



Admin Training Center

L2-1 Osnove administriranja Linuxa

Admin Training Center
Studentski trg 4, VI sprat
11000 Beograd, Srbija
<http://www.atc.rs/>



Copyright ©2014 Veselin Mijušković, Ljubiša Radivojević, Marko Uskoković.

Ovaj tekst je licenciran pod Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. Da biste videli kopiju ove licence posetite:
<http://creativecommons.org/licenses/by-nc-nd/4.0/>

Sadržaj

<u>Uvod - tipografske konvencije.....</u>	7
<u>Tipografske konvencije.....</u>	7
<u>Upravljanje diskovima.....</u>	9
<u>Označavanje diskova.....</u>	9
<u>Moderni diskovi.....</u>	9
<u>IDE diskovi.....</u>	9
<u>Ostali načini označavanja diskova.....</u>	10
<u>Particionisanje diskova.....</u>	10
<u>Označavanje particija.....</u>	10
<u>Programi za particionisanje diskova pod Linuxom.....</u>	11
<u>fdisk.....</u>	11
<u>GNU parted.....</u>	12
<u>Fizički fajl sistemi i formatiranje particija.....</u>	13
<u>Sopstveni fajl sistemi.....</u>	14
<u>Pozajmljeni fajl sistemi.....</u>	15
<u>Strani fajl sistemi.....</u>	15
<u>Formatiranje particija.....</u>	15
<u>Pripremanje swap particije.....</u>	17
<u>Kačenje i otkačinjanje uređaja.....</u>	17
<u>Rad sa swap uređajima.....</u>	19
<u>Fajl /etc/fstab.....</u>	20
<u>Provera ispravnosti fajl sistema i ispravljanje grešaka.....</u>	21
<u>Upravljanje logičkim volumenima.....</u>	25
<u>Uvod.....</u>	25
<u>Arhitektura LVM.....</u>	25
<u>PV – fizički volumeni.....</u>	25
<u>Kreiranje fizičkih volumena.....</u>	26
<u>Listanje fizičkih volumena.....</u>	26
<u>Promena veličine fizičkog volumena.....</u>	27
<u>Uklanjanje PV.....</u>	27
<u>VG – volumenske grupe.....</u>	28
<u>Kreiranje VG.....</u>	28
<u>Dodavanje PV u VG.....</u>	28
<u>Izbacivanje PV iz VG.....</u>	28
<u>Listanje volumenskih grupa.....</u>	28
<u>Skeniranje VG i rekreiranje keš fajlova.....</u>	29
<u>Aktiviranje i deaktiviranje VG.....</u>	29
<u>Uklanjanje VG.....</u>	30
<u>Preimenovanje VG.....</u>	30
<u>LV – logički volumeni.....</u>	30
<u>Kreiranje LV.....</u>	30
<u>Kreiranje snapšot volumena.....</u>	31
<u>Promena veličine LV.....</u>	33
<u>Preimenovanje LV.....</u>	34
<u>Aktiviranje i deaktiviranje LV.....</u>	34

<u>Uklanjanje LV</u>	34
<u>Listanje LV</u>	35
<u>Praćenje sistemskih resursa</u>	37
<u>Praćenje stanja sistema</u>	37
<u>Praćenje CPU</u>	37
<u>Komanda 'ps'</u>	37
<u>Komanda 'top'</u>	38
<u>Praćenje memorije</u>	39
<u>Komanda 'free'</u>	39
<u>Praćenje podsistema diskova i zauzeća prostora na njima</u>	40
<u>Komanda 'blkid'</u>	40
<u>Komanda 'df'</u>	41
<u>Komanda 'du'</u>	42
<u>Praćenje ostalog hardvera</u>	43
<u>Komanda 'lspci'</u>	43
<u>Komanda 'lsusb'</u>	44
<u>Komanda 'lscpu'</u>	45
<u>Upravljanje softverskim paketima</u>	47
RedHat Package Manager – RPM	47
<u>RPM paket</u>	47
Razrešavanje problema međuzavisnosti paketa	48
<u>Komanda rpm</u>	48
Instaliranje softverskih paketa	49
Deinstaliranje paketa	50
Postavljanje upita	50
Upravljanje DEB paketima	50
Deb softverski paketi	50
Alati za upravljanje paketima	51
<u>Instaliranje softvera iz repozitorijuma</u>	53
APT repozitorijumi	53
<u>APT podsistem</u>	53
YUM podsistem	55
Ažuriranje paketa	55
Pretraživanje repozitorijuma	56
Prikaz informacija o paketima	58
Instaliranje paketa	59
Uklanjanje paketa	59
Dodavanje novih repozitorijuma	60
Aktiviranje i deaktiviranje repozitorijuma	62
Brisanje yum keševa	62
<u>Umrežavanje servera</u>	65
Pojam mreže	65
IP adrese	65
<u>Klase i opsezi adresa</u>	65
IP mreže	67
Rutiranje	67
Podrazumevana ruta (default gateway)	67
Privatne mreže	67

<u>DNS</u>	68
<u>Podešavanje mrežnih parametara iz komandne linije</u>	68
<u>Komanda 'ifconfig'</u>	68
<u>Komanda 'route'</u>	69
<u>Mapiranje DNS imena u IP adresu i obatno</u>	71
<u>Komanda 'ip'</u>	71
<u>Rutiranje pomoću komande 'ip'</u>	72
<u>Konfigurisanje mrežnog interfejsa</u>	72
<u>Upravljanje IP adresama</u>	73
<u>Osnovni mrežni konfiguracioni fajlovi</u>	74
<u>/etc/hosts</u>	74
<u>/etc/resolv.conf</u>	74
<u>Sistemsko podešavanje parametara mreže na Debian distribucijama</u>	75
<u>Sistemsko podešavanje parametara mreže na RedHat baziranim distribucijama</u>	75
<u>Konfiguracioni fajlovi za interfejse</u>	76
<u>Statičke rute</u>	77
<u>Administriranje korisnika i grupa</u>	79
<u>Dodavanje novog korisnika</u>	79
<u>Modifikovanje parametara korisnika</u>	80
<u>Brisanje korisnika</u>	80
<u>Zaključavanje i otključavanje korisnikovog naloga</u>	80
<u>Struktura /etc/passwd i /etc/shadow fajlova</u>	81
<u>Skeleton direktorijum</u>	82
<u>Administriranje grupa</u>	83
<u>Dodavanje nove grupe</u>	83
<u>Brisanje grupe</u>	83
<u>Učlanjivanje korisnika u grupu</u>	83
<u>Fajl /etc/group</u>	83
<u>Administriranje udaljenih servera</u>	85
<u>SSH pristup</u>	85
<u>SSH pristup preko komandne linije</u>	85
<u>Generisanje javnog i tajnog ključa za SSH autentifikaciju</u>	86
<u>Kopiranje fajlova iz komandne linije na udaljeni računar</u>	87
<u>Konfigurisanje osnovnih lokalnih servisa</u>	89
<u>Konfigurisanje osnovnih lokalnih servisa na RedHat i izvedenim distribucijama</u>	89
<u>Upotreba chkconfig programa</u>	89
<u>Prikaz servisa i njihovog statusa u različitim ranlevelima</u>	89
<u>Aktiviranje servisa</u>	90
<u>Deaktiviranje servisa</u>	90
<u>Pokretanje servisa iz komandne linije</u>	90
<u>Dobijanje trenutnog statusa servisa</u>	91
<u>Pokretanje servisa</u>	91
<u>Zaustavljanje servisa</u>	92
<u>Restartovanje servisa</u>	92
<u>Ostale varijante 'service' komande</u>	92
<u>Cron servis</u>	93
<u>Kreiranje cron tabela</u>	93
<u>Editovanje cron tabela</u>	94

<u>Konfigurisanje osnovnih mrežnih servisa.....</u>	97
<u>Konfigurisanje OpenSSH servisa.....</u>	97
<u>Startovanje OpenSSH servisa.....</u>	98
<u>Upotreba autentifikacije bazirane na PKI.....</u>	98
<u>Instaliranje CentOS 6 servera.....</u>	101
<u>Razumevanje boot procesa.....</u>	103
<u>Uvod.....</u>	103
<u>Butovanje PC BIOS-a.....</u>	103
<u>Pokretanje instaliranog but loudera.....</u>	103
<u>Konfigurisanje GRUB but loudera.....</u>	104
<u>Pokretanje Linux kernela.....</u>	105
<u>Kačenje / fajlsistema i pokretanje init procesa.....</u>	106
<u>Pokretanje programa prilikom butovanja sistema.....</u>	108
<u>Izvršavanje programa u zadatom ranlevelu.....</u>	108
<u>Programi koji se izvršavaju u posebnim slučajevima.....</u>	109
<u>Pokretanje programa koji upravljaju terminalima.....</u>	110
<u>Menjanje tekućeg ranvela.....</u>	111

Uvod - tipografske konvencije

Skripta je podeljena na poglavlja, a na sekcije. Unapred skrećemo pažnju polaznicima da je ova skripta samo deo dokumentacije koju oni treba da koriste. Polaznicima se savetuje da pročitaju man i help strane za svaku komandu, kao i da potraže na Internetu dodatne informacije i načine kako da iste korsite.

Tipografske konvencije

Radi lakšeg snalaženja u tekstu, koristili smo neke tipografske konvencije na koje vam ovde skrećemo pažnju:

- ukoliko se uvodi neki značajan pojam, on će u prvom pomenu biti ispisан **proporcionalnim bold** tekstrom;
- boldovano su prikazane i neke **značajne tvrdnje** na koje treba obratiti pažnju u tekstu;
- *proporcionalnim italikom* su napisane reči na stranom jeziku, najčešće engleskom;
- nazivi programa, opcija i fajlova u tekstu su ispisani **neproporcionalnim fontom**

Deo teksta koji se odnosi na direktni unos korisnika i ispis računara, kao i sadržaji fajlova i sl. će biti prikazani u zasebnom bloku:

```
$ ls -F  
myscript.sh*    Vezbe/
```

Boldovanim tekstrom je prikazano ono što polaznik treba da unese onako kako je napisano u skripti. Regularnim tekstom je prikazan ispis programa koji ne treba unositi. Ukoliko unutar specifikacije komande ili dela fajla postoji tekst koji je promenljiv, on će biti dodatno prikazan *italics* stilom:

```
$ cp [opcije] source... dest
```

U ovom slučaju je boldovano prikazana sama komanda i tako je treba uneti. Bold-italikom su prikazani promenljivi parametri i to znači da na tom mestu treba uneti neku stvarnu vrednost, a ne ono što piše u komandnoj liniji. Na kraju, opcioni parametri, koji se mogu izostaviti su navedeni u srednjim zagradama []. Tri tačke (...) označavaju da je dati parametar moguće navesti više puta.

Gde god smo mislili da nešto posebno treba naglasiti, to smo naveli u zasebnom boksu, koji obično ima naslov 'Važno!' ili 'Napomena', kao ovde:

Važno!

Hard linkovi ne mogu biti kreirani za direktorijume, već samo za regularne fajlove!

Upravljanje diskovima

Označavanje diskova

Svaki od diskova ima pridružen odgovarajući blok-specijalni fajl preko kojeg se može pristupiti disku kao celini.

Moderni diskovi

Bez obzira da li su diskovi zakačeni na SATA, PATA, SCSI, USB ili neku drugu sabirnicu, Linux sve njih tretira kao jedan virtuelni SCSI lanac, tako da diskovi imaju imena nasleđena od starijih SCSI sistema:

Redni broj	Blok-specijalni uređaj
0	/dev/sda
1	/dev/sdb
2	/dev/sdc
3	/dev/sdd
4	/dev/sde
5	/dev/sdf
6	/dev/sdg
7	/dev/sdh
...	...

IDE diskovi

Na starijim sistemima, PATA diskovi su imali poseban, odvojen sistem drajvera za rad sa diskovima i drugačije označavanje samih uređaja.

IDE kontroler	Tip	Blok-specijalni uređaj
ide0	master	/dev/hda
	slave	/dev/hdb

IDE kontroler	Tip	Blok-specijalni uređaj
ide1	master	/dev/hdc
	slave	/dev/hdd
ide2	master	/dev/hde
	slave	/dev/hdf
ide3	master	/dev/hdg
	slave	/dev/hdh
...

Noviji sistemi sa IDE/PATA/SATA diskovima emuliraju SCSI set komandi tako da koriste iste oznake kao i SCSI diskovi.

Ostali načini označavanja diskova

Neki RAID kontroleri ili sistemi za virtuelizaciju, kao što je Xen, imaju svoje oznake diskova. Npr. Xen virtuelni diskovi se na Linuxu vide kao /dev/xvda, /dev/xvdb, /dev/xvdc itd.

Particionisanje diskova

Linux podržava standardno DOS PC partacionisanje diskova. Za diskove veće od 2TB koristi se noviji sistem partacionanja baziran na GUID tabelama (GPT) koji je takođe kompatibilan sa Microsoft Windows OS.

Standardno PC partacionisanje diskova, kakvo koriste i Microsoft Windows operativni sistemi, deli disk na:

- maksimalno 4 primarne particije, ili
- maksimalno 3 primarne i jednu proširenu (*extended*) particiju.

Proširena particija može sadržati maksimalno do 250 logičkih diskova.

Podaci i proširenje virtuelne memorije (*swap*) se mogu smestiti samo na primarne particije i logičke diskove.

Označavanje particija

Particije takođe imaju pridružene svoje blok-specijalne uređaje preko kojih im se može pristupati direktno. Označavanje je identično za sve diskove. Naziv blok-specijalnog fajla je identičan nazivu diska kojoj particija pripada, sa dodatkom rednog broja

particije, po sledećem rasporedu:

Redni broj	Tip	Blok-specijalni uređaj
1	primarna ili proširena	/dev/sda1
2	primarna ili proširena	/dev/sda2
3	primarna ili proširena	/dev/sda3
4	primarna ili proširena	/dev/sda4
5	prvi logički disk	/dev/sda5
6	drugi logički disk	/dev/sda6
7	treći logički disk	/dev/sda7
...

Kao što se vidi, označavanje logičkih diskova počinje od rednog broja 5, bez obzira da li na disku uopšte postoje sve primarne particije.

Programi za particionisanje diskova pod Linuxom

Pod Linuxom se generalno koriste dva programa za particionisanje diskova: **fdisk** i **parted**.

fdisk

Komanda **fdisk** se može koristiti za formatiranje diskova manjih od 2TB i ima sledeći format:

```
# fdisk [opcije] blok-uređaj
```

gde su najčešće opcije:

-l prikazuje postojeće particije na zadatom disku bez ulaska u interaktivni mod

U interaktivnom modu najčešće su sledeće komande:

m pomoć

- p** prikazuje privremenu particionu tabelu
- n** kreira novu particiju
- d** briše postojeću particiju
- t** menja tip particije
- a** menja vrednost bootable flega
- w** zapisuje privremenu particionu tabelu na disk (ova opcija je destruktivna!)
- q** izlazak iz programa bez zapisivanja privremene particione tabele na disk.

Prilikom kreiranja particija potrebno je navesti koji tip particije želimo da kreiramo (primarni, *extended* ili logički, ako postoji kreirana *extended* particija), početni i krajnji cilindar. Umesto krajnjeg cilindra možemo navesti i željenu veličinu particije u kilobajtima ili megabajtima, u kom slučaju će program sam sračunati najpribližniju veličinu tako da se particija završava na granici cilindra.

Prilikom kreiranja particija možemo im dodeliti tip. Linux podržava veliki broj tipova particija, tako da možemo kreirati i particije koje koriste drugi operativni sistemi, npr. Microsoft Windows. Što se tiče Linuxa, on koristi sledeće tipove particija: Linux (83), Linux swap (82) i Linux LVM (8e). Prilikom menjanja tipa, unosi se ID tipa particije, naveden u zagradama.

Pošto je kreiranje particija potencijalno destruktivna operacija koja može narušiti integritet celog sistema, sve izmene se čuvaju u privremenoj particionoj tabeli koje program **fdisk** čuva u memoriji. Tek kada eksplisitno naredimo zapisivanje privremene particione tabele komandom '**w**' se izmene zapisuju na disk.

Po završetku programa potrebno je izmenjenu particionu tabelu učitati i u kernel memoriju. Ukoliko smo menjali particije na primarnom disku (disku na kojem je / particija) to nije moguće uraditi i sistem i dalje koristi staru particionu tabelu za pristup particijama primarnog diska, na šta program **fdisk** i upozorava. Iz tog razloga potrebno je restartovati sistem da bi se nova particiona tabela primarnog diska učitala u kernel memoriju. Ukoliko se radi o drugim diskovima, njihove izmenjene particione tabele će biti učitane u kernel memoriju odmah po završetku rada **fdisk** programa i nije potrebno iz tog razloga restartovati sistem.

GNU parted

GNU parted je napredniji program za particionisanje diskova koji pruža i dodatne mogućnosti kao formatiranje particija, njihovo kopiranje, prebacivanje i promenu veličine. Takođe, GNU parted može kreirati particije i na diskovima većim od 2TB.

```
# parted [blok-uredaj] [komanda [opcije...]]]
```

Ukoliko se blok-uredaj izostavi kao argument, parted će uzeti prvi blok-uredaj koji nađe.

Komande se mogu zadavati interaktivno ili u sklopu komandne linije, što je pogodno za skriptove. Komande mogu imati svoje opcije i parametre. Najčešće su:

- **help [komanda]** – ispisuje kratko uputstvo (za komandu)
- **mklabel tip** – kreira praznu particionu tabelu tipa 'tip'. Tip može biti 'msdos' (standardni tip za diskove manje od 2TB), 'gpt' (standardni tip za diskove veće od 2TB), 'bsd' (za kompatibilnost sa BSD-olikim OS), 'mac' (za kompatibilnost sa MacOS operativnim sistemima) itd.
- **mkpart tip-particije [tip-fs] početak kraj** – kreira particiju tipa 'tip-particije' ('primary', 'logical' ili 'extended'), na kojoj će biti fajl-sistem tipa 'tip-fs' ('fat16', 'fat32', 'ext2', 'HFS', 'linux-swap', 'NTFS', 'reiserfs' ili 'ufs'). Parametri 'početak' i 'kraj' se podrazumevano zadaju u megabajtima.
- **print** – ispisuje sadržaj partacione tabele
- **rm particija** – briše zadatu particiju
- **resize particija početak kraj** – zadaje novu veličinu particije
- **select blok-uredaj** – postavlja "blok-uredaj" za podrazumevani uredaj na koji deluju komande
- **set particija fleg stanje** – menja stanje flegova na zadatoj particiji. Flegovi mogu biti: 'boot', 'root', 'swap', 'hidden', 'lvm', 'lba', 'raid' itd.). Stanje može biti 'off' ili 'on'.
- **quit** – izlazak iz programa.

Napomena: za razliku od fdisk-a kod kojeg se zapisivanje izmena u particionoj tabeli dešava tek kad se eksplisitno zada komanda 'w', parted odmah zapisuje sve izmene, posle svake komande.

Fizički fajl sistemi i formatiranje particija

Fizički fajl sistem je način organizovanja podataka na nekom uredaju, najčešće particiji diska i treba ga razlikovati od logičkog fajl sistema koji predstavlja organizaciju podataka na nekom Linux sistemu.

Linux u različitoj meri podržava veliki broj raznih fizičkih fajl sistema. Ove fajl sisteme možemo podeliti na nekoliko grupa, u zavisnosti od toga u kojoj su meri povezani sa Linux operativnim sistemom. Ove grupe su:

- *sopstveni* fajl sistemi
- *pozajmljeni* fajl sistemi
- *strani* fajl sistemi

Sopstveni fajl sistemi

Pod *sopstvenim* fizičkim fajl sistemima podrazumevamo one fajl sisteme koji su razvijeni specifično za Linux. U ovu grupu spadaju ext2, ext3, ext4, reiserfs i btrfs fajl sistemi.

Ext2 ili *second extended* fajl sistem je nastao iz *extended* fajl sistema koji je bio prvi *sopstveni* fajl sistem, ali zbog svojih loših osobina nije opstao. U pitanju je standardni Unix fajl sistem sa podrškom za kvote i proširene attribute. Osobine koje ga krase su velika izdržljivost na greške i prilična brzina.

ReiserFS je fajl sistem kojeg je dizajnirala i razvila firma Namesys, i prvi je žurnalski (*journaling*) fajl sistem na Linuxu, što je donelo dodatnu robustnost i brzinu oporavka. ReiserFS se odlikuje dobrom brzinom, posebno je uspešan u radu sa velikim brojem malih ili veoma velikih fajlova. Nova verzija ovog fajl sistema Reiser4 donela je revolucionarne promene u radu sa fajlovima i tipovima podataka koristeći sopstvenu *plug-in* tehnologiju. Na žalost, ovaj sistem se više ne razvija budući da je glavni autor i vlasnik firme Namesys, Hans Reizer na odsluženju dugogodišnje kazne zatvora zbog ubistva svoje bivše supruge.

Ext3 je proširenje ext2 fajl sistema sa podrškom za žurnale. Vertikalno je kompatibilna sa ext2 fajl sistemom i zbog toga veoma popularna.

Ext4 je proširenje ext3 fajl sistema sa optimizacijom za velike particije i drugim unapređenjima.

Btrfs je najnoviji fajl sistem razvijen za Linux koji je dizajniran na principima naprednog ZFS fajl sistema koji se koristi na Solaris OS. Takođe bi trebalo da je optimizovan za upotrebu na fleš i SSD drajvovima.

Žurnalska tehnologija omogućava da se izmene koje treba da se izvrše na fajl sistemu, prevashodno u upisu metapodataka, prethodno zabeleže u posebnom delu diska, tzv. žurnalu. Žurnal periodično iščitava posebna nit kernela koja unosi izmene u sam fajl sistem po transakcionom metodu – ili će izmena biti kompletno urađena i zapis uklonjen iz žurnala ili će sistem biti vraćen u predašnje stanje. Prilikom nasilnog prekida rada sistema, sve što je potrebno da se žurnalski fajl sistem oporavi jeste da se iščitaju preostali zapisi u žurnalu, koji zbog nasilnog prekida nisu bili izvršeni i urade izmene na fajl sistemu. Na ovaj način fajl sistem uvek ostaje konzistentan.

Pozajmljeni fajl sistemi

Pozajmljeni fajl sistemi su oni fizički fajl sistemi koji se na Linuxu koriste za standardno smeštanje podataka, kao i sopstveni fajl sistemi, ali koji nisu specifično razvijeni za Linux, već su portovani sa drugih operativnih sistema. U ovu grupu spadaju JFS firme IBM, razvijen za AIX operativni sistem i XFS firme SGI, razvijen za njihovu verziju Unixa, Irix.

JFS je žurnalski fajl sistem sa solidnim karakteristikama.

XFS je takođe žurnalski fajl sistem odličnih karakteristika i sa posebnom podrškom za rad u realnom vremenu.

Strani fajl sistemi

Eksterni (*foreign*) fajl sistemi su oni fizički fajl sistemi koji su razvijeni za druge operativne sisteme, a u Linuxu postoji podrška za njih, koja je različitog nivoa. Ovi operativni sistemi se generalno ne koriste kao osnovni fajl sistemi za smeštanje podataka na Linuxu, već za mogućnost čitanja i pisanja podataka na particije i diskove koji rade pod drugim operativnim sistemima.

U ovu grupu možemo ubrojiti i fajl sisteme razvijene za prenosne medijume, kao što su CD-ROM-ovi i DVD-ROM-ovi.

Formatiranje particija

Formatiranje particije je njena priprema za prihvatanje podataka i uključivanje u logički fajl sistem. Ovom operacijom se inicijalizuju metapodaci po metodi definisanoj za odabrani fizički fajl sistem. Standardna komanda za formatiranje particija je `mkfs`, čiji je format:

```
# mkfs [opcije] [fs-opcije] blok-uredaj
```

gde su najčešće opcije:

- V prikaz više informativnih poruka o tome šta se radi
- c provera particije na loše blokove
- t **tip-fajl-sistema** specificiranje tipa fizičkog fajl sistema kojim će se formatirati particija. Ako se izostavi podrazumeva se *ext2* fajl sistem

Iza opštih opcija mogu se navesti opcije specifične za dati tip fizičkog fajl sistema (*fs-opcije*). Ove opcije ne smeju se mešati sa opštim opcijama i moraju biti navedene iza njih.

Program `mkfs` je samo zajednički interfejs ka programima za formatiranje specifičnih tipova fizičkih fajl sistema. Ovi programi imaju imena oblika `mkfs.tip-fajl-`

sistema i njima se prosleđuju opcije specifične za dati fajl sistem.

Neke opcije za *ext2* fajl sistem su:

-b *veličina-bloka* specificira veličinu bloka u bajtovima. Dozvoljene vrednosti su 1024, 2048 i 4096. Ukoliko ova opcija nije specificirana komanda će sama odrediti veličinu bloka na osnovu veličine particije i nameravane upotrebe ove particije (opcija -T)

-f *veličina-framenta* specificira veličinu fragmenta u bajtovima

-i *broj-bajtova-po-i-nodu* specificira broj bajtova po jednom kreiranom i-nodu.

Ovaj odnos definiše koliko će i-nodova biti kreirano u datoj particiji, što određuje maksimalni broj fajlova koji mogu da se kreiraju. Ovaj broj nebi trebao da bude manji od veličine bloka da se nebi razbacivao prostor za alociranje i-nodova kojih bi u tom slučaju bilo više od maksimalno mogućeg broja fajlova koji se mogu kreirati na datoj particiji

-L *labela* postavlja labelu za datu particiju

-m *procenat-rezervisanih-blokova* procenat od ukupnog broja blokova na sistemu koji se odvaja za superusera. Podrazumevano 5%.

-N *broj-i-nodova* direktno postavljanje ukupnog broja i-nodova za datu particiju.

-O *osobina*, ... lista osobina koje dati fajl sistem treba da ima. Moguće vrednosti su:

- ***dir_index*** upotreba heširanih b-stabala da bi se ubrzalo pretraživanje u velikim direktorijumima,
- ***filetype*** smeštanje tipa fajla u zapis u direktorijumu
- ***sparse_super*** kreiranje fajl sistema sa manje kopija superbloka (što štedi prostor na velikim particijama)

-T *tip-particije* utiče na više parametara kako bi se prepodesili za optimalno obavljanje zadate funkcije. Parametar tip-particije može biti:

- ***news*** jedan i-nod na jedan blok od 4KB
- ***largefile*** jedan i-nod na 1MB prostora
- ***largefile4*** jedan i-nod na 4MB prostora

Specifične opcije za *ext3* fajl sistem uključuju sve opcije za *ext2* fajl sistem i sledeće navedene:

-j kreiranje žurnala

-J *žurnal-opcija*, ... specificira žurnal opcije ako se one razlikuju od podrazumevanih:

- ***size=veličina-žurnala-u-megabajtima*** specificira veličinu internog žurnala. Žurnal ne može biti manji od 1024 x veličina-bloka niti veći od 102400 x veličina-bloka
- ***device=spoljni-žurnal*** žurnal može biti smešten na drugoj particiji koja unapred mora biti pripremljena.

Pripremanje swap particije

I swap particija mora biti prethodno pripremljena da bi mogla da posluži svojoj svrsi. Pripremanje nove swap particije se vrši komandom **mkswap**, čiji je format:

```
# mkswap [opcije] blok-uredaj
```

gde je najčešće korišćena opcija

-c testiranje swap particije na loše blokove.

Posebnu pažnju treba obratiti na činjenicu da Linux kernel ne obraća pažnju na ID particije, tako da program neće prijaviti grešku ako pokušate da neku particiju označenu kao Linux formatirate kao swap.

Kačenje i otkačinjanje uređaja

Da bi neku particiju diska, prenosni medijum (CD-ROM, DVD-ROM, flopi disk, USB disk) mogi koristiti za smeštanje podataka, potrebno je da ih prethodno uključimo u logički fajl sistem. Operacija uključivanja nekog uređaja na logički fajl sistem se naziva kačenje ili 'mauntovanje' (*mounting*). Komanda koja odradjuje ovaj posao je **mount** i ima sledeći format:

```
# mount [opcije] [fs-opcije] uređaj direktorijum
```

uređaj je particija diska, specijalni fajl asociran prenosnom medijumu ili specifikacija mrežnog diska. Direktorijum je tzv. tačka kačenja, tj. direktorijum od kojeg počinje da se vidi zakačeni uređaj.

Komanda **mount** ima opšte i opcije specifične za tip fajl sistema. Opšte opcije su:

- a** zakači sve uređaje koji su navedeni u fajlu /etc/fstab i nemaju noauto opciju navedenu
- r** zakači uređaj kao read-only
- w** zakači uređaj kao read-write (podrazumevana vrednost izuzev kod read-only uređaja kao što je CD-ROM)
- t tip-fajl-sistema** zakači uređaj kao fizički fajl sistem navedenog tipa
- o opcija, . . .** zakači uređaj sa navedenim opcijama, koje mogu biti opšte i

specifične za dati fajl sistem. Opšte opcije su:

- **async** I/O pristup treba da bude asinhroni
- **atime** ažuriraj vreme pristupa i-nodu za svaki pristup (podrazumevana opcija)
- **auto** moguće kačenje sa mount -a komandom
- **defaults** sinonim za skup opcija: rw,suid,dev,exec,auto,nouser,async
- **dev** interpretiraj blok i karakter specijalne fajlove na datom fajl sistemu
- **exec** omogući da se binarni programi sa date particije mogu izvršavati
- **noatime** suprotno od atime opcije
- **noauto** ne može biti zakačen sa mount -a komandom
- **nodev** ne interpretiraj blok i karakter specijalne uređaje na datom fajl sistemu
- **noexec** ne dozvoli izvršavanje binarnih programa na datom fajl sistemu (može se zaobići!)
- **nosuid** ne interpretiraj uid i sgid bitove
 - **nouser** zabrani običnim korisnicima da mauntuju uređaj
- **remount** ponovo zakači već zakačen uređaj na istom mestu. Ova opcija se koristi da bi se promenile mount opcije za već zakačeni fajl sistem (npr. read-only u read-write)
- **ro** zakači sistem read-only
- **rw** zakači sistem read-write
- **suid** suprotno od nosuid
- **sync** I/O pristup treba da bude sinhron
- **dirsync** ažuriranje direktorijuma treba da je sinhrono
- **user** omogući običnim korisnicima da kače uređaj
- **users** omogući da svaki korisnik može kačiti i otkačinjati uređaj (koji je zakačio neki drugi korisnik)

Za opcije specifične za pojedine tipove fajl sistema pogledajte man stranu za komandu **mount**.

Komandom

```
# mount -a
```

mogu se automatski zakačiti svi uređaji koji su u fajlu `/etc/fstab` označeni (eksplicitno ili implicitno) kao auto.

Komandom:

```
# mount
```

mogu se videti trenutno zakačeni fajl sistemi, koje opcije su korišćene za kačenje pojedinih fajl sistema i koji direktorijumi su njihove tačke kačenja.

Operacija suprotna kačenju je otkačinjanje ili 'anmauntovanje' (*unmounting*). Ovu operaciju je moguće izvesti na nekom zakačenom uređaju samo ukoliko ni jedan proces nema svoj tekući direktorijum na delu stabla direktorijuma koji se fizički nalazi na uređaju koji se želi otkačiti.

Komanda za otkačinjanje je `umount`, čiji je format:

```
# mount uređaj | tačka mountovanja
```

Dakle, kao parametar komande `umount` mogu se navesti ili uređaj, odnosno particija koju želimo da otkačimo, ili direktorijum koji je tačka kačenja za dati uređaj. Umesto ovih parametara, možemo navesti opciju `-a` kojom ćemo otkačiti sve particije osim `root` particije, a koje u fajlu `/etc/fstab` imaju, eksplisitno ili implicitno, opciju `auto`. Otkačinjanje će uspeti samo u slučaju da ni jedan proces nema svoj tekući direktorijum na podstablu koje se otkačuje, niti da postoji neki otvoreni fajl na tom podstablu. Pre same opcije će se svi podaci iz bafera, odnosno sistemske keš memorije biti snimljeni na datu particiju, kako bi se održala konzistentnost podataka i metapodataka.

Rad sa swap uređajima

Swap je produženje fizičke memorije i povećavanje efektivne virtualne memorije sistema. Sistem na *swap* particiju (ili fajl) smešta delove memorije koji se trenutno ne koriste, da bi istu oslobođio za druge procese. Kada se želi pristupiti delu memorije, bilo da su na njemu podaci ili program, a koji se trenutno nalazi na *swap* particiji, sistem će transparentno učitati taj deo memorije sa *swap* uređaja u RAM a možda neki drugi deo memorije koji se duže vreme nije koristio prebaciti iz RAM-a na *swap*.

Najbolji način za implementaciju *swap*-a je na zasebnoj particiji. Obično se za veličinu *swap* prostora uzima dvostruka veličina RAM-a, a preporuka je da je minimalna veličina jednaka veličini RAM-a (minimum za standardne 2.4 kernele verzije veće od 2.4.10 je dvostruka veličina RAM-a). *Swap* može biti i veći od dvostrukе veličine RAM-a, što čak neke aplikacije i zahtevaju. Ukupan *swap* prostor ne mora biti ceo na jednoj particiji, već može biti razbijen na više particija, što se čak i preporučuje ukoliko

postoji više diskova na računaru. Zasebnim *swap* particijama možemo dodeliti različite nivoje prioriteta koji definišu kako će se ta *swap* particija koristiti.

Komanda za uključivanje neke *swap* particije u sistem virtuelne memorije je:

```
# swapon [opcije] blok-uredaj
```

gde su najčešće opcije:

- p** **prioritet** prioritet *swap* uređaja
- a** uključuje sve *swap* uređaje navedene u fajlu /etc/fstab. Parametar *uredaj* u ovom slučaju se ne navodi
- e** opcija slična opciji -**a**, s tim što ne daje upozorenja za uređaje koji ne postoje, a navedeni su u /etc/fstab fajlu.

Komanda za isključivanje neke particije iz sistema virtuelne memorije je:

```
# swapoff blok-uredaj
```

gde se, umesto uređaja, tj. particije koja se isključuje iz sistema virtuelne memorije može navesti opcija -**a**, koja će isključiti sve particije navedene u /etc/fstab fajlu. Prethodno će se svi delovi memorije koji se trenutno nalaze na *swap* particiji koja se isključuje biti prebačeni u RAM i ostale *swap* particije.

Fajl /etc/fstab

Fajl /etc/fstab sadrži informacije o uređajima (particije, mrežni diskovi, izmenljivi mediji), njihovim tačkama kačenja, tipovima fajl sistema i opcijama prilikom kačenja. Svaka linija koja nije komentar ili prazna linija, definiše zapis za jedan uređaj. Format linije je:

```
uredaj tačka_kačenja tip_fs opcije dump fsck
```

gde su polja:

- **uredaj** specijalni fajl (particija) ili oznaka mrežnog diska. Umesto specijalnog fajla moguće je staviti logičko ime particije (labelu) u obliku 'LABEL=logicko_ime', odnosno UUID¹ diska u obliku 'UUID=uuid_particije'.
- **tačka_kačenja** direktorijum koji predstavlja tačku kačenja za dati uređaj, ili *none* za one uređaje koji se ne uključuju u logički fajl sistem
- **tip_fs** tip fizičkog fajl sistema na uređaju

¹ UUID (*Universally Unique Identifier*) je standard za označavanje u softverskim konstrukcijama koji je standardizovala OSF (*Open Source Foundation*) kao deo DCE (*Distributed Computing Environment*). Identičan je GUID standardu koji koristi Microsoft.

- **opcije** lista opcija mount komande razdvojena zarezima, bez belina
- **dump** zastavica koja označava da li se dati uređaj može bekapovati komandom dump (1) ili ne (0)
- **fsck** redosled kojim se obavlja operacija oporavka sistema fsck u slučaju da radi u paralelnom modu. Naime, komanda fsck može istovremeno da čekira više particija ukoliko su one na različitim diskovima. Za eksterne tipove fajl sistema se u ovo polje upisuje 0.

Primer jednog /etc/fstab fajla:

/dev/sdc5	/	ext4	noatime,acl,user_xattr	1	1
/dev/sdc2	/boot	ext4	noatime,acl,user_xattr	1	2
LABEL=home	/home	ext4	acl,user_xattr	1	2
UUID=f0543f31-ae31-41d3-b01b-dfe2f4c7e287		swap	swap		
pri=42		0	0		
devpts	/dev/pts	devpts	mode=0620,gid=5	0	0
proc	/proc	proc	defaults	0	0
sysfs	/sys	sysfs	noauto	0	0

Kao što se iz primera vidi, specijalni, tzv. pseudo-fajl sistemi nemaju asociran uređaj i u prvom polju imaju oznaku pseudo-fajl sistema.

Provera ispravnosti fajl sistema i ispravljanje grešaka

Iako se Linuxovi sopstveni i pozajmljeni fajl sistemi smatraju veoma robustnim, ponekad se može doći u situaciju da je potrebno proveriti ispravnost nekog fajl sistema i, ako postoje greške u njemu, pokušati ispraviti te greške i fajl sistem dovesti u ispravno i konzistentno stanje. U stvarnosti, da bi mogao da održi reputaciju stabilnog i robustnog sistema, Linux prilikom svakog kačenja nekog fajl sistema u stablo direktorijuma proverava da li je dati fajl sistem ispravan i da li treba uraditi dodatnu proveru ispravnosti i ispravljanje grešaka. Kao glavni indikator da li je neki fajl sistem ispravan služi jedna zastavica (*flag*) koja se naziva 'prljavi bit' (*dirty-bit*) i koja se postavlja svaki put kada se fajl sistem zakači u stablo, a briše svaki put kada se fajl sistem otkačinje od stabla. Otkačinjanje fajl sistema iz stabla označava da su svi podaci i metapodaci smešteni na uređaj na ispravan način i da su podaci na otkačenom uređaju konzistentni. Prilikom ponovnog kačenja datog fajl sistema, ukoliko je prljavi bit i dalje postavljen, to znači da fajl sistem nije regularno otkačen prethodni put (pad sistema i sl.) i da dati fajl sistem možda sadrži neispravnu strukturu i nekonzistentan sistem. U tom slučaju treba pokrenuti odgovarajuću komandu za proveru sistema i oklanjanje grešaka. Kod nekih fajl sistema kao što su ext2 i ext3, sistem će periodično vršiti forsiranu proveru ispravnosti sistema čak i ako je prljavi bit obrisan (obično posle nekog zadatog broja kačenja ili posle nekog zadatog vremenskog perioda od poslednjeg kačenja sistema).

Komanda za proveru fajl sistema je **fsck** (*file system check*). Ova komanda ima sledeći format:

```
# fsck [opcije] [uredaj...] [-- fs_opcije]
```

gde su najčešće korištene opcije:

- A** paralelna provera svih fajl sistema koji imaju *fsck* zastavicu različitu od 0 u */etc/fstab* fajlu.
- a** automatsko ispravljanje grešaka na fajl sistemu bez manuelne potvrde
- r** interaktivno ispravljanje grešaka. Kada **fsck** detektuje grešku korisnik mora da odobri akciju ispravljanja date greške ili da odbije ispravljanje.
- s** ako je zadato više fajl sistema, oni će biti proveravani jedan za drugim, a ne paralelno, što je standardni način rada (paralelna provera se obavlja kada su dati fajl sistemi na različitim fizičkim uređajima).
- t tip_fajl_sistema** ručno se specificira tip fajl sistema koji se proverava
- y** kod nekih tipova fajl sistema podrazumevano ponašanje je interaktivna provera, kao da je zadata opcija -r. U tom slučaju opcija -y će automatski odgovarati potvrđeno (bez interakcije korisnika) na sve upite za ispravljanje uočenih grešaka.

uredaj može biti naziv uređaja ili direktorijum na koji se dati uredaj kači (ako je naveden u */etc/fstab* fajlu).

fs_opcije su opcije koje su specifične za pojedine tipove fajl sistema.

Komanda **fsck** sama ne izvršava proveru ispravnosti i ispravljanje grešaka, već je samo front-end za **fsck** komande specifične za dati tip fajl sistema. Ove komande se obično nalaze u */sbin* direktorijumu i imaju imena oblika **fsck.tip_fajl_sistema**.

Povratna vrednost komande **fsck** označava rezultat provere:

0 fajl sistem ne sadrži greške

1 fajl sistem je sadržao greške koje su opravljene

2 komanda je opravila fajl sistem ali je potrebo restartovati računar

4 neke greške nisu ispravljene

8 došlo je do greške prilikom izvršavanja komande

16 greška u sintaksi komande

32 korisnik je prekinuo izvršavanje komande

128 greška u deljenim bibliotekama

Ukoliko je komanda uspešno opravila fajl sistem, prljavi bit će biti obrisan i dati fajl sistem proglašen ispravnim.

Property of Admin Training Center

Upravljanje logičkim volumenima

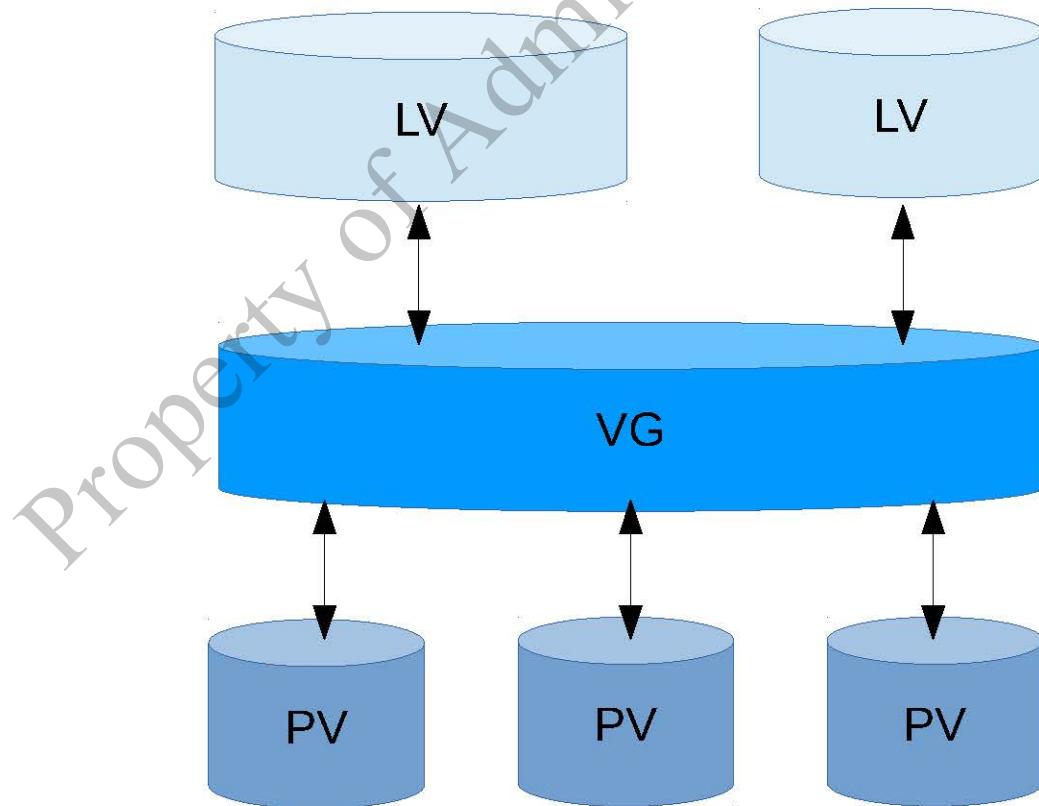
Uvod

Logički volumeni su virtuelne particije koje omogućavaju različite manipulacije prostorom za smeštanje podataka. Na Linuxu, logički volumeni su implementirani preko podsistema LVM2 (*Logical Volume Manager*)

Arhitektura LVM

Logička struktura LVM se sastoji od tri 'sloja':

- PV (*physical volume* – fizički volumen) – blok uređaji kao što su particije diska ili čitavi diskovi,
- VG (*volume group* – volumenska grupa) – struktura koja objedinjava jedan ili više PV-ova,
- LV (*logical volume* – logički volumen) – virtualni blok uređaj koji predstavlja particiju a nalazi se unutar neke VG.



PV – fizički volumeni

PV mogu biti disk particije ili celi diskovi. Ukoliko želimo da neku particiju pretvorimo u PV, potrebno je da prethodno promenimo njen tip u '8e' (Linux LVM). Ukoliko želimo ceo disk da pretvorimo u PV moramo ukloniti particionu tabelu ukoliko ona postoji. To možemo uraditi sledećom komandom:

```
# dd if=/dev/zero of=blok-uredaj bs=512 count=1
```

Kreiranje fizičkih volumena

PV je potrebno kreirati, odnosno inicijalizovati pre nego što ih ubacimo u VG. Ukoliko niste sigurni koji diskovi i particije su pripremljene za konvertovanje u PV, to možete videti komandom 'lvmdiskscan'.

