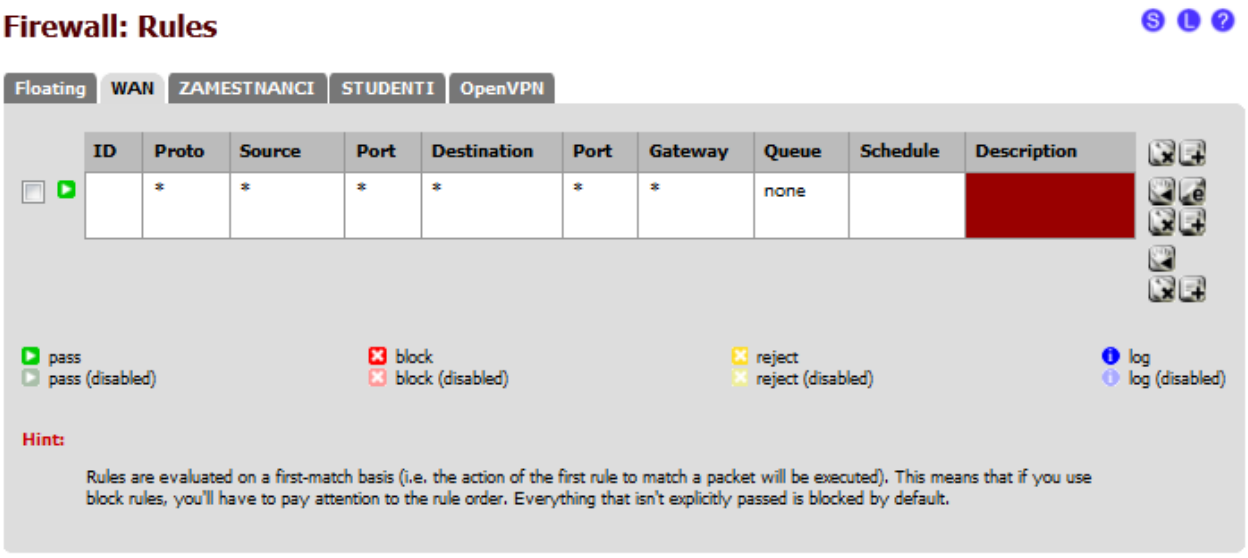
Obrázok č. 11: Možnosti nastavenia pfSense WEB GUI

V úvodnom okne sú zobrazené podrobnosti o systéme a jeho využití, v okne naľavo sa zobrazujú sieťové rozhrania a ich základné nastavenia. V hornej lište sú zobrazené primárne sekcie nastavení, po stlačení sa rozložia na podrobný zoznam nastavení.

Pre funkcionalitu smerovača, nielen v prostredí pfSense, sa musia nastaviť základné súčasti každého smerovača, Prvou z nich je firewall. Firewall plní funkciu programu, ktorý v sebe nesie všetky pravidlá pre komunikáciu a na ich základe v reálnom čase vyhodnocuje všetky prijaté dáta jednoduchým spôsobom, buď ich povolí alebo odmietne. Čo však jednoduché nie je, sú pravidlá ktoré sa dajú pre firewall vytvoriť, tvoria skutočne komplexný systém povolení a zákazov pre všetky prichádzajúce požiadavky na komunikáciu zo stanice a do stanice (firewall sa väčšinou nastavuje pre každý sieťový adaptér zvlášť), hodnotené premenné sú zdrojová alebo cieľová IP adresa, protokol transportnej vrstvy, aplikačný protokol, zdrojový port a cieľový port. Politika

firewallu sa špecifikuje pre každú organizáciu a situáciu zvlášť, patrí k najzložitejším činnostiam pri vytváraní siete a neustále sa mení podľa potrieb siete.

Nastavenie firewallu nie je predmetom praktickej ukážky, samotné prostredie realizácie sa nachádza v demilitarizovanej zóne ústavu a nastavovať firewall nebolo cieľom, v praxi by stačilo povoliť komunikáciu pre všetky zdrojové a cieľové adresy ak budú smerovať na porty, ktoré sú potrebné pre OpenVPN, webový prehliadač a vzdialený konzolový prístup, aj takéto pravidlá tvoria základný prístup, všetko ostatné by samozrejme bolo zakázané. V reálnej situácii by smerovač v roli firewallu filtroval všetku komunikáciu zvonku a veľmi mierne komunikáciu medzi podsieťami, komunikácia medzi jednotlivými stanicami na podsieťach by mala ostať úplne otvorená. V tomto prípade sa firewall nastavil na povolenie absolútne všetkých spojení, dalo by sa hovoriť o vypnutí, v praxi je takéto riešenie neprípustné a má za následok vysokú pravdepodobnosť úspešného útoku a získanie kontroly nad stanicou. Rozhranie pre nastavovanie firewallu sa nachádza na obrázku č.12.



Obrázok č.12: Nastavenie firewallu

V nastaveniach pfSense sa vyžaduje vypnúť predstavenú voľbu NAT (z angl. Network address translation) aby bolo smerovanie v podsieťach korektné, pre ďalšiu konfiguráciu OpenVPN môžeme smerovaču nastaviť aj DNS servery v sekcii System a časti General Setup.

Užívateľské stanice tvorili dve stanice, z ktorých každá zastupuje jedného jedinca v každej podsieti, zamestnanca v zamestnaneckej podsieti a študenta v študentskej podsieti. Na obidve stanice bol inštalovaný Linux Debian 5.6 a už spomínané nastavenia sieťových adaptérov boli jednoducho implementované pomocou príkazu ifconfig.

2.2.5 Konfigurácia OpenVPN serveru

2.2.5.1 Tvorba kľúčov, certifikátov a digitálne podpisovanie

Všetky potrebné užívateľské kľúče a certifikáty sa vytvoria pomocou aplikácie OpenVPN a jej vstavaných nástrojov ako OpenSSL a pkitool. Všetky kľúče, certifikáty a certifikačné autority by bolo možné vytvoriť aj v prostredí pfSense, ale toto prostredie iba spúšťa základné príkazy OpenVPN pod plášťom príjemného užívateľského rozhrania. Všetky úkony boli vykonávané na užívateľskej stanici zamestnanec, na prácu sa hodí ľubovoľné prostredie, priateľskejšie je prostredie s unixovým alebo linuxovým jadrom.

Zdrojový súbor sa stiahne so stránky projektu (napríklad príkazom wget) do zložky, ktorá sa ľubovoľne zvolí, v tomto prípade priečinok */home*, rozbalí sa príkazom

```
tar -xfz openvpn-2.2.1.tar.gz
```

V novovzniknutom priečinku sa uskutoční presun do priečinku *easy-rsa/2.0*, kde sa nachádzajú pracovné skripty.

```
root@debian:/home/openvpn_zamestnanci/easy-rsa/2.0# ls
build-ca           build-key-server  Makefile                    README
build-dh           build-req         openssl-0.9.6.cnf           revoke-full
build-inter        build-req-pass    openssl-0.9.8.cnf           sign-req
build-key          clean-all         openssl-1.0.0.cnf           tmp
build-key-pass     inherit-inter     openssl-1.0.0.cnf-old-copy  vars
build-key-pkcs12   list-crl          pkitool                     whichopensslcnf
```

Ako prvý krok sa edituje súbor vars, ktorého údaje slúžia na vytvorenie certifikačnej autority ale aj všetkých kľúčov a certifikátov, v tomto súbore sú ukladané hodnoty špecifické pre certifikačnú autoritu a neskôr pri podpisovaní certifikátov pre všetkých užívateľov. Najdôležitejšie premenné sa nastavia nasledovne:

```
export KEY_COUNTRY="SK"
export KEY_PROVINCE="SK"
export KEY_CITY="Bratislava"
export KEY_ORG="KIRP"
export KEY_EMAIL="noreply@notexisting.sk"
```

Tieto atribúty sú zároveň povinné, v súbore *vars* sa nachádza mnoho ďalších, ale tie sa môžu nechať prázdne alebo sa doplniť o údaje zvyšujúce výpovednú hodnotu certifikátu. Súbor *vars* sa musí inicializovať, čím sa alokujú všetky premenné, potom sa použije príkaz, ktorý zmaže všetky prípadné doteraz vytvorené certifikáty alebo kľúče a pripraví miesto pre príchod nových.

```
. ./vars

. ./clean-all
```

Prípravné kroky sú týmto ukončené a nasledovným príkazom *build-ca* sa vytvori základný certifikát a to certifikačná autorita:

```
root@debian:/home/openvpn_zamestnanci/easy-rsa/2.0# ./build-ca
Generating a 1024 bit RSA private key
.............................++++++
..................++++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [SK]:
State or Province Name (full name) [SK]:
Locality Name (eg, city) [Bratislava]:
Organization Name (eg, company) [KIRP]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:
Name []:
Email Address [noreply@notexisting.sk]:
```

Pri každej položke sa systém opýtal, či sa má skutočne identifikačná premenná použiť, nakoľko nie je potreba meniť z prednastaveného súboru *vars*, stačí zakaždým stlačiť enter. Po prebehnutí procedúry sa v adresári zobrazil prvý kľúčový pár, ktorý patrí certifikačnej autorite. Druhým krokom je vytvorenie kľúčového páru pre server, tentokrát pomocou príkazu *build-key-server*:

```
root@debian:/home/openvpn_zamestnanci/easy-rsa/2.0#           ./build-key-server
opvpn_key_zamestnanci
Generating a 1024 bit RSA private key
......++++++
.............................................................++++++
writing new private key to 'opvpn_key_zamestnanci.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [SK]:
State or Province Name (full name) [SK]:
```

```
Locality Name (eg, city) [Bratislava]:
Organization Name (eg, company) [KIRP]:
Organizational Unit Name (eg, section) [changeme]:
Common Name (eg, your name or your server's hostname) [opvpn_key_zamestnanci]:
Name [changeme]:
Email Address [noreply@notexisting.sk]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using    configuration    from    /home/openvpn_zamestnanci/easy-rsa/2.0/openssl-
0.9.8.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'SK'
stateOrProvinceName   :PRINTABLE:'SK'
localityName          :PRINTABLE:'Bratislava'
organizationName      :PRINTABLE:'KIRP'
organizationalUnitName:PRINTABLE:'changeme'
commonName            :T61STRING:'opvpn_key_zamestnanci'
name                  :PRINTABLE:'changeme'
emailAddress          :IA5STRING:'noreply@notexisting.sk'
Certificate is to be certified until Dec  8 09:59:49 2021 GMT (3650 days)

Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Pri vytváraní kľúča je opäť potrebné potvrdzovať hodnoty, ktoré ako sám príkaz upozorňuje, budú zahrnuté do certifikátu, v istej časti príkaz ponúka automatickú možnosť digitálne podpísať novovzniknutý certifikát certifikačnou autoritou, táto vlastnosť je samozrejme vyžadovaná takže áno. Veľmi podobný príkaz sa použije na vytvorenie používateľského kľúčového páru aj s digitálnym podpisom, tento príkaz berie ako argument budúci spoločný názov kľúčového páru, vytvárať sa takto dá množstvo párov, pri každom nasleduje rovnaká postupnosť ako pri vytváraní kľúčového páru serveru..

```
./build-key zamestnanec1
./build-key zamestnanec2
./build-key zamestnanec3
```

Všetky vytvorené kľúčové páry sú umiestnené v adresári *keys* v príslušnej zložke, platí prísne pravidlo bezpečnosti, súkromné kľúče, súbory s príponou key sú tajné a nikdy nesmú byť prezradené nikomu okrem vlastníka. V adresári *keys* sa nachádzajú ešte iné súbory, ktoré vznikali v priebehu vytvárania kľúčových párov, ako napríklad CSR súbory, pre skutočné použitie majú význam len súbory s príponou key a crt.

```
root@debian:/home/openvpn_zamestnanci/easy-rsa/2.0# ls keys/
01.pem  index.txt                  opvpn_key_zamestnanci.key  zamestnanec2.crt
02.pem  index.txt.attr             serial                     zamestnanec2.csr
03.pem  index.txt.attr.old         serial.old                 zamestnanec2.key
04.pem  index.txt.old              zamestnanec1.crt           zamestnanec3.crt
ca.crt  opvpn_key_zamestnanci.crt  zamestnanec1.csr           zamestnanec3.csr
ca.key  opvpn_key_zamestnanci.csr  zamestnanec1.key           zamestnanec3.key
```

Absolútne celý uvedený postup sa dá analogicky zopakovať pre študentský prístup, na serveri vykonávajúcom funkciu smerovača budú totiž dve inštancie OpenVPN servera, a na každý zvlášť sa budú pripájať zamestnanci a študenti, je to nutné z hľadiska aplikácie rôznych pravidiel pre tieto skupiny. Postup vytvárania bude rovnaký až na to že bude prebiehať v inej rozbalenej inštalácii OpenVPN, aby nedochádzalo k premazaniu pôvodnej certifikačnej autority a problémom pri vytváraní. Pracuje sa v zložke:

```
root@debian:/home/openvpn_studenti/easy-rsa/2.0# ls keys/
```

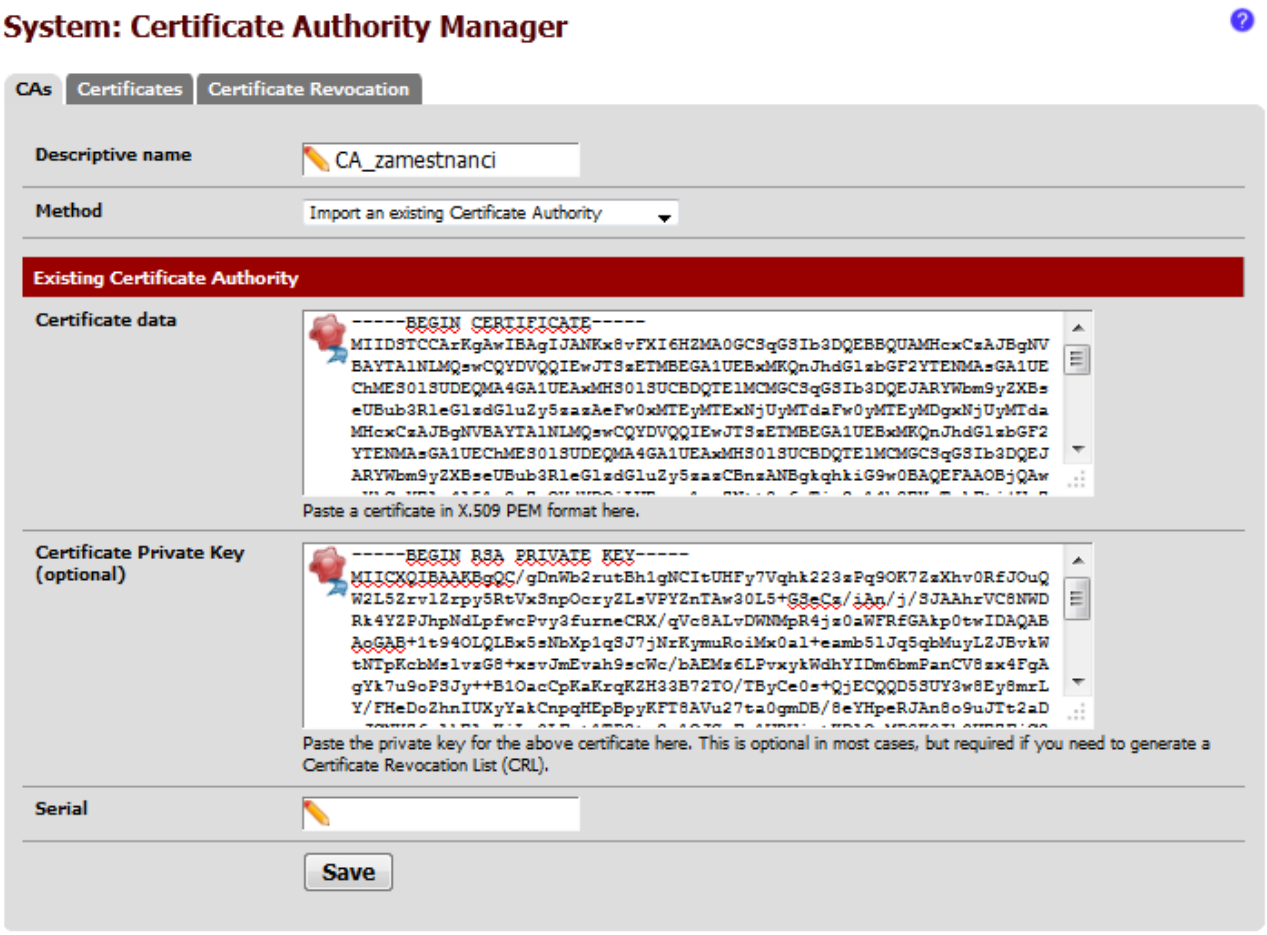
Ako prvý sa opäť upraví súbor vars o potrebné údaje, nasledovať bude súbor príkazov, ktoré sú postupne spracovávané stláčaním enter pri každej požiadavke.

```
. ./vars
. ./clean-all
. ./build-ca
. ./build-key-server opvpn_key_studenti
. ./build-key student1
. ./build-key student2
. ./build-key student3
```

podobne ako v prvom prípade sa priečinok *keys* zaplní súbormi, ktoré stačí už iba zadať.

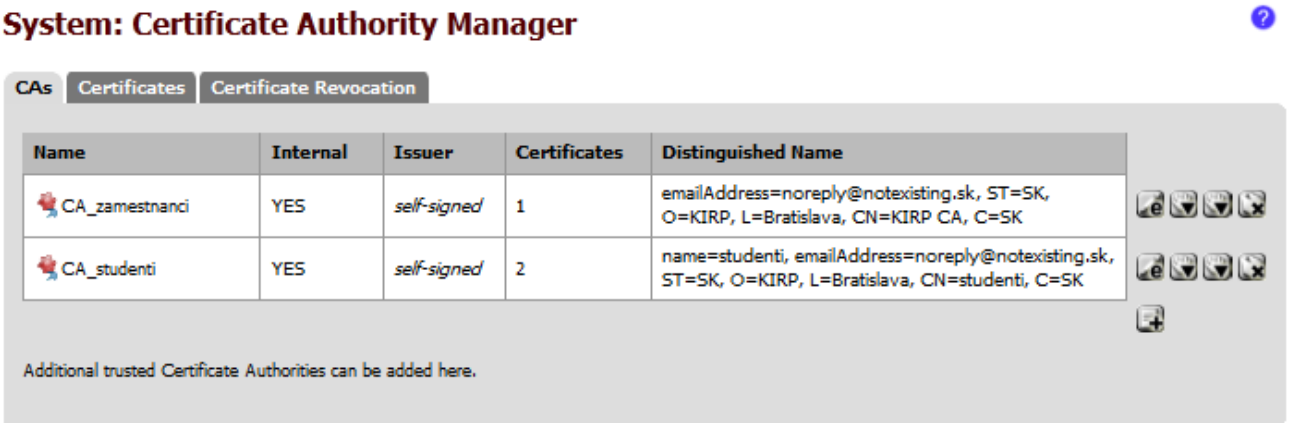
2.2.5.2 Zadávanie kľúčov

Všetky vytvorené súbory stačí už iba skopírovať a vložiť do príslušných polí v grafickom prostredí pfSense. Ako prvý sa vloží kľúčový pár certifikačnej autority do sekcie *System* časti c*ert manager*



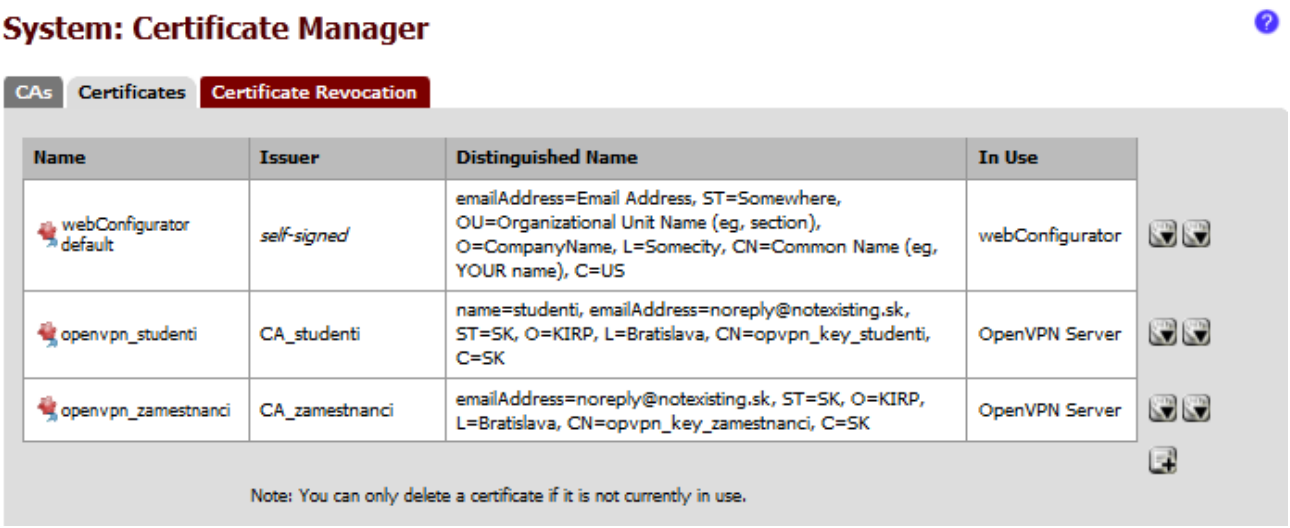
Obrázok č. 13: Vloženie kľúčového páru certifikačnej autority

rovnaká činnosť sa vykoná pre certifikačnú autoritu študentskej časti siete a výsledný náhľad vyzerá ako obrázok č. 14.



Obrázok č. 14: Vložené kľúčové páry certifikačných autorít

Do podobného rozhrania sa vkladajú kľúčové páry pre OpenVPN server, výsledný náhľad na obrázku č. 15.



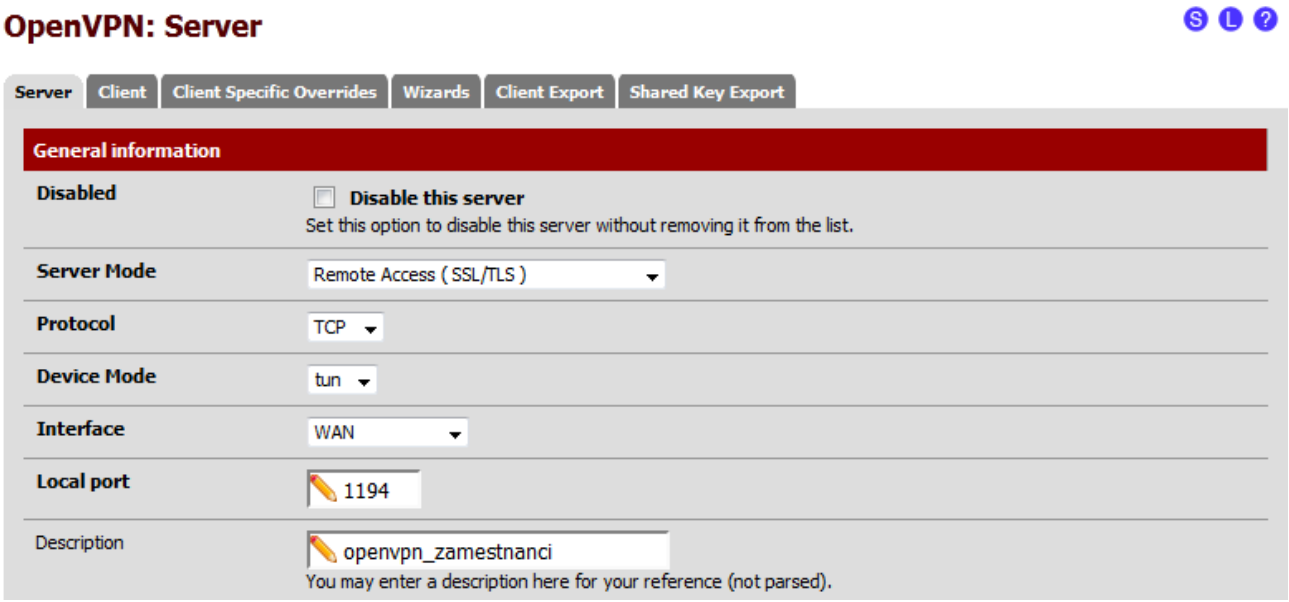
Obrázok č. 15: Vložené kľúčové páry OpenVPN serverov

Certifikáty sú úspešne vložené, to že ich rozhranie prijalo, značí správu väzbu medzi kľúčovým párom.

Pre úspešnú komunikáciu potrebuje server poznať súkromný aj verejný kľúče OpenVPN servera a verejný kľúč certifikačnej autority, súkromný kľúč nemusí byť zadaný napriek uvedenému.

2.2.5.3 Nastavenia serverovej časti Open VPN

Celá konfigurácia prebieha pomocou grafického rozhrania, toto rozhranie iba vytvorí konfiguračný súbor pre serverovú časť OpenVPN, to ale bez toho aby bol užívateľ zaťažovaný správnym syntaxom. OpenVPN server sa nastaví v sekcii VPN v časti OpenVPN, v otvorenej ponuke stačí zvoliť ikonu pridať nový server. Prvú hlavnú časť nastavení zobrazuje obrázok č. 16.



Obrázok č. 16: Hlavné nastavenia OpenVPN serveru

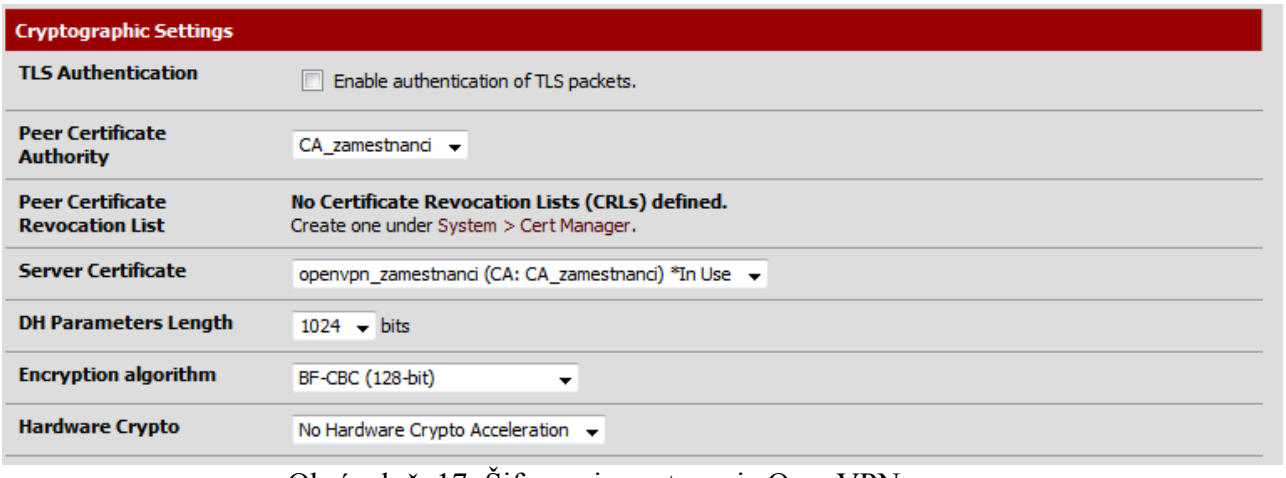
Možnosti:

- Server Mode – Zobrazuje ponúknuté možnosti vzťahov medzi serverom a klientom, na výber sú možnosti, kedy môžu naraz komunikovať iba dve strany, či budú komunikovať pomocou symetrického alebo asymetrického šifrovania prípadne, či má byť komunikácia na začiatku potvrdená heslom. Výber na obrázku č. 16 zodpovedá situácii, kedy server slúži na komunikáciu s ľubovoľným počtom klientov pomocou asynchónneho šifrovania bez potreby dodatočnej autentifikácie hesla (táto je nepotrebná kvôli zvolenej politike certifikačnej autority). Celá ponuka nastavení sa dynamicky mení podľa zvoleného nastavenia, ak sa zvolí táto možnosť, ďalšie nastavenia sa prispôsobia podľa jeho možných konfigurácií
- Protocol - Volí sa protokol transportnej vrstvy, na výber je iba TCP alebo UDP, rozdiel spočíva v činnosti protokolu. Každý protokol transportnej vrstvy dostáva dáta od vyšších vrstiev, ale on rozhoduje, ako tieto dáta rozdelí a doručí. Protokol TCP vyžaduje po každej zaslanej časti potvrdenie o doručení a následne zašle ďalší diel, protokol UDP zasiela časti dát bez potvrdení. Ak druhá stanica zaregistruje, že časť dát neobdržala, vyžiada si ju znovu a protokol UDP ju znovu zašle. Naznačený rozdiel je v rýchlosti, kde vyhráva protokol UDP, a v kvalite a istote doručenia, kde víťazí protokol TCP, zvolená bola možnosť TCP pre kvalitnejšiu integritu dát.
- Device Mode – Na výber sú dve možnosti, TUN a TAP, ide o virtuálne zariadenie systémového jadra, TUN pracuje na tretej vrstve a TAP na druhej. Po zvolení systém vytvorí virtuálne sieťové zariadenie, na ktoré sú odosielané všetky dáta, ktoré sú určené aplikácii

a zároveň aplikácia na ňu odosiela všetky dáta určené na prenos, systémové jadro potom rozhoduje o ich smerovaní. Voľba TUN či TAP vytvorí virtuálny tunel, v ktorom sa komunikuje pomocou druhej alebo tretej vrstvy, pri tomto móde (označovanom aj ako Road Warrior, kedy sa môže stanica pripojiť kedykoľvek, zároveň môže byť pripojených viac staníc). Vhodnejším ostáva TUN adaptér, pretože sa pracuje zo zložitejším systémom podsietí. Voľba TAP sa používa len zriedka a väčšinou pri komunikácii dvoch statických staníc. Voľba TUN alebo TAP adaptéru musí byť na serverovej aj klientskej strane určená.

- Interface – Názov sieťového adaptéru z ktorého bude možné prijímať pripojenie pre OpenVPN relácie, zvoliť možno aj všetky porty, ale predpokladá sa, že relevantné spojenia môžu byť nadväzované iba z vonkajšej siete.
- Local Port – Určuje číslo portu, na ktorom bude celá služba komunikovať a vôbec celý program fungovať, pochopiteľnej po vytvorení spojenia môžu stanice komunikovať v súkromnej sieti na rôznych portoch a celú komunikáciu zastrešuje jeden virtuálny tunel bežiaci na zvolenom porte. (Port 1194 využíva služba OpenVPN predvolene).
- Description – Neurčený názov využívaný len ako identifikácia pre ľudskú jednotku obsluhy systému.

Ďalšia časť na obrázku č. 17 má za dôsledok nastavenia šifrovania



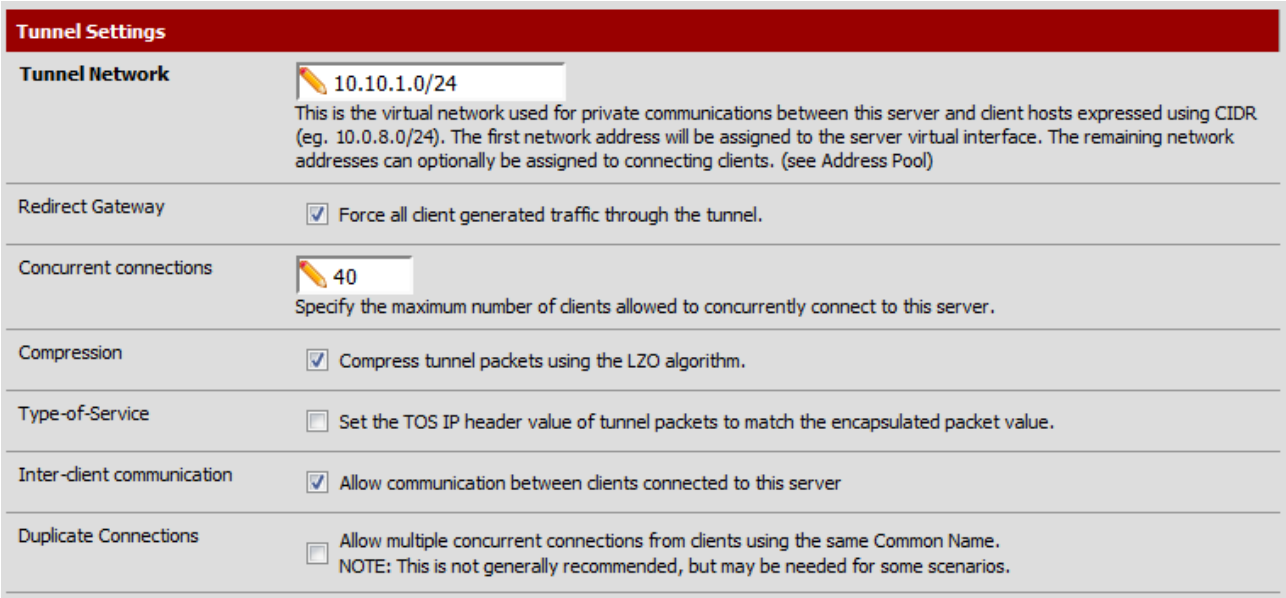
Obrázok č. 17: Šifrovacie nastavenia OpenVPN serveru

- TLS Authentication – Voľbá dáva možnosť vložiť zdieľaný kľúč použitý na symetrické šifrovanie, jeho použitie opäť otvára nekonečný kruh otázky bezpečnosti pri doručení zdieľaného kľúča druhej strane, známy tiež ako egg-chicken paradox. Použitím

asymetrického šifrovania by bola táto forma ochrany už redundantná, preto ostáva nevyužitá.

- Peer Certificate Authority – Systém zobrazuje možné certifikáty certifikačných autorít, zvolí sa certifikačná autorita vytváraná pre potreby OpenVPN serveru zamestnancov
- Peer Certificate Revocation List – V prípade voľby komunikácie iba medzi dvomi stanicami sa používajú jednorázové symetrické kľúče, táto voľba dáva prístup k databáze už využitých kľúčov a ponúka možnosť znovu ich využiť, opäť sa z relevantných dôvodov nepoužije.
- Server Certificate – Volí sa kľúčový pár pre komunikáciu, zvolil sa zamestnanecký kľúčový pár vytvorený práve pre serverovú komunikáciu.
- DH Parameters Length – Veľkosť parametra diskrétneho matematického algoritmu výmeny kľúčov Diffie-Hellmana, používa sa hľavne pri symetrickom šifrovaní, pri asymetrickom šifrovaní nemá veľký význam ale môže zvýšiť bezpečnosť pri počiatočnej iniciácii spojení.
- Encryption algorithm – Voľba konkrétnej metódy, ktorou sa bude prenos šifrovať s využitím verejného kľúča (certifikátu) druhej strany, výber pozostáva s veľkého zoznamu metód z rôznou dĺžkou šifrovaného bloku, nie všetky metódy musia byť podporované obidvoma stranami.
- Hardware Crypto – Ak dostupný hardvér podporuje konkrétnu kryptovaciu metódu, môže sa použiť na urýchlenie komunikácie. Možnosť sa využíva len vo výnimočných prípadoch, existujú ovládače, ktoré túto činnosť môžu vyvíjať, ale prax ukázala minimálnu efektívnosť bez využitia špeciálneho hardvéru.

Ďalšia časť nastavení na obrázku č 18 pojednáva o možnostiach vytvoreného sieťového tunelu :



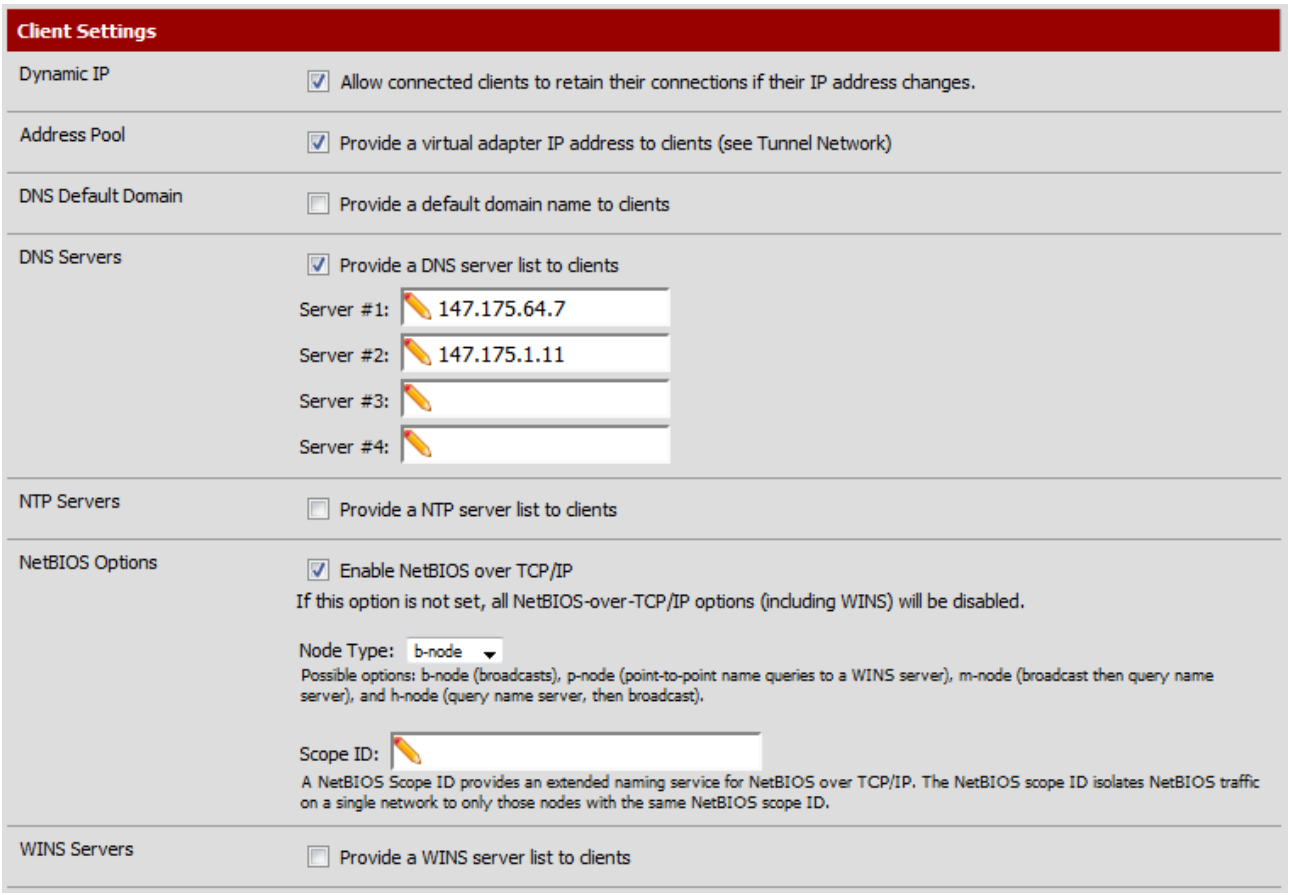
Obrázok č. 18: Tunelové nastavenia OpenVPN serveru

- Tunnel Network – Týmto sa určí rozsah IP adries, ktorý chceme poskytovať pripojeným staniciam, používa sa CIDR notácia, zvolený prípad prideľuje 254 možných IP adries, zvolená bol rozsah vnútornej podsiete. Prvá adresa je automaticky priradená predvolenej bráne v tejto sieti. Po úspešnom pripojení samozrejme každá stanica vystupuje pod IP adresou smerovača voči vonkajšiemu svetu.
- Redirect Gateway – Dôležité nastavenie, určuje aká sieť je dostupná po pripojení, po zaškrtnutí danej možnosti budú absolútne všetky pripojenia presmerované cez router a zároveň bude mať pripojený klient všetky možnosti pripojenia, ktoré mu firewall povolí. Voľba vykresľuje potrebu zamestnancov pripájať sa do akademickej siete a mať do nej plný prístup, to je možné iba ak je entite pridelená akademická IP adresa, týmto prístupom má entita zároveň môže pristupovať do obidvoch podsietí, študentskej aj učiteľskej.
- Concurrent connections – Znamená zvoliť rozumný počet maximálne súčasne pripojených klientskych staníc, určuje sa rôzne podľa potrieb.
- Compression – Zvolením sa využije bezstratový LZO algoritmus na kompiláciu obsahu v reálnom čase, pri zložitosti šifrovania sa zmenšením obsahu zrýchľuje čas prenosu, táto voľba napomáha k výkonu VPN okruhu.
- Type-of-Service – Pomerne nebezpečné nastavenie, ktoré v hlavičke prenášaných dát na tretej úrovni, paketu, prepisuje namiesto druhu služby VPN druh služby na skutočný druh služby prenášaný v šifrovanom obsahu, nastavenie má viacero bezpečnostných dôvodov pre

ktoré sa nepoužije, existujú mimoriadne situácie, kedy tento bod opodstatnenie má.

- Inter-client communication – Povoľuje pripojeným staniciam vzájomnú komunikáciu v súkromnej sieti VPN okruhu, zamestnanci môžu slobodne komunikovať, rozhodnutie aplikovať túto položku vyplýva iba z uplatňovanej politiky.
- Duplicate Connections – Možnosť vytvárať duplicitné pripojenia, tiež nebezpečné nastavenie, neodporúča sa používať sa ho.

Posledná séria nastavení na obrázku číslo č. 19 sa týka možností klientskeho nastavenia



Obrázok č. 19: Nastavenie klientskych vlastností OpenVPN serveru

- Dynamic IP – Touto voľbou sa klientovi povoľuje klientovi uchovať spojenie, ak počas neho zmení IP adresu, častý jav vo verejne dostupných sieťach.
- Address Pool – Nastavenie určuje, či sa má jedincovi poskytnúť IP adresa v prípade, ak využíva TUN adaptér, vyžadovaná vlastnosť.
- DNS Default Domain – Užívateľa zahrnie do domény pridaním doménového mena, táto

možnosť nebude potrebná na aktuálne účeli využívania siete

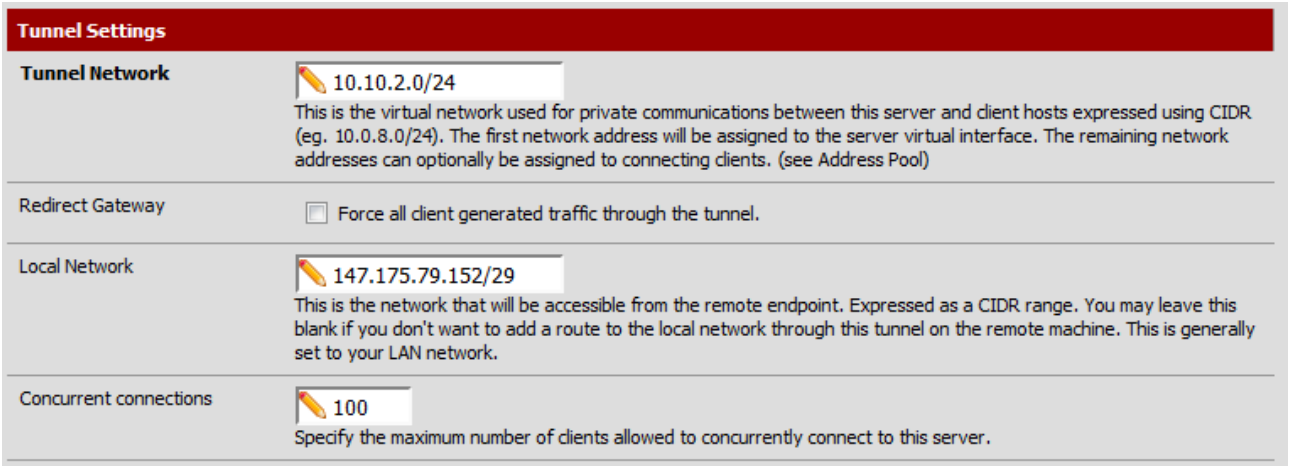
- DNS Servers – Zoznam DNS serverov, ktoré budú poskytnuté pri pridelení virtuálnej IP adresy, v prípade smerovania všetkej komunikácie cez VPN okruh sú potrebné na správne fungovanie webového prehliadača. Použité boli univerzitné DNS.
- NTP Servers – Poskytne stanici na výber vlastný zoznam serverov určených na synchronizáciu času. Zbytočné nastavenia vzhľadom na množstvo týchto serverov a ich dostupnosť, používa sa len výnimočne.
- NetBIOS Options – Protokol NetBIOS pôsobí na piatej úrovni OSI modelu, má za účel prideľovanie vlastných znakových mien zariadeniam v lokálnej sieti, vykonáva podobnú činnosť ako DNS server ale iba lokálne. Tento protokol je dôležitou súčasťou, ktorá spája rôzne systémové platformy a umožňuje využívať rôzne zariadenia ako tlačiarne, dátové úložiská a rôzne iné. V konkrétnej implementácii má najdôležitejšie uplatnenie v protokole SAMBA, ktorý umožňuje zdieľanie súborového priestoru po sieti a je to vyžadovanou súčasťou. Voľbou typu uzla sa dá určiť či má protokol kontaktovať lokálny NetBIOS server a zoznam mien si vypýtať alebo pravidelne vysielať do siete oznamy a takto mapovať všetky mená a IP adresy v podsieti. Scope ID ostáva prázdne, používa sa ak chceme NetBIOS súčasti siete rozdeliť do viacerých sekcií.
- WINS Servers – Servery pre službu WINS, nadstavbu NetBIOS, pri využití uzlového módu broadcast nemá význam.

Ak má užívateľ pocit chýbajúceho článku konfigurácie OpenVPN, môže ho doplniť formou príkazu do poslednej časti na obrázku č. 20:

Obrázok č. 20: Prípadné dodatočné nastavenia

Stlačení voľby Save sa uloží aktuálna konfigurácia a vytvorí OpenVPN server.

Študentský server sa vytvorí veľmi podobne s rozdielmi, ktoré sú dané politikou prístupu, okrem použitia študentskej certifikačnej autority a kľúčového páru pre študentský OpenVPN server sa bude študentská časť rozlišovať v časti nastavenia sieťového tunelu na obrázku č. 21:



Obrázok č. 21: Dodatočné nastavenia študentského OpenVPN servera

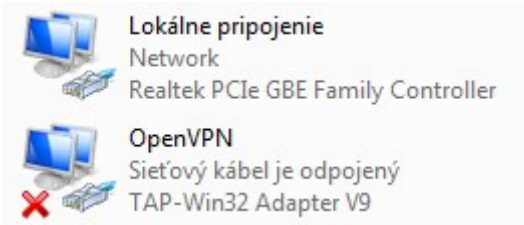
- Tunnel Network – Nastaví sa rozdielna virtuálna sieť, v možnostiach firewallu na serveri sa dá následne presne špecifikovať možnosť a úroveň komunikácie medzi pripojenými klientami.
- Redirect Gateway – Narozdiel od zamestnaneckej časti študenti nesmú mať prístup do akademickej siete ale iba do študentskej časti podsiete.
- Local Network – Tu sa pomocou CIDR notácie presne špecifikuje časť siete, ktorá má byť prístupná po vytvorení OpenVPN okruhu, po výpočte rozsahu má študent prístup iba do študentskej časti siete a na stanice v nej prístupné. Zatiaľ čo všetka zamestnanecká komunikácia je smerovaná cez VPN okruh, aj keby bol cieľ bližšie, v študentskej časti je smerovaná cez VPN okruh iba komunikácia smerovaná do študentskej časti podsiete, všetka ostatná prúdi cez vlastné internetové pripojenie študenta.
- Concurrent connections - sa môžu zväčšiť vzhľadom na väčší počet študentov.
- Port – Služba sa nemôže používať na dvoch rovnakých portoch, preto bude študentský OpenVPN prijímať na porte 1195.

Okrem uvedeného sa dobrým nápadom javí vypnúť možnosť vzájomnej komunikácie medzi pripojenými stanicami a možnosti NetBIOS serveru, všetko preto minimalizáciu možných škôd.

2.2.6 Konfigurácia klienta OpenVPN

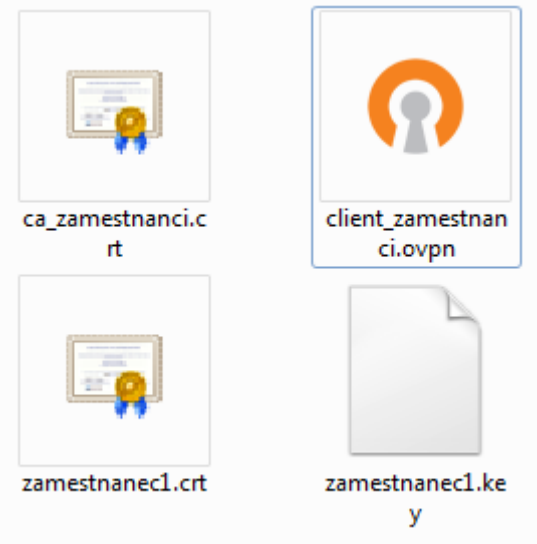
2.2.6.1 Konfigurácia v prostredí Windows 7

Prvá úlohou je nainštalovať OpenVPN v prostredí Windows, uvedený návod platí pre Windows 7 ale s veľmi malými odchýlkami je platný aj pre Windows XP. Inštalačný balík sa nachádza voľne dostupný na stránke projektu, po jeho stiahnutí a úspešnej inštalácii, ktorú tvorí len potvrdenie niekoľkých krokov, vytvára prvú interakciu vznik nového sieťového adaptéru v nastaveniach siete, tento sa musí nutne upraviť do tvaru bez špeciálnych a bielych znakov ako na obrázku číslo 22, napríklad na OpenVPN.



Obrázok č. 22: Sieťový adaptér OpenVPN aplikácie

Nasleduje konfigurácia klientskej strany, tú tvorí tvorba konfiguračného súboru, tvorba klientskeho kľúčového páru a získanie certifikátu certifikačnej autority. Všetky súbory stačí uložiť do adresára, v prípade predvolenej cesty pri inštalácii, : *C:\Program Files\OpenVPN\config*



Obrázok č. 23: Potrebné súbory

Tri so štyroch potrebných súborov boli vytvorené v predchádzajúcich krokoch, stačí ich skopírovať prípadne vytvoriť a zmeniť typ súboru. Ostáva vytvorenie konfiguračného súboru s príponou ovpn. Jeho obsah je predmetom výberu zo všetkých možností obsahu základného konfiguračného súboru na stránke projektu, zostáva zvoliť konkrétne súčasti, tvar konečného súboru:

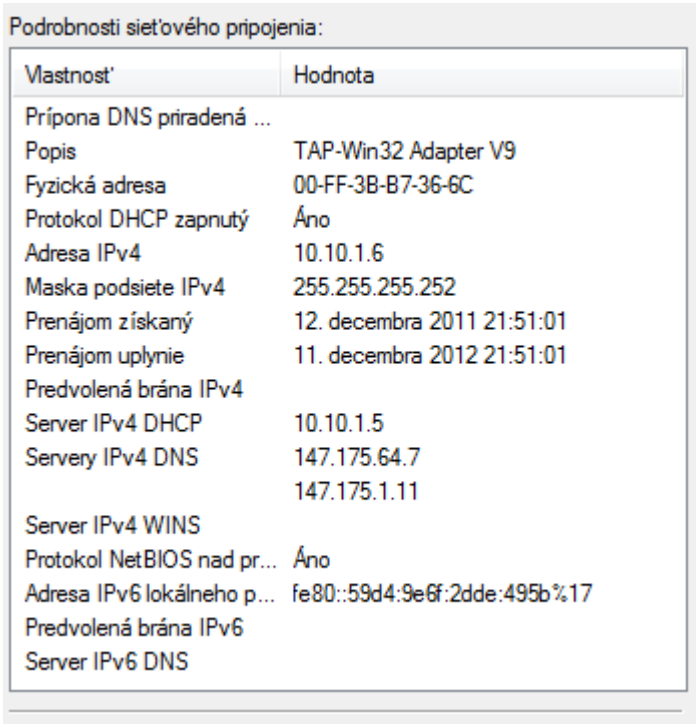
```
client
dev tun
dev-node OpenVPN
proto tcp
remote 147.175.79.140 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca_zamestnanci.crt
cert zamestnanec1.crt
key zamestnanec1.key
ns-cert-type server
cipher BF-CBC
keysize 128
comp-lzo
verb 3
mute 20
```

Každý riadok zodpovedá o spôsobe konfigurácie, musí byť v súlade zo stranou serveru, konfiguračný súbor v sebe nesie nasledovné informácie:

- client – Stanica obsadzuje klientsku pozíciu vo vzťahu pripojenia
- dev tun – Komunikuje pomocou adaptéru TUN
- dev-node OpenVPN – Názov sieťového rozhrania OpenVPN vzniknutého po inštalácii.
- proto tcp – Použitý bude protokol TCP
- remote 147.175.79.140 1194 – adresa serveru na, ktorý sa má stanica pripojiť
- resolv-retry infinite – Počet pokusov koľkými sa bude klient snažiť pripojiť, v tomto prípade nekonečno.
- nobind – Klient týmto nastavením nemá stanovený konkrétny port, ktorý bude používať na klientskej strane, bude si voliť s určitého rozsahu
- persist-key – Toto nastavenie bude zachovávať načítané konfiguračné súbory od štartu až po ukončenie jednej relácie programu v systéme, keby sa súbory zmenili počas pripojenia, aplikácia to nebude brať na vedomie
- persist-tun – Rovnaké nastavenie ako persist-key ale pre virtuálny adaptér TUN

- ca ca_zamestnanci.crt – Názov certifikátu certifikačnej autority
- cert zamestnanec1.crt – Názov certifikátu klienta
- key zamestnanec1.key – Názov kľúča klienta
- ns-cert-type server – Toto nastavenie zabraňuje potencionálnym útokom pri falšovaní certifikátu servera
- cipher BF-CBC – Druh použitej metódy šifrovania
- keysize 128 – Veľkosť šifrovaného bloku
- comp-lzo – Použitie kompresie
- verb 3 – Úroveň hĺbky tvorby logov
- mute 20 – Ne zaznamenávanie prudko sa opakujúcich systémových správ

Po vytvorení konfiguračného súboru a zhromaždenie všetkých potrebných súborov sa stačí pripojiť spustením spustiteľného súboru v zložke inštalácie, prípadne odkazom na ploche alebo štarte. Pohodlné GUI rozhranie po chvíli vyhlási na paneli úloh úspešné pripojenie a parametre pripojenia si môžeme overiť vlastnosťami adaptéra virtuálneho pripojenia TUN služby OpenVPN na obrázku č. 24 :



Obrázok č. 24: Pridelená konfigurácia úspešného VPN pripojenia

Rovnakým spôsobom sa vytvorí nastavenie pre VPN okruh určený študentom. Vymenia sa iba certifikáty, kľúč a zmenia sa ich názvy v konfiguračnom súbore ešte sa zmení port serveru, tiež v konfiguračnom súbore.

2.2.6.2 Konfigurácia klienta v prostredí linux

Testovaný systém bol linux debian 5.6, kvôli podobnej stavbe jadra by bolo možné túto stavbu implementovať, s malými úpravami, aj v prostredí Suse, Mandriva, Fedora, Mint a Ubuntu.

Prvú časť opäť obnáša inštalácia OpenVPN balíka, v prípade linuxu sa to obmedzí na zadanie príkazu, v prípade užívateľského konta root, apt-get install openvpn, po jednom potvrdení je balík nainštalovaný. Druhá úloha plní konfiguračný priečinok */etc/openvpn/* konfiguračnými súbormi. Táto je však už nevykonáva pod užívateľom root ale pod zvoleným užívateľom. Do pričinka sa skopíruje certifikát certifikačnej autority, kľúčový pár klienta a konfiguračný súbor:

```
uzivatel@debian:/etc/openvpn# ls
client.conf  zamestnanec1.key  zamestnanec1.csr zamestnanec1.crt
```

Priečinok obsahuje potrebné súbory, konfiguračný súbor client.conf má úplne rovnaký tvar ako v prostredí Windows. Výborná vlastnosť OpenVPN je univerzálnosť, pre linuxové prostredie sa vykoná niekoľko ďalších úprav, editovanie súboru /etc/default, tu sa môže obsah buď zmazať alebo iba zapoznámkovať. Dôležitým riadkom, ktorý musí súbor obsahovať je **AUTOSTART="client"**, teda názov konfiguračného súboru. Po uložení súborDLu sa OpenVPN spustí príkazom

```
uzivatel@debian:/etc/default# service openvpn start
```

Nadviazanie OpenVPN pripojenia sa dá overiť prezeraním konfigurácie sieťových pripojení, s použitím konfiguračného súboru by mal po správnosti vzniknúť virtuálny sieťový adaptér TUN, výpis konkrétneho adaptéru po použití príkazu ifconfig:

```
uzivatel@debian:/etc/default#
tun0
Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
      inet addr:10.10.1.6  P-t-P:10.10.1.5  Mask:255.255.255.255
      UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Adaptér bol skutočne vytvorený a VPN okruh vytvorený

2.2.7 Testovacie scenáre

Žiadaný účel vytvorených VPN okruhov možno overiť splnením vyžadovaných potrieb pre ktoré bol scenár realizovaný.

Po vytvorení zamestnaneckého VPN okruhu má zamestnanec prístup do verejnej siete internet iba cez smerovač, aby nebolo potrebné definovať všetky rozsahy akademických sietí sveta, týmto riešením sa počíta ako člen akademickej siete automaticky. Po zapnutí prehliadača a spustení napríklad stránky http://www.whatismyip.com/ sa na nej zobrazuje informácia o IP adries cez ktorú sa na ňu pristupuje, zobrazuje sa IP adresa smerovača 147.175.79.140. Zamestnanec môže mať prístup do obidvoch podsietí, overiť to môže pomocou príkazu ping na ľubovoľné stanice v obidvoch podsieťach, stanice musia byť dostupné. Zamestnanec by mal byť tiež schopný pripojiť si vzdialené sieťové jednotky pomocou služby Samba.

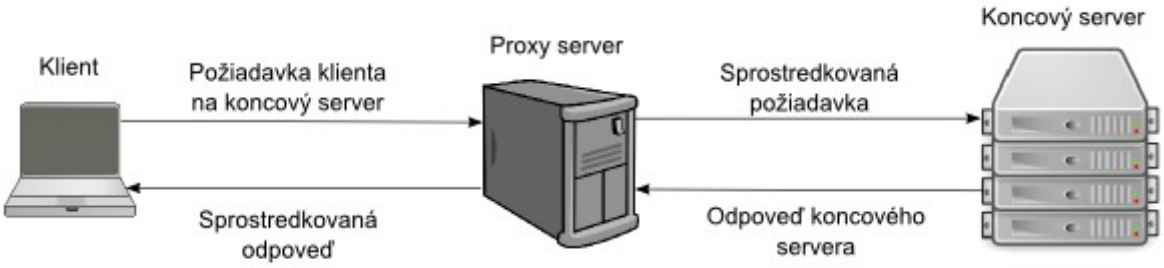
Študent po vytvorení VPN okruhu na stránke http://www.whatismyip.com/ vystupuje pod svojou lokálne pridelenou IP adresou , prístup môže mať len na stanice v študentskej podsieti, ktorú si môže overiť príkazom ping, zároveň nesmie mať prístup do zamestnaneckej podsiete, túto podmienku si môže tiež overiť príkazom ping, pričom žiadna stanica dostupná byť nesmie.

3 Implementácia proxy servera

3.1 Funkcie proxy servera

3.1.1 Základný princíp činnosti proxy servera

Proxy server sa ako názov technológie a aj ako súbor princípov činnosti radí k najstarším technológiám používanými v súvislosti s prevádzkou siete. Od vzniku svojej existencie na počiatkoch vývoja sieťovej technológie ušiel proxy server dlhú cestu plnú zmien a noviniek v možnostiach využívania, preto je dnes proxy možné použiť na široké spektrum činností. Samotný názov proxy server nemožno s určitosťou priradiť ku konkrétnej činnosti bez poznania podrobností jeho nastavenia a funkcie v konkrétnom bode siete. Ako najvšeobecnejší popis by sa dalo použiť slovo prostredník prípadne sprostredkovateľ, proxy server vystupuje v sieťovej komunikácii medzi dvomi koncovými bodmi a určitým spôsobom sa podieľa na ich vzájomnej komunikácii. Na rozdiel od základných sieťových prenosov existujúcich na druhej a tretej vrstve OSI modelu, funguje proxy server na vyšších aplikačných vrstvách, kde sa zapája do dátového toku. Proxy server vždy vystupuje ako klient aj server zároveň, koncový bod, ktorý inicioval spojenie sa na proxy server dopytuje ako na server, zatiaľ čo proxy server sa po iniciácii dopytuje na cieľový server ako klient, obdržanú odpoveď následne proxy server posunie pôvodnému bodu tak, ako je to vyobrazené na obrázku č.25 .

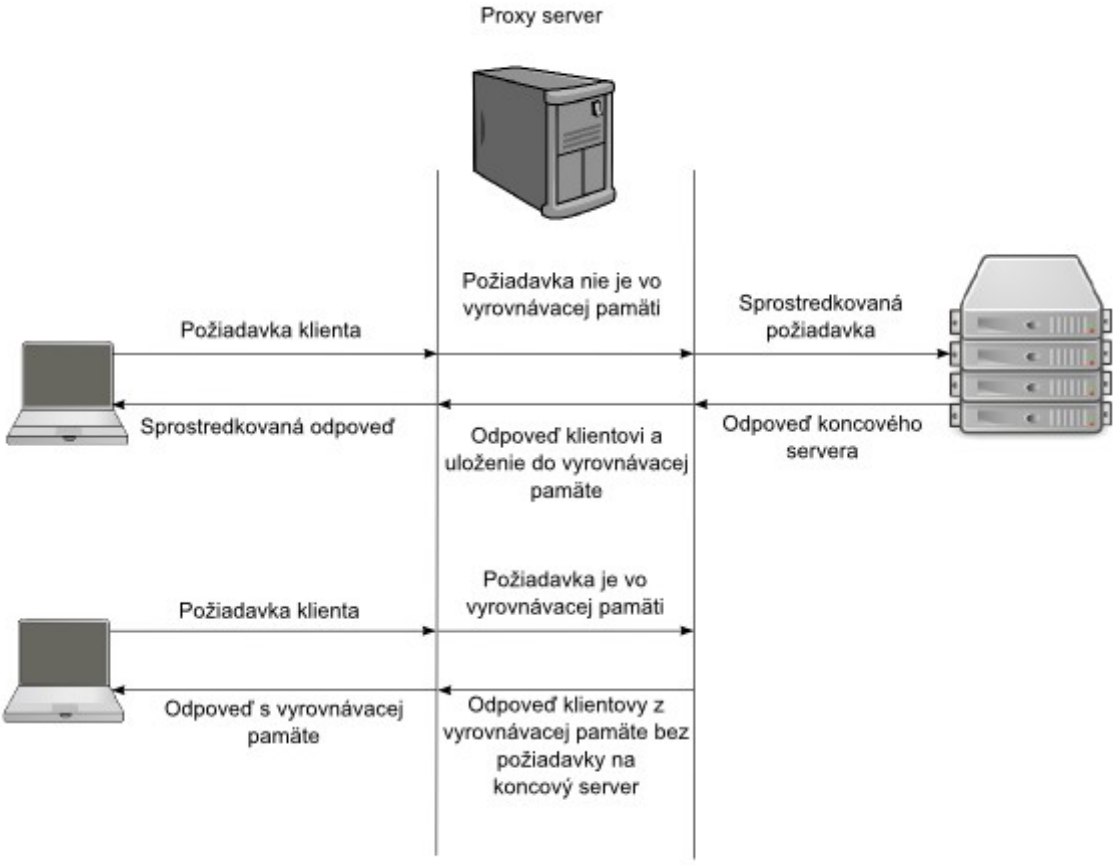


Obrázok č.25: Činnosť proxy servera

Tento náčrt komunikácii vytvára mnoho priestoru na spôsoby vykonávania tejto činnosti, ich princípy, možné nastavenia a vôbec široké možnosti, ktoré by sa dali realizovať. V skutočnosti ich proxy server môže vykonávať takmer všetky a jeho dlhý vývoj k tomu značne prispel. Hlboké spektrum možností takéhoto spojenia možno zhrnúť do dvoch veľmi hrubých oblastí, ktorými sú ukladanie do vyrovnávacej pamäte a filtrovanie. Pre objasnenie ich významu je potrebné vrátiť sa k dôvodom vzniku takejto technológie a prvých princípom jej fungovania [13],[16].

3.1.2 Ukladanie do vyrovnávacej pamäte (Cache)

Pri počiatkoch internetovej komunikácie existovalo, narozdiel od dnešnej takmer miliardy web stránok, iba niekoľko desiatok tisícov stránok pričom takmer všetky plnili vedecký, vojenský alebo štátny význam. V tej dobe bolo pripojenie dostupné hlavne pomocou analógového pripojenia dostupnej telefonickej siete a rýchlosť prenosu sa pohybovala na úrovni niekoľkých kilo bytov za sekundu. Stiahnuť si pri takejto rýchlosti akúkoľvek stránku bolo veľmi náročné, preto vznikol nápad ukladať už stiahnuté dáta na akýsi prechodný server v lokálnej sieti, odkiaľ by mohla byť daná stránka stiahnutá namiesto koncového servera v prípade opätovného dopytu z lokálnej siete. Takéto riešenie v danom čase spôsobilo dramatické zvýšenie dostupnosti webových stránok, nakoľko boli v rámci lokálnej siete dostupné z vyrovnávacej pamäte (Cache) lokálneho servera, kde sa uložili po prvom dopyte z lokálnej siete. Týmto riešením sa taktiež zväčšila priepustnosť vtedy malej siete internet keďže mnoho staníc sa nemuselo dopytovať na koncový server ale stačil im záznam na lokálnom servery, takýto server sa nazval proxy server. Proxy server v tomto prípade teda funguje ako prostredník, pričom sa klient na neho dopytuje s požiadavkou na konkrétnu web stránku. Proxy server najprv skontroluje, či sa na ňu už niekto nedopytoval a nemá ju uloženú vo vyrovnávacej pamäti, ak áno tak ju vráti klientovi bez dopytu na požadovaný server. V prípade zistenia neprítomnosti danej stránky vo vyrovnávacej pamäti sa proxy server dopytuje ako klient na server s vyžadovanou stránkou, tú nielenže poskytne klientovi, ktorý vydal pôvodný dopyt, ale taktiež ju uloží do vyrovnávacej pamäte pre prípadné ďalšie využitie, tento postup ilustruje obrázok č.26.



Obrázok č.26: Práca s vyrovnávacou pamäťou

Dlhodobé používanie tejto možnosti proxy serveru danú metódu značne zdokonalilo a moderný proxy server určený na ukladanie obsahu webových stránok do vyrovnávacej pamäti disponuje širokou paletou pokročilých nástrojov a algoritmov na správu uloženého obsahu. Medzi tieto metódy patria hlavne algoritmy, ktoré riešia ako dlho dáta uchovávať, v prípade vyplnenia vyrovnávacej pamäte vybrať, ktoré dáta sa už nepoužívajú a nahradiť ich, v prípade menenia obsahu zistiť odlišnosti medzi uloženou verziu a aktuálnou verziu [13],[16].

3.1.3 Cache a dnešné využitie

Napriek zaujímavému návrhu služby, ktorú tvorí proxy server ukladajúci do vyrovnávacej pamäte, má dnes toto ukladanie len malý význam. V minulosti tvoril web stránku väčšinou statický obsah, ktorý sa len málokedy menil. V dnešnej dobe dynamických web stránok s často jednorázovými adresami má cache len malý význam a dokáže ukladať iba niektoré všeobecné súčasti stránky ako najmä drobné statické časti, logá a iné obrázky. V kombinácii s dnešným bežne

dostupným vysoko rýchlostným pripojením je bežné, že desiatky užívateľov na rovnakej sieti navštívia tisícky rôznych stránok, ktoré sa nemusia vôbec zhodovať, pričom v takejto situácii nie je proxy server veľmi efektívny. Dnes jeho využitie možno nájsť len ako doplnkovú implementáciu v sieti organizácie, ktorá má pomalé pripojenie na internet, a teda chce minimalizovať objem prenesených dát a vyťaženosť linky. Ďalšiu implementáciu možno nájsť vo väčších organizáciach, v ktorých užívatelia často pristupujú na stránku, ktorej serverová časť je pomalá a server nestíha plne obsluhovať všetky dopyty. Zavedenie proxy servera môže znížiť vyťaženosť koncového servera [13],[16].

3.1.4 Filtrovanie

3.1.4.1 Vyhodnocovanie dátového toku

Filtrovanie je druhou veľkou funkciou proxy serveru, svojou podstatou predstavuje ďalšie logické využitie princípu fungovania proxy serveru. Vo funkcii prostredníka na aplikačnej úrovni sieťovej komunikácie musí proxy server vidieť obsah dopytu na vzdialený server a rovnako aj odpoveď z neho. Na základe tohto vzťahu sa otvárajú nové možnosti zasahovania, ovplyvňovania, blokovania, analýzy alebo iného vyhodnocovania dát prechádzajúcich cez proxy server. Možností, akým spôsobom to robiť, je rovnako veľa, ako dôvodov, prečo to robiť. V tomto ponímaní sa môže proxy server javiť ako škodlivý útočník, ktorý nazerá do dátového toku, avšak tok dát cez proxy server má vždy svoj význam a opodstatnenie v rámci organizačnej infraštruktúry [13],[16].

3.1.4.2 Proxy server ako firewall

Prechádzajúci dátový tok možno vyhodnocovať na základe viacerých hľadísk, medzi základné patrí vyhodnocovanie na základe zdroja dát a ich príjemcu. Vzhľadom na vyhodnocovanie dáta na aplikačnej vrstve, narozdiel od nižších úrovní OSI modelu, je možné hodnotiť dátový aj prihliadnuc na obsah dát. Obsahom dát najčastejšie býva vyžadovaná alebo vrátená adresa stránky, prípadne kľúčové slovo v názve stránky. Proxy server môže pri svojej činnosti funkčne zavádzať celé série obmedzení a blokovania konkrétnych IP alebo stránok. Táto činnosť je porovnateľná s činnosťou ako firewall, proxy server môže v určitom zmysle fungovať ako firewall, ale nasadiť ho iba pre tento účel nie je dobrý nápad. Pri vyhodnocovaní vyšších aplikačných protokolov zaberá proxy server viac výpočtového času ako nižší firewall, ktorý pracuje len s dátovými jednotkami tretej vrstve OSI modelu a pracuje tak skutočne rýchlo. Ďalším dôvodom prečo nenasadzovať proxy server ako firewall, je slabšie zabezpečenie proxy servera, ten môže totiž popri firewalle vykonávať

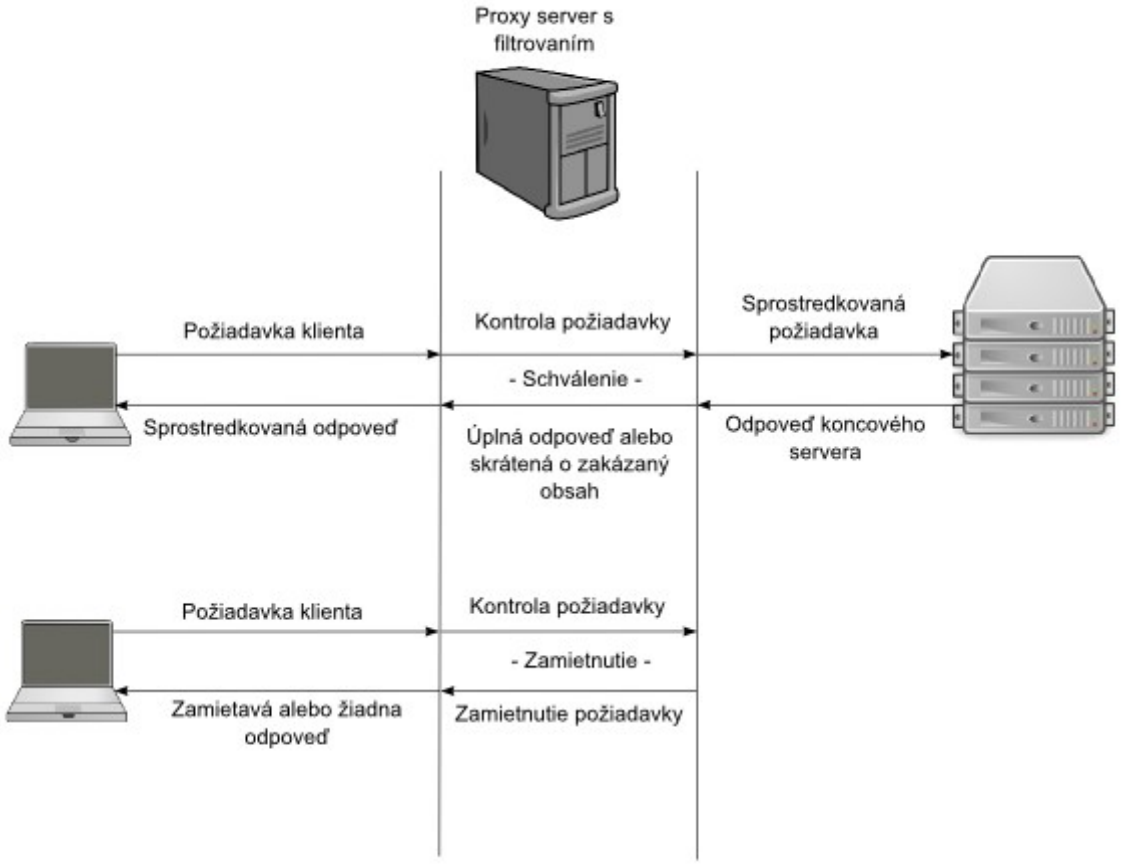
plno iných činností, ktoré môžu byť v rozpore s pravidlami firewallu a môže dochádzať k narušeniu bezpečnosti. Funkcie firewallu je vhodné pre proxy server zavádzať doplnkovo pre ostatné veci funkcie, ktoré vykonáva a tým mu priamo určiť aký dátový tok môže a nemôže prechádzať a ktorý sa má a nemá vyhodnocovať [13],[16].

3.1.4.3 Proxy server ako anti vírus

Narozdiel od funkcie firewallu dochádza pri snahe odhaliť škodlivý kód ku skutočnému vyhodnocovaniu obsahu prenášaných informácií. Proxy server môže a často aj plní bezpečnostnú funkciu odhaľovania útokov a prenosu škodlivého kódu. Toto možno ľahko dosiahnuť analýzou prenášaných dát a pátranie po vzoroch škodlivého kódu alebo útokoch tretích strán. Proxy server dokáže zistiť pokusy o infiltrácie, útoky, prenos vírov a iné bezpečnostné hrozby, zistené nebezpečné dáta môže zablokovať a jednoducho neposunúť ďalej pôvodnému klientovi. K pokročilejším metódam patrí priame odfiltrovanie nebezpečných častí a klientovi sa vráti iba bezpečná časť dát. Množstvo implementácii proxy serverov takto pôsobí ako proaktívna ochrana pred hrozbami z vonkajšej siete, dobre nasadený proxy server vie ochrániť klientov a aj servery pred radom rôznych útokov. Nevie však ochrániť klienta pred škodlivým účinkom aplikácie samotnej, aj možnosť zistenia hrozby v prenášaných dátach má svoj limit vzhľadom na protokol, ktorý proxy server spracováva [13],[16].

3.1.4.4 Blokovanie zvoleného obsahu

Proxy server tiež umožňuje blokovať rôzne súčasti prenášaných dát, najzákladnejším princípom je blokovanie konkrétnych URL alebo konkrétnych častí, ktoré vyhodnotila organizácia ako neprístupné pre vnútornú sieť. Bežnou záležitosťou je, že vo firemnej sieti dochádza k blokácii rôznych stránok, ktoré znižujú zamestnanecký výkon, takýto prístup je však značne zastaralý a neefektívny vzhľadom na možnosti dnešného internetu môže zamestnanec nájsť mnoho spôsobov ako obísť blokovanie jednej stránky. Zobrazenie takéhoto filtrovania je na obrázku č.27. Bezpečnostným prístupom v rámci blokovania prístupu môže byť zablokovanie rôznych aplikácii v rámci web stránky, napríklad blokovanie activeX alebo java scriptu v rámci zlepšenia bezpečnosti v rámci vnútornej siete a zníženie potencionálneho výskytu vírusov. Ďalším dôvodom pre blokovanie obsahu môže byť jeho vysoký stupeň podozrenia po vykonaní určitých analýz, prípadne môže byť obsah dát vyslovene nelegálny [13],[16].



Obrázok č.27: Využívanie obsahové filtra

3.1.5 Anonymita

Pri bežnej činnosti vykazuje proxy server vo svojom vzťahu zaujímavú vlastnosť, a to že bez ohľadu na klienta, ktorý od neho vyžiada obsah stránky a jeho IP adresu, ďalej proxy server pri sprostredkovaní žiadosti vystupuje pod svojou IP adresou. Pri bežnej činnosti proxy servera teda dochádza k akémusi skrývaniu všetkých jeho klientov za jednu jeho adresu. Táto najprv nepodstatná vlastnosť sa neskôr stala kľúčovým prvkom funkcie proxy serveru a možno povedať, že drvivú väčšinu všetkých proxy serverov na svete tvoria práve takzvané anonymné proxy, ktoré majú jediný účel, a to skryť pôvodnú IP klienta pri jeho prístupoch na sieti. Využívanie tejto vlastnosti má okrem účelu skrytia totožnosti aj účel pripájať sa cez proxy server, ktorý môže pristupovať do oblastí do ktorých bežný klient nemôže pristúpiť, typickým príkladom sú obmedzovania IP vzhľadom na geografickú alebo štátnu príslušnosť. Ďalším príkladom je nasadenie obmedzujúceho proxy serveru v zamestnaní, proti nemu sa dá bojovať iným proxy serverom, na ktorý sa dá pristúpiť bez blokácie a z neho následne pristupovať kdekoľvek. V rámci

vnútornej sieti organizácie má takáto anonymita aj bezpečnostný charakter, kde sú všetci klienti skrytí za proxy serverom, ktorý je vyzbrojený silnou bezpečnostnou politikou a chráni tak klientov citlivých na útoky [13],[16].

3.1.6 Variácie funkcií proxy servera

Možnosti výkonu činností proxy servera sú veľmi široké a možno ich nakonfigurovať veľmi podrobne a veľmi špecificky. V dnešnej dobe sa v rámci organizačných štruktúr nasadzujú vopred zvolené konfigurácie rôznych pôvodov, ktoré majú presne definovaný význam a chovanie. Okrem proxy serverov určených priamo na filtrovanie obsahu alebo ukladanie do vyrovnávacej pamäte je vhodné spomenúť ešte dve často využívané formy proxy serverov:

- **Reverzný proxy server** – Ide o proxy server, ktorý je umiestnený v organizačnej infraštruktúre, ktorá poskytuje veľké množstvo služieb v rámci serverovej časti. Takýto druh proxy servera je umiestnený pred servermi a funguje reverzne, teda namiesto bežného dopytu od klientov zachytáva automaticky všetky požiadavky na serveri za ním. Tieto požiadavky následne pošle ďalej na jeden zo serverov alebo vráti obsah, ktorý má uložený vo vyrovnávacej pamäti, vzhľadom na rovnaký charakter služby, ktorú servery poskytujú, je veľmi vysoká pravdepodobnosť žiadania stále rovnakého obsahu. Proxy server najčastejšie vráti z vyrovnávacej pamäte veľkú časť dát a len veľmi malú nutnú časť pošle aplikačným serverom za ním na spracovanie. Takto dochádza k prudkému zníženiu zátaže na serveroch, ktoré vykonávajú dôležitejšiu činnosť v rámci poskytovanej služby. Väčšinou sa pri tejto implementácii nachádza za proxy serverom veľké množstvo serverov plniacich rovnakú činnosť a proxy server často funguje aj ako rozdeľovač zátaže (load balancer), ktorý alikvotne prideľuje serverom úlohy. Toto rozloženie je veľmi typické a často využívanie v stredných a väčších organizáciách [13],[16].
- **Transparentný proxy server** – Ide o zvláštnu implementáciu proxy servera možnú len v špecifických prostrediach informačnej infraštruktúry organizácie. Najčastejšie sa implementuje tam, kde tvorí výstup z vnútornej siete iba jeden bod, ktorým je zároveň hlavný smerovač (router). Vo väčšine implementácii proxy serverov je nutná konfigurácia na strane klienta, prípadne poskytnutia informácie o existencii proxy servera, moderné aplikácie a prehliadače často dokážu zistiť možnosť využívať proxy server dostupný na sieti. Teda takmer vždy sa jedná o vedomé využívanie. Transparentný proxy server sa implementuje na hlavnom prístupovom bode a väčšinou nemá žiadny vplyv na činnosť

klientov. Rozdiel je v tom ,že každý klient prechádzajúci daným bodom proxy server využíva bez ďalších nastavení a o jeho existencii nemusí ani vedieť. Takto implementovaný proxy server môže vykonávať jeho bežné funkcie alebo byť takzvaný pasívny proxy server a iba zaznamenávať informácie o dátovom toku klientov pre potreby analýzy bez akéhokoľvek zasahovania [13],[16].

3.1.7 Vlastnosti proxy servera

Proxy server možno realizovať v takmer každej aplikačnej oblasti a pre takmer každý aplikačný protokol. Najčastejšie sa sa proxy server využíva pre HTTP protokol, pre ktorý pôvodne vznikol a jeho implementácie pre tento protokol sú najčastejšie a najbežnejšie. Proxy server však možno nasadiť aj pre iné základné protokoly ako napríklad SMTP alebo DNS. Pre každý protokol sa vyžaduje vlastný proxy server, nie je možné používať proxy server pre viacero protokolov, ak áno tak len v okrajových prípadoch pri veľmi podobných protokoloch. Pri nastavení a vlastnostiach proxy serveru je kritická otázka bezpečnosti a to hlavne zabezpečiť proxy server proti využívaniu nechcenými klientmi, ktorý môžu využívať základnú anonymnú funkciu proxy serveru, a tak v jeho mene vyvíjať škodlivú činnosť. Proxy server sa často stáva jediným prístupovým bodom pripojenia pre vnútornú sieť, preto treba dbať na rozumnú záťaž a funkcie proxy serveru. V mieste jeho implementácie môže v dôsledku zložitých politík dochádzať k veľkému vyťaženiu a tým zhoršeniu priepustnosti sieti a zužovaniu kapacity hlavného prístupového bodu. Nie je vhodné používať proxy server na viac ako jednu presne určenú úlohu, zloženými politikami môže dochádzať k ich vzájomnému narušovaniu a bezpečnostným problémom, navyše môže dojsť ku konfliktu s inými aplikáciami v sieti, ktoré majú na starosť danú činnosť [13],[16].

3.2 Inštalácia proxy servera

3.3 Ciele implementácie

Implementácia proxy servera má za cieľ inštaláciu zvoleného proxy servera v už funkčnom produkčnom prostredí s už nainštalovaným VPN serverom a ostatnými súčasťami v prostredí distribúcie pfSense. Konfigurácia proxy servera a má jednoduchú úlohu, a to zhromažďovať informácie o dátovom toku v uskutočňovanom prvkami sieťovej infraštruktúry do vhodných logov. Tieto logy budú priebežne vyhodnocované a spracovávané vytvoreným skriptom, ktorý bude vytvárať logy dostupné správe sieti. Pre bližšie vysvetlenie je potrebné rozobrať koncept činnosti proxy servera.

3.4 Vyžadovaná funkcionalita

Vyžadovanou funkciou je transparentný proxy server, ktorý pracuje na báze HTTP protokolu. Tento proxy server nemá svojou činnosťou nijako ovplyvňovať klientov a ich činnosť, systém bude iba vytvárať logy o činnosti klientov v rámci daného protokolu a tie budú ďalej spracovávané externým systémom na reportovanie. Inštalácia a testovanie prebieha v už vytvorenom testovacom prostredí, v ktorom bol nainštalovaný Open VPN server. Inštalácia je opäť vykonávaná na hlavný smerovač (router) v rámci distribúcie pfSense a jej dostupných súčastí.

3.4.1 Existujúce riešenia proxy servera

3.4.1.1 Proxy server na báze pfSsense

Inštalácia súčastí v rámci distribúcie pfSense sa musí uskutočňovať striktne v rámci balíčkov systémových súčastí dostupných pre túto distribúciu. Vzhľadom na charakter distribúcie by mohla mať inštalácia v rámci príkazového riadku nežiadúce až škodlivé dôsledky na stabilitu alebo funkčnosť celého systému. Preto musí byť vybraný proxy server dostupný v rámci množiny balíkov aplikácii dostupných k vybranej distribúcii. Prostredie pfSense ponúka všetky svoje dostupné balíky.

3.4.1.2 HAProxy

Jedná sa o rozšírenú open source aplikáciu, ktorej hlavné zameranie sa zaoberá vysokou dostupnosťou veľmi vyťažených serverov a balansovaním záťaže na týchto serveroch. Samotný názov je skratkou anglického „High Availability Proxy“. Projekt má začiatky už v roku 1996, kedy vznikal ešte ako základná implementácia HTTP proxy servera, neskôr pokračoval v roku 2000 už pod vlastným názvom a bližšie špecifikovaným zameraním. Od svojho začiatku ušiel veľkú trať a neskôr sa zameral na model, ktorého hlavným cieľom bolo v rámci jedného procesu zvládať vysoký počet simultánnych pripojení pri vysokej rýchlosti a vyťaženej prevádzke serverov. Svojím spôsobom sa implementácie HAProxy zameriava hlavne na funkcie reverzného proxy servera. Jedným s cieľov HAProxy je vysoký výkon pri nízkych zdrojoch informačnej infraštruktúry, a to najmä nízky výkon hardvérových prostriedkov organizácie. HAProxy má dlhodobo povesť vysoko stabilnej aplikácie v produkčnom prostredí, ktorá má vývoj zameraný na silnú stabilitu a žiadnu poruchovosť, samotný program je napísaný v jazyku C. Medzi základné funkcionality patrí rad techník, ktoré majú dosiahnuť maximálny výkon na danej platforme a využiť systémové zdroje na maximum. Patrí tam mnoho hĺbkových optimalizácií spracovania prichádzajúcich klientských

žiadostí, a to hlavne ich hromadné simultánne spracovanie. HAProxy má taktiež vysoko optimalizovanú prácu s pamäťou, čo dramaticky znižuje výpočtový čas. Rozvinutou súčasťou je tiež optimalizovaná analýza hlavičky HTTP požiadavky taktiež zameraná na výkon a rýchlosť spracovania. HAProxy sa vďaka veľmi prísnemu prístupu vo vývoji stal taktiež vysoko bezpečnou aplikáciou, ktorá nemá počas posledných 10 rokov zaznamenanú žiadnu medzeru v bezpečnosti aplikácie. Podpora ukladania do vyrovnávacej pamäte je taktiež implementovaná, nie je to však hlavný cieľ aplikácie, preto je táto časť skôr optimalizovaná na bezpečnosť než na výkon a pokročilé techniky spracovania. Svojou podstatou sa teda HAProxy proxy server snaží fungovať hlavne ako distribútor záťaže (Load Balancer) a v druhom rade ako reverzný proxy server. Cieľovými platformami sú hlavne linuxové distribúcie, nakoľko bol HAProxy vyvíjaný na linuxovom jadre. V rámci svetovej infraštruktúry je tento projekt implementovaný na viacerých stránkach celosvetovo známych spoločností zahrňujúcich Reddit, Tumblr a Twitter, čo samé o sebe svedčí o jeho funkcionalite. HAVProxy server sa vzhľadom na jeho zameranie optimalizácie výkonnosti pri reverznom proxy nehodí na vyžadovanú funkcionalitu transparentného proxy servera[15].

3.4.1.3 HAVP antivirus

Jedná sa o ďalšiu implementáciu HTTP proxy servera pričom hlavný dôraz sa kladie analýzu celkového dátového toku pri HTTP protokole a jeho analýzu vzhľadom na škodlivý kód, samotný názov je z anglického „HTTP Anti Virus Proxy“. Projekt datuje začiatky v roku 2005 kedy bol ako prvý krát vydaný pod licenciou GNU/Linux ako balík zdrojového kódu pre platformu linux, neskôr sa možnosti platforiem rozšírili a projekt začal fungovať pod licenciou GPL. Ide o svojou podstatou o veľmi smelý a ojedinelý projekt, ktorého cieľom bola online analýza HTTP prenosov s čo najmenší vplyvom na rýchlosť prenosu a činnosť klientov. HAVP antivirus v sebe spája techniky z veľkého množstva malých, niekedy už neexistujúcich, projektov zameraných na bezpečnosť, skenovanie vírusov, firewally a iné bezpečnostné prvky. HAVP antivirus proxy server sa zameriava na zisťovanie vírusov a inak škodlivého kódu hlavne v aktívnych častiach web stránok, ktorými sú napríklad java script alebo active X prvky. Hlavným cieľom je však výkon pri tomto skenovaní a neovplyvňovania činnosti klientov. Medzi tieto techniky patrí hlavne možnosť neblokovania sťahovaných súborov a ich nepretržité sťahovanie bez nutnosti prerušení kvôli analýze. Druhou hlavnou technikou je hladké skenovanie všetkých súčastí HTTP prenosu ako sú obrázky a iné dáta, taktiež hlbšie skenovanie dynamických stránok. Princípom fungovania je vytvorenie malého

dočasného súboru, s veľkosťou definovanou konfiguráciou, po obdržaní odpovede z koncového servera na dopyt klienta. Tento súbor pôsobí ako malé dopravné oneskorenie a ihneď počas prijímania odpovede zo strany serveru sú do súboru zapisované dáta, ktoré sú zároveň skenované v rámci celej dĺžky súboru. Dáta sú ihneď po skenovaní posielané klientovi, takýmto spôsobom za cenu veľmi malého dopravného oneskorenia, možno analyzovať sťahované súbory simultánne. Nevýhodou takého prístupu ostáva fakt, že ak je vzor vírusu väčší ako veľkosť dočasného súboru, môže byť tento vírus nepovšimnutý a ďalej prenesený. Bližšie popisovať heuristiku algoritmov tejto aplikácie pri vyhľadávaní nebezpečného kódu nemá veľký význam, zaujímavé sú však ostatné funkcie, ktorými disponuje. Základnými funkciami, okrem aktívneho skenovania všetkého HTTP dátového toku, sú možnosti aktívneho skenovania množstva klientov naraz, zvýšený dôraz na skenovanie web stránok spojených s heslom a prihlasovaním, podpora transparentného módu proxy serveru a podpora iného proxy serveru. Ďalšou výbornou funkciu je aktívne budovanie takzvaných white listov a black listov, ktoré môžu byť zdielané s komunitou. Pri podpore iného proxy serveru je možné s HAVP skĺbiť iný proxy server, ktorý bude vykonávať činnosti ako spracovanie, vyhodnocovanie, filtrovanie, ukladanie do vyrovnávacej pamäti a inú činnosť bežne spojenú s proxy serverom. HAVP sa v takom prípade transformuje len na malú súčasť materského proxy serveru pričom sa zameriava len na analýzu dát. Takúto podporu má rozvinutú hlavne pre Squid proxy server. Samotný HAVP antivirus proxy server nemá podporu ukladania cache, reverzného proxy serveru, distribútora záťaže alebo iných funkcií, hlavným zameraním tohto proxy servera ostáva aktívne skenovanie škodlivého kódu. HAVP antivirus proxy server má dnes podporu väčších antivírusových spoločností, ktoré mu umožňujú vykonávať jeho analýzu spolu s analýzou ich softvéru. HAVP antivirus sa však vzhľadom na jeho obmedzené možnosti funkcie ako transparentného servera a vytvárania logov nehodí na vyžadovanú funkcionalitu [14].

3.4.1.4 Squid

Jedná sa o široko rozšírenú aplikáciu s veľkým spektrom funkcií podporujúcich takmer každú činnosť proxy servera. Projekt začal ako aplikácia proxy serveru zameraná na ukladanie do vyrovnávacej pamäti pôvodne vyvíjaná na Colorado University, neskôr bol projekt financovaný národnou vedeckou nadáciou a dokončený na University of California, San Diego. Projekt bol vydaný pod verziou 1.0.0 v roku 1996. Dnes je squid vyvíjaný výlučne skupinou dobrovoľníkov a vydaný pod licenciou GPL. Squid proxy server, alebo jeho rozmanité súčasti sú dnes súčasťou obrovského množstva iných aplikácií, ktoré sa využívajú v rámci sieťovej infraštruktúry

neopísateľného množstva organizácií. Samotný squid dnes využívajú stovky poskytovateľov Internetu na celom svete pre jeho robustnosť a veľké možnosti konfigurácie. Squid ako taký je HTTP proxy server, ktorý však môže podporovať HTTPS, FTP a iné protokoly čo s neho robí oproti iným riešeniam podstatne komplexnejšiu a viac využiteľnejšiu aplikáciu. Squid môže vykonávať funkciu ako bežného proxy serveru tak aj reverzného proxy serveru, distribútora záťaže, filtrovania obsahu, ukladania do vyrovnávacej pamäte a iné funkcie proxy serveru, všetky môže vzhľadom na rozšírenú bázu techník vykonávať plnohodnotne. Najviac cenenou funkciou Squid proxy serveru sa javí schopnosť ukladania do vyrovnávacej pamäte (cache). Napriek dnešnej dobe, kedy dynamické stránky a rozmach množstva internetových stránok spôsobil veľmi malé využitie vyrovnávacej pamäte proxy serverov, squid dokázal toto ukladanie optimalizovať a dosahovať vysoké percento využiteľnosti vyrovnávacej pamäte. Samotný vývojári squidu uvádzajú až 75 % využiteľnosti vyrovnávacej pamäte pri implementácii reverzného proxy serveru, pri nárazovej záťaži udávajú využiteľnosť blížiacu sa k 100 %, čím sa dramaticky zníži vyťaženie prenosovej kapacity. Squid sa teda pri ukladaní do vyrovnávacej pamäte zameriava na zložité algoritmy vyhodnocovania dát na uloženie, taktiež si necháva záležať na algoritmoch, ktoré vyhodnocujú ponechanie dát v pamäti. Ďalšou výraznou funkciou Squid je filtrovanie a politiky prístupu, kde je možné definovať prístupné a neprístupné URL, množiny klientov, ktorý sa majú filtrovať na základe rozličných politík. V rámci proxy servera, ktorý pracuje na viacerých protokoloch, je možné definovať aj obmedzenia na úrovni portov a čiastočne tak preberať úlohu firewallu. Na aplikačnej vrstve je možné definovať pravidlá na základe povoleného množstva dáta, maximálne rýchlosti, dokonca je možné definovať politiky pre sťahovanie rôznych druhov súborov a veľkostí súborov. Bezpečnostná politika Squid taktiež nezaostáva, implementovaná podpora môže pracovať zo súčasťami ako LDAP, RADIUS a iné prihlasovacie aplikácie tretích strán ale aj s vlastnými možnosťami autentifikácie pre využívanie proxy servera. V rámci modelov fungovania proxy servera môže Squid fungovať v ako v reverznom tak aj transparentom móde ale uplatní sa aj ako distribútor záťaže. Okrem toho má squid bohaté nastavenia na sieťovej úrovni, čo sa týka nastavení jeho zdrojov a metód fungovania na nainštalovanej platforme. Squid taktiež disponuje rozšíreným spôsobom vytváranie logov, má bohaté možnosti nastavenia informácií, ktoré má zaznamenávať, ale aj tvaru, v akom ich má zaznamenávať. V rámci činnosti Squid servera sa vyvinula komunita aplikácií, ktoré pracujú čisto na výsledkoch jeho práce alebo priamo spracovávajú logy, ktoré produkuje do rôznych podôb. Samotný squid má podporu pre veľa konkrétnych platforiem operačných systémov, okrem linux a unix platforiem aj pre platformu windows. Aplikácia Squid proxy server sa najmä vďaka

pokročilým metódam vytváraním logov radí ako najvhodnejšia voľba na inštaláciu v prostredí platformy pfSense [12],[17].

3.4.1.5 Ostatné dostupné riešenia

Balíky aplikácií dostupných pre pfSense zahŕňajú aj niektoré menšie riešenia proxy serverov často spojené s inými aplikáciami [11].

- **Imspector** – Proxy server nepracujúci na báze HTTP ale na báze portov spoločných pre IM „Instant Messagging“ ako sú ICQ, Jabber, MSN, Skype a podobne. Táto aplikácia je zameraná najmä na blokovanie alebo filtrovanie týchto aplikácií a taktiež hlavne na odchytávanie obsahu v týchto aplikáciách pokiaľ nie sú zabezpečené. Tento druh proxy servera vôbec nepracuje na báze HTTP preto nemá zmysel nad ním uvažovať ako nad možným riešením.
- **Mod_security proxy server** – Implementácia open source proxy servera, ktorá má za cieľ fungovať ako čiastočný firewall na aplikačnej vrstve HTTP a zabraňovať tak útokom na web. Taktiež poskytuje možnosti analýzy web obsahu v reálnom čase. Používa sa hlavne ako súčasť aplikácie Apache pre zvýšenie jej bezpečnosti. Táto aplikácia svojím fungovaním nespĺňa kritériá vyžadovanej konfigurácie.
- **Siproxd** – Open source projekt, ktorý má využívať schopnosť proxy servera zakrývať svojich klientov a pôsobiť tak čiastočne ako NAT. Táto funkcia je výhradne pre SIP klientov a uplatňuje sa tak pre technológiu IP telefonovania. S vyžadovanou konfiguráciou má len veľmi málo spoločné.

3.4.2 Inštalácia Squid

3.4.2.1 Voľba proxy servera Squid

Vzhľadom na svoju robustnosť, rozšírenosť a veľké možnosti konfigurácie sa ako najvhodnejšia na použitie javí aplikácia Squid proxy server. Dôležitým aspektom pri výbere je aj pravdepodobnosť zaniknutia v budúcnosti alebo tiež možnosť ďalšieho vývoja. Pri menších projektoch totiž často dochádza k ukončeniu ich vývoja a podpori, čo má za následok v rámci informačnej infraštruktúry organizácie nutné zmeny aplikácií čo vedie k zvýšeným nákladom alebo problémom pri migrácii. Squid sa vzhľadom na vznik a súčasnosť javí ako stabilná, výkonná a dlhotrvajúca platforma s dobrými vyhliadkami na podporu v budúcnosti. Pre pfSense je

momentálne dostupná stabilná verzia Squid 2.7.9. Pre inštaláciu sa využije už vytvorené testovacie prostredie na ktorom bol nainštalovaný VPN server [12],[17].

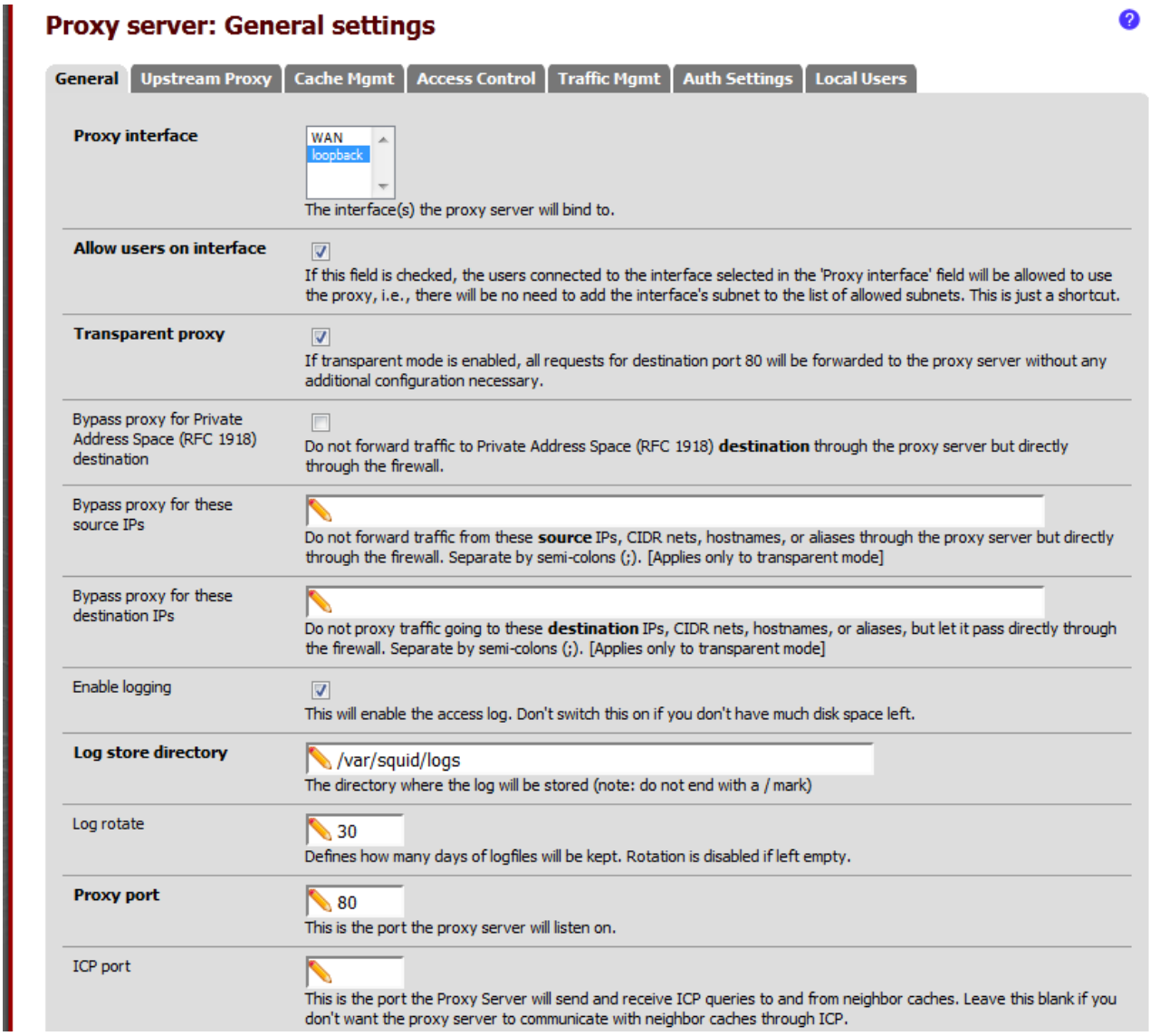
3.4.2.2 Inštalácia

Samotná inštalácia je v prostredí pfSense veľmi pohodlná vďaka GUI prostrediu a časti package manager, kde si stačí zvolený balík jedným kliknutím nainštalovať. Presný postup zahŕňa viacero krokov:

1. V rámci lišty z hlavnej ponuky treba zvoliť položku „System“ a pod položku „packages“.
2. V novom vzniknutom okne treba v časti „Available Packages“ nájsť v zozname dostupných balíčkov balíček označený „squid“ v ostatných poliach je možné hneď prečítať podrobnosti činnosti aplikácie a jej verzii, čím sa dá uistiť o správnosti balíčku.
3. V rámci nájdenej položky v zozname balíčkov stačí kliknúť na posledný symbol v danom riadku symbolizujúci inštaláciu, aplikácia bude následne inštalovaná. Po ukončení inštalácie systém sám vykoná všetky potrebné úpravy, nastavenia, prípadný reštart niektorých súčastí, pre možnosť okamžite používať novo nainštalovaný balík.

3.4.2.3 Konfigurácia Squid proxy servera

Konfiguráciu je možné opäť vykonať pomocou GUI prostredia pfSense. Samotný konfiguračný súbor Squid proxy servera má veľmi veľké možnosti konfigurácie, prostredie pfSense v rámci GUI nastavení vybrané tie najvhodnejšie voľby konfigurácie. Voľby konfigurácie, ktoré nie sú zobrazené v GUI, je možné doplniť do konfiguračného súboru pomocou textu, ktorý sa dá doplniť do textového pola v GUI a následne je pridaný do konfiguračného súboru. Samotnú konfiguráciu proxy serveru môžeme v GUI nájsť v časti „Services“ na hlavnej lište a následne v jej pod časti „Proxy server“, časť tejto ponuky možno vidieť na obrázku č. 28. Dostupné nastavenia Squid značne presahujú nastavenia potrebné na vyžadovanú konfiguráciu, v dostupnom GUI prostredí stačí nastaviť zlomok zobrazených možností. Nastavenia v rámci v karty „General“:



Obrázok č.28: Ponuka "General"

- **Proxy interface** – V tejto možnosti treba označiť všetky adaptéry, na ktorých má proxy server prijímať požiadavky od klientov. Táto položka je mimoriadne dôležitá pre rôznu voľbu funkcie proxy servera. V prípade transparentného proxy servera sa výberom adaptérov určujú adaptéry, na ktorých budú všetky žiadosti automaticky spracovávané proxy serverom. V tomto prípade môžu byť vybrané akékoľvek adaptéry na vnútornej sieti avšak v žiadnom prípade nie adaptér WAN, ktorý je spojením s vonkajšou sieťou. Pokiaľ sa zvolí ako adaptér na ktorom má proxy server pracovať, tak budú spracovávané aj všetky požiadavky z vonkajšej siete na prípadné web servery vo vnútornej sieti, proxy server nadobudne časť funkcie reverzného proxy servera. Avšak bez ďalších nastavení pre túto

situáciu spôsobí toto nastavenie zablokovanie dostupnosti služieb na vnútornej sieti pre vonkajší svet, bez toho aby ktokoľvek z vnútornej siete niečo spoznal. Navyše zavedenie adaptéra WAN do množiny proxy server adaptérov spôsobí, že budú môcť jedinci z vonkajšej siete využívať proxy server a skrývať sa zaň, čo je vyslovene nebezpečná situácia s bezpečnostného hľadiska. Označiť je potrebné všetky adaptéry, na ktorých je možná prítomnosť klientov, neoznačené musia ostať adaptéry WAN, loopback a adaptéry na ktorých sa činnosť proxy servera nevyžaduje.

- **Allows users on interface** – Pomocou tejto voľby sa dajú povoliť všetci klienti nachádzajúci sa na danom adaptéri automaticky. Nie je potrebné ich povoľovať ručne v rámci prístupovej politiky a dá sa tak vyhnúť predvolenému blokovaniu neznámych užívateľov. Voľbu zaškrtnúť.

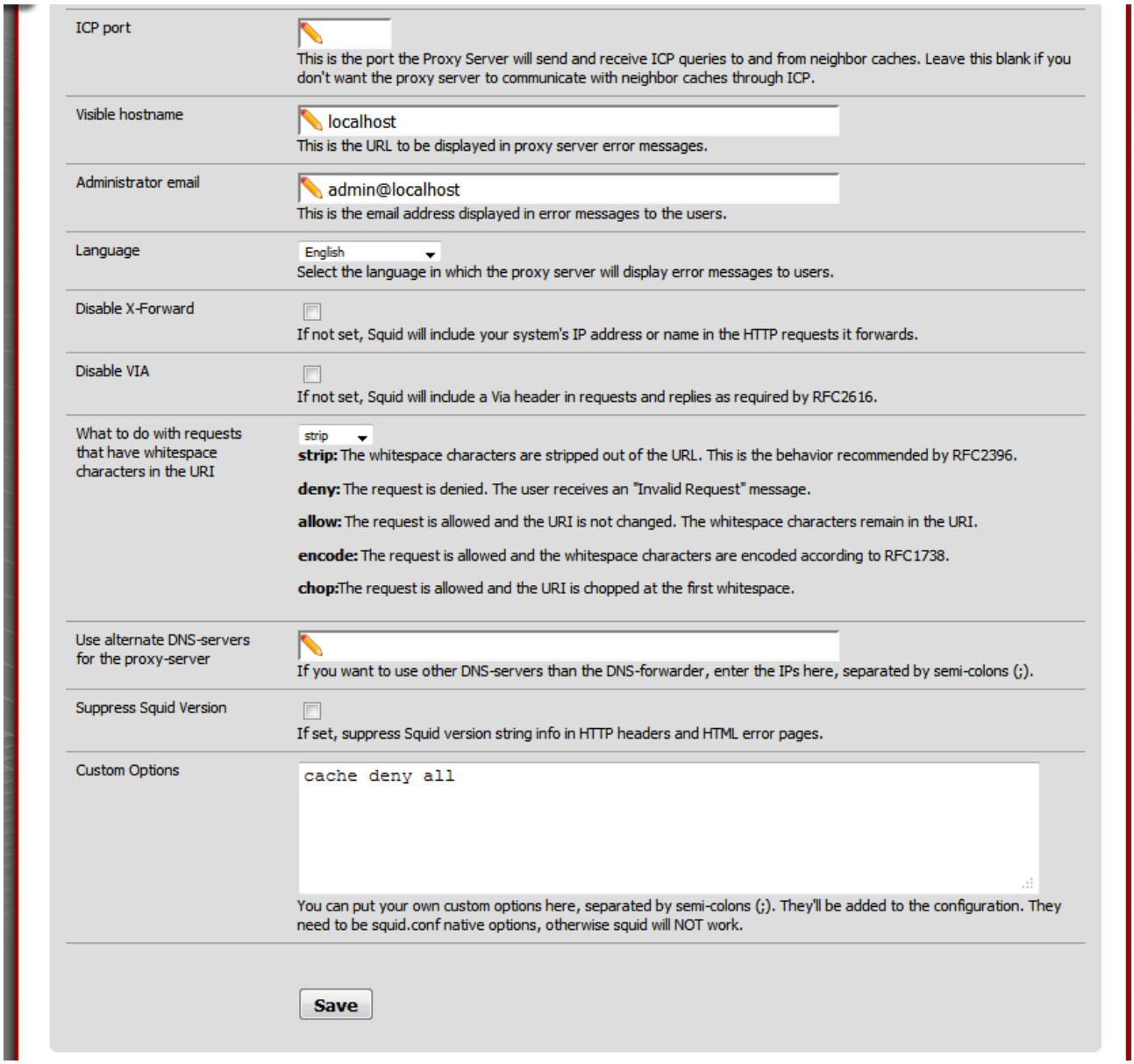
- **Transparent proxy** - Ako napovedá názov, voľbou tejto možnosti sa zvolí, či má fungovať proxy server v transparentom móde. Voľbu je nutné zaškrtnúť.

- **Enable logging** - Táto položka zapne vytváranie logov informácií dátových HTTP transakcií klientov transparentným proxy serverom.

- **Log rotate** – Číselná hodnota vyjadruje počet dní zachovania logov, každý log zaznamenáva presne jeden deň informácií. Počet dní je potrebné vhodne zvoliť na základe objemu dát v logoch a dostupného miesta pričom treba brať do úvahy aj minimálny nutný počet dní uchovávania logov.

- **Proxy port** – Port na ktorom má proxy server prevádzkovať transparentné chovanie, v tomto prípade štandardný HTTP port 80.

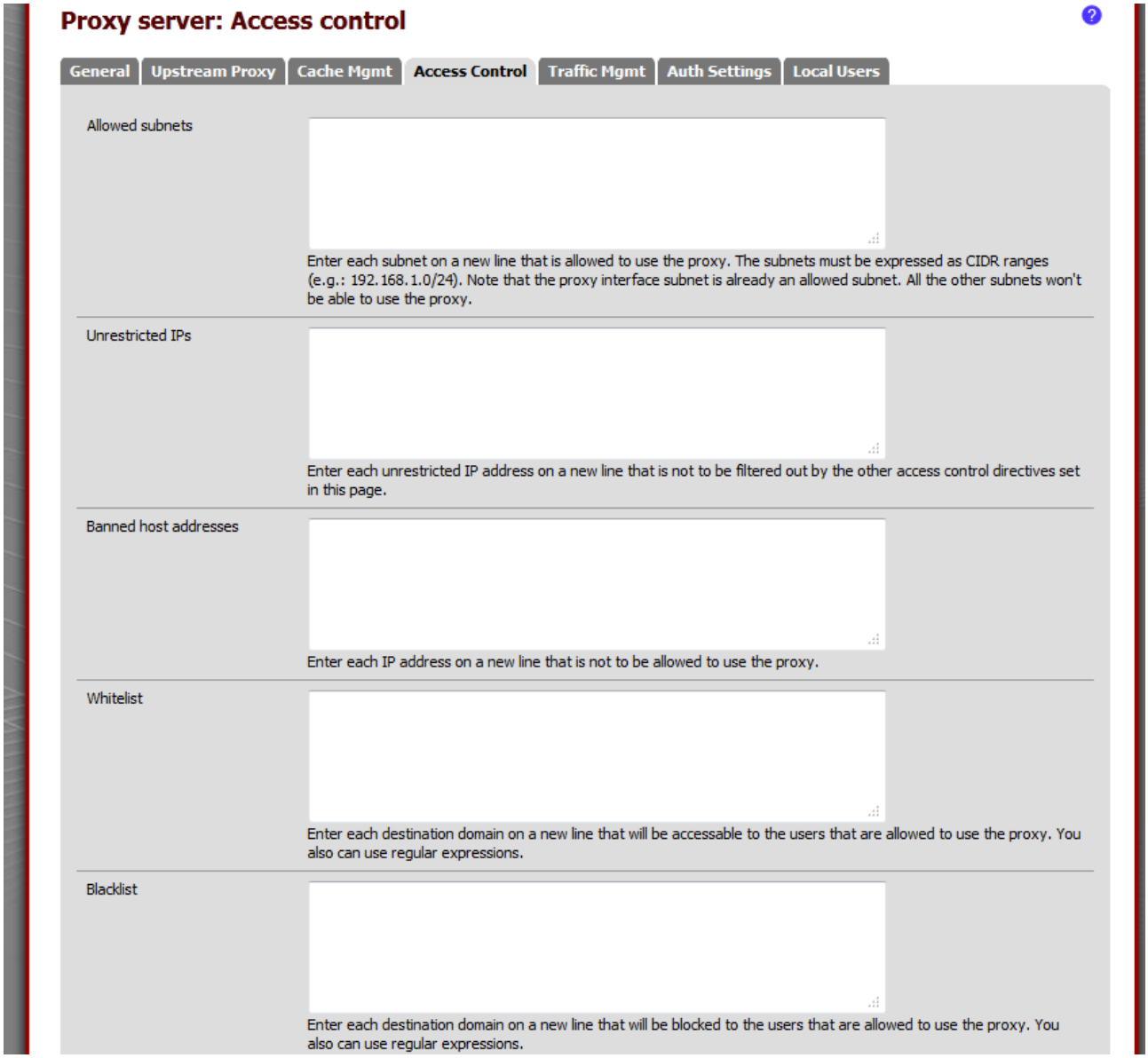
- **Custom Options** – Toto pole pohodlne rieši absenciu niektorých podrobných nastavení Squid v rámci GUI, obsah tohto poľa je jednoducho pridaný do konfiguračného súboru. Predpokladá sa že užívateľ vie čo robí. Do poľa sa doplní text „cache deny all“, toto nastavenie spôsobí vypnutie funkcie ukladania do vyrovnávacej pamäti, textové pole možno vidieť na obrázku č.29. Aj keď môže byť táto funkcia užitočná, na danej topológii nemá veľký význam a mohla by zbytočne oberať hlavný vstupný prvok do siete o výpočtovú kapacitu. Navyše nie je súčasťou vyžadovanej konfigurácie.



Obrázok č.29: Záver ponuky "General"

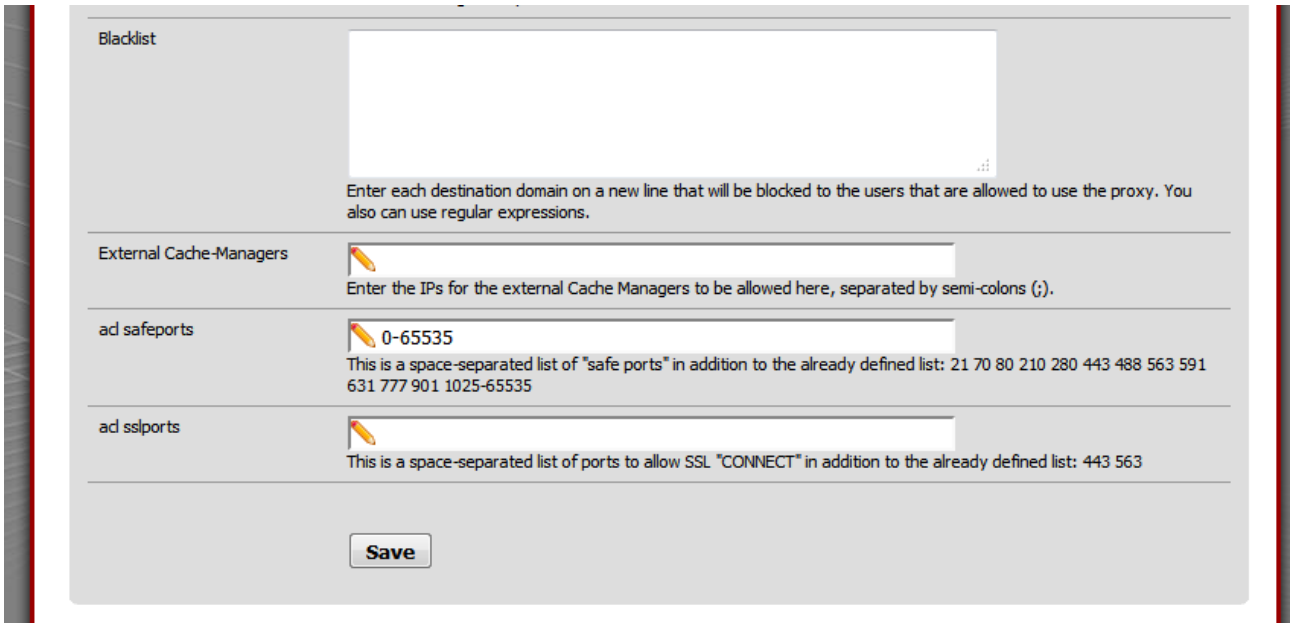
V rámci karty „General“ je dostupných veľa iných nastavení, tieto však nie sú potrebné pre vyžadovanú konfiguráciu a navyše nepotrebujú meniť stav vzhľadom na predvolenú konfiguráciu.

Posledné nastavenie sa uskutoční v rámci karty „Access control“ (Obrázok č.30), v tejto karte je rôzne politiky prístupu, keďže funkcia firewallu sa na transparentom proxy serveri nevyžaduje, ostanú tieto nastavenia prázdne až na jedno.



Obrázok č.30: Ponuka "Caption control"

- **acl safeports** – Nastaví sa hodnota „0-65535“ (Obrázok č.31) , týmto sa potlačí predvolené chovanie proxy servera povoľovať prenos informácií iba na niektorých portoch a proxy server úplne prestane plniť funkcie firewallu, ktoré ostanú na skutočnom firewalle.



Obrázok č.31: Záver ponuky "Caption control"

Nastavenia iných častí v rámci vyžadovanej konfigurácie nie sú potrebné. Zvyšné karty ponúkajú možnosti podrobného nastavenia politík filtrácie obsahu, ukladania obsahu do vyrovnávacej pamäte alebo možnosti autentifikácie užívateľov. Tieto časti sú nepotrebné pre funkciu transparentného proxy servera a predvolene sú vypnuté, čo eliminuje potrebu dodatočne ich konfigurovať.

3.5 Výsledný stav konfigurácie

Po správnej konfigurácii Squid proxy servera vznikol stav, kedy sa hlavný smerovač (router) v sieti stal transparentným proxy serverom. Implementovaný proxy server funguje ako doplnková aplikácia na hlavnom smerovači, ktorý zároveň plní funkciu hlavného a jediného prístupového bodu do vonkajšej siete. Ide o HTTP implementáciu proxy serveru, ktorý všetky požiadavky klientov na vnútornej sieti najprv zaznamená a následne zašle koncovému serveru. Proxy server funguje bez toho aby bola potrebná akákoľvek zmena konfigurácie u klientov. Hlavný výsledok implementácie spočíva v tvorbe logov, ktoré vytvára Squid na základe prebiehajúcej dátovej komunikácie klientov na porte 80, tieto informácie sú zbierané a poslúžia ako zdroj informácii pre potreby reportovania

4 Implementácia reportovania

4.1 Účel reportovania

Účelom reportovania je pomocou informácií získaných z implementácie transparentného HTTP proxy servera vytvárať informačné štatistiky súvisiace so stránkami navštívenými klientmi. Spracovávané informácie sú štatistického charakteru a to hlavne veľkosť prenesených dát, počet navštívených stránok a celkový počet unikátnych názvov navštívených stránok za zvolené obdobie. Informácie je možné vyhodnocovať buď vzhľadom na jednotlivých klientov alebo globálne pre všetkých klientov, pričom sa vyhodnocujú rovnaké objektívne kritériá ako pre jednotlivcov. Vypracované hlásenia sú následne distribuované jednotlivým klientom ako zdroj informácií pre nich samotných. Výstup z reportovania tvorí HTML súbor, ktorý pomocou prehľadných tabuliek poskytuje náhľad na činnosť klienta s dobrou výpovednou hodnotou.

4.2 Možnosti zobrazovania

Zvolený prístup reportovania vytvára prehľadné záznamy, ktoré sú podávané textovou formou. Väčšinu hlásenia tvoria HTML tabuľky spracovaných informácií zoradených podľa relevantnosti. Tento prístup bol zvolený kvôli jednoduchosti hlásenia, okamžitej prehľadnosti a možnosti spracovávať vytvorený report externými programami. Takéto reportovanie však nezahŕňa grafickú formu. Pokiaľ ide o grafickú formu, tak v rámci Squid proxy servera je voľne dostupných viacero programov vytvárajúcich zložité grafické výstupy na základe dostupných výstupných súborov s činnosti Squid proxy servera. V rámci pfSense distribúcie sú dostupné hneď dva balíky aplikácii, ktoré vykonávajú túto činnosť[11]:

- **LightSquid** – Aplikácia vytvára jednoduché grafy pre jednotlivých klientov, ktoré zahŕňajú navštívené stránky, prenesené dáta, rôzne percentuálne hodnotenia. LightSquid ďalej môže vytvárať rôzne globálne náhľady a tabuľky všetkých užívateľov. Samotný prístup aplikácie je zameraný na jednoduchosť a existenciu iba tých najzákladnejších funkcií, ktoré sa naozaj využívajú.
- **Sarg** – Aplikácia podobná LightSquid. Tiež poskytuje grafické výstupy činnosti klientov, ale vo viac podobách a zložitejších štruktúrach. Okrem toho poskytuje množstvo tabuliek s rôznymi analýzami a vyhodnoteniami činnosti jednotlivých alebo všetkých klientov.

4.3 Zdroj spracovávaných dát

Kmeňovým zdrojom dát sú logy so záznamami o činnosti, ktoré vytvára Squid svojou činnosťou transparentného proxy servera. Z informácií, ktoré môže zbierať, je najdôležitejší súbor „access.log“, ktorý obsahuje potrebné informácie. Pokiaľ nie je zvolené inak, tento súbor naberá štruktúru v ktorom sa vyskytuje jeden model riadkového záznamu, ktorý mení svoje vlastnosti podľa toho aký požiadavok klienta na stránku sa zaznamenal. Predvolený formát má 10 základných polí, ktoré nasledujú v riadku za sebou v nasledovnom poradí:

1. **Čas** – Ide o časový záznam vo formáte Unix, čiže počet sekúnd ktoré uplynuli od 01.01.1970, tento časový formát je veľmi pohodlný a vhodný na spracovanie
2. **Trvanie spracovania** – Záznam toho, ako dlho požiadavka vyťažovala vyrovnávaciu pamäť alebo tiež ako dlho trvalo spracovanie a vyhodnotenie požiadavky.
3. **Adresa klienta** – IP adresa klienta ktorý vyslal požiadavku
4. **Kód výsledku** – Dve hodnoty oddelené lomítkom, ide o informáciu či požiadavka mohla byť spracovaná záznamom vo vyrovnávacej pamäti alebo nie, druhá hodnota je HTTP kód prenosu. Pri vypnutí ukladania do vyrovnávacej pamäti bude tento kód samozrejme vždy rovnaký.
5. **Velkosť** – Celková veľkosť prenesených dát v rámci požiadavky v bytoch.
6. **Metóda vyžiadania** – Metóda prenosu vyžiadania webovej stránky, väčšinou GET alebo POST.
7. **URL** – Adresa web stránky na ktorú sa pristupovalo, zaznamenáva sa celá URL
8. **RFC931** – Informácia o klientovi špecifikovaná v danom RFC, predvolene je vypnutá. Používa sa len pri určitých metódach autentifikácie klientov, v logu sa namiesto tohto poľa zjaví len pomlčka.
9. **Hierarchický kód** – Informácia zložená z dvoch častí oddelených lomítkom. Prvá časť hovorí o tom, ako bola požiadavka vybavená, možnosti sú buď priamo zo servera, z vyrovnávacej pamäte, presmerovaním alebo mnoho iných. Najčastejšou možnosťou je priame spojenie zo servera. Druhým údajom je IP adresa koncového servera.
10. **Typ obsahu** – Obsah prenosu tak ako je opísaný v http hlavičke prichádzajúcej z koncového servera späť ku klientovi, najčastejšie vo forme „text/html“.

Z uvedených dátových polí majú najdôležitejšiu výpovednú hodnotu polia 1,3,5 a 7. Teda polia obsahujúce čas žiadosti, jej veľkosť, IP adresu klienta a vyžiadanú URL. Ostatné polia nemajú pre zvolený druh reportovania dôležitý význam

4.4 Koncept spracovania

4.4.1 Voľba prostriedkov

Spracovávanie bude vykonávané vlastným, na presný účel zostrojeným skriptom, ktorý bude dané hlásenia pre reportovanie vytvárať. Ako jazyk v ktorom bude skript napísaný bol zvolený perl a to hlavne pre jeho existujúcu inštaláciu v produkčnom prostredí kde je súčasťou iného aktívne sa používajúceho balíčka. Okrem toho je perl dlhodobou súčasťou takmer každej distribúcie linuxových alebo unixových systémov a na týchto systémoch sa dlhodobo a úspešne používa. Skript bude podľa potreby alebo pravidelne spúšťaný na produkčnej distribúcii za účelom vytvárania hlásení reportovania a ich posielanie klientom. Z dôvodu dĺžky samotného skriptu, viac než 2000 riadkov, sa v práci skript nenachádza, je však priložený ako príloha. Skript je vytvorený s licenciou s otvoreným kódom typu GPL.

4.4.2 Vstupy skriptu

Zdrojom dát sú už spomínané logy z činnosti Squid proxy serveru vo funkcii transparentného proxy serveru. Skript pre svoju funkčnosť a pokročilé vytváranie hlásení o reportovaní potrebuje ešte iné či už konfiguračné súbory alebo zdroje iných dát. Okrem dát na analýzu skript potrebuje aj zdroje dát v ktorých sú informácie o klientoch ako napríklad ich meno, emailová adresa alebo iný identifikačný prvok, keďže produkčnú sieť možno rozdeliť na pevnú sieť identifikovaných klientov a variabilne sa meniacu skupiny klientov pripojených bezdrôtovo bez bližšej identifikácie. Pri vývoji skriptu boli definované nasledovné zdroje:

- **dhcpd.leases** – Súbor ktorý obsahuje informácie činnosti DHCP protokolu. Z tohto súboru je možné získať informácie pre časť siete v ktorej sa nachádzajú klienti bez identifikácie. Hlavnými zdrojmi informácií sú IP adresa, MAC adresa a host name stanice klienta.
- **database.txt** – Súbor vytvorený pre účely skriptu, obsahuje informácie o identifikovanej časti siete. Identifikačnými údajmi sú hlavne IP adresa, meno, email a informácia či sa má daný klient spracovať, nakoľko niektoré koncové prvky siete ako napríklad tlačiarne a servery nemusia byť zahrnuté do hlásení reportovania. V prípade dynamickej konfigurácie

je možné namiesto IP adresy zadať MAC adresu, skript následne MAC adresu vyhľadá v súbore dhcpd.leases a priradí jej identifikačné údaje.

- **paths.conf** – Vytvorený súbor ktorý obsahuje cesty k predchádzajúcim súborom a ďalším zložkám ako napríklad zložke, v ktorej sa nachádzajú logy Squid, zložka v ktorej sa majú vytvárať reporty a umiestnenie vlastného logu skriptu. Tento súbor musí byť v rovnakej zložke ako skript a otvára sa ako prvý po spustený skriptu pre získanie potrebných informácií.
- **access.log.n** – Množina súborov s vstupnými dátami v rozsahu access.log.0 až access.log.n ktoré reprezentujú n-1 dní ktoré sa uchováva log.

Medzi vstupy skriptu by sa dali zaradiť aj vstupné argumenty, ktoré svojou definíciou môžu značne meniť beh skriptu a jeho výstupy.

4.4.3 Činnosť skriptu

Vykonávaná analýza má na starosti pomerne jednoduché úkony. Ide najmä o spracovanie štyroch hlavných údajov z riadkového záznamu a to:

1. Čas spracovania požiadavky zaslanej klientom
2. Veľkosť prenesených dát pre požiadavku v bytoch
3. IP adresa klienta
4. URL ktorú klient požadoval

S týmito údajmi je možné vytvárať prehľadné tabuľky, ktoré sú vždy stiahnuté na inú jednotku a iný identifikačný prvok. Typický ukazovateľ pre jedného klienta je tabuľka, v ktorej sú zoradené jednotlivé unikátne domény prvého stupňa podľa počtu navštívení. Z danej tabuľky ihneď vidieť, ktoré stránky sú klientom najnavštevovanejšie. Podobnú tabuľku možno vytvoriť nahradením počtu navštívením celkovou prenesenou veľkosťou, z tejto tabuľky ihneď vidieť z ktorých stránok užívateľ uskutočňoval najväčšie prenosy. Možné je tiež vytvoriť globálnu tabuľku, v ktorej sú všetky IP adresy zoradené podľa počtu prenesených dát, počtu navštívených stránok, celkového počtu navštívených unikátnych stránok. Na základe tohto sumáru možno určiť, ktorý z klientov je najaktívnejší. Podobnú tabuľku možno určiť pre URL a zoradiť ich podľa počtu zobrazení alebo prenesených dát, čo dáva informáciu o najnavštevovanejších stránkach v rámci skupiny klientov. Následne je potrebné vygenerovaný report zaslať danému klientovi, túto funkcionalitu plní skript

pomocou emailu, pričom nevyužíva lokálny poštový server, ale pripája sa na vytvorené externé konto, odkiaľ posiela emaily spolu s hlásením reportovania ako prílohou. Pôvodné pokusy posielať tieto správy z poštového servera priamo na distribúcii pfSense sa ukázali ako neúčinné. Táto distribúcia je silne prispôsobená svojej funkcii a inštalácia prvkov, ktoré pre ňu nie sú prirodzené, sa ukázala ako problémová a pre danú distribúciu nebezpečná.

4.4.4 Spúšťanie skriptu a výstupy

Konkrétne používanie skriptu a jeho spúšťanie s argumentami možno rozdeliť do dvoch častí. Prvou časťou je definícia zdrojových dát, kedy sa určí konkrétny súbor na spracovanie, presný časový rozsah alebo počet dní od aktuálneho dátumu smerom do minulosti, ktoré sa majú analyzovať. Zdrojové dáta vždy predstavujú presnú množinu riadkov z rôznych acces.log súborov. Argumenty ktoré skriptu definujú časovú oblasť

- **-f access.log.N** – Argument „f“ musí byť nasledovaný názvom súboru, ktorý má byť spracovaný, tento súbor sa musí nachádzať v adresári, v ktorom sú log súbory Squid proxy servera.
- **-d N** – Argument „d“ musí byť nasledovaný kladným celým číslom, ktoré symbolizuje počet dní od aktuálneho dátumu, v rámci ktorých sa majú vyskytovať záznamy na spracovanie
- **-t YYYY-MM-DDTHH:MM:SS** – Argument „t“ musí byť nasledovaný jedným alebo dvomi časovými záznamami, v prípade dvoch záznamov sú tak definované dve hranice, v rámci ktorých sa majú vyskytovať všetky záznamy zdrojových dát. V prípade jedného záznamu sa druhou hranicou rozsahu stáva dnešný dátum. Pri zadávaní je časť YYYY-MM-DD povinná, časť s konkrétnym časom nie je povinná.

V prípade že zdrojové dáta neboli definované žiadnym spôsobom, dôjde k spracovaniu predvoleného súboru ktorým je access.log.1, teda posledný uzatvorený log, predpokladá sa že access.log.0 sa stále plní.

Druhou časťou, ktorá sa definuje pri spúšťaní, je konkrétna forma výstupu hlásenie o reportovaní. Skript dokáže generovať dva základné druhy hlásení. Prvým je HTML súbor, ktorý je uložený na pevný disk, druhým je výstup priamo v v príkazovom riadku po spustení skriptu. Druhý výstup je ideálny pre rýchlu kontrolu alebo ďalšie spracovávanie výstupu systémovými nástrojmi, zatiaľ čo prvý výstup je vhodnejší na posielanie prípadne inú prezentáciu. Najzákladnejším druhom hlásenia je HTML report pre jednu IP. Tento HTML súbor obsahuje časti:

1. **Identifikačná hlavička** – Krátka HTML tabuľka s identifikáciou, či už menom alebo MAC adresou a host name. Ďalej obsahuje sumár ako celkový počet prenesených dát, počet navštívených stránok, čas vytvorenia hlásenia, časový rozsah dát v hlásení a podobne.
2. **Tabuľka navštívení** – Tabuľka ktorá obsahuje unikátne domény prvého stupňa a počet ich navštívení daným klientom
3. **Tabuľka prenesených dát** - Tabuľka, ktorá obsahuje unikátne domény prvého stupňa a celkovú veľkosť prenesených dát pri požiadavkách na túto stránku.
4. **Tabuľka všetkých URL** – Tabuľka, v ktorej sú chronologicky zoradené všetky navštívené URL spolu s časom požiadavky

Druhým typom HTML hlásenia reportovania je globálny report zvlášť pre IP adresy a zvlášť pre domény. V prípade domén vytvorí súbor, ktorý obsahuje dve tabuľky:

1. **Tabuľka navštívení** – Tabuľka, ktorá obsahuje unikátne domény prvého stupňa a počet ich navštívení všetkými klientmi.
2. **Tabuľka prenesených dát** - Tabuľka, ktorá obsahuje unikátne domény prvého stupňa a celkovú veľkosť prenesených dát pri požiadavkách na túto stránku všetkými klientmi.

Globálny report pre IP adresy vytvára tri tabuľky:

1. **Tabuľka prenesených dát** - Tabuľka, ktorá obsahuje všetky IP adresy, ktoré sa nachádzajú v zdrojových dátach, tie sú zoradené podľa počtu dát, ktoré preniesli
2. **Tabuľka unikátnych domén** – Tabuľka, ktorá obsahuje všetky IP adresy, ktoré sa nachádzajú v zdrojových dátach, tie sú zoradené podľa počtu unikátnych domén prvého stupňa ktoré navštívili
3. **Tabuľka všetkých požiadaviek** – Tabuľka, ktorá obsahuje všetky IP adresy, ktoré sa nachádzajú v zdrojových dátach, tie sú zoradené podľa celkového počtu požiadaviek, ktoré v danom období mali.

Druhá skupina hlásení reportovania, ktorá vytvára výstup do príkazového riadku má dvoch členov

spoločných a to globálny report pre domény a globálny report pre IP adresy. Tretím hlásením je URL report, ktorý po zadaní konkrétnej URL zobrazí dostupné informácie o tom, ktoré IP navštívili danú URL, koľkokrát a koľko dát pri tom preniesli. Tento report je ideálny pre rýchle zistenie množiny klientov, ktorý pristupujú na konkrétnu stránku. Uvedené zobrazenia sa spúšťajú s argumentmi:

- **-i IP** – Argument „i“ musí byť nasledovaný IP adresou v správnom tvare, pre túto IP adresu bude následne vygenerovaný HTML report.
- **-w** – Argument „w“ vygeneruje globálny HTML report pre domény, alebo tiež web stránky.
- **-a** – Argument „a“ vygeneruje globálny HTML report pre IP adresy.
- **-wn** – Argument „w“ spolu s argumentom „n“ vygeneruje globálny report pre domény, alebo tiež web stránky ale iba ako výstup do príkazového riadku.
- **-an** – Argument „a“ spolu s argumentom „n“ vygeneruje globálny report IP adresy ale iba ako výstup do príkazového riadku.
- **-u URL** – Argument „u“ musí byť nasledovaný URL adresou, pre ktorú má byť vygenerovaný ako výstup do príkazového riadku.

Predvolený beh programu, teda bez argumentov má za následok vygenerovanie základných HTML hlásení pre všetky IP. Program možno použiť pre rovnaký druh hlásení aj pri akejkoľvek definícii zdrojových dát, pokiaľ nie sú definované iné argumenty manipulujúce s formátom hlásenia. Špeciálnym prepínačom je prepínač, ktorý spôsobí zaslanie výsledkov reportovania emailom klientom definovaných v súbore database.txt, tento prepínač je možné aplikovať len na zasielanie základných logov,. Teda buď pri základnom behu programu, alebo pri špecifikácii časovej oblasti, alebo pri špecifikácii konkrétnej IP. Pri zvolení globálnych hlásení sa použiť nedá.

- **-e** – Argument „e“ spôsobí zaslanie hlásení reportovania emailom po ich vygenerovaní.

Príklad príkazu ktorý spracuje zdrojové dáta za posledných 7 dní, vytvorí základné hlásenie reportovania pre jednotlivé IP a rozošle ich emailom.

```
proxyparser.pl -d 7 -e
```

Samostatnú kapitolu výstupov zo skriptu tvorí jeho vlastný log súbor. Tento log zaznamenáva svoju činnosť a plní ho informáciami o každom spracovaní dát, môže sa hodiť pri zisťovaní prípadných problémov s aplikáciou.

4.5 Presun na skutočnú sieť ústavu

Takáto implementácia skriptu má viacero krokov a náležitostí, ktoré treba dodržať, ešte pred samotným presunom z testovacieho prostredia je potrebné rozhodnúť o umiestnení zložky, v ktorej sa bude skript nachádzať. Taktiež treba určiť, do ktorej zložky sa budú ukladať vytvorené hlásenia reportovania a do ktorej zložky sa bude ukladať vlastný log aplikácie. Emailové konto, pomocou ktorého skript rozosiela hlásenia o reportovaní musí byť nakonfigurované na zvolenom serveri, musí byť vyjasnená metóda autentifikácie a skript musí mať k dispozícii prihlasovacie údaje do emailovej schránky. Samotný spôsob reportovania nemusí byť jasne určený, to kedy a čo bude skript odosielať zostáva na voľbe správy siete. Pre posielanie emailov je však nutná inštalácia troch knižníc v rámci distribúcie pfSense a produkčného prostredia ústavu. Prvým krokom pre inštaláciu určenie umiestnenia, odkiaľ má systém knižnice stiahnuť, v príkazovom riadku pfSense je to možné vykonať príkazom:

```
setenv PACKAGESITE
ftp://ftp-archive.freebsd.org/pub/FreeBSD-Archive/ports/i386/packages-8.1-release/All/
```

Ďalšou sériou troch príkazov s rozličnou knižnicou sa dané knižnice nainštalujú.

```
pkg_add -r p5-Net-SMTP-SSL-1.01.tbz
pkg_add -r p5-IO-Socket-SSL-1.33.tbz
pkg_add -r p5-Authen-SASL-2.14.01.tbz
```

Po ich úspešnej inštalácii je možné skript používať v produkčnom prostredí ústavu. Možnosti, ako ho používať sú voľbou správy siete na ústave. Pri tomto rozhodnutí je dôležitým aspektom množstvo zdrojových dát pre prácu skriptu. Pokiaľ sa skript spustí s priveľkým množstvom dát, jeho beh sa predĺži a výsledné súbory môžu naberať na veľkosti. Pokiaľ je objemový prietok záznamov počas prevádzky identifikovaný, spúšťanie vytvárania hlásení sa môže vhodne nastaviť. Najvhodnejším spôsobom spúšťania sa javí zavedenie príkazu do programu cron, ktorý ho bude následne spúšťať v presne určených časových oblastiach. Program cron sa v prostredí pfSense spúšťa príkazom „crontab -e“. Príkaz sa zapíše ako riadok, v ktorom sa pomocou úvodných polí definuje, v ktorom časovom období má byť príkaz spustený, ktorého definícia nasleduje po

časových poliach. Príklad príkazu v programe cron pre vytváranie hlásení každý deň pre všetkých klientov a zasielanie na email:

```
# Minute (0-59)   Hour(0-24)  Day(1-31)   Month(1-12) Weekday(0-6)      Command
    10    0    *    *    *    /usr/bin/perl /UMIESTNENIE/ proxyparser.pl -e
```

Uvedený príklad spustí program každý deň po polnoci, pričom spracuje predvolený súbor ktorým je súbor obsahujúci všetky záznamy z predchádzajúceho dňa. Skript následne vytvorené hlásenia rozošle definovaným klientom. Samotný príkaz sa skladá s dvoch častí. Prvá hovorí o interpreteri, ktorý ho má spustiť, v tomto prípade perl, druhá časť definuje program aj s umiestnením a argumentmi s ktorými má byť spustený. Pokiaľ je objem dát malý, skript možno spúšťať aj počas dlhšieho obdobia, napríklad jeden týždeň

```
# Minute (0-59)   Hour(0-24)  Day(1-31)   Month(1-12) Weekday(0-6)      Command
    10    0    *    *    1    /usr/bin/perl /UMIESTNENIE/proxyparser.pl -d 7 -e
```

Tento príklad aktivuje skript vždy v pondelok desať minút po polnoci a vypracuje správy za posledných sedem dní, pričom ich rozošle jednotlivým klientom pomocou emailov.

4.6 Overenie funkčnosti navrhnutého riešenia

Squid implementácia transparentného proxy servera beží na sieť ústavu už dlhší čas bez problémov, rovnako bez problémov sú vytvárané logy. Tie sú spracovávané balíčkom LightSquid ktorý graficky vykresľuje činnosť klientov z týchto logov, čím overuje správnu implementáciu, proxy servera a funkčné vytváranie logov o funkčnosti. Implementácia skriptu pre spracovanie Squid logov sa po prvotných problémoch a ich odstránení ukázala funkčná a pracuje tak, ako bolo vyžadované. S pohľadu funkčnosti sa úspešne podarilo implementovať transparentný proxy server a taktiež riešenie pre správu reportovania na ústave, ktorá bola viackrát úspešne odskúšané a overené.

5 Záver

Úlohou diplomovej práce bolo implementovať VPN server a proxy server na sieti ústavu, vytvoriť riešenie reportovania a tiež ho implementovať. Pri implementácii VPN servera boli rozobrané základné pojmy v sieťovom prenose a teória šifrovaného prenosu dát. Pri voľbe konkrétnej aplikácie boli rozobrané aspekty práce VPN serveru pri vyžadovanej konfigurácii. Z dostupných možností bola vybraná aplikácia OpenVPN server, ktorá bola nainštalovaná na vytvorenom testovacom prostredí. Testovacie prostredie predstavuje analogickú infraštruktúru ako skutočná sieť ústavu. Tvorili ho tri stanice, z čoho jedna predstavovala prístupový bod do siete pre dve testovacie siete, pričom v každej bola zapojená jedna stanica, ktorá predstavovala klienta v danej sieti. Vytvorenie testovacie prostredia zahŕňalo inštaláciu zvolených operačných systémov na všetky stanice, návrh a konfiguráciu sieťovej topológie. Aplikácia OpenVPN bola inštalovaná a konfigurovaná v serverovom prostredí distribúcie pfSense. VPN sa podarilo úspešne nainštalovať a nakonfigurovať oddelené prístupy pre zamestnancov a študentov.

Ďalšou úlohou bolo implementovať proxy server. V rámci jeho implementácie boli rozobrané základné definície proxy servera, jeho činnosti, najpoužívanejšie kombinácie jeho funkcií a využitie v praxi. Pri zvažovaní vhodnej aplikácie bol vypracovaný literárny prehľad existujúcich riešení na báze distribúcie pfSense. Vzhľadom na vyžadovanú funkcionalitu bol vybraný a nainštalovaný Squid proxy server. Inštalácia a konfigurácia sa uskutočnila v už vytvorenom testovacom prostredí na testovacom serveri s distribúciou pfSense.

Riešenie pre reportovanie bolo uskutočňované pomocou vlastného riešenia, ktorým bol skript v jazyku perl. Zdrojom dát pre vytváranie hlásení boli výstupy činnosti klientov, ktoré vytvára Squid proxy server v rámci svojej činnosti transparentného proxy servera. Vytvorené riešenie spracováva informácie o navštívených stránkach, počte prenesených dát a čase návštevy ako hlavné vstupy. Výstupom riešenia sú súbory obsahujúce sumárne informácie o činnosti klientov. Všetky vytvorené súčasti boli úspešne nainštalované a otestované v testovacom prostredí. Následne bola vytvorená infraštruktúra prenesená a integrovaná do produkčnej siete na ústave. V rámci produkčnej siete sa taktiež podarilo všetky súčasti úspešne nainštalovať, nakonfigurovať a otestovať.

6 Zoznam použitej literatúry

[1] *Cisco DocWiki – Internetworking Basics,* [on line], 25.5.12.2013. Dostupné z <http://docwiki.cisco.com/wiki/Internetworking_Basics>

[2] Alena Kabelová, Libor Dostálek : *Velký prúvodce protokolz TCP/IP a systémov DNS, 2. aktualizované vydanie*

[3] Martin Struhár : *Virtuálna privátna sieť ústavu, diplomová práca*

[4] Matthew D. Wilson : *VPN HOWTO,* [on line], 25.5.12.2013. Dostupné z <http://www.faqs.org/docs/Linux-HOWTO/VPN-HOWTO.html>

[5] SANS Institute : *OpenVPN and the SSL VPN Revolution,*[on line], 25.5.12.2013. Dostupné z <http://www.sans.org/reading_room/whitepapers/vpns/openvpn-ssl-vpn-revolution_1459>

[6] OpenManiak : *OpenVPN Introduction*, [on line], 25.5.12.2013. Dostupné z <http://openmaniak.com/openvpn.php>

[7] ORACLE : *Oracle Directory Server Enterprise Administration Guide 11g – Managing Certificates,* [on line], 25.5.12.2013. Dostupné z <http://docs.oracle.com/cd/E20295_01/html/821-1220/bcauo.html>

[8] Wikipedia: *Digital signature,* [on line], 25.5.12.2013. Dostupné z <http://en.wikipedia.org/wiki/Digital_signature>

[9] Wikipedia: *Public-key cryptography,* [on line], 25.5.12.2013. Dostupné z <http://en.wikipedia.org/wiki/Public-key_cryptography>

[10] OpenVPN Technologies, Inc. : *HOWTO,* [on line], 25.5.12.2013. Dostupné z <http://openvpn.net/index.php/open-source/documentation/howto.html>

[11] BSD Perimeter : *pfSense Documentation,* [on line], 25.5.12.2013. Dostupné z <http://doc.pfsense.org/index.php/Main_Page>

[12] Duane Wessels : *Squid The Definitive Guide*

[13] Matthew Strebe, Charles Perkins : *Firewally a proxy-servery Praktický prúvodce*

[14] Christian Hilgers : *HAVP Features,* [on line], 25.5.12.2013. Dostupné z <http://www.server-side.de/features.htm>

[15] HAProxy : *HAProxy description,* [on line], 25.5.12.2013. Dostupné z <http://haproxy.1wt.eu/>

[16] Wikipedia: *Proxy server,* [on line], 25.5.12.2013. Dostupné z <http://en.wikipedia.org/wiki/Proxy_server>

[17] Kulbir Saini : *Squid Proxy Server 3.1 Beginner's Guide*

7 Prílohy

[1] Kompaktný disk (CD) so skriptom na riešenie reportovania proxyparser.pl, disk ďalej obsahuje vzory súborov paths.conf a database.txt