UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Nejc Škoberne

NADGRADITEV POŽARNEGA ZIDU pfSense Z UPORABNIŠKIMI STORITVAMI

Diplomska naloga
na univerzitetnem študiju

Mentor: prof. dr. Borut Robič

Ljubljana, 2008

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Namesto te strani **vstavite** original izdane teme diplomskega dela s podpisom mentorja in dekana ter žigom fakultete, ki ga diplomant dvigne v študentskem referatu, preden odda izdelek v vezavo!

Zahvala

Zahvaljujem se mentorju prof. dr. Borutu Robiču, ki mi je pomagal pri nastanku tega diplomskega dela. Hvaležen sem tudi svoji ženi Mici in staršem, ki so me dejavno podpirali na celotni študijski poti, in nenazadnje svojemu še nerojenemu otroku, saj je bil pri pisanju moja največja motivacija. Hvala tudi Marku Toplaku in Mitarju Milutinoviću, ki sta prispevala veliko tehtnih pripomb. Za lektoriranje naloge se zahvaljujem Martini Eyer.

Kazalo

Povzetek

Požarni zid pfSense je poceni in zanesljiva rešitev na področju omrežnih operacijskih sistemov. Kljub temu pogrešamo bolj uporabniško naravnane funkcionalnosti, kot sta poštni in spletni strežnik. Največja težava, ki jo povzročijo dodatne uporabniške funkcionalnosti na pomembnih gradnikih omrežja (požarnih zidovih, usmerjevalnikih itn.), je zmanjšanje varnosti celotnega omrežja.

V diplomskem delu smo zato preučili možnosti za nadgradnjo sistema pfSense z dodatnimi uporabniškimi storitvami na način, ki ne bo ogrozil varnosti požarnega zidu in omrežij za njim. To smo storili z uporabo ječ FreeBSD (angl. *FreeBSD jail*), ki omogočajo virtualizacijo na stopnji operacijskega sistema, kar omeji posledice morebitnega vdora. Poleg tega smo načrtovani sistem primerjali z nekaterimi drugimi podobnimi sistemi.

Pripravili smo tudi načrt za izvedbo nadgradnje tako, da smo opisali posamezne ječe z uporabniškimi storitvami in nekatera razmerja med njimi. Preučili smo še možnosti za posodabljanje tako nadgrajenega sistema pfSense ter nekaj idej za razvoj uporabniškega vmesnika in za nadaljnje delo.

Ključne besede:

FreeBSD, pfSense, ječa, virtualizacija, storitev, uporabniške storitve, operacijski sistem, strežnik, varnost, poslovni model ...

Poglavje 1

Uvod

Načrtovalec ali vzdrževalec informacijske infrastrukture v organizaciji ima danes veliko možnosti pri izbiri posameznih gradnikov sistema – osebnih računalnikov, strežnikov in drugih omrežnih naprav, ki skupaj sestavljajo informacijsko omrežje poslovnega sistema. Značilnost večjih poslovnih sistemov je (ali je vsaj pred nedavnim porastom uporabe virtualiziranih sistemov bila), da je večina gradnikov fizično neodvisna od drugih. Za posamezne funkcije v računalniškem omrežju se uporabljajo različne naprave, kot so usmerjevalniki, stikala, podatkovni strežniki, poštni strežniki, požarni zidovi itd. V srednje velikih in majhnih podjetjih po drugi strani zaradi sorazmerno manjših potreb in navadno tudi zaradi manjših investicij v informacijsko infrastrukturo prevladuje načelo vse-v-enem (angl. *all-in-one*), pri čemer se skuša s čim manj strojne opreme omogočiti čim več različnih funkcionalnosti. Tipičen primer je, denimo, Microsoftov izdelek Windows Small Business Server, ki v enem samem operacijskem sistemu združuje vrsto strežniških aplikacij – od poštnega strežnika do požarnega zidu.

Posebno za manjša podjetja (pa tudi za večja, vendar zaradi drugih lastnosti, ki se jim bomo posvetili kasneje) so predvsem zanimivi sorodni izdelki, ki temeljijo na odprtokodnih tehnologijah, saj imajo pogosto privlačno ceno ali pa so (navadno samo programska oprema) celo zastonj. Seveda pa tudi v takih primerih t. i. skupni stroški lastništva (TCO) niso enaki nič. Nasprotno, pri nesmotrnih in nepremišljenih uvedbah takšnih rešitev so lahko ti stroški celo večji od tistih, ki bi jih ustvarile na prvi pogled dražje rešitve.

Odprtokodni izdelki so gotovo eden zanimivejših fenomenov današnje informacijske družbe. Poleg manjših stroškov, ki jih prinašajo podjetjem, so pomembni tudi z vidika odprtosti. Predvsem večje združbe, oz. njihovi razvojno-informacijski oddelki, lahko takšne rešitve v veliki meri ali celo popol-

noma prilagodijo lastnim potrebam.

Kljub razmahu odprtokodnih rešitev na področju informacijske infrastrukture je mogoče zaznati pomanjkanje rešitev vse-v-enem, ki bi bile varne in preproste za uvedbo (namestitev), nastavitev ter vzdrževanje. Takšne rešitve morajo biti tudi zanesljive in, če podjetje nima sklenjene vzdrževalne pogodbe s podjetjem, ki zastopa to rešitev, dobro podprte s strani uporabniških skupnosti (angl. *user groups*). To pomeni, da morajo te rešitve imeti dovolj uporabnikov. Le tako lahko hišni (angl. *in-house*) vzdrževalec rešuje morebitne težave, ki se pojavljajo pri uporabi rešitve.

V nadaljevanju bomo preučili možnost nadgradnje že obstoječega izdelka – požarnega zidu pfSense, ki temelji na odprtokodnem operacijskem sistemu FreeBSD. Izdelek trenutno ponuja funkcionalnosti omrežne naprave (angl. *appliance*): zmogljiv požarni zid, strežnik VPN, visoko razpoložljivost, porazdeljevanje obremenitve, strežnik DHCP itd. Raziskali bomo možnosti (predvsem varne) uvedbe dodatnih funkcionalnosti (poštni strežnik, datotečni strežnik, spletni strežnik itd.), ki niso strogo vezane na osnovne omrežne funkcije.

1.1 Omrežni operacijski sistemi

V diplomskem delu se bomo ukvarjali z nekaterimi omrežnimi operacijskimi sistemi (NOS). To je "programska oprema, ki nadzoruje omrežje in sporočilni promet, ki se po njem prenaša, ter čakalne vrste, dostop več uporabnikov do omrežnih virov (datoteke, tiskalniki), in nudi določene administrativne funkcije, vključno z varnostjo" [1].

Nekatere osnovne značilnosti omrežnih operacijskih sistemov so naslednje:

- **varnost:** avtentikacija, avtorizacija, prijavne omejitve, omejitev dostopov;
- osnovna podpora omrežnim povezavam (Ethernet);
- osnovne lastnosti operacijskih sistemov – podpora računalniškim arhitekturam, različnim protokolom, prepoznavi strojne opreme in večprocesnim programom;
- podpora imeniškim in imenskim storitvam (ADS, DNS ...);
- podpora datotečnim in spletnim storitvam ter storitvam za arhiviranje.

Vidik varnosti je pri omrežnih operacijskih sistemih zelo pomemben, saj ima lahko njihova izvedba nanjo velik vpliv. Z začetkom prenosa pomembnih informacij preko računalniških omrežij je postalo zelo pomembno tudi njihovo varovanje: tako dostop do samih informacij kot njihova neokrnjenost. Avtentikacija je preverjanje istovetnosti osebe ali naprave in ne zadošča za zagotavljanje varnosti (če bi vsi uporabniki imeli vsa pooblastila na nekem sistemu, ta sistem ne bi bil varen, čeprav se uporabniki lahko avtenticirajo). Potrebna je še avtorizacija, ki zagotavlja nadzor nad dostopom posameznih avtenticiranih uporabnikov do informacij, operacij itn. Če sta ena ali druga okrnjeni, je okrnjena varnost celotnega sistema.

Kadar govorimo o varnosti, imamo v mislih predvsem varnost računalniškega omrežja in informacij, ki se po njem prenašajo. Varnost računalniškega omrežja sestavljajo ukrepi (oz. njihova (ne)učinkovitost), izvedeni na omrežni infrastrukturi, in varnostna politika, ki jo je sprejel vzdrževalec omrežja, da bi zaščitil omrežje pred nepooblaščenimi dostopi.

Operacijski sistem, ki ima orodja za delo z omrežji (npr. Microsoft Windows XP), ni nujno tudi omrežni operacijski sistem. Mednje štejemo predvsem tiste sisteme, ki so bili napisani z namenom pripomoči k optimalnemu delovanju omrežij.

Nekateri primeri omrežnih operacijskih sistemov so: GNU/Linux, Novell NetWare, Microsoft Windows NT, Digital OpenVMS, pa tudi vse različice sistemov BSD, kot so OpenBSD, FreeBSD, NetBSD. V tej diplomski nalogi nas bo zanimal predvsem omrežni operacijski sistem FreeBSD.

1.2 FreeBSD in pfSense

Družina današnjih sistemov BSD (FreeBSD, NetBSD, OpenBSD, PC-BSD ...) ima skupna prednika: operacijska sistema 4.4BSD in 386BSD. V času t. i. vojn UNIX, ko so se odvijali veliki tektonski premiki na področju operacijskih sistemov UNIX, sta se iz teh dveh sistemov ločeno razvila operacijska sistema FreeBSD in NetBSD. OpenBSD, DragonFly BSD, PC-BSD in v določeni meri tudi Mac OS X so bili izpeljani pozneje, nekateri iz sistema FreeBSD, drugi iz NetBSD.

Danes najbolj znani operacijski sistemi BSD se razlikujejo predvsem v poudarkih, ki jim razvijalci sledijo pri razvoju:

- **FreeBSD** – poudarek na zmogljivosti in zanesljivosti;
- **NetBSD** – poudarek na prenosljivosti med različnimi procesorskimi arhitekturami;

- **OpenBSD** – poudarek na varnosti;
- **DragonFly BSD** – poudarek na večprocesorskih sistemih in gručah;
- **PC-BSD**, **DesktopBSD** – poudarek na preprosti uporabi;
- **Mac OS X** – komercialen izdelek, ki ga podjetje Apple prodaja skupaj s svojo strojno opremo.

Prednost sistema FreeBSD pred drugimi sistemi BSD je predvsem v tem, da ima veliko uporabnikov (ki jih sicer še vedno precej manj od uporabnikov GNU/Linux), kar pomeni dobro podprtost raznovrstne strojne opreme ter enostavno in hitro pomoč v težavah.

Ker bomo v petem poglavju navedli nekaj izdelkov, ki temeljijo na nekaterih distribucijah operacijskega sistema GNU/Linux, naredimo na tem mestu kratko primerjavo med sistemom FreeBSD in sistemi, ki temeljijo na jedru Linux. V primerjavi s sistemi, ki temeljijo na GNU/Linux, je FreeBSD namenjen predvsem strežniškim sistemom. Lehey v [4] naredi preprosto primerjavo med sistemi FreeBSD in sistemi, ki temeljijo na jedru Linux – v tabeli 1.1 je naštetih le nekaj bistvenih razlik.

Operacijski sistem pfSense je odprtokodni požarni zid, ki nudi vse pomembne značilnosti dragih komercialnih požarnih zidov (vključno z enostavnostjo uporabe), le da je na voljo zastonj. Sistem pfSense temelji na prilagojeni različici FreeBSD, kar je veljalo že za njegovega predhodnika – požarni zid m0n0wall. Za razliko od sistema m0n0wall pfSense namenja več pozornosti delovanju na sistemih PC in strežnikih, in ne toliko na vgrajenih (angl. *embedded*) napravah, kar velja za m0n0wall. Poleg tega je avtor in vzdrževalec izdelka m0n0wall en sam, pfSense pa je rezultat skupnega dela ekipe programerjev.

1.3 Cilji diplomske naloge

Enega izmed vidikov filozofije razvoja sistema pfSense opiše Mott v [2] takole: "Sistem pfSense je požarni zid in cilj požarnega zidu je zagotavljanje varnosti. Čim več funkcionalnosti dodamo, tem bolj povečamo možnost, da bo varnostna luknja v tej funkcionalnosti ogrozila varnost požarnega zidu. Ustanovitelji in glavni programerji požarnega zidu pfSense so mnenja, da nobena izmed storitev, ki deluje zunaj 2. do 4. plasti (modela OSI, op. p.), ne sodi v osnovni sistem pfSense. Storitve lahko dodajamo v obliki paketov, vendar mora biti vzdrževalec sistema previden pri odločitvah za njihovo uporabo. Večinoma bi

samostojen računalnik, ki bi nudil dodatne storitve, deloval bolje (v sodelovanju s požarnim zidom pfSense) v smislu zagotavljanja največje varnosti."

FreeBSD	**Linux**
Poleg jedra, ki ga razvija število razvijalcev (več kot 300 programerjev ima pravico neposredno spreminjati uradno izvorno kodo (angl. *commit bit*)), vključuje široko paleto uporabniških programov – obstaja samo ena "distribucija" sistema FreeBSD.	Linux je jedro, ki ga vzdržujejo Linus Torvalds osebno (vse spremembe jedra mora potrditi lastnoročno) in nekaj njegovih sodelavcev. Programi, ki niso del jedra, se od distribucije do distribucije lahko zelo razlikujejo.
Je neposredno izpeljan iz izvornega sistema UNIX, čeprav ne vsebuje več nobenih ostankov kode podjetja AT&T.	Je klon sistema UNIX (napisan popolnoma na novo) in ni nikoli vseboval kode podjetja AT&T.
Je manj znan od Linuxa, saj je bilo njegovo širjenje v začetku oteženo zaradi pravnih sporov s podjetjem AT&T.	Nikoli ni imel težav s pravnimi spori, zato so zanj nekaj časa mislili, da je edini prosti in odprti UNIX-u podoben operacijski sistem na voljo.
Zaradi nepoznanosti je zanj manj lastniške programske opreme in tudi gonilnikov za strojno opremo.	Za GNU/Linux je danes na voljo veliko programske opreme. Prav tako je vsaka nova strojna oprema, ki pride na trg, praktično takoj podprta tudi v Linuxu.
Na voljo je pod licenco BSD, ki določa zelo malo omejitev glede ravnanja z izvorno kodo (mogoče jo je uporabiti tudi za zaprtokodne izdelke).	Na voljo je pod licenco GNU GPL, ki je v primerjavi z licenco BSD veliko strožja glede uporabe izvorne kode (ni je mogoče uporabiti za zaprtokodne izdelke, ki bi jih želeli razpečevati).

Tabela 1.1: Primerjava operacijskih sistemov FreeBSD in Linux

Cilj diplomske naloge je zasnovati načrt nadgraditve sistema pfSense prav s funkcionalnostmi v obliki dodatnih monolitnih paketov, o katerih piše Mott. Paketi morajo biti enostavno namestljivi in nastavljivi z uporabo spletnega vmesnika pfSense. Poleg tega, da morajo biti dodatne funkcionalnosti na voljo kot samostojne nadgradnje in ne smejo biti del osnovnega sistema pfSense (nekatere nadgradnje v takšni obliki pfSense že vsebuje), bomo v diplomski nalogi dodali še dodatno stopnjo varnosti. Vse dodatne funkcionalnosti se bodo izvajale v t. i. ječah FreeBSD, ki so z vidika uporabnika popolnoma ločene od

gostiteljskega sistema. To pomeni, da gre v resnici za virtualizacijo sistema FreeBSD v sistemu FreeBSD, kar omogoča varnost osnovnih funkcionalnosti požarnega zidu tudi ob vdoru v katero izmed ječ.

V drugem poglavju bomo na kratko preučili izdelek pfSense; podlago, na kateri je zgrajen, njegov poslovni model, strukturo ter filozofijo razvoja in možnosti za razširitve, na katere se bomo osredinili v nadaljevanju naloge. V tretjem poglavju bomo preučili tehnologijo virtualizacije na stopnji operacijskega sistema – ječe FreeBSD, ki so temeljni del ideje tega diplomskega dela. V četrtem poglavju bomo podrobneje navedli razloge za uporabo ječ FreeBSD v sistemu pfSense, nato pa ga primerjali z nekaterimi podobnimi že obstoječimi izdelki: SME Server, ClarkConnect in eBox. Nato se bomo lotili načrtovanja vgradnje ječ v sistem pfSense in opisali strukturo posameznih ječ z uporabniškimi storitvami. Na koncu pa bomo analizirali še možnosti za posodabljanje tako nadgrajenega sistema pfSense ter navedli nekaj idej za razvoj uporabniškega vmesnika in nadaljnje delo.

Poglavje 2

Študij sistema pfSense

Da bi lahko nadgradili zdajšnji sistem pfSense, ga je potrebno dobro razumeti. Ker knjig in člankov na tem področju še ni, je potrebno uporabiti druge vire: spletne strani wiki[1] za razvijalce sistema pfSense ([3]) in sámo izvorno kodo, do katere je mogoče dostopati preko javne shrambe CVS (angl. *repository*).

2.1 Operacijski sistem FreeBSD kot podlaga

Izdelek m0n0wall, ki ga razvija Manuel Kasper, sta Chris Buechler in Scott Ullrich septembra 2004 vzela kot osnovo za novi izdelek pfSense. Pri tem sta ohranila veliko lastnosti m0n0walla:

- sistem FreeBSD kot osnova za gradnjo;
- spletni vmesnik za upravljanje, preprost za uporabo;
- enostavna namestitev;
- uporaba jezika PHP za zagonske skripte FreeBSD (namesto lupinskih skript FreeBSD);
- hranjenje vseh sistemskih nastavitev v eni sami datoteki XML.

Razlogov za izbor sistema FreeBSD za osnovo gradnje novega, uporabniku prijaznejšega sistema je več (povzeto po [5]):

[1] Poseben tip zbirke nadbesedilnih spisov ali skupinskega programja, s katerim je izdelana.

- **podpora različnim arhitekturam:** FreeBSD lahko teče ne samo na najbolj razširjeni arhitekturi Intel® i386™, ampak tudi na arhitekturah alpha, amd64, ia64, pc98, Sparc64® in po novem tudi na ARM®, MIPS® in PowerPC®;

- **razširljiva ogrodja (*Extensible Frameworks*)**:

  - **Netgraph:** modularen omrežni podsistem, ki lahko dopolnjuje obstoječo omrežno infrastrukturo v jedru. Razvijalcem omogoča izpeljavo lastnih omrežnih modulov, kar omogoča hiter in enostaven razvoj naprednih omrežnih storitev. Module za nekatere storitve (PPPoE, ATM, ISDN, Bluetooth, HDLC, EtherChannel, Frame Relay, L2TP ...) vključuje že uradno jedro FreeBSD.

  - **GEOM:** modularno vhodno/izhodno diskovno ogrodje za pretvorbo zahtev (angl. *I/O request transformation framework*) omogoča enostavno, čisto in hitro vgradnjo novih storitev za hrambo podatkov v jedro FreeBSD. Primeri uporabe: razvoj različnih tehnologij RAID, kriptografskih zaščit podatkov itd.

  - **GBDE:** nudi močno kriptografsko zaščito datotečnih sistemov, izmenjalnih (angl. *swap*) in drugih naprav za hrambo podatkov. Poleg tega lahko GBDE šifrira celotne datotečne sisteme, ne samo posameznih datotek. Nešifrirani podatki se nikoli ne "dotaknejo" plošč trdih diskov.

  - **PAM:** tako kot Linux tudi FreeBSD nudi podporo modelu PAM. Ta omogoča upravljalcu nadgradnjo tradicionalnega avtentikacijskega modela UNIX® "uporabniško ime/geslo". FreeBSD nudi module, ki omogočajo več različnih vrst avtentikacijskih mehanizmov, vključno s Kerberos 5, OPIE, RADIUS in TACACS+.

Poleg tega velja omeniti, da FreeBSD slovi tudi po svoji varnosti, zanesljivosti in robustnosti. Med operacijskimi sistemi, ki znajo natačno sporočati čas od zadnjega ponovnega zagona (angl. *uptime*), je FreeBSD najpogosteje na seznamu 50 najzanesljivejših spletnih strežnikov, ki ga vzdržuje organizacija Netcraft[2].

[2]Netcraft je angleško podjetje, ki nudi internetne storitve in izvaja raziskave o tržnih deležih na področju spletnih strežnikov in operacijskih sistemov.

2.2 Odprtokodna zasnova kot poslovni model

Z vzponom interneta se je predvsem v svetu programske opreme začel svojevrsten fenomen, ki ima vidnejše začetke v osemdesetih letih. Takrat sta univerza v Berkeleyu v ZDA in Richard M. Stallman začela izdajati programsko opremo pod drugačnimi licencami, kot je bilo dotlej v navadi. Prosta programska oprema (angl. *free software*) je bila sprva predvsem domena akademskih raziskovalcev in programerskih navdušencev. Šele v drugi polovici devetdesetih let so začela nastajati podjetja, ki so v prosti programski opremi (termin odprta koda je namreč nastal šele leta 1998) videla poslovne priložnosti. Primer so podjetja, ki so prodajala zgoščenke z distribucijami GNU/Linux, saj jih takrat praktično še ni bilo mogoče distribuirati po internetu. Kupci so tem podjetjem začeli pošiljati vprašanja glede uporabe programske opreme, ta pa so spet zaznala novo tržno nišo – nudenje pomoči pri namestitvi, tehnično podporo in svetovanje. Kmalu so svojo dejavnost razširila z razvojnimi oddelki, ki so pripravljali do uporabnika prijaznejše vmesnike za uporabo programske opreme. Podobno so se razvila nekatera podjetja, ki še danes uspešno poslujejo na ravno teh področjih (tipičen primer je podjetje Red Hat, Inc.).

Golden v [6] zastavlja vprašanje, kako je mogoče služiti s prosto programsko opremo. Podoben primer bi bil prodajanje vode, ki je dandanes prav donosen posel, čeprav teče (pitna) voda iz vsake pipe. Na vprašanje avtor odgovori s predstavitvijo treh osnovnih poslovnih modelov, ki so se do sedaj uveljavili v praksi:

- **Model britve:** podjetje izda en (navadno bistveni) del izdelka kot odprtokodno programsko opremo (rezilo), drugi del pa ohrani pod komercialno licenco. Mogoče je tudi, da je celoten izdelek na voljo pod odprtokodno licenco in je plačljiva le strokovna podpora. Tako precej zmotivira uporabniško skupnost, da aktivno sodeluje pri razvoju odprtega dela izdelka. To je navadno izjemno uspešno, saj podjetja praktično nič ne stane, hkrati pa v kratkem času pridobi veliko uporabnikov, kar bi pri izključno komercialnem izdelku navadno trajalo dlje. Model britve uporablja večina odprtokodnih projektov, med drugim tudi eBox in ClarkConnect, ki ju v nadaljevanju primerjamo z načrtovanim sistemom.

- **Model "naprave" (angl. *appliance*):** odprta programska oprema sama po sebi ni cilj, ampak sredstvo za doseg cilja. Primer takega modela so denimo poceni brezžični usmerjevalniki, ki jih je mogoče kupiti kot "črne škatle", ki jih (skoraj) samo priklopimo in delujejo. Sámo jedro Linux je v tem primeru praktično neuporabno, zato ga podjetja uporabijo

kot osnovo za operacijske sisteme v takšnih in podobnih napravah. Tako zmanjšajo skupne stroške proizvodnje izdelka. Kot primer navajamo podjetje ASUS, katerega usmerjevalniki za domačo uporabo temeljijo na operacijskem sistemu Linux.

- **Hibridni model:** (nekateri ga imenujejo tudimodel dualnega licenciranja) isti izdelek se izda pod dvema (ali več) različnimi licencami. Komercialne licence omogočajo manj omejeno uporabo kupljenega izdelka, odprte licence pa navadno zahtevajo, da tudi nadaljnji (iz njega izpeljani) izdelki ostanejo odprti, kar za nekatera podjetja morda ni primerno. Primer podjetja, ki uporablja hibridni model, je MySQL AB, ki izdeluje MySQL.

Poleg tega Golden našteje naslednje prednosti odprtokodne usmeritve:

- hitro pridobivanje uporabnikov zaradi večje motivacije in možnosti sodelovanja;
- cenejši razvoj zaradi vključevanja že napisane kode;
- hitrejši razvoj in morebiten prodor na trg (angl. *time to market*) zaradi vključevanja že napisane kode.

Tim O'Reilly, ustanovitelj podjetja O'Reilly Media, Inc., največje založbe z računalniško literaturo, je na konferenci EclipseCon 2005 1. marca 2005 poudaril naslednje smernice za razvoj proste oz. odprtokodne programske opreme:

- **Načrtovanje za integracijo:**
  - programska oprema naj bo kot komponenta enostavno vgradljiva v druge, morda večje sisteme;
  - izbor primerne licence, ki omogoča zgornjo smernico;
  - modularnost;
  - dokumentiranje vseh vmesnikov API.

- **Uporabniško naravnan razvoj:**
  - hitre (zgodne) in pogoste izdaje;
  - učinkovit mehanizem za pošiljanje poročil o napakah in popravkov uporabnikov;
  - povišanje najdejavnejših uporabnikov v vloge z večjo odgovornostjo.
- **Osredinjenje na:** hitrost testiranja, razvoja in integracije, in ne na dodajanje zaprtih in plačljivih dodatkov ter funkcionalnosti (tako imajo uporabniki vedno najboljše na trgu);
- **Sledenje standardom v industriji:** ponuditi izdelek v nekaj dokazano dobrih konfiguracijah, ki ustrezajo industrijskim standardom, in prepustiti uporabnikom možnost izbire. Ko se pojavijo nova področja uporabe, razviti nove konfiguracije, ki jih podpirajo.
- **"Večni beta":** nove funkcionalnosti naj ne bodo na voljo le v enotnih (monolitnih) izdajah, ampak naj bodo uporabnikom predstavljane redno, kot del vsakdanje uporabniške izkušnje. Vključiti uporabnike kot vir sprotne povratne informacije – kot primer lahko navedemo Google Mail (beta).

Izvorna koda izdelka pfSense je izdana pod licenco BSD, kar pomeni, da jo lahko kdorkoli uporabi, spreminja in vgrajuje v svoje potencialno komercialne izdelke. Avtorji bi zato lahko izbrali tudi drugačen način licenciranja (česar v primeru licence GPL osnovnega sistema ne bi mogli storiti zaradi njene že omenjene "virusne" narave), vendar ostajajo pri odprtosti.

Podjetje BSD Perimeter, LLC, ki stoji za izdelkom pfSense, se ukvarja z nudenjem podpore, svetovanjem in razvojem dodatnih funkcionalnosti za izdelka pfSense in m0n0wall. Gre torej za poslovni model britve, saj je izdelek prosto dostopen, za dopolnilne storitve pa je potrebno plačati.

Bistvene prednosti izbora takega modela za izdelek pfSense so naslednje:

- veliko uporabnikov – velika skupnost, ki na poštnih seznamih in spletnih forumih nudi veliko brezplačne pomoči (uporabniki si med seboj pomagajo);
- zaradi veliko uporabnikov ("veliko oči") je na voljo sorazmerno veliko dobrih povratnih informacij (poročila o hroščih, (ne)želene funkcionalnosti ...);

- izkušeni in tehnično podkovani uporabniki lahko sami popravljajo izdelek in popravke pošiljajo neposredno razvojni ekipi - tako so tudi uporabniki razvijalci;

- izkušeni programerji lahko razvijajo dodatne funkcionalnosti za lastne potrebe, ki jih pogosto podarijo tudi uporabniški skupnosti (in razvijalcem, ki lahko to funkcionalnost vgradijo v uradni izdelek);

- odprt pogovor med vsemi razvijalci o pomembnih odločitvah – katere funkcionalnosti vključiti v izdelek in katerih ne, način implementacije, ohranjanje prožnosti, možnosti vzdrževanja ipd.;

- težave se rešujejo demokratično in odprto – udeležen je lahko vsak, ne glede na vlogo in izkušnje – ob hujšem nesoglasju lahko kadarkoli vsak odide in celo nadaljuje razvoj v okviru novega izdelka.

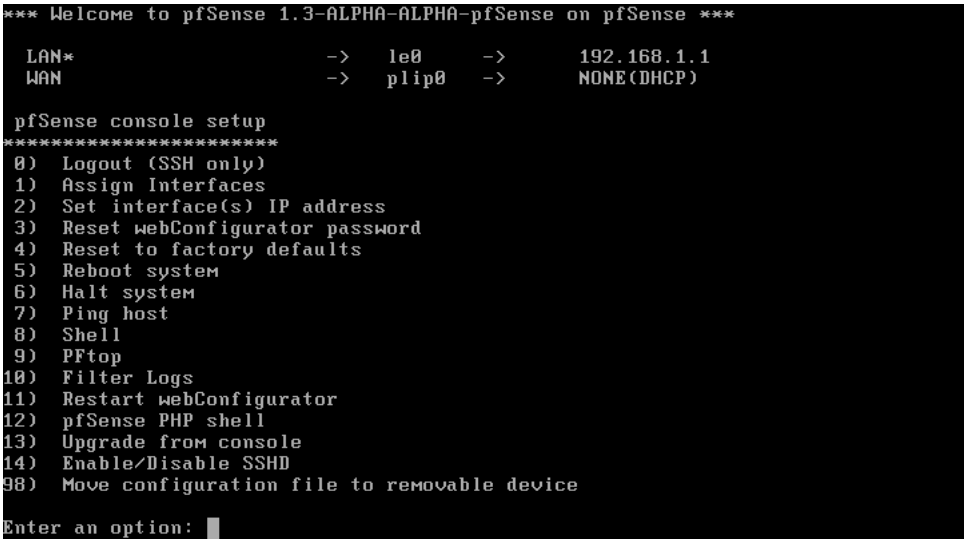
Takšen model pa ima tudi nekatere slabe lastnosti:

- pomoč uporabnikom, ki nimajo sredstev za nakup tehnične podpore, ni zanesljiva in je zato lahko zavajajoča (uporabnik se tega ne zaveda dovolj, dokler ne naleti na hude, morda usodne težave);

- razvijalci niso "prisiljeni" v razvoj in se lahko polenijo, kar povzroči zastanek ali celo prenehanje razvoja. Kljub takemu razpletu dogodkov bi se verjetno vedno našla nova skupina ljudi (sploh če izdelek kvaliteten), ki bi prevzela razvoj istega ali novega izdelka, ki bi temeljil na zapuščenem.

2.3 Struktura sistema in filozofija razvoja

Vzpostavitev osnovnega in delujočega sistema pfSense je zahvaljujoč enostavnemu namestitvenemu postopku zelo pregledna, kar se ujema z osnovno filozofijo razvoja izdelka – enostavnost za uporabo ter varne in delujoče privzete nastavitve. Takšna miselnost ima sicer lahko tudi slabe strani – predvsem za varnost omrežja, ki ga bo takšen požarni zid varoval. Če pravila požarnega zidu načrtuje nekdo brez potrebnih strokovnih izkušenj, lahko zaradi enostavnosti namestitve vzpostavi delujoč, a neustrezno zavarovan sistem. Kljub temu je treba poudariti, da še tako kompleksen namestitveni postopek ne more zagotoviti varnega sistema, zato je bolje staviti na enostavnost in usposobljeno osebje.



Slika 2.1: Pregled osnovnih sistemskih nastavitev nastavljivih z uporabo konzole.

Po uspešni namestitvi sistema na trdi disk[3] moramo z uporabo konzole (tekstovni vmesnik) nastaviti osnovne parametre, ki jih pfSense potrebuje za dokončen zagon. Pomembna je predvsem nastavitev omrežnih naprav, saj dotlej ni mogoče uporabljati spletnega vmesnika za upravljanje. Po nastavitvi omrežnih naprav lahko z uporabo konzole nastavljamo tudi druge osnovne sistemske nastavitve, katerih pregled je na sliki 2.1.

2.3.1 Centralizirane nastavitve

Ena večjih slabosti sistemov UNIX (in njim podobnih) je razpršenost nastavitev. To je posledica osnovne filozofije teh sistemov, ki jih sestavlja množica manjših, navadno samostojno vzdrževanih programov. Skoraj vsak program ima svojo nastavitveno datoteko in tudi svoj način zapisa (sintakso) te datoteke. Primer: ko želimo spremeniti nastavitve strežnika DNS, ki je del osnovnega sistema FreeBSD, moramo spremeniti samo njegovo nastavitveno datoteko, ki je sintaktično različna od vseh drugih. Zato mora vzdrževalec dobro poznati vse te programe, če želi vzpostaviti celovit sistem. Eden izmed poskusov poenotenih nastavitev v sistemih FreeBSD (in drugih sistemih BSD) je uvedba datoteke `/etc/rc.conf`, ki vsebuje nekatere centralizirane nastavitve

[3]Namesto trdega diska lahko uporabimo tudi kakšno drugo bločno napravo, katere vsebina se po ponovnem zagonu računalnika ne izgubi.

samega sistema, pa tudi nekaterih programov. Kljub temu je to navadna besedilna datoteka, ki ne vsebuje (večine) nastavitev v nastavitvenih datotekah programov.

Razvijalci izdelka pfSense so zato sklenili vse nastavitve, tako sistemske kot tudi nastavitve posameznih programov, popolnoma centralizirati. Tako je del sistema pfSense datoteka `/cf/conf/config.xml`, ki vsebuje vse nastavitve v obliki XML. Seveda morajo programi (denimo prej omenjeni strežnik DNS) nekako prebrati nastavitve, ki so zapisane v tej datoteki. Ker pa bi bilo spreminjanje vseh programov nemogoče, je bilo potrebno razviti sistem pomožnih programov in skript, ki po vsaki spremembi centralne nastavitvene datoteke iz nje ustvarijo nastavitvene datoteke za programe. Kljub temu hranjenje nastavitev v datoteki XML ne zagotavlja preglednosti vmesnika, zato mu je treba nameniti pri razvoju posebno pozornost.

Pomembna prednost centraliziranih nastavitev je tudi enostavno arhiviranje in ponovna vzpostavitev stanja sistema pfSense. Ob strojni okvari je mogoče okvarjeni požarni zid pfSense zelo hitro zamenjati z rezervnim in restavrirati nastavitve iz arhiva.

2.3.2 PHP kot sistemski skriptni jezik

Druga bistvena značilnost sistema pfSense in njegovega predhodnika m0n0wall je uporaba programskega jezika PHP za razvoj sistemskih skript in ne samo za spletni vmesnik. Ker je PHP pravi višjenivojski objektno orientiran jezik, je mogoče na ta način enostavno brati nastavitve iz centralizirane datoteke XML.

2.4 Možnosti razširitve

Na operacijski sistem FreeBSD je mogoče namestiti veliko dodatne programske opreme in z njo ponuditi široko paleto storitev: spletni strežnik, poštni strežnik, datotečni strežnik itd. Na pfSense lahko trenutno z uporabo upravljalca paketov, ki so ga razvili razvijalci pfSense, že namestimo dodatne pakete. Na različico pfSense 1.3-ALPHA je tako mogoče namestiti 24 dodatnih storitev, ki pa so bolj ali manj enake omrežnim storitvam, ki jih ponujajo podobni izdelki – TinyDNS, OpenBGP, razna pomožna orodja za nadzor in diagnostiko ipd.

Ker požarni zid pfSense temelji na sistemu FreeBSD, ga je mogoče nadgraditi z vsakršno programsko opremo, ki teče na tem sistemu. Ker pa vsaka dodatna storitev pomeni tudi varnostno tveganje, je potrebno zagotoviti, da ranljivost dodatnih storitev ne bo povzročila ranljivosti celotnega požarnega

zidu, saj lahko to za omrežje za njim pomeni katastrofo. Potencialni napadalec lahko namreč vdor na požarni zid prikriva in tako potuhnjeno prestreza prihajajoči in odhajajoči promet omrežja.

Poglavje 3

Virtualizacija v FreeBSD – ječe

Bill Joy je leta 1982 za potrebe testiranja med razvojem operacijskega sistema 4.2BSD napisal sistemsko funkcijo `chroot(2)`, ki deluje tako, da spremeni vidni korenski imenik trenutnega procesa in njegovih potomcev. Program, ki se mu tako spremeni korenski imenik, ne more dostopati do datotek zunaj tega korenskega imenika, ki se navadno imenuje "ječa chroot".

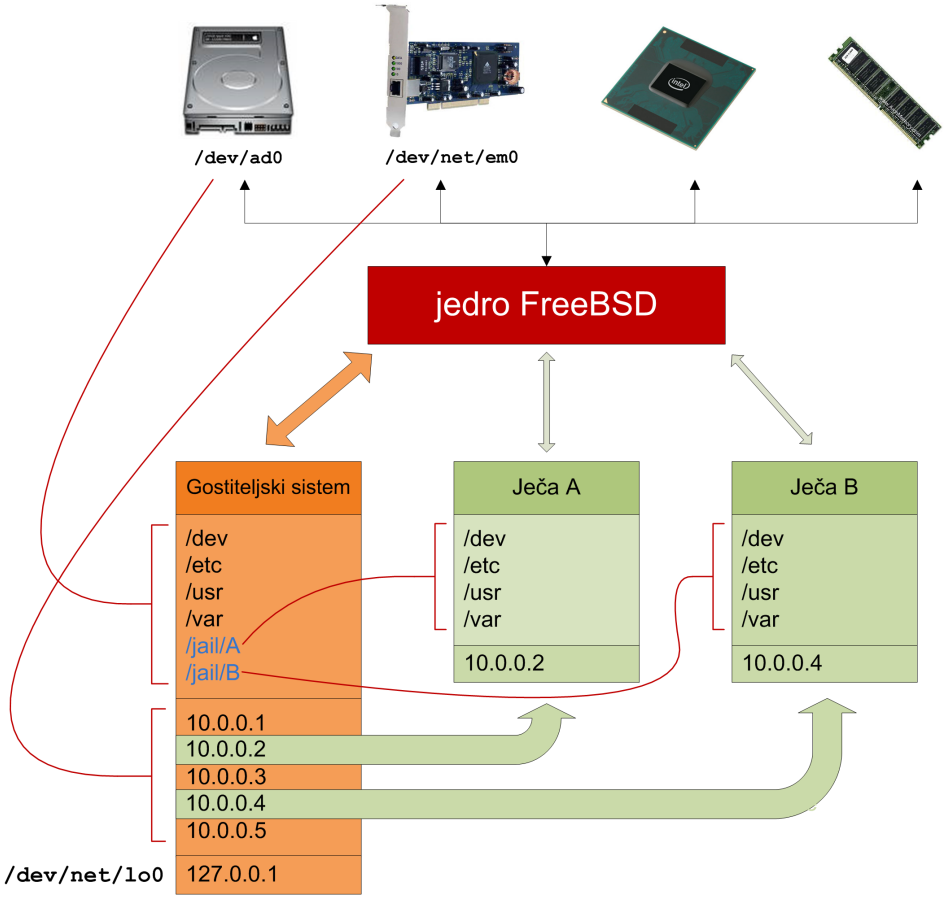
Znane pomanjkljivosti sistemskega klica `chroot(2)` so bile največja motivacija za razvijalce FreeBSD, da bi naredili zanesljivejši sistem z razširjenimi funkcionalnostmi. Poul-Henning Kamp in Robert Watson sta na več mestih spremenila jedro FreeBSD tako, da so vsi načini pobegov iz okolja `chroot(2)` odtlej nemogoči [7]. Čeprav se jedro komajda zaveda, da nek proces teče v okolju `jail(2)` (ječi), izvaja vrsto mehanizmov v različnih podsistemih, ki skrbijo, da procesi ostanejo v njem.

Cilji, ki bi jih z ječami FreeBSD radi dosegli, so:

- **Virtualizacija:** vsaka ječa je navidezno okolje, ki teče na gostiteljskem sistemu in ima svoje lastne datoteke, procese, uporabnike in superuporabnika. Z vidika procesov v ječah je okolje (skoraj) enako nevirtualiziranemu.

- **Varnost:** vsaka ječa je popolnoma ločena od drugih, kar pomeni dodatno stopnjo varnosti.

- **Enostavnost pooblaščanja:** zaradi omejenega območja delovanja ječe lahko vzdrževalci brez težav pooblastijo uporabnike za izvajanje opravil, ki zahtevajo pravice superuporabnika – če gre karkoli narobe, napaka prizadene samo eno ječo, preostali sistem ostane nedotaknjen.



Slika 3.1: Shematski prikaz sistema FreeBSD z dvema ječama.

Na sliki 3.1 je shematski prikaz sistema FreeBSD z dvema ječama. Takšen sistem bi lahko uporabili za potrebe krajevnega spletnega strežnika. V ječi A in B bi lahko tekel spletni strežnik Apache; prvi bi bil dostopen na naslovu IP 10.0.0.2, drugi pa na naslovu 10.0.0.4.

3.1 Osnovne lastnosti

Korff in sodelavci ([7]) ter prej omenjena avtorja ([8]) so opisali osnovne lastnosti ječ FreeBSD in njihove bistvene razlike v primerjavi z uveljavljenim, a zastarelim sistemom `chroot(2)`.

3.1.1 Omejen medsebojen vpliv procesov

Za posamezne ječe FreeBSD je značilen njihov `JID`. V navadnem okolju `chroot(2)` lahko procesi, ki imajo isti `UID`, vplivajo eden na drugega, predvsem pa imajo procesi s številko `UID` superuporabnika 0 popoln nadzor nad vsemi ostalimi procesi. To pomeni, da jim lahko pošiljajo katere koli signale in tako povzročijo tudi zaključitev ali "smrt" drugih procesov. Procesi, ki tečejo v ječah FreeBSD, pa imajo poleg številk `UID` in `GID` dodeljeno tudi številko ječe `JID`. Procesi s številkami `JID`, različnimi od 0, tako ne morejo nikakor vplivati na procese (niti ne morejo vedeti, da obstajajo), ki imajo drugačen `JID` od njihovega. Procesi z `JID`, enakim 0, lahko vidijo vse procese, če pa je tudi `UID` 0, lahko tudi ti pošiljajo katere koli signale vsem procesom.

3.1.2 Omejen dostop do omrežnih virov

Sistemski klic `chroot(2)` na noben način ne omejuje procesov pri dostopanju do omrežnih virov. Procesi lahko neomejeno ustvarjajo tudi surove vtičnice (angl. *raw sockets*). To pomeni, da lahko poljubno spreminjajo zaglavja paketov in tako pošiljajo (ponarejene) pakete z lažnimi naslovi IP ali izvajajo napade DoS.

Sistemski klic `jail(2)` procese omeji na več načinov:

- vsaka ječa (skupina procesov z istim `JID`) ima natanko en naslov IP in nanj vezano ime (angl. *host name*);
- procesi v ječi lahko pošiljajo samo pakete, ki imajo izvorni naslov IP enak naslovu IP svoje ječe;
- ponarejanje naslovov MAC v ječah ni mogoče;
- dostop do surovih vtičnic privzeto ni mogoč (na voljo je le uporaba protokolov TCP in UDP, je pa ta možnost nastavljiva).

Naj poudarimo, da dostop do protokolov TCP in UDP v praksi pomeni dostop do vseh storitev krajevnega omrežja, saj večina uporabniških protokolov uporablja ali TCP ali UDP. Zato je zelo pomembno, da je dostop ječe do zaupnega omrežja omejen s pravili požarnega zidu na gostitelju (pfSense uporablja požarni zid pf).

3.1.3 Omejen dostop do naprav

Poleg naštetih omejitev imajo procesi, ki tečejo v ječah FreeBSD, omejen tudi dostop do naprav v imeniku `/dev`. S prihodom datotečnega sistema *devfs*, ki je vpet v imenik `/dev`, ročno ustvarjanje posameznih datotek naprav s sistemskim klicem `mknod(2)` ni več potrebno. Proces v ječi FreeBSD lahko bere s katere koli naprave, ki jo vidi v (svojem) imeniku `/dev`. Zato je potrebno poskrbeti, da ječam FreeBSD v njihovih imenikih `/dev` zagotovimo samo nujno potrebne naprave. FreeBSD za to uporablja nastavitveno datoteko `/etc/devfs.conf`.

3.2 Primerjava z navideznimi stroji

Poleg ječ FreeBSD obstajajo tudi drugi podobni sistemi za virtualizacijo na stopnji operacijskega sistema: Linux-VServer, OpenVZ, FreeVPS (za Linux), Container/Zone (za Solaris), sysjail (za OpenBSD in NetBSD) itd. V splošnem lahko posameznim tako virtualiziranim sistemom rečemo tudi navidezne particije (angl. *virtual partition*), saj gostiteljski sistem na neki način particionira svoje vire med samega sebe in vse navidezne sisteme. Ta način virtualizacije se od izdelkov za virtualizacijo z navideznimi stroji (angl. *virtual machine*), kot sta VMWare in XeN, razlikuje predvsem v naslednjih lastnostih:

- **Učinkovitost:** virtualizacija na stopnji operacijskega sistema je učinkovitejša. To pomeni, da je zanjo značilna manjša skupna poraba (angl. *overhead*) virov. Programi v navideznih particijah namreč uporabljajo kar isti vmesnik za sistemske klice kot gostiteljski sistem, kar pomeni, da klicev ni treba programsko posnemati (angl. *emulation*).

- **Prožnost:** virtualizacija z navideznimi stroji je prožnejša, saj posnema strojno opremo. To pomeni, da lahko v navideznih strojih poganjamo kateri koli operacijski sistem na arhitekturi, ki jo navidezni stroj virtualizira. Pri virtualizaciji na stopnji operacijskega sistema to ni mogoče, zato lahko v navideznih particijah tečejo le sistemi, ki uporabljajo isto jedro kot gostiteljski sistem.

- **Hranjenje podatkov:** nekateri sistemi za virtualizacijo na stopnji operacijskega sistema nudijo mehanizme kopiraj-ob-pisanju (angl. *copy-on-write*) na stopnji datotek. To pomeni, da se kopije datotek, ki so na voljo več navideznim particijam, ustvarijo šele takrat, ko jih navidezne particije prvič spremenijo. Tak način omogoča enostavno ustvarjanje varnostnih kopij in je prostorsko učinkovitejši od podobnih mehanizmov

na stopnji podatkovnih blokov na disku, ki se jih poslužujejo sistemi za virtualizacijo z navideznimi stroji.

Poglavje 4

Uporaba ječ FreeBSD v sistemu pfSense

V tem poglavju bomo raziskane sisteme in tehnologije združili v delujočo celoto. Varnost in neodvisnost ječ FreeBSD bomo izkoristili pri nadgradnji požarnega zidu pfSense z uporabniškimi storitvami.

4.1 Motivacija

Vzdrževalci sistemov navadno vsak zase in vedno znova razvijamo poenotene, za vzdrževanje enostavne sisteme, ki jih uporabljajo za svoje potrebe in potrebe različnih organizacij. Kar je po eni strani lepota odprtokodnih sistemov, po drugi strani lahko zavira razvoj. Pomanjkanje komunikacije med vzdrževalci, ki bi omogočala razvoj skupnega sistema, za vse ali vsaj večino, ni redek pojav. Z odpravo te težave bi se zmanjšalo podvajanje opravljenega dela. Poleg tega bi se povečala tudi zanesljivost in varnost skupnega sistema, saj bi vsi razvijalci in vzdrževalci (torej uporabniki) hkrati programsko opremo tudi testirali in poročali o napakah. Seveda pa nimajo vsi uporabniki enakih potreb, zato lahko vsak, ki pogreša kakšno funkcionalnost, to prispeva. Zelo verjetno bo prišla prav tudi komu drugemu.

Za izhodišče takega sistema se zdi pfSense zelo primeren – je stabilen, uveljavljen omrežni operacijski sistem, ki ga je zaradi njegove proste narave (licence) mogoče poljubno dopolnjevati.

4.2 Pregled podobnih sistemov

Kljub temu da že obstaja nekaj izdelkov z nekaterimi lastnostmi, ki si jih vzdrževalci sistemov želimo, z nobenim izmed teh ne moremo biti povsem zadovoljni. Bistvena razlika med spodaj navedenimi sistemi in sistemom, ki bi temeljil na izdelku pfSense in ječah FreeBSD, je prav virtualizacija uporabniških storitev na stopnji operacijskega sistema in s tem neprimerno povečana varnost. Ne glede na to je dobro, da glavne "tekmece" (SME Server, ClarkConnect in eBox) na kratko preučimo.

Primerjava vseh treh izdelkov glede na njihove funkcionalnosti, ki nas zanimajo, je prikazana v tabeli 4.1.

4.2.1 SME Server

Izdelek SME Server temelji na sistemu GNU/Linux in uporablja paketni sistem uveljavljene strežniške distribucije Linux CentOS, kar zagotavlja dolgoročno dostopnost varnostnih posodobitev. SME Server je bil razvit s poudarkom na uporabniških storitvah in na preprostosti, zato ne vsebuje večine naprednejših funkcionalnosti drugih podobnih sistemov (predvsem tistih, ki se tičejo požarnega zidu). Izdelek je tako namenjen predvsem manj zahtevnim uporabnikom, ki bi ga radi hitro začeli uporabljati.

Kljub zanimivemu naboru funkcionalnosti se zdi, da SME Server počasi zamira. Večina sicer obsežne dokumentacije se nanaša na zastarelo programsko opremo, predvsem na starejše operacijske sisteme Microsoft Windows. Vmesnik je relativno preprost in ne uporablja naprednih tehnologij (JavaScript, AJAX ...). Marsikatera storitev ni več posebno aktualna (strežnik FTP, statične spletne strani ipd.), zato je lahko razumeti potrebe po novejših in modernejših izdelkih.

4.2.2 ClarkConnect

ClarkConnect temelji na distribuciji RedHat Enterprise Linux. Podjetje Point Clark Networks, ki operacijski sistem razvija in vzdržuje, je temeljno distribucijo nadgradilo z lastnimi moduli. Jedro sistema ClarkConnect je Linux 2.6. Bistvena razlika med njim in ostalimi opisanimi sistemi, je v poslovnem modelu – podjetje se je odločilo za model britve, zato izdelek ponuja v dveh izdajah: *Community Edition* in *Enterprise Edition*. Prva je namenjena predvsem domačim uporabnikom, ki ne potrebujejo tehnične podpore in nekaterih dodatnih funkcionalnosti. Odsotnost teh funkcionalnosti pa lahko za

	SME Server	ClarkConnect	eBox
E-poštni strežnik	Strežniki POP3, IMAP, SMTP vključno s podporo varnim povezavam TLS/SSL in načinu preverjanja istovetnosti uporabnika SMTP-AUTH. Možen je tudi dostop do e-pošte preko spletnega vmesnika ter uporaba (skupnih) imenikov naslovov, prav tako pa je mogoče nastaviti pregledovanje elektronske pošte s protivirusnim programom, zaznavanje in izločevanje nezaželene pošte.	Strežniki POP3, IMAP, SMTP vključno s spletnim vmesnikom za elektronsko pošto, dodatek za boljše povezovanje z odjemalcem Microsoft Outlook, več rešitev za zaznavanje nezaželene pošte (SpamAssassin, Dspam, greylisting) in protivirusna programska oprema.	Strežniki POP3, IMAP, SMTP vključno z orodjem za filtriranje nezaželene pošte SpamAssassin in protivirusnim programom ClamAV.
Datotečni strežnik	Omogoča deljenje datotek med računalniki v omrežju, ki uporabljajo različne operacijske sisteme in podpirajo Microsoftovo skupino protokolov CIFS. Poleg tega omogoča prijavljanje računalnikov Microsoft Windows v t. i. domeno Windows NT (angl. *Windows NT domain*) in nastavitev kvot (angl. *quotes*) porabe diskovnega prostora za posamezne uporabnike.	Skupina programov projekta Samba omogoča izmenjavo datotek z uporabo Microsoftove skupine protokolov CIFS.	Skupina programov projekta Samba omogoča deljenje datotek z uporabo Microsoftove skupine protokolov CIFS. Podatke o uporabnikih in skupinah uporabnikov upravlja imeniški sistem OpenLDAP.
Spletni strežnik	Osnovne možnosti spletnega strežnika brez možnosti povezave s podatkovno bazo in uporabe dinamičnih spletnih strani.	Omogoča poganjanje dinamičnih spletnih strani, razvitih za okolje LAMP. Vgrajena je tudi spletna fotogalerija.	Omogoča javno objavo datotek, do katerih lahko dostopamo tudi preko Microsoftovih omrežnih protokolov.
Tiskalniški strežnik	Možnost skupne uporabe tiskalnika med računalniki v omrežju.	Možnost deljenja tiskalnika med računalniki v omrežju z uporabo programske opreme CUPS.	Možnost deljenja tiskalnika med računalniki v omrežju z uporabo programske opreme CUPS.
Podpora poljem RAID	Možnost uporabe diskov v obliki diskovnih polj RAID-1, RAID-5 in RAID-6.	Možnost uporabe diskov v obliki diskovnih polj RAID-0, RAID-1 in RAID-5.	
Uporabniške shrambe	T. i. shrambe i-bay omogočajo deljenje istih datotek z uporabo protokolov FTP, HTTP ali Microsoftovih omrežnih protokolov.	T. i. Flexshare omogoča deljenje istih datotek z uporabo protokolov FTP, HTTP(S), e-poštnih (SMTP, MIME, S/MIME) in Microsoftovih omrežnih protokolov.	
Nadgradljivost	Uporabiti je mogoče katerekoli druge programske pakete, ki so na voljo za distribucijo CentOS, vendar jih je treba večinoma nastavljati in upravljati z njimi »ročno« (z uporabo lupine).	Prosta (angl. *community*) različica zagotavlja brezplačno dostopnost nadgradenj 1-2 leti po namestitvi, plačljiva pa 6 let.	
Požarni zid		Omogoča nadzor nad prometom na treh slojih OSI (omrežni, transportni in aplikacijski), zmogljivost Multi-WAN, ki omogoča več (redundantnih) povezav v zunanja omrežja, in nadzor nad uporabo pasovne širine na omrežnem sloju.	Omogoča določitev abstraktnih objektov, ki jih uporabimo v pravilih, filtriranje, preusmeritve, podporo povezavam VLAN in Multi-WAN (za porazdeljevanje obremenitve) ter nadzor nad porabo pasovne širine na omrežnem sloju.
Povezljivost VPN		Podpora protokolom OpenVPN, IPSec in PPTP VPN za varno povezovanje med posameznimi omrežji ali med omrežji in odjemalci – t. i. cestnimi bojevniki (angl. *roadwarriors*).	Je omogočena z uporabo protokola OpenVPN.
Spletno predpomnjenje in filtriranje		Mogoča je uporaba spletnih posrednikov (angl. *web proxy*) za dostop do zunanjih spletnih strani, ki poleg tega, da zmanjšajo porabo pasovne širine internetne povezave, nudijo tudi možnost filtriranja določenih spletnih vsebin na aplikacijskem sloju.	
Orodja za sodelovanje (angl. *groupware*)		Omogočajo uporabnikom enostavnejšo komunikacijo z uporabo skupnih koledarjev, stikov, opravil, zapiskov, e--poštnih seznamov itd.	
Podatkovna baza		Relacijska podatkovna baza MySQL.	
Arhiviranje		Mogoče je arhiviranje nastavitev sistema ClarkConnect, ne pa tudi samih uporabniških podatkov.	
Šifriran datotečni sistem		Mogoče je šifrirati posamezne datotečne sisteme in se tako zavarovati pred morebitno fizično krajo podatkov.	
Strežnik za hitro sporočanje			Strežnik Jabber omogoča hitro sporočanje (angl. *instant messaging*) z uporabo ustreznih odjemalcev, ki je lahko omejeno na krajevno omrežje.

Tabela 4.1: Primerjava sistemov SME Server, ClarkConnect in eBox. Opisane so funkcionalnosti, ki jih bo nudil nadgrajeni izdelek pfSense, in tudi nekatere, ki jim v tem delu ne bomo posvečali posebne pozornosti.

zahtevnejšega uporabnika pomeni znatno omejitev: več kot deset poštnih predalov večje, programske posodobitve, dodatek za združljivost z odjemalci Microsoft Outlook™, dinamični VPN (povezave med napravami brez statičnih naslovov IP), nekatere funkcionalnosti požarnega zidu, kot so 1-na-1 NAT, DMZ in Multi-WAN. Poleg tega vse elektronske pošte, ki gre skozi sistem, ni mogoče hraniti v skupnem arhivu.

ClarkConnect je tudi v svoji okrnjeni različici neprimerno zmogljivejši od popolnoma prostega izdelka SME Server. Očitna je tudi večja tehnološka dovršenost izdelka in komercialna naravnanost (profesionalno oblikovanje vmesnika, na nekaterih mestih pomanjkljiva dokumentacija, velika hitrost delovanja). Ker temelji izdelek na drugačni distribuciji Linux, je mogoče tudi prilagajanje sistema po svoji meri, vendar na lastno pest v okviru pomoči uporabniške skupnosti.

4.2.3 eBox

Pretežno mlad projekt eBox temelji na distribuciji Debian GNU/Linux. To pomeni, da gre za moderen sistem, temelječ na najnovejših tehnologijah. Špansko podjetje Warp Networks S. L., ki izdelek razvija, se je odločilo za poslovni model britve, ki pa je nekoliko drugačen od poslovnega modela sistema ClarkConnect. Izdelek je v celoti na voljo zastonj, za strokovno in zanesljivo tehnično pomoč pa je potrebno plačati.

Sistem eBox pa kljub svoji tehnološki naprednosti ne deluje kot stabilen in robusten sistem, temveč kot razvojna različica za predogled. Hitrost delovanja je tudi na razmeroma zmogljivi strojni opremi neprimerno manjša od prej omenjenih sistemov. Intuitivnost vmesnika je negotova, saj v nekaterih primerih uporabnik ne ve točno, kaj se od njega pričakuje. Je torej obetaven projekt, ki pa v tem trenutku še ni primeren za resnejša produkcijska okolja.

4.3 Vgradnja ječ v pfSense

Glavna smernica pri uvedbi ječ FreeBSD v izdelek pfSense je ta, da ga bo mogoče še naprej namestiti in uporabljati kot doslej, torej brez dodatnih uporabniških storitev. Te bodo namreč na voljo kot poljubne nadgradnje, ki jih bo vzdrževalec lahko namestil in z njimi upravljal neposredno preko grafičnega spletnega vmesnika.

Preden začnemo vgradnjo ječ FreeBSD v pfSense, je treba sprejeti nekaj pomembnih odločitev, ki jih pozneje ne bo več mogoče enostavno spreminjati:

- **Lokacija:** določiti je potrebno imenik za hranjenje ječ in ga umestiti v obstoječo strukturo imenikov sistema pfSense.
- **Izbira pomožnega datotečnega sistema:** vsaka ječa potrebuje za delovanje osnovni sistem FreeBSD (bazo). Ker bi bilo nesmiselno imeti toliko kopij istih datotek, kot je ječ, je potrebno razmisliti, kako bomo z eno samo bazo (fizično kopijo datotek) oskrbeli vse ječe. Poleg tega nam bo taka zasnova omogočala lažje nadgrajevanje ječ.
- **Imeniška struktura:** določiti je potrebno imeniško strukturo, kamor bodo umeščene posamezne komponente – baza in ječe.

V naslednjih poglavjih bomo preučili možnosti za oblikovanje nadgrajenega sistema in utemeljili posamezne odločitve.

4.3.1 Ločena osnovna namestitev FreeBSD – baza

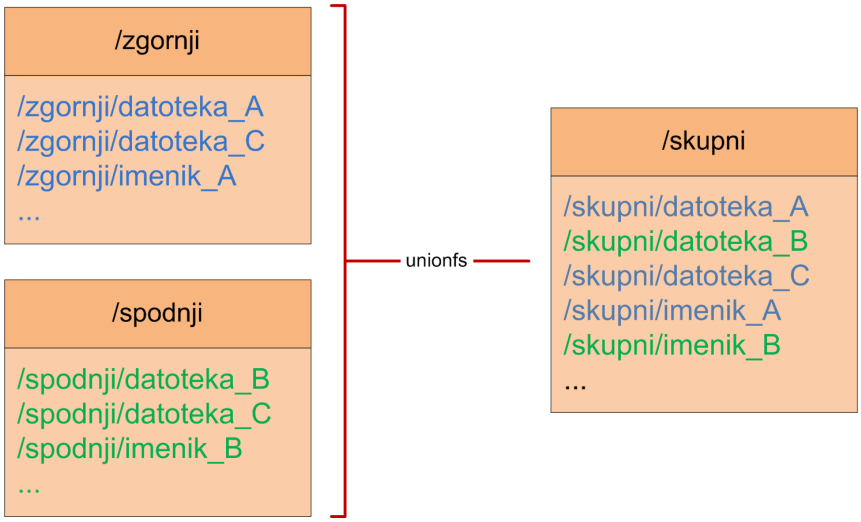
Ker je glavni (in tudi največji) diskovni razdelek (angl. *partition*), kjer je nameščen pfSense, navadno vpet v korenski imenik `/`, lahko hranimo vse ječe v imeniku `/jail`, ne da bi nas skrbelo pomanjkanje diskovnega prostora. Če se vzdrževalec odloči za ječe uporabiti ločen razdelek (morda na drugem fizičnem podatkovnem nosilcu), lahko to naredi v postopku namestitve sistema pfSense.

Baza je od osnovnega (gostiteljskega) sistema ločena predvsem zaradi glavne smernice, opisane v začetku poglavja. Želimo torej, da sta obstoječi sistem pfSense in sistem uporabniških storitev popolnoma ločena.

Razvijalci sistema pfSense so skrbno načrtovali izbor datotek iz operacijskega sistema FreeBSD. Le tako je mogoče pfSense namestiti tudi na vgrajene naprave, saj te navadno ne omogočajo hranjenja večje količine podatkov. Namestitvena slika sistema pfSense je tako velika nekaj čez 50 MB, kar je malo, v primerjavi z namestitvenimi slikami operacijskega sistema FreeBSD, katerih skupna velikost je skoraj 2 GB. Baza za ječe FreeBSD, ki bo v imeniku `/jail/base`, bo vsebovala tudi nekatere knjižnice in programe, ki jih osnovni sistem pfSense ne potrebuje.

4.3.2 Datotečna sistema *unionfs* in *nullfs*

Ker datoteke, ki sestavljajo bazo za ječe FreeBSD, zavzamejo sorazmerno veliko diskovnega prostora, je smiselno poiskati rešitev za eno samo fizično kopijo datotek, ki bi bila na voljo vsem ječam. Poleg tega je postopek nadgradnje vseh ječ hkrati tako enostavnejši, saj je potrebno nadgraditi samo bazo. Pri



Slika 4.1: Shematski prikaz dveh imenikov, združenih v datotečni sistem unionfs. V imeniku `/skupni` je na voljo datoteka iz zgornjega sloja (imenik `/zgornji`), saj ob istoimenskih datotekah ta vedno prevlada nad spodnjim.

tem gre za programsko opremo, ki je privzeto vključena v operacijski sistem FreeBSD (angl. *base*), in ne za dodatne programe iz zbirke programov (angl. *ports*), združljivih s sistemom FreeBSD. Zato je nadgrajevanje posameznih ječ s storitvami vseeno precej bolj zapleteno.

Unionfs je datotečni sistem, ki vsebuje dva sloja – zgornjega in spodnjega. Z njim lahko združimo dva imenika tako, da so v priklopni točki (angl. *mount point*) vidni vsi imeniki in datoteke, ki so ali v prvem (spodnjem) ali v drugem (zgornjem) imeniku (slika 4.1). Bistvena lastnost datotečnega sistema unionfs je, da se spremembe nad datotekami in imeniki v priklopni točki hranijo samo na zgornjem sloju. To ima vsaj dve posledici: poraba diskovnega prostora je minimalna, saj se za vsak priklop datotečnega sistema unionfs hranijo le spremenjeni imeniki in datoteke; enostavno arhiviranje – redno je treba arhivirati le zgornji sloj, saj ostane spodnji navadno nespremenjen in ga je potrebno arhivirati le enkrat.

Kljub temu pa ima unionfs (predvsem njegova izvedba v sistemu FreeBSD) nekaj težav. Na prvi pogled je morda celoten mehanizem videti preprost, vendar ko se poglobimo v delovanje in možnosti uporabe, ugotovimo, da je problem razmeroma kompleksen. Težave nastanejo pri brisanju datotek – tako z zgornjega kot s spodnjega sloja. Datotečni sistem UFS, ki je privzeti datotečni sistem v operacijskem sistemu FreeBSD, vsebuje podporo za poseben

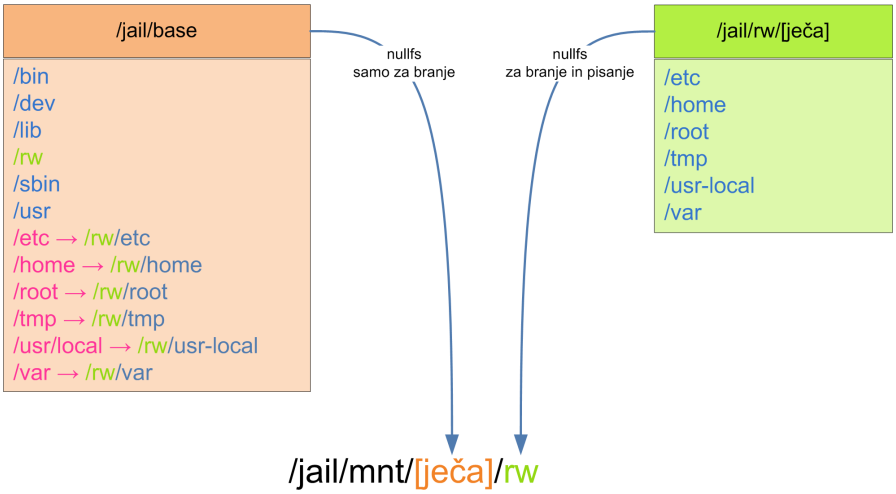
tip datotek – t. i. korektor (angl. *whiteout*). Če datoteko izbrišemo z zgornjega imenika, bo ta v datotečnem sistemu označena kot datoteka tipa korektor (analogija s korekturno tekočino), kar je preprosta oznaka za izbrisano datoteko. Če želimo, da se datoteka iz spodnjega imenika spet prikaže na točki priklopa, moramo to korekturno datoteko izbrisati (z ukazom `rm -W`). Če izbrišemo datoteko iz spodnjega imenika, bo ta pričakovano izginila tudi iz zgornjega. Kaj pa se zgodi, če datoteka iz spodnjega imenika izgine po tem, ko smo jo na točki prilopa že spremenili? V resnici nič posebnega, vendar to lahko pomeni težave pri načrtovanju nadgradenj programske opreme. Poleg tega je potrebno odgovoriti na vprašanje, ali naj se korekturne datoteke ustvarijo vedno ("za vsak slučaj") ali samo če obstaja istoimenska datoteka v spodnjem imeniku. Ta možnost je pri izvedbi datotečnega sistema unionfs v sistemu FreeBSD nastavljiva s posebnim parametrom pri uporabi programa `mount_unionfs`, s katerim datotečni sistem ustvarimo in ga tudi priklopimo na ustrezen imenik.

Ne nazadnje je v našem primeru kritična tudi sama kakovost izvedbe datotečnega sistema unionfs. Ta je v operacijskem sistemu FreeBSD še v razvojni fazi, saj se pojavljajo določene težave, ki jih uporabnik ne pričakuje (nevidnost datotek v spodnjih imenikih v nekaterih posebnih primerih, težave s premikanjem imenikov itd.). Do teh ugotovitev smo prišli s temeljitim testiranjem datotečnega sistema pri poskusni namestitvi osnovne ječe FreeBSD.

Nullfs je po drugi strani datotečni sistem, ki ne počne skoraj ničesar. Z orodjem `mount_nullfs` lahko namreč neki imenik priklopimo na drugega in s tem za vsako datoteko in imenik prvega imenika določimo alternativno pot dostopa – preko drugega imenika. Nullfs pa pridobi uporabno vrednost v našem primeru šele takrat, ko ga uporabimo v načinu samo-za-branje (angl. *read-only*). To pomeni, da lahko večino imenikov iz baze, katerih vsebina se v posameznih ječah ne bo spreminjala, priklopimo v ječe z uporabo datotečnega sistema nullfs z možnostjo samo-za-branje. Poleg tega moramo v vsako ječo skopirati tudi tiste imenike, katerih vsebina se bo morda spremenila. To pomeni, da bo uporaba sistema nullfs nekoliko potratnejša, kar se tiče diskovnega prostora, vendar zato precej preprostejša in lažja za vzdrževanje. To pa je dovolj dober razlog, da za svoje potrebe namesto datotečnega sistema unionfs izberemo datotečni sistem nullfs.

4.3.3 Struktura imenikov

Da bi določili imeniško strukturo v imeniku `/jail`, je bilo potrebno preučiti delovanje sistema FreeBSD. Morali smo izvedeti, v katere imenike sistem med



Slika 4.2: Shematski prikaz strukture podimenikov v imeniku `/jail`. Z modro barvo so označeni fizični imeniki, z roza pa mehke povezave.

navadnim delovanjem ne zapisuje novih oz. ne spreminja obstoječih datotek. Ti imeniki sestavljajo bazo, preostali pa so v imeniku `/jail/rw/[ječa]`, kjer je `[ječa]` ime ječe.

Imenik `/jail` vsebuje naslednje podimenike (navedene so absolutne poti):

- `/jail/base`: vsebuje nekoliko spremenjeno bazo sistema FreeBSD brez naslednjih imenikov (navedene so relativne poti):
  - **/etc:** vsebuje nastavitvene datoteke sistema in sistemskih programov, podatke o uporabnikih in skupinah itd.;
  - **/home:** vsebuje domače imenike nepriviligiranih uporabnikov;
  - **/root:** domači imenik priviligiranega (root) uporabnika;
  - **/tmp:** vsebuje morebitne začasne datoteke;
  - **/usr/local:** vsebuje uporabniške programe in njihove nastavitve ter navadno tudi večino uporabniških podatkov (elektronska pošta, datoteke datotečnega strežnika ...);
  - **/var:** vsebuje različne datoteke, ki se pogosto spreminjajo: dnevniške datoteke, podatkovne baze, pomožne datoteke za zagon različnih programov, baza paketnega sistema itd.

  Namesto fizičnih imenikov vsebuje mehke povezave (angl. *soft link*) do ustreznih imenikov v datotečnem sistemu, ki je priklopljen za branje in pisanje (podimenik imenika `/jail/rw`).

- `/jail/rw:` za vsako ječo v njem obstaja ločen imenik, v katerem so podimeniki, specifični za to ječo. Ti imeniki so namenjeni tako branju kot pisanju, zato so z datotečnim sistemom nullfs priklopljeni v bralno-pisalnem načinu na podimenik `/jail/mnt/[ječa]/rw`.
- `/jail/mnt:` za vsako ječo v njem obstaja ločen imenik, kamor je v načinu samo-za-branje priklopljen imenik z bazo (`/jail/base`).

Grafična predstavitev strukture imenikov je na sliki 4.2.

Sistemi UNIX za nastavitev priklopov datotečnih sistemov ob zagonu sistema uporabljajo datoteko `/etc/fstab`. Ta vsebuje definicije – preslikave v obliki "naprava (`Device`) → priklopna točka (`Mountpoint`)" s še nekaj dodatnimi nastavitvami za posamezen priklop (tip datotečnega sistema (`FStype`), način priklopa (`Options`) in navodila za arhiviranje priklopne točke (`Pass#`)). Datoteka */etc/fstab* na gostiteljskem sistemu vsebuje naslednje dodatne definicije priklopov datotečnih sistemov nullfs (navedene so definicije za n ječ):

```
# Device          Mountpoint             FStype Options Dump Pass#
/jail/base        /jail/mnt/[ječa_1]     nullfs ro      0    0
/jail/rw/[ječa_1] /jail/mnt/[ječa_1]/rw  nullfs rw      0    0
/jail/base        /jail/mnt/[ječa_2]     nullfs ro      0    0
/jail/rw/[ječa_2] /jail/mnt/[ječa_2]/rw  nullfs rw      0    0
...
/jail/base        /jail/mnt/[ječa_n]     nullfs ro      0    0
/jail/rw/[ječa_n] /jail/mnt/[ječa_n]/rw  nullfs rw      0    0
```

Vrstni red priklopov za posamezno ječo je pomemben, saj priklop na točko `/jail/mnt/[ječa_n]/rw` ni mogoč, preden baza s podimenikom `rw`, ni priklopljena na `/jail/mnt/[ječa_n]`.

4.4 Storitve v ječah

Zaradi zagotavljanja kar največje varnosti je potrebno čim bolj vestno upoštevati načelo "ena ječa – ena storitev". Kljub temu včasih ni mogoče (ali vsaj ni smiselno) ločiti nekaterih komponent, saj je njihova medsebojna odvisnost tako močna, da jih je zelo težko razporediti po različnih ječah. V primeru poštnega strežnika bo tako v eni sami ječi kar veliko različnih programov, ki se med seboj dopolnjujejo.

Predlagani koncept predpostavlja, da bi razvijalci v okviru razvojnega cikla pripravili posamezne ječe kot monolitne pakete, ki bi jih uporabnik nato enostavno namestil in nastavil z uporabo grafičnega vmesnika.

4.4.1 Uporabniki in skupine

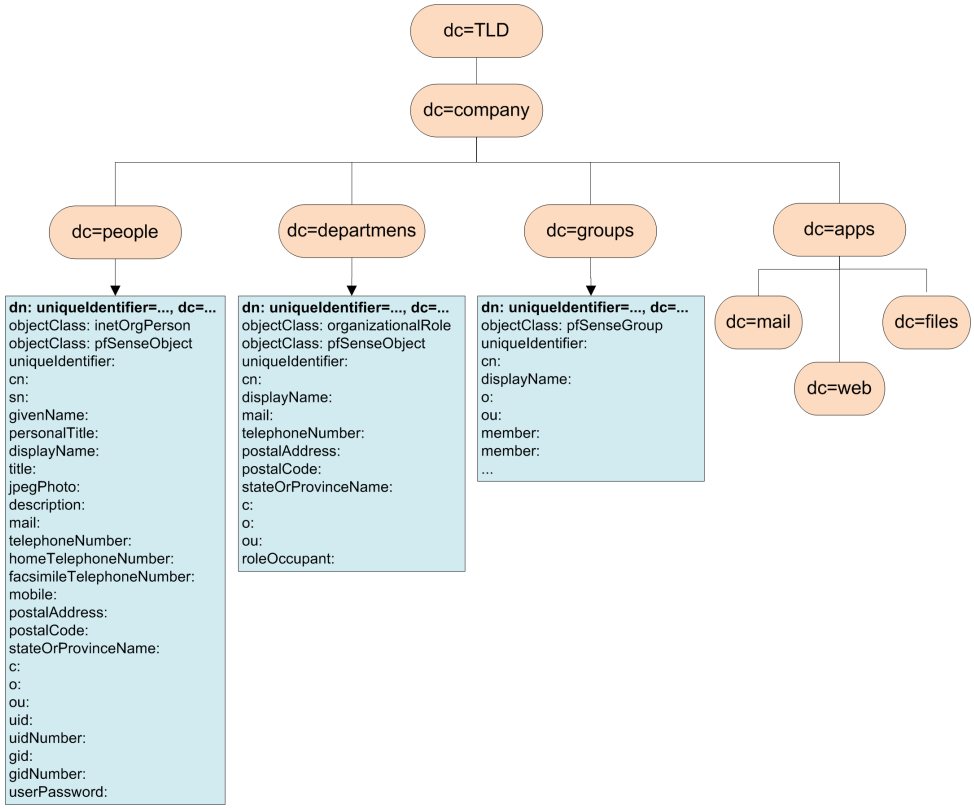
Sistem pfSense za potrebe vzdrževanja že vsebuje lasten sistem za upravljanje z uporabniki in skupinami, ki za hranjenje uporabnikov in skupin privzeto uporablja kar datoteko XML, v kateri so vse druge nastavitve. Ta način je zelo preprost in nudi le najosnovnejše funkcionalnosti (hrani le nekatere osnovne podatke o uporabnikih in skupinah). Največja pomanjkljivost pa je ta, da so vsi ti podatki na voljo le programom, ki tečejo na sistemu pfSense in ne na celotnem (krajevnem) omrežju.

Za potrebe ostalih storitev je zato zaželena zmogljivejša shramba podatkov o uporabnikih. V ta namen smo uporabili odprtokodni izdelek OpenLDAP, ki je prosta izvedba imeniškega sistema LDAP.

LDAP je protokol, ki deluje po načelu strežnik – odjemalec. Strežnik hrani drevesno strukturirane objekte z natančno določenimi atributi. Tipe objektov in atributov določa shema LDAP (angl. *scheme*). Ker potrebujemo imenik, ki bo hranil podatke o objektih, uporabljenih v več različnih storitvah, je potrebno na podlagi standardne sheme LDAP, ki je definirana v dokumentih RFC, zasnovati lastno shemo LDAP. Za to je treba najprej pridobiti lasten identifikator objektov (OID). Če bi nadgradnjo sistema pfSense z ječami FreeBSD izvedli razvijalci kar v okviru projekta pfSense, bi bilo najbolje pri organizaciji IANA pridobiti lasten OID za projekt pfSense. Njegova pridobitev je brezplačna.

Razlog za izbiro izdelka OpenLDAP je preprost – je zanesljiv, zrel in dobro preizkušen odprtokodni strežnik LDAP. Poleg tega obstoječi sistem za upravljanje uporabnikov omogoča priklop na imenik LDAP, kar pomeni, da je lahko hranjenje podatkov o uporabnikih in skupinah popolnoma centralizirano v celotnem sistemu pfSense.

Poraja se vprašanje, zakaj ne bi bilo za hranjenje vseh teh podatkov bolje izbrati relacijske podatkovne baze. Razlogov je več. Imenik LDAP je danes standardni način za hranjenje hierarhično urejenih podatkov. Če imamo torej podatke, ki so večinoma urejeni drevesno oz. na način starš – otrok, je LDAP primernejši od relacijske podatkovne baze. Poleg tega sam protokol LDAP omogoča spreminjanje sheme, kar je težje izvedljivo pri RDBMS. Nenazadnje omogoča LDAP preprosto sinhronizacijo strežnikov LDAP. Slaba stran večine teh je počasnost zapisovanja in spreminjanja podatkov, zato je primernejši za okolja, kjer je branj veliko več kot pisanj. Naše okolje temu ustreza, saj bomo v imeniku LDAP hranili podatke o osebah, oddelkih, skupinah, dovoljenjih ter nekatere nastavitve programov. “Težkih” podatkov (elektronska pošta, datoteke datotečnega strežnika), ki se hitro spreminjajo, imenik LDAP ne bo



Slika 4.3: Osnutek drevesne strukture imenika LDAP. Z roza barvo so označeni razredi objektov, v modrih okvirih pa so navedeni atributi za posamezne razrede.

vseboval.

Ker je zasnova sheme LDAP razmeroma zahtevna naloga, je na tem mestu ne bomo podrobneje opisovali, ampak samo navedli okvirno strukturo, kjer so vidni najnujnejši tipi objektov in atributov. Približek sheme takšne strukture je v obliki entitet z atributi na sliki 4.3.

4.4.2 Elektronska pošta

Eden najbolj zapletenih podsistemov v smislu nastavitve in vzdrževanja je poštni podsistem oziroma ječa, v kateri bodo tekle poštne storitve. Težava je v tem, da je potrebno razmeroma širok nabor različnih programov natančno prilagoditi, da skupaj delujejo kot usklajena celota. Poleg tega odločitev o izbiri programov, ki bodo opravljali delo posameznih komponent podsistema,

ni enostavna, saj je možnost izbire na tem področju v odprtokodnem svetu zelo velika.

Po natančni preučitvi in temeljitem testiranju smo ugotovili, da lahko z naslednjim naborom različnih samostojnih odprtokodnih izdelkov sestavimo stabilen in robusten e-poštni podsistem:

- **Postfix:** izdelek, ki ga razvija Wietse Venema, velja za enega kakovostnejših odprtokodnih e-poštni strežnikov (MTA). Kot referenčno organizacijo, ki ga uporablja, lahko navedemo zavod ARNES. Služil nam bo kot e-poštni strežnik za protokol SMTP, torej za pošiljanje pošte drugim strežnikom SMTP, in za prejemanje pošte tako od drugih strežnikov SMTP kot od odjemalcev, ki uporabljajo ta strežnik za pošiljanje. V tem primeru bomo omogočili uporabo strežnika SMTP za pošiljanje pošte na naslove, katerih e-poštni predali niso na njem samem, ampak jih je potrebno posredovati naprej ustreznemu strežniku SMTP. Ker bi bila možnost pošiljanja pošte zaradi morebitnih zlorab preko strežnika SMTP za vse odjemalce prenevarna (pošiljanje nezaželene pošte), je potrebno odjemalce avtenticirati. Zato bomo uporabili razširitev protokola SMTP, ki se imenuje SMTP-AUTH. Postfix prav tako skrbi za dostavo elektronske pošte v ustrezne e-poštne predale (z uporabo v nadaljevanju opisanega vmesnika, ki ga nudi strežnik Dovecot), za klic zunanjih programov za protivirusno pregledovanje elektronske pošte in za nadzor nad nezaželeno pošto. Omogoča tudi uporabo varnega povezovanja z uporabo protokola SSL oz. TLS.

- **Dovecot:** glavni razvijalec je Timo Sirainen, prvič je bil izdan julija 2002. Tudi Dovecot je uveljavljen strežnik za protokola IMAP in POP3, ki torej odjemalcem omogoča dostop do e-poštnih predalov. Podpira več vrst e-poštnih predalov: mbox, Maildir, dbox in Cydir, od katerih sta zadnja dva značilna za izdelek Dovecot, torej jih drugi strežniki IMAP in POP3 ne poznajo. V našem primeru bomo uporabili vrsto dbox, ker ima vsak izmed preostalih kakšno pomanjkljivost:

  - **mbox**: način zapisa je preprost in deluje tako, da so vsa sporočila hranjena v eni sami datoteki, kar pomeni težave s hitrostjo nekaterih operacij nad posameznimi sporočili;
  - **Maildir**: način zapisa je danes sicer zelo pogost in standarden, vendar je počasnejši in tudi že nekoliko star. Deluje tako, da je vsako sporočilo v svoji datoteki, kar pri izvajanju določenih operacij pomeni veliko dela za datotečni sistem;

  - **Cydir**: najnovejši Dovecotov način zapisa e-poštnih predalov in še ni primeren za produkcijsko uporabo.

  Vrsta dbox združuje dobre lastnosti vrst mbox in Maildir, saj jo lahko uporabimo na dva načina: ena datoteka na sporočilo, kot pri vrsti Maildir, ali ena datoteka za več sporočil, vendar v primerjavi z mbox tudi več datotek na poštni predal (to trenutno še ne deluje povsem). Glavni razlog za hitro delovanje vrste dbox pa je, da uporablja Dovecotove indeksne datoteke za hranjenje večine ključnih besed in oznak sporočil. Poleg tega je dbox razširljiv in omogoča zanimive nadgradnje, kot so: več predalov v eni shrambi dbox, priponke za več predalov so na disku zapisane samo enkrat.

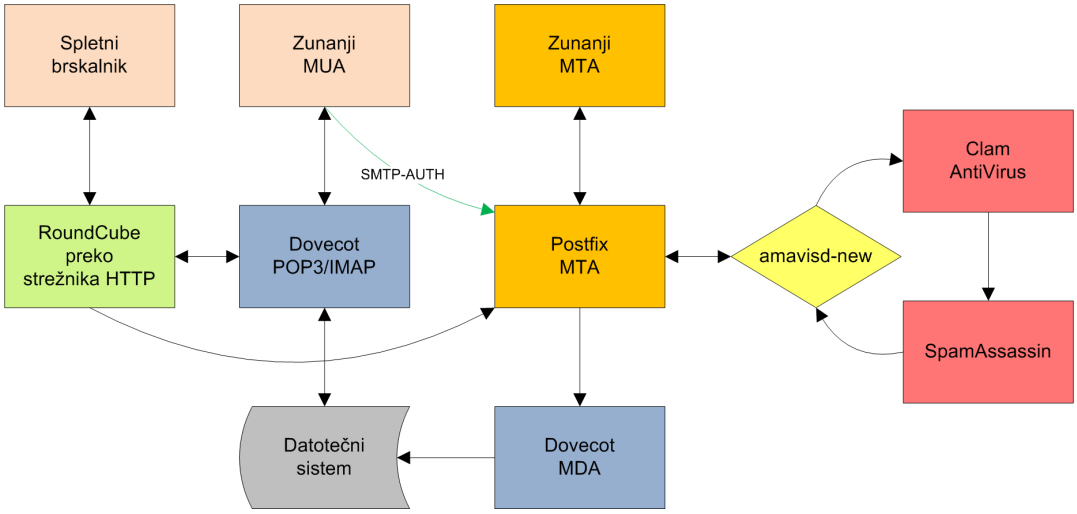
  Tudi Dovecot omogoča varno povezovanje (šifriranje povezave) med odjemalci in strežnikom z uporabo protokola SSL oz. TLS.

- **SpamAssassin:** program, ki z različnimi metodami poskuša ločiti neželeno elektronsko pošto (angl. *spam*) od želene. V resnici je to pomožen program, ki ga bo klical program amavisd-new. Za uspešno ločevanje uporablja SpamAssassin naslednje metode: Bayesov klasifikator, ki ga lahko "uči" uporabnik sam z ustrezno povratno informacijo; vrsto javnih črnih list, ki omogočajo iskanje pošiljateljevega naslova IP; metode, ki temeljijo na kontrolnih vsotah sporočil, kot sta Distributed Checksum Clearinghouses in Razor; metodo SPF, ki temelji na zapisih v strežnikih DNS.

- **Clam AntiVirus:** protivirusni program, ki omogoča zaznavanje in odstranjevanje računalniških virusov iz vseh vrst datotek, tudi iz takih, ki vsebujejo e-poštna sporočila. ClamAV je edini tovrstni odprtokodni program, žal pa se s komercialnimi protivirusnimi izdelki težko kosa. Tudi ta ne deluje samostojno, temveč ga kliče program amavisd-new.

- **amavisd-new:** vmesnik med strežnikom SMTP (MTA) in enim ali več pregledovalnimi programi (v našem primeru Clam AntiVirus in SpamAssassin). Program je razvil Mark Martinec z Instituta "Jožef Stefan".

- **RoundCube:** spletni vmesnik za dostop do e-poštnih predalov. Gre za zelo mlad, a obetaven izdelek, saj v primerjavi s podobnimi programi združuje visoke tehnologije (AJAX) ter hkrati ohranja preprostost za uporabo in preglednost. Zadnje različice so že primerne tudi za produkcijsko uporabo, čeprav bo treba morda na nekaterih mestih spremeniti izvorno kodo ali vključiti nekatere dodatke, ki uradno niso del



Slika 4.4: Shema poštnega podsistema. Povezave nakazujejo potek in smer komunikacije med posameznimi komponentami.

programa. RoundCube prikaže vsebino e-poštnih predalov ter omogoča pregledovanje in ustvarjanje e-poštnih sporočil. Poleg tega podpira delo z zbirkami podatkov o osebah (stikih) v imeniku LDAP.

Vse te programe je potrebno ustrezno nastaviti, kar pa že presega cilje te diplomske naloge. V postopku nastavitve je potrebno sprejeti še nekaj odločitev o tem, kateri podatki naj se hranijo v podatkovni bazi RDBMS (v našem primeru MySQL), kateri pa v imeniku LDAP. Brez podatkovne baze namreč ni mogoče hraniti nekaterih podatkov, ki se pogosto spreminjajo (npr. učni podatki za Bayesov klasifikator). Ker v tem trenutku ni popolnoma jasno, kako našteti programi podpirajo hranjenje nastavitev in podatkov za delovanje v imeniku LDAP, je pri izvedbi potrebno paziti tudi na to in se glede na ugotovitve primerno odločiti za uporabo RDBMS ali imenika LDAP.

Shema celotnega poštnega podsistema je prikazana na sliki 4.4.

4.4.3 Podatkovna baza (RDBMS)

Relacijske podatkovne baze se danes uporabljajo zelo pogosto – ne več samo za hranjenje uporabniških podatkov, temveč tudi za hranjenje nastavitev programov (poštni podsistem), metapodatkov[1] (spletne fotogalerije) itd. V tem primeru bomo uporabili odprtokodni izdelek MySQL podjetja AB, ki je na

[1] To so podatki o podatkih.

voljo pod hibridno licenco. Ker je tudi pfSense odprtokodni izdelek, je v našem primeru na voljo pod licenco GNU GPL. Poleg MySQL bi našim potrebam ustrezal tudi izdelek PostgreSQL, ki pa je nekoliko počasnejši, zahtevnejši za uporabo in vsebuje funkcionalnosti, ki so namenjene naprednejši uporabi.

Nastavitev ječe s podatkovno bazo MySQL ni posebno težavna, nekaj pazljivosti je treba pri določanju velikosti podatkovnih datotek, ki jih baza nato hrani v delovnem pomnilniku (RAM). Ker je pfSense omrežna naprava, ni nujno, da ima na voljo veliko pomnilnika RAM, zato je ob namestitvi ječe smiselno od vzdrževalca zahtevati, naj sam določi želeno porabo, privzeta vrednost pa naj bo primerno nizka.

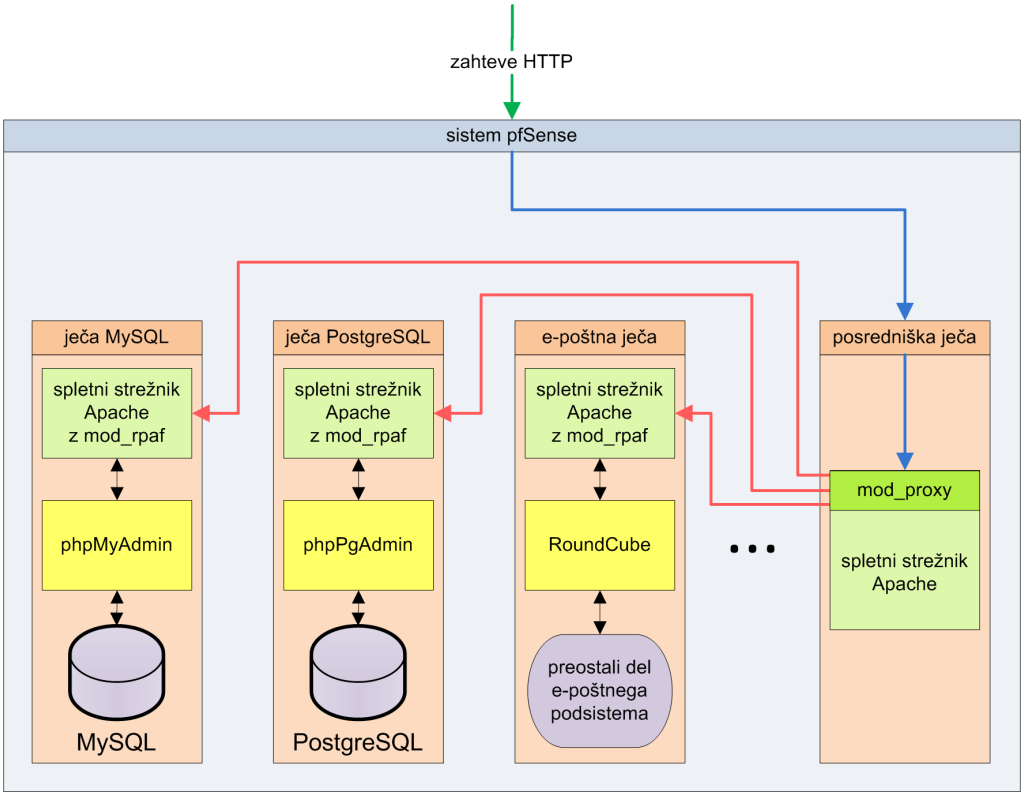
Za upravljanje s podatkovno bazo ni smiselno programirati lastnega vmesnika znotraj spletnega vmesnika pfSense, saj je to razmeroma kompleksno. Zato je primerneje uporabiti že uveljavljen spletni vmesnik phpMyAdmin, ki se ga lahko prilagodi v smislu grafičnega poenotenja s spletnim vmesnikom pfSense. Vmesnik phpMyAdmin za svoje delovanje potrebuje spletni strežnik s podporo jeziku PHP.

Za vzpostavitev ustreznih podatkovnih baz, uporabniških imen in gesel za potrebo drugih storitev v ječah bodo poskrbele njihove namestitvene skripte. Za ostalo vzdrževanje (ustvarjanje novih baz in uporabniških imen za spletne strani) pa bo imel vzdrževalec na voljo preprost vmesnik znotraj vmesnika pfSense.

Če se pojavi potreba po drugačni programski opremi za podatkovno bazo (denimo že prej omenjeni PostgreSQL), se lahko pripravi ločeno ječo, ki jo bo vsebovala. Za upravljanje baze PostgreSQL prav tako obstaja uveljavljen spletni vmesnik phpPgAdmin, ki za delovanje potrebuje spletni strežnik s podporo jeziku PHP.

4.4.4 Posredniški spletni strežnik

Postavlja se vprašanje, ali naj imajo ječe (za podatkovno bazo, elektronsko pošto ...) svoj lasten spletni strežnik za potrebe poganjanja spletnih vmesnikov ali naj se razvije poseben "posredniški" (angl. *proxy*) spletni strežnik. Ta bi bil ločen podsistem (ječa) in bi preusmerjal zahteve HTTP, ki pridejo na določene naslove, v ustrezne ječe. To bi bilo mogoče z uporabo dveh dodatkov za spletni strežnik Apache: `mod_proxy`, ki izvaja dejanske preusmeritve na strani posredniškega spletnega strežnika, in `mod_rpaf`, ki skrbi za ustrezno ohranitev izvornih naslovov IP posameznih zahtev HTTP (drugače končni spletni strežnik zahteve vidi, kot da prihajajo od posredniškega spletnega strežnika, in ne od izvornih odjemalcev). Nagibamo se k drugi rešitvi (slika 4.5), ki



Slika 4.5: Prikaz posredovanja HTTP zahtev, ki ga izvaja posredniški spletni strežnik z uporabo `mod_proxy` dodatka za Apache spletni strežnik.

omogoča večjo neodvisnost posameznih ječ, čeprav morda za ceno nekoliko večje porabe sistemskih sredstev (predvsem pomnilnika RAM). Vseeno pa je to končna odločitev programerjev, ki bi rešitev razvili, saj se bo optimalna rešitev pokazala med razvojem in testiranjem.

4.4.5 Datotečni in tiskalniški strežnik

Na področju odprtokodnih datotečnih in tiskalniških strežnikov imata izdelka Samba in CUPS tako rekoč monopol, zato se bomo omejili nanju.

Nastavitev datotečnega strežnika Samba sama po sebi ni težavna, vendar ga je v našem primeru potrebno nastaviti za sodelovanje z imenikom LDAP, kar nastavitev nekoliko oteži. Večje težave pa se pojavijo, ko želimo strežnik Samba poganjati znotraj ječe FreeBSD.

Samba namreč posnema operacijske sisteme Microsoft Windows in nji-

hove protokole (skupino protokolov CIFS). Ena izmed lastnosti sistemov Windows je ta, da brez prestanka poslušajo na vratih 137 in 138 UDP. Vzrok je specifikacija protokola NetBIOS, ki omogoča enostavno preslikavanje imen računalnikov[2] v omrežne naslove IP. To v praksi omogoča, da se računalniki med seboj "vidijo" po imenih NetBIOS, ne da bi bilo treba uporabnikom vedeti naslove IP. Operacijski sistemi ali programi, ki uporabljajo protokol NetBIOS, pošiljajo poizvedbe na t. i. naslov IP za razpršeno oddajanje[3] (angl. *broadcast address*), saj tako poizvedbo prejmejo vsi računalniki v krajevnem omrežju. Če ime katerega izmed računalnikov ustreza tistemu v poizvedbi, se ta javi neposredno povprašujočemu računalniku.

Težava nastopi, ker ječe FreeBSD omogočajo programom, ki tečejo v njih, poslušanje samo na enem naslovu IP – na naslovu sáme ječe (glej poglavje 3.1.2). Strežnik Samba, če teče v ječi FreeBSD, tako ne more poslušati na naslovu za razpršeno oddajanje, kar pomeni, da ne vidi povpraševanj, ki jih pošiljajo drugi računalniki. To tudi pomeni, da se ne odziva na zahteve, ki se sklicujejo na njegovo ime NetBIOS.

Rešitev, ki bi se je lahko poslužili takoj, je enostavna: poganjajmo datotečni strežnik Samba v gostiteljskem sistemu FreeBSD. To seveda pomeni izenačenje programa Samba s preostalimi dodatnimi paketi, ki so že na voljo v sistemu pfSense, kar ne ustreza zamisli o visoki stopnji varnosti, zapisani v ciljih tega dela.

Zato je treba poiskati trajnejšo rešitev: idealno bi bilo, če bi lahko vzdrževalec gostiteljskega sistema določil, katere ječe lahko dovolijo poslušanje na naslovih za razpršeno oddajanje in katere ne. To bi bila zanimiva funkcionalnost, ki bi jo bilo potrebno vgraditi v jedro FreeBSD ali zanjo navdušiti odgovornega razvijalca FreeBSD.

Tiskalniški strežnik CUPS vsebuje lasten (vgrajeni) spletni vmesnik za upravljanje, ki teče na vratih 631 (TCP), zato je njegova nastavitev trivialna. Vgraditev vmesnika v obstoječi spletni vmesnik pfSense je možna na dva načina: z razvojem ustreznega vmesnika, ki bi znal komunicirati s strežnikom CUPS (težji način), ali pa se vmesnik pfSense prilagodi tako, da se vmesnik za upravljanje strežnika CUPS v spletnem brskalniku prikaže znotraj vmesnika pfSense. To je sicer precej cenena rešitev, vendar bi večini uporabnikov najverjetneje zadostovala.

[2]To niso imena DNS računalnikov, temveč imena NetBIOS, kar je značilno le za računalnike, ki uporabljajo protokol NetBIOS.

[3]To je najvišji (zadnji) naslov IP naslovnega prostora omrežja IP.

4.4.6 Spletni strežnik

Posebnost spletnih strežnikov je v tem, da lahko obremenitev (število zahtev na časovno enoto) lahko skozi čas zelo niha. To pomeni, da se težko pripravimo na dogodke, kot je na primer t. i. učinek Slashdot [4].

Zato je spletni strežnik, ki bo gostil obsežnejše in dobro obiskane spletne strani, navadno dobro namestiti na ločeno strojno opremo, namenjeno samo njemu. Kljub temu pa se včasih pojavi potreba po postavitvi manj zmogljivega spletnega strežnika: za spletno stran manjšega podjetja, društva, šole ipd. Statične spletne strani danes niso več tako razširjene kot nekoč, zato se večina izdelovalcev odloča za uporabo podatkovne baze in dinamičnih programskih jezikov, kot je PHP. Zato bo tudi podsistem s spletnim strežnikom omogočal uporabo jezika PHP. Ustvarjanje potrebnih podatkovnih baz, uporabniških imen in gesel za dostop do njih bi se sicer dalo deloma avtomatizirati, vendar ta opravila zaenkrat lahko prepustimo vzdrževalcem sistema pfSense (ali vzdrževalcem ječe za spletni strežnik).

Vsekakor bi bilo potrebno razviti spletni vmesnik za enostavno upravljanje spletnih map in za spreminjanje nastavitvenih možnosti strežnika Apache, predvsem raznih dovoljenj in možnosti za posamezne mape. Za Apache obstajajo nekateri dodatki, ki omogočajo hranjenje dela nastavitev v podatkovni bazi. Eden takšnih je dodatek `mod_shapvh`, ki omogoča hranjenje nastavitev o navideznih gostiteljih (ta možnost omogoča streženje spletnih strani z različnimi naslovi URL z enega samega spletnega strežnika) v podatkovni bazi MySQL. Podobno delo opravlja `mod_ldapcfg`, ki omogoča hranjenje nastavitev v imeniku LDAP. Odločitev o tem, kateri dodatek bi bilo v tem primeru bolje uporabiti, bo moral sprejeti razvijalec, saj brez testiranja ni mogoče zagotoviti enakovrednosti teh dodatkov.

Uporabniki (izdelovalci spletnih strani) danes za prenos vsebin na spletni strežnik večinoma še vedno uporabljajo protokol FTP, kar se odsvetuje iz več razlogov. Prvi je zagotovo ta, da protokol FTP ni varen, saj ne podpira šifriranja povezave. Drugi pa je ta, da je bil protokol FTP zasnovan za delo v zaupanja vrednem omrežju, torej omrežju brez požarnih zidov in preslikavanja naslovov IP v druge (NAT). Vse te značilnosti modernih omrežij namreč pomenijo težave za delovanje protokola FTP, ki včasih niso enostavno premostljive.

Zato predlagamo, da se za dostop do spletnih strani uporabi protokol SFTP, ki ni prav nič podoben protokolu FTP, saj deluje na podlagi protokola SSH. To

[4](Pre)obremenitev, ki doleti spletni strežnik, če je naslov gostujoče spletne strani objavljen na dobro obiskani novičarski strani, kot je na primer http://www.slashdot.org.

pomeni, da je potrebno uporabljati drugačne odjemalce (na primer WinSCP za operacijski sistem Microsoft Windows), saj odjemalci FTP navadno ne podpirajo protokola SFTP. Ta je namreč deloma tudi kriptografski protokol, ki je za izvedbo nekoliko kompleksnejši. Druga slaba stran protokola SFTP je tudi ta, da se uporabnike neprimerno težje "omeji". Večina strežnikov FTP danes podpira navidezne uporabnike, torej takšne, ki nimajo svojega sistemskega uporabniškega imena in gesla (zapisanega v datoteki `/etc/passwd`), temveč le uporabniško ime in geslo, zapisano v poljubni nesistemski shrambi (pogosta je podatkovna baza RDBMS). To omogoča večjo varnost, saj se uporabnik ne more prebiti do datotek celotnega sistema, ampak ima na voljo samo svoj domači imenik FTP. Kljub tej pomanjkljivosti je tveganje majhno. Operacijski sistem FreeBSD slovi med drugim ravno po tem, da navaden uporabnik brez skrbniških privilegijev teh skoraj nikoli ne more pridobiti z raznimi zlonamernimi programi, kar na primer ne drži za operacijski sistem Linux. Od različice 4.3 naprej ni bila v FreeBSD odkrita nobena taka varnostna luknja. Poleg tega gre v tem primeru za ječo. To pomeni, da v najslabšem primeru morebitni napadalec še vedno ogrozi samo spletni strežnik, ne pa ostalih ječ ali gostiteljskega sistema.

4.4.7 Imenski strežnik

Ena izmed storitev, ki ji je smiselno dodeliti lastno ječo, je tudi imenski strežnik (DNS). Navadno se ta v organizaciji lahko uporabi na dva načina:

- **Krajevne domene:** za potrebe preslikavanja imen krajevnih omrežnih naprav (računalnikov, tiskalnikov, usmerjevalnikov ...) v krajevne naslove IP se ponekod vzdrževalci sistemov odločajo za uporabo krajevnega strežnika DNS. Tak strežnik deluje tako, da če prejme povpraševanje za preslikavo imena, ki se nanaša na domeno, za katero je avtoritativen[5], na povpraševanje odgovori. V nasprotnem primeru (navadno) posreduje povpraševanje naprej – največkrat imenskemu strežniku internetnega ponudnika. Krajevni imenski strežnik lahko povežemo tudi s strežnikom DHCP (če oba razumeta standard RFC 2136), ki nato imenskemu sporoča trenutno veljavne preslikave imen v naslove IP.
- **Javne domene:** za potrebe preslikovanja imen domen, ki so registrirane pri registrarjih in so dosegljive preko interneta. Imenski strežnik, ki služi samo temu namenu, lahko za vsa ostala povpraševanja (za domene, za katere ni avtoritativen) vrne napako – to je popolnoma normalno.

[5]To pomeni, da je odgovoren za hranjenje preslikav DNS za to domeno.

Zelo uveljavljen imenski strežnik je BIND, katerega razvoj krije organizacija ISC. Vendar pa ta strežnik omogoča hranjenje nastavitev in preslikav samo v navadnih datotekah, kar je včasih omejujoče. Zato predlagamo, da se v ječi poganja imenski strežnik PowerDNS, ki omogoča vrsto načinov hranjenja preslikav. Referenčnih namestitev izdelka PowerDNS je kar nekaj (registrar Register.com, vrhnja domena .mp, nemški registrar 1and1.com ...), kar priča o njegovi kvaliteti.

Ker razvoj vmesnika za upravljanje s strežnikom PowerDNS ni preveč kompleksen (razmeroma običajne operacije na preprostih tabelah podatkovne baze), bi bilo najbolje razviti vmesnik v okviru ogrodja pfSense. Drugače pa se lahko uporabi kakšnega izmed mnogih že obstoječih vmesnikov, kot so PowerAdmin, PDNS-Admin itd.

4.4.8 Sistem za nadzor različic

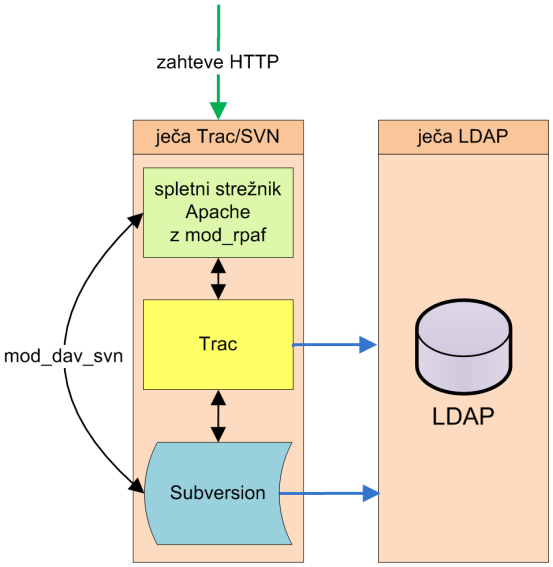
Program za nadzor različic in spletni vmesnik za komunikacijo med uporabniki lahko bistveno pripomoreta k organizaciji in izvedbi projektov, tudi ne na strogo računalniškem področju (razvoj programske opreme). Zato je smiselno eno izmed ječ nameniti prav za to namenjeni programski opremi.

Izdelek CVS je uveljavljen in preizkušen sistem za nadzor različic, saj ga je Dick Grune razvil že v osemdesetih letih prejšnjega stoletja. Kljub temu ima za današnjo uporabo kar nekaj pomembnih omejitev, ki včasih znatno otežujejo delo. Nekaj primerov: CVS ne zna nastavljati različic datotek in imenikov, ki jih premikamo ali preimenujemo; omejena podpora datotekam Unicode in drugim ne-ASCII datotekam; netransakcijske izvedbe sprememb datotek (kar lahko povzroči okvare na datotekah, če se sistem CVS sesuje ob neprimernem času).

Da bi odpravili večino najbolj neprijetnih težav izdelka CVS, je podjetje CollabNet inc. leta 2000 izdalo prvo različico programa Subversion (SVN). Danes ga uporabljajo znani projekti, kot so Apache Software Foundation, Free Pascal, GCC, FreeBSD, Python.

Posebno dobro zna Subversion sodelovati z orodjem Trac, ki nudi spletni vmesnik za shrambo SVN, sistem za sledenje hroščev, želenih izboljšav in dodatkov – torej za delo z "listki" (angl. *tickets*), pa tudi okolje wiki. Poleg tega omogoča pregled nad dogodki (spreminjanje datotek v shrambi SVN in na strani wiki), kar zagotavlja preglednost.

Ječa mora tako vsebovati spletni strežnik Apache, Subversion in programsko opremo Trac. Namestitev ni prezahtevna, potrebno je spet nekaj pozornosti pri nastavitvi orodja Trac, saj bi bilo primerno, da uporablja za av-



Slika 4.6: Shema delovanja ječe s sistemom za nadzor različic.

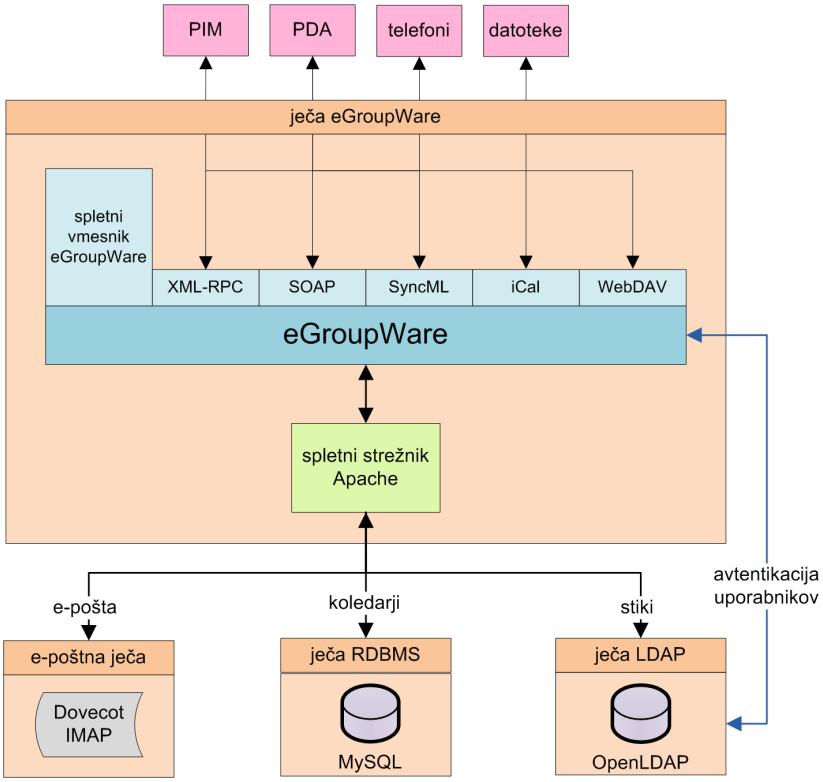
tentikacijo imenik LDAP, pa tudi za avtorizacijo. Da bi to dosegli, je potrebno namestiti in nastaviti dodatek LdapPlugin za Trac in na strežni Apachek namestiti dodatke za protokol WebDAV, posebno tistega, ki omogoča delo s shrambo SVN, to je `mod_dav_svn`. Shema delovanja te ječe je na sliki 4.6.

4.4.9 Okolje za skupinsko delo

Predvsem v poslovnih okoljih so danes dobrodošle rešitve, ki omogočajo uporabo istega vmesnika za vrsto različnih storitev. Primer iz Microsoftovega sveta je uporabniški program Microsoft Outlook, ki v sodelovanju s strežnikom Microsoft Exchange omogoča poleg e-poštnih storitev tudi deljene koledarje, stike, zapiske, naloge ipd. V povezavi s strežnikom Microsoft Project omogoča tudi uporabo orodij za vodenje projektov.

V odprtokodnem svetu obstaja mnogo programov, ki skušajo združiti prej naštete funkcionalnosti, med katerimi so najbolj vidni Kolab, OpenGroupware.org, Scalix, Zimbra in eGroupWare. Zadnji se za uporabo v okviru izdelka pfSense zdi primeren, saj temelji na jeziku PHP in potrebuje spletni strežnik ter podatkovno bazo. Možna je tudi povezava z imenikom LDAP za sámo prijavljanje v eGroupWare kot tudi za e-poštni imenik (angl. *address book*).

Program eGroupWare je kompleksen in zahteva precej pozornosti za pravilno nastavitev, sploh v okviru našega sistema, v katerem želimo, da vse



Slika 4.7: Shema povezav ječe eGroupWare z drugimi ječami ter zunanjimi napravami in viri podatkov.

storitve uporabljajo enoten vir avtentikacijskih podatkov (imenik LDAP). Kljub temu je okolje za skupinsko delo ena pomembnejših storitev za uporabnike, saj bistveno pripomore k dobri uporabniški izkušnji.

Na sliki 4.7 je shema povezav med ječo eGroupWare ter ostalimi ječami in drugimi morebitnimi napravami in programi. Ta shema je zgolj osnutek in ne odraža nujnega končnega stanja povezav.

4.5 Možnosti nadaljnjih nadgradenj

Zaradi odprtokodne narave izdelkov, ki smo jih uporabili v celotnem predlogu nadgradnje sistema pfSense z uporabniškimi storitvami, je možnosti za nadaljnje delo še veliko. Dve izmed njih, ki se nam zdita najbolj zanimivi, bomo opisali v naslednjih podpoglavjih.

4.5.1 Integracija z imenikom Microsoft ADS

Veliko informacijskih sistemov danes sestavljajo pretežno računalniki, na katerih je nameščena ena izmed različic operacijskega sistema Microsoft Windows. Za lažje vzdrževanje osebnih računalnikov zaposlenih, pa tudi drugih računalnikov s sistemom Windows, se je zelo uveljavila uporaba tehnologije Microsoft Active Directory (pogosto tudi ADS). To je LDAP-u podoben imeniški sistem, ki nudi tudi druge storitve: centralizirano avtentikacijo (z uporabo t. i. prijave v domeno AD) na podlagi protokola Kerberos, preslikavanje imen v naslove IP z uporabo strežnika DNS itd. Gre za lastniško tehnologijo podjetja Microsoft, vendar jo je zaradi LDAP-u podobne narave možno uporabiti tudi z drugo programsko opremo.

Zelo zanimiva bi torej bila možnost uporabe imenika ADS namesto posebne ječe z imenikom LDAP. Problem je sicer vse prej kot trivialen, saj v tem primeru ne moremo po svoje oblikovati sheme LDAP. Zato bi bilo verjetno potrebno razviti nekakšen "vmesni sloj", ki bi hranil potrebne podatke, ki jih ne moremo hraniti v imeniku ADS. To bi bile, denimo, nekatere nastavitve programske opreme, nekatera dovoljenja (kateri uporabniki smejo dostopati do katerih storitev) ipd. Za vmesni sloj bi lahko uporabili tudi ločen imenik LDAP, podatkovno bazo ali celo preprosto datoteko XML.

4.5.2 Centralizirane nastavitve več sistemov pfSense

V večji organizaciji, kjer morda potrebujemo več sistemov pfSense za različne segmente omrežij ali za različne funkcionalnosti (poštni strežnik, požarni zid ...), lahko postane vzdrževanje vsakega sistema posebej z uporabo spletnih vmesnikov težaško opravilo. Ena izmed rešitev bi bila že deljenje istega imenika LDAP (ječe). Imenik, nameščen samo na enem sistemu pfSense, bi bilo mogoče uporabiti na vseh ostalih. Tako bi imeli poenoteno shrambo uporabnikov in skupin ter nekaterih nastavitev.

Kljub temu pa bi bila še bolj zanimiva funkcionalnost centraliziranega vmesnika za vse sisteme pfSense. Da bi to dosegli, bi bilo potrebno razviti poseben vmesnik API, ki bi omogočal hranjenje nastavitev ter operacije (brisanje, ustvarjanje) nad nastavitvami in sistemom samim (ponovni zagon storitev, celotnega sistema). V tem primeru bi bilo potrebno razmisliti o načinu uporabe skupnega vmesnika. Če naj bo to spletni vmesnik, mora biti nekje nameščen. Najbolj smiselno bi bilo določiti enega izmed sistemov pfSense za "gospodarja", ostale pa za "služabnike". Morda pa bi bil smiseln tudi razvoj posebne programske opreme (denimo v programskem jeziku Java ali C++), ki bi tekla na različnih operacijskih sistemih. To je seveda obsežen zalogaj, vendar je takšen

sistem vsekakor mogoče razviti in pridobiti uporabnike (morda tudi take, ki bi bili pripravljeni takšno dodatno storitev plačati).

Poglavje 5

Vzdrževanje ječ

5.1 Posodabljanje ječ

Osnovno vodilo pri načrtovanju postopka posodabljanja ječ, ki tečejo v sistemu pfSense, je enostavnost. Posodobitve morajo biti preproste (izvedljive z nekaj preprostimi koraki v grafičnem vmesniku) in varne. To pomeni, da mora sistem brezhibno delovati tudi po nadgradnji in da ni možnosti za izgubo uporabniških podatkov. Poleg tega bi bila zelo dobrodošla funkcionalnost, ki bi omogočala razveljavitev nadgradnje in povrnitev v stanje pred nadgradnjo. To bi prišlo prav, ko uporabnik z nadgradnjo ne bi bil zadovoljen ali pa ta zaradi kakršnega koli razloga ne bi pravilno delovala.

Nadgradnje naj bi bile na voljo v obliki paketov na strežnikih projekta pfSense, kot je to sedaj urejeno z obstoječimi dodatki pfSense. Tako se posodobitve vedno sproti prenesejo z interneta, kar zagotavlja ažurnost. Ponekod (kjer zaradi varnostnih ali drugih razlogov ni prostega dostopa do interneta) bi gotovo bila uporabna tudi funkcionalnost "odklopljene" (angl. *off-line*) posodobitve. To pomeni, da bi vzdrževalec posodobitveni paket, ki ga je predhodno prenesel z interneta, naložil preko spletnega brskalnika.

5.1.1 Posodabljanje baze

Posodabljanje osnovnega sistema (baze) v `/jail/base` je razmeroma kompleksno, saj je potrebno veliko pazljivosti, da ne povzročimo regresij programske opreme. Od iste baze je namreč odvisnih več ječ in vsaka ječa morda uporablja del baze, ki ga druge ječe ne uporabljajo. To pomeni, da lahko z nadgradnjo mimogrede povzročimo nezdružljivo spremembo, ki se bo poznala v nedelovanju kakšne ječe.

Da bi zmanjšali verjetnost regresij, bi bilo potrebno pripraviti vrsto regresijskih testov, ki bi po nadgradnji baze preverjali delovanje čim širšega spektra funkcionalnosti vseh ječ. To zahteva razmeroma veliko dela, vendar edino tako lahko zagotovimo največjo verjetnost, da bodo ječe delovale tudi po nadgradnjah.

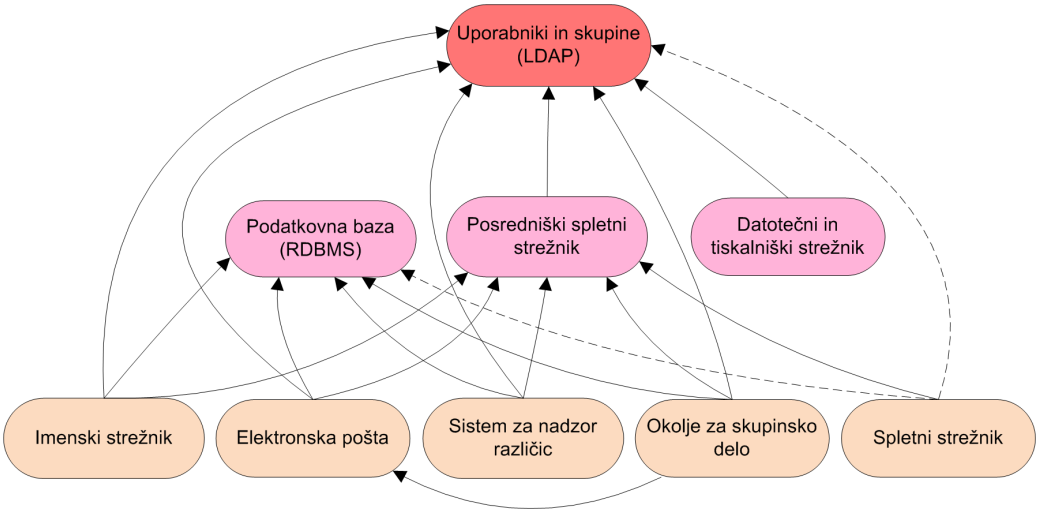
Če regresijski testi kakšne napake kljub testiranju ne bi zaznali, mora biti omogočena razveljavitev nadgradnje. To se najlažje doseže tako, da je izdelava rezervne kopije trenutne baze vključena v postopek nadgradnje. To sicer nekoliko podaljša celoten postopek, vendar omogoči hitro povrnitev v staro stanje.

Poskrbeti je potrebno tudi za to, da so nadgradnje izvedene transakcijsko, za kar imamo dva načina. Izkoristimo lahko mehke povezave (ukaz `ln -s`). V tem primeru bi bilo potrebno celoten sistem ječ zasnovati tako, da bi bila datoteka `/jail/base` mehka povezava na imenik, npr. `/jail/base.MM-DD-YY@HH-MM-SS`, kjer so `MM`, `DD`, `YY`, `HH`, `MM` in `SS` po vrsti mesec, dan, leto, ure, minute in sekunde. Na koncu postopka nadgradnje se samo pobriše mehko povezavo in ustvari novo, ki kaže na novi imenik z drugačnim imenom (saj bo ustvarjen ob drugačnem času). Drugi način pa je z uporabo premikanja imenikov. Postopek nadgradnje pričnemo tako, da ne okrnemo trenutnega delujočega sistema, kar pomeni, da paket z novimi datotekami namestimo v ločen imenik, na koncu postopka nadgradnje pa starega preimenujemo v arhivski imenik, novega pa v izvorni imenik. Ukaz za preimenovanje `mv` deluje zelo hitro, saj le popravi kazalce v datotečnem sistemu. Obe rešitvi sta podobni, le da je prva preglednejša, saj omogoča, če se odločimo ohraniti vse stare imenike, lepše zapisovanje zgodovine.

Zgornje razmišljanje upošteva, da se pri posodobitvi z interneta prenese celoten posodobljeni paket (in ne samo spremembe), torej monolitno posodobitev. Če bi želeli sistem nadgraditi samo z datoteko sprememb (diferencialna posodobitev), je potrebno drugače zastaviti arhiviranje trenutnega sistema. Najbolj smiselno bi bilo prekopirati celotno bazo v drug začasni imenik in tja nato namestiti posodobljene datoteke. Od tukaj dalje pa velja zgornje razmišljanje.

5.1.2 Posodabljanje ječ s storitvami

V primerjavi z bazo imamo pri ječah opravka tudi z uporabniškimi podatki, zato je bolj smiselno uporabiti diferencialne posodobitve namesto monolitnih, če je to le mogoče. Ob nadgradnji celotnega sistema pfSense (vključno z jedrom FreeBSD) bi bilo potrebno v celoti nadgraditi tudi bazo in ječe. V tem primeru diferencialne posodobitve ne pridejo v poštev, zato bo treba pripraviti



Slika 5.1: Odvisnosti med posameznimi ječami.

postopke, ki bodo prenesli podatke v nove ječe (in jih po potrebi prilagodili).

5.2 Upravljanje ječ in uporabniški vmesnik

Pomembno je, da upravljanje nadgrajenega sistema pfSense ostane enostavno in pregledno. Glede na to, da je celoten sistem ječ popolnoma ločen od preostalega sistema (oz. da pfSense lahko deluje kot doslej brez njih), je v grafičnem spletnem vmesniku smiselno dodati nov zavihek samo za ta namen, torej za upravljanje z ječami in storitvami v njih. Poleg obstoječih zavihkov Sistem (angl. *System*), Vmesniki (angl. *Interfaces*), Požarni zid (angl. *Firewall*), Storitve (angl. *Services*), VPN, Status in Diagnostika (angl. *Diagnostics*) je treba dodati zavihek Podsistemi (angl. *Subsystems*).

Po sveži namestitvi nadgrajenega sistema pfSense bo pod zavihkom Podsistemi na voljo le možnost Nastavitve (angl. *Configuration*). Tam bo mogoče vključiti ali izključiti podsisteme ter, ko bodo ječe vključene, upravljati uporabnike in skupine (torej z vnosi v imeniku LDAP). Vključitev ječ torej pomeni namestitev baze in ječe z imenikom LDAP.

Izkaže se, da med ječami veljajo nekatere odvisnosti, ki so prikazane na sliki 5.1. To so kritične odvisnosti, kar pomeni, da, denimo, ječa za elektronsko pošto ne more delovati brez ječe z imenikom LDAP in ječe s podatkovno bazo. Zato je prva ječa, ki jo lahko namestimo, prav ječa z imenikom LDAP. To pomeni, da je struktura uporabnikov in skupin popolnoma neodvisna od vseh

ostalih ječ. Odvisnost spletnega strežnika od podatkovne baze in strežnika LDAP je poljubna, saj spletni strežnik lahko streže tudi statične strani, ki ne potrebujejo povezljivosti s podatkovno bazo ali strežnikom LDAP. Odvisnosti se v praksi poznajo tako, da moramo, preden namestimo ječo, odvisno od druge ječe, najprej namestiti zahtevano ječo.

V istem zavihku Podsistemi se glede na dodane ječe prikazujejo dodatne možnosti, za vsako ječo po ena. Z izbiro teh možnosti nato nastavljamo posamezne ječe (nastavitve poštnega, spletnega ali drugega strežnika).

Poglavje 6

Sklepne ugotovitve

Po analizi obstoječega izdelka pfSense lahko rečemo, da je ta primeren za zasnovo novega, izboljšanega sistema, ki bo poleg sistemskih omrežnih podpiral tudi širok nabor uporabniških storitev. Razlog za to je predvsem njegova odprtokodna zasnova ter premišljen in nadgradljiv koncept centraliziranih nastavitev. Poleg tega dejstvo, da sistem temelji na operacijskem sistemu FreeBSD, samo po sebi pomeni garancijo za zanesljiv in robusten sistem, kar je pomembna lastnost omrežnih operacijskih sistemov.

Ječe FreeBSD, ki so bistveni del naše nadgradnje, so dovolj preproste in hkrati prožne, da lahko z njimi zgradimo skoraj popolnoma izolirane podsisteme, ki bodo nudili okolje operacijskega sistema programom uporabniških storitev. Glavni razlog za njihovo uvedbo je doprinos k varnosti celotnega sistema, ki bi jo lahko uporabniške storitve sicer ogrozile. Okolje ječ je na več načinov omejeno le na tiste funkcije operacijskega sistema, ki jih programi uporabniških storitev nujno potrebujejo.

Pregled drugih sistemov, podobnih načrtovanemu, je nakazal nekatere smernice in ideje za načrtovanje nabora uporabniških storitev, ki jih je smiselno ponuditi v novem sistemu. Pri načrtovanju ječ s storitvami smo ugotovili, da je kot pomožni datotečni sistem, ki bi služil varčevanju z diskovnim prostorom, najbolje izbrati datotečni sistem nullfs, saj bi uporaba sistema unionfs, ki smo ga tudi preučili, prinesla določene težave. Poleg tega smo zasnovali strukturo imenikov, ki naj bi ji morebitni razvijalci sledili skozi celoten nadaljnji razvoj. Nadalje smo izbrali nabor programske opreme za posamezne ječe in zasnovali nekatere povezave med njimi. Ta del načrtovanja je najzahtevnejši, saj je večidel odvisen od ugotovitev, ki bi jih prinašal sprotni razvoj.

V zadnjem poglavju smo navedli še nekaj zamisli glede vzdrževanja ječ s storitvami, ki bodo lajšale izvedbo postopkov nadgradenj. Te navadno za-

htevajo veliko izkušenj in "uvida v prihodnost", saj ni enostavno načrtovati sistema, ki bo ostal nadgradljiv in nespremenjen v nekaj let odmaknjeni prihodnosti. Bistveni ideji sta predvsem enostavnost in varnost oz. zanesljivost, ki ju je mogoče doseči z načrtovanjem nadgradenj kot samostojnih paketov in s temeljitim testiranjem. Transakcijske posodobitve in nadgradnje pa so pogoj za konsistentnost sistema tudi ob nepričakovanih dogodkih, kot je izpad električne energije.

Na podlagi povedanega in izkušenj s samostojnimi strežniki, ki nudijo uporabniške storitve, lahko ugotovimo, da je problem integracije več uporabniških storitev v en sam izdelek izjemno kompleksen. Razlogov za to je več. Največji je verjetno ta, da je skoraj vsak izdelek razvila druga skupina razvijalcev z različnimi poudarki v različnih programskih jezikih in okoljih. Gre torej za izjemno heterogen nabor programov, ki se razlikujejo tudi po svoji zrelosti in stabilnosti. Po drugi strani imamo nemalo težav zaradi veliko različnih protokolov, ki včasih služijo istemu namenu. Večino teh protokolov mora namreč programska oprema podpirati, če jo želimo povezati z drugimi programi, predvsem uporabniškimi. Tipičen primer je protokol POP3 za odjemalce elektronske pošte – strežnik POP3 mora namreč podpirati različne izvedbe istega protokola, saj prihaja pri nekaterih proizvajalcih programske opreme do odstopanj pri tolmačenju specifikacij protokolov. Tipična primera takih razmeroma dvoumnih protokolov sta tudi IPSec in IMAP.

Kljub temu smo z izsledki testiranj uspešno pripravili osnutek razvojnega dokumenta, na podlagi katerega se razvoj (nadgradnja) lahko začne. Ker je snov obsežna, bi lahko skoraj za vsako izmed ječ (predvsem za tiste zahtevnejše, denimo za ječo s sistemom za elektronsko pošto in ječo za skupinsko delo) napisali razvojni dokument, ki ne bi bil krajši od tega diplomskega dela. Testiranje posameznih nastavitev vseh komponent ječ je namreč zelo obsežno in zamudno, saj večina odprtokodnih programov nudi izjemno prožnost pri uporabi in nastavitvi ter zagotavlja obilo prostora za domišljijo.

Nadaljevanje tega dela bi torej bilo potrebno začeti z natančnejšim načrtom – predvsem imenika LDAP tin njegove sheme, pa tudi nekaterih ječ, predvsem ječe za skupinsko delo. V nadaljevanju bi bilo seveda potrebno vse načrtovano tudi razviti.

Slike

Tabele

Seznam uporabljenih kratic in simbolov

ADS – (Microsoft) Active Directory Service (lastniški protokol za imeniške storitve)

AJAX – Asynchronous JavaScript and XML (skupina tehnik za razvoj spletnih aplikacij)

API – Application programming interface (vmesnik, ki določa funkcije operacijskega sistema ali računalniškega programa, ki so na razpolago drugemu računalniškemu programu)

ARM – Advanced RISC Machine (napredni RISC računalnik, tip RISC arhitekture CPE)

ASCII – American Standard Code for Information Interchange (ameriški standard za zapis črkovnih, številskih, posebnih in krmilnih znakov, med katerimi ni znakov za šumnike in preglase)

AT&T – AT&T Inc. (ameriško komunikacijsko podjetje)

ATM – Asynchronous Transfer Mode (protokol za prenos enako dolgih podatkovnih paketov, namenjen večjim prenosnim hitrostim)

BGP – Border Gateway Protocol (internetni protokol za prenos usmerjevalnih informacij med avtonomnimi sistemi)

BIND – Berkeley Internet Name Domain (najpogosteje uporabljani imenski strežnik (strežnik DNS) za sisteme UNIX)

BSD – Berkeley Software Distribution (izvedba sistema UNIX, ki so jo konec sedemdesetih let začeli razvijati Bill Joy in sodelavci na kalifornijski univerzi v Berkeleyju)

CIFS – Common Internet File System (omrežni protokol za izmenjavo datotek)

CUPS – Common Unix Printing System (modularen tiskalniški sistem za operacijske sisteme UNIX, ki omogoča, da računalnik nudi storitve tiskalniškega strežnika)

CVS – Concurrent Versions System (sistem za nadzor vzporednih različic izvorne kode, zasnovan na sistemu RCS)

devfs – device file system (datotečni sistem za datoteke naprav)

DHCP – Dynamic Host Configuration Protocol (protokol, ki omogoča dinamično uskladitev naslovov IP z računalniki v krajevnem omrežju)

DMZ – DeMilitarized Zone (demilitarizirano območje, koncept ločitve zunanjih strežnikov od krajevnega omrežja)

DNS – Domain Name System (sistem, ki se v internetu uporablja za preslikavo med domenskimi imeni in naslovi IP)

DoS – Denial-of-Service (dejanje, ki prepreči informacijskemu sistemu ali njegovemu delu normalno delovanje)

Ethernet – družina omrežnih tehnologij za krajevna računalniška omrežja

FTP – File Transfer Protocol (aplikacijski protokol, ki skrbi za prenos datotek med omrežnimi vozlišči)

GBDE – GEOM Based Disk Encryption (šifriranje diska, ki temelji na ogrodju GEOM)

GEOM – modularno vhodno/izhodno diskovno ogrodje

GID – Group IDentifier (identifikator uporabniške skupine)

GNU – GNU's Not UNIX (projekt, začet v osemdesetih letih, ki si je za cilj zastavil prosto izvedbo sistema UNIX)

GPL – General Public Licence (dovoljenje za uporabo programja, ki spremlja vse programe projekta GNU in mnoge druge proste programe)

HDLC – High-Level Data Link Control (protokol za visokostopenjski nadzor nad podatkovno povezavo)

HTTP – HyperText Transfer Protocol (protokol za izmenjavo nadbesedil ter grafičnih, zvočnih in drugih večpredstavnostnih vsebin na spletu)

HTTPS – HyperText Transfer Protocol over Secure Socket Layer (protokol za varno izmenjavo nadbesedil ter grafičnih, zvočnih in drugih večpredstavnostnih vsebin na spletu)

IMAP – Internet Message Access Protocol (protokol za dostop do elektronske pošte)

I/O – Input/Output (vhod/izhod)

IANA – Internet Assigned Numbers Authority (organizacija za dodeljevanje internetnih številk)

IP – Internet Protocol (protokol omrežnega sloja v protokolnem skladu TCP/IP)

IPsec – Internet Protocol Security (zbirka ukrepov za zagotavljanje varnostnih storitev v protokolu IP)

ISC – Internet Systems Consortium (neprofitna organizacija, ki se ukvarja z razvojem internetnih tehnologij)

ISDN – Integrated Services Digital Network (digitalno omrežje, v katerem se digitalne centrale in prenosni mediji uporabljajo za vzpostavitev povezav za integrirane storitve)

JID – Jail IDentifier (identifikator ječe FreeBSD)

L2TP – Layer 2 Tunneling Protocol (protokol za tuneliranje omrežnega prometa drugega sloja OSI)

LAMP – Linux-Apache-MySQL-PHP (skupek odprtokodne programske opreme, ki tvori popolnoma delujoč spletni strežnik)

LDAP – Lightweight Directory Access Protocol (protokol za dostop do imenikov na osnovi modela strežnik – odjemalec)

MAC – Media Access Control (naloga povezavnega sloja, ki ureja dostop posamezne naprave do medija)

MDA – Mail Delivery Agent (programska oprema, ki dostavlja e-poštna sporočila v ustrezne poštne predale)

MIME – Multipurpose Internet Mail Extensions (večnamenske razširitve internetne pošte)

MIPS – Microprocessor without Interlocked Pipeline Stages (tip arhitekture RISC CPE)

MTA – Mail Transfer Agent (strežniški program, ki skrbi za prepošiljanje elektronske pošte s protokolom SMTP)

MUA – Mail User Agent (uporabniški program za ravnanje z elektronsko pošto)

Multi-WAN – možnost uporabe več povezav WAN sočasno

MySQL – odprtokodni sistem za upravljanje z relacijsko podatkovno bazo

NAT – Network Address Translation (omogoča prevod naslova IP, ki ga uporablja eno omrežje, v drug naslov IP, ki ga prepozna drugo omrežje)

NetBIOS – Network Basic Input/Output System (omogoča programom na različnih računalnikih, da med seboj komunicirajo preko krajevnega omrežja)

NOS – Network Operating System (omrežni operacijski sistem)

OPIE – One Time Passwords in Everything (omogoča uporabo gesel za enkratno uporabo v raznih storitvah UNIX)

OID – Object Identifier (identifikator objektov za razvoj shem LDAP, SNMP ali drugih)

OpenVPN – odprtokodni program za upravljanje povezav VPN

OSI – Open Systems Interconnection (referenčni model, ki definira funkcije povezovanja v sedmih slojih)

PAM – Pluggable Authentication Modules (mehanizem za vgradnjo nizkostopenjske avtentikacije v visokostopenjske vmesnike API)

PC – Personal Computer (osebni računalnik, namenjen enemu uporabniku)

PHP – PHP: Hypertext Preprocessor (splošno uporaben skriptni programski jezik, ki ga tolmači strežnik in je namenjen izdelavi dinamičnih spletnih strani)

POP3 – Post Office Protocol version 3 (protokol za prenos elektronske pošte s strežnika k odjemalcu)

PPTP – Point-to-Point Tunneling Protocol (protokol za oddaljeno povezavo računalnika s tunelom skozi internet)

PPP – Point-to-Point Protocol (protokol na dostopovnem vodu od uporabnika do ponudnika internetnih storitev)

PPPoE – Point-to-Point Protocol over Ethernet (protokol PPP preko omrežja Ethernet)

RADIUS – Remote Authentication Dial In User Service (omrežni protokol za centralizirano upravljanje dostopa, avtorizacije in obračunavanja)

RAID – Redundant Array of Independent Disks (več fizičnih diskov v računalniškem sistemu, povezanih kot logična enota, s čimer je dosežena večja zmogljivost in/ali zanesljivost)

RAM – Random-Access Memory (bralno-pisalni neobstojni pomnilnik z naključnim dostopom)

RDBMS – Relational Database Management System (sistem za upravljanje relacijskih podatkovnih baz)

RFC – Request For Comments (oznaka dokumentov, ki določajo tehnične vidike interneta, predvsem njegove protokole)

RISC – Reduced Instruction Set Computing (računalnik s procesorjem, ki uporablja malo preprostih ukazov)

SFTP – SSH File Transfer Protocol (omrežni protokol, ki omogoča prenos in delo z datotekami preko varne povezave SSH)

S/MIME – Secure/Multipurpose Internet Mail Extensions (varni protokol MIME)

SMTP – Simple Mail Transfer Protocol (protokol za prenos eletronske pošte)

SMTP-AUTH – razširitev protokola SMTP za podporo avtentikaciji odjemalcev

SPF – Sender Policy Framework (vrsta zaščite proti nezaželeni pošti, ki temelji na ustreznih zapisih za domene v strežnikih DNS)

SQL – Structured Query Language (strukturiran povpraševalni jezik za delo s podatkovnimi bazami)

SSH – Secure SHell (omrežni protokol, ki omogoča prenos podatkov preko šifriranega kanala med dvema omrežnima napravama)

SSL – Secure Sockets Layer (protokol, ki omogoča šifrirano povezavo med strežnikom in odjemalcem)

TACACS+ – Terminal Access Controller Access-Control System Plus (protokol za nadzor dostopa do usmerjevalnikov)

TCO – Total Cost of Ownership (vsi stroški razvoja, uporabe in vzdrževanja informacijskega sistema)

TCP – Transmission Control Protocol (povezavni protokol transportnega sloja v protokolnem skladu TCP/IP)

TCP/IP – Transmission Control Protocol/Internet Protocol (standardiziran sklad protokolov, na katerem temelji internet)

TLD – Top-Level Domain (vrhnji del domen, ki se uporabljajo v sistemu DNS)

TLS – Trans Layer Security (naslednik protokola SSL)

UDP – User Datagram Protocol (nepovezavni protokol transportnega sloja v protokolnem skladu TCP/IP)

UID – User IDentifier (identifikator uporabnika)

Unicode – standardni kodirani nabor znakov, razvit z namenom zajeti znake vseh pisav sveta; koordiniran s standardom ISO/IEC 10646

UNIX – večopravilni večuporabniški operacijski sistem, ki sta ga l. 1969 v Bellovih laboratorijih začela razvijati Ken Thompson in Dennis Ritchie; v letih 1972-1974 prepisan v programski jezik C

URI – Universal Resource Identifier (zgoščen niz znakov, ki označuje ali poimenuje vir na internetu)

URL – Universal Resource Locator (URI, ki poleg poimenovanja vira navaja tudi protokol in kraj, preko katerih je vir dostopen)

VLAN – Virtual LAN (skupina gostiteljev, ki komunicirajo v isti razpršeni domeni (*angl. broadcast domain*) ne glede na svojo fizično lokacijo)

VPN – Virtual Private Network (telekomunikacijska storitev, ki zagotavlja naročniku zasebno omrežje, realizirano z viri javnega omrežja)

XML – Extensible Markup Language (oblika zapisa podatkov za izmenjavo strukturiranih dokumentov)

WAN – Wide Area Network (omrežje, ki pokriva večje geografsko področje)

WebDAV – Web-based Distributed Authoring and Versioning (množica razširitev za protokol HTTP, ki omogoča uporabnikom skupno delo z datotekami na spletnih strežnikih)

Literatura

[1] *Wikipedia: The Free Encyclopedia.* Wikimedia Foundation Inc. Posodobljeno 20. junija 2008, 15:04 UTC. Spletna enciklopedija. Dostopno na:
http://en.wikipedia.org/wiki/Network_operating_system. Internet. Preneseno 1. avgusta 2008.

[2] *pfSense Official Documentation.* Posodobljeno 11. junija 2007, 07:06 UTC. Dostopno na:
http://doc.pfsense.org/index.php?title=Introducing_pfSense&action=history. Internet. Preneseno 2. avgusta 2008.

[3] *pfSense Developers Wiki.* Posodobljeno 23. julija 2008, 00:07 UTC. Dostopno na:
http://devwiki.pfsense.org/PfSenseDevHome. Internet. Preneseno 6. avgusta 2008.

[4] G. Lehey, *The Complete FreeBSD®, Fourth Edition*, O'Reilly Media, Inc., 2003.

[5] D. Lavigne, *FreeBSD: An Open Source Alternative to Linux.* Posodobljeno 8. avgusta 2006, 19:35 UTC. Dostopno na:
http://www.freebsd.org/doc/en/articles/linux-comparison/freebsd-features.html. Internet. Preneseno 6. avgusta 2008.

[6] B. Golden, *Succeeding With Open Source*, Addison-Wesley Professional, 2005.

[7] Y. Korff, P. Hope, B. Potter, A. Randal, *Mastering FreeBSD and OpenBSD security*, O'Reilly Media, Inc., 2005.

[8] P. Kamp, R. Watson, *Jails: Confining the omnipotent root*, SANE 2000 Conference proceedings.

Izjava

Izjavljam, da sem diplomsko nalogo izdelal samostojno pod vodstvom mentorja prof. dr. Boruta Robiča. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

Ljubljana, 5. september 2008, Nejc Škoberne