Konvertovanje diska ili particije u PV se vrši komandom 'pvcreate':

```
# pvcreate blok-uredaj ...
```

Donji primer kreira PV na disku /dev/sdd i na particijama /dev/sde1 i /dev/sdf1.

```
# pvcreate /dev/sdd /dev/sde1 /dev/sdf1
```

Listanje fizičkih volumena

Komandom 'pvdisplay' možemo pogledati informacije o pojedinom fizičkom volumenu (ako je kao argument zadat blok-uredaj koji je konvertovan u PV) ili za sve fizičke volumene na sistemu (ako je komanda zadata bez argumenata):

```
# pvdisplay [blok-uredaj] ...
```

Primer ispisa:

```
# pvdisplay
--- Physical volume ---
PV Name          /dev/sdc1
VG Name          new_vg
PV Size          17.14 GB / not usable 3.40 MB
Allocatable      yes
PE Size (KByte) 4096
Total PE         4388
Free PE          4375
Allocated PE     13
PV UUID          Joqlch-yWSj-kuEn-IdwM-01S9-X08M-mcpsVe
```

Komanda 'pvscan' skenira sve podržane LVM blok-uredaje na sistemu tražeći fizičke volumene i daje skraćeni ispis njihovih parametara:

```
# pvscan
PV /dev/sdb2   VG vg0    lvm2 [964.00 MB / 0   free]
PV /dev/sdc1   VG vg0    lvm2 [964.00 MB / 428.00 MB free]
PV /dev/sdc2   VG vg0    lvm2 [964.84 MB]
Total: 3 [2.83 GB] / in use: 2 [1.88 GB] / in no VG: 1
[964.84 MB]
```

Promena veličine fizičkog volumena

Ukoliko iz nekog razloga želite da promenite veličinu nekog fizičkog volumena, to možete uraditi iz dva koraka:

- promena veličine fizičkog volumena unutar LVM
- promena veličine particije

Redosled koraka zavisi od toga da li želimo da povećamo ili smanjimo veličinu fizičkog volumena. Ukoliko želimo da povećamo veličinu PV, onda je redosled koraka sledeći:

1. povećanje veličine particije (`fdisk` ili `parted`)
2. povećanje veličine fizičkog volumena unutar LVM

Ukoliko želimo da smanjimo veličinu PV, onda je redosled koraka:

1. smanjenje veličine fizičkog volumena unutar LVM
2. smanjenje veličine particije (`fdisk` ili `parted`)

Promena veličine fizičkog volumena unutar LVM se može odraditi i dok je taj volumen aktivan u nekoj VG. Drugim rečima, nije potrebno otkačiti fajl sisteme niti restartovati server da bi se ova operacija izvela. Naravno, kod smanjenja veličine PV koji je aktivan u nekoj volumenskoj grupi moramo prethodno biti sigurni da je unutar VG nezauzeta količina prostora za koju smanjujemo PV.

Menjanje veličine PV se vrši komandom:

```
# pvresize [--setphysicalvolumesize veličina] PV-blok-uređaj ...
```

Ukoliko se opcija --setphysicalvolumesize izostavi, nova veličina će biti celokupna veličina particije. Ovo praktično znači da će se ova opcija koristiti samo kod smanjenja veličine particije.

Uklanjanje PV

Ako više ne želimo da neka particija ili disk bude PV, možemo ukloniti metapodatke sa njе komandom 'pvremove'. Prethodno moramo izbaciti dati PV iz volumenske grupe, što se radi komandom 'vgreduce'.

Oblik komande 'pvremove' je:

```
# pv remove PV-blok-uređaj ...
```

VG – volumenske grupe

Volumenske grupe okupljaju jedan ili više fizičkih volumena i njihov ukupan kapacitet dele na logičke volumene.

Kreiranje VG

Kreiranje VG se obavlja komandom 'vgcreate' koja kreira novu VG od zadatih PV:

```
# vgcreate [opcije] ime PV-blok-uređaj ...
```

U primeru:

```
# vgcreate vg1 /dev/sdd1 /dev/sde1
```

biće kreirana nova VG pod imenom 'vg1' a koja se sastoji iz dva PV: /dev/sdd1 i /dev/sde1.

Novokreirana volumenska grupa će svoj ukupan prostor za podatke podeliti u kontinualne blokove (ekstente) podrazumevane veličine od 4MB. Pošto logički volumeni unutar neke VG moraju imati ceo broj ovih ekstenata, to znači da je se povećanje ili smanjivanje veličine LV može izvesti samo u koracima veličine jednog ekstenta. Ukoliko želite da vaša VG ima neku drugu veličinu ekstenta, nju zadajte sa opcijom '-s' kod kreiranja VG.

Maksimalna veličina VG je 8EB na 64-bitnim platformama.

Dodavanje PV u VG

Proširenje VG se može izvršiti 'u letu' (dok je VG aktivna) komandom:

```
# vgextend ime-VG PV-blok-uredaj ...
```

Izbacivanje PV iz VG

Neki fizički volumen možemo ukloniti iz volumenske grupe samo ako on ne sadrži podatke. Komanda za uklanjanje je:

```
# vgreduce ime-VG PV-blok-uredaj ...
```

Listanje volumenskih grupa

Komandom:

```
# vgdisplay [ime-VG ...]
```

možemo prikazati parametre date VG. Na primer:

```
# vgdisplay new_vg
--- Volume group ---
VG Name           new_vg
System ID
Format           lvm2
Metadata Areas   3
Metadata Sequence No 11
VG Access        read/write
VG Status         resizable
MAX LV
Cur LV           1
Open LV          0
Max PV
Cur PV           3
Act PV           3
VG Size          51.42 GB
PE Size          4.00 MB
Total PE         13164
Alloc PE / Size  13 / 52.00 MB
Free  PE / Size  13151 / 51.37 GB
VG UUID          jxQJ0a-Zkk0-0pM0-0118-nlw0-wwqd-fD5D32
```

Skeniranje VG i rekreiranje keš fajlova

Komanda 'vgscan' skenira sve dostupne blok-uređaje tražeći fizičke volumene i volumenske grupe na njima. Ova komanda generiše LVM keš fajl (/etc/lvm/cache/.cache) koji sadrži listing svih VG na sistemu. Ova komanda se startuje automatski pri startovanju servera i posle zadate 'vgcreate' komande. Ukoliko ste dodali diskove koji su sadržali neku VG (npr. sa nekog drugog sistema), potrebno je manuelno

zadati ovu komandu kako bi LVM podsistem uključio tu novu grupu u spisak trenutno poznatih VG.

Primer:

```
# vgscan
Reading all physical volumes. This may take a while...
Found volume group "new_vg" using metadata type lvm2
Found volume group "officevg" using metadata type lvm2
```

Aktiviranje i deaktiviranje VG

Da bi VG mogla da se koristi na sistemu potrebno je da bude proglašena aktivnom, tj. aktivirana. Ovo se dešava automatski kod svakog kreiranja grupe, ali je ponekad potrebno grupu manuelno aktivirati ili deaktivirati. To se radi komandom:

```
# vgchange -a stanje [ime-VG ...]
```

gde je stanje:

y – grupa treba biti aktivirana;

n – grupa treba biti deaktivirana.

Ukoliko je grupa deaktivirana kernel više ne vidi logičke volumene koji pripadaju toj VG.

Uklanjanje VG

Da bismo uklonili tj. obrisali neku VG prvo moramo izbrisati sve LV koji su postojali unutar nje. Komanda za uklanjanje VG je:

```
# vgremove ime-VG
```

Preimenovanje VG

Volumensku grupu možemo preimenovati komandom:

```
# vgrename staro-ime-VG novo-ime-VG
```

Napomena: Vodite računa da će posle izmene imena VG biti promenjene sve putanje do njenih LV!

LV – logički volumeni

Logički volumeni su virtualne particije koje se nalaze unutar neke VG. Sa logičkim volumenima postupamo kao i sa bilo kojim drugim disk-particijama.

Kreiranje LV

Logički volumen možete kreirati komandom:

```
# lvcreate [opcije] ime-VG
```

gde su opcije:

- n **ime-LV** – ime logičkog volumena koji kreiramo. Bez ove opcije, volumen će biti imenovan kao 'lvol#', gde je '#' redni broj volumena. Novokreirani LV će biti dostupan preko blok uređaja '/dev/<ime-VG>/<ime-LV>'.
- L **apsolutna-veličina** – zadata veličina za novi LV. Isti će biti kreiran ako unutar VG ima dovoljno prostora. Ako se ova opcija izostavi biće pokušano kreiranje LV koji zauzima ceo prostor VG. Parametar 'apsolutna-veličina' može se zadati kao broj, u kom slučaju se to računa kao broj MB, ili kao 'brojG' (broj gigabajta) itd.
- l **veličina** – veličina se može zadati i u broju ekstenata, kada se koristi ova opcija, ili procentualno, kao 'br_proc%OSNOVA', gde je 'br_proc' broj procenata u odnosu na OSNOVU, koja može biti 'VG' (cela volumenska grupa) ili 'FREE' (preostali slobodan prostor unutar VG).

Na primer:

```
# lvcreate -L 10G vg1
```

kreira LV pod imenom /dev/vg1/lvol1 veličine 10GB unutar volumenske grupe vg1.

```
# lvcreate -L 1500 -n testlv testvg
```

kreira LV /dev/testvg/testlv unutar testvg VG veličine 1500MB.

```
# lvcreate -l 60%VG -n mylv testvg
```

kreira LV pod imenom /dev/testvg/mylv čija je veličina 60% od ukupne veličine VG testvg.

```
# lvcreate -l 100%FREE -n yourlv testvg
```

kreira LV pod imenom /dev/testvg/yourlv koji će zauzeti sav slobodan prostor unutar VG testvg.

```
# vgdisplay testvg | grep "Total PE"  
Total PE 10230  
# lvcreate -l 10230 testvg -n mylv
```

kreira LV pod imenom /dev/testvg/mylv veličine 10230 ekstenata (komandom 'vgdisplay' je utvrđeno da je tolika veličina VG).

Kreiranje snapšot volumena

Snepšot volumen je kopija zadatog volumena u trenutku kreiranja snepšota. Od trenutka kreiranja snepšot i original od kojeg je snepšot volumen nastao se mogu razlikovati. Snepšotove najčešće koristimo da bismo bekapovali neke podatke koji se tokom vremena često menjaju a moraju biti konzistentni unutar bekapa. Dobar primer za to je npr. bekapovanje velike baze podataka. Regularan bekap zahteva da tokom njegovog trajanja ne bude izmena u podacima unutar baze. Ako takav bekap traje neko duže vreme, baza neće biti dostupna za promene što može biti neprihvatljivo. U tom slučaju, najbolje rešenje je da se bazni fajlovi čuvaju na LV particiji. Onda je moguće za trenutak zaustaviti izmene u bazi (time su podaci unutar baze u konzistentnom stanju), zatim napraviti snapšot LV particije koja sadrži bazne fajlove i odmah nakon kreiranja snapšota (što se izvršava skoro trenutno), omogućiti ponovnu izmenu podataka u bazi. Snapšot LV treba zakačiti u fajl sistem na neki direktorijum i onda odatle praviti bekap jer fajlovi unutar snepšota sadrže konzistentne podatke. Posle završenog bekapa snepšot LV možemo otkačiti i ukloniti iz VG.

Kada kreiramo novi LV automatski se alocira maksimalna zadata veličina unutar VG. Kod snepšotova se zadaje veličina koja će sadržati samo razlike u odnosu na originalni LV. Drugim rečima, LVM zapisuje samo 'razlike' u odnosu na originalni LV koje nastaju od trenutka kada je snepšot kreiran. Iz tog razloga, ukoliko planiramo da koristimo snepšotove unutar neke VG, potrebno je uvek ostaviti nešto slobodnog prostora koji neće biti alociran logičkim volumenima.

Snepšot LV se takođe kreira komandom 'lvcreate':

```
# lvcreate -L veličina-snepšota --snapshot \
-n ime-snepšota ime-LV
```

U primeru:

```
# lvcreate --size 100M --snapshot --name snap \
/dev/vg00/lvol1
```

kreiran je snepšot imena /dev/vg00/snap koji je kopija /dev/vg00/lvol1 LV-a i kojem je alocirano 100M prostora.

Komandom:

```
# lvdisplay /dev/vg00/lvol1
--- Logical volume ---
LV Name          /dev/vg00/lvol1
VG Name          vg00
LV UUID          LBy1Tz-sr23-0jsI-LT03-nHLC-y8XW-
EhC178
LV Write Access  read/write
LV snapshot status source of
                   /dev/vg00/snap [active]
LV Status        available
# open           0
LV Size          52.00 MB
Current LE       13
Segments         1
Allocation       inherit
Read ahead sectors 0
Block device    253:2
```

Komanda 'lvs' nam može prikazati koliko alociranog prostora snepšota je zauzeto:

# lvs	LV	VG	Attr	LSize	Origin	Snap%	Move	Log	Copy%
	lvol1	vg00	owi-a-	52.00G					
	snap	vg00	swi-a-	100.00M	lvol1	0.20			

Napomena: ukoliko se nakupi toliko izmena u odnosu na originalni LV da snepšot volumen postane pun, isti će biti proglašen nevalidnim. Originalni LV neće biti afektiran ovom promenom, pošto je snepšot taj koji čuva izmene u odnosu na originalne ekstentove. Iz tog razloga uvek je dobra praksa periodično pozivati komandu 'lvs' da bi se pratilo zauzeće prostora alociranog snepšotu.

Promena veličine LV

Smanjivanje veličine LV se obavlja 'lvreduce' komandom. Ako LV čiju veličinu želimo smanjiti sadrži fajl sistem, isti prvo moramo smanjiti pre smanjivanja veličine LV. LV mora minimalno biti onoliko velika koliki je fajl-sistem.

Komanda:

```
# lvreduce -l -3 vg00/lvol1
```

će smanjiti LV vol1 na VG vg00 za tri ekstenta.

Povećavanje veličine LV se obavlja komandom 'lvextend', gde se može zadati nova veličina LV ili količina prostora za koji povećavamo postojeći LV. Naravno, preduslov je da VG u kojoj se nalazi dati LV ima dovoljno nealociranog prostora.

Komanda:

```
# lvextend -L12G /dev/myvg/homevol
lvextend -- extending logical volume "/dev/myvg/homevol" to 12
GB
lvextend -- doing automatic backup of volume group "myvg"
lvextend -- logical volume "/dev/myvg/homevol" successfully
extended
```

povećava veličinu LV homevol na 12GB.

Komanda:

```
# lvextend -L+1G /dev/myvg/homevol
lvextend -- extending logical volume "/dev/myvg/homevol" to 13
GB
lvextend -- doing automatic backup of volume group "myvg"
lvextend -- logical volume "/dev/myvg/homevol" successfully
extended
```

povećava LV homevol za 1GB.

Komanda:

```
# lvextend -l +100%FREE /dev/myvg/testlv
Extending logical volume testlv to 68.59 GB
Logical volume testlv successfully resized
```

povećava veličinu LV testlv za kompletan slobodan prostor na VG myvg.

Posle povećavanja veličine LV potrebno je i povećati veličinu fajl-sistema koji isti sadrži.

Preimenovanje LV

Logički volumen možemo preimenovati komandom:

```
# lvrename staro-ime-LV novo-ime-LV
```

gde su 'staro-ime-LV' i 'novo-ime-LV' putanje do blok-uredaja. Ukoliko ne želimo da navodimo pune putanje, možemo kao prvi argument navesti ime VG u kojoj se nalazi LV koji želimo da preimenujemo:

```
# lvrename vg02 lvold lvnew
```

Aktiviranje i deaktiviranje LV

LV je automatski aktivirana po kreiranju, no ponekad je potrebno deaktivirati i aktivirati LV manuelno komandom:

```
# lvchange -a stanje ime-LV ...
```

gde je stanje:

y – LV treba biti aktiviran;

n – LV treba biti deaktiviran.

Uklanjanje LV

Uklanjanje LV se obavlja komandom:

```
# lvremove ime-LV
```

Ukoliko nismo prethodno deaktivirali LV, komanda 'lvremove' će tražiti potvrdu da stvarno želimo da uklonimo aktivni LV:

```
# lvremove /dev/testvg/testlv
Do you really want to remove active logical volume "testlv"?
[y/n]: y
Logical volume "testlv" successfully removed
```

Listanje LV

Komanda 'lvdisplay' će prikazati podatke o zadatom LV:

```
# lvdisplay -v /dev/vg00/lvol2
```

Komanda 'lvscan' će skenirati sve LVM particije i tražiti volumenske grupe i logičke volumene koji su kreirani unutar njih:

```
# lvscan
ACTIVE          '/dev/vg0/gfslv' [1.46 GB] inherit
```

Property of Admin Training Center

Praćenje sistemskih resursa

Praćenje stanja sistema

Često je potrebno utvrditi trenutno zauzeće resursa kao što su CPU, memorija i prostor na disku, kako bismo utvrdili da li naš sistem ima dovoljno resursa da izdrži zahtevano opterećenje.

Praćenje CPU

Za praćenje opterećenja CPU najčešće koristimo komande 'ps' i 'top'.

Komanda 'ps'

Komanda 'ps' omogućava prikaz informacija o procesima koji se trenutno izvršavaju. Komanda generiše statičku listu koja ukazuje na stanje kakvo je bilo u trenutku izvršavanja same komande. Da biste videli sve procese koji se trenutno izvršavaju, zadajte komandu:

```
# ps ax
```

Komanda prikazuje stanje tabelarno, svaki proces u zasebnom redu. Za svaki proces se prikazuju redom:

- **PID** – ID procesa
- **TTY** – kontrolni terminal (terminal na kojem se ispisuju standardni izlaz i standardni izlaz za greške i sa kojeg se učitava standardni ulaz – znak '?' označava da je proces otkačen od svog kontrolnog terminala)
- **STAT** – trenutno stanje u kojem se nalazi proces (D – proces u stanju spavanja koje se ne može prekinuti (obično znači da je proces blokirana od strane IO), R – proces se izvršava ili je u redu za izvršavanje, S – proces je u stanju spavanja koje se može prekinuti (obično čeka da se završi neki događaj), T – proces je zaustavljen eksterno ili se prati, X – proces je mrtav (ne bi trebalo da se pojavljuje na ispravno radećem sistemu), Z – proces je 'zombi')
- **TIME** – kumulativno vreme izvršavanja na procesoru
- **COMMAND** – komanda koja se izvršava, sa argumentima

Komandom:

```
# ps aux
```

dobićemo još i kolone:

- **USER** - efektivni vlasnik procesa
- **%CPU** - procenat korišćenja CPU
- **%MEM** – procenat korišćenja memorije
- **VSZ** - veličinu virtuelne memorije procesa u KB
- **RSS** – veličina memorije u KB koja uvek mora biti u RAM
- **START** – datum i vreme startovanja procesa

Komanda 'top'

Komanda 'top' prikazuje skoro iste parametre procesa kao i komanda 'ps' ali to čini u realnom vremenu, sa određenom frekvencijom osvežavanja ispisa (podrazumevano na svake 3 sekunde). Komanda takođe u zaglavlju ispisuje zbirne vrednosti nekih parametara, a omogućava i interaktivno sortiranje ispisa i ubijanje procesa.

Komanda:

```
# top
```

za svaki prikazani proces ispisuje:

- **PID** – ID procesa
- **USER** – efektivni vlasnik procesa
- **PR** – trenutni prioritet procesa
- **NI** – 'nice' vrednost procesa
- **VIRT** – količina virtuelne memorije koju proces koristi
- **RES** – količina memorije koja mora stalno biti u RAM-u
- **SHR** – količina deljene memorije procesa
- **%CPU** – procenat korišćenja CPU
- **%MEM** – procenat korišćenja memorije
- **TIME+** - kumulativno vreme izvršavanja
- **COMMAND** – komanda koja se izvršava

Pored ovoga, u zaglavlju ispisa se nalaze sledeći parametri:

- trenutno vreme
- koliko je vremena prošlo od poslednjeg startovanja sistema

- prosečno opterećenje sistema u poslednjih 1 minut, 5 minuta i 15 minuta
- broj procesa: ukupno, u stanju izvršavanja, u stanju spavanja, zaustavljenih i zombija
- procentualno zauzeće CPU:
 - us – u korisničkom prostoru sa standardnim prioritetom
 - sy – u kernelskom prostoru
 - ni – u korisničkom prostoru sa promenjenim prioritetom
 - wa – vreme potrošeno na čekanje nekog I/O događaja
 - hi – vreme potrošeno na servisiranje hardverskih interapta
 - si – vreme potrošeno na servisiranje softverskih interapta
 - st – vreme koje je hipervizor 'ukrao' (ukoliko je sistem u stvari virtuelna mašina)
- zauzeće memorije: ukupno, zauzeta, slobodna, u dinamičkim baferima
- zauzeće svop memorije: ukupno, zauzeto, slobodno, keširano

Interaktivne komande programa 'top' su:

- Enter, Space – momentalno osvežavanje ispisa
- h, ? - ispis ekrana za pomoć
- k – ubijanje procesa. Program će pitati za PID i signal koji mu se prosleđuje
- n – menja broj prikazanih procesa. Program će pitati za novi broj
- u – sortira spisak po vlasniku procesa
- M – sortira spisak po zauzeću memorije (od najvećeg ka najmanjem)
- P – sortira spisak po zauzeću CPU (od najvećeg ka najmanjem)
- q – izlaz iz programa

Praćenje memorije

Komanda 'free'

Komanda:

```
# free
```

ispisuje trenutno zauzeće memorije. Ovo zauzeće je prikazano tabelarno, gde su redovi prikaz: RAM memorije (Mem), keš bafera (-/+ buffers/cache) i svop memorije (Swap), a kolone su: prikaz ukupne memorije (total), prikaz iskorišćene memorije (used), prikaz slobodne memorije (free), prikaz deljene memorije (shared), prikaz kernel bafera (buffers) i prikaz keševa (cached). Sve vrednosti su prikazane u KB.

Praćenje podsistema diskova i zauzeća prostora na njima

Komanda 'blkid'

Komanda:

```
# blkid
```

prikazuje informacije o dostupnim blok-uređajima, podrazumevano njihov UUID i tip fajl-sistema:

```
# blkid
/dev/sda1: UUID="1BAD-00DF" TYPE="vfat"
/dev/sda2: UUID="f0543f31-ae31-41d3-b01b-dfe2f4c7e287"
TYPE="ext2"
/dev/sda3: UUID="PbmArb-7gYa-0bKs-2AYB-q8NL-s0gf-fj5gzS"
TYPE="LVM2_member"
/dev/mapper/ubuntu-root: UUID="495499c2-1a00-47ee-bd89-
92a43a663c12" TYPE="ext4"
/dev/mapper/ubuntu-swap_1: UUID="a82fee77-ba1c-43bc-8d3b-
0adfde4165aa" TYPE="swap"
```

Ovom komandom možemo dobiti i više podataka o pojedinačnom blok-uređaju, npr:

```
# blkid -ip /dev/sda1
UUID=1BAD-00DF
VERSION=FAT32
TYPE=vfat
USAGE=filesystem
MINIMUM_IO_SIZE=512
PHYSICAL_SECTOR_SIZE=512
LOGICAL_SECTOR_SIZE=512
PART_ENTRY_SCHEME=gpt
PART_ENTRY_UUID=81143dfe-f8ef-4f8d-8bea-cb2ab83d413f
PART_ENTRY_TYPE=c12a7328-f81f-11d2-ba4b-00a0c93ec93b
PART_ENTRY_NUMBER=1
PART_ENTRY_OFFSET=2048
PART_ENTRY_SIZE=389120
PART_ENTRY_DISK=8:0
```

Komanda 'df'

Komanda 'df' prikazuje detaljan izveštaj o zauzeću prostora na zakačenim blok-uređajima. Oblik ove komande je:

```
# df [opcije]
```

gde su najčešće opcije:

- h** – ispisuje veličinu u razumljivijim jedinicama (mega, giga, tera) u odnosu na podrazumevanu vrednost (KB za prostor, jedinična količina za i-nodove)
- i** – prikazuje iskorišćenost i-nodova umesto prostora na disku

Primeri:

```

# df
Filesystem      1K-blocks      Used  Available Use% Mounted on
/dev/vda1        20642364    4323032   15270760  23% /
tmpfs            1959488       0     1959488   0% /dev/shm
/dev/vdc1        412849208   374576628  17301068  96% /srv
# df -h
Filesystem      Size   Used  Avail Use% Mounted on
/dev/vda1        20G    4.2G   15G  23% /
tmpfs            1.9G    0     1.9G  0% /dev/shm
/dev/vdc1        394G   358G   17G  96% /srv
# df -i
Filesystem      Inodes  IUsed  IFree IUse% Mounted on
/dev/vda1        1310720  50249  1260471  4% /
tmpfs            489872    1    489871  1% /dev/shm
/dev/vdc1        26214400 905527 25308873  4% /srv
# df -ih
Filesystem      Inodes  IUsed  IFree IUse% Mounted on
/dev/vda1        1.3M    50K    1.3M  4% /
tmpfs            479K    1     479K  1% /dev/shm
/dev/vdc1        25M    885K   25M  4% /srv

```

Komanda 'du'

Komanda 'du' prikazuje koliko koji direktorijum ili fajl zauzimaju prostora na fajl-sistemu. Bez opcija i argumenata, 'du' daje prikaz zauzeća prostora u tekućem direktorijumu u kilobajtima:

```

# du
14972  ./Downloads
4       ./gnome2
4       ./mozilla/extensions
4       ./mozilla/plugins
12      ./mozilla
15004  .

```

Ukoliko želimo da ispis veličine zauzeća prostora bude u čitljivijem obliku, koristićemo opciju '-h':

```

# du -h
5M     ./Downloads
4.0K   ./gnome2
4.0K   ./mozilla/extensions
4.0K   ./mozilla/plugins
12K   ./mozilla
15M   .

```

Komanda može dati i zbirni prikaz zauzeća za zadati direktorijum, odnosno tekući direktorijum ukoliko je komanda pozvana bez argumenata. Zbirni prikaz se dobija navođenjem opcije '-s':

```
# du -sh  
15M .
```

Napomena

U slučajevima kada neka aplikacija (koja se obično dugo izvršava ili radi kao dimon, npr. rsyslog) otvorit će pri startovanju neki fajl, koji zatim mi obrišemo dok data aplikacija i dalje radi, 'du' prilikom računanja prostora neće uračunati i taj obrisani fajl u prikazu zauzeća (jer ne postoji ni u jednom direktorijumu). Međutim, taj fajl, iako obrisan, i dalje zauzima prostor na disku, sve dok ga aplikacija drži otvorenim i u takvom slučaju može doći do različitog prikaza zauzeća koje za dati fajl-sistem vraćaju komande 'df' i 'du'. U takvim slučajevima 'df' je pouzdaniji izvor, ali sama razlika znači da postoje obrisani fajlovi koji su još uvek otvoreni i još uvek alociraju prostor na disku.

Praćenje ostalog hardvera

Komanda 'lspci'

Komanda 'lspci' prikazuje informacije o PCI sabirnicama i uređajima zakačenim na nju:

```
# lspci  
00:00.0 Host bridge: Intel Corporation 82X38/X48 Express DRAM Controller  
00:01.0 PCI bridge: Intel Corporation 82X38/X48 Express Host-Primary PCI Express Bridge  
00:1a.0 USB Controller: Intel Corporation 82801I (ICH9 Family)  
USB UHCI Controller #4 (rev 02)  
00:1a.1 USB Controller: Intel Corporation 82801I (ICH9 Family)  
USB UHCI Controller #5 (rev 02)  
00:1a.2 USB Controller: Intel Corporation 82801I (ICH9 Family)  
USB UHCI Controller #6 (rev 02)  
[ispis odsečen]
```

Detaljnije informacije se dobijaju opcijama '-v' ili '-vv':

```
# lspci -v
[ispis odsečen]
01:00.0 VGA compatible controller: nVidia Corporation G84
[Quadro FX 370] (rev a1) (prog-if 00 [VGA controller])
    Subsystem: nVidia Corporation Device 0491
    Physical Slot: 2
    Flags: bus master, fast devsel, latency 0, IRQ 16
    Memory at f2000000 (32-bit, non-prefetchable)
[size=16M]
    Memory at e0000000 (64-bit, prefetchable) [size=256M]
    Memory at f0000000 (64-bit, non-prefetchable)
[size=32M]
    I/O ports at 1100 [size=128]
    Expansion ROM at <unassigned> [disabled]
    Capabilities: <access denied>
    Kernel driver in use: nouveau
    Kernel modules: nouveau, nvidiafb
[ispis odsečen]
```

Ova komanda će prikazati sve uređaje na PCI sabirnicama, bez obzira da li ih sam kernel prepoznaje ili ne. Ukoliko kernel, odnosno neki drajver zadužen za upravljanje tom klasom uređaja, ne prepoznaje dati uređaj, 'lspci' neće prikazati naziv samog uređaja već će umesto toga stajati 'unknown'. Ukoliko takav uređaj ne radi na vašem sistemu, moguće je da je problem do kernel drajvera koji ne prepoznaje taj uređaj.

Komanda 'lsusb'

Komandom 'lsusb' možemo dobiti informacije o USB sabirnicama i uređajima zakačenim na njih:

```
# lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
[ispis odsečen]
Bus 001 Device 002: ID 0bda:0151 Realtek Semiconductor Corp.
Mass Storage Device (Multicard Reader)
Bus 008 Device 002: ID 03f0:2c24 Hewlett-Packard Logitech M-
UAL-96 Mouse
Bus 008 Device 003: ID 04b3:3025 IBM Corp.
```

Slično kao i kod 'lspci' komande, opcija '-v' će dati detaljniji prikaz:

```

# lsusb -v
[ispis odsečen]
Bus 008 Device 002: ID 03f0:2c24 Hewlett-Packard Logitech M-
UAL-96 Mouse
Device Descriptor:
  bLength          18
  bDescriptorType   1
  bcdUSB         2.00
  bDeviceClass      0 (Defined at Interface level)
  bDeviceSubClass    0
  bDeviceProtocol    0
  bMaxPacketSize0     8
  idVendor        0x03f0 Hewlett-Packard
  idProduct        0x2c24 Logitech M-UAL-96 Mouse
  bcdDevice       31.00
  iManufacturer      1
  iProduct          2
  iSerial           0
  bNumConfigurations  1
Configuration Descriptor:
  bLength          9
  bDescriptorType   2
[ispis odsečen]
```

Komanda 'lscpu'

Komanda 'lscpu' prikazuje informacije o CPU-ovima:

```

# lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
CPU(s):                 4
On-line CPU(s) list:  0-3
Thread(s) per core:    1
Core(s) per socket:    4
Socket(s):              1
NUMA node(s):           1
Vendor ID:              GenuineIntel
CPU family:             6
Model:                  23
Stepping:                7
CPU MHz:                 1998.000
BogoMIPS:                4999.98
Virtualization:        VT-x
L1d cache:               32K
L1i cache:               32K
L2 cache:                 3072K
NUMA node0 CPU(s):      0-3
```

Property of Admin Training Center

Upravljanje softverskim paketima

RedHat Package Manager – RPM

RedHat Package Manager je najpoznatiji sistem za upravljanje paketima u Linux svetu i podrazumevani je takav sistem na većini najpoznatijih i najraširenijih Linux distribucija (RedHat/Fedora, SuSE, Mandriva itd.).

RPM paket

RPM paket je specifična vrsta arhive koja sadrži fajlove koji će biti instalirani kao i dodatne informacije u paketu. RPM paket se instalira isključivo 'rpm' komandom i arhivu nije moguće raspakovati nekom drugom komandom bez modifikacije samog RPM paketa. Nomenklatura paketa obično prati sledeći format:

naziv_softvera-varijanta-verzija-izdanje.arhitektura.rpm

gde su:

- *naziv_softvera* je naziv softverskog paketa spakovanog u RPM arhivu
- *varijanta* je opcionalni dodatak imenu paketa koja detaljnije određuje namenu fajlova u RPM arhivi. Standardne varijante su:
 - *devel* – razvojni deo paketa
 - *common* – zajednički deo paketa (obično i za client i za server deo)
 - *client* – deo paketa koji sadrži samo klijentske programe
 - *server* – deo paketa koji sadrži samo serverske programe
 - *docs* – dokumentacija za softverski paket
- *verzija* je izvorna verzija softverskog paketa, prema nomenklaturi proizvodača softvera
- *revizija* je revizija izvornog softverskog paketa od strane onog ko je pravio RPM paket
- arhitektura je oznaka arhitekture za koju je paket namenjen. Standardne oznake su:
 - *i386* – RPM paket sadrži izvršne fajlove koji se mogu izvršavati na Intel 386 kompatibilnim procesorima.

- *i586* – RPM paket sadrži izvršne fajlove koji se mogu izvršavati na Intel Pentium kompatibilnim procesorima.
- *i686* – RPM paket sadrži izvršne fajlove koji se mogu izvršavati na Intel PentiumPro kompatibilnim procesorima.
- *athlon* – RPM paket sadrži izvršne fajlove koji se mogu izvršavati samo na AMD Athlon 32-bitnim procesorima.
- *ia64* – RPM paket sadrži izvršne fajlove koji se mogu izvršavati samo na Intel Itanium i Itanium2 procesorima.
- *ppc64* – RPM paket sadrži izvršne fajlove koji se mogu izvršavati samo na PowerPC 64-bitnim procesorima.
- *noarch* – RPM paket ne sadrži izvršne fajlove i može se koristiti na bilo kojoj platformi
- *src* – RPM paket sadrži izvorni kod softverskog paketa sa instrukcijama za generisanje jednog ili više RPM paketa sa izvršnim fajlovima.

RPM sistem sve podatke vezane za instalaciju nekog paketa čuva u svojoj bazi podataka. Informacije koje se čuvaju obuhvataju informacije o paketima, spisak fajlova koji čini jedan paket, heš vrednosti za svaki instalirani fajl u okviru paketa (da bi se utvrdilo da li je fajl menjan nakon instalacije), vreme instalacije i sl.

Razrešavanje problema međuzavisnosti paketa

RPM pokušava da razreši problem međuzavisnosti tako što omogućava autoru RPM paketa da eksplicitno navede spisak fajlova ili paketa koji su potrebni da bi se dati RPM paket instalirao. Prilikom instalacije, RPM softver pretražuje sopstvenu bazu podataka da bi našao informacije da li su potrebni fajlovi/paketi instalirani, i ako nisu daje upozorenje i odbija instalaciju paketa. Na žalost, ako je potrebni fajl instaliran na datom Linux sistemu van RPM softvera, 'rpm' komanda neće biti u mogućnosti da pronađe taj fajl u svojoj bazi i pogrešno će javiti da taj fajl ne postoji na sistemu. Naknadno ubacivanje informacija o instaliranom softveru van RPM metoda nije moguće i jedina opcija je da se zada tzv. forsirano instaliranje gde se neće voditi računa o međuzavisnosti paketa.

Komanda rpm

Sav posao oko upravljanja paketima u RPM sistemu je izведен preko komande **rpm**, korišćenjem različitih opcija. Komanda **rpm** je veoma moćna komanda i ovde će biti naveden samo deo opcija koji se najčešće koristi. Za potpunije upoznavanje sa **rpm** komandom i RPM sistemom za upravljanje softverskim paketima pogledajte

odgovarajuće man strane.

Komanda **rpm** ima tri osnovna moda rada:

- instaliranje softverskih paketa
- deinstaliranje softverskih paketa
- zadavanje upita o paketima

Instaljanje softverskih paketa

Kod RPM sistema postoje tri načina instaliranja paketa:

- instaliranje paketa
- ažuriranje (*upgrade*) instaliranog paketa
- osvežavanje (*freshen*) instaliranog paketa

Instaliranje paketa je osnovna opcija i ona će instalirati pakete koji su zadati kao argumenti u komandnoj liniji. Format ove komande je:

```
# rpm -i [opcije] rpm_paket ...
```

gde su najčešće opcije:

- v ispisivanje informacija vezanih za zadatu operaciju
- h ispisivanje 'progres trake' (sastavljene od znakova '#') kao indikacija procesa instaliranja
- nodeps forsirano instaliranje bez provere međuzavisnosti paketa

Argument **rpm_paket** je putanja do odgovarajućeg *.rpm fajla.

Ažuriranje paketa je operacija identična instalaciji uz tu razliku da posle operacije ažuriranja na sistemu ostaje instalirana samo najnovija verzija paketa, dok sve ostale verzije bivaju deinstalirane. Format komande **rpm** u ovom slučaju je:

```
# rpm -U [opcije] rpm_paket ...
```

gde su opcije iste kao kod instaliranja.

Osvežavanje paketa je operacija instaliranja paketa ali samo u slučaju da je neka prethodna verzija paketa već bila instalirana. U slučaju da prethodna verzija datog paketa nije instalirana, osvežavanje neće instalirati dati paket. Format **rpm** komande u ovom slučaju je:

```
# rpm -F [opcije] rpm_paket ...
```

gde su opcije iste kao kod instaliranja.

Deinstaliranje paketa

Deinstaliranje paketa se takođe obavlja **rpm** komandom koja u ovom slučaju ima format:

```
# rpm -e [opcije] paket ...
```

gde su najčešće korišćene opcije:

--nodeps forsiranje deinstaliranja paketa bez razrešavanja međuzavisnosti

Argument paket je naziv softverskog paketa za koji je, ukoliko njih više nije instalirano, dovoljno navesti samo naziv paketa i varijantu, ako postoji kao deo naziva. Ekstenzija .rpm se u ovom slučaju nikada ne navodi.

Deinstaliranje, u standardnoj formi, neće uspeti ako neki drugi instalirani paket zavisi od paketa kojeg hoćemo da deinstaliramo.

Postavljanje upita

Korisnik RPM sistema može pregledati informacije iz baze instaliranih paketa i iz samih paketa (koji nisu instalirani) korišćenjem komande **rpm**. Osnovni format komande 'rpm' u ovom slučaju je:

```
# rpm -q [opcije] [paket...]
```

gde se opcijama određuje koje informacije želimo da dobijemo:

-p rpm_paket umesto iz baze, informacije treba čitati iz navedenog *rpm_paket* fajla

-a ispis liste svih instaliranih paketa koji su zavedeni u bazi

-f fajl pronalaženje paketa kojem pripada zadati fajl *fajl*

-i dobijanje ekrana sa informacijama za dati paket ili rpm fajl

-l dobijanje liste fajlova koji sačinjavaju dati paket ili rpm fajl

-R dobijanje ispisa liste fajlova od kojih dati paket zavisi

Upravljanje DEB paketima

Deb softverski paketi

Debian je prilično razvijen projekat koji obuhvata veliki broj volontera koji su kreirali vrlo popularnu distribuciju, sa gomilom pomoćnih programa kako bi se distribucija lakše održavala i bila upotrebljivija većem broju korisnika.

Jedan od najpoznatijih podsistema koju je Debian razvio, a koju su, logično, preuzele sve izvedene distribucije jeste sistem pakovanja i organizovanja softverskih paketa.

Tako je razvijen DEB fomat paketa koji se koristi na ovim distribucijama.

Naravno, postavlja se pitanje zašto uopšte pakovati softver u pakete i koje su prednosti DEB paketa u odnosu na npr. običnu arhivu?

Prvo, jedan softverski paket obično se sastoji iz više fajlova:

- jednog ili više programa,
- jednog ili više konfiguracionih fajlova,
- odgovarajućih fajlova koji sačinjavaju dokumentaciju,
- ako je softverski paket složeniji, verovatno su u njega uključene i odgovarajuće biblioteke zajedničkog koda koji omogućavaju da druge aplikacije koriste API koji je u osnovi datog softverskog paketa.

Naravno, sve ovo mora biti instalirano u odgovarajućim direktorijumima, kako bi se zadržala struktura fajl sistema na Linux mašini. Sve ovo, naravno, može biti spakovano u obliku arhive i, u slučaju DEB paketa i jeste! Naime, u svojoj osnovi, DEB je tzv 'ar' arhiva koja u osnovi sadrži fajlove koji sačinjavaju softverski paket. No, pored tih fajlova, i dodatne informacije (takođe organizovane u obliku fajlova) se nalaze unutar jednog DEB paketa. Ti dodatni fajlovi sadrže informacije koje se tiču paketa kao što su:

- informacije o samom paketu (ime, spisak autora, odakle je uzet izvorni kod, opis paketa, interna verzija paketa, licenca pod kojom je paket licenciran i sl.)
- informacije o softveru koji je potreban za ispravno funkcionisanje paketa
- spisak fajlova koji sačinjavaju paket, sa njihovim izračunatim heševima (koji se koriste da bi se moglo utvrditi da li je fajl menjan u odnosu na originalni – tako se može utvrditi da li je npr. konfiguracioni fajl menjan u odnosu na onaj izvorni i, u slučaju kada se paket ažurira, utvrditi da li je moguće prebrisati postojeći konfiguracioni fajl ili ga treba ostaviti netaknutog)
- komande koje treba izvršiti pre instalacije paketa
- komande koje treba izvršiti po instalaciji paketa

Ove dodatne informacije se ubacuju u odgovarajuće baze: u sklopu repozitorijuma i u sklopu same instalacije.

Alati za upravljanje paketima

Na prethodnim kursevima smo upoznali neke alate, kao što je Synaptic, koji su namenjeni upravljanju paketima. U ovom kursu ćemo upoznati još neke, kako bismo imali opšti uvid u to kako sistem paketa i repozitorijuma funkcioniše.

Osnovni alat za upravljanje DEB paketima je komanda 'dpkg'. Njen format je:

```
# dpkg [opcija...] akcija [paket...]
```

Ova komanda je namenjena instalaciji, uklanjanju, pravljenju i pružanju informacija o paketima. Neke od ovih aktivnosti su implementirane kao zasebne komande koje 'dpkg' internu poziva.

Neki primeri komande 'dpkg' su:

```
# dpkg -i paket.deb
```

Ova komanda će instalirati paket 'paket.deb' koji se nalazi u tekućem direktorijumu.

```
# dpkg -l
```

Ova komanda će izlistati spisak instaliranih paketa i prikazati verziju i kratak opis za svaki od paketa.

```
# dpkg -L paket
```

Ova komanda će izlistati spisak fajlova prethodno instaliranog paketa 'paket.deb'.

```
# dpkg -S /putanja/fajl
```

Ova komanda će prikazati kojem paketu pripada fajl /putanja/fajl.

```
# dpkg -r paket
```

Ova komanda će ukloniti prethodno instalirani paket 'paket.deb'.

Kao što se iz primera vidi, 'dpkg' ne radi direktno sa repozitorijumima već upravlja prethodno instaliranim paketima ili paketima koji su prethodno skinuti sa mreže i postavljeni na lokalni fajl sistem. Takođe, 'dpkg' komanda ne ume da razlučuje međuzavisnosti izuzev što će detektovati da iste nisu ispunjene i neće u tom slučaju instalirati zadati paket.

Instaliranje softvera iz repozitorijuma

APT repozitorijumi

APT podsistem koristi /etc/apt direktorijum za smeštanje svoje konfiguracije, uključujući i spisak repozitorijuma kojima se obraća. Ovaj spisak se nalazi u fajlu /etc/apt/sources.list. Njegov format je:

```
tip_repozitorijuma URL verzija grupa ...
```

gde su:

- **tip_repozitorijuma** - 'deb' ili 'deb-src', prvi označava da su u pitanju paketi sa izvršnim fajlovima, a drugi označava da su u pitanju paketi sa spakovanim izvornim kodom (radi rebuildovanja binarnih paketa)
- **URL** - URL repozitorijuma (ukazuje na početni direktorijum stabla)
- **verzija** - verzija na koju se odnosi repozitorijum (istи repozitorijum može sadržati pakete za više različitih verzija iste distribucije)
- **grupa** - paketi su grupisani u grupe radi lakše manipulacije

APT podsistem

Podsistem APT se sastoji od nekoliko programa koji služe da olakšaju instalaciju, uklanjanje i razlučivanje međuzavisnosti među paketima. APT može da:

- dohvati informacije o različitim repozitorijumima,
- odredi međuzavisnosti među paketima, pronađe koji su dodatni paketi koje treba da budu instalirani da bi se međuzavisnosti razlučile ispravno
- pronađe iz kojih repozitorijuma je potrebno skinuti pakete za instalaciju
- dohvatiti pakete iz repozitorijuma
- instalirati pakete ispravnim redosledom tako da su međuzavisnosti uvek ispunjene,
- prikazati informacije o paketima
- pronaći pakete prema zadatim regularnim izrazima
- ukloniti pakete

- ukloniti pakete koji su instalirani kako bi se zadovoljile međuzavisnosti koje više nisu potrebne (npr. ako je prethodno uklonjen paket koji je zahtevao te međuzavisnosti),
- izvršiti kompletno ažuriranje sistema
- dodavanje novih repozitorijuma, posebno tzv. PPA repozitorijuma.

Komanda:

```
# apt-get update
```

će iz prethodno definisanog spiska repozitorijuma skinuti potrebne podatke (spisak paketa i njihovih međuzavisnosti) i ažurirati lokalnu bazu. Ovu proceduru automatski izvršava 'software-updater' aplikacija, ali se ista može pokretati i ručno.

```
# apt-get upgrade
```

Ova komanda će ažurirati ceo sistem na najnovije verzije paketa.

```
# apt-get install paket...
```

Ova komanda će instalirati navedeni spisak paketa, vodeći računa da se zadovolje međuzavisnosti. Ovo znači da će pored navedenih paketa možda biti instalirani i neki dodatni paketi (komanda će vas pitati o tome). Svi paketi će biti skinuti iz odgovarajućih repozitorijuma i instalirani redosledom koji u svakom momentu obezbeđuje da su međuzavisnosti uvek ispunjene.

```
# apt-get remove paket...
```

Ova komanda će ukloniti paket, ukoliko to ne narušava trenutne međuzavisnosti. Ukoliko uklanjanje paketa narušava međuzavisnosti, korisnik će biti upozoren na to i biće mu prezentovani koji paketi zavise od paketa koji je korisnik nameravao da ukloni.

```
$ apt-cache search izraz...
```

Ova komanda će pretražiti lokalnu bazu podataka tražeći koji paketi odgovaraju zadatom 'izrazu' i, ukoliko ih nađe, prikazaće ih kao spisak, uključujući i kratak opis svakog od pronađenih paketa.

Komanda:

```
$ apt-cache show paket...
```

će prikazati detaljnije informacije o paketima navedenim kao argumenti ove komande.

YUM podsistem

Yum je pandan 'apt' podsistemu na RedHat izvedenim distribucijama. Baziran je na sistemu repozitorijuma i, za razliku od 'apt' podsistema, implementiran je kao jedna komanda. Yum omogućava dobijanje informacija o dostupnim paketima, pretraživanje po delu imena ili opisa paketa, skidanje paketa sa repozitorijuma, instaliranje i uklanjanje paketa, kao i ažuriranje sistema na najnoviju verziju svih paketa. Yum automatski razrešava međuzavisnosti paketa koji se ažuriraju, instaliraju ili uklanjaju i automatski ažurira i instalira odgovarajuće verzije. Za razliku od 'apt'-a, yum može grupisati softverske pakete u grupe i jednom komandom instalirati celu grupu paketa. Takođe, yum je lako proširljiv sistem jer se bazira na sistemu plugin-ova. Yum podržava autentifikaciju paketa kroz metod potpisivanja paketa GPG ključevima koji moraju unapred biti instalirani na sistemu koji se instalira/ažurira. Autentifikacija paketa se može podešavati po pojedinim repozitorijumima.

Ažuriranje paketa

Komandom:

```
# yum check-update
```

dobijamo spisak paketa za koje postoje novije verzije i koje možemo ažurirati, npr:

```
# yum check-update
Loaded plugins: product-id, refresh-packagekit, subscription-manager
Updating Red Hat repositories.
INFO:rhsm-app.repolib/repos updated: 0
PackageKit.x86_64          0.5.8-2.el6      rhel
PackageKit-glib.x86_64       0.5.8-2.el6      rhel
PackageKit-yum.x86_64        0.5.8-2.el6      rhel
PackageKit-yum-plugin.x86_64 0.5.8-2.el6      rhel
glibc.x86_64                2.11.90-20.el6   rhel
glibc-common.x86_64         2.10.90-22     rhel
kernel.x86_64               2.6.31-14.el6   rhel
kernel-firmware.noarch       2.6.31-14.el6   rhel
rpm.x86_64                  4.7.1-5.el6    rhel
rpm-libs.x86_64              4.7.1-5.el6    rhel
rpm-python.x86_64            4.7.1-5.el6    rhel
udev.x86_64                 147-2.15.el6   rhel
yum.noarch                   3.2.24-4.el6   rhel
```

Ispis komande je:

- ime paketa i arhitektura
- verzija u repozitorijumu
- ime repozitorijuma gde se paket nalazi

Samo ažuriranje paketa obavljamo komandom:

```
# yum update [ime-paketa...]
```

Gornjom komandom će yum ažurirati zadate pakete. Ako izostavimo ime paketa, ceo sistem će biti ažuriran na najnoviju verziju. Yum će razrešiti međuzavisnosti i, pre nego što krene sa ažuriranjem, prikazati spisak paketa koji će biti ažurirani, a zatim tražiti potvrdu da to i odradi. Ukoliko ne želite da vas yum pita za dozvolu da odradi aktivnost koju ste mu zadali, koristite opciju -y.

Pretraživanje repozitorijuma

Yum ima nekoliko komandi kojima možete pretraživati repozitorijume u traganju za odgovarajućim paketima. Najčešće korišćena je:

```
# yum search izraz...
```

Ovom komandom će yum prikazati sve pakete koji u imenu ili svom opisu imaju zadati *izraz*. Na taj način ne morate znati kako se paket tačno zove da bi ste ga pronašli u nekom repozitorijumu.

```
# yum list glob-izraz...
```

Gornja komanda će prikazati sve pakete čije ime odgovara zadatom glob-izrazu. Unutar glob-izraza možete koristiti:

- * - zamenjuje bilo koji niz znakova
- ? - zamenjuje pojedinačan znak

Napomena

da bi izbegli da vaš komandni interpreter ekspanduje glob znakove u yum komandi, poništite njihovo specijalno značenje stavlјajući znak '\\' ispred glob znaka.

Primer:

```
# yum list abrt-addon\* abrt-plugin\*
Loaded plugins: product-id, refresh-packagekit, subscription-
manager
Updating Red Hat repositories.
INFO:rhsm-app.repolib:repos updated: 0
Installed Packages
abrt-addon-ccpp.x86_64           1.0.7-5.el6      @rhel
abrt-addon-kerneloops.x86_64      1.0.7-5.el6      @rhel
abrt-addon-python.x86_64          1.0.7-5.el6      @rhel
abrt-plugin-bugzilla.x86_64       1.0.7-5.el6      @rhel
abrt-plugin-logger.x86_64         1.0.7-5.el6      @rhel
abrt-plugin-sosreport.x86_64      1.0.7-5.el6      @rhel
abrt-plugin-ticketuploader.x86_64 1.0.7-5.el6      @rhel
```

Varijacije prethodne komande su:

```
# yum list all [glob-izraz...]
```

prikazuje sve instalirane i dostupne pakete koji odgovaraju zadatom glob-izrazu.

```
# yum list installed [glob-izraz...]
```

prikazuje sve instalirane pakete koji odgovaraju zadatom glob-izrazu.

```
# yum list available [glob-izraz...]
```

prikazuje sve pakete koji nisu instalirani a dostupni su u aktiviranim repozitorijumima.

Napomena: ako izostavite glob-izraz u gornjim komandama to je ekvivalentno zadavanju glob-izraza '*'.

Primer:

```
# yum list installed "krb?-*"
Loaded plugins: product-id, refresh-packagekit, subscription-
manager
Updating Red Hat repositories.
INFO:rhsm-app.repolib:repos updated: 0
Installed Packages
krb5-libs.x86_64                 1.8.1-3.el6      @rhel
krb5-workstation.x86_64            1.8.1-3.el6      @rhel
```

Prikaz svih dostupnih softverskih grupa se dobija komandom:

```
# yum grouplist
```

dok se spisak svih aktiviranih repozitorijuma dobija sa:

```
# yum repolist
```

Prikaz informacija o paketima

Detaljniji prikaz informacija o jednom ili više paketa možete dobiti komandom:

```
# yum info glob-izraz...
```

gde glob-izraz predstavlja ime paketa:

```
# yum info abrt
Loaded plugins: product-id, refresh-packagekit, subscription-
manager
Updating Red Hat repositories.
INFO:rhsm-app.repolib/repos updated: 0
Installed Packages
Name        : abrt
Arch       : x86_64
Version    : 1.0.7
Release    : 5.el6
Size       : 578 k
Repo       : installed
From repo : rhel
Summary    : Automatic bug detection and reporting tool
URL        : https://fedorahosted.org/abrt/
License    : GPLv2+
Description: abrt is a tool to help users to detect
              : defects in applications and to create a bug
              : report with all informations needed by
              : maintainer to fix it. It uses plugin system
              : to extend its functionality.
```

Ako želite da dobijete spisak svih definisanih softverskih grupa, to možete uraditi komandom:

```
# yum grouplist [glob-izraz...]
```

Ova komanda će prikazati imena svih dostupnih (instaliranih ili ne) grupa. Ako želite da dobijete ID-ove grupa, koristite opciju '-v'.

Ako želite da dobijete informaciju kojem paketu pripada neki fajl (bez obzira da li je taj paket instaliran ili ne), koristite komandu:

```
# yum provides glob-izraz...
```

Glob izraz ovde označava putanju do nekog fajla:

```
# yum provides "*bin/named"
Loaded plugins: product-id, refresh-packagekit, subscription-manager
Updating Red Hat repositories.
INFO:rhsm-app.repolib:repos updated: 0
32:bind-9.7.0-4.P1.el6.x86_64 : The Berkeley Internet Name Domain
(BIND)
                                                : DNS (Domain Name System) server
Repo        : rhel
Matched from:
Filename   : /usr/sbin/named
```

Instaliranje paketa

Instaliranje paketa se obavlja komandom:

```
# yum install ime...
```

Na mesto imena paketa možemo koristiti za yum standardne glob-izraze. U slučaju da želite da instalirate celu softversku grupu, koristite ID grupe sa prefiksom '@'.

Primeri:

```
# yum install sqlite.i686
# yum install perl-Crypt-\*
```

Umesto imena paketa možete zadati punu putanju nekog fajla koji pripada paketu kojeg želite da instalirate:

```
# yum install /usr/sbin/named
```

Već smo rekli da se cela grupa može instalirati preko 'yum install' komande, zadavanjem ID-a grupe:

```
# yum install @kde-desktop
```

Umesto ID-a grupe, možete koristiti ime grupe, ali u tom slučaju komanda je 'yum groupinstall':

```
# yum groupinstall „KDE Desktop“
```

Ime grupe se uvek zadaje bez ID-a grupe, mada komanda 'yum groupinstall' prima i ID grupe (bez imena) kao argument.

Prilikom instalacije, kao i pri ažuriranju paketa, yum će razrešiti sve međuzavisnosti i ponuditi da instalira sve nedostajuće pakete, odnosno ažurira postojeće ako njihova verzija ne zadovoljava zateve paketa koje treba instalirati.

Uklanjanje paketa

Uklanjanje pojedinačnih paketa se vrši komandom:

```
# yum remove ime-paketa...
```

Prilikom uklanjanja paketa, yum će odrediti da li neki instalirani paket zahteva paket koji želimo da uklonimo i ako je takav slučaj odbije da ukloni zahtevani paket.

Uklanjanje cele grupe se može obaviti sa:

```
# yum remove @ID-grupe...
```

ili sa:

```
# yum groupremove ime-grupe-ili-ID-grupe...
```

Napomena

ukoliko u `/etc/yum.conf`, u sekciji `[main]` nije postavljena direktiva `'groupremove_leaf_only=1'`, yum će ukloniti sve pakete iz grupe, bez obzira da li neki drugi paket zavisi od njih.

Dodavanje novih repozitorijuma

Svaki repozitorijum kojem možete pristupiti je opisan sa svojom strukturom. Obično se za svaki repozitorijum opis nalazi u zasebnom fajlu, koji se nalazi u `/etc/yum.repos.d/` direktorijumu. Primer opisa jednog repozitorijuma dat je za poznati RPMForge repozitorijum:

```
[rpmforge]
name = RHEL $releasever - RPMforge.net - dag
baseurl = http://apt.sw.be/redhat/e16/en/$basearch/rpmforge
mirrorlist = http://mirrorlist.repoforge.org/e16/mirrors-rpmforge
#mirrorlist = file:///etc/yum.repos.d/mirrors-rpmforge
enabled = 1
gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-rpmforge-dag
gpgcheck = 1

[rpmforge-extras]
name = RHEL $releasever - RPMforge.net - extras
baseurl = http://apt.sw.be/redhat/e16/en/$basearch/extras
mirrorlist = http://mirrorlist.repoforge.org/e16/mirrors-rpmforge-extras
#mirrorlist = file:///etc/yum.repos.d/mirrors-rpmforge-extras
enabled = 0
gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-rpmforge-dag
gpgcheck = 1

[rpmforge-testing]
name = RHEL $releasever - RPMforge.net - testing
baseurl = http://apt.sw.be/redhat/e16/en/$basearch/testing
mirrorlist = http://mirrorlist.repoforge.org/e16/mirrors-rpmforge-testing
#mirrorlist = file:///etc/yum.repos.d/mirrors-rpmforge-testing
enabled = 0
gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-rpmforge-dag
gpgcheck = 1
```

Kao što se može videti, u fajlu /etc/yum.repos.d/rpmforge.repo se nalaze opisi tri RPMForge repozitorijuma: rpmforge, rpmforge-extras i rpmforge-testing. Minimalan skelet opisa jednog repozitorijuma je:

```
[id-repozitorijuma]
name = Ime repozitorijuma
baseurl = URL-repozitorijum
```

gde su:

- ***id-repozitorijuma*** – niz znakova, bez blanko znaka, koji definiše repozitorijum
- ***Ime repozitorijuma*** – opisno ime repozitorijuma
- ***URL-repozitorijuma*** – URL gde se nalazi osnovni direktorijum repozitorijuma.

Najčešće dodatne direktive su:

- ***mirrorlist*** – URL do fajla koji sadrži spisak mirora originalnog sajta
- ***enabled*** – da li je repozitorijum aktiviran (1) ili ne (0)
- ***gpgkey*** – putanja do lokalnog fajla gde se nalazi javni GPG ključ kojim su potpisani paketi iz ovog repozitorijuma
- ***gpgcheck*** – da li se proverava GPG potpis paketa prilikom instalacije (1) ili ne (0)

Yum dozvoljava upotrebu varijabli u definiciji direktiva:

- `$releasever` – verzija distribucije
- `$arch` – tip arhitekture, kao što vraća 'uname -m'
- `$basearch` – tip arhitekture koji koristi yum

Aktiviranje i deaktiviranje repozitorijuma

Globalno aktiviranje repozitorijuma se obavlja postavljanjem direktive 'enabled=1' u fajl sa opisom repozitorijuma.

Globalno deaktiviranje repozitorijuma se obavlja postavljanjem direktive 'enabled=0' u fajl sa opisom repozitorijuma.

Yum kontaktira jedino aktivirane repozitorijume prilikom svojih operacija.

Privremeno aktiviranje globalno deaktiviranog repozitorijuma se obavlja zadavanjem opcije '`--enablerepo=repo-glob-izraz`' u yum komandi.

Privremeno deaktiviranje globalno aktiviranog repozitorijuma se obavlja zadavanjem opcije '`--disablerepo=repo-glob-izraz`' u yum komandi.

Brisanje yum keševa

Yum, radi povećanja brzine rada, često kešira informacije (metapodatke, pakete) na lokalnom disku. Ponekad ovi keševi ne vraćaju realne podatke i potrebno ih je obrisati (svaka yum operacija automatski osvežava keševe). Opšti oblik komande za to je:

```
# yum clean cilj...
```

gde je cilj:

- `expire-cache` – briše vremenske zapise o metapodacima i miror-listama koje su prethodno dovučene iz repozitorijuma. Naredna yum komanda će ponovo učitati ove podatke
- `packages` – briše sve lokalno keširane pakete
- `headers` – briše sva zaglavila koja su prethodne verzije yum-a koristile za razrešavanje međuzavisnosti
- `metadata` – briše sve metapodatke o dostupnosti paketa u repozitorijumima, tako da sledeća yum komanda mora te podatke ponovo dovlačiti iz repozitorijuma
- `dbcache` – briše sqlite keševe koji služe za brži pristup metapodacima, tako da sledeća yum komanda mora te keševe ponovo dovlačiti iz repozitorijuma

- **rpmb** – briše sve podatke iz lokalne rpmb baze
- **plugins** – signalizira svim aktiviranim plugin-ovima da obrišu svoje keševe
- **all** – zamena za navođenje svih gorepomenutih opcija.

Property of Admin Training Center

Umrežavanje servera

Pojam mreže

Računari se povezuju u mreže da bi se postiglo deljenje resursa (npr. štampača), razmena fajlova i elektronska komunikacija. Računari se u mrežu mogu povezati putem kablova, telefonskih linija, radio talasa, satelita...

Local Area Network (LAN) je mreža računara na relativno maloj međusobnoj udaljenosti. Najčešće se formiraju unutar poslovnih zgrada, škola, organizacija i sl.

Wide Area Network (WAN) je mreža povezuje udaljena geografska područja.

Računarske mreže povezuju računare različitih arhitektura, mogućnosti i performansi i zbog toga moraju biti dizajnirane tako da prevaziđu sve te razlike. Mrežni protokoli, kao što je TCP/IP obezbeđuju funkcije adresiranja računara, transporta i isporuke podataka i obezbeđuju uniformnost u radu sa različitim arhitekturama računara i tipova mreža.

IP adrese

IP adrese su logički način za adresiranje mrežnih uređaja na trećem nivou OSI modela. U Internet protokolu verzije 4 (IPv4) adrese su predstavljene sa 32 bita koji su podeljeni u 4 grupe decimalnih brojeva od 0-255. Primer IPv4 adrese je: 147.91.8.4.

Tradicionalno su postojale 3 osnovne klase mreža: A, B i C, svaka od kojih je imala predviđen određeni broj bitova za oznaku, a ostatak za oznaku uređaja u toj mreži. Klasa A ima najviše hostova ali postoji najmanje takvih mreža jer je odvojeno 24 bita za hostove). U klasi B je za oznaku hosta odvojeno 16 bita, a u klasi C 8 bita. Ovakve mreže se nazivaju Classful networks.

CIDR (Classless Inter-Domain Routing) je noviji i fleksibilniji način tumačenja IP adresa. Zajedno sa IP adresom host dobija i masku subneta koja određuje koji deo IP adrese određuje mrežu a koji deo određuje hosta. U primeru: 192.168.0.8/255.255.255.0 (/24) prvi deo, 192.168.0.8 predstavlja IP adresu, drugi deo 255.255.255.0 označava masku kojom se 24 bita odvaja za oznaku mreže, a ostalih 8 bita za oznaku hosta (ekvivalent klase C).

Klase i opsezi adresa

Opsezi adresa koje obuhvataju klase su prikazani u sledećoj tabeli:

Klasa	Početni bitovi	Start	Kraj	Broj mreža	Broj hostova po mreži
Klasa A	0	0.0.0.0	127.255.255.255	127	16,777,214
Klasa B	10	128.0.0.0	191.255.255.255	16,384	65,534
Klasa C	110	192.0.0.0	223.255.255.255	2,097,152	256
Klasa D	1110	224.0.0.0	239.255.255.255		268,435,456
Klasa E	1111	240.0.0.0	255.255.255.255		268,435,456

Pojedini opsezi adresa prema RFC 3330 imaju specijalne namene:

Klasa	Adrese	CIDR ekvivalent	Broj hostova	Svrha
A	0.0.0.0 - 0.255.255.255	0.0.0.0/8	16,777,216	Zero Addresses
A	10.0.0.0 - 10.255.255.255	10.0.0.0/8	16,777,216	Private IP addresses
A	127.0.0.0 - 127.255.255.255	127.0.0.0/8	16,777,216	Localhost Loopback Address
B	169.254.0.0 - 169.254.255.255	169.254.0.0/16	65,536	Microsoft APIPA
B	172.16.0.0 - 172.31.255.255	172.16.0.0/12	1,048,576	Private IP addresses
C	192.0.2.0 - 192.0.2.255	192.0.2.0/24	256	Documentation and Examples
C	192.88.99.0 - 192.88.99.255	192.88.99.0/24	256	IPv6 to IPv4 relay Anycast
C	192.168.0.0 - 192.168.255.255	192.168.0.0/16	65,536	Private IP addresses
C	198.18.0.0 - 198.19.255.255	198.18.0.0/15	131,072	Network Device Benchmark
D	224.0.0.0 - 239.255.255.255	224.0.0.0/4	268,435,456	Multicast
E	240.0.0.0 - 255.255.255.255	240.0.0.0/4	268,435,456	Reserved

Privatne IP adrese su u opsezima 10.0.0.0/8, 172.16.0.0/12 i 192.168.0.0/16 i one se ne rutiraju. Opseg adresa 127.0.0.0/8 naziva se loopback i referencira tu istu mašinu.

Koristi se radi testiranja mrežnih protokola.

IP mreže

IP mrežama se postiže logičko grupisanje mrežnih uređaja, tj. hostova. Svaki mrežni uređaj dobija jedinstvenu adresu koji ga, zajedno sa subnet maskom jedinstveno identificuje. Host se podesi tako da sluša mrežnu aktivnost vezanu za taj IP broj i na taj način se uspostavlja konekcija u trećem sloju OSI modela. Svaka mreža ima dve rezervisane IP adrese: adresu mreže i broadcast adresu. Adresa mreže identificuje samu mrežu, a broadcast adresu slušaju svi hostovi u toj mreži.

192.168.9.0/24	ip/subnet mask
192.168.9.0	adresa mreže
192.168.9.255	broadcast adresa
192.168.9.1 – 192.168.9.254	adrese hostova

Rutiranje

Pronalaženje puta od nekog hosta u jednoj mreži do drugog u nekoj drugoj mreži obavlja se procesom koji se naziva rutiranje. Mrežni uređaj koji je povezan na više od jedne mreže se ponaša kao ruter kada na osnovu različitih pravila i protokola za rutiranje odlučuje gde će da prosledi pakete koje primi, a koji nisu za njega.

Podrazumevana ruta (default gateway)

Kada je neki paket upućen ka mašini koja pripada nekoj drugoj mreži bilo gde na Internetu, komunikacija sa tim hostom se obavlja preko default gateway-a koji tu mrežu povezuje sa svim ostalim mrežama. Svaka mreža koja je povezana sa drugim mrežama mora da ima podešen default gateway koji pomoći različitim protokola za rutiranje određuje gde treba uputiti paket da bi on stigao na odredište.

Privatne mreže

Dogovorom je usvojeno da postoji tri opsega ip adresa koje se ne rutiraju, već ostaju privatne unutar jedne mreže. Sve ostale ip adrese se smatraju javnim i mogu se rutirati. To su opsezi:

- A klasa 10.0.0.0 do 10.255.255.255 (10.0.0.0/8)
- B klasa 172.16.0.0 do 172.31.255.255 (172.16.0.0/12)
- C klasa 192.168.0.0 do 192.168.255.255 (192.168.0.0/16)

DNS

Radi lakšeg pamćenja adresa hostova, uveden je domain name system kao način na koji se smislena tekstualna imena kao sto je www.google.com prevode u IP adresu. Name server je ima ulogu da održava bazu podataka IP adresa i imena hostova za jedan deo Interneta, a postoje i name serveri koji održavaju jedino bazu adresa drugih name servera (top level domain).

Podešavanje mrežnih parametara iz komandne linije

Komanda 'ifconfig'

Komanda ifconfig vrši podešavanje mrežnih interfejsa koji postoje na sistemu. Potrebno je pokrenuti komandu oblika:

```
# ifconfig INTERFACE IP_ADDRESS broadcast BROADCAST \ netmask  
      NETMASK
```

gde su *IP_ADDRESS*, *BROADCAST* i *NETMASK* odgovarajući parametri tog hosta, a *INTERFACE* je neki mrežni interfejs. Mrežni interfejsi su:

- eth0 - prvi ethernet interfejs
- lo - loopback interfejs koji pokazuje na istu mašinu
- ppp0 - prvi interfejs Point to Point Protokola, dakle dial-up

Primer podešavanja mreže hosta 192.168.0.10 sa netmaskom 255.255.255.0 je:

```
# ifconfig eth0 192.168.0.10 broadcast 192.168.0.255 \  
      netmask 255.255.255.0
```

Sama komanda ifconfig ispisuje već podešene interfejse, sa dodatnim informacijama među kojima je i MAC adresa uređaja (HWaddr),

```
# ifconfig
eth0 Link encap:Ethernet Hwaddr 00:04:76:8C:8C:51
      inet addr:147.91.13.14 Bcast:147.91.13.127 Mask:255.255.255.128
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:544238 errors:0 dropped:0 overruns:23 frame:0
          TX packets:394901 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:193251891 (184.2 Mb) TX bytes:50753617 (48.4 Mb)
          Interrupt:19 Base address:0xd000

lo  Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:459 errors:0 dropped:0 overruns:0 frame:0
          TX packets:459 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:41933 (40.9 Kb) TX bytes:41933 (40.9 Kb)
```

Komandi **ifconfig** mogu se proslediti mnogi parametri kojima će se podešiti različite opcije. Najvažniji su sledeći:

- **up** - ovo je podrazumevana vrednost kojom se interfejs stavlja u funkciju
- **down** - deaktivira interfejs
- **-a** - prikazuje konfiguracije svih interfejsa na sistemu
- **[-]arp** - omogućuje ili onemogućuje korišćenje ARP protokola za rezoluciju adresa
- **mtu *N*** - postavlja MTU (Maximum Transfer Unit)
- **netmask *addr*** - podešava subnet masku kojom se naznačava kojoj mreži host pripada
- **[-]broadcast [*addr*]** - podešava broadcast adresu protokola ukoliko se navede adresa, inače je briše
- **[-]pointopoint [*addr*]** - podešava adresu mašine na drugom kraju Point-to-Point linka

Komanda 'route'

Nakon podizanja interfejsa pomoću komande ifconfig, treba postaviti rute ka mrežama, dakle default gateway i ostale po potrebi.

Dodavanje podrazumevanje rute se vrši komandom:

```
# route add default gw ADDRESS metric 1
```

gde je *ADDRESS* adresa default gateway-a, npr:

```
# route add default gw 192.168.0.1
```

Uklanjanje podrazumevane rute se sa:

```
# route del default gw ADDRESS
```

Izlaz same komande route daje informacije o već definisanim rutama.

```
# route
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
192.168.0.0     *              255.255.255.0  U     0      0        0 eth0
link-local      *              255.255.0.0   U     0      0        0 eth0
loopback        *              255.0.0.0    U     0      0        0 lo
default         192.168.0.1   0.0.0.0     UG    0      0        0 eth0
```

Ove informacije uključuju:

- Destination - odredišni host ili mreža
- Gateway - adresa gateway-a ili '*' ako nije podešen za tu rutu
- Genmask - subnet maska za odredišnu mrežu. '255.255.255.255' za odredišni host, a '0.0.0.0' za podrazumevanu rutu.
- Flags
 - U (ruta je postavljena)
 - H (odnosi se na host)
 - G (koristi gateway)
 - R (za dinamičko rutiranje)
 - D (dinamički postavljeno od strane daemon-a ili redirekcijom)
 - M (prepravljeno od strane daemon-a ili redirekcijom)
 - A (postavljeno addrconf-om)
 - C (podatak je keširan)
 - ! (odbijena ruta)
- Metric - razdaljina do odredišta (obično navedena u hopovima). Više je kernel ne koristi, ali može biti potrebna daemonima za rutiranje.
- Ref - broj referenci na ovu rutu (Ne koristi se u Linux kernelu)
- Use - broj promašaja (uz -F) ili pogodaka (uz -C) ove rute
- Iface - interfejs preko kog se šalju paketi za ovu rutu.

Mapiranje DNS imena u IP adrese i obratno

Za podešavanje DNS servera potrebno je editovati fajl /etc/resolv.conf

Direktiva 'nameserver' označava redom od kojih DNS servera će se tražiti rezolvovanje IP adresa, a search (ili domain za samo jedan argument) omogućava da se specificira domen kome host pripada i da se on korisiti kao podrazumevan ukoliko se ne navede.

Primer /etc/resolv.conf fajla:

```
search etf.bg.ac.yu fon.bg.ac.yu
nameserver 147.91.8.62
nameserver 147.91.8.6
```

Komanda 'ip'

Komanda 'ip' dolazi sa velikim skupom mogućnosti koje nam mogu pomoći u upravljanju mrežnim interfejsima, podešavanju protokola i adresa, filtriranju mrežnog saobraćaja, listanju MAC tabela, rutiranju.

Osnovna sintaksa komande ip je:

```
# ip [opcija] [objekat] [komanda [ARGUMENTI . . . ]]
```

Opcije su:

- v prikaz verzije
- s statistika
- f, -t ip osnovnog seta protokola (familija) { inet, inet6, link }
 - 4 --> familija inet
 - 6 --> familija inet6
 - 0 --> familija link
- o ispis u jednoj liniji
- r rezolvovanje ip adresa

Objekti su:

- link** - fizički ili logički mrežni interfejs
- address** - IPv4 ili IPv6 adresa mrežnog interfejsa
- neighbour** - ARP ili NDSC sadržaj keša
- route** - ruting tabela

KOMANDE:

add

delete (del, d)

show (list, ls, l)

Rutiranje pomoću komande 'ip'

Upravljanje tabelama rutiranja se lako obavlja pomoću komande 'ip'. Dodavanje nove rute se zadaje na sledeći način:

```
# ip route add cilj via gateway
```

Primer:

```
# ip route add 192.168.2.1 via 212.200.61.2
```

Ostale komande vezane za rutiranje su:

ip route change - promena rute

ip route replace - promena rute ili dodavanje nove

ip route delete - brisanje rute

ip route show - lista ruta

Konfigurisanje mrežnog interfejsa

Promena atributa mrežnog interfejsa se obavlja pomoću komande ip u obliku:

```
# ip link set argument
```

gde je argument:

up/down - promena statusa UP ili DOWN

arp on/off - promena NOARP flega interfejsa

mtu NUMBER - promena MTU interfejsa

Primer: podizanje i spustanje interfejsa

```
# ip link set up eth0
# ip link set down eth0
```

Prikazivanje atributa interfejsa zadaje se komandom:

```
# ip link show [argument]
```

gde su argumenti:

dev NAME - prikaz atributa navedenog interfejsa (ako je ovaj argument izostavljen izlistaće se svi interfejsi)

UP - samo pokrenuti interfejsi

Primer: prikaz atributa interfejsa eth0 i lo

```
# ip link show eth0  
# ip link show lo  
# ip link show up
```

Primer: prikaz linkova

```
# ip link list
```

Upravljanje IP adresama

Komanda ip address omogućava pregledanje i upravljanje adresama i drugim osobinama konekcije.

Postavljanje nove IP adrese interfejsa se obavlja komandom:

```
# ip address add [local] ip-adresa argumenti...
```

gde su argumenti:

dev NAME - ime interfejsa kome će se dodati nova adresa

local ADDRESS - adresa interfejsa, format adrese zavisi od protokola

peer ADDRESS - adresa krajnje tačke za Point to Point Protokol

broadcast ADDRESS - broadcast adresa interfejsa

local NAME - svaka adresa može biti označena stringom

scope SCOPE_VALUE - zona validnosti adrese; SCOPE_VALUE je jedna od sledećih:
`site, link, host`

Primer: dodavanje ip adrese eth0 interfejsu

```
# ip address add 212.200.61.1/24 dev eth0
```

Brisanje adrese interfejsa se zadaje komandom:

```
# ip addr[ess] del[ete] adresa dev interfejs
```

Primer: brisanje ip adrese eth0 interfejsa

```
# ip address del 212.200.61.1/24 dev eth0
```

Prikaz adresa nekog interfejsa (ili svih interfejsa na sistemu) dobija se komandom:

```
# ip addr[ess] show [argument]
```

gde su argumenti:

dev NAME - ime interfejsa

scope SCOPE_VAL - samo adrese iz datog opsega

Upravljanje ARP tabelama se takođe može odraditi preko komande 'ip':

```
# ip neighbour [komanda [argument...]]
```

gde su komande:

add - dodavanje novog arp keša

change - promena postojećeg arp keša

replace - dodavanje novog arp keša ili promena postojećeg

delete - brisanje arp keša

show - listanje arp keša

Osnovni mrežni konfiguracioni fajlovi

/etc/hosts

Glavna namena ovog fajla je da omogući neposredno rezolvovanje imena u IP adrese, za ona imena koja ne mogu da se rezolvuju ni na koji drugi način. Ukoliko imate malu, zatvorenu mrežu, često je jednostavnije koristiti /etc/hosts umesto DNS servisa za rezolvovanje imena u adrese. Bez obzira da li koristite DNS ili ne, ovaj fajl treba da postoji i da sadrži minimum:

```
127.0.0.1 localhost.locaLdomain
```

/etc/resolv.conf

Ovaj fajl definiše IP adrese DNS servera i, eventualno, podrazumevani domen. Njegov format je:

```
nameserver ip-adresa
search domain ...
```

Možete definisati do tri 'nameserver' linije, tj. tri DNS servera. Na nekim sistemima se ovaj fajl automatski generiše od strane mrežnih startup skriptova.

Sistemsko podešavanje parametara mreže na Debian distribucijama

Ukoliko želimo da se postavljene adrese pamte i automatski postavljaju, možemo koristiti mrežni konfiguracioni fajl `/etc/network/interfaces`.

Preko sledećeg primera pokazaćemo format ovog fajla:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 192.168.1.1
    netmask 255.255.255.0
    gateway 192.168.1.254
    up flush-mail

auto eth1
iface eth1 inet dhcp
```

Linije koje su oblika 'auto iface' označavaju da se dati interfejs setuje automatski, prema zadatim pravilima, prilikom startovanja sistema.

Linije 'iface iface inet metod' označavaju način na koji se dodeljuje adresa interfejsa. Parametar 'metod' može biti 'static', gde se statički dodeljuje IP adresa, 'dhcp', gde se adresa dodeljuje preko DHCP protokola ili 'loopback' što je namenjeno isključivo za 'lo' interfejs (dodeljuje se adresa 127.0.0.1/8).

Kod 'static' metoda, dodatni parametri su:

- **address** – IP adresa interfejsa
- **netmask** – mrežna maska
- **gateway** – podrazumevana ruta sa adresom gejtveja
- **up|pre-up|post-up|down|pre-down|post-down** – komanda koja se izvršava pre ili posle podizanja odnosno zaustavljanja interfejsa.

Sistemsko podešavanje parametara mreže na RedHat baziranim distribucijama

Osnovno podešavanje mrežnih parametara se obavlja editovanjem fajla `/etc/sysconfig/network`.

Ovaj fajl definiše ime mašine, kao i podrazumevanu rutu (default gateway). U pitanju je običan sh skript fajl, tako da su njegove direktive u stvari dodele vrednosti specifičnim promenljivim:

```
NETWORKING="zahiev"  
HOSTNAME="fqdn-ime-hosta"
```

gde su:

zahiev - „yes“ ili „no“

fqdn-ime-hosta – puno ime koje je dodeljeno lokalnom serveru

Konfiguracioni fajlovi za interfejse

Pojedinačni interfejsi se konfigurišu fajlovima čije je ime oblika `/etc/sysconfig/network-scripts/ifcfg-interfejs`, gde je 'interfejs' ime interfejsa. I ovo su obični sh skript fajlovi, tako da se konfigurisanje svodi na dodelu vrednosti odgovarajućim promenljivim.

Osnovni skelet ovih fajlova sadrži sledeće direktive:

```
DEVICE="uredaj"  
BOOTPROTO="protokol"  
ONBOOT="zahiev"
```

gde su:

uredaj – ime interfejsa

protokol – specifikacija načina dodelje IP adrese interfejsu i može biti 'none' (postavljanje statičke adrese) ili 'dhcp' (postavljanje adrese preko DHCP protokola);

zahiev – 'yes' ili 'no', definiše da li će interfejs biti automatski aktiviran i dodeljena mu adresa prilikom startovanja sistema.

Ukoliko je u pitanju ethernet interfejs, poželjno je specificirati i njegovu MAC adresu u obliku:

```
HWADDR="ethernet-adresa"
```

Za statičko dodeljivanje IP adresu, potrebno je dodati i sledeće direktive:

```
IPADDR="ip-adresa"  
NETMASK="mrežna-maska"
```

gde su:

ip-adresa – IP adresa interfejsa

mrežna-maska – mrežna maska interfejsa, u obliku a.b.c.d (dotted quad)

Opcionie direktive su i:

```
GATEWAY="ip-adresa"  
USERCTL="zahiev"
```

gde su:

GATEWAY – adresa podrazumevane rute i dodaje se samo za onaj interfejs preko koga ta ruta prelazi

USERCTL – definiše da li običan korisnik može menjati mrežne parametre interfejsa

Statičke rute

Statičke rute definišu putanje do onih mreža ili pojedinačnih hostova koji ne mogu biti dosegnuti kroz podrazumevanu rutu. Ukoliko je potrebno definisati neku statičku rutu koja će ići kroz neki od interfejsa na serveru, koristi se fajl:

/etc/sysconfig/network-scripts/route-interfejs

Na primer, statičke rute koje prolaze kroz interfejs 'eth0' će biti definisane u fajlu */etc/sysconfig/network-scripts/route-eth0*, u obliku:

```
ciljna-adresa/maska via adresa-rutera dev interfejs
```

gde su:

ciljna-adresa – adresa mreže ili pojedinačnog hosta za koji postavljamo statičku rutu; ukoliko u ovom fajlu želimo da postavimo podrazumevanu rutu onda se ciljna-adresa/maska može zameniti rečju 'default'

maska – CIDR specifikacija mrežne maske (32 za rutu ka hostu)

adresa-rutera – adresa rutera koji je sledeći čvor u ruti (next-hop)

interfejs – ime lokalnog mrežnog interfejsa preko kojeg ruta prolazi

Primer:

```
default via 192.168.0.1 dev eth0  
10.10.10.0/24 via 192.168.0.1 dev eth0  
172.16.1.0/24 via 192.168.0.1 dev eth0
```

Property of Admin Training Center

Administriranje korisnika i grupa

Administriranje korisnika

Administriranje korisnika se sastoji od dodavanja korisnika, postavljanja ili menjanja njihovih lozinki, promene parametara korisnika, i brisanje korisnika sa sistema. Korisnicima se takođe može privremeno zabraniti rad na sistemu.

Dodavanje novog korisnika

Dodavanje novog korisnika se sastoji iz dve operacije: definisanja korisnika i postavljanja početne lozinke korisniku.

Definisanje korisnika se može obaviti korišćenjem komande `useradd` ili direktnim editovanjem `/etc/passwd` fajla.

Komanda `useradd` ima sledeći format:

```
# useradd [opcije] korisnik
```

gde su najčešće opcije:

-m popunjavanje home direktorijuma fajlovima i direktorijumima definisanim u skeleton direktorijumu

-c komentar specificiranje informacija o korisniku kao što su njihovo ime i sl.

Parametar *komentar* mora biti pod navodnicima ako sadrži beline i ne sme sadržati znak `:`

-d direktorijum home direktorijum korisnika

-g grupa osnovna grupa kojoj korisnik pripada (zadata numerički ili preko imena grupe). Navedena grupa mora unapred postojati.

-s shell zadavanje komandnog interpretera korisnika

-u UID zadavanje specificiranog UID-a korisniku

-e datum_isticanja zadavanje datuma isticanja naloga (u formatu GGGG-MM-DD)

-f vreme_zaključavanja zadavanje vremena (u danima) posle isticanja lozinke nakon kojeg se nalog automatski zaključava. Vrednost parametra 0 označava zaključavanje odmah po isticanju lozinke, -1 označava isključivanje ove opcije.

Ukoliko opcije *d*, *s*, *e*, *f*, *u* nisu navedene, uzimaju se podrazumevane vrednosti. Ostale opcije nemaju podrazumevane vrednosti.

Dodeljivanje početne lozinke korisniku se obavlja komandom `passwd`, čiji je format u

ovom slučaju:

```
# passwd korisnik
```

Ova komanda se koristi i za promenu lozinke korisniku.

Modifikovanje parametara korisnika

Modifikovanje parametara korisnika se može obaviti komandom usermod ili direktnim editovanjem /etc/passwd, odnosno /etc/shadow fajla.

Komanda usermod ima sledeći format:

```
# usermod [opcije] username
```

gde su najčešće opcije iste kao kod komande useradd.

Brisanje korisnika

Brisanje korisnika može se izvesti komandom userdel ili manuelno, brisanjem odgovarajućih podataka iz fajlova /etc/passwd, /etc/shadow i /etc/group, kao i brisanja fajlova koji su pripadali obrisanom korisniku.

Komanda userdel ima sledeći format:

```
# userdel [opcije] username
```

gde su opcije:

-r briše rekurzivno korisnikov home direktorijum. Ostali fajlovi koji su van korisnikovog home direktorijuma moraju biti ručno obrisani ili će njihov vlasnik postati neki drugi novokreirani korišnik koji bude dobio UID koji je imao obrisani korisnik.

Zaključavanje i otključavanje korisnikovog naloga

Korisnički nalog se privremeno može zaključati i otključati komandama passwd, odnosno usermod.

Komanda passwd u ovom slučaju ima sledeći format:

```
# passwd [opcije] username
```

gde su opcije:

-l zaključavanje naloga (ispred enkriptovane lozinke se upiše znak '!' što onemogućava njeno dekriptovanje)

-u otključavanje naloga (vraćanje lozinke na vrednost pre zaključavanja).

Komanda usermod u ovom slučaju ima format:

```
# usermod [opcije] username
```

gde su opcije:

- L** zaključavanje naloga (ispred enkriptovane lozinke se upiše znak '!' što onemogućava njeno dekriptovanje)
- U** otključavanje naloga (vraćanje lozinke na vrednost pre zaključavanja).

Struktura /etc/passwd i /etc/shadow fajlova

U oba slučaja fajlovi se sastoje iz linija čiji je format naveden u nastavku.

/etc/passwd ima sledeći format linije:

```
korisnik:lozinka:UID:GID:komentar:homedir:shell
```

gde su polja:

korisnik ime korisnika na sistemu

lozinka kriptovana lozinka ili samo x ukoliko se koristi /etc/shadow fajl

UID numerički korisnički ID

GID numerički ID primarne grupe kojoj korisnik pripada

komentar ovo polje je opciono i koristi se samo za informacije. Obično se koristi samo za puno ime korisnika

homedir korisnički home direktorijum.

shell Program koji se startuje prilikom logina (ukoliko je prazno koristi se /bin/sh). Ukoliko se postavi nepostojeća putanja do programa korisnik neće moći da se uloguje.

/etc/shadow ima sledeći format:

```
korisnik:lozinka:lastchange:min:max:warn:inact:expire:res
```

gde su polja:

korisnik ime korisnika

lozinka kriptovani password

lastchange broj dana od 1970-01-01 kada je password poslednji put promenjen

min broj dana koji mora isteći pre nego što password može biti promenjen

max broj dana posle kojih password **mora** biti promenjen

warn broj dana pre isteka passworda od kada korisnik biva upozoren o isteku passworda

inact broj dana od isticanja passworda posle kojih će nalog biti zaključan

expire broj dana od 1970-01-01 kada nalog biva zaključan

res rezervisano za buduća proširenja

Skeleton direktorijum

Direktorijum koji služi kao šablon za korisničke direktorijume novih korisnika. Administrator može popunjavati ovaj direktorijum sa fajlovima za koje želi da se nađu u korisnikovom direktorijumu prilikom kreiranja novog naloga. Svi fajlovi iz skeleton direktorijuma će se rekurzivno prekopirati u *home* direktorijum novog korisnika koji će postati njihov vlasnik.

Postavljanje podrazumevanih parametara za autentifikaciju korisnika

Mnogi parametri **useradd** komande mogu biti unapred definisani. U ovom slučaju, **useradd** komanda ima sledeći format:

```
# useradd -D [opcije]
```

gde su opcije:

-b postavljanje prefiksa za *home* direktorijume korisnika. Prefiks je nadređeni direktorijum u kojem će se kreirati *home* direktorijum korisnika *korisnik* (podrazumevano '/home')

-s shell postavljanje podrazumevanog komandnog interpretera

-g grupa postavljanje podrazumevane grupe koja može biti zadata numerički ili preko naziva grupe

-e datum_isticanja zadavanje podrazumevanog datuma isticanja naloga (u formatu GGGG-MM-DD)

-f vreme_zaključavanja zadavanje podrazumevanog vremena (u danima) posle isticanja lozinke nakon kojeg se nalog automatski zaključava. Vrednost parametra 0 označava zaključavanje odmah po isticanju lozinke, -1 označava isključivanje ove opcije.

Ukoliko se koristi bez opcija, komanda **useradd -D** prikazuje tekuće podrazumevane vrednosti parametara, koje se inače čuvaju u fajlu */etc/default/useradd*.

Administriranje grupa

Korisnici na Linux sistemu su podeljeni na grupe. Svaki korisnik mora pripadati makar jednoj grupi, a može biti član do 32 grupe istovremeno.

Dodavanje nove grupe

Dodavanje nove grupe na sistem može se izvesti upotrebom komande **groupadd** ili direktnim editovanjem /etc/group fajla.

Komanda **groupadd** ima format:

```
# groupadd [opcije] naziv_grupe
```

gde su opcije :

-g GID postavljanje specifičnog GID-a za novokreiranu grupu

groupadd dodaje novu grupu. Bez opcija, sistem bira prvi sledeći ceo broj za novi GID. Opcijom **-g GID** možemo sami postaviti GID za novokreiranu grupu. Novokreirana grupa nema članova.

Brisanje grupe

Brisanje grupe sa sistema može se izvesti korišćenjem komande **groupdel** ili direktnim editovanjem /etc/group fajla.

Komanda **groupdel** ima format:

```
# groupdel naziv_grupe
```

groupdel briše grupu. Obrisana grupa ne sme biti primarna grupa nijednog korisnika.

Učlanjivanje korisnika u grupu

Učlanjivanje korisnika u grupu se obavlja prilikom dodavanja novog korisnika na sistem (komanda **useradd**), modifikovanjem parametara korisnika (komanda **usermod**) ili, najčešće, direktnim editovanjem sistemskog fajla /etc/group u kojem su definicije grupe.

Fajl /etc/group

Na standardnom Linux sistemu, definicija grupe se čuva u fajlu /etc/group. Svaka linija ovog fajla definiše jednu grupu. Jedna linija ima sledeći format zapisa:

```
naziv_grupe:lozinka:GID:korisnik, korisnik...
```

Na mestu lozinke se obično nalazi znak 'x' koji označava da je lozinka definisana na drugom mestu (fajl /etc/gshadow). Ukoliko u ovom fajlu ne postoji zapis za datu

grupu, smatra se da je lozinka nepostojeća. Ukoliko ovaj fajl ne postoji, smatra se da ni jedna grupa nema lozinku. Ukoliko lozinka za grupu postoji, korisnik koji je član grupe mora uneti ovu lozinku da bi promenio svoju aktivnu grupu.

Administriranje udaljenih servera

SSH pristup

SSH je poznati protokol enkriptovane konekcije ka udaljenom računaru. SSH je više od običnog softvera za logovanje na udaljeni sistem – preko njega je moguće kopirati fajlove (koristeći SCP ili SFTP protokol koji ide u paketu sa SSH-om) ili postaviti enkriptovani VPN tunel prema udaljenom računaru. SSH takođe ume da prosleđuje konekciju udaljene X aplikacije ka lokalnom X serveru što omogućava da se jednostavno, iz komandne linije, pokreću grafičke aplikacije koje će se izvršavati na udaljenoj mašini ali će se prikazivati i primati input sa tastature i pointerskog uređaja lokalne grafičke sesije. SSH nudi različite načine autentifikacije koje čine da ova konekcija bude veoma sigurna – SSH je osnovni alat administratora koji administriraju servere!

OpenSSH je standardna implementacija SSH protokola na Linuxu (OpenSSH je nastao kao podprojekat OpenBSD operativnog sistema koji je poznat po veoma visokom stepenu sigurnosti).

Preduslov za udaljeni pristup preko SSH protokola je da na udaljenoj mašini bude pokrenut SSH servis. Komunikacija ide po portu 22/tcp pa ovaj port treba biti otvoren na udaljenom računaru. Dakle, ukoliko nije instaliran, treba instalirati paket 'openssh-server'.

Sa instaliranim OpenSSH serverom na udaljenoj mašini, moguće je koristiti oba načina rada na Linuxu – grafički i rad iz komandne linije.

SSH pristup preko komandne linije

Standardna komanda za pristup udaljenom serveru je:

```
$ ssh username@hostname.domain.tld
```

Ova komanda će otvoriti terminalski pristup ka udaljenom računaru 'hostname.domain.tld' na koji ćete se prijaviti kao korisnik 'username'.

Standardno, prijavite se na udaljeni sistem unošenjem lozinke za korisnika 'username' na udaljenom računaru, ali to nije jedini način autentifikacije koji SSH podržava. Iako SSH ostvaruje kriptovanu konekciju između klijenta i servera, puno bolji način autentifikacije je korišćenjem tajnog i javnog ključa. Prednosti ove autentifikacije, izuzev što se lozinka nikada ne prenosi kroz konekciju su ta, da sa jednim setom

ključeva (javni i tajni) možete pristupati neograničenom broju udaljenih računara.

Generisanje javnog i tajnog ključa za SSH autentifikaciju

Da bismo ostvarili autentifikaciju preko javnog i tajnog ključa, moramo ih prvo generisati. Ovo se izvodi komandom:

```
$ ssh-keygen -t rsa
```

Ovo je interaktivna komanda i prvo će ponuditi fajl u koji će snimiti vaš tajni i javni ključ. Naziv ovog fajla možete promeniti jer možete imati neograničen broj tajnih i javnih ključeva. Napomena: svakom tajnom ključu odgovara **tačno jedan** javni ključ i oni uvek idu u paru. Nakon toga treba da unesete lozinku kojom možete otključati svoj tajni ključ (softver podržava i izostavljanje lozinke – samo pritisnite ENTER na upit). Nakon toga, biće generisan par tajni-javni ključ i biće vam isписан 'otisak' ovog ključa (radi dodatne provere prilikom logovanja).

Ako niste menjali naziv i lokaciju fajlova, oni će biti kreirani u '.ssh' poddirektorijumu vašeg početnog direktorijuma. Obratite pažnju na prava pristupa za ovaj poddirektorijum (sva prava za vlasnika, nikakva prava za grupu i ostale), kao i za fajl u kojem se nalazi vaš tajni ključ (podrazumevano 'id_rsa', vlasnik ima 'rw' prava, grupa i ostali nemaju nikakva). Ovakva prava su neophodna da bi sistem funkcionišao – posebno vodite računa o vašim tajnim ključevima: ukoliko se neko zlonameran dokopanjih imaće pristup svim udaljenim računarima na kojima ste ih instalirali.

Drugi potreban korak da biste mogli da se autentifikujete na udaljeni računar preko javnog ključa je da morate taj ključ instalirati u '.ssh' direktorijum korisnika na čiji nalog se logujete na udaljenom računaru. Prvo prekopirajte svoj **javni ključ** (podrazumevano, to je fajl 'id_rsa.pub' iz vašeg .ssh/ direktorijuma) na udaljeni računar (možete koristiti 'scp' komandu ili preko fajl brauzera). Zatim na udaljenom računaru, ako već ne postoji, kreirajte u početnom direktorijumu korisnika na čiji nalog se logujete direktorijum '.ssh/' koji treba da ima sva prava pristupa za vlasnika, i nikakva prava za grupu i ostale. Zatim ubacite vaš javni ključ u fajl 'authorized_keys' u udaljenom '.ssh/' direktorijumu. Ako ovaj fajl ne postoji, najjednostavnije je prekopirati vaš javni ključ (koji ste već prebacili na udaljeni računar) u fajl 'authorized_keys'. Ako, pak, ovaj fajl već postoji, treba dodati sadržaj vašeg javnog ključa na kraj ovog fajla. Ovo možete uraditi iz editora (vodite računa da je sadržaj vašeg javnog ključa isписан u jednom redu i tako treba da bude i u fajlu 'authorized_keys'). Alternativno, možete iskoristiti komandu 'cat' na udaljenom računaru (u primeru dole, prepostavljamo da je vaš javni ključ iskopiran u fajl id_rsa.username u početnom direktorijumu udaljenog korisnika):

```
$ cat ~/id_rsa.hostname >> ~/.ssh/authorized_keys
```

Posle ovoga, prilikom logovanja na udaljeni računar nećete morati unositi lozinku

korisnika već lozinku kojom ste zaključali svoj privatni ključ! U slučaju da to radite iz grafičkog okruženja, poseban softver koji se automatski pokreće prilikom prijavljivanja na sistem (ssh-agent) će pokrenuti dijalog prozor u kojem trebate uneti ovu lozinku i to samo prvi put u toku jedne sesije. Sve kasnije ssh, scp i fajl brauzer komande koje se kače na udaljeni sistem će lozinku za otključavanje vašeg tajnog ključa dobijati direktno od softvera 'ssh-agent' što znači da nećete morati da unosite svoju lozinku.

Kopiranje fajlova iz komandne linije na udaljeni računar

Kopiranje fajlova na udaljeni računar možete izvesti komandom:

```
$ scp lok_fajl username@hostname.domain.tld:/putanja/fajl
```

odnosno

```
$ scp username@hostname.domain.tld:/putanja/fajl lok_fajl
```

Rekurzivno kopiranje se postiže dodavanjem opcije '-r' komandi 'scp'.

Property of Admin Training Center

Konfigurisanje osnovnih lokalnih servisa

Konfigurisanje osnovnih lokalnih servisa na RedHat i izvedenim distribucijama

Upotreba chkconfig programa

Program `chkconfig` je alat koji omogućava specificiranje u kom ranlevelu će se startovati zadati servis. Ranleveli su skupovi pravila koji definišu ponašanje servera.

Servisi su definisani preko svojih start skriptova, koji se nalaze u `/etc/init.d/` direktorijumu.

Prikaz servisa i njihovog statusa u različitim ranlevelima

Komandom:

```
# chkconfig --list
```

dobićete prikaz stanja instaliranih servisa u odnosu na ranlevele. Ranleveli su označeni brojevima i predstavljaju kolone ispisa:

NetworkManager	0:off	1:off	2:on	3:on	4:on	5:on	6:off
abrt	0:off	1:off	2:off	3:on	4:off	5:on	6:off
acpid	0:off	1:off	2:on	3:on	4:on	5:on	6:off
anamon	0:off	1:off	2:off	3:off	4:off	5:off	6:off
atd	0:off	1:off	2:off	3:on	4:on	5:on	6:off
auditd	0:off	1:off	2:on	3:on	4:on	5:on	6:off
avahi-daemon	0:off	1:off	2:off	3:on	4:on	5:on	6:off
... nekoliko linija izostavljeno ...							
wpa_supplicant	0:off	1:off	2:off	3:off	4:off	5:off	6:off

Status 'on' označava da će servis biti automatski pokrenut u tom ranlevelu, dok status 'off' označava da taj server neće biti pokrenut, odnosno da će biti ugašen ako je bio pokrenut pre prelaska u dati ranlevel.

Za prikaz stanja pojedinačnog servisa koristite istu komandu:

```
# chkconfig --list ime-servisa
```

Na primer, da vidite status OpenSSH servisa, zadajte komandu:

```
# chkconfig --list sshd  
sshd          0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

Aktiviranje servisa

Ranleveli 2, 3, 4 i 5 služe za pokretanje sistema. Ranleveli 0 i 6 predstavljaju sistem u stanju restarta, odnosno gašenja, dok ranlevel 1 predstavlja sistem u tzv. single-user modu.

Da biste podesili da se zadati servis automatski startuje u ranlevelima 2, 3, 4 i 5, zadajte komandu:

```
# chkconfig ime-servisa on
```

Na primer, da aktivirate Apache veb servis, kucajte:

```
# chkconfig httpd on
```

Ako želite da aktivirate neki servis samo u određenim ranlevelima, zadajte istu komandu sa dodatnom opcijom:

```
# chkconfig ime-servisa on --level ranleveli
```

gde je parametar 'runleveli' zadat kao niz cifara koje definišu ranlevele u kojima želite da startujete servis.

Na primer, da aktivirate abrtd servis u ranlevelima 3 i 5, zadajte komandu:

```
# chkconfig abrtd on --level 35
```

Aktivirani servisi će biti startovani kada sistem sledeći put bude prelazio u zadate ranlevele.

Deaktiviranje servisa

Da biste deaktivirali servis u standardnim ranlevelima zadajte komandu:

```
# chkconfig ime-servisa off
```

Na primer, da deaktivirate httpd servis (Apache veb server), kucajte:

```
# chkconfig httpd off
```

Ako želite da deaktivirate servis u zadatim ranlevelima, koristite '--level' opciju, kao kod aktiviranja servisa.

Pokretanje servisa iz komandne linije

Start skriptovi koji se nalaze u /etc/init.d/ direktorijumu, omogućavaju da

upravljate servisima pozivajući ove skriptove sa odgovarajućim argumentima – komandama. Alternativno, možete koristiti komandu 'service' za upravljanje servisima, ako ne želite da kucate punu putanju do start skripta.

Dobijanje trenutnog statusa servisa

Ako želite da u kom stanju je trenutno zadati servis, koristite komandu:

```
# service ime-servisa status
```

ili pozovite start skript direktno sa:

```
# /etc/init.d/ime-servisa status
```

Na primer, da biste utvrdili da li je servis 'httpd' pokrenut ili ne, zadajte komandu:

```
# service httpd status
httpd (pid 7474) is running...
```

Da biste dobili status svih dostupnih servisa odjednom, zadajte komandu:

```
# service --status-all
abrt (pid 1492) is running...
acpid (pid 1305) is running...
atd (pid 1540) is running...
auditd (pid 1103) is running...
automount (pid 1315) is running...
Avahi daemon is running
cpuspeed is stopped
... nekoliko linija izostavljeno ...
wpa_supplicant (pid 1227) is running...
```

Pokretanje servisa

Komandama:

```
# service ime-servisa start
```

ili

```
# /etc/init.d/ime-servisa start
```

možete pokrenuti zadati servis. Na primer, ako želite da pokrenete 'httpd' servis, kucajte:

```
# service httpd start
Starting httpd: [ OK ]
```

ili

```
# /etc/init.d/httpd start  
Starting httpd:
```

[OK]

Zaustavljanje servisa

Servis možete regularno zaustaviti sledećom komandom:

```
# service ime-servisa stop
```

ili

```
# /etc/init.d/ime-servisa stop
```

Na primer, da biste zaustavili 'httpd' servis prvom komandom, kucajte:

```
# service httpd stop  
Stopping httpd:
```

[OK]

Restartovanje servisa

Restartovanje servisa je obično operacija koja prvo zaustavlja servis pa ga startuje, tako da možete da koristite dve 'service' komande jednu za drugom. No, start skriptovi omogućavaju da se to obavi automatski, zadavanjem komandi:

```
# service ime-servisa restart
```

ili

```
# /etc/init.d/ime-servisa restart
```

Na primer, da biste restartovali 'httpd' servis, zadajte komandu:

```
# service httpd restart  
Stopping httpd:  
Starting httpd:
```

[OK]
[OK]

Ostale varijante 'service' komande

Komanda 'service' prosleđuje zadatu komandu (start/stop/restart) start skriptu, tako da, ako skript ima još definisanih komandi, one mogu da se koriste i u 'service' komandi za dati servis. Najčešće od njih su:

reload – ponovno učitavanje konfiguracionog fajla, ako servis omogućava rekonfiguraciju u letu,

force-reload – forsirano učitavanje nove konfiguracije u servis koji se trenutno izvršava.

Cron servis

Servis cron je namenjen periodičnom izvršavanju komandi. Ovaj servis se pokreće prilikom startovanja sistema i aktivan je sve dok se sistem ne spusti. Svakog minuta, cron nanovo učitava svoje tabele sa komandama i izvršava one komande čija specifikacija vremena izvršavanja odgovara trenutnom datumu i vremenu. Ukoliko komanda ispisuje bilo kakav ispis na standardni izlaz ili standardni izlaz za greške, a koji nije preusmeren u neki fajl, takav ispis će biti sačuvan od strane cron servisa i prosleđen kao mejl poruka vlasniku tabele u kojoj se data komanda nalazi.

Kreiranje cron tabela

Specifikacije komandi koje treba da se izvrše, kao i vremenski intervali u kojima te komande treba izvršavati se zadaju preko cron tabela. Inicijalno, postoji samo sistemska cron tabela, koja se nalazi u fajlu `/etc/crontab`. Takođe, root korisnik može imati svoju tabelu. Ostali korisnici mogu imati svoje cron tabele samo ako postoji neki od fajlova `/etc/cron.allow`, odnosno `/etc/cron.deny`. Ako postoji `/etc/cron.allow` cron tabele mogu imati samo oni korisnici čija su korisnička imena navedena u tom fajlu. Ukoliko `/etc/cron.deny` postoji, samo korisnici koji nisu navedeni u ovom fajlu mogu koristiti cron servis i imati svoje cron tabele. Ako ni jedan od ova dva fajla ne postoji, samo root korisnik može koristiti cron servis.

Korisničke cron tabele čuvaju se u zasebnim fajlovima unutar `/var/spool/cron` direktorijuma. Ove tabele su u obliku tekstualnih fajlova, ali se ne smeju direktno editovati, već se to radi pozivanjem komande `crontab`. Format ove komande je:

```
# crontab [opcija]
```

gde su opcije:

- l listanje cron tabele
- e editovanje cron tabele
- r brisanje cron tabele
- u *username* ukoliko je ova opcija zadata, ostale opcije se odnose na cron tabelu navedenog korisnika, a ne na sopstvenu.

Sistemska cron tabela se edituje direktno kao fajl `/etc/crontab` i ima isti format kao korisničke cron tabele.

Pored sistemske cron tabele, postoji mogućnost zadavanja sistemskih komandi koje će se izvršavati na svaki sat, jednom dnevno, jednom nedeljno ili jednom u mesecu. Ove komande se u obliku skriptova koje treba izvršiti smeštaju u direktorijume `/etc/cron.hourly`, `/etc/cron.daily`, `/etc/cron.weekly` i

/etc/cron.monthly respektivno. Pored ovoga, parcijalne cron tabele se mogu definisati u direktorijumu /etc/cron.d.

Editovanje cron tabela

Editovanje cron tabele se obavlja u editoru koji je naveden u \$EDITOR promenljivoj okruženja. Format fajla je tekstualni, gde se linije koje počinju sa znakom '#' tretiraju kao komentari. Prazne linije se zanemaruju. Ostale linije moraju biti ili specifikacije parametara cron servisa ili specifikacije komandi i intervala u kojima se data komanda izvršava.

Specifikacije parametara su oblika

```
ime = vrednost
```

gde je *ime* naziv neke promenljive okruženja.

Specifikacije komandi imaju sledeći format:

```
minut sat dan_u_mesecu mesec dan_u_nedelji komanda
```

gde su:

- *minut* vremenska specifikacija za minut u kojem se komanda izvršava
- *sat* vremenska specifikacija za sat u kojem se komanda izvršava
- *dan_u_mesecu* vremenska specifikacija za dan u mesecu kada se komanda izvršava
- *mesec* vremenska specifikacija za mesec u kojem se komanda izvršava
- *dan_u_nedelji* vremenska specifikacija za dan u nedelji u kojem se komanda izvršava
- *komanda* komandna linija koja treba da se izvrši

Vremenske specifikacije za navedene parametre su brojčane vrednosti iz sledećeg opsega:

- minut 0-59
- sat 0-23
- dan_u_mesecu 1-31
- mesec 1-12
- dan_u_nedelji 0-7 (i 0 i 7 predstavljaju nedelju)

Pored navedenih vrednosti, vremenske specifikacije mogu biti i opsezi oblika *početak-*

kraj što označava vremenski interval koji počinje vrednošću *početak* a traje do vrednosti *kraj*, uključivo. Celokupan opseg za neki vremenski parametar se alternativno može zadati u obliku ***. Opseg može biti zadat i u koraku različitom od 1, oblika '*početak-kraj/korak*' ili '**/korak*'.

Vremenski opsezi mogu biti i liste termina, razdvojene znakom ',' bez belina između. Termini mogu biti pojedinačne vrednosti ili opsezi.

Posle editovanja neke cron tabele nije potrebno restartovati cron servis pošto on sve tabele učitava iznova svakog minuta.

```
# Everyday at 2:00 am run evl log manager and remove records
# that are older than 30 days
0 2 * * * root /sbin/evlogmgr -c 'age > "30d"'  
  
# Everyday at 1:00 am run evl log manager and remove records
# with severity=DEBUG
0 1 * * * root /sbin/evlogmgr -c "severity=DEBUG"
```

Property of Admin Training Center

Konfigurisanje osnovnih mrežnih servisa

Konfiguriranje OpenSSH servisa

U /etc/ssh direktorijumu nalaze se dva konfiguraciona fajla:

- `ssh_config` – namenjen konfiguraciji klijentskih programa (ssh, scp, sftp)
- `sshd_config` – namenjen konfiguraciji servera (sshd dimon)

Korisnik može imati svoju konfiguraciju klijentskih programa, u fajlu `~/.ssh/config`. Ova, lokalna, podešavanja imaju prioritet u odnosu na sistemska podešavanja u `/etc/ssh/ssh_config`.

Ostali fajlovi u /etc/ssh direktorijumu su:

- `/etc/ssh/moduli` – fajl koji sadrži Diffie-Hellman grupe korišćene prilikom razmene ključeva i uspostavljanja bezbedne konekcije
- `/etc/ssh/ssh_host_dsa_key` – privatni DSA ključ servera
- `/etc/ssh/ssh_host_dsa_key.pub` – javni DSA ključ servera
- `/etc/ssh/ssh_host_key` – privatni RSA ključ servera za SSH protokol verzije 1
- `/etc/ssh/ssh_host_key.pub` – javni RSA ključ servera za SSH protokol verzije 1
- `/etc/ssh/ssh_host_rsa_key` – privatni RSA ključ servera za SSH protokol verzije 2
- `/etc/ssh/ssh_host_rsa_key.pub` – javni RSA ključ servera za SSH protokol verzije 2

U korisnikovom `~/.ssh` direktorijumu mogu se naći i:

- `~/.ssh/authorized_keys` – lista javnih ključeva korisnika koji imaju pristup korisnikovom nalogu
- `~/.ssh/known_hosts` – lista DSA javnih ključeva servera kojima je korisnik pristupao

U istom direktorijumu mogu se naći i kombinacije javnih i privatnih ključeva (DSA i RSA) koje je korisnik sam kreirao. Tih ključeva može biti više i oni služe da

implementiraju autentifikaciju preko PKI, što je sigurniji način autentifikacije od zadavanja korisničke lozinke. Podrazumevana imena fajlova su: **id_dsa** (privatni DSA korisnikov ključ), **id_dsa.pub** (javni DSA korisnikov ključ), **id_rsa** (privatni RSA korisnikov ključ), **id_rsa.pub** (javni RSA korisnikov ključ).

Startovanje OpenSSH servisa

OpenSSH servis možete startovati komandom:

```
# service sshd start
```

a zaustaviti sa:

```
# service sshd stop
```

Ako želite da se OpenSSH servis startuje automatski prilikom startovanja sistema, zadajte komandu:

```
# chkconfig sshd on
```

(RedHat i izvedene distribucije), ili

```
# update-rc.d sshd defaults
```

na Debian izvedenim distribucijama.

Prilikom prvog startovanja OpenSSH servisa, kreiraće se odgovarajući DSA i RSA setovi ključeva. Ovo znači da, prilikom reinstalacije servera, dobijate novi set ključeva, ukoliko niste sačuvali stari.

Ukoliko ste dobili novi set ključeva za server na koji ste se već prethodno kačili ssh komandom, ssh će javiti da su serverski ključevi neodgovarajući:

Upotreba autentifikacije bazirane na PKI

Standardna je praksa da dodatno obezbedite server tako što ćete onemogućiti ssh autentifikaciju preko korisničkih lozinki, a omogućiti autentifikaciju preko RSA i DSA ključeva.

UPOZORENJE: Da ne biste došli u situaciju da ne možete da se ulogujete preko mreže

na sopstveni server, uvek prvo postavite autentifikaciju preko ključeva na sve naloge na koje želite da se logujete (i preko kojih možete doći do root privilegija) i to detaljno PROVERITE!!!

Uključite autentifikaciju preko PKI postavljanjem direktiva:

```
RSAAuthentication yes  
PubkeyAuthentication yes
```

u /etc/ssh/sshd_config i restartuje servis.

Zatim podesite ssh autentifikaciju preko ključeva na svim nalozima na koje želite da se logujete (vi i ostali korisnici), što je prikazano u sekciji 'Administriranje udaljenih servera'.

Nakon što ste proverili da možete da se autentifikujete preko ključeva na sve naloge na koje želite, ponovo izmenite fajl /etc/ssh/sshd_config i postavite direktivu:

```
PasswordAuthentication no
```

i ponovo restartujte OpenSSH servis.

Property of Admin Training Center

Instaliranje CentOS 6 servera

Napomena

Ova tema će biti obradena kroz pokaznu vežbu

Property of Admin Training Center

Property of Admin Training Center

Razumevanje boot procesa

Uvod

Standardni proces butovanja PC servera se izvodi po sledećoj sekvenци:

- butovanje PC BIOS-a
- pokretanje instaliranog but loudera
- pokretanje Linux kornela
- kačenje / fajl sistema i pokretanje init procesa
- pokretanje programa prilikom butovanja sistema
- pokretanje programa u zadatom ranlevelu

Nakon izvršavanja poslednjeg koraka, sistem je spreman za interaktivan rad korisnika koji mogu da se uloguju na sistem.

Butovanje PC BIOS-a

Prilikom startovanja računara počinje se sa izvršavanjem programskog koda smeštenog u fleš ROM memoriji računara. Namena ovog programskog koda je da detektuje i inicijalizuje hardver računara i izvrši osnovna testiranja pravnosti. PC BIOS zatim postavlja korisnički zadate parametre i rekonfiguriše sistem u skladu sa njima. Jedan od parametara je i definisanje jedinice sekundarne memorije (hard disk, CD/DVD uređaj i sl.) sa kojeg će početi da se butuje sam operativni sistem.

Ovaj korak je zajednički za bilo koji operativni sistem instaliran na datom računaru.

Pokretanje instaliranog but loudera

Poslednji korak koji PC BIOS izvršava je startovanje rutine za učitavanje operativnog sistema – but loudera (*boot loader*). Pošto je PC BIOS veoma jednostavan deo softvera (sa stanovišta kornela), on pokušava da pronađe but louder na tačno zadatim mestima na butabilnom medijumu. Ukoliko je ovaj butabilni medijum hard disk, but louder bi trebalo da se nalazi u prvom sektoru hard diska, tj. u prvih 512 bajtova. Ova lokacija se kod PC računara naziva 'master but rekord' (*master boot record – MBR*). Ukoliko se but louder ne nalazi u MBR-u, PC BIOS proverava početak primarne particije diska koja je proglašena butabilnom. U slučaju da ni tamo ne nađe but louder, PC BIOS prijavljuje grešku. I ovaj korak je zajednički za sve operativne sisteme.

Postoje različiti but louderi koji se razlikuju u karakteristikama i mogućnostima za

butovanje različitih operativnih sistema. U najvećem broju slučajeva, moderni but louderi imaju karakteristike koje ne mogu biti isprogramirane u 512 bajtova, tako da se u početnom sektoru nalazi samo početni deo but loudera koji učitava ostatak but loudera.

Osnovni zadatak but loudera je učitavanje operativnog sistema u RAM i njegovo pokretanje.

Konfigurisanje GRUB but loudera

GRUB (*Grand Unified Boot Loader*) je produkt GNU projekta za butovanje njihovih Linux i Hurd kernela. U pitanju je napredan but louder koji omogućava izbor operativnog sistema/kernela koji se butuje preko but menija, kao i direktno zadavanje parametara prilikom butovanja. GRUB zna da pristupa sopstvenim i pozajmljenim fajl sistemima kod Linuxa a može butovati i strane operativne sisteme, kao što je Microsoft Windows.

Konfiguracioni fajl je obično /boot/grub/menu.lst ili /boot/grub/grub.conf (obično je jedan simbolički link na drugi). Konfiguracioni fajl je obično kratak i sastoji se iz bloka globalnih parametara i specifikacija stavki u but meniju, zajedno sa odgovarajućim komandnim linijama.

Specifična stvar vezana za GRUB but louder je način označavanja disk particija koji se razlikuje od načina na koji to čini Linux. Naime ovde se specifikacija diska zadaje u obliku '(hdX)', gde je X redni broj diska, koji počinje od 0. Oznaka particije je u obliku '(hdX,Y)', gde je X redni broj diska a Y redni broj particije, počevši takođe od 0.

Globalni parametri se obično svode na tri parametra a to su:

```
default X
```

koji zadaje podrazumevanu stavku koja se butuje ako korisnik ne promeni meni stavku. X označava redni broj stavke, počevši od 0.

```
timeout X
```

Ovim se definiše vreme u sekundama koje but louder čeka sa menijem pre nego što automatski počne butovanje podrazumevane stavke.

```
splashimage=putanja_do_pozadinske_slike
```

Ovim parametrom se zadaje putanja do slike koja se pokazuje u pozadini ekrana sa menijima.

Za svaku stavku menija potrebno je prvo zadati naziv koji se pojavljuje u meniju, parametrom:

```
title Naslov
```

Zatim je potrebno zadati parametar

```
root particija
```

koji označava fizičku particiju gde se nalazi kernelski binarni fajl, a to mora biti primarna particija. Ovaj parametar je moguće izbeći ali u tom slučaju specifikacija particije se navodi kao deo putanje do kernel fajla. Za butovanje Linuxa je potrebno zadati komandnu liniju za butovanje kernela, koja je oblika

```
kernel kernel_fajl opcije
```

gde su

kernel_fajl absolutna putanja do kernel fajla na fizičkoj particiji sa koje se butuje
opcije parametri kernela

Primer jednog konfiguracionog fajla GRUB but loudera je dat u nastavku:

```
default=0
timeout=30
splashimage=(hd0,0)/boot/grub/splash.xpm.gz

title CentOS (2.6.32-431.el6.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-2.6.32-431.el6.x86_64 ro root=/dev/vda1
initrd /boot/initramfs-2.6.32-431.el6.x86_64.img

title CentOS (2.6.32-358.6.2.el6.x86_64)
root (hd0,0)
kernel /boot/vmlinuz-2.6.32-358.6.2.el6.x86_64 ro root=/dev/vda1
initrd /boot/initramfs-2.6.32-358.6.2.el6.x86_64.img
```

Pokretanje Linux kernela

Prilikom butovanja Linux operativnog sistema, but louder treba na butabilnom medijumu locira binarni fajl koji sadrži kernel, učita ga u RAM i predaje mu kontrolu. Različiti but louderi odradjuju ovaj posao na različite načine: neki, jednostavniji, moraju da imaju podatak gde se tačno fizički na disku nalaze sektori u kojima se nalazi kernel i u kom redosledu su ti sektori i onda učitavaju kernel sektor po sektor i na kraju predaju kontrolu izvršavanja programa, tj. pokreću kernel; drugi, složeniji, but louderi znaju da čitaju fajl sistem i mogu da pronađu kernelski fajl i tako ga učitaju u RAM i predaju mu kontrolu izvršavanja.

U svakom slučaju, kernel se startuje kao neki program, kome se mogu proslediti parametri u obliku standardne komandne linije. Jedan od obaveznih parametara je naziv particije diska na kojoj se nalazi / fajlsistem. Ostali parametri se takođe mogu nавести i oni definišu izvršavanje kernela. Parametri navedeni u komandnoj liniji kernela ne

moraju da se odnose na kernel – parametri koje kernel ne prepozna će biti tiho preskočeni, a celokupna komandna linija kojom je 'startovan' kernel se, po podizanju kernela mogu videti kroz `/proc` pseudo-fajlsistem, čitanjem virtuelnog fajla `/proc/cmdline`. Na ovaj način prilikom startovanja kernela se mogu zadati neki parametri koji i nisu namenjeni kernelu već nekom drugom programu koji se pokreće automatski prilikom podizanja sistema.

Zajedno sa učitavanjem kernela se može učitati, ukoliko postoji, i sadržaj tzv. ramdiska – sadržaj ovog diska je u bitske slike fajl sistema ramdiska i nalazi se u zasebnom fajlu.

Po učitavanju u RAM kernel počinje da se izvršava. Prvo se izvršavaju rutine koje detektuju hardver i inicijalizuju ga. Zatim se inicijalizuju ostale kernelske strukture i instaliraju različite dodatne rutine koje se sve nalaze ugrađene u kernel. Na kraju, kernel treba da zakači / fajlsistem i sa njega učita i pokrene inicijalni proces.

Kačenje / fajlsistema i pokretanje init procesa

Kernel saznaće na kojoj particiji se nalazi / fajlsistem iz sopstvene komandne linije. Ovaj fajlsistem može biti na disku ili se privremeno može koristiti fajlsistem na ramdisku. Ramdisk sa privremenim / fajlsistemom se koristi da bi se eventualno učitali dodatni moduli koji su potrebni da bi se sistem dalje pokrenuo, prevashodno da bi se moglo pristupiti uređaju na kojem se nalazi stvarni / fajlsistem. U ovom slučaju se takođe pokreće i inicijalni proces koji je najčešće neki komandni skript.

U slučaju da je odgovarajući drajver već ugrađen u kernelski fajl, kernelu nije potrebno da učitava ramdisk i fajlsistem na njemu već se direktno kači stvarni / fajlsistem, učitava i pokreće inicijalni proces.

Inicijalni proces se može zadati u komandnoj liniji kernela parametrom `init=/putanja/do/programa`. Ukoliko on nije zadat na ovakav način, kao što je slučaj u svakodnevnom radu, onda se kao inicijalni program pokreće izvršni fajl `/sbin/init`.

U svakom slučaju inicijalni program mora biti fizički prisutan na / fajlsistemu, a ne na nekom od ostalih fajlsistema koji u ovom trenutku još uvek nisu zakačeni u logičko stablo direktorijuma.

Init proces, čiji se kod nalazi u izvršnom fajlu `/sbin/init` je standardni inicijalni proces koji je zadužen za dalje podizanje sistema i njegovo dovođenje u operativno stanje. Ovaj proces učitava svoj konfiguracioni fajl `/etc/inittab` u kojem se nalaze specifikacije za pokretanje ostalih programa koji se automatski pokreću prilikom podizanja sistema.

Fajl `/etc/inittab` je tekstualni fajl u kojem se linije koje su prazne i linije koje

počinju sa znakom '#' zanemariju. Ostale linije su direktive programu init i imaju generalni format:

```
identifikator:ranleveli:akcija:komanda
```

gde su:

identifikator – jedinstveni identifikator direktive dužine 1-4 znaka

ranleveli – spisak ranlevela u kojima se akcija specificirana direktivom izvršava. Ranleveli su 'nivoi' izvršavanja sistema i označeni su ciframa 0-6. Standardno značenje pojedinih ranlevela je sledeće:

0 – zaustavljanje sistema

1 ili S – jednokorisnički mod

2 – višekorisnički mod bez mrežne podrške

3 – puni višekorisnički mod bez grafičkog moda

4 – ne koristi se

5 – puni višekorisnički mod sa grafičkim modom

6 – restartovanje sistema

Više ranlevela u spisku se navodi kao niz cifara bez belina.

akcija – specifikacija akcije koja će biti izvršena. Akcija može biti:

respawn – program će biti automatski restartovan kada završi sa izvršavanjem

wait – program će biti izvršen jednom prilikom ulaska u zadati ranlevel i init će čekati dok se program ne završi pre nego što nastavi sa radom (sekvencijalno pokretanje).

once – program će biti izvršen jednom prilikom ulaska u zadati ranlevel. Init proces neće čekati da se ovaj program završi već će odmah nastaviti sa radom (paralelno pokretanje).

boot – program će biti izvršen prilikom butovanja sistema. Polje 'ranleveli' se ignoriše.

bootwait – program će biti izvršen prilikom butovanja sistema ali će init čekati dok se program ne završi. Polje 'ranleveli' se ignoriše.

off – akcija se ne izvršava

initdefault – akcija kojom se specificira u koji od ranlevela će se ući po butovanju sistema. U polju 'ranleveli' može biti naveden tačno jedan ranlevel. Polje 'komanda' se ignoriše.

sysinit – program će biti izvršen prilikom butovanja sistema, pre bilo koje 'boot' ili 'bootwait' akcije. Polje 'ranleveli' se ignoriše.

powerwait – program će biti izvršen kada se signalizira prekid napajanja. Init čeka da se ovaj program izvrši.

powerfail – program će biti izvršen kada se signalizira prekid napajanja, a init proces neće čekati da se program završi.

powerfailnow – program će biti izvršen kada se init procesu bude signalizirano da je baterija UPS-a skoro ispražnjena i da napajanje još nije uspostavljeno.

resume – program će biti izvršen kada init procesu bude signalizirano da je napajanje normalizovano.

kbrequest – program će biti izvršen ako se na konzoli sistema pritisne odgovarajuća kombinacija tastera.

ctrlaltdel – program (obično 'shutdown' komanda) koji se izvršava kada se na konzoli sistema pritisne kombinacija tastera CTRL-ALT-DEL.

komanda – program i njegovi parametri koji treba da budu pokrenuti da bi se akcija izvršila.

Pokretanje programa prilikom butovanja sistema

Primer konfiguracionog fajla /etc/inittab se nalazi u nastavku i na njemu ćemo objasniti pojedine direktive:

```
# The default runlevel is defined here  
id:3:initdefault:
```

Inicijalni ranlevel je ranlevel 3.

```
# First script to be executed,  
si::bootwait:/etc/init.d/boot
```

Skript koji se izvršava prilikom butovanja sistema. On je zadužen za kačenje lokalnih fajl sistema, učitavanje zadatih modula i sl.

Izvršavanje programa u zadatom ranlevelu

Specifikacija komandi koje će se izvršiti kada se uđe u pojedini ranlevel. Kod skoro svih distribucija zadati program je komandni skript koji, između ostalog, pokreće skriptove za startovanje pojedinih servisa. Ovi skriptovi se nalaze u direktorijumima /etc/rc.d/rcX.d gde je *X* oznaka ranlevela. Svaki od skriptova se napisan tako da kao argument uzima parametre 'start', 'stop' ili 'restart'. Skript pokrenut sa parametrom 'start' pokreće dati servis. Isti skript sa parametrom 'stop' zaustavlja servis. Parametar

'restart' služi za restartovanje servisa.

Svi navedeni skriptovi imaju imena oblika *SXYservis*, odnosno *KXYservis*, gde *S* na početku imena označava da prilikom ulaska u dati ranlevel servis treba pokrenuti sa parametrom 'start', dok *K* na početku imena označava servis koji treba pokrenuti sa parametrom 'stop' (tj. u datom ranlevelu navedeni servis treba ugasiti). *XY* u imenu označava dvocifrenu oznaku koja omogućava da servisi budu startovani po zadatom redosledu, dok je *servis* ime servisa.

```
# /etc/init.d/rc takes care of runlevel handling
#
# runlevel 0 is System halt (Do not use this for initdefault!)
# runlevel 1 is Single user mode
# runlevel 2 is Local multiuser without remote network (e.g. NFS)
# runlevel 3 is Full multiuser with network
# runlevel 4 is Not used
# runlevel 5 is Full multiuser with network and xdm
# runlevel 6 is System reboot (Do not use this for initdefault!)
#
10:0:wait:/etc/init.d/rc 0
11:1:wait:/etc/init.d/rc 1
12:2:wait:/etc/init.d/rc 2
13:3:wait:/etc/init.d/rc 3
#14:4:wait:/etc/init.d/rc 4
15:5:wait:/etc/init.d/rc 5
16:6:wait:/etc/init.d/rc 6

# what to do in single-user mode
ls:S:wait:/etc/init.d/rc S
~~:S:respawn:/sbin/sulogin
```

Pošto različiti ranleveli pokreću/zaustavljaju iste servise, to se u */etc/rc.d/rcX.d* direktorijumima umesto stvarnih skriptova nalaze simbolički linkovi gore navedenog formata imena koji ukazuju na stvarne skriptove koji se nalaze pod imenom *servis* u direktorijumu */etc/init.d*.

Programi koji se izvršavaju u posebnim slučajevima

Posebni slučajevi mogu da se dogode u bilo kom 'standardnom' ranlevelu tako da se za takve akcije specificiraju određeni programi. Ovi posebni slučajevi uključuju pritiskanje CTRL-ALT-DEL kombinaciju tastera (obično da bi restartovali sistem), odnosno situacije kada dolazi do poremećaja napajanja.

```
# what to do when CTRL-ALT-DEL is pressed
ca::ctrlaltdel:/sbin/shutdown -r -t 4 now

# special keyboard request (Alt-UpArrow)
# look into the kbd-0.90 docs for this
Kb::kbrequest:/bin/echo "Keyboard Request -- edit /etc/inittab to let
this work."

# what to do when power fails/returns
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
#pn::powerfail:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop

# for ARGO UPS
sh:12345:powerfail:/sbin/shutdown -h now THE POWER IS FAILING
```

Pokretanje programa koji upravljaju terminalima

Logovanje na virtuelnim terminalima na konzoli, odnosno terminalima povezanim preko serijske veze, odnosno modema je kontrolisano posebnim programima nazvanim 'getty' programi. Postoji više varijacija ovih programa sa različitim karakteristikama – jednostavniji se koriste isključivo za kontrolu virtuelnih terminala na konzoli, dok se neki složeniji koriste za kontrolu serijskih linija i modema i umeju prepoznati i dolazeći glasovni poziv ili faks i adekvatno reagovati.

'getty' program prilikom startovanja preuzima posao nadgledanja zadate terminalske linije. Ukoliko detektuje signal na liniji, getty će startovati program 'login' na datom terminalu. Posao *login* programa je da izvrši autentifikaciju korisnika, i ako je ona bila uspešna, startuje korisnikov podrazumevani komandni interpreter. Pošto se ovo sve dešava u istom procesu, isti će biti završen kada se korisnik izloguje i time okonča izvršavanje komandnog interpretera. Tada će *init* proces detektovati da je proces koji se inicijalno počeo kao 'getty' završio sa radom i zbog 'respawn' akcije pokrenuti novi getty program.

```

# getty-programs for the normal runlevels
# <id>:<runlevels>:<action>:<process>
# The "id" field MUST be the same as the last
# characters of the device (after "tty").
1:2345:respawn:/sbin/mingetty --noclear tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
#
#S0:12345:respawn:/sbin/agetty -L 9600 ttyS0 vt102

#
# Note: Do not use tty7 in runlevel 3, this virtual line
# is occupied by the programm xdm.
#

# This is for the package xdm, after installing and
# and configuration you should remove the comment character
# from the following line:
#7:3:respawn:+/etc/init.d/rx tty7

# modem getty.
# m0:235:respawn:/usr/sbin/mgetty -s 38400 modem

# fax getty (hylafax)
# m0:35:respawn:/usr/lib/fax/faxgetty /dev/modem

# vbox (voice box) getty
# I6:35:respawn:/usr/sbin/vboxgetty -d /dev/ttYI6
# I7:35:respawn:/usr/sbin/vboxgetty -d /dev/ttYI7

# end of /etc/inittab

```

Menjanje tekućeg ranlevela

Pošto je *init* inicijalni proces čiji je PID 1, prestanak rada ovog procesa nije dozvoljen i pokušaj da se ovaj proces prekine dovešće do fatalne greške na sistemu. No, program *init* ima i drugu namenu – pozivanjem ovog programa iz komandne linije i zadavanjem odgovarajućeg parametra moguće je promeniti tekući ranlevel u neki drugi, pri čemu će se prekinuti izvršavanje onih pokrenutih servisa koji nisu predviđeni za izvršavanje u novom ranlevelu, a pokrenuti izvršavanje onih servisa koji se ne izvršavaju u prethodnom ranlevelu a izvršavaju u novom. Dakle, format komande je:

```
# init ranlevel
```

Pored navedenih ranlevela moguće je zadati i parametar '*q*', odnosno '*Q*' što znači da treba ponovo izvršiti postojeći ranlevel, što efektivno znači da treba ponovo pregledati

dati direktorijum sa skriptovima koji se pokreću i zaustavljaju u datom ranlevelu i startovati naknadno dodate skriptove.



Admin Training Center

L2-2 Administracija Linux servera

Veselin Mijušković, Marko Uskoković, Ljubiša Radivojević

Copyright © 2014 Veselin Mijušković, Marko Uskoković, Ljubiša Radivojević

OBJAVIO ADMIN TRAINING CENTER

www.atc.rs

Licencirano po Creative Commons Attribution-NonCommercial 3.0 Unported License (the "License"). You may not use this file except in compliance with the License. You may obtain a copy of the License at <http://creativecommons.org/licenses/by-nc/3.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Prvo izdanje, 2014

Sadržaj

Uvod	7
Tipografske konvencije	7
Modul 1: Log fajlovi	9
1.1 Rsyslog	9
1.1.1 Konfigurisanje rsyslog-a	9
1.1.2 Globalne direktive	9
1.1.3 Moduli	10
1.1.4 Pravila	10
1.1.5 Filteri	11
1.1.6 Akcije	14
1.1.7 Šabloni	15
1.2 Rotiranje logova	16
1.3 Vežbe	17
Modul 2: Init ramdisk, kernel i moduli	19
2.1 Initramfs	19
2.2 Konfigurisanje GRUB boot loader-a	20
2.3 Rad sa kernel modulima	21
2.3.1 Prikaz trenutno učitanih modula	22
2.3.2 Prikaz informacija o modulu	23
2.3.3 Učitavanje modula	24
2.3.4 Uklanjanje modula iz kernela	24
2.3.5 Postavljanje parametara modula	25
2.4 Vežbe	26
Modul 3: Proširena kontrola pristupa fajlovima	27
3.1 Uvod	27
3.2 Omogućavanje ACL	27
3.3 Postavljanje prava preko ACL	28
3.4 Pregledanje ACL	28

3.5	Uklanjanje prava iz ACL i same pristupne ACL	29
3.6	Određivanje pristupa fajlu preko ACL	29
3.7	Vežbe	31
Modul 4: Linux firewall		33
4.1	OSI model	33
4.1.1	Slojevi OSI modela	33
4.2	TCP/IP set protokola	34
4.2.1	IP protokol	34
4.2.2	TCP protokol	34
4.2.3	UDP protokol	35
4.3	Namena firewall-a	36
4.4	Tipovi firewalla	36
4.5	Netfilter princip rada	38
4.6	Iptables userspace program	38
4.7	Iptables komande	39
4.8	Iptables provere pravila (rule matches)	41
4.8.1	Generičke provere	41
4.8.2	Implicitne provere	42
4.8.3	Eksplisitne provere	44
4.8.4	Provera kernel oznaka	45
4.8.5	Multiport provera	45
4.8.6	Provera vlasnika	46
4.8.7	Provera stanja paketa	46
4.8.8	Provera polja TOS	46
4.8.9	Provera polja TTL	47
4.8.10	Provera ispravnosti paketa	47
4.9	Iptables AKCIJE (targets/jumps)	47
4.9.1	Akcija ACCEPT	47
4.9.2	Akcija DROP	47
4.9.3	Akcija REJECT	47
4.9.4	Aktivnost LOG	48
4.9.5	Aktivnost MARK	48
4.9.6	Aktivnost TOS	49
4.9.7	Aktivnost TTL	49
4.9.8	Aktivnost REDIRECT	49
4.9.9	Aktivnost MIRROR	50
4.9.10	Aktivnost QUEUE	50
4.9.11	Aktivnost RETURN	50
4.10	Postupak Masquerading-a	50
4.10.1	Aktivnost DNAT	50
4.10.2	Aktivnost SNAT	51
4.10.3	Aktivnost MASQUERADE	51
4.11	Implementacija firewall-a na RedHat/CentOS distribucijama	51
4.12	Vežbe	51

Modul 5: Kontrola pristupa u Linuxu	53
5.1 SELinux koncepti	54
5.2 SELinux sigurnosni kontekst	54
5.3 Tranzicija domena	55
5.4 Polise i konfiguracioni fajlovi	56
5.4.1 Strict polisa	56
5.4.2 Targeted polisa	56
5.5 SELinux alatke	57
5.5.1 Paket coreutils	57
5.5.2 Paket libselinux-utils	57
5.5.3 Paket policycoreutils	57
5.5.4 Paket checkpolicy	57
5.5.5 Paket policycoreutils-python	57
5.5.6 Paketi setroubleshoot i setroubleshoot-server	58
5.6 Označavanje fajlova	58
5.7 Označavanje fajlova	58
5.8 Log fajlovi	58
5.8.1 Nepravilno označeni fajlovi	59
5.9 Prilagodavanje polisa	59
5.10 Dodatni moduli polise	60
5.11 Vežbe	60
Modul 6: Nenadgledana instalacija Linuxa	63
6.1 Pozivanje nenadgledane instalacije	63
6.2 Kickstart konfiguracioni fajl	64
6.3 Komande Kickstart-a	65
6.3.1 Komande koje definišu osnovne parametre sistema	65
6.3.2 Komande koje definišu autentifikaciju	65
6.3.3 Komande za hardver	67
6.3.4 Komande koje definišu tip instalacije	67
6.3.5 Komande koje definišu način instalacije	68
6.3.6 Komande koje definišu particionisanje diska	68
6.3.7 Komande koje se tiču bezbednosti sistema	71
6.3.8 Komande za mrežni pristup	72
6.3.9 Komande koje definišu praćenje instalacije	72
6.3.10 Komande koje definišu pakete koji se instaliraju	73
6.3.11 Opcije koje definišu konfiguraciju X-a	73
6.3.12 Komande za konfiguraciju sistema nakon instalacije	74
6.3.13 Komande koje definišu kako se završava instalacija	74
6.3.14 Ostale komande	75
6.3.15 Dodatne sekcije kickstart fajla	75
6.4 Vežbe	77
Indeks	79

Property of Admin Training Center

Uvod

Skripta je podeljena na poglavlja, a poglavlja na sekcije. Unapred skrećemo pažnju polaznicima da je ova skripta samo deo dokumentacije koju oni treba da koriste. Polaznicima se savetuje da pročitaju man i help strane za svaku komandu, kao i da potraže na Internetu dodatne informacije i načine kako da iste koristite.

Tipografske konvencije

Radi lakšeg snalaženja u tekstu, koristili smo neke tipografske konvencije na koje vam ovde skrećemo pažnju:

- ukoliko se uvodi neki značajan pojam, on će u prvom pomenu biti isписан **proporcionalnim bold tekstom**;
- boldovano su prikazane i neke značajne tvrdnje na koje treba obratiti pažnju u tekstu;
- *proporcionalnim italicom* su napisane reči na stranom jeziku, najčešće engleskom;
- nazivi fajlova u tekstu su ispisani *neproporcionalnim fontom*

Oblik neke komande prikazan je na sledeći način:

```
# komanda [opcije] argument...
```

Deo koji je napisan uspravnim fontom se unosi kako je napisan. Opcioni deo, koji se može izostaviti je naveden uvek u uglastim zagradama (same zgrade se ne unose). Ukoliko je tekst isписан *italicom* to znači da je u pitanju neki opšti naziv i umesto njega treba uneti neku stvarnu vrednost. Ukoliko iza teksta stoje tri tačke '...' to znači da se taj deo može ponavljati.

Deo teksta koji se odnosi na direktni unos korisnika i ispis računara, kao i sadržaji fajlova i sl. će biti prikazani u zasebnom bloku:

```
$ ls -F  
myscript.sh*  Vezbe/
```

Boldovanim tekstom je prikazano ono što polaznik treba da unese onako kako je napisano u skripti. Regularnim tekstom je prikazan ispis programa koji ne treba unositi.

Gde god smo mislili da nešto posebno treba naglasiti, to smo naveli na kao napomenu ili upozorenje na sledeći način:

Napomena:

Hard linkovi ne mogu biti kreirani za direktorijume, već samo za regularne fajlove!

Upozorenje!

Uvek koristite 'modprobe' za učitavanje modula jer ćete tako učitati i zavisne module!

Na kraju, ukoliko se zadaje deo koda (npr. skripte komandnog interpretera bash), onda će to izgledati ovako:

```
1 #!/bin/bash
2 #
3 # helloworld.sh - standardni Hello, World skript
4 #
5
6 echo "Hello, World!"
```

Modul 1: Log fajlovi

Log fajlovi su fajlovi koji sadrže poruke o sistemu, uključujući kernel, servise i aplikacije koje se na njemu izvršavaju.

Log fajlovi su vrlo korisni u slučajevima otkrivanja grešaka na sistemu i dokumentovanja šta je i kada urađeno.

1.1 Rsyslog

Sistemski log fajlovi su kontrolisani od strane 'rsyslogd' dimona. Njegov konfiguracioni fajl `/etc/rsyslog.conf` sadrži, između ostalog i listu log fajlova koju kontroliše ovaj program.

Uloga programa 'rsyslog' je da prihvati log poruke iz različitih izvora preko svojih ulaznih modula, prosledi poruke odgovarajućim setovima pravila, gde se pravila uslovno primenjuju. Svako pravilo se sastoji iz filtera i akcije. Ukoliko pravilo odgovara, poruka se prosleđuje akciji koja uradi nešto sa porukom, bilo da je zapiše u lokalni fajl ili da je prosledi na udaljeni rsyslog/syslog program.

1.1.1 Konfigurisanje rsyslog-a

Konfiguracioni fajl za program 'rsyslog' je `/etc/rsyslog.conf`. Sastoji se iz:

- globalnih direktiva
- pravila
- komentara (prazne linije i tekst od znaka '#' do kraja linije)

1.1.2 Globalne direktive

Globalne direktive definišu opcije koje se odnose na sam program. Najčešće su u pitanju vrednosti neke predefinisane varijable koja utiče na ponašanje samog programa ili pravila koje je u nastavku fajla. Globalne direktive počinju sa znakom '\$'. U jednom redu može biti samo jedna globalna direktiva.

Primer direktive je:

```
$MainMsgQueueSize 50000
```

Ova direktiva redefiniše veličinu reda sa porukama sa podrazumevane vrednosti (10.000) na novu (50.000).

Jedna ista globalna direktiva može se pojaviti u konfiguracionom fajlu više puta. Svako novo pojavljivanje redefiniše vrednost navedene globalne promenljive.

Kompletan spisak svih globalnih direktiva je dostupan u fajlu:

```
/usr/share/doc/rsyslog-verzija/rsyslog_conf_global.html
```

1.1.3 Moduli

Moduli programa 'rsyslog' obezbeđuju dinamičku funkcionalnost. Većina modula obezbeđuje dodatne načine unosa poruka (ulazni moduli) ili izlaza poruka (izlazni moduli). Moduli mogu imati svoje konfiguracione direktive koje postaju dostupne pošto se modul učita.

Učitavanje modula se vrši direktivom:

```
$ModLoad modul
```

Na primer, ako želimo da učitamo modul 'imfile' koji konvertuje tekstualne fajlove u syslog poruke, to ćemo uraditi na sledeći način:

```
$ModLoad imfile
```

Standardna instalacija programa 'rsyslog' nudi mnoštvo modula koji su grupisani u sledeće kategorije:

ulazni moduli Ovi moduli skupljaju poruke iz različitih izvora. Njihova imena uvek počinju prefiksom 'im'.

izlazni moduli Ovi moduli omogućavaju da se poruke smeštaju na različite lokacije, kao što je slanje poruka preko mreže, smeštanje poruka u različite baze ili enkriptovanje poruka. Imena ovih modula uvek počinju prefiksom 'om'.

filterski moduli Ovi moduli implementiraju različite metode filtriranja poruka. Imena im počinju prefiksom 'fm'.

parserski moduli Parserski moduli omogućavaju različite načine parsiranja ulaznih poruka. Imena im počinju sa 'pm'.

moduli koji modifikuju poruke Ovi moduli implementiraju različite metode modifikovanja ulaznih poruka.

moduli generatori stringova Ovi moduli generišu tekstualne stringove na osnovu poruka koje obrađuju. Tesno su povezani sa sistemom šablonu koje nudi 'rsyslog'. Njihova imena počinju sa 'sm'.

bibliotečki moduli Ovi moduli implementiraju zajedničke funkcije koje koriste ostali moduli. Ovi moduli se automatski učitavaju kada su potrebni i ne mogu biti konfigurisani od strane korisnika.

Spisak dostupnih modula možete dobiti u fajlu:

```
/usr/share/doc/rsyslog-verzija/rsyslog_conf_modules.html
```

1.1.4 Pravila

Pravila su grupisana u setove. Svako pravilo se sastoji iz dva dela:

- filter
- akcija

Pravila se definišu u obliku:

filter akcija

1.1.5 Filteri

Različiti filterski moduli implementiraju različite načine filtriranja poruka.

Objekat/prioritet filteri

Najpoznatiji način filtriranja poruka, koji implementiraju i drugi syslog programi. Filter je oblika:

objekat.prioritet

gde je 'objekat' podsistem koji generiše poruku i može biti jedan od:

- auth, authpriv
- cron
- daemon
- kern
- lpr
- mail
- news
- syslog
- user
- uucp
- local0...;local7

a 'prioritet' nivo ozbiljnosti poruke koji može biti (od najnižeg ka najvišem):

- debug
- info
- notice
- warning
- err
- crit
- alert
- emerg

Navedeni prioritet znači da će filter prihvati sve poruke koje su tog ili višeg prioriteta. Ako ispred specifikacije prioriteta stavimo znak '=', onda će biti selektovane samo poruke tog prioriteta. Znak '!' označava da će poruke navedenog prioriteta biti ignorisane, a sve ostale selektovane.

Znak '*' može stajati umesto objekta i/ili prioriteta i označava sve objekte, odnosno prioritete. Više različitih objekata ili prioriteta u filteru se navodi tako što se navode razdvojeni zarezom (bez razmaka). Više filtera se može navesti u istom redu razdvojenih sa ','.

Primeri filtera su dati u sledećem listingu:

```

kern.*    # Selects all kernel syslog messages with any priority
mail.crit # Selects all mail syslog messages with priority crit and higher.
cron.!info,!debug   # Selects all cron syslog messages except those
                    # with the info or debug priority.

```

Filteri bazirani na osobinama

Filteri bazirani na osobinama omogućavaju da se poruke filtriraju prema bilo kojoj njihovoj osobini. Trenutno definisane osobine su prikazane u tabeli 1.1.

Filteri bazirani na osobinama se zadaju u obliku:

```
:osobina, [!]operacija-poređenja, "string"
```

gde su:

osobina Osobina koja se poredi (npr. 'timegenerated' ili 'hostname')

! Opciona negacija operacije poređenja

operacija-poređenja Jedna od operacija poređenja iz tabele 1.2

string Argument sa kojim se osobina poredi

Sledeći primeri prikazuju filtere bazirane na osobinama:

```

# The following filter selects syslog messages which
# contain the string error in their message text:
:msg, contains, "error"

# The following filter selects syslog messages received
# from the hostname host1:
:hostname, isEqual, "host1"

# The following filter selects syslog messages which do
# not contain any mention of the words fatal and error
# with any or no text between them (for example, fatal
# lib error):
:msg, !regex, "fatal .* error"

```

Filteri bazirani na izrazima

Filteri bazirani na izrazima selektuju poruke prema definisanim aritmetičkim, logičkim ili string operacijama. Izrazi koriste skripting jezik definisan u samom programu 'rsyslog'. Sintaksa ovog jezika je definisana u /usr/share/doc/rsyslog-verzija/rscontrol_abnf.html. Format ovog tipa filtera je:

```
if izraz then akcija
```

gde su:

izraz Izraz koji se izračunava, kao npr:

```
$msg startswith 'DEVNAME' or $syslogfacility-text == 'local0'
```

akcija Akcija koja se izvodi ako je izraz tačan.

U ovim filterima se ne mogu koristiti regularni izrazi.

Filteri bazirani na izrazima moraju biti napisani u jednoj liniji.

Osobina	Opis
msg	Tekst poruke
rawmsg	Tekst poruke kako je primljen sa soketa
hostname, source	Ime hosta koji je generisao poruku
fromhost	Ime hosta koji je poslao poruku
fromhost-ip	IP adresa hosta koji je poslao poruku
syslogtag	TAG poruke (oznaka koji program/PID je generisao poruku)
programname	Ime programa koji je generisao poruku
pri	objekat.prioritet u obliku broja
pri-text	objekat.prioritet u obliku teksta
syslogfacility	objekat u numeričkoj formi
syslogfacility-text	objekat u tekstu formi
syslogseverity, syslogpriority	prioritet poruke u numeričkoj formi
syslogseverity-text, syslogpriority-text	prioritet poruke u tekstu formi
timegenerated	<i>timestamp</i> kada je poruka primljena
timereported, timestamp	<i>timestamp</i> koja se nalazi u samoj poruci
protocol-version	verzija protokola iz IETF drafata <i>draft-ietf-syslog-protocol</i>
structured-data	STRUCTURED-DATA polje iz IETF drafata <i>draft-ietf-syslog-protocol</i>
app-name	APP-NAME polje iz IETF drafata <i>draft-ietf-syslog-protocol</i>
procid	PROCID polje iz IETF drafata <i>draft-ietf-syslog-protocol</i>
msgid	MSGID polje iz IETF drafata <i>draft-ietf-syslog-protocol</i>
inputname	Ime ulaznog modula kroz koji je poruka stigla
\$bom	UTF-8 BOM
\$uptime	<i>uptime</i> u sekundama
\$now	Tekući datum u formatu YYYY-DD-MM
\$year	Tekuća godina (4 cifre)
\$month	Tekući mesec (2 cifre)
\$day	Tekući dan u mesecu (2 cifre)
\$hour	Tekući čas 0-24h (2 cifre)
\$hhour	Tekući polučas (0-29 minuta je uvek 0, 30-59 minuta je uvek 1)
\$qhour	Tekući četvrt-čas (0-3)
\$minute	Tekući minut (2 cifre)
\$myhostname	Ime lokalnog hosta

Tabela 1.1: Spisak osobina

Operacija	Opis
contains	Proverava da li zadata osobina sadrži 'string'
isequal	Proverava da li zadata osobina odgovara 'stringu'
startswith	Proverava da li zadata osobina počinje sa 'stringom'
regex	Primenjuje POSIX osnovni regularni izraz sa sadržajem osobine
ereregex	Primenjuje POSIX prošireni regularni izraz sa sadržajem osobine

Tabela 1.2: Operacije poređenja

BSD blokovi

BSD blokovi su modifikatori filtera koji se odnose na filtere navedene u sledećem redu. Odnose se na programe koji generišu poruke, odnosno na hostove sa kojih poruke dolaze:

!program Uključi sve poruke koje dolaze od programa 'program'.

-program Isključi sve poruke koje dolaze od programa 'program'.

+host Uključi sve poruke koje dolaze sa hosta 'host'.

-host Isključi sve poruke koje dolaze sa hosta 'host'.

Sledeći primer smešta sve poruke koje dolaze od programa 'yum' a prioriteta su 'notice' ili višeg u fajl `/var/log/yum.log`

```
!yum
*.notice    /var/log/yum.log
```

1.1.6 Akcije

Akcije definišu šta će biti urađeno sa porukama koje su prošle kroz filtere. Neke akcije koje možete definisati u pravilima navedene u nastavku.

Snimanje log poruka u log fajlove

Najčešće korišćena akcija je snimanje poruka u neki fajl. U specifikaciji pravila jednostavno za akciju upišite putanju do fajla u koji želite da snimite poruke, kao u primeru gde se sve poruke objekta 'cron' snimaju u fajl `/var/log/cron.log`:

```
cron.* /var/log/cron.log
```

Putanja fajla može biti statička (kao u primeru) ili dinamička, oblika:

?šablon

Takođe, ako ispred putanje dodamo znak '-' izlazni fajl se neće sinhronizovati svaki put kada se upiše poruka. Na kraju, ako putanja ukazuje na neki terminalni fajl ili konzolu, onda će se poruka ispisivati na ekran.

Prosleđivanje poruka na udaljeni server

Poruke mogu da se prosleđuju na udaljeni računar ili da se primaju sa njega. Format akcije koja šalje poruku na udaljeni server je:

```
@[@]{(opcija)}[host:[port]}
```

Prefiks '@' označava da se koristi UDP protokol, a '@@' da se koristi TCP protokol. 'opcija' se piše u zagradama, više njih može se grupisati razdvojene zarezima. 'host' je ime ili IP adresa

udaljenog servera, a 'port' je port na kojem sluša program syslog. Primeri za ovu vrstu akcije su dati u nastavku:

```
*.* @192.168.0.1      # Forwards messages to 192.168.0.1 via the UDP protocol  
.*.* @@example.com:18    # Forwards messages to "example.com" using port  
                           # 18 and the TCP protocol  
.*.* @({z9})[2001::1]  # Compresses messages with zlib (level 9 compression)  
                           # and forwards them to 2001::1 using the UDP protocol
```

Slanje poruke korisniku

Moguće je poruku poslati korisniku. U tom slučaju samo je potrebno na mestu akcije upisati korisničko ime onog kojem šaljemo poruku (ako je ulogovan u tom trenutku). Ako želimo da pošaljemo istu poruku za više korisnika njihova korisnička imena razdvajamo zarezima. Ako umesto korisničkog imena stavimo znak '*' poruka će biti poslata svim korisnicima koji su u tom trenutku ulogovani na sistem.

Prosleđivanje poruke eksternom programu

Poruku je moguće proslediti eksternom programu, kao parametar komande. U tom slučaju format poruke mora biti definisan šablonom (o šablonima videti više u delu 1.1.7). Format ove akcije je:

```
^program;šablon
```

Odbacivanje poruka

Poruke možemo odbaciti/zanemariti tako što na mesto akcije stavimo znak '~'.

Unutar jednog pravila moguće je navesti više akcija koje će se odvijati paralelno. To se radi tako što se prva akcija navodi po pokazanom formatu, a ostale akcije se navode u narednim redovima gde na mestu filtera stoji znak '&' kao u primeru:

```
kern.=crit joe  
&          ^test-program;temp  
&          @192.168.0.1
```

U gornjem primeru sve kritične poruke od strane kernela se šalju korisniku 'joe', primenjuje se šablon 'temp' i zatim se poruka prosleđuje programu 'test-program' i šalje se na udaljeni server 192.168.0.1 u originalnom obliku.

1.1.7 Šabloni

Šabloni definišu oblik poruke koji će biti prosleđen izlaznom modulu. Sam program 'rsyslog' dolazi sa hardkodovanim najčešće korišćenim šablonima, ali korisnik može kreirati svoje. Šabloni moraju biti definisani pre same upotrebe.

Šabloni se definišu na sledeći način:

```
$template ime-šablona, "tekst-šablona", [opcije]
```

gde su:

ime-šablona Ime pod kojim se šablon referencira.

tekst-šablona Sam tekst šablona. Unutar teksta šablona moguće je referencirati osobine poruke na sledeći način:

```
%osobina[:početni-znak:krajnji-znak:opcija]%
```

'početni-znak' i 'krajnji-znak' omogućavaju da 'odsečemo' deo teksta osobine. Opcija modifikuje tekst osobine na neki specifičan način.

opcije Ovo su opcije šablona i ne treba ih mešati sa opcijama osobine. Trenutno definisane opcije šablona su 'sql' i 'stdsql' i one prilagođavaju izlaz šablona tako da produkuju SQL naredbe.

Šabloni se koriste i kod generisanja dinamičkih imena fajlova. Primer jednog takvog šablona i njegova upotreba su dati u sledećem primeru:

```
$template DynamicFile,"/var/log/test_logs/%timegenerated%-test.log"
*.* ?DynamicFile
```

Ovaj, malo nezgrapan primer, će generisati zaseban fajl za svaku primljenu syslog poruku.

1.2 Rotiranje logova

Program 'logrotate' se poziva iz 'cron'-a i prati sve log fajlove za koje je konfigurisan, proveravajući njihove parametre i, ako je to potrebno, kreirajući nove, ispravne log fajlove i pohranjujući stare.

Problem kreiranja novih log fajlova jeste što kod mnogih aplikacija koje kreiraju log fajlove brisanje log fajla u toku izvršavanja aplikacije neće dovesti do uklanjanja njegovog sadržaja, već samo imena pod kojim je taj log fajl zaveden. Aplikacija će i dalje koristiti isti fajl (koji sada nema ime ali i dalje zauzima prostor na disku). Čak i kreiranje fajla sa istim imenom neće dovesti do njegove upotrebe jer taj fajl ima različit i-nod od originalnog, i različit fajl-deskriptor (ako ga aplikacija uopšte i otvorи). Program 'logrotate' zna da se snađe u takvim situacijama (obično je dovoljno poslati odgovarajući signal procesu da bi on otpustio obrisani fajl i otvorio novokreirani sa istim imenom). Program 'logrotate' zna, u zavisnosti od konfiguracije, i da arhivira starije log fajlove (npr. da ih kompresuje) i da obriše one koji više nisu potrebni.

Konfiguracija programa 'logrotate' se nalazi u `/etc/logrotate.conf`. Deo ovog fajla je prikazan na sledećem listingu:

```
# rotate log files weekly
weekly
# keep 4 weeks worth of backlogs
rotate 4
# uncomment this if you want your log files compressed
compress
```

Gornji primer prikazuje globalne direktive koje se primenjuju na sve log fajlove koji su konfigurisani u nastavku ovog fajla:

- log fajlovi se rotiraju nedeljno
- čuvaju se četiri prethodna fajla
- svi fajlovi osim aktivnog se kompresuju

Direktive specifične za neki fajl mogu se smestiti u `/etc/logrotate.conf` ali je puno bolje da se smeštaju kao zasebni fajlovi u direktorijum `/etc/logrotate.d/`.

Primer jednog takvog fajla je:

```
/var/log/messages {  
    rotate 5  
    weekly  
    postrotate  
        /usr/bin/killall -HUP syslogd  
    endscript  
}
```

Ova konfiguracija je specifična za fajl `/var/log/messages`:

- čuva se pet starijih log fajlova
- fajlovi se rotiraju nedeljno
- posle rotacije fajla treba izvršiti komandu `'/usr/bin/killall -HUP syslogd'`

U nastavku je lista najčešće korišćenih direktiva za 'logrotate':

daily Definiše dnevno rotiranje logova

weekly Definiše nedeljno rotiranje logova

monthly Definiše mesečno rotiranje logova

yearly Definiše godišnje rotiranje logova

compress Definiše da se rotirani fajlovi kompresuju

nocompress Definiše da se rotirani fajlovi ne kompresuju

compresscmd Definiše komandu za kompresovanje fajlova

uncompresscmd Definiše komandu za dekompresiju fajlova

compressext Definiše ekstenziju koju će dobiti kompresovani fajlovi

compressoptions Definiše opcije koje se prosleđuju programu za kompresiju

delaycompress Definiše da se rotirani fajl kompresuje tek pri sledećem rotiranju

rotate broj Definiše koliko poslednjih rotiranih fajlova treba čuvati

mail adresa Definiše da se rotirani fajlovi šalju na navedenu email adresu

nomail Definiše da se rotirani fajlovi ne šalju emailom

mailfirst Definiše da se poslednji rotirani fajl šalje preko emaila

maillast Definiše da se prvi sačuvani rotirani fajl (fajl koji upravo treba da se obriše) šalje preko emaila. Ovo je podrazumevano ponašanje za 'mail' direktivu.

1.3 Vežbe

1. Kreirajte potreban šablon i pravilo kojim se sve poruke prioriteta 'notice' i većeg snimaju u fajl `/var/log/allmsgs.log`, tako da je format poruke:

```
YYYY-MM-DD-HH-MM:host:program:objekat:prioritet:tekst-poruke
```

Property of Admin Training Center

Modul 2: Init ramdisk, kernel i moduli

2.1 Initramfs

Prilikom startovanja sistema, kernel mora da učita dodatne module koji upravljaju hardverom sistema, kao i da obavi osnovnu konfiguraciju sistema pre nego što zakači / (root) fajlsistem.

Potrebni moduli se podrazumevano nalaze na / fajlsistemu, u poddirektorijumu `/lib/modules` ali postoji mogućnost da kernel nema u sebi potrebne drajvere za pristup uređaju na kojem se nalazi ovaj fajlsistem, već da je taj drajver u obliku kernel modula. Takva situacija dovodi do *deadlock-a*: kernel mora da zakači / fajlsistem, ali da bi to odradio mora prethodno da učita modul koji se nalazi na tom fajlsistemu.

Iz tog razloga, svi potrebni moduli za hardver datog sistema se nalaze u init ramdisku, koji se naziva *initramfs*. Ovaj ramdisk predstavlja potpuno ogoljen / fajlsistem u kojem se nalaze samo moduli, potrebni konfiguracioni fajlovi i minimalan set komandi koje su potrebne da bi se zakačio 'pravi' file/ fajlsistem.

Iako se 'initramfs' nalazi takođe na / fajl sistemu, njega, kao i sam kernelski fajl učitava *boot loader* preko BIOS funkcija koje kernel ne koristi. Iz tog razloga je moguće učitati init ramdisk iako kernel za to ne poseduje odgovarajuće drajvere.

Sam 'initramfs' se generiše automatski prilikom instalacije sistema i svaki put kada se ažurira kernel ili odgovarajući moduli. No, ponekad je potrebno regenerisati samostalno ovaj fajl i za to služi komanda:

```
# dracut [opcije] [initramfs-fajl.img kernel-verzija]
```

Initramfs mora tačno odgovarati verziji kernela (naziv fajla je *initramfs-kernel-verzija* i nalazi se u `/boot` direktorijumu). Komanda 'dracut' inače neće prebrisati postojeći initramfs fajl ukoliko se ne navede opcija '--force'.

Šta se od modula treba naći u initramfs fajlu definiše se konfiguracionim fajlom 'dracut' programa, `/etc/dracut.conf`, u direktivi:

```
add_dracutmodules = module1 module2...
```

Sadržaj initramfs se može dobiti komandom:

```
# lsinitrd initramfs-fajl
```

Na primer:

```
# lsinitrd initramfs-2.6.32-22.el6.x86_64.img
initramfs-2.6.32-22.el6.x86_64.img:
=====
dracut-004-17.el6
=====
drwxr-xr-x 23 root      root          0 May  3 22:34 .
drwxr-xr-x  2 root      root          0 May  3 22:33 proc
-rw xr-xr-x  1 root      root 7575 Mar 25 19:53 init
drwxr-xr-x  7 root      root          0 May  3 22:34 etc
drwxr-xr-x  2 root      root          0 May  3 22:34 etc/modprobe.d
[ostatak ispisa odsečen]
```

2.2 Konfigurisanje GRUB boot loader-a

Konfiguracioni fajl GRUB boot loader-a `/boot/grub/grub.conf` sadrži nekoliko opštih direktiva (default, timeout, splashimage, hiddenmenu). Ostatak fajla sadrži sekcije od po 4 linije koje definišu različite kernele prisutne na sistemu. Ove sekcije uvek počinju direktivom 'title' koja sadrži ime kernela koje se prikazuje u meniju, a a praćene su direktivama koje su uvučene u odnosu na 'title' direktivu:

root Definiše particiju na kojoj se nalazi kernel. Sve putanje u ostalim direktivama su relativne u odnosu na koreni direktorijum ove particije. Obično je to `/` ili `/boot` particija. Označavanje particije je oblika:

`(hdredni-broj-diska,redni-broj-particije)`

gde su 'redni-broj-diska' i 'redni-broj-particije' redni brojevi koji počinju sa 0. Redosled diskova je onakav kakav ga vidi BIOS.

kernel Komandna linija kernela. Na početku linije se nalazi putanja do kernel fajla a ostatak linije su parametri kernela.

initrd Definiše koji initramfs fajl treba koristiti.

Primer:

```
# grub.conf generated by anaconda
[comments omitted]
default=1
timeout=0
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu

title Red Hat Enterprise Linux (2.6.32-22.el6.x86_64)
root (hd0,0)
kernel /vmlinuz-2.6.32-22.el6.x86_64 ro \
        root=/dev/mapper/vg_vm6b-lv_root rd_LVM_LV=vg_vm6b/lv_root \
        rd_NO_LUKS rd_NO_MD rd_NO_DM LANG=en_US.UTF-8 \
        SYSFONT=latarcyrheb-sun16 KEYBOARDTYPE=pc KEYTABLE=us \
        rhgb quiet crashkernel=auto
initrd /initramfs-2.6.32-22.el6.x86_64.img

title Red Hat Enterprise Linux (2.6.32-19.el6.x86_64)
root (hd0,0)
kernel /vmlinuz-2.6.32-19.el6.x86_64 ro \
        root=/dev/mapper/vg_vm6b-lv_root rd_LVM_LV=vg_vm6b/lv_root \
        rd_NO_LUKS rd_NO_MD rd_NO_DM LANG=en_US.UTF-8 \
        SYSFONT=latarcyrheb-sun16 KEYBOARDTYPE=pc KEYTABLE=us \
```

```

    rhgb quiet crashkernel=auto
    initrd /initramfs-2.6.32-19.el6.x86_64.img

title Red Hat Enterprise Linux 6 (2.6.32-17.el6.x86_64)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-17.el6.x86_64 ro \
        root=/dev/mapper/vg_vm6b-lv_root rd_LVM_LV=vg_vm6b/lv_root \
        rd_NO_LUKS rd_NO_MD rd_NO_DM LANG=en_US.UTF-8 \
        SYSFONT=latacyrheb-sun16 KEYBOARDTYPE=pc KEYTABLE=us \
        rhgb quiet
    initrd /initramfs-2.6.32-17.el6.x86_64.img

```

Opšte direktive GRUB boot loader-a su:

default= Definiše koja sekcija (počevši od rednog broja 0) će biti iskorišćena kao definicija za podrazumevani kernel ako korisnik manuelno ne odabere neki drugi kernel.

timeout= Definiše koliko dugo (u sekundama) će biti dostupna mogućnost izbora stavke menija, tj. kornela prilikom startovanja sistema. Jednom kada je neka stavka odabrana, korisnik treba da pritisne 'ENTER' ili 'b' kako bi se taj kernel startovao. Parametar '0' označava da se meni sa stavkama ne ispisuje. U tom slučaju, ako na vreme pritisnete bilo koji alfanumerički taster (odmah posle BIOS poruka), meni će se pojaviti.

splashimage= Ova direktiva definiše sliku koja se pojavljuje kada se meni prikazuje. Putanja treba da sadrži oznaku particije i putanju od korenog direktorijuma te particije.

hiddenmenu Meni se podrazumevano neće prikazati čak ni ako je 'timeout=' različit od nule, već će se prikazivati poruka koja objašnjava šta treba uraditi da bi se meni pojavio. Poruka će trajati onoliko koliko je definisano u direktivi 'timeout='.

2.3 Rad sa kernel modulima

Teoretski, Linux kernel spada u monolitne kernele, što znači da se sav kod koji sačinjava kernel izvršava u zaštićenom modu procesora, u kojem su dostupne sve njegove komande i gde postoji neograničen pristup memoriji. Standardno, monolitni korneli su obično u obliku jednog velikog fajla sa kodom koji se učitava prilikom startovanja sistema. Bilo kakva izmena zahteva da se kernelski kod ponovo kompajlira kako bi se dobio novi kernel fajl.

Suprotno od monolitnih kornela su tzv. mikrokerneli — oni se sastoje od jednog vrlo malog i ogoljenog kernelskog fajla koji se izvršava u zaštićenom modu i koji samo obezbeđuje osnovne mogućnosti i komunikaciju između podsistema kornela. Ostatak čine moduli koji se izvršavaju u korisničkom okruženju i po potrebi se učitavaju ili izbacuju. Prednost mikrokernela je ta da se lako rekonfigurišu i da moduli nemaju pristup određenim komandama procesora i nekim zaštićenim delovima memorije. Loše strane mikrokernela su upravo iste: moduli ne mogu da izvršavaju neke specifične komande niti da pristupe sami određenim delovima memorije i onda zahtevaju da to umesto njih odradjuje mikrokernel. To jeste, sa stvarišta bezbednosti bolja opcija ali se ta dodatna bezbednost plaća smanjenjem performansi (usled konstantnog menjanja konteksta iz kernel u korisnički mod i obratno).

Linuxov kernel je iz tog razloga krenuo nekom srednjom putanjom: u stvarnosti je to i dalje monolitan kernel, ali sa ugrađenim sistemom za dinamičko učitavanje i izbacivanje modula, tako da je konfigurabilan skoro kao i svaki drugi mikrokernel, ali bez smanjenja performansi jer svaki modul može pristupiti svim delovima memorije i svim instrukcijama procesora.

Izuvez onih osnovnih sistema koji u najvećem delu spadaju u neki viši nivo apstrakcije sistema u samom kernelu, sve ostalo je definisano i dostupno preko modula, kao npr:

- drajveri za različite hardverske podsisteme
- podrška za različite tipove fajl sistema, mrežnih protokola i sl.

Sami moduli mogu modifikovati svoje ponašanje na osnovu parametara koji im se mogu zadati prilikom pozivanja. Takođe, pojedinačan modul ne mora biti nezavisna celina već može zavisiti od drugih modula. Kernel je sposoban da razreši ove međuzavisnosti samostalno i automatski učita nedostajuće module.

Takođe, modul se ne može izbaciti iz memorije ukoliko se:

- pristupa nekom hardverskom uređaju čiji drajver implementira modul
- u memoriji nalazi drugi modul koji zavisi od onog modula koji nameravamo da izbacimo iz memorije

Moduli se većinom automatski učitavaju, ali postoje mogućnost manuelne manipulacije njima:

- učitavanje (sa i bez razrešenja međuzavisnosti)
- dobijanje informacija o učitanim i neučitanim modulima
- uklanjanje modula (nenasilno i nasilno — uz nepoštovanje međuzavisnosti)

2.3.1 Prikaz trenutno učitanih modula

Komandom 'lsmod' možemo prikazati trenutno učitane kernel module kao i videi njihovu međuzavisnost, kao na primeru:

```
# lsmod
Module           Size  Used by
xfs              803635  1
exportfs          3424  1 xfs
vfat              8216  1
fat               43410  1 vfat
tun               13014  2
fuse              54749  2
ip6table_filter   2743  0
ip6_tables        16558  1 ip6table_filter
ebtable_nat       1895  0
ebtables          15186  1 ebtable_nat
ipt_MASQUERADE    2208  6
iptable_nat       5420  1
nf_nat            19059  2 ipt_MASQUERADE,iptable_nat
rfcomm            65122  4
ipv6              267017 33
sco               16204  2
bridge             45753  0
stp                1887  1 bridge
llc                4557  2 bridge,stp
bnep              15121  2
l2cap             45185  16 rfcomm,bnep
cpufreq_ondemand  8420  2
acpi_cpufreq       7493  1
freq_table         3851  2 cpufreq_ondemand,acpi_cpufreq
usb_storage        44536  1
sha256_generic     10023  2
aes_x86_64          7654  5
aes_generic         27012  1 aes_x86_64
cbc                2793  1
```

```
dm_crypt          10930  1
kvm_intel        40311  0
kvm              253162  1 kvm_intel
[ispis programa odsečen]
```

Komanda 'lsmod' prikazuje sledeće podatke:

- naziv modula
- količinu memorije koju modul trenutno zauzima
- zbir broja procesa koji koriste dati modul i broja zavisnih modula, praćen spiskom zavisnih modula.

2.3.2 Prikaz informacija o modulu

Komandom 'modinfo' možemo dobiti detaljan prikaz informacija o modulu. Komanda zahteva ime modula kao argument, kao u primeru:

```
# modinfo e1000e
filename:      /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/net/e1000e/e1000e.ko
version:       1.2.7-k2
license:       GPL
description:   Intel(R) PRO/1000 Network Driver
author:        Intel Corporation, <linux.nics@intel.com>
srcversion:    93CB73D3995B501872B2982
alias:         pci:v00008086d00001503sv*sd*bc*sc*i*
alias:         pci:v00008086d00001502sv*sd*bc*sc*i*
[neke alias linije su izostavljene]
alias:         pci:v00008086d0000105Esv*sd*bc*sc*i*
depends:
vermagic:     2.6.32-71.el6.x86_64 SMP mod_unload modversions
parm:          copybreak:Maximum size of packet [ispis skraćen] (uint)
parm:          TxIntDelay:Transmit Interrupt Delay (array of int)
parm:          TxAbsIntDelay:Transmit Absolute Interrupt Delay (array of int)
parm:          RxIntDelay:Receive Interrupt Delay (array of int)
parm:          RxAbsIntDelay:Receive Absolute Interrupt Delay (array of int)
parm:          InterruptThrottleRate:Interrupt Throttling Rate (array of int)
parm:          IntMode:Interrupt Mode (array of int)
parm:          SmartPowerDownEnable:Enable PHY smart power down (array of int)
parm:          KumeranLockLoss:Enable Kumeran lock loss workaround (array of int)
parm:          WriteProtectNVM:Write-protect NVM [ispis skraćen] (array of int)
parm:          CrcStripping:Enable CRC Stripping, [ispis skraćen] (array of int)
parm:          EEE:Enable/disable on parts that support the feature (array of int)
```

Sledi opis nekih polja ispisa:

filename Apsolutna putanja do .ko kernel-objektnog fajla modula. Opcija '-n' će prikazati samo ovo polje.

description Kratak opis modula. Opcija '-d' će prikazati samo ovo polje.

alias Ova linija pojavljuje se za svaki alias, tj. alternativno ime modula.

depends Ova linija sadrži spisak modula od kojih posmatrani modul zavisi.

parm Svaki parametar modula je prikazan u zasebnoj 'parm' liniji u obliku 'ime-parametra:kratak opis parametra'. Kratak opis parametra sadrži tip argumenta koji parametar očekuje. Ime parametra se koristi kada se navode parametri modula, bilo u komandnoj liniji, bilo u 'option' liniji u /etc/modprobe.d/ konfiguracionim fajlovima. Sažetija lista parametara se može dobiti opcijom '-p' komande 'modprobe'.

2.3.3 Učitavanje modula

Manuelno učitavanje modula u kernel se obavlja komandom:

```
# modprobe ime-modula [parametar1=vrednost1 parametar2=vrednost2...]
```

gde je 'ime-modula' ime kernel-objektnog fajla bez putanje i bez ekstenzije '.ko'. Uz komandu 'modprobe' mogu se zadati i vrednosti nekih parametara modula, ukoliko je to potrebno.

Komanda 'modprobe' će dati modul potražiti u podstablu:

```
/lib/modules/kernel-verzija/kernel/drivers/
```

U ovom podstablu postoje poddirektorijumi kao što su: `net/`, `scsi/` ili `fs/`, u kojima se nalaze moduli, grupisani po svojoj svrsi. Komanda će, pre učitavanja samog modula, izvući informacije o tome od kojih modula ovaj modul zavisi, a onda pokušati na isti način da učita i te module. Na kraju, kada se sračuna lista zavisnih modula, svi nedostajući moduli će biti učitani korišćenjem komande 'insmod'.

Komanda 'insmod' je slična komandi 'modprobe' ali umesto samog imena modula njoj treba kao argument navesti puno ime kernel-objektnog fajla. Parametri se prosleđuju na isti način kao i kod komande 'modprobe'. Najvažnija razlika u odnosu na komandu 'modprobe' je što 'insmod' **ne razlučuje međuzavisnosti između modula**.

Upozorenje!

Uvek koristite 'modprobe' za učitavanje modula jer ćete tako učitati i zavisne module!

Opcija '-v' će prikazati koji se sve moduli učitavaju, kao u primeru:

```
# modprobe -v fcoe
insmod /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/scsi/scsi_tgt.ko
insmod /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/scsi/scsi_transport_fc.ko
insmod /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/scsi/libfc/libfc.ko
insmod /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/scsi/fcoe/libfcoe.ko
insmod /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/scsi/fcoe.fcoe.ko
```

2.3.4 Uklanjanje modula iz kernela

Uklanjanje modula se može izvesti na dva načina. Sigurniji način je komanda:

```
# modprobe -r ime-modula
```

Ova komanda će pokušati da ukloni navedeni modul i sve njegove zavisne module. Ukoliko bilo koji od modula ne može biti uklonjen, komanda neće ukloniti nijedan od modula.

Druga komanda, koja je nešto opasnija je:

```
# rmmod ime-modula
```

Ona će ukloniti samo navedeni modul ukoliko on nije u upotrebi od strane nekog uređaja ili drugih modula koji su zavisni od njega.

Razlika između ove dve komande je najvidljivija na nekom primeru:

```
# modprobe -r wacom
```

Ova komanda će pokušati da iz kernela ukloni modul za upravljanje Wacom grafičkim tablama. Komanda neće ukloniti nijedan modul ako je ispunjen bilo koji od sledećih uslova:

- neki proces koristi wacom modul

- neki proces koristi modul od kojeg wacom modul direktno zavisi
- neki proces koristi neki od modula od kojih zavise moduli od kojih zavisi sam wacom modul

Dakle, bez obzira da li neko koristi Wacom grafičku tablu ili ne, moguće je da wacom modul neće biti moguće ukloniti iz kornela jer su neki drugi moduli, od kojih wacom modul direktno ili indirektno zavisi, u upotrebi od strane nekih drugih procesa ili uređaja.

U slučaju komande:

```
# rmmod wacom
```

modul neće biti uklonjen isključivo ako postoji neki proces ili modul koji koristi modul 'wacom'. Da li taj uslov postoji možete proveriti komandom 'lsmod | grep wacom' i proveriti da li u trećem polju postoji broj različit od 0.

U svakom slučaju, budući da je komanda 'rmmod' opasnija, uvek prvo probajte da uklonite modul komandom 'modprobe -r'. Ukoliko to nije moguće, a sami ste, izlistavanjem svih informacija putem 'modprobe' i 'lsmod' komandi došli do zaključka da je moguće ukloniti dati modul bez da on ugrozi rad sistema, probajte 'rmmod' komandom da uklonite taj modul.

2.3.5 Postavljanje parametara modula

Ukoliko želimo da neki modul učitate sa parametrima različitim od onih koji su podrazumevani ili sa kojima je dati modul učitan, onda to možete uraditi iz komandne linije navođenjem liste parametara kao što je opisano na strani 24).

No, ako želite da dati modul uvek bude učitavan sa vašim parametrima, onda je potrebno te parametre ubaciti u odgovarajući fajl u `/etc/modprobe.d/` direktorijumu. Fajl može imati bilo kakvo ime, ali mora imati ekstenziju `.conf` da bi ga 'kmod' sistem za učitavanje modula registrovao.

Pored parametara, ovi fajlovi mogu imati još neke direktive koje utiču na učitavanje modula, kao što su:

- mogućnost definisanja dodatnih aliasa za dati modul, uključujući i *wildcard globing*.
- mogućnost specificiranja posebnog niza komandi koji treba izvršiti kada se neki modul učitava
- mogućnost da aliasi imaju različite parametre od originalnog modula
- mogućnost da se blokiraju interni aliasi za neki modul (ukoliko više modula ima isti alias a vama se učitava pogrešan modul, ili neki modul pogrešno tvrdi da je alias za nešto)
- mogućnost da se definiše redosled učitavanja zavisnih modula

Direktive koje se mogu koristiti u `.conf` fajlovima su:

alias wildcard ime-modula Definiše dodatne aliase za dati modul 'ime-modula'. Ovako definisani aliasi mogu imati svoje setove parametara. U imenu aliasa može se naći i znak '*' koji zamenjuje pojavljivanje 0 ili više znakova, pa je moguće na taj način jednom 'alias' direktivom definisati veliki broj aliasa za isti modul.

blacklist ime-modula Označava da svi interni aliasi navedenog modula trebaju biti ignorisani. Koristi se kada više modula definiše isti alias ili kada modul definiše pogrešan alias. Efektivno, blokira učitavanje navedenog modula, izuzev ako se on ne učitava pod svojim originalnim imenom.

install ime-modula komanda... Definiše da se prilikom učitavanja navedenog modula trebaju izvršiti zadate komande (koje su obično 'modprobe' komande, ali ne moraju biti). Na ovaj način moguće je npr. definisati međuzavisnost između modula koja nije navedena u samim modulima.

options ime-modula parametri Specificira podrazumevane parametre sa kojim će se modul učitavati bilo direktno preko komande 'modprobe', bilo indirektno, ako zavisni modul. Deo linije 'parametri' je isti kao spisak parametara prilikom pozivanja komande 'modprobe'. Ako je modprobe pozvana sa navedenim parametrima, onda ti parametri, ako su imena parametara ista kao kod onih u .conf fajlu onda parametri iz komandne linije imaju prednost, inače se i jedni i drugi spajaju u zajedničku listu parametara sa kojom se modul učitava. Aliasi mogu imati svoje sopstvene liste parametara, ako se na mesto imena modula unese ime aliasa.

remove ime-modula komanda... Slično 'install' direktivi, samo važi kod uklanjanja modula preko 'modprobe -r' komande.

softdep ime-modula pre: modul... post: modul... Definiše 'meku' zavisnost među modulima. Koristi se u slučajevima gde ne postoji predefinisane međuzavisnosti, ali je potrebno zajedno učitati module. Parametar 'pre:' definiše module koji trebaju biti učitani pre originalnog modula, a parametar 'post:' definiše module koji trebaju biti učitani posle originalnog modula. Primetite da se ista funkcionalnost može postići odgovarajućom 'install...' direktivom ali da je 'softdep' jasnija i ima prioritet u odnosu na 'install' i 'remove' komande.

Ako želimo da učitamo neki modul prilikom startovanja sistema, onda je potrebno napraviti odgovarajući izvršni komandni skript u direktorijumu /etc/sysconfig/modules/. Fajl može imati bilo kakvo ime ali je obavezno da ima .modules ekstenziju i da bude izvršan. Primer fajla je dat u sledećem listingu:

```
#!/bin/sh

if [ ! -c /dev/input/uinput ] ; then
    exec /sbin/modprobe uinput >/dev/null 2>&1
fi
```

U gornjem skriptu proverava se da li postoji karakter specijalni fajl /dev/input/uinput i ako ne postoji, učitava se modul 'uinput' koji implementira Bluetooth input uređaj.

2.4 Vežbe

1. Izlistajte sadržaje svih initramfs fajlova iz vašeg /boot direktorijuma.
2. Promenite definiciju GRUB boot loadera tako da uvek učitava prvi zapis za kernel i da uvek prikazuje meni u trajanju od 10 sekundi.
3. Izlistajte fajlove u /etc/modprobe.d/ direktorijumu i pokušajte da utvrdite koji moduli imaju specifične 'install' direktive i šta se tim direktivama radi.

Modul 3: Proširena kontrola pristupa fajlovima

3.1 Uvod

Standardna kontrola pristupa fajlovima na Linuxu podrazumeva da se prava pristupa definišu nad tri grupe korisnika:

- vlasnika fajla (grupa koja broji samo jednog člana)
- grupu asociranu fajlu
- sve ostale korisnike koji nisu vlasnik ili članovi grupe asocirane fajlu

Ova podela je prilično gruba, jer nije moguće podešavati prava pristupa za pojedinačnog korisnika koji nije vlasnik fajla. Iz tog razloga, na ext3/4 sistemima i NFS šerovima su implementirane liste kontrole pristupa (ACL - Access Control Lists).

Upotreba ACL na fajl sistemima zahteva dve stvari:

- podršku u kernelu (podrazumevano je ova podrška uključena u standardne kernele za RedHat/CentOS i Debian/Ubuntu sisteme)
- paket pomoćnih programa 'acl' koji treba instalirati iz standardnih repozitorijuma a koji omogućava manipulaciju listama.

Takođe, komande 'cp' i 'mv' će kopirati, odnosno prebaciti ACL zajedno sa fajlovima i direktorijumima koji se kopiraju/prebacuju.

3.2 Omogućavanje ACL

Ako želimo da koristimo ACL na nekom fajl sistemu, moramo ga zakačiti sa uključenom 'acl' opcijom:

```
# mount -t ext3 -o acl ime-particije direktorijum
```

Ukoliko želimo da permanentno dodamo podršku za ACL na nekoj particiji onda opciju 'acl' dodajemo u **/etc/fstab** fajl:

LABEL=/home	/home	ext4	acl,defaults	1 2
-------------	-------	------	--------------	-----

3.3 Postavljanje prava preko ACL

Postoje dve vrste ACL:

- pristupne ACL (*access ACL*)
- podrazumevane ACL (*default ACL*)

Pristupna ACL je lista kontrole pristupa za specifični fajl ili direktorijum. Podrazumevana ACL je lista kontrole pristupa koja je asocirana isključivo direktorijumima i služi da definiše prava pristupa za one fajlove unutar tog direktorijuma koji nemaju svoje specifične pristupne ACL.

ACL mogu biti definisane:

- po korisniku
- po grupi
- preko maske efektivnih prava
- za korisnike koji nisu u grupi asociranoj fajlu

Komanda 'setfacl' postavlja ACL za fajlove i direktorijume. Koristite opciju '-m' za dodavanje ili menjanje ACL na nekom fajlu ili direktorijumu:

```
# setfacl -m prava fajlovi
```

Prava moraju biti specificirana po sledećim formatima:

u:uid:perms 'u' označava da se postavljaju prava za korisnika; 'uid' je UID ili korisničko ime; 'perms' je kombinacija slova 'r', 'w' i 'x';

g:gid:perms 'g' označava da se postavljaju prava za grupu; 'gid' je GID ili ime grupe; 'perms' je kombinacija slova 'r', 'w' i 'x';

m:perms 'm' označava postavljanje efektivne maske; 'perms' je kombinacija slova 'r', 'w' i 'x';

o:perms 'o' označava postavljanje prava za ostale korisnike; 'perms' je kombinacija slova 'r', 'w' i 'x'.

Više prava može biti postavljeno na istom fajlu jednim pozivom komande 'setfacl' ako ih razdvojimo zarezom unutar komandne linije:

```
# setfacl -m pravo1,pravo2,... fajl...
```

Ako fajl koji navodimo kao argument komande 'setfacl' već ima postavljenu ACL, onda će komanda dodati nova prava odnosno promeniti ona prava koja su već definisana za datog korisnika ili grupu.

Podrazumevana prava se postavljaju na isti način i istom komandom samo što se kao argument mora pojaviti direktorijum i svako pravo mora imati dodatni prefiks 'd:' koji se dodaje nad pravo specificirano na neki od prethodno navedenih načina.

3.4 Pregledanje ACL

Komanda:

```
# getfacl [opcije] fajl...
```

će prikazati ACL sa odgovarajućim zaglavljem (ime fajla, vlasnik, grupa). Najčešća opcija je '-omit-header' koja ne prikazuje zaglavje već samo ACL za zadati fajl.

3.5 Uklanjanje prava iz ACL i same pristupne ACL

Komanda:

```
# setfacl -b fajl...
```

će ukloniti sve stavke iz ACL. Komanda:

```
# setfacl -x pravo fajl...
```

će ukloniti zadata prava ako ona postoje u ACL za zadati fajl.

3.6 Određivanje pristupa fajlu preko ACL

Ako su ACL omogućene na nekom fajl sistemu, onda se prava pristupa određuju preko ACL. Standardna prava nad fajlovima i direktorijumima se automatski mapiraju u odgovarajuće stavke unutar ACL. Na primer ako kreiramo fajl 'proba.dat' negde na fajl sistemu koji ima uključenu podršku za ACL, uz pretpostavku da na nadređenom direktorijumu nije postavljena podrazumevana ACL, dobijamo:

```
$ id
uid=1000(user1) gid=1000(user1) groups=1000(user1)
$ pwd
/home/user1
$ touch proba.dat
$ chmod 0750 proba.dat
$ ls -l proba.dat
-rwxr-x--- 1 user1 user1 0 Jul 14 12:13 proba.dat
```

Prethodnim nizom komandi smo kreirali fajl proba.dat u početnom direktorijumu korisnika 'user1' i promenili smo mu prava pristupa tako da vlasnik ima sva prava, grupa 'user1' koja je vlasnik fajla ima prava r i x a ostali nemaju nikakva prava.

ACL za ovaj fajl izgleda ovako:

```
$ getfacl proba.dat
# file: proba.dat
# owner: user1
# group: user1
user::rwx
group::r-x
other::---
```

Obratite pažnju da su se standardna prava pristupa mapirala u odgovarajuće stavke ACL koje nisu imenovane! Takođe, efektivna maska pristupa nije definisana.

Ako sada dodamo sva prava pristupa preko ACL za korisnika 'pera' koji inače nije član grupe 'user1', dobijamo sledeći ACL:

```
$ setfacl -m u:pera:rwx proba.dat
$ getfacl proba.dat
# file: proba.dat
# owner: user1
# group: user1
```

```
user::rwx
user:pera:rwx
group::r-x
mask::rwx
other::---
```

Kao što se vidi iz listinga, dodate su dve nove stavke u ACL za fajl proba.dat: jedna za korisnika 'pera', a druga kao maska efektivnih prava, koja je implicitno kreirana. Ova maska je unija svih prava koja su definisana na fajlu proba.dat. Značenje maske će biti objašnjeno dalje u tekstu.

Ako ponovo izlistamo podatke o fajlu proba.dat, dobijamo:

```
$ ls -l proba.dat
-rwxrwx---+ 1 user1 user1 0 Jul 14 12:13 proba.dat
```

Kao što se vidi, prava za grupu su promenjena tako da odgovaraju maski, a dodat je i znak '+' koji označava da postoji pristupna ACL koja definiše prava pristupa za fajl proba.dat.

Ako sada promenimo masku tako da sadrži samo prava 'r' i 'x', bilo komandom:

```
$ setfacl -m m:rx proba.dat
```

bilo komandom

```
$ chmod g-w proba.dat
```

dobićemo sledeću ACL:

```
$ getfacl proba.dat
# file: proba.dat
# owner: user1
# group: user1
user::rwx
user:pera:rwx #effective:r-x
group::r-x
mask::r-x
other::---
```

Kao što se vidi, bez obzira što korisnik pera ima stavku sa eksplisitnim 'w' pravom, on neće moći da menja sadržaj fajla proba.dat jer maska ne sadrži to pravo, pa je ono svima uskraćeno, izuzev vlasniku fajla.

Iz ovoga možemo odrediti algoritam po kojem se određuju prava pristupa kada se koriste ACL:

1. Ako je UID procesa koji pristupa fajlu identičan UID-u vlasnika fajla, onda željeni pristup dozvoljen ukoliko zapis vezan za vlasnika fajla (user::) ima taj pristup, inače
2. ako je UID procesa koji pristupa fajlu identičan sa UID-om nekog imenovanog korisničkog zapisa u ACL onda je dati pristup dozvoljen ako postoji u tom zapisu (user:UID:) i u masci (mask:), inače
3. ako je jedan od GID-ova procesa identičan GID-u grupe koja je vlasnik fajla, onda je pristup dozvoljen ukoliko istovremeno postoji taj pristup u zapisu grupe vlasnika fajla (group::) i maske, inače
4. ako je jedan od GID-ova procesa identičan sa nekim imenovanim grupnim zapisom (group;GID:) i istovremeno u tom zapisu i u masci postoji to pravilo, onda je pristup dozvoljen, inače

5. ako je jedan od GID-ova procesa identičan grupi vlasnika fajla ili jednom ili više imenovanih grupnih zapisa ali ni u jednom ne postoji dozvoljeno pravo za zahtevani pristup ili je ono maskirano zapisom za masku, pristup je odbijen, inače
6. pristup određuje zapis za ostale korisnike (other:).

Ukoliko zapis za masku efektivnih prava ne postoji u ACL-u smatra se da su sva prava dozvoljena, tj. sistem se ponaša kao da postoji implicitan zapis 'm:rwx'.

3.7 Vežbe

1. Omogućiti upotrebu ACL na /home particiji testnog servera.
2. Kreirati dva korisnika (user1 i user2) koja zajedno nisu članovi nijedne grupe. Kreirati direktorijum čiji je vlasnik user1 i postaviti ACL-ove tako da:
 - grupa koja je vlasnik fajla ima prava 'r' i 'x'
 - ostali nemaju nikakva prava
3. Dodeliti prava 'r', 'w' i 'x' korisniku 'user2'.
4. Postaviti podrazumevana prava na kreiranom direktoriju takva da:
 - vlasnik ima sva prava
 - grupa koja je vlasnik ima sva prava
 - ostali nemaju nikakva prava
5. Unutar kreiranog direktorijuma kreirati neki poddirektorijum. Da li korisnik 'user2' ima pravo pisanja u novokreiranom poddirektorijumu i zašto?

Property of Admin Training Center

Modul 4: Linux firewall

4.1 OSI model

OSI (*Open Systems Interconnect*) model je teorijski opis funkcionisanja mreže kompjutera koji se sastoji od više slojeva od kojih svaki rešava jednu oblast problema vezanih za transport podataka kroz mrežu. Svaki sloj se oslanja na niži sloj da dobro obavlja svoju funkciju, i obezbeđuje dobro definisane usluge višim slojevima u modelu. Viši slojevi su apstraktniji, bliži korisnicima, dok su niži bliži računarima i ostalim mrežnim uređajima.

4.1.1 Slojevi OSI modela

1. Fizički sloj — UTP bakarni kabl
2. Data link sloj — Ethernet
3. Mrežni sloj — IP protokol
4. Transportni sloj — TCP/UDP protokoli
5. Sloj sesije — RPC, NFS
6. Prezentacioni sloj — JPEG slika
7. Aplikacioni sloj — HTTP, Telnet, SMTP i sl.

Fizički sloj (*layer 1*) definiše sve električne i fizičke specifikacije uređaja. Ovo uključuje raspored pinova, voltažu na kojoj uređaji rade, specifikacije kablova... Ovaj sloj obezbeđuje uspostavljanje i prekidanje veze (handshake), modulaciju signala, kontrolu toka podataka (flow control) i sl. Uređaji koji rade na ovom nivou su hub-ovi i repeater-i.

Data link sloj (*layer 2*) obezbeđuje prenošenje podataka između mrežnih entiteta, detektovanje i ispravljanje kolizija koje nastupaju u fizičkom sloju (collision detection) ili obezbeđuju izbegavanje kolizija (collision avoidance). Šema adresiranja je fizička, što znači da su adrese uređaja fizički kodirane unutar mrežnih karti. Najpoznatiji primer data link sloja je Ethernet, a uređaji koji rade na ovom nivou su bridge-vi i switch-evi.

Mrežni sloj (*layer 3*) obezbeđuje prenos paketa podataka različite veličine preko jedne ili više mreža. Šema adresiranja je hijerarhijska, a adrese uređaja su logičke. U ovom sloju se obavlja rutiranje (usmeravanje) putanje paketa do destinacije putem uređaja koji se zovu router-i. Primer protokola ovog sloja je IP (Internet Protokol).

Transportni sloj (*layer 4*) pruža transparentan i pouzdan prenos segmenata podataka između dva korisnika, brinući o tome da svi segmenti zaista i stignu na odredište. Protokoli koji rade na ovom nivou su TCP, UDP, ATM...

Sloj sesije (*layer 5*) pruža upravljanje dijalogom između dve korisničke aplikacije, bilo putem duplex (obosrana komunikacija) ili half-duplex (samo jedan priča) operacije. Ovim slojem se obezbeđuje uspostavljanje i prekidanje TCP/IP sesija.

Prezentacioni sloj (*layer 6*) obezbeđuje da gornji, aplikacioni sloj, ne mora da brine o razlikama u načinima reprezentovanja podataka između sistema. Ovde se obavlja MIME enkodiranje, enkripcija, konverzija kodova i sl.

Aplikacioni sloj (*layer 7*) kao najviši nivo direktno obavlja servise koje korisničke aplikacije zahtevaju. Na ovom sloju rade HTTP, telnet i drugi protokoli.

4.2 TCP/IP set protokola

TCP/IP je jezik kojim komunicira većina mrežnih uređaja. Ime je sačinjeno od dva protokola: TCP – Transmission Control Protocol i IP – Internet Protocol

4.2.1 IP protokol

IP spada u treći sloj OSI modela i predstavlja metod kojim se podaci od izvorišta do odredišta šalju preko niza drugih mrežnih uređaja. Svaki mrežni uređaj ima svoj jedinstveni IP broj kojim se identificuje kojoj mreži pripada i kuda treba poslati podatke za njega. Proces odlučivanja kuda idu podaci naziva se rutiranje.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31								
Verzija	IHL	ToS										Ukupna dužina paketa																											
16-bitna identifikacija																Flegovi	Offset fragmenta																						
TTL								Protokol								Kontrolna suma zaglavljia																							
Izvorišna IP adresa																Odredišna IP adresa																							
Opcije																																							

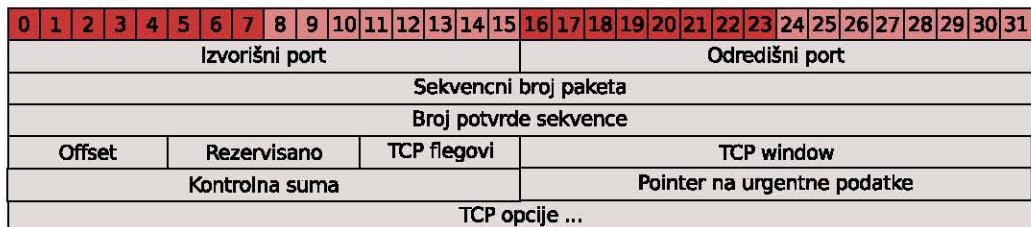
Slika 4.1: Zaglavje IP paketa

4.2.2 TCP protokol

TCP je protokol koji radi na četvrtom sloju OSI modela i koji obezbeđuje da podaci koji su poslati zaista ispravni i stignu na odredište. TCP deli podatke na fragmente i dodeljuje im redne brojeve, tako da se na taj način podeljeni podaci opet mogu sklopiti kada stignu do odredišta. Oni paketi koji ne stignu, ili stignu oštećeni, ponovo se šalju sve dok se ne prenese kompletan i ispravan podatak.

Paketi se u TCP/IP modelu sastoje od zaglavja i tela. Zaglavje paketa sadrži razne informacije o paketu: protokol, tip servisa, izvorišna i odredišna IP adresa, identifikacioni broj i redni broj fragmenta, dužinu života, checksum, kao i razne opcije i flegove. Evo kako šematski izgleda IP zaglavje:

Flegovi kojima mogu biti označeni TCP paketi prikazani su u sledećoj tabeli:



Slika 4.2: Zaglavje TCP paketa

Bit	Oznaka	Naziv
0	URG	Urgent
1	ACK	Acknowledgement
2	PSH	Push
3	RST	Reset
4	SYN	Synchronize
5	FIN	Finished

Tabela 4.1: TCP flegovi

TCP uspostavljanje konekcije

TCP je protokol kojim se prilikom razmene podataka između hostova ostvaruje konekcija koja garantuje ispravnu isporuku podataka. Da bi podaci, koji se kroz mrežu šalju podeljeni u pakete, mogli da se rekonstruišu kada stignu na odredište, paketi moraju biti označeni sekvenčnim brojem, obzirom da se dešava da stignu i različitim redosledom od onog kojim su slati. Sa jedne strane konekcije nalazi se aktivni host koji započinje konekciju, a sa druge je pasivni host, koji je u stanju LISTEN, što znači da očekuje takve konekcije. Njihovo 3 way handshake (rukovanje) se ostvaruje na sledeći način:

1. Inicijator konekcije šalje SYN paket koji sinhronizuje sekvenčni broj u paketima koji će biti poslati.
2. Drugi host uzvraća SYN/ACK pri čemu inicijator opet radi sinhronizaciju sekvence i potvrđuje inicijalni paket.
3. Inicijator uzvraća ACK koji potvrđuje paket koji je upravo primio.

Nakon ovoga konekcija se smatra uspostavljenom.

TCP prekid konekcije

Konekcija se prekida kada:

- neka od strana pošalje FIN paket
- neka od strana pošalje RST paket
- vreme za konekciju istekne

4.2.3 UDP protokol

UDP (user datagram protocol) se takođe koristi na četvrtom (transportnom) nivou OSI modela. Predstavlja nepouzdani protokol kojim se razmena podataka vrši bez uspostavljanja konekcije i ponovnog slanja neispravnih ili neisporučenih paketa. Koristi se tamo gde je

potrebna veća brzina, a manja pouzdanost prenosa podataka u odnosu na TCP, kao što je slučaj kod prenosa multimedijalnih podataka i sl.

UDP paketi se sastoje od samo četiri podatka u zaglavljaju: izvořišni i odredišni port, dužina paketa i checksum.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Izvořišni port																Odredišni port															
Dužina																Kontrolna suma															

Slika 4.3: Zaglavljje UDP paketa

4.3 Namena firewall-a

Osnovni način za obezbeđivanje mrežnog okruženja je kontrola i filtriranje paketa koji se razmenjuju između unutrašnje (zaštićene) i spoljne (nezaštićene) mreže, tako da se neki paketi propuštaju, a neki ne (paketi su segmenti podataka na trećem sloju OSI modela). **Firewall** (*firewall*, zid ili prepreka koja sprečava širenje požara) je sistem koji stoji između unutrašnje i spoljne mreže i vrši to filtriranje.

Odluke o tome koji se paketi propuštaju a koji ne, donose se na osnovu zaglavja paketa. Ono sadrži podatke o odredišnoj i izvořišnoj IP adresi, odredišnom i izvořišnom portu i dr. Pored zaglavja paket sadrži i telo, u kome se nalaze konkrentni podaci koji se prenose. Firewall analizira sadržaj zaglavja paketa i upoređuje ga sa prethodno definisanim pravilima koja se jedno za drugim nižu kao karike u lancu, sve dok se, prvim zadovoljenjem nekog pravila, ne odluči šta treba uraditi sa paketom. Paket se može prihvati, odbaciti ili preusmeriti na neki drugi lanac pravila radi dodatne analize.

Linux firewall u verzijama kernela počev od 2.4 se sastoji iz dva dela: kernelspace deo (netfilter) i userspace program (komanda iptables). Iptables je razvijen proširivanjem mogućnosti paketa Ipchains, firewall-a iz kernela verzije 2.2. Iako se značajno razlikuju, Iptables je unazad kompatibilan, pa se podešavanja za Ipchains mogu iskoristiti bez prepravljanja.

4.4 Tipovi firewalla

Postoji više tipova firewall-a u zavisnosti od toga koje funkcionalnosti pružaju. Osnovna podela je na filtere paketa i proxy firewall-ove, gde se podela vrši prema tome da li se (kao kod filtera paketa) filtriranje vrši samo na osnovu zaglavja paketa, ili i na osnovu sadržaja podataka koje ti paketi prenose (kako je to kod proxy firewall-ova). Proxy firewall-ovi pružaju veću bezbednost, ali su dosta kompleksniji i zahtevniji, a veliki broj protokola i servisa ne radi dobro sa ovakvom vrstom firewall-a.

Druga podela firewall-ova je na *stateless* i *stateful*:

stateless firewall obrađuje svaki paket kao zaseban entitet i ne pamti vezu između paketa koji su deo iste konekcije/sesije;

stateful firewall pamti, tamo gde je to moguće, informacije o konekciji i svaki paket pokušava da asocira nekoj od konekcija.

Iptables podržava stateful filtriranje pomoću modula **conntrack** (*connection tracking*) tako što se paketi posmatraju kao deo konekcije, a u internim tabelama sistema se čuvaju podaci o stanju u kome se konekcije nalaze. Konekcije mogu biti u sledećim stanjima:

NEW stanje označava pakete koji nisu deo uspostavljene konekcije već oni kojima se konekcija započinje.

ESTABLISHED je stanje u koje se prelazi nakon uspostavljanja konekcije

RELATED je stanje konekcije koja je u vezi sa nekom ESTABLISHED konekcijom (npr. FTP kontrolna i FTP data konekcija)

INVALID stanje označava sve pakete koji se ne mogu identifikovati

Kada aktivni host koji zahteva konekciju pošalje Syn paket, firewall registruje konekciju kao NEW, a kada pasivni host koji osluškuje konekcije uzvrati Syn/Ack paketom konekcija prelazi u stanje ESTABLISHED. I ako prvi paket koji stigne nije Syn (bilo zbog prirode protokola, bilo zbog neregularnog toka paketa) konekcija se označava kao NEW.

Po pravilu se svi paketi koji dolaze na firewall sa spoljne mreže propuštaju jedino ukoliko su deo ESTABLISHED ili RELATED konekcije, dok se zahtevi za uspostavljanjem konekcije dopuštaju jedino ukoliko dolaze iz unutrašnje mreže i ukoliko idu ka serverima u demilitarizovanoj zoni (DMZ), tj. zoni u kojoj su serveri kojima je dopušteno pristupanje i sa spoljne i sa unutrašnje mreže (npr. WWW, mail, FTP...).

Netfilter terminologija:

Lanci (Chains) nizovi pravila kojim se paketi filtriraju, tj. faze obrade paketa

Tabele tabele u kojima se vodi evidencija o konekcijama

Provere pravila provere koje se vrše u lancima (eng. rule matches)

Akcije akcije koje se preduzimaju nad paketom (eng. rule targets)

Netfilter lanci:

PREROUTING svi paketi koji ulaze u firewall (kroz npr: eth, lo, ppp...)

INPUT svi paketi koji su adresirani za sam firewall

FORWARD svi paketi koji se rutiraju preko firewall-a

OUTPUT svi paketi koje firewall generiše

POSTROUTING svi paketi koji napuštaju firewall

Netfilter tabele:

filter spisak filtera koji se primenjuju na konekcije

Dozvoljene akcije: ACCEPT, REJECT, DROP, LOG

nat (Network Address Translation) - prevodenje mrežnih adresa

Dozvoljene akcije: SNAT, DNAT, MASQUERADE

mangle Prepravljanje podataka u zaglavljtu

Dozvoljene akcije: Promena TTL, TOS / DSCP, postavljanje MARK

Svrha: promena rutiranja, praćenje paketa, statistika

conntrack Praćenje konekcija. Nije direktno dostupna korisnicima.

Netfilter pravila su provere po kojima se vrši filtriranje. Primeri su:

```
-p tcp      - svi paketi TCP protokola
-d a.b.c.d/n - odredišna (destination) adresa je a.b.c.d netmaska n
--dport x    - odredišni broj porta je x
--length     - broj bajtova u paketu
--mac-source  - MAC adresa uređaja koji je poslao paket
-i, -o       - ulazni (-i) ili izlazni (-o) interfejs kojim ide paket
```

Netfilter akcije su akcije koje se sprovode nad paketom kad zadovoljava pravilo:

ACCEPT paket se dozvoljava

DROP paket nije dozvoljen i odbacuje se bez obaveštenja pošiljaoca

REJECT paket nije dozvoljen i odbacuje se uz obaveštenje pošiljaocu

DNAT menja se odredišna adresa paketa

SNAT menja se izvorišna adresa paketa

LOG paket se zabeležava u syslog

MARK označava paket radi kasnijeg dodatnog procesiranja

MIRROR obrću se izvorišna i odredišna adresa

Radi dodatnog filtriranja akcija takođe može biti i neki korisnički definisan lanac.

4.5 Netfilter princip rada

Kada paket dođe na mrežni interfejs firewall-a prvu obradu preduzima odgovarajući deo operativnog sistema zadužen za kontrolisanje mrežne aktivnosti. Paket nakon toga prolazi kroz određena faze pre nego što se prosledi drugom mrežnom uređaju ili nekom višem servisu koji pruža firewall. Faze obrade se nazivaju lanci (chains).

Svi paketi koji dolaze do firewall-a prolaze kroz PREROUTING lanac MANGLE tabele, zatim kroz PREROUTING lanac NAT tabele i onda dolaze do trenutka odluke rutiranja kada se ispituje da li su paketi namenjeni za sam firewall ili za neku drugu mrežu.

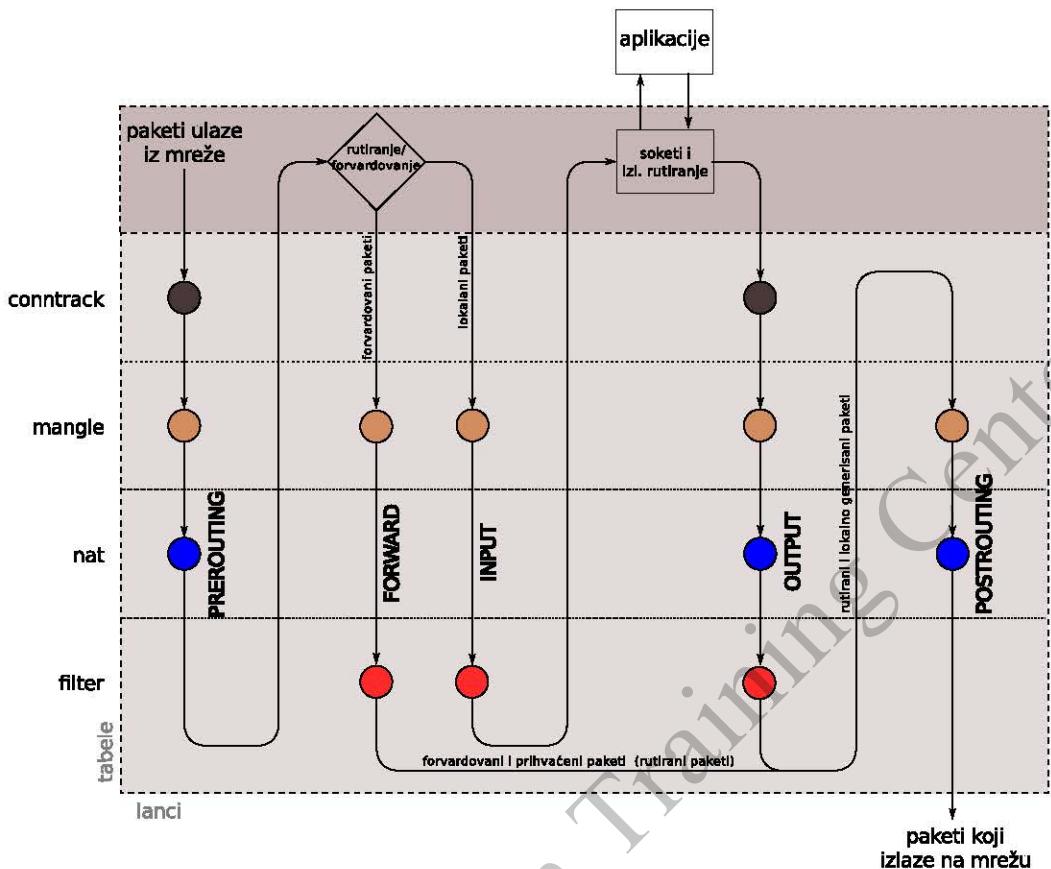
Kada je paket namenjen za sam firewall, biće prosleđen na INPUT lanac FILTER tabele, zatim na INPUT lanac MANGLE tabele da bi zatim bio prosleđen nekom višem servisu firewall-a.

Kada je paket namenjen nekoj drugoj mreži on ulazi u FORWARD lanac MANGLE tabele, zatim u FORWARD lanac FILTER tabele, da bi onda bio poslat na POSTROUTING lanac MANGLE i POSTROUTING lanac NAT tabele.

Ukoliko firewall generiše paket on prolazi kroz OUTPUT lanac MANGLE tabele, OUTPUT lanac NAT tabele i OUTPUT lanac FILTER tabele, da bi nakon toga bio prosleđen u POSTROUTING lanac MANGLE i POSTROUTING lanac NAT tabele.

4.6 Iptables userspace program

Komanda iptables dodaje i briše pravila iz kernela. Pravila koja su uneta važe do sledećeg restartovanja sistema. Postoje i skripte iptables-save i iptables-restore koje obavljaju snimanje i učitavanje svih pravila.



Slika 4.4: Prolazak paketa kroz netfilter

4.7 Iptables komande

Pravila se unose u sledećem obliku:

```
iptables [-t tabela] komanda [provera] [akcija]
```

Sledi spisak komandi, provera i akcija koji se mogu navesti u pravilu.

Komanda -A, --append

```
iptables -A INPUT ...
```

Ova komanda dodaje pravilo na kraju lanca INPUT. Obzirom da se pravila proveravaju redom kako su uneta, ovo pravilo će biti poslednje proveravano, osim ukoliko se doda još pravila posle ovog.

Komanda -D, --delete

```
iptables -D INPUT --dport 80 -j DROP, iptables -D INPUT 1
```

Ova komanda briše pravilo koje je ranije uneto i to na jedan od dva načina: unošenjem kompletног pravila koje želite da obriшете (prvi primer), ili unošenjem rednog broja pravila koje želite da obriшете. Pravila se broje odozgo na dole počev od broja 1.

Komanda -R, --replace

```
iptables -R INPUT 1 -s 192.168.0.1 -j DROP
```

Ova komanda zamenjuje pravilo sa rednim brojem 1 drugim pravilom.

Komanda -I, --insert

```
iptables -I INPUT 1 --dport 80 -j ACCEPT
```

Ubacuje pravilo na (u ovom primeru) prvo mesto u lancu.

Komanda -L, --list

```
iptables -L INPUT
```

Izlistava sva pravila koja su definisana u naznačenom lancu, tj. svim lancima ukoliko se ne navede u kom.

Komanda -F, --flush

```
iptables -F INPUT
```

Ova komanda briše sva pravila iz naznačenog lanca, odnosno iz svih ukoliko nije navedeno u kom.

Komanda -Z, --zero

```
iptables -Z INPUT
```

Postavlja na nulu sve brojače u naznačenom lancu, odnosno u svim lancima ukoliko se ne navede u kom.

Komanda -N, --new-chain

```
iptables -N allowed
```

Ova komanda kreira novi lanac sa naznačenim imenom.

Komanda -X, --delete-chain

```
iptables -X allowed
```

Komanda briše lanaz iz tabele. Da bi to bilo ostvareno ni jedno pravilo kao

svoju akciju ne sme da ima ovaj lanac (da referencira na njega).

Komanda -P, --policy

```
iptables -P INPUT DROP
```

Ova komanda postavlja lancu podrazumevanu akciju (tj. polisu). Nad svim paketima koji se ne poklapaju ni sa jednim pravilom u lancu biće preduzeta ta podrazumevana akcija. Dozvoljene akcije su DROP i ACCEPT

Komanda -E, --rename-chain

```
iptables -E allowed disallowed
```

Menja ime lanca u naznačeno (allowed u disallowed u ovom primeru).

Ukoliko se ne unese cela komanda izlistaće se kratak opis sintakse komandi i ugrađena pomoć za korišćenje. Opcija -v se koristi za ispisivanje verzije programa, a opcija -h za ispisivanje kratkog opisa komandi i pomoći pri korišćenju.

Sledi spisak opcija koje se koristite sa komandama da bi se postigli pojedini efekti.

Opcija -v, --verbose

Koristi se sa komandama: **--list, --append, --insert, --delete, --replace**.

Ova komanda više informacija o onome šta se izvršava sa naznačenom komandom. Kada se koristi uz komandu --list biće ispisane mnoge dodatne informacije, kao što su: adresa interfejsa, opcije pravila, TOS maske, razne brojače i drugo.

Opcija -x, --exact

Koristi se sa komandama: **--list**.

Ova opcija se koristi sa komandom --list da bi brojači koji se ispisuju bili prikazani tačnim vrednostima, a ne skraćenim (K za hiljadu, M za milion i G za milijardu).

Opcija -n, --numeric

Koristi se sa komandama: **--list**.

Ova opcija naznačava da se ispisuju numeričke vrednosti IP adrese i porta, umesto podrazumevanog ispisivanja imena hosta, mreže i aplikacije koja se koristi. Ova opcija znatno ubrzava izlistavanje pravila kada ih ima mnogo, a neophodno ju je koristiti kada postoji problem u prevodenju IP adresa u hostname (npr. konekcija sa DNS-om nije u redu)

Opcija --line-numbers

Koristi se sa komandama: **--list**.

Ova opcija naznačava da izlaz komande --list sadrži i redne brojeve pravila u lancu.

Opcija -c, --set-counters

Koristi se sa komandama: **--insert**, **--append**, **--replace**.

Ova opcija se koristi kada se kreira novo pravilo ili se postojeće prepravlja da bi se inicijalizovali brojači na neke početne vrednosti.

Opcija --modprobe

Koristi se sa svim komandama.

Opcija se koristi da bi se naznačilo koji kernel modul iptables treba da učita

ukoliko već nije učitan.

4.8 Iptables provere pravila (rule matches)

Provere koje je moguće sprovesti pomoću programa iptables dele se u pet podkategorija.

U prvu spadaju sve generičke provere koji se mogu koristiti u svim pravilima. Zatim su tu 3 kategorije provera paketa pojedinih protokola i to provere TCP, UDP i ICMP paketa. Na kraju postoje i specijalne provere kao što su provere stanja konekcije, vlasnika, provera limita itd.

4.8.1 Generičke provere

Generičke provere su one koje su dostupne sa bilo kojim protokolom koji je u pitanju. Nije potrebno navoditi nikakve parametre da bi se koristile ove provere.

Provera -p, --protocol

```
iptables -A INPUT -p tcp
```

Ovim se proverava koji je protokol u pitanju. Mogu se koristiti tcp,udp,icmp za odgovarajuće protokole, numeričke označke protokola koje se mogu naći u fajlu /etc/protocols, ili ključna reč ALL koja označava zajedno tcp,udp i icmp protokole. Takođe se može koristiti i lista protokola, razdvojena zarezima. Provere se mogu i invertovati pomoću znaka uzvika, tako da --protocol ! tcp označava provere UDP and ICMP protokola. Podrazumevano je da se proveravaju TCP, UDP i ICMP protokoli.

Provera -s, --src, --source

```
iptables -A INPUT -s 192.168.1.1
```

Ovim se proverava da li se izvorišna IP adresa paketa poklapa sa navedenom. Mogu se navesti pojedinačne IP adrese, ili opsezi adresa u CIDR notaciji. Može se koristiti i znak uzvika radi invertovanja rezultata provere. Podrazumevano je da se proveravaju sve IP adrese.

Provera -d, --dst, --destination

```
iptables -A INPUT -d 192.168.1.1
```

Proverava se odredišna IP adresa paketa, na isti način kao i --source.

Provera -i, --in-interface

```
iptables -A INPUT -i eth0
```

Proverava se da li paketi dolaze na navedeni interfejs. Ova provera je jedino

dozvoljena u INPUT, FORWARD i PREROUTING lancima inače će iptables prijaviti grešku. Ukoliko nije naveden interfejs, podrazumeva se da se proveravaju paketi sa svih interfejsa. Ukoliko se navede npr. eth+ provera će se vršiti na svim Ethernet uređajima, obzirom da se "+" koristi kao džoker znak. Može se takođe i invertovati rezultat provere tako da -i ! eth0 označava sve interfejse osim eth0.

Provera -o, --out-interface

```
iptables -A FORWARD -o eth0
```

Proveravaju se paketi koji napuštaju eth0 interfejs. Ova provera je dostupna

jedino u OUTPUT, FORWARD i POSTROUTING lancima. Mogu se koristiti džoker znak + za uređaje istog tipa, i znak ! za invertovanje rezultata provere. Podrazumevano je da se posmatraju svi interfejsi.

Provera -f, --fragment

```
iptables -A INPUT -f
```

Ovom proverom se ispituju drugi i treći deo fragmentovanog paketa, obzirom da se kod fragmentovanih paketa ne mogu znati ni izvorišni ni odredišni port fragmenata, niti tipovi ICMP poruka i sl. Ukoliko se koristi inverzija, posmatraju se prvi fragment paketa kao i svi paketi koji nisu fragmentovani.

4.8.2 Implicitne provere

Implicitne provere su one provere koje su implicitno učitane u kernel tako da ih nije potrebno dodatno navoditi kao što je slučaj sa eksplisitim proverama. Postoje implicitne provere za TCP, UDP i ICMP protokole, svaka sa jedinstvenim kriterijumima dostupnim jedino za odgovarajuće protokole.

TCP provere

Ovim proverama se ispituju kriterijumi specifični za TCP protokol, pa su jedino dostupne kada se radi sa TCP paketima i tokom podataka. Da bi se koristile ove provere potrebno je navesti argument --protocol tcp pre navođenja kriterijuma provere.

Provera --sport, --source-port

```
iptables -A INPUT -p tcp --sport 22
```

Ova provera se vrši na osnovu izvorišnog porta. Može se navesti ili ime servisa koje se nalazi u fajlu /etc/services ili broj porta, što je brže ukoliko postoji veliki broj pravila. Takođe, može se navesti opseg portova navođenjem argumenta npr. --source-port 22:80, a može se

koristiti i invertovanje rezultata provere pomoću znaka užvika. Ukoliko je potrebno navesti veći broj portova koji nisu uzastopni mora se koristiti proširenje multiport.

Provera --dport, --destination-port

```
iptables -A INPUT -p tcp --dport 22
```

Ova provera se vrši na osnovu odredišnog porta. Korišćenje je analogno korišćenju provere --source-port

Provera --tcp-flags

```
iptables -p tcp --tcp-flags SYN,FIN,ACK SYN
```

Vrši se provera TCP flegova u paketu i to SYN, ACK, FIN, RST, URG i PSH. Mogu se navesti i ključne reči ALL i NONE. Prvi argument je lista flegova koji se porede (maska), a drugi lista flegova koji bi trebalo da su postavljeni na 1. Za razdvajanje se u listi koristi zarez, bez znakova razmaka. Dakle, provera --tcp-flags ALL NONE će proveriti sve flegove i biti ispunjena ukoliko ni jedan fleg nije postavljen na 1. Može se koristiti invertovanje provere tako da je --tcp-flags ! SYN,FIN,ACK SYN ispunjeno kod onih paketa koji umaju postavljene ACK i FIN flegove, a nemaju postavljen SYN.

Provera --syn

```
iptables -p tcp --syn
```

Služi za proveru da li je paketu postavljen SYN bit. Postoji radi kompatibilnosti sa ipchains, inače je ne treba koristiti, obzirom da se isti efekat može postići sa proverom --tcp-flags SYN,RST,ACK SYN

Provera --tcp-option

```
iptables -p tcp --tcp-option 16
```

Ovim argumentom se vrši provera TCP opcija koje su specificirane u zaglavju TCP paketa. Opcije se sastoje od 3 polja: prvo dužine 8 bita koje govori koje su opcije navedene u zaglavju, drugo takođe dužine 8 bita koje govori koje dužine je treće polje koje sadrži vrednosti opcija. Provera se vrši na osnovu decimalne vrednosti koja se navede. Moguće je koristiti i inverznu proveru, kako bi se proveravale sve opcije osim onih koje su navedene. Listu svih opcija koje se mogu navesti u zaglavju TCP paketa održava Internet Engineering Task Force, organizacija zadužena za objavljivanje standarda u vezi Interneta.

UDP provere

UDP provere se koriste jedino sa UDP paketima. Da bi se koristile dovoljno je navesti argument --protocol UDP i one će biti implicitno učitane. Navode se nakon tog prvog argumenta. Obzirom da je UDP protokol takav protokol da ne koristi konekcije, u paketima ne postoje flegovi koji govore kako se ti podaci koriste. Ukoliko je paket izgubljen ili oštećen ništa se ne preduzima kako bi se ponovo poslao. Zbog prirode UDP protokola postoji malo provera koje se mogu izvoditi. Treba znati da se statefull filtriranje ipak može obavljati, obzirom da tokovi UDP paketa označavaju sa NEW i ESTABLISHED prvim paketom koji se salje i njegovim odgovorom.

Provere koje se mogu vršiti na UDP paketima su:

Provera --sport, --source-port

```
iptables -A INPUT -p udp --sport 53
```

Ovom proverom proverava se izvorišni port preko kog je poslat paket, baš kao i analogna provera kod TCP konekcija. Može se navesti jedan port, rang portova i inverzija rezultata

provere. Osim decimalnih brojeva portova mogu se koristiti i imena servisa koji se obavljaju na tim portovima, onako kako je to navedeno u fajlu /etc/services . Ukoliko vam je potrebno da proveravate više portova koji nisu uzastopni moraćete da koristite ekstenziju multiport.

Provera --dport, --destination-port

```
iptables -A INPUT -p udp --dport 53
```

Ovom proverom proverava se odredišni port preko kog stiže paket, baš kao i analogna provera kod TCP konekcija. Može se navesti jedan port, rang portova i inverzija rezultata provere. Osim decimalnih brojeva portova mogu se koristiti i imena servisa koji se obavljaju na tim portovima, onako kako je to navedeno u fajlu /etc/services . Ukoliko vam je potrebno da proveravate više portova koji nisu uzastopni moraćete da koristite ekstenziju multiport.

ICMP provere

ICMP (Internet Control Message Protocol) je protokol za kontrolu konekcije i prijavljivanje grešaka. Program ping npr. šalje ICMP echo zahtev radi provere postojanja hosta sa tom IP adresom. Ovo je protokol bez konekcije a zaglavje ICMP paketa sadrži informacije o tipu ICMP poruke koja se šalje.

Postoji samo jedna provera –icmp-type za ispitivanje tipa poruke, i ona se implicitno učitava sa --protocol ICMP, ali se, pomoću generičkih provera, mogu vršiti i provere izvorišne i odredišne IP adresе i sl

Provera --icmp-type

```
iptables -A INPUT -p icmp --icmp-type 8
```

Ovom proverom proverava se tip ICMP poruke koja se šalje. Može se navesti broj ili naziv, koji se može dobiti komandom iptables --protocol icmp --help . Naravno, može se koristiti i invertovanje rezultata provere pomoću znaka “!”.

4.8.3 Eksplisitne provere

Eksplisitne provere su one koje je neophodno učitati sa -m ili --match argumentom. One nisu deo osnovnog skupa provera jer se ne koriste uvek, već za neke posebne primene. Neke od ovih provera zavise od protokola koji se koristi, a neke ne.

Provera Limit

Ovo je ekstenzija koja se mora učitati eksplisitno pomoću -m limit argumenta. Omogućava kontrolisanje broja ispunjavanja uslova provere u toku nekog vremenskog perioda.

Provera --limit

```
iptables -A INPUT -m limit --limit 3/hour
```

Ovim se podešava prosečan maksimalni broj puta ispunjavanja provere po jedinici vremena. Moguće je koristiti /second /minute /hour i /day intervale. Podrazumevana vrednost je 3/hour.

Provera --limit-burst

```
iptables -A INPUT -m limit --limit-burst 5
```

Maksimalni broj paketa koji se proveravaju: ovaj se smanjuje za 1 svaki put kada istekne vremenski interval naveden pod --limit opcijom za koji pravilo nije bilo ispunjeno, a povećava se za jedan (ako je manje od navedene vrednosti), svaki put kada se pravilo ispunji. Podrazumevana vrednost je 5.

Provera MAC adrese

Provera se eksplisitno učitava sa opcijom -m mac i omogućava proveravanje MAC adrese uređaja koji je poslao paket.

Provera --mac-source

```
iptables -A INPUT -m mac --mac-source 00:00:00:00:00:01
```

Ovom proverom proverava se MAC adresa uređaja koji je poslao paket. MAC adresa mora biti navedena u obliku XX:XX:XX:XX:XX:XX, inače će doći do greške. Rezultat provere može biti i invertovan korišćenjem znaka uzvika, npr. --mac-source ! 00:00:00:00:00:01, što bi bilo ispunjeno za sve uređaje koji imaju neku drugu MAC adresu. Provera je moguća samo na Ethernet tipovima interfejsa, a ispravno je koristiti ovu proveru jedino u PREROUTING, FORWARD i INPUT lancima.

4.8.4 Provera kernel oznaka

Linux kernel označava sve pakete koji prolaze kroz računar, radi obavljanja različitih rutina kao što su filtering i traffic shaping. Oznake se mogu postaviti kroz iptables preduzimanjem akcije MARK na paket (FWMARK u ipchains). Oznaka je pozitivan ceo broj. Važno je zapamtitи да je oznaka jedino važeće na računaru koji ih je postavio.

Provera --mark

```
iptables -t mangle -A INPUT -m mark --mark 1
```

Ovom proverom proverava se vrednost oznake koja je ranije bila postavljena. Može se koristiti i maska za logičku AND operaciju, tako što bi --mark 1/1 izvršilo AND operaciju sa maskom pre izvršenja poređenja.

4.8.5 Multiport provera

Ova ekstenzija se može koristiti da bi se navelo više portova koji nisu uzastopni, ili više rangova portova, da bi se izbeglo korišćenje više pravila istog tipa samo zbog različitih portova. Ne može se koristiti sa standardnim načinom za proveru portova.

Provera --source-port

```
iptables -A INPUT -p tcp -m multiport --source-port 22,53,80,110
```

Ovom proverom proverava se veći broj izvořišnih portova koji nisu uzastopni. Može se navesti maksimalno 15 portova, razdvojenih zarezom. Provera se može vršiti jedino sa TCP i UDP protokolima, tj. -p tcp i -p udp proverama.

Provera --destination-port

```
iptables -A INPUT -p tcp -m multiport --destination-port 22,53,80,110
```

Ovom proverom proverava se veći broj odredišnih portova koji nisu uzastopni. Može se navesti maksimalno 15 portova, razdvojenih zarezom. Provera se može vršiti jedino sa TCP i UDP protokolima, tj. -p tcp i -p udp proverama.

Provera --port

```
iptables -A INPUT -p tcp -m multiport --port 22,53,80,110
```

Ovom proverom proverava se više portova istih i na izvořišnom i odredišnom hostu. Može se navesti maksimalno 15 portova, razdvojenih zarezom. Provera se može vršiti jedino sa TCP i UDP protokolima, tj. -p tcp i -p udp proverama.

4.8.6 Provera vlasnika

Ova ekstenzija proverava koji proces je generisao pakete. Provera se može vršiti na osnovu PID-a procesa, UID-a korisnika koji je pokrenuo proces ili GID-a grupe kojoj pripada korisnik koji je pokrenuo proces. Moguće ju je primeniti jedino u OUTPUT lancu. ICMP paketi gotovo nikada nemaju vlasnika. Ovim se postiže npr. da se onemogući svima osim root-u da otvaraju spoljne konekcije, ili svima osim korisniku http da šalju pakete kroz HTTP port.

Provera --uid-owner

```
iptables -A OUTPUT -m owner --uid-owner 500
```

Ovom proverom proverava se da li je korisnik čiji UID je naveden pokrenuo proces koji je generisao paket.

Provera --gid-owner

```
iptables -A OUTPUT -m owner --gid-owner 0
```

Ovom proverom proverava se da li korisnik koji je pokrenuo proces koji je generisao paket pripada grupi čiji je GID naveden.

Provera --pid-owner

```
iptables -A OUTPUT -m owner --pid-owner 78
```

Ovim se proverava da li je paket generisao proces čiji je PID naveden.

Provera --sid-owner

```
iptables -A OUTPUT -m owner --sid-owner 100
```

Ovim se proverava da li proces koji je generisao paket ima odgovarajući SID (Session ID). Neophodno je koristiti proveru SID-a kod procesa koji su multitredovani.

4.8.7 Provera stanja paketa

Ova ekstenzija se koristi za proveru stanja paketa. Radi za sve protokole koji održavaju konekciju (dakle ne radi za ICMP i UDP). Potrebno je eksplicitno učitati ekstenziju pomoću -m state.

Provera --state

```
iptables -A INPUT -m state --state RELATED,ESTABLISHED
```

Ovom proverom proverava se da li se konekcija nalazi u navedenom stanju. Postoje 4 stanja u kojima može biti konekcija: INVALID, ESTABLISHED, NEW i RELATED. Ukoliko se navodi više stanja, neophodno ih je razdvojiti zarezom.

4.8.8 Provera polja TOS

Polje TOS u zaglavljtu paketa sadrži informacije o tipu servisa i dužine je 8 bita. Predstavlja način za označavanje paketa radi boljeg rutiranja i sl. Provera ovog polja se eksplicitno učitava navođenjem -m tos.

Provera --tos

```
iptables -A INPUT -p tcp -m tos --tos 0x16
```

Ovom proverom proverava se ToS polje u IP zaglavljtu paketa. Navodi se heksadecimalna vrednost ili naziv servisa koji se može pronaći u izlazu komande 'iptables -m tos -h'.

4.8.9 Provera polja TTL

Polje TTL (Time To Live) sadrži informacije o dužini validnosti paketa. Nalazi se u IP zaglavljiju paketa i dužine je 8 bita. Vrednost ovog polja se smanjuje svaki put kada prelazi sa jednog na drugi host na putu do odredišta. Kada dođe do vrednosti 0, pošiljaocu se šalje ICMP paket tipa 11 informišući ga da paket nije stigao do odredišta. Provera polja TTL se eksplisitno učitava sa -m ttl.

Provera --ttl

```
iptables -A OUTPUT -m ttl --ttl 60
```

Ovom proverom proverava se da li je polje TTL u zaglavljiju paketa jednako navedenoj vrednosti.

4.8.10 Provera ispravnosti paketa

Provera ispravnosti paketa se koristi kada želite da proverite ispravnost

paketa na osnovu njihovog polja Checksum. Nije potrebno navoditi bilo kakve druge opcije osim eksplisitnog učitavanja sa -m unclean. Neki protokoli su takve prirode da ne rade dobro sa ovom ekstenzijom, pa treba biti obazriv prilikom korišćenja.

4.9 Iptables AKCIJE (targets/jumps)

Nakon što se neko pravilo zadovolji, nad paketom se sprovodi određena aktivnost. To može akcija kao što je ACCEPT, DROP ili REJECT, ili preusmeravanje na neki drugi lanac radi dalje obrade. Ukoliko u tom lancu ni jedno pravilo ne bude ispunjeno, paket će biti vraćen u prvočitni lanac radi dalje analize.

4.9.1 Akcija ACCEPT

ACCEPT je aktivnost koja se preduzima nad paketom koji se želi propustiti kroz firewall. Nakon obavljanja ove aktivnosti, paket se dalje ne ispituje i izlazi iz svih lanaca kroz koje je prolazio u trenutno ispitivanoj tabeli (ali prolazi kroz lance u drugim tabelama). Nije potrebno navoditi bilo kakve druge opcije uz ove aktivnosti osim -j ACCEPT.

4.9.2 Akcija DROP

Aktivnost DROP se sprovodi nad onim paketima kojima se ne dozvoljava da prođu kroz firewall. Ne sprovodi se obaveštavanje izvorišnog i odredišnog hosta o tome da je paket odbijen kao što to radi akcija REJECT. To može značiti da će na izvorišnom hostu ostati otvorena konekcija, pa treba razmatrati u kojim situacijama to treba izbeći.

4.9.3 Akcija REJECT

Aktivnost REJECT sprovodi blokiranje paketa, kao što to radi aktivnost DROP, ali šalje poruku da je paket odbijen hostu koji ga je poslao. Dozvoljeno je koristiti akciju REJECT jedino u INPUT, FORWARD i OUTPUT lancima ili lancima na koje oni referenciraju (oni na koje se paketi preusmeravaju iz ovih lanaca).

Pomoću parametra --reject-with može se navesti razlog zbog kog je paket odbijen.

Opcija --reject-with

```
iptables -A FORWARD -p TCP --dport 22 -j REJECT --reject-with tcp-reset
```

Ovaj parametar se prosleđuje akciji REJECT i predstavlja odgovor koji će biti poslat hostu od kog paket počinje kao razlog zašto je paket odbijen. Moguće je navesti sledeće razloge: icmp-net-unreachable, icmp-host-unreachable, icmp-port-unreachable, icmp-proto-unreachable,

icmp-net-prohibited i icmp-host-prohibited. Podrazumevano je da se šalje poruka port-unreachable . Sve ove poruke su ICMP poruke o grešci. Ukoliko je u pitanju TCP protokol, može se poslati i poruka tcp-reset koja će vratiti TCP RST paket kao odgovor na uspostavljanje konekcije, kako bi konekcija bila zatvorena i na izvorišnom hostu.

4.9.4 Aktivnost LOG

Ukoliko je potrebno zabeležiti informacije o paketima koji prolaze kroz firewall i ispušnjavaju određene kriterijume, nad paketima se može preduzeti aktivnost LOG, navodeći -j LOG. Informacije će biti zapisane u sistemski log, tako da se mogu čitati komandom dmesg. Postoji pet opcija koje se mogu prolsediti ovoj aktivnosti da bi se postigli različiti efekti.

Opcija --log-level

```
iptables -A FORWARD -p tcp -j LOG --log-level debug
```

Ovom opcijom se navodi koji log nivo će iptables koristiti kada bude zapisivao informacije u syslog. Kompletna lista nivoa se može naći u man strani syslog.conf fajla, a neki od njih su: debug, info, notice, warning, warn, err, error, crit, alert, emerg, panic....

Opcija --log-prefix

```
iptables -A INPUT -p tcp -j LOG --log-prefix "INPUT packets"
```

Ovom opcijom se syslog logovi označavaju stavljanjem navedenog parametra ispred informacija koje se zapisuju kako bi bilo lako pronalaženje tih informacija npr. korišćenjem komande grep. Parametar može biti dugačak 29 znakova, uključujući i beline.

Opcija --log-tcp-sequence

```
iptables -A INPUT -p tcp -j LOG --log-tcp-sequence
```

Omogućava zapisivanje u log TCP Sekvencnog broja iz zaglavlja paketa zajedno sa ostalim informacijama.

Opcija --log-tcp-options

```
iptables -A FORWARD -p tcp -j LOG --log-tcp-options
```

Omogućava zapisivanje u log mnogih dodatnih informacija iz zaglavlja TCP paketa koje mogu biti korisne prilikom rešavanja različitih problema.

Opcija --log-ip-options

```
iptables -A FORWARD -p tcp -j LOG --log-ip-options
```

Omogućava zapisivanje mnogih dodatnih informacija iz IP zaglavlja paketa, koje mogu biti korisne prilikom otklanjanja greški.

4.9.5 Aktivnost MARK

Ovom aktivnošću se postavljaju MARK vrednosti koje su unutar kernela pridružene paketima. Moguće ju je koristiti jedino u MANGLE tabeli.

Opcija --set-mark

```
iptables -t mangle -A PREROUTING -p tcp --dport 22 -j MARK --set-mark 2
```

Označava paket navedenom celobrojnom vrednošću. Obzirom da oznaka nije deo zaglavlja paketa biće vidljiva jedino na sistemu koji ju je postavio.

4.9.6 Aktivnost TOS

Polje TOS u IP zaglavljtu paketa se koristi prilikom rutiranja sa iproute2 da bi se odabrala optimalna ruta kojom paket treba da se pošalje. Za razliku od aktivnosti MARK – koja paket označava nekom vrednošću samo unutar kernela, TOS polje je sastavni deo IP zaglavlja pa se može preneti drugom hostu i biti tamo procesirana. Vrednosti koje se mogu upisati u TOS polje IP zaglavlja su definisane u Linux/ip.h kernel include fajlu, a mogu se dobiti komandom iptables -j TOS -h.

Ova aktivnost je jedino validna u okviru MANGLE tabele.

Opcija --set-tos

```
iptables -t mangle -A PREROUTING -p TCP --dport 22 -j TOS --set-tos 0x10
```

Ova opcija postavlja TOS vrednost IP zaglavlja na navedenu vrednost. Vrednost može biti numerička, bilo decimalna, bilo heksadecimalna ili po nazivu servisa. Najčešće korišćene vrednosti su: Minimize-Delay, Maximize-Throughput, Maximize-Reliability, Minimize-Cost i Normal-Service. Podrazumevana vrednost je Normal-Service.

4.9.7 Aktivnost TTL

Ovom aktivnošću se može prepravljati TTL (Time To Live) vrednost polja u IP zaglavlju paketa. Aktivnost je jedino dozvoljena u MANGLE tabli, a ima 3 moguće opcije.

Opcija --ttl-set

```
iptables -t mangle -A PREROUTING -i eth0 -j TTL --ttl-set 64
```

Ova opcija postavlja TTL polje na navedenu vrednost. Najoptimalnije vrednosti su oko 64, a treba izbegavati postavljanje ove vrednosti suviše velike, jer to može loše uticati na performanse mreže ukoliko dođe do vraćanja paketa.

Opcija --ttl-dec

```
iptables -t mangle -A PREROUTING -i eth0 -j TTL --ttl-dec 1
```

Ova opcija smanjuje vrednost TTL polja za navedenu vrednost. Treba voditi računa da će TTL vrednost u svakom slučaju biti smanjena za jedan, pa bi u gornjem primeru vrednost TTL polja bila smanjena za 2 po izlasku iz hosta.

Opcija --ttl-inc

```
iptables -t mangle -A PREROUTING -i eth0 -j TTL --ttl-inc 1
```

Ovom opcijom se povećava vrednost polja TTL navedenom vrednošću. Treba imati na umu da će TTL vrednost pri prolasku kroz naš host biti smanjena za jedan, pa bi u gornjem primeru vrednost TTL polja po izlasku iz hosta ostala ista kao i pri ulasku, čime se naš host efektivno sakriva od programa tipa traceroute, što može biti dobra stvar sa stanovišta sigurnosti, ali loša stvar što se tiče otklanjanja grešaka u radu sa mrežom.

4.9.8 Aktivnost REDIRECT

Ovom aktivnošću se paketi mogu preusmeriti na neki drugi port iste mašine. Koristi se npr. da se svi paketi namenjeni HTTP portovima preusmere transparentnom HTTP proxy serveru na lokalnoj mašini.

Aktivnost je jedino moguće preduzeti u okviru PREROUTING i OUTPUT lanca NAT tabele i korisnički-definisanim lancima koje oni referenciraju. Postoji jedan parametar koji se može proslediti ovoj aktivnosti.

Opcija --to-ports

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8080
```

Ovim parametrom se određuje odredišni port na koji će paketi biti preusmereni. Ako se ovaj parametar ne navede, biće korišćen isti port. Moguće je navesti i opseg portova u obliku --to-ports 8080-8090. Opcija je jedino validna kada su u pitanju TCP i UDP protokoli.

4.9.9 Aktivnost MIRROR

Ova aktivnost služi za eksperimentisanje i demonstraciju i ne treba je koristiti u druge svrhe. Njome se invertuju izvorišna i odredišna adresa u IP zaglavljaju paketa. Aktivnost je dozvoljena jedino unutar INPUT, FORWARD i PREROUTING lancima i drugim korisnički-definisanim lancima koje ovi lanci referenciraju.

4.9.10 Aktivnost QUEUE

Ova aktivnost se koristi da bi se paketi prosledili zasebnim programima operativnog sistema, npr. radi vođenja dnevnika aktivnosti ili naplaćivanja usluga – dakle ono izlazi van okvira aktivnosti Netfilter paketa.

4.9.11 Aktivnost RETURN

Ova aktivnost će izazvati završetak proveravanja pravila u lancu u kom se nalazi. Ukoliko je u pitanju korisnički-definisan lanac, paket će biti vraćen na analizu originalnom lancu iz kog je preusmeren na taj lanac, a ukoliko je u pitanju glavni lanac (npr. INPUT) nad paketom će biti preduzeta podrazumevana aktivnost (default policy).

4.10 Postupak Masquerading-a

Masquerading je specifični tip NAT-a (network address translation – prevodenja mrežnih adresa). Može se koristiti da bi se računari sa privatnim IP adresama povezali na Internet preko jednog mrežnog uređaja (router-a ili firewall-a) sa dva mrežna interfejsa, od kojih jedan ima privatnu a drugi javnu IP adresu. Računari privatnog LAN-a će pakete koji su namenjeni ka spoljnoj mreži slati na default gateway, koji će biti podešen da bude privatna IP adresa router-a (tj. firewall-a). On će tim paketima vršiti prevođenje izvorišne IP adrese (source network address translation – sNAT) i te pakete preusmeravati na svoj interfejs sa javnom IP adresom, dakle na spoljnu mrežu. Kada se paket vratí, router će znati koji je paket namenjen za koji privatni host, zatim izvršiti prevođenje odredišne mrežne adrese (destination network address translation) i preusmeriti taj paket sa javnog interfejsa na privatni, tj. ka unutrašnjoj mreži. Što se tiče odredišnog hosta na javnoj mreži, on ne zna ništa o privatnoj mreži i hostovima u njima, već jedino zna da je paket stigao od router-a (tj. firewall-a).

Na ovaj način jedino hostovi iz privatne mreže mogu da uspostavljaju konekcije, obzirom da spoljni host ne može da adresira privatne IP adrese, a u masquerading tabeli postoji jedino zapisi o konekcijama koje su zatražili hostovi iz unutrašnje-privatne mreže.

4.10.1 Aktivnost DNAT

Opcija --to-destination

```
iptables -t nat -A PREROUTING -p tcp -d 15.45.23.67 --dport 80 \
          -j DNAT --to-destination 192.168.1.1-192.168.1.10
```

Ova opcija nagoveštava IP adresu koju DNAT mehanizam treba da upiše u zaglavje paketa. Ako se navede opseg adresa, paketi koji stignu na firewall će biti prosleđeni nekom nasumično

odabranom hostu sa IP adresom iz navedenog opsega. Ovim se na jednostavan način može postići balansiranje opterećenja hostova radi poboljšanja performansi. Takođe se može navesti i port ili opseg portova na koje se žele preusmeravati paketi.

4.10.2 Aktivnost SNAT

Opcija --to-source

```
iptables -t nat -A POSTROUTING -p tcp -o eth0 \
          -j SNAT --to-source 194.236.50.155-194.236.50.160:1024-32000
```

Ova opcija nagoveštava IP adresu koju SNAT mehanizam treba da upiše u zaglavlje paketa. Ako se navede opseg adresa, biće korišćena nasumično odabrana IP adresa iz tog opsega. Ukoliko dva ili više hostova pokuša da koristi isti port, iptables će mapirati nekog od njih na neki drugi port iz opsega.

4.10.3 Aktivnost MASQUERADE

Opcija --to-ports

```
iptables -t nat -A POSTROUTING -p TCP -j MASQUERADE --to-ports 1024-31000
```

Ova opcija postavlja vrednosti portovar koje mogu biti iskorišćene za odlazeće pakete. Može se koristiti jedino sa TCP i UDP protokolima.

4.11 Implementacija firewall-a na RedHat/CentOS distribucijama

Za razliku od Debian distribucija, RedHat/CentOS distribucije imaju servis 'iptables' preko kojeg se može uključiti ili isključiti paketski filter. Ovaj servis je podrazumevano uključen. Ovaj servis implementira perzistentnost pravila tako što ih čuva u fajlu /etc/sysconfig/iptables.

Startovanje prethodno zaustavljenog 'iptables' servisa se obavlja komandom:

```
# service iptables start
```

Ova komanda će učitati prethodno snimljena pravila iz /etc/sysconfig/iptables i učitati odgovarajuće kernel module ukoliko nisu učitani.

Zaustavljanje 'iptables' servisa se obavlja komandom:

```
# service iptables stop
```

Ova komanda će prvo snimiti postojeća pravila a zatim obrisati sve korisnički kreirane lance, obrisati sva pravila iz ugrađenih lanaca i postaviti polise na ACCEPT, tako da je mrežni pristup u potpunosti omogućen.

Privremeno snimanje pravila se obavlja komandom:

```
# service iptables save
```

koja će snimiti pravila u /etc/sysconfig/iptables.

4.12 Vežbe

1. Konfigurisati netfilter servis tako da dozvoljava pristup serveru:

- preko 80/tcp sa svih adresa
- preko porta 22/tcp sa lokalne mreže

2. Konfigurisati netfilter servis tako da server izigrava ivični firewall neke mreže koja u unutrašnjem delu ima adrese iz privatnog opsega a u spoljašnjem delu ima dinamički dodeljenu adresu. Od spolja treba propustiti port 22/tcp na sam server i port 80/tcp koji treba proslediti na server u unutrašnjem delu mreže. Adrese dodelite sami.

Property of Admin Training Center

Modul 5: Kontrola pristupa u Linuxu

Standardna kontrola pristupa u Linuxu je bazirana na sledećim premisama:

- procesi se izvršavaju pod UID-om i GID-om (realnim i efektivnim);
- objekti pripadaju određenom UID-u i GID-u i imaju pristupne dozvole (čitanje, pisanje i izvršavanje za vlasnika, grupu i ostale);
- pristup se dozvoljava ili odbija na osnovu pristupnih dozvola za korisnika, grupu kojoj resurs pripada i ostatak sveta;
- korisnici određuju ko sme da pristupi njihovim objektima;
- sistem ne razlikuje korisnika od njegovog procesa.

Primer:

Veb čitač koga korisnik pokreće se izvršava pod UID-om i GID-om tog korisnika. Kompromitovan veb čitač ne može da naudi sistemu jer ni korisnik to ne može da uradi. ALI: kompromitovan veb čitač može da pročita sve fajlove korisnika (npr. privatni ssh ključ)

Problemi sa standardnom kontrolom pristupa u Linuxu su:

- standardna kontrola pristupa je diskretna
- ne postoji koncept vlasništva resursa
- procesi mogu promeniti pristupne dozvole
- root korisnik može zaobići sigurnosne mehanizme
- nije lako pružiti samo minimum privilegija

Da bi se rešili ovi problemi, Američka nacionalna agencija za bezbednost (NSA) je implementirala SELinux kao niz prepravki Linux kernela koje omogućavaju finiju kontrolu. Red Hat je nastavio razvoj i SELinux je sada standardni deo kernela, podržan od strane većine distribucija.

5.1 SELinux koncepti

SELinux implementira MAC (*Mandatory Access Control*, obavezujuća kontrola pristupa) sa sledećim karakteristikama:

- procesi i resursi imaju pridružene sigurnosne kontekste
- sigurnosnom polisom se definišu veze između konteksta
- sigurnosnu polisu definiše administrator, a primenjuje kernel — korisnici je ne mogu zaobići

Osobine SELinux kontrole pristupa su da je obavezna, da omogućava definisanje minimuma privilegija, da je fino granulirana, modularna i da root korisnik nije svemoguć, kao kod standardne Linux kontrole pristupa. Takođe, SELinux je transparentan za većinu aplikacija, ali neke aplikacije još uvek nisu podešene da rade sa sistemom sa uključenim SELinuxom.

Ove osobine su implementirane kroz tri forme kontrole pristupa:

- *Type Enforcement* (TE, forsiranje tipova)
- *Role Based Access Control* (RBAC, kontrola pristupa bazirana na rolama)
- *Multi Level Security* (MLS, višeslojna sigurnost)

Kontrola pristupa se definiše polisom. Primeri SELinux polisa na RedHat/CentOS sistema su: Targeted, Strict i MLS. Sav pristup se podrazumevano odbija, eksplicitno se navodi što treba da se odobri.

5.2 SELinux sigurnosni kontekst

SELinux definiše *subjekte* (entitet koji vrši neku radnju) i *objekte* (entiteti nad kojima se radnja vrši). U SELinux implementaciji, subjekti su procesi, a objekti su fajlovi, direktorijumi i ostali resursi. Osnovni mehanizam SELinuxa definiše kako subjekti pristupaju objektima i kako subjekti interaguju međusobno.

Kao osnova za navedeni mehanizam služi kontekst. Kontekst je labela koja je eksplicitno dodeljena svakom subjektu i objektu. Kontekst nekog objekta ili subjekta možete očitati standardnim alatima:

```
$ ls -Z /etc/passwd
-rw-r--r--. root root system_u:object_r:etc_t:s0  /etc/passwd

$ ps aux | head -3
LABEL PID TTY STAT TIME COMMAND
system_u:system_r:init_t:s0 1 ? Ss 0:03 /sbin/init
system_u:system_r:kernel_t:s0 2 ? S< 0:00 [kthreadd]

$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

$ netstat -antZ
```

Novi fajlovi dobijaju kontekst direktorijuma:

```
# ls -Z /etc/fstab; cp /etc/fstab /tmp; ls -Z /tmp/fstab
-rw-r--r--. root root system_u:object_r:etc_t:s0  /etc/fstab
```

```
-rw-r--r--. root root unconfined_u:object_r:user_tmp_t:s0 /tmp/fstab
```

Pomereni fajlovi zadržavaju kontekst:

```
# ls -Z /tmp/fstab; mv /tmp/fstab ~; ls -Z ~/*fstab
-rw-r--r--. root root unconfined_u:object_r:user_tmp_t:s0 /tmp/fstab
-rw-r--r--. root root unconfined_u:object_r:user_tmp_t:s0 /root/fstab
```

Komandom 'install' fajlovi dobijaju kontekst koji im je definisan u polisi.

SELinux kontekst ima sledeću sintaksu:

korisnik:uloga:tip:nivo

korisnik je SELinux korisnik i on je polisom autorizovan za specifičan skup uloga i specifičan MLS opseg. SELinux korisnik je različit entitet u odnosu na korisnika na sistemu. Mapiranje korisnika na sistemu u SELinux korisnike je definisano SELinux polisom.

uloga je atribut koji se koristi za RBAC. Polisa određuje koji SELinux korisnici su autorizovani za koje uloge, i koje uloge su autorizovane za koje domene. Na osnovu uloga se definiše kako subjekt može pristupiti nekom objektu.

tip je atribut TE. Tip definiše domen procesima i tip fajlovima. Polisa određuje na koji način tipovi pristupaju jedni drugima, što se, u stvari, preslikava na to kako subjekt sa datim tipom pristupa objektu sa datim tipom i kako dva subjekta sa zadatim tipovima mogu da interaguju međusobno. Pristup je moguć samo ako postoji odgovarajuće pravilo unutar polise koje taj pristup dozvoljava.

nivo je atribut MLS i definiše kojim nivoima osetljivosti neki proces može prići, odnosno kojim nivoima osetljivosti neki objekt pripada.

5.3 Tranzicija domena

Procesi mogu da prelaze iz domena u domen i to je striktno definisano polisom. Tranzicija domena se dešava kada proces sa nekim domenom izvršava aplikaciju koja ima entrypoint dozvolu za neki drugi domen. Prilikom izvršavanja aplikacije proces iz svog tekućeg domena prelazi u domen koji je definisan polisom. Primer je promena lozinke:

1. da bi promenio svoju lozinku korisnik mora da izvrši komandu `passwd`
2. fajl `/usr/bin/passwd` ima 'passwd_exec_t' tip
3. komanda `passwd` mora da pristupi fajlu `/etc/shadow` koji ima tip 'shadow_t'
4. u polisi stoji da:
 - domen 'passwd_t' ima pravo pisanja i čitanja fajlova koji imaju tip 'shadow_t'
 - polisa definiše da domen 'passwd_exec_t' ima entrypoint dozvolu za domen 'passwd_t'
5. kada korisnik izvrši komandu `passwd`, domen njegovog procesa prelazi u 'passwd_t' i na kada pokuša da pročita i/ili izmeni fajl `/etc/shadow`, SELinux će mu to dozvoliti jer 'passwd_t' domen ima pravo pisanja i čitanja domena 'shadow_t'.

Sigurnosni kontekst fajlova i direktorijuma se čuva u ekstended atributima fajl sistema (ext2,3,4, raiserfs...) Sigurnosni kontekst se dodeljuje labelovanjem fajl sistema. Novi fajlovi nasleđuju kontekst direktorijuma u kom se prave.

Procesi svoj kontekst nasleđuju od roditelja ili prelaze u drugi domen ako je tako definisano polisom.

5.4 Polise i konfiguracioni fajlovi

5.4.1 Strict polisa

Strict polisa zabranjuje sve što nije eksplisitno definisano. SELinux je dizajniran sa strict polisom na umu, teško se primenjuje na operativnom sistemu opšte namene (npr. desktop radna stаница). MLS (Multi Layer Security) polisa omogućava ograničenje pristupa objektima prema nivoima poverljivosti informacija koje ti objekti čuvaju.

5.4.2 Targeted polisa

Targeted polisa je podrazumevana polisa na RedHat/CentOS sistemima. Procesi su podrazumevano neograničeni, ali polisa ograničava odabrane servise (zato targeted, ciljana). Podrazumevani tip je unconfined_t za korisničke programe, a initrc_t za sistemske procese. Unconfined procesi se izvršavaju (skoro) kao da je SELinux onemogućen. Tranzicija iz unconfined_t u tip definisan polisom se vrši automatski prilikom pokretanja na način definisan polisom.

SELinux polise se čuvaju u podstablu /etc/selinux. /etc/selinux/config definiše polisu i režim rada:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing   SELinux security policy is enforced.
#       permissive  SELinux prints warnings instead of enforcing.
#       disabled    No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#       targeted    Only targeted network daemons are protected.
#       strict      Full SELinux protection.
SELINUXTYPE=targeted
```

Parametri komandne linije kernela su važniji:

selinux = 0 — onemogućava SELinux i potrebno je ponovo labelovati fajlove kada se SELinux ponovo omogući.

enforcing = 0 — startuje permissive režim. Novi fajlovi će biti ispravno labelirani. Ono što bi SELinux zabranio se beleži u logu.

Polise se nalaze unutar /etc/selinux/ direktorijuma i sadrže poddirektorijume:

contexts — sadrži definicije konteksta za objekte

policy — sadrži binarnu (kompajliranu) polisu

modules — sadrži module koji se uključuju u polisu

Targeted polisa se nalazi u \$TARGETED/ (u daljem tekstu \$TARGETED). Konteksti fajlova se nalaze u \$TARGETED/contexts/files/:

file_contexts — osnovni konteksti

file_contexts.local — lokalne prepravke konteksta

file_contexts.homedir — korisnički fajlovi

media — uklonjivi uređaji (CD, DVD, USB disk...)

Polise zaštićenih servisa imaju svoje man strane:

```
apropos selinux | grep _selinux
ftpd_selinux  (8) - Security-Enhanced Linux policy for ftp daemons
httpd_selinux (8) - Security Enhanced Linux Policy for the httpd
named_selinux (8) - Security Enhanced Linux Policy for named
nfs_selinux   (8) - Security Enhanced Linux Policy for NFS
pam_selinux   (8) - PAM module to set the default security context
rsync_selinux (8) - Security Enhanced Linux Policy for the rsync
samba_selinux (8) - Security Enhanced Linux Policy for Samba
ypbind_selinux (8) - Security Enhanced Linux Policy for NIS
```

5.5 SELinux alatke

5.5.1 Paket coreutils

chcon menja kontekst navedenog fajla

5.5.2 Paket libselinux-utils

getenforce vraća status SELinux-a i režim rada

setenforce postavlja režim rada

selinuxenabled za skripte

getsebool vraća vrednosti promenljivih

togglesebool menja vrednost promenljive

matchpathcon vraća podrazumevani sigurnosni kontekst za navedenu putanju ili proverava (-V) da li je kontekst dobar

5.5.3 Paket policycoreutils

sestatus prikazuje status SELinux-a

semodule učitava modul u polisu

setsebool postavlja vrednosti promenljivih

genhomedircon generiše file_contexts.homedir

setfiles vrši inicijalno labelovanje fajl sistema

restorecon postavlja kontekste prema polisi

fixfiles pametno postavljanje konteksta

restorecond daemon koji pazi na kontekste

5.5.4 Paket checkpolicy

checkmodule kompajlira modul

checkpolicy kompajlira polisu

5.5.5 Paket policycoreutils-python

semanage vrši upite i menjanje polise

audit2allow generiše allow pravila na osnovu logova odbijenih aktivnosti

5.5.6 Paketi setroubleshoot i setroubleshoot-server

setroubleshootd pretražuje i obrađuje AVC poruke

sealert prikazuje objašnjenja AVC poruka

seapplet uznemirava korisnika obaveštenjima

5.6 Označavanje fajlova

Kontekst fajla se može promeniti chcon komandom koja je nalik chmod komandi:

```
# chcon -R -t httpd_sys_content_t /srv/www
```

Ovakve promene će važiti do ponovnog labelovanja.

Komanda 'restorecon' postavlja kontekste kako je definisano polisom.

```
# restorecon [opcije] putanja...
```

Ako nije navedeno -F ignorišu se tipovi koji su navedeni u \$TARGETED-contexts/customizable_types. Sa parametrima -n i -vv samo prijavljuje šta bi promenio. Ako postoji fajl /.autorelabel fajl sistem će biti labelovan prilikom startovanja sistema.

5.7 Označavanje fajlova

Fajlom \$TARGETED-contexts/files/file_contexts polisa definiše kontekste fajlova. Koriste se regularni izrazi za naznačavanje putanja.

Oznaka '--' u drugoj koloni označava samo fajlove, '-d' direktorijume, '-b' blok uređaje, '-c' karakter uređaje, '-l' linkove i sl.

Specijalni kontekst «none» označava da kontekst ne treba da se menja. Primeri:

```
/usr/(s)?bin/[xgkw]dm -- system_u:object_r:xdm_exec_t:s0
```

Postavlja xdm_exec_t tip za xdm, gdm, kdm i wdm programe koji se nalaze u /usr/sbin ili /usr/bin direktorijumima.

```
/mnt(/[!]*|? -d system_u:object_r:mnt_t:s0  
/mnt/[!]*/* <<none>>
```

Direktorijumi u /mnt dobijaju kontekst, ali njihovi sadržaji ne.

Komanda 'genhomedir' formira fajl \$TARGETED-contexts/files/file_contexts.homedirs na osnovu početnog direktorijuma korisnika. Prijaviće greške ukoliko je početni direktorijum korisnika na neuobičajenim lokacijama. U tom slučaju potrebno je ručno dodati kontekste u fajl \$TARGETED-contexts/files/file_contexts.local.

5.8 Log fajlovi

SELinux poruke (AVC – access vector cache) se beleže u fajl /var/log/messages. Ako je pokrenut servis 'auditd' kompletne AVC poruke se beleže u /var/log/audit/audit.log. Objašnjenja AVC poruka se mogu dobiti komandom 'sealert'. 'Setroubleshootd' analizira poruke, a može i da pošalje mail. Podešavanja za setroubleshootd su u /etc/setroubleshoot/setroubleshoot.cfg.

AVC poruke se javljaju zbog:

- nepravilno označenih fajlova
- procesa koji se izvršava sa pogrešnim kontekstom
- nedovršene polise
- upada na sistem

5.8.1 Nepravilno označeni fajlovi

Poruke u vezi fajlova sa tipom file_t se javljaju kada ima neoznačenih fajlova:

- fajlovi koji su napravljeni dok je SELinux bio onemogućen
- fajlovi na novom disku
- poruke u vezi direktorijuma sa tipom default_t
- fajlovi koji su pomereni komandom mv

Najčešće rešenje za ove probleme je zadavanje komande

```
# restorecon -R -F /
```

AVC poruke se mogu ignorisati u polisi (dontaudit pravilo). To može praviti probleme kada ispitujemo zašto neki program ne radi kada je SELinux u enforcing režimu. Da bismo onemogućili dontaudit pravilo potrebno je učitati enableaudit modul:

```
# semodule b /usr/share/selinux/targeted/enableaudit.pp
```

5.9 Prilagođavanje polisa

Polise često uključuju promenljive čije vrednosti se mogu promeniti bez prepravljanja same polise.

Komanda 'getsebool' prikazuje vrednosti promenljivih. Komanda 'toggesebool' će promeniti vrednost promenljive. Komandom 'setsebool' sa parametrom -P se vrednost može trajno promeniti. Man strane polisa daju detaljna objašnjenja promenljivih.

Komanda 'semanage' može promeniti parametre polise:

Pregled SELinux korisnika:

```
# semanage user -l
```

Podešavanje da Apache služi sadržaje iz /web direktorijuma:

```
# semanage fcontext -a -t httpd_sys_content_t "/web(/.*)?"
```

Podešavanje da Apache može da sluša na portu 81:

```
# semanage port -a -t http_port_t -p tcp 81
```

Isključivanje blokiranja Apache procesa:

```
# semanage permissive -a httpd_t
```

Uključivanje blokiranja Apache procesa:

```
# semanage permissive -d httpd_t
```

5.10 Dodatni moduli polise

Počev od RHEL 5 i Fedora Core 5 SELinux koristi modularne polise. U polisu se mogu učitati moduli tokom rada. Polisa će biti kompajlirana u letu.

Komanda 'audit2allow' omogućava kreiranje modula na osnovu log fajla sa AVC porukama.

Komanda:

```
# semodule -l
```

daje spisak učitanih modula.

Komanda:

```
# semodule -b /usr/share/selinux/targeted/enableaudit.pp
```

omogućava beleženje svih poruka (onemogućava dontaudit).

Komanda:

```
# semodule -b /usr/share/selinux/targeted/base.pp
```

omogućava dontaudit pravila.

Komanda:

```
# semodule -i mojapolisa.pp
```

instalira polisu mojapolisa.pp, snima je u \$TARGETED/modules/active/modules.

Komanda:

```
# semodule -r mojapolisa
```

izbacuje modul iz upotrebe.

Polisa se sastoji od sledećih fajlova:

Type Enforcement (.te) Sadrži allow pravila i interfejse kojima pristupa ovaj domen

File Context (.fc) Sadrži označavanja fajlova za ovaj modul

Interface fajl (.if) Sadrži interfejse koje drugi domeni koriste za pristup ovom domenu

5.11 Vežbe

1. Proverite da li je SELinux omogućen i u kom režimu radi.
2. Proverite kontekst fajla /etc/shadow.
3. Proverite da li se kontekst fajla čuva 'cp' i 'mv' komandama.
4. Proverite da li se kontekst fajla čuva kada se fajl kopira/pomera na zaseban fajl sistem.
5. Proverite da li se kontekst fajla čuva kada se fajl kopira/pomera dok je SELinux u permissive režimu rada.
6. Proverite da li fajl /etc/passwd ima dobar kontekst.
7. Trajno dozvolite Apache veb serveru da prikazuje sadržaje iz /srv/www direktorijuma.
8. Izlistajte sve promenljive definisane u polisi

9. Pronađite u man stranici httpd_selinux promenljivu koja omogućava prikazivanje korisničkih prezentacija Koja je vrednost te promenljive? Demonstrirajte šta se dešava kada promenite vrednost te promenljive Napomena: U novijim targeted polisama direktorijski korisnika imaju tip httpd_user_content_t koji nije zaštićen ovom promenljivom. Promenljiva httpd_enable_homedirs se odnosi na tip user_home_dir_t.

Property of Admin Training Center

Modul 6: Nenadgledana instalacija Linuxa

Nenadgledana instalacija je izraz koji opisuje automatizovanu instalaciju Linuxa, gde su podaci koji se standardno unose interaktivno tokom procesa instalacije već unapred dostupni programu za instalaciju. Administrator na taj način može da unapred pripremi kako će njegov novi server izgledati, kako će se organizovati prostor za smeštaj podataka, koje mrežne parametre treba postaviti i koje pakete treba instalirati.

Nenadgledana instalacija korišćenjem Kickstart sistema omogućava i izvršavanje kustomizovanih skriptova pre i posle same instalacije, što omogućava da se automatizuju i neke aktivnosti koje se obavljaju pošto je server instaliran.

Automatizovana instalacija preko Kickstart sistema je dostupna i na RedHat/CentOS distribucijama (gde je i nastala) i na novijim Ubuntu distribucijama (uz neka ograničenja).

Konfiguracija programa za instalaciju se obavlja preko Kickstart fajla `ks.cfg` (može imati i neko drugo ime). Ovaj fajl mora biti dostupan kada se sistem pokreće sa svog instalacionog medijuma. Lokacija tog fajla se navodi kao parametar u komandnoj liniji Boot Manager programa. Sam fajl može biti smešten na instalacionom mediju, na zasebnom disku ili disketi ili negde na mreži gde je dostupan preko HTTP ili FTP protokola ili se nalazi na nekom NFS deljenom direktorijumu.

6.1 Pozivanje nenadgledane instalacije

Kickstart sistem se poziva navođenjem putanje do kickstart konfiguracionog fajla u komandnoj liniji Boot Manager-a. Koristi se opcija 'ks':

```
linux ks=location/kickstart-file.cfg [kssendmac]
```

Opcija 'kssendmac' ima smisla ukoliko se kickstart fajl nalazi na nekom HTTP serveru. Ova opcija će u sklopu HTTP zahteva poslati i MAC adresu servera koji se instalira tako da, ukoliko je sa druge strane neka aplikacija, ona može, na osnovu te MAC adrese da pošalje nazad kickstart konfiguracioni fajl skrojen posebno za dati server.

Napomena:

Firstboot je set skriptova koji se izvršava kada se server prvi put pokrene. Njihova namena je da kompletiraju konfiguraciju sistema. Ukoliko se kroz nenadgledanu instalaciju ne instalira desktop grafičko okruženje, firstboot skriptovi neće biti izvršavani. Iz tog razloga je potrebno da se kroz opcije kickstart konfiguracionog fajla dodaju podaci koji bi se unosili prilikom pokretanja firstboot skriptova.

Mogući formati 'ks' opcije u zavisnosti od lokacije kickstart fajla su navedeni u sledećoj tabeli:

Lokacija	Format opcije
DVD uređaj	ks=cdrom:/directory/ks.cfg
Hard disk	ks=hd:/device/directory/ks.cfg
Drugi uređaji	ks=file:/device/directory/ks.cfg
HTTP server	ks=http://server.mydomain.com/directory/ks.cfg
HTTPS Server	ks=https://server.mydomain.com/directory/ks.cfg
FTP Server	ks=ftp://server.mydomain.com/directory/ks.cfg
NFS Server	ks=nfs:server.mydomain.com:/directory/ks.cfg

Tabela 6.1: Formati 'ks' opcije u zavisnosti od lokacije kickstart fajla

6.2 Kickstart konfiguracioni fajl

Kickstart konfiguracioni fajl je tekstualni fajl koji sadrži opcije automatizovane instalacije u formatu 'jedna opcija u jednom redu'. Dozvoljeno je imati i prazne redove, a linija može biti komentar ako počinje znakom '#'.

Najjednostavniji način kreiranja kickstart konfiguracionog fajla je preuzimanje nekog već gotovog. Naime, svaka instalacija RedHat/CentOS servera automatski, u početnom direktorijumu korisnika 'root' kreira `ks.cfg` koji sadrži sve opcije te instalacije. Na taj način može se jednostavno klonirati server korišćenjem već gotovog kickstart konfiguracionog fajla.

Primer kickstart konfiguracionog fajla prikazan je u nastavku:

```
#Generated by Kickstart Configurator
#platform=x86

#System language
lang en_US
#System keyboard
keyboard us
#System timezone
timezone Europe/Belgrade
#Root password
rootpw --iscrypted $1$3ndkIUa$f5ec561Bc5MWH68hpZTY0/
#Reboot after installation
reboot
#Use text mode install
text
#Install OS instead of upgrade
install
#Use CDROM installation media
cdrom
#System bootloader configuration
bootloader --location=mbr
#Clear the Master Boot Record
zerombr yes
#Partition clearing information
clearpart --all --initlabel
#Disk partitioning information
part /boot --fstype ext4 --size 500 --asprimary
```

```
part / --fstype ext4 --size 20480
part swap --recommended
part /home --fstype ext4 --size 1 --grow
#System authorization infomation
auth --useshadow --enablemd5
#Network information
network --bootproto=dhcp --device=eth0
#Firewall configuration
firewall --disabled
#Run the Setup Agent on first boot
firstboot --reconfig
```

6.3 Komande Kickstart-a

Komande Kickstart-a su direktive kojima se definiše instalacija sistema. Komande mogu imati opcije, od kojih neke mogu imati svoje parametre. Neke komande su obavezne i ne smeju biti izostavljene. Ostale komande su opcione.

Napomena:

Ako je komanda praćena znakom '*' onda je ona obavezna.

Ako je opcija neke komande praćena znakom '=' onda ona zahteva neku vrednost.

6.3.1 Komande koje definišu osnovne parametre sistema

keyboard*

Obavezna komanda koja postavlja odgovarajući raspored tastera na tastaturi. Raspored se navodi kao argument komande.

lang*

Obavezna komanda koja definiše podrazumevani jezik na sistemu koji se instalira. Jezik se definiše preko koda koji se prosleđuje kao argument komande. Kodove možete pogledati u fajlu /usr/share/system-config-language/locale-list.

timezone*

Ova komanda definiše vremensku zonu u kojoj se server nalazi. Zona se prosleđuje kao argument komande i identična je zonama koje se dobijaju komandom 'timeconfig'. Opcija:

--utc Definiše da je interni časovnik servera podešen na UTC zonu

6.3.2 Komande koje definišu autentifikaciju

auth ili authconfig*

Postavlja opcije za autentifikaciju novoinstaliranog servera. Opcije su slične opcijama komande 'authconfig'. Podrazumevano stanje je da se lozinke kriptuju ali da se ne koristi 'shadow' sistem.

--enablemd5 Uključuje kriptovanje preko MD5 heš algoritma

--enablenis Uključuje podršku za NIS autentifikaciju

--nisdomain= Postavlja domen za NIS

--nisserver= Postavlja server za autentifikaciju

--useshadow --enableshadow Koristi shadow sistem za smeštanje lozinki

--enableldap Uključuje podršku za LDAP u `/etc/nsswitch.conf`. Potrebno je i uključiti paket **nss_ldap** u grupu paketa koja se instalira. Takođe je potrebno podesiti parametre LDAP konekcije sa opcijama `--ldapserver=` i `--ldapbasedn=`.

--enableldapauth Uključuje autentifikaciju preko LDAP-a.

--ldapserver= Postavlja ime LDAP servera.

--ldapbasedn= Postavlja bazni DN.

--enableldaptls Uključuje TLS kriptovanu konekciju ka LDAP serveru

--enablekrb5 Uključuje Kerberos 5 autentifikaciju. Kerberos samo obavlja autentifikaciju tako da mora da se koristi ili sa LDAP-om ili sa NIS-om

--krb5realm= Postavlja Kerberos 5 *realm*

--krb5kdc= Postavlja servere koji obavljaju ulogu KDC-a u zadatom realmu.

--krb5adminserver Definiše koji od KDC servera izvršavai kadmin.

--enable smbauth Uključuje autentifikaciju korisnika prema SMB serveru (Samba ili Windows server). Slično kao kod Kerberosa, mora se koristiti u paru sa LDAP ili NIS autentifikacijom.

--smbservers= Postavlja servere za SMB autentifikaciju. Ako je više od jednog servera navedeno, njihova imena se razdvajaju zarezom

--smbworkgroup= Definiše radnu grupu za SMB autentifikaciju

--enablecache Uključuje 'nscd' servis koji kešira informacije o korisnicima

rootpw*

Postavlja lozinku za root korisnika. Lozinka se prosleđuje kao argument komande. Opcija

--iscrypted označava da je prosleđena lozinka već kriptovana

user

Ova komanda kreira korisnika na sistemu. Opcije su joj:

--name= Korisničko ime korisnika.

--groups= Spisak grupa, razdvojene zarezima, u koje je učlanjen korisnik

--homedir= Početni direktorijum korisnika, podrazumevano `/home/username`

--password= Početna lozinka korisnika. Ako se ova opcija izostavi nalog će biti zaključan.

--iscrypted= Da li je prosleđena lozinka u kriptovanom obliku ili čist tekst

--shell= Komandni interpreter korisnika

--uid= UID korisnika. Ako se izostavi uzima se prvi slobodan.

6.3.3 Komande za hardver

device

Ova komanda se koristi kada sistem sam ne može da prepozna uređaj i učita odgovarajući modul. Na ovaj način se definiše koji modul da se učitava i sa kakvim parametrima:

```
device tip modul  
--opts = "opcija1 opcija2 ..."
```

tip Oznaka za tip uređaja (**scsi** ili **eth**)

modul Naziv kernel modula

--opts= Opcije koje se prosleđuju kernel modulu. Više opcija treba biti navedeno sa razmakom i sve zajedno pod znacima navoda (kao u primeru)

driverdisk

Komanda koja govori gde je locirana disketa sa drajverima koji će trebati tokom instalacije a nisu u samoj distribuciji. Sadržaj diskete mora biti instaliran ili na početnom direktorijumu particije na hard disku ili negde na mreži gde mu se može pristupiti preko HTTP, FTP ili NFS protokola:

```
driverdisk particija  
--type = tip_fajlsistema
```

particija Particija na kojoj se nalaze drajveri

--type= Tip fajlsistema particije

--source= Ukoliko se imidž diskete (napravljen 'dd' komandom) nalazi na mreži, onda se umesto gornjeg primera koristi ova opcija. Argument opcije je URL sa putanjom do imidža diska.

mediacheck

Ovom komandom se zadaje testiranje instalacionog medijuma. Podrazumevana vrednost je da je ova opcija isključena.

6.3.4 Komande koje definišu tip instalacije

install

Definiše kakvu vrstu akcije (instalacija ili ažuriranje sistema) program za instalaciju treba da sproveđe. Ova komanda kao parametar prima medijum sa kojeg se instalacija izvršava (cdrom, harddrive, nfs, url). Detaljnije definicije medijuma se obavljaju kroz opcije komande:

cdrom Instalacija će se obaviti sa prvog CD/DVD ROM uređaja na sistemu

harddrive Instalacija će se obaviti sa lokalne particije, koja mora biti tipa 'vfat' ili 'ext2':

--biospart= Oznaka BIOS particije

--partition= Particija sa koje se instalira sistem.

--dir= Direktorijum koji sadrži instalaciju

nfs Instalacija sa NFS particije:

--server= Server sa kojeg se instalira

--dir= Direktorijum koji sadrži instalaciju

-[-opts=] Opcije koje treba postaviti prilikom kačenja NFS deljenog diektorijuma.

url Instalacija će e obaviti sa udaljenog FTP ili HTTP servera:

--url= URL na kojem se nalazi instalacija

upgrade

Označava da je u pitanju ažuriranje sistema a ne nova instalacija.

6.3.5 Komande koje definišu način instalacije

cmdline

Izvodi instalaciju u potpuno neinteraktivnom modu komandne linije. Bilo kakav prompt za interakciju će prekinuti instalaciju.

graphical

Uključuje instalaciju u grafičkom modu, što je i podrazumevana vrednost.

interactive

Ova komanda uključuje interaktivni mod rada programa za instalaciju. Parametri iz kickstart konfiguracionog fajla se tretiraju kao podrazumevane vrednosti ali korisnik može da ih promeni interaktivno.

text

Označava da će se instalacija izvršiti u tekst modu.

6.3.6 Komande koje definišu particionisanje diska

autopart

Automatsko kreiranje particija - min. 1GB za root (/) particiju, swap particija i boot particija, u zavisnosti od hardverske arhitekture. Uz pomoć 'part' direktive mogu se specificirati veličine ovih particija.

ignoredisk

Komanda kojom se nalaže programu za instalaciju da ignoriše pojedine diskove. Ima smisla ako se koristi sa komandom 'autopart' kako bi sprečila da se particije kreiraju na pojedinim diskovima.

--drives=a Imena diskova koje treba ignorisati (sda, sdb,...)

clearpart

Uklanja postojeće particije sa sistema, pre nego što se kreiraju nove. Podrazumevano je da se postojeće particije sačuvaju.

--all Uklanja sve particije na disku

--drives= Uklanja particije sa navedenih diskova

--initlabel Inicijalizuje disk za odgovarajuću arhitekturu.

--linux Uklanja samo Linux particije sa diska

--none Podrazumevano stanje: ne uklanja nijednu particiju

part ili partition*

Ova komanda je obavezna samo kod instalacije. Kod ažuriranja sistema se ignoriše. Format ove komande je:

```
part mountpoint opcije...
```

Parametar 'mountpoint' može biti:

putanja Apsolutna putanja do direktorijuma na koji se particija kači

swap U pitanju je swap particija

raid.id U pitanju je particija za softverski RAID (vidi komandu 'raid')

pv.id U pitanju je particija za LVM.

Ostale opcije su:

--size= Minimalna veličina particije u megabajtima.

--grow Veličina particije treba da poraste dok ne zauzme sav slobodan prostor ili dok ne dostigne zadatu maksimalnu veličinu

--maxsize= Maksimalna veličina do koje particija može da raste, u megabajtima

--noformat Sprečava formatiranje particije

--onpart=, --usepart= Definiše da će se koristiti postojeća particija. Upotrebite '--noformat' da biste sačuvali podatke na ovakvoj particiji.

--ondisk=, --ondrive= Forsira kreiranje particije na zadatom disku

--asprimary Particija mora biti primarna

--fstype= Forsira tip fajlsistema na particiji

--start= Definiše početni cilindar particije

--end= Definiše poslednji cilindar particije

--bytes-per-inode= Definiše broj i-nodova po particiji, tamo gde taj podatak ima smisla

--recommended Definiše veličinu particije automatski

raid

Kreira softverski RAID. Format ove komande je:

```
raid mountpoint --level=raid-nivo --device=ime  
particije [opcije...]
```

Parametar 'mountpoint' je direktorijum na koji se kači particija. Ako će data particija sadržati kernel (root ili /boot particija), onda ona mora biti nivoa 1.

Parametar 'particije' je spisak particija koje treba da sačinjavaju RAID niz.

Ostale opcije su:

--level= RAID nivo (0, 1 ili 5)

--device= Ime RAID uređaja, može biti md0...;md7

--bytes-per-inode= Definiše broj i-nodova po particiji, tamo gde taj podatak ima smisla

--spares= 'hot spare' particije

--fstype= Tip fajlsistema koji će biti kreiran na RAID particiji

--fsoptions= String sa opcijama koji se direktno kopira u /etc/fstab, treba da bude uokviren znakovima navoda.

--noformat Sprečava formatiranje već postojeće RAID particije

--useexisting Koristi postojeći RAID uređaj i formatiraj ga

volgroup

Kreira volumensku grupu na LVM-u. Komanda ima sledeći format:

```
volgroup ime particije opcije...
```

Opcije su:

--noformat Koristi postojeću VG i nemoj je formatirati

--useexisting Koristi postojeću VG i formatiraj je

--pesize= Definiše veličinu fizičkog ekstenta (PE)

logvol

Definiše logički volumen LVM-a. Ukoliko želite da kreirate LVM i koristite LV za particije, prvo particionišite disk komandama 'part' a zatim kreirajte VG komandom 'volgroup', pa tek onda zadajte komandu 'logvol'. Format komande je:

```
logvol mountpoint --vgname=VG --name=ime  
ostale_opcije
```

gde su ostale opcije:

--noformat Upotrebi postojeću LV sa postojeće VG i nemoj je formatirati

--useexisting Upotrebi postojeću LV sa postojeće VG ali je formatiraj prethodno

--fstype= Postavlja tip fajlsistema za LV

--fsoptions= Postavlja opcije fajlsistema

--grow Definiše da LV zauzme sav slobodan prostor na VG.

--maxsize= Definiše maksimalnu veličinu u MB do koje LV može da raste.

--recommended Definiše da se veličina LV određuje automatski

--size= Definiše veličinu u MB

--percent= Definiše veličinu u procentima slobodnog prostora na VG.

zerombr

Ovom komandom se brišu sve neispravne particione tabele. Komanda ima oblik:

```
zerombr yes
```

bootloader*

Specificira kako bi bootloader trebao biti instaliran.

--append= Navodi ekstra parametre za kernel komandnu liniju. Ako treba navesti više parametara onda se oni razdvajaju blanko znakom:

```
bootloader --location=mbr --append="hda=ide-scsi ide=nodma"
```

--driveorder= Specificira redosled diskova. Diskovi se označavaju imenima i razdvojeni su zarezima, bez belina

--location= Specificira gde će biti bootloader smešten. Vrednosti parametra su: **mbr** za Master Boot Record (podrazumevana vrednost), **partition** za instaliranje na prvi sektor particije koja sadrži kernel i **none** da se boot loader ne instalira

Primer za komande koje definišu particije:

```
clearpart --drives=hda,hdc --initlabel
# Raid 1 IDE config
part raid.11    --size 1000    --asprimary    --ondrive=hda
part raid.12    --size 1000    --asprimary    --ondrive=hda
part raid.13    --size 2000    --asprimary    --ondrive=hda
part raid.14    --size 8000    --ondrive=hda
part raid.15    --size 1 --grow    --ondrive=hda
part raid.21    --size 1000    --asprimary    --ondrive=hdc
part raid.22    --size 1000    --asprimary    --ondrive=hdc
part raid.23    --size 2000    --asprimary    --ondrive=hdc
part raid.24    --size 8000    --ondrive=hdc
part raid.25    --size 1 --grow    --ondrive=hdc

# You can add --spares=x
raid /          --fstype ext3 --device md0 --level=RAID1 raid.11 raid.21
raid /safe      --fstype ext3 --device md1 --level=RAID1 raid.12 raid.22
raid swap       --fstype swap  --device md2 --level=RAID1 raid.13 raid.23
raid /usr       --fstype ext3 --device md3 --level=RAID1 raid.14 raid.24
raid pv.01      --fstype ext3 --device md4 --level=RAID1 raid.15 raid.25

# LVM configuration so that we can resize /var and /usr/local later
volgroup sysvg pv.01
logvol /var        --vgname=sysvg  --size=8000    --name=var
logvol /var/freespace --vgname=sysvg --size=8000    --name=freespace
logvol /usr/local   --vgname=sysvg  --size=1 --grow --name=usrlocal
```

6.3.7 Komande koje se tiču bezbednosti sistema

firewall

Omogućava automatsko konfigurisanje iptables pravila.

--enabled, --enable Uključuje firewall sa blokadom dolaznih paketa koji inicijalizuju konekciju.

--disabled, --disable Isključuje firewall

--trust= Definiše tačno jedan interfejs za koji firewall propušta sve dolazne i odlazne pakete. Ukoliko je potrebno navesti više interfejsa, treba koristiti više puta ovu opciju.

--ssh, --telnet, --smtp, --http --ftp Propušta dolazne pakete koji inicijalizuju konekcije za zadate protokole. Pretpostavlja se da navedeni protokoli slušaju na standardnim portovima.

--port= Propuštaju se paketi koji stižu na dati port sa zadatim protokolom. Port i protokol treba navesti u obliku 'port:protokol', a sam port može biti naveden kao broj ili kao ime servisa koji sluša na ovom portu (prema stavkama u `/etc/services`). Ukoliko želite da propustite više portova, navedite ih razdvojene zarezima.

selinux

Ova komanda definiše u kom modu će SELinux biti startovan. Komanda prima jednu od opcija:

--enforcing SELinux će biti u enforcing modu

--permissive SELinux će biti u permissive modu

--disabled SELinux će biti isključen

6.3.8 Komande za mrežni pristup

network

Ovom komandom se definišu mrežni parametri kako kod sistema koji se instalira tako i za potrebe same instalacije (ako se koristi bilo kakav mrežni pristup). Ukoliko se ova komanda izostavi, a sama instalacija ne zahteva mrežni pristup (npr. instalira se sa DVD-a na kojem se nalazi i kickstart fajl) onda se neće konfigurisati ni mreža na instaliranom sistemu. Ako je ova komanda izostavljena a instalacija zahteva mrežni pristup, onda se podrazumeva da je taj pristup preko interfejsa 'eth0' i da će se mrežni parametri dobiti preko DHCP servisa. Na isti način će biti konfigurisani i interfejsi na instaliranom sistemu.

--bootproto= Definiše način konfigurisanja interfejsa ('dhcp' ili 'static'). Ukoliko se koristi 'static' način, ostali mrežni parametri se navode kao dodatne opcije (moraju sve biti u istoj liniji):

--ip= IP adresa

--netmask= Mrežna maska

--gateway= Podrazumevana ruta

--nameserver= DNS server (može se navesti samo jedan, ostali se mogu dodati kroz postinstalacioni skript)

--onboot= Da li interfejs treba podići prilikom startovanja sistema

--device= Definiše interfejs koji se konfiguriše. Ova opcija ne važi za mrežnu konfiguraciju u toku samog procesa instalacije ako je kickstart fajl na mreži.

6.3.9 Komande koje definišu praćenje instalacije

autostep

Slična komandi 'interactive' samo što automatski prebacuje na sledeći ekran. Koristi se najčešće za debagovanje konfiguracionog fajla.

--autoscreenshot Generiše skrinšotove svih ekrana programa za instalaciju. Koristi se pri dokumentovanju instalacije. Slike ekrana se smeštaju u direktorijumi `/root/anaconda-screenshots/`

logging

Ovom komandom možemo definisati da će se log informacije same instalacije zapisivati na udaljeni server. Udaljeni server mora da izvršava syslogd ili rsyslogd proces koji je konfigurisan da prima poruke sa udaljenih mašina. Ova komanda ne utiče na log informacije sistema koji se instalira. Opcije su:

--host= Adresa udaljenog hosta

--port= Port na kojem sluša syslogd/rsyslog proces na udaljenom serveru.

--level= Označava koji nivo log poruka će se prikazivati na virtuelnoj konzoli 3. Argument ove opcije treba biti jedno od: debug, info, warning, error ili critical. Što se tiče udaljenog servera, njemu se šalju poruke svih nivoa, bez obzira na ovu opciju.

vnc

Ova opcija definiše parametre VNC servera uz pomoć kojeg može da se prati grafička instalacija sistema sa udaljenog računara, koristeći VNC klijent. Ovaj način je preporučljiviji od instalacije u tekst modu zbog nekih ograničenja koja ima taj metod. Bez dodatnih opcija, komanda će startovati VNC server na standardnom portu, bez lozinke i ispisaće na ekran komandu kojom treba pozvati VNC klijentsku aplikaciju da bi se ona mogla zakačiti na server.

Dodatne opcije su:

--host= Umesto da se pokreće VNC server na lokalnoj mašini (koja se instalirat), ova opcija instruira VNC da se zakači na udaljenu mašinu gde bi trebao da je već pokrenut VNC klijent koji čeka konekciju.

--port= Definiše port na kojem sluša VNC klijentska aplikacija na udaljenoj mašini kada se sistem kači na nju umesto da pokreće lokalni VNC server proces.

--password= Specificira lozinku koju klijent treba da unese da bi mogao da se poveže na lokalni VNC server.

6.3.10 Komande koje definišu pakete koji se instaliraju

repo

Definiše dodatne repozitorijume koji se mogu koristiti prilikom instalacije sistema. Moće je definisati više dodatnih repozitorijuma, svaki zasebnom komandom:

```
repo --name=repoid [--baseurl=url |  
--mirrorlist=url]
```

--name= ID repozitorijuma. Ova opcija je obavezna.

--baseurl= URL repozitorijuma. Ne mogu se koristiti varijable, kao kod definicija repozitorijuma u `/etc/yum.repos.d/`. Ova opcija ne može se koristiti sa opcijom '`--mirrorlist=`'.

--mirrorlist= URL liste miror sajtova za repozitorijum. Lista ne sme sadržati varijable koje se koriste kod `/etc/yum.repos.d/` fajlova. Ova opcija se ne može koristiti zajedno sa opcijom '`--baseurl=`'.

6.3.11 Opcije koje definišu konfiguraciju X-a

skipx

Ova opcija definiše da se preskače konfiguracija X-a.

xconfig

Ovom opcijom se konfiguriše X Windows System, ako je isti selektovan za instalaciju. Ako je X selektovan za instalaciju a ova komanda nije ubaćena u kickstart fajl onda korisnik mora manuelno da konfiguriše X Window System tokom instalacije. Ova komanda treba biti izostavljena ako se ne instalira X Window System.

--driver= Specificira X drajver koji treba koristiti za grafički hardver.

--videoram= Specificira veličinu video RAM-a grafičke kartice.

--defaultdesktop= Specificira da li se koristi GNOME ili KDE dekstop okruženje. Prepostavlja se da je navedeno okruženje u spisku paketa za instalaciju.

--startxonboot= Da li će se pokretati grafičko okruženje automatski prilikom startovanja sistema.

--resolution= Specificira podrazumevanu rezoluciju grafičkog podsistema.

--depth= Specificira broj bitova po pikselu (dubina boja). Validne vrednosti argumenta su: 8, 16, 24 ili 32.

6.3.12 Komande za konfiguraciju sistema nakon instalacije

firstboot

Određuje da li se pokreće Setup Agent prilikom prvog startovanja servera (videti napomenu na strani 63!). Podrazumevano stanje je da se Setup Agent ne pokreće.

--enabled, --enable Setup Agent će se startovati

--disabled, --disable Setup Agent se neće startovati

--reconfig Setup Agent će se startovati u modu za rekonfiguraciju gde je moguće promeniti različite druge parametere pored onih standardnih.

services

Definiše koji servisi trebaju biti startovani a koji ugašeni prilikom startovanja sistema. Servisi su zadati u obliku liste imena razdvojenih zarezima.

--disabled= Spisak servisa koji ne trebaju biti startovani prilikom startovanja sistema.

--enabled= Spisak servisa koji trebaju biti startovani prilikom startovanja sistema.

6.3.13 Komande koje definišu kako se završava instalacija

halt

Gasi server nakon uspešno završene instalacije. Podrazumevana akcija je 'reboot'.

poweroff

Ukoliko je instalacija uspešno okončana sistem će biti ugašen kao da je zadata komanda 'shutdown -p'.

reboot

Ukoliko je instalacija uspešno okončana, sistem će biti restartovan. Ovo je podrazumevan način završavanja instalacije ako se nijedna komanda iz ove grupe ne navede.

shutdown

Ukoliko je instalacija uspešno okončana, ovom komandom će sistem biti ugašen.

6.3.14 Ostale komande

%include

Ova komanda omogućava da se deo kickstart konfiguracije učita iz fajla koji je prosleđen kao argument.

6.3.15 Dodatne sekcije kickstart fajla

%packages

Sekcija '%packages' služi da specificira koji paketi treba da se instaliraju. Paketi se mogu navoditi pojedinačno, kao i preko paketskih grupa. Imena dostupnih paketskih grupa su navenene u fajlovima *varijanta/repoadata/comps-*.xml*, koji se nalazi na instalacionom DVD-u. Unutar grupe, svaki paket ima dodeljenu jednu od opcija: 'mandatory', 'default' ili 'optional'. Ako je grupa navedena za instalaciju instaliraće se svi paketi sa opcijom 'mandatory'. Paketi koji imaju opciju 'default' će takođe biti instalirani ako negde drugde nisu eksplisitno označeni da ne treba da se instaliraju. Paketi sa opcijom 'optional' će biti instalirani samo ako su negde eksplisitno navedeni za instalaciju. Ako eksplisitno ne želimo da instaliramo neki paket ili paketsku grupu, pred njegovu specifikaciju ćemo staviti znak '-'. Takođe, moguće je koristi fajl globing znak '*' koji zamenjuje 0 ili više znakova.

Svaki paket ili paketska grupa treba biti upisan u zaseban red iza reda kojim se označava početak sekcije (%packages). Paketske grupe se navode preko svojih punih imena ili ID-ova, u obliku:

@ Puno ime grupe

Paketska grupa '@ Base' se uvek instalira, bez obzira da li je navedena u spisku paketa ili ne, ukoliko nije eksplisitno navedena opcija za neinstaliranje '@ Base' grupe.

Opcije se navode iza ključne reči '%packages':

--nobase Označava da se '@ Base' grupa ne instalira automatski.

--ignoremissing Ignoriše nedostajuće pakete i ne prekida instalaciju ako program za instalaciju identificuje da nekog od paketa koji treba instalirati nema na naznačenom instalacionom medijumu.

Program za instalaciju će, pre početka instalacije paketa, napraviti stablo paketa kako bi razrešio sve međuzavisnosti. Ukoliko se detektuje da neki od paketa koji je potreban nije naveden kroz grupu ili eksplisitno, biće automatski dodat na spisak za instalaciju.

%pre

Kickstart sistem omogućava izvršavanje skripta neposredno pošto se kickstart fajl parsira, a pre nego što počne izvršavanje komandi. Sekcija '%pre' mora biti navedena iza svih komandi i mora biti terminisana direktivom '%end'.

Komandni interpreter kojem se prosleđuju linije između '%pre' i '%end' je podrazumevano /bin/sh, ali se to može promeniti preko opcije:

--interpreter koja u nastavku navodi putanju do željenog komandnog interpretera.

Napomena:

Ukoliko je mrežni pristup konfigurisan tokom instalacije, isti se može koristiti u '%pre' skriptu.

Ipak, ne mogu se koristiti FQDN već samo IP adrese.

Sledeći primer ilustruje upotrebu '%pre' sekcije, gde skript, na osnovu toga da li je na sistemu dostupno jedan ili dva diska, definiše kako diskove treba particionisati tako

što odgovarajuće kickstart komande upisuje u fajl /tmp/part-include. U tom slučaju, sve komande u kickstart skriptu koje definišu particioniranje novog sistema trebaju biti zamenjene sa:

```
%include /tmp/part-include

%pre
#!/bin/sh
hds=""
mymedia=""
for file in /proc/ide/h* do
    mymedia=`cat $file/media`
    if [ $mymedia == "disk" ] ; then
        hds="$hds `basename $file`"
    fi
done

set $hds
numhd=`echo $#`
drive1=`echo $hds | cut -d' ' -f1`
drive2=`echo $hds | cut -d' ' -f2`
#Write out partition scheme based on whether there are 1 or 2 hard drives
if [ $numhd == "2" ] ; then
    #2 drives
    echo "#partitioning scheme generated in %pre for 2 drives" \
        > /tmp/part-include
    echo "clearpart --all" >> /tmp/part-include
    echo "part /boot --fstype ext3 --size 75 --ondisk hda" \
        >> /tmp/part-include
    echo "part / --fstype ext3 --size 1 --grow --ondisk hda" \
        >> /tmp/part-include
    echo "part swap --recommended --ondisk $drive1" >> /tmp/part-include
    echo "part /home --fstype ext3 --size 1 --grow --ondisk hdb" \
        >> /tmp/part-include
else
    #1 drive
    echo "#partitioning scheme generated in %pre for 1 drive" \
        > /tmp/part-include
    echo "clearpart --all" >> /tmp/part-include
    echo "part /boot --fstype ext3 --size 75" >> /tmp/part-include
    echo "part swap --recommended" >> /tmp/part-include
    echo "part / --fstype ext3 --size 2048" >> /tmp/part-include
    echo "part /home --fstype ext3 --size 2048 --grow" >> /tmp/part-include
fi
%end

%post
```

Sekcija '%post' služi da definiše skript koji će se izvršiti po instalaciji sistema. Ova sekcija, kao i sekcija '%pre' mora biti navedena na kraju kickstart fajla (redosled '%pre' i '%post' sekcija nije bitan). Sekcija mora biti terminisana direktivom '%end', kao i '%pre' sekcija.

Skriptovi se izvršavaju u 'chroot' okruženju, tj. kao da se izvršavaju na novoinstaliranom serveru.

Opcije '%post' su:

--interpreter Navodi putanju do željenog komandnog interpretera.

--nochroot Dozvoljava da se navedene komande ne izvršavaju pod 'chroot'-om. Ovo dozvoljava da se, npr. mogu kopirati fajlovi sa instalacionog medijuma na novoinstalirani

server.

--log Navodi (kao argument) gde se smešta log ispisa skripta.

Napomena:

Ukoliko je mreža na novoinstaliranom sistemu konfigurisana statički, onda se u '%post' skriptu mogu koristiti i IP adrese i FQDN imena. No, ukoliko je mreža konfigurisana dinamički, preko DHCP-a, onda rezolvovanje imena još uvek nije dostupno i mogu se koristiti samo IP adrese.

Primer skripta koji kači NFS šerovani direktorijum i sa njega izvršava komandni skript:

```
%post --log=/root/ks-post.log
mkdir /mnt/temp
mount -o noblock 10.10.10.2:/usr/new-machines /mnt/temp
openvt -s -w -- /mnt/temp/runme
umount /mnt/temp
%end
```

6.4 Vežbe

1. Napraviti kickstart konfiguracioni fajl koji će instalirati CentOS 6 distribuciju na sistem sa dva diska od 20GB kojih treba napraviti:

- jednu `/boot` particiju veličine 500M koja mora biti primarna
- jednu `/` particiju veličine 5GB na logičkom volumenu `/dev/vg0/lv-root`
- jednu `/srv` particiju koja će zauzeti ostatak slobodnog prostora na VG `vg0`
- jednu swap particiju veličine 4GB

Postaviti mrežne parametre za interfejs `eth0` koje ćete dobiti od predavača. Postaviti firewall pravila tako da se propušta SSH i HTTP saobraćaj. Postaviti vremensku zonu 'Europe/Belgrade' i UTC vreme internog časovnika. Jezik sistema treba da bude 'en_US'. Instalirati osnovni sistem i `httpd` server.

Property of Admin Training Center

Indeks

A

ACL.....	25
Određivanje pristupa fajlu	27
Omogućavanje ACL.....	25
podrazumevane ACL.....	26
Postavljanje prava preko ACL	26
Pregledanje ACL.....	26
pristupe ACL	26
Uklanjanje prava iz ACL.....	27
Uvod	25
Vežbe.....	29

G

GRUB.....	18
-----------	----

I

init ramdisk.....	17
Initramfs.....	17
dracut.....	17
iptables.....	36
akcije.....	45
ACCEPT, 45	
DROP, 45	
REJECT, 45	
aktivnosti	
DNAT, 48	
LOG, 46	
MARK, 46	
MASQUERADE, 49	
MIRROR, 48	
QUEUE, 48	

K

Kernel moduli.....	19
--------------------	----

insmod	22
lsmod	20
modinfo	21
modprobe	22
postavljanje parametara	23
rmmod	22
uklanjanje	22
Kickstart	61
komande	63
konfiguracioni fajl	62
pozivanje	61
sekcije	73

L

Linux firewall	31
connection tracking	34
conntrack modul	34
filter paketa	34
implementacija	49
iptables	36
princip rada	36
proxy firewall	34
stateful	34
stateless	34
Tipovi firewalla	34
Vežbe	49
Log fajlovi	7
Rsyslog	7
Logrotate	14

N

Nenadgledana instalacija Linuxa	61
Netfilter	35
akcije	36
chains	35
lanci	35
princip rada	36
tabele	35

O

OSI model	31
slojevi	31

R

Rotiranje logova	14
Rsyslog	7
šabloni	13
akcije	12
filteri	9
globalne direktive	7
konfigurisanje	7
moduli	8
pravila	8

S

SELinux	51
alatke	55
koncepti	52
korisnik	53
log fajlovi	56
moduli	58
nivo	53
označavanje fajlova	56
polise	54
prilagođavanje polisa	57
sigurnosni kontekst	52
tip	53
tranzicija domena	53
uloga	53

T

TCP/IP	32
IP protokol	32
TCP prekid konekcije	33
TCP protokol	32
TCP uspostavljanje konekcije	33
UDP protokol	33



Admin Training
Center

L2-3 Administracija infrastrukturnih servisa

Veselin Mijušković, Marko Uskoković, Ljubiša Radivojević

Copyright © 2014 Veselin Mijušković, Marko Uskoković, Ljubiša Radivojević

OBJAVIO ADMIN TRAINING CENTER

www.atc.rs

Licencirano po Creative Commons Attribution-NonCommercial 3.0 Unported License (the "License"). You may not use this file except in compliance with the License. You may obtain a copy of the License at <http://creativecommons.org/licenses/by-nc/3.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Prvo izdanie, 2014

Sadržaj

Uvod	5
Tipografske konvencije	5
Modul 1: NTP - Network Time Protocol	7
1.1 Stratumi	7
1.2 Kako NTP radi?	8
1.3 NTP konfiguracioni fajl	9
1.4 Pokretanje NTP servisa	11
Modul 2: Domain Name System	13
2.1 Organizacija DNS	13
2.1.1 DNS zone	13
2.2 Komponente DNS sistema	14
2.3 Zapis resursa - Resource Records	15
2.4 DNS upiti	15
2.4.1 Rekurzivni upiti	16
2.4.2 Iterativni (re-rekurzivni) upiti	17
2.5 Reverzno mapiranje	17
2.6 Transfer zona	18
2.6.1 Transfer cele zone - AXFR	18
2.6.2 Transfer izmenjenih delova zone - IXFR	18
2.6.3 Obaveštavanje o izmenama - NOTIFY	18
2.7 BIND server	18
2.7.1 BIND konfiguracije	19
2.7.2 Obavezne zone u named.conf fajlu	20
2.7.3 Ostali tipovi zona	20
2.7.4 Kompletan sadržaj named.conf fajla	21

Modul 3: DHCP - Dinamičko konfigurisanje hostova	23
3.1 DHCP server	23
3.1.1 Parametri	23
3.1.2 Deklaracije	24
3.1.3 Deklaracija subnet	24
3.2 Najam adresa	26
3.3 DHCP relaying	26
3.4 Podešavanje DHCP klijenta	26
Modul 4: LDAP	27
4.1 Organizacija LDAP-a	27
4.1.1 Atributi LDAP i klase objekata	27
4.2 LDIF fajl	28
4.3 LDAP šeme	29
4.3.1 OID	29
4.3.2 Primeri definicija klasa i atributa	30
4.4 OpenLDAP	30
4.5 Konfiguriranje OpenLDAP servera	32
4.5.1 Konfiguriranje OpenLDAP klijenata	32
4.5.2 Konfiguracija slapd servera	32
4.6 Konfiguriranje sistema da koristi OpenLDAP za autentifikaciju	33
Modul 5: NFS	35
5.0.1 RPC	35
5.1 NFS servisi	36
5.2 Konfiguriranje NFS servera	36
5.2.1 Komanda exportfs	39
5.3 Prikaz eksportovanih fajl sistema na klijentskoj mašini	39
5.4 Kačenje eksportovanog fajl sistema	39
Indeks	41

Uvod

Skripta je podeljena na poglavlja, a poglavlja na sekcije. Unapred skrećemo pažnju polaznicima da je ova skripta samo deo dokumentacije koju oni treba da koriste. Polaznicima se savetuje da pročitaju man i help strane za svaku komandu, kao i da potraže na Internetu dodatne informacije i načine kako da iste koristite.

Tipografske konvencije

Radi lakšeg snalaženja u tekstu, koristili smo neke tipografske konvencije na koje vam ovde skrećemo pažnju:

- ukoliko se uvodi neki značajan pojam, on će u prvom pomenu biti isписан **proporcionalnim bold tekstom**;
- boldovano su prikazane i neke značajne tvrdnje na koje treba obratiti pažnju u tekstu;
- *proporcionalnim italicom* su napisane reči na stranom jeziku, najčešće engleskom;
- nazivi fajlova u tekstu su ispisani *neproporcionalnim fontom*

Oblik neke komande prikazan je na sledeći način:

```
# komanda [opcije] argument...
```

Deo koji je napisan uspravnim fontom se unosi kako je napisan. Opcioni deo, koji se može izostaviti je naveden uvek u uglastim zagradama (same zgrade se ne unose). Ukoliko je tekst isписан *italicom* to znači da je u pitanju neki opšti naziv i umesto njega treba uneti neku stvarnu vrednost. Ukoliko iza teksta stoje tri tačke '...' to znači da se taj deo može ponavljati.

Deo teksta koji se odnosi na direktni unos korisnika i ispis računara, kao i sadržaji fajlova i sl. će biti prikazani u zasebnom bloku:

```
$ ls -F
myscript.sh*  Vezbe/
```

Boldovanim tekstom je prikazano ono što polaznik treba da unese onako kako je napisano u skripti. Regularnim tekstom je prikazan ispis programa koji ne treba unositi.

Gde god smo mislili da nešto posebno treba naglasiti, to smo naveli na kao napomenu ili upozorenje na sledeći način:

Napomena:

Hard linkovi ne mogu biti kreirani za direktorijume, već samo za regularne fajlove!

Upozorenje!

Uvek koristite 'modprobe' za učitavanje modula jer ćete tako učitati i zavisne module!

Na kraju, ukoliko se zadaje deo koda (npr. skripte komandnog interpretera bash), onda će to izgledati ovako:

```
1 #!/bin/bash
2 #
3 # helloworld.sh - standardni Hello, World skript
4 #
5
6 echo "Hello, World!"
```

Modul 1: NTP - Network Time Protocol

Jedan od važnijih zahteva unutar neke lokalne mreže jeste uskladivanje internih časovnika na čvorovima u mreži (serveri, desktop računari, laptopovi, mobilni računari, svičevi, ruteri itd.). Razlog za to je dvojak:

1. sa usklađenim časovnicima moguće je korelirati aktivnosti koje su se istovremeno događale na različitim serverima
2. neki servisi i aplikacije zavise od usklađenosti časovnika u mreži i tačnog vremena.

Prvi razlog može izgledati nepotreban ali je ipak vrlo značajan. Naime, aktivnosti oko troubleshooting-a i odgovaranja na incidente skoro uvek zahtevaju da se utvrdi redosled događaja, a isti se može utvrditi samo kroz analiziranje log fajlova. Svi događaji u log fajlovima su zabeleženi preko vremenskih oznaka (engl. *timestampos*), tako da je vrlo bitno tačno znati kada se neki događaj desio u odnosu na neki drugi. Ovaj zahtev čak i prevazilazi granice vaše mreže - svi događaji koji se odvijaju na Internetu trebali bi imati ispravnu vremensku oznaku.

1.1 Stratumi

Da bi se implementirao sistem koji omogućava da čvorovi u mreži imaju tačno vreme podešeno u odnosu na neki referentni časovnik, koristi se Network Time Protocol (NTP). Umesto da se svi čvorovi sinhronizuju sa referentnim časovnikom, ovaj protokol definiše hijerarhiju čvorova, u zavisnosti od toga koliko su "sinhronizaciono udaljeni" od referentnog časovnika. Svi čvorovi su svrstani u 16 "stratuma" (nivoa), u zavisnosti koliko su "udaljeni" od izvora tačnog vremena (referentni časovnik).

Stratum 0 Ovaj stratum čine precizni izvori tačnog vremena, u koje spadaju atomski časovnici, GPS sistem, mobilne telefonske mreže i sl.

Stratum 1 Ovaj stratum čine računari koji su direktno nakačeni na neki od stratum 0 uređaja. Oni predstavljaju glavne čvorove u mreži NTP.

Stratum 2-14 Ove stratume sačinjavaju uređaji koji se sinhronizuju sa uređajima koji imaju za jedan niži stratum od njih samih. Takvi uređaji i sami mogu biti konfigurisani da budu referentni izvori tačnog vremena za uređaje čiji je stratum za jedan veći od njihovog.

Stratum 15 Ovo je najniži stratum u kojem se nalaze uređaji koji se sinhronizuju sa uređajima sa stratumom 14. Oni ne mogu biti referentni izvori tačnog vremena ni za jedan drugi uređaj.

Stratum 16 Ovo je oznaka za uređaje koji nisu sinhronizovani.

1.2 Kako NTP radi?

NTP prikazuje vreme kao broj sekundi od 00:00:00, 1. januara 1900. godine. Pošto se za prikaz vremena koriste 32-bitni brojevi to će "premotavanje" časovnika biti u 2036. godini (dve godine pre Unixovog "premotavanja"). No, NTP se više bavi razlikama među časovnicima pa ovo premotavanje neće imati toliki uticaj. NTP je sposoban da omogući kontrolu vremena ispod jedne sekunde. U LAN mrežama, pod idealnim okolnostima preciznost je oko 1ms. Na Internetu, preciznost je standardno u okvirima 10s.

NTP uračunava vreme za koje paket putuje kroz mrežu, kao i vreme koje serveru treba da odgovori. To radi tako što NTP klijent prilikom slanja zabeleži vremensku oznaku kao 'originating'. NTP server, odmah prilikom prijema paketa zabeleži unutar njega još jednu vremensku oznaku, kao 'receive', a zatim, nakon obrade a neposredno pre slanja paketa doda još jednu oznaku kao 'transmit'. NTP klijent prilikom prijema paketa zabeležava vremensku oznaku kada je paket pristigao (oznaka 'incoming') pa tako može da sračuna povratno vreme paketa kao:

$$RTT = t_{\text{incoming}} - t_{\text{transmit}} + t_{\text{receive}} - t_{\text{originating}}$$

Parametar RTT se dalje koristi za proračun 'pomaka' (engl. *drift*), tj. vrednosti koja predstavlja razliku između frekvencije lokalnog i referentnog časovnika i koji se čuva u fajlu `/var/lib/ntp/drift`.

Razlika označava koliko treba ažurirati klijentov časovnik da bi bio u sinhronizaciji sa časovnikom NTP servera. No, ovo podešavanje se ne obavlja odmah već se svake sekunde lokalni časovnik pomera za 0.5ms. Na ovaj način časovnik koji kasni 1 sekundu će se sinhronizovati za 2000 sekundi. Razlog ovome je što neke aplikacije ili delovi sistema mogu da se počnu ponašati čudno ukoliko se desi da je lokalni časovnik išao ispred referentnog časovnika NTP servera (npr. fajl može imati vreme kreiranja u budućnosti jer je neposredno po kreiranju fajla lokalni časovnik vraćen unazad za razliku).

Razlika veća od 1000s se neće sinhronizovati, već je potrebno da se to izvrši manuelno, npr. komandom `ntpdate`.

Sva vremena u NTP protokolu su UTC vremena. Primena vremenskih zona i zimsko-g/letnjeg računanja vremena se primenjuje samo na klijentima. UTC, kao zvanično vreme, je bazirano na IAT (*International Atomic Time*), odnosno SI definiciji sekunde i dana kao jedinice vremena jednake 81400 SI sekundi. Međutim, srednji solarni dan koji je definisan GMT (*Greenwich Mean Time*) i njegovim naslednikom UT1 (*Universal Time, 1*) se razlikuje neznatno od UTC pa se na nivou godine pojavljuje razlika. Dopushteno je da razlika između UTC i UT1 vremena bude ispod 1 sekunde. Razlika preko 1 sekunde se anulira uvođenjem 'prestupne sekunde' (23:59:60). Zbog varijacija u brzini rotacije Zemlje, nije moguće unapred odrediti pojavu 'prestupne sekunde' pa nju proglašava telo zvano IERS (*International Earth Rotation and Reference Systems Service*).

Kako bi izbegao mogućnost da maliciozni korisnik naruši rad sistema tako što će slati lažne NTP informacije, NTP protokol omogućava da se klijenti mogu autentifikovati servere. Postoje različiti načini implementacije autentifikacije, uključujući simetrične i asimetrične algoritme.

1.3 NTP konfiguracioni fajl

Već smo rekli da je `/etc/ntp.conf` konfiguracioni fajl NTP servera. U pitanju je standardni tekstualni fajl koji sadrži prazne linije, komentare (počinju znakom '#') i završavaju se do kraja linije) i direktivama. Direktive mogu biti sledeće:

driftfile

Direktiva `driftfile` definiše putanju do drift fajla, kao u primeru:

```
driftfile /var/lib/ntp/drift
```

restrict

Direktivom `restrict` se definišu prava pristupa serveru. Unutar konfiguracionog fajla može se naći više 'restrict' direktiva, posebno zato što se ista direktiva, ali sa različitim opcijama koristi za ograničavanje pristupa kod IPv4 i kod IPv6 protokola.

Format ove komande je:

```
restrict destinacija [opcije...]
restrict -6 destinacija [opcije...]
```

(druga linija predstavlja verziju direktive za IPv6 protokol)

Parametar `destinacija` predstavlja entitet na koji se komanda odnosi i može biti oblika:

default označava podrazumevano pravilo

a.b.c.d označava specifičnu IP adresu *a.b.c.d*

a.b.c.d mask m.n.o.p označava podmrežu čija je mrežna adresa *a.b.c.d* i mrežna maska *m.n.o.p*

Opcije definišu koja ograničenja se primenjuju na zadati entitet i mogu biti:

ignore svi paketi poslati od strane klijenta će biti ignorisani

kod označava da će server poslati klijentu 'kiss-of-death' paket kojim ga upozorava da ograniči količinu upita

limited server neće odgovarati na upite čija učestanost prevaziđa limite postavljene komandom *discard*

lowpriotrap trapovi koje postavljaju zadati klijenti će biti tretirani kao događaji niskog nivoa

nomodify onemogućava izmene konfiguracije od strane klijenta

noquery onemogućava upite koje šalju komande *ntpq* i *ntpd*

nopeer onemogućava stvaranje 'peer' odnosa

noserve onemogućava sve pakete izuzev upita komandi *ntpq* i *ntpd* (opcija sa suprotnim značenjem od opcije 'noquery')

notrap onemogućava trapove koje zadaje komanda *ntpd*

notrust odbija pakete koji nisu kriptografski autentifikovani

ntpport prihvata samo pakete koji dolaze sa porta 123/tcp

version odbija pakete koji ne pripadaju tekućoj verziji NTP servisa

discard

Ovom direktivom se definišu limiti propusnog opsega paketa upućenih serveru. Format ove direktive je:

```
discard opcija argument
```

Parametar 'opcija' je:

average specificira minimalni prosečni razmak između dva paketa. Argument je \log_2 vrednost vremena u sekundama. Podrazumevana vrednost argumenta je 3 (tj. $2^3 = 8$ sekundi).

minimum specificira minimalni apsolutni razmak između dva paketa, takođe u vremenu izraženom u \log_2 vremenu. Podrazumevana vrednost je 1 ($2^1 = 2$ sekunde)

server

Direktiva **server** definiše referentne časovnike za lokalni server, tj. NTP servere koje će lokalna mašina propitivati za tačno vreme. Obično se navodi više od jedne linije, kako bi se implementirala visoka dostupnost i mogućnost da jedan od servera vraća pogrešno vreme. Naime, lokalni NTP servis će propitivati sve servere koji su navedeni u 'server' direktivama kako bi dobio što precizniju predstavu o tekucem vremenu vršeci usrednjavanje dobijenih razlika. No, ukoliko neki server prikazuje znatno veću razliku od ostalih, njegov odgovor će biti odbačen.

Format ove direktive je:

```
server adresa [opcija...]
```

Parametar 'adresa' označava ime ili IP adresu NTP referentnog servera.

broadcast / manycastclient

Direktiva **broadcast** definiše broadcast ili multicast IP adresu na koju server šalje pakete. Direktiva **multicastclient** postavlja adresu kojoj se šalju upiti u adresiranom multicast modu. Klijenti odgovarajućom direktivom takođe mogu postaviti istu adresu kao adresu za prijem. Format direktive je:

```
broadcast adresa [opcija...]  
multicastclient adresa
```

peer

Direktivom **peer** se dva servera istog stratuma povezuju međusobno. Ideja je da se serveri istog nivoa uvežu u mrežu, sa tim da to ima smisla samo ako serveri imaju makar jednu različitu 'server' direktivu. Format ove direktive je:

```
peer adresa [opcija...]
```

Parametar 'adresa' je ime ili IP adresa NTP servera istog stratuma sa kojim se uspostavlja 'peer' veza. 'Peer' serverima najčešće upravlja isti administrator.

Opcije direktiva server, peer, broadcast i manycastclient

Parametar 'opcija' kod komandi **server**, **peer** i **broadcast** može biti:

burst - označava da se umesto jednog svaki put šalje 8 paketa (ovu opciju treba izbegavati što je više moguće!!!) [server]

iburst - ubrzava inicijalnu sinhronizaciju. Ako server ne odgovori, paketi se šalju sa razmaka od 16 sekundi. U protivnom, šalju se na svaka 2 sekunda. [server]

minpoll vrednost definiše minimalni interval za propitivanje referentnog ili peer servera. Vrednost je izražena kao \log_2 , tj. vreme je izraženo kao $2^{vrednost}$ [server, peer]

maxpoll vrednost definiše maksimalni interval za propitivanje referentnog ili peer servera. Vrednost se iskazuje kao kod opcije 'minpoll' [server, peer]

prefer definiše preferentni server [server, peer]

ttl vrednost definiše vrednost TTL paketa (podrazumevana vrednost je 127) [server, peer]

version verzija definiše podrazumevanu verziju NTP protokola (podrazumevana vrednost je najviša verzija protokola koju tekuća implementacija podržava, obično 4) [server, peer]

key broj definiše koji ključ se koristi za kriptografsku autentifikaciju paketa [server, peer, broadcast]

broadcastclient, manycastserver, multicastclient

Direktive **broadcastclient**, **manyজcastserver** i **multicastclient** definišu da će lokalni server primati upite koji idu preko broadcast ili multicast saobraćaja (standardno ili adresirano). Format ovih direktiva je:

```
broadcastclient
manyজcastserver adresa
multicastclient adresa
```

Primer jednog konfiguracionog fajla je dat u sledećem listingu:

```
# Prava pristupa: podrazumevano je onemogućen pristup svemu
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery

# Dozvoljavamo sve za loopback port
restrict 127.0.0.1
restrict -6 ::1

# Mašine iz lokalne mreže mogu da koriste ovaj server kao referentni
restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap nopeer

# Koristimo servere iz poola pool.ntp.org
server 0.pool.ntp.org
server 1.pool.ntp.org
server 2.pool.ntp.org
```

1.4 Pokretanje NTP servisa

Opcije za pokretanje NTP servisa su definisane u `/etc/sysconfig/ntpd`. Ovaj fajl obično sadrži sledeće opcije:

```
# Drop root to id 'ntp:ntp' by default.
OPTIONS="-u ntp:ntp -p /var/run/ntpd.pid -g"
```

Opcija '-g' označava da će, prilikom startovanja NTP servera, on ignorisati limit od 1000 sekundi razlike na početku sinhronizacije, i automatski će postaviti dobijeno vreme. Time ne mora pre pokretanja NTP servera ručno da se sinhronizuje reme.

Sam servis se pokreće komandom:

```
# service ntpd start
```

Da biste proverili da li NTP server radi, zadajte sledeću komandu:

```
$ ntpq -p
      remote          refid      st t when poll reach   delay    offset  jitter
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+clock.util.phx2 .CDMA.        1 u  111  128  377  175.495    3.076   2.250
*clock02.util.ph .CDMA.       1 u   69  128  377  175.357   7.641   3.671
 ms21.snowflakeh .STEP.      16 u     - 1024   0   0.000   0.000   0.000
 rs11.lvs.iif.hu .STEP.      16 u     - 1024   0   0.000   0.000   0.000
 2001:470:28:bde .STEP.      16 u     - 1024   0   0.000   0.000   0.000
```

Komanda 'ntpq' prikazuje prozvane NTP servere i informacije o njihovim odgorima. Značenja kolona su sledeća:

remote - referentni server

refid - referentni server referentnog servera

st - stratum servera

t - tip servera (lokalni, unicast, multicast ili broadcast)

when - vreme u sekundama proteklo od poslednjeg propitivanja servera

poll - interval u sekundama između dva propitivanja servera

reach - oktalna bit-maska koja pokazuje uspešnost poslednjih 8 propitivanja servera (krajnji levi bit je poslednje propitivanje)

delay - RTT u milisekundama

offset - razlika između lokalnog i serverovog časovnika, u milisekundama

jitter - razlika sukcesivnih vremena dobijenih od servera (visok jitter označava da su serveri ili mreža nestabilni)

Kraći izvešta može se dobiti komandom **ntpstat**:

```
$ ntpstat
unsynchronised
  time server re-starting
  polling server every 64 s

$ ntpstat
synchronised to NTP server (10.5.26.10) at stratum 2
  time correct to within 52 ms
  polling server every 1024 s
```

Modul 2: Domain Name System

DNS – *Domain Name System, engl. sistem domenskih imena* je softverski sistem mapiranja imena u IP adresu, radi lakšeg pamćenja adresa.

2.1 Organizacija DNS

DNS je distribuirana baza podataka organizovana hijerarhijski tako što postoji mali broj top-level domain name servera koji su odgovorni za domene najvišeg nivoa i one koji su ispod njih. Potpuno kvalifikovano ime hosta se sastoji od dva dela: imena mašine i imena domena, a domen može uključivati i pod-domene. Domeni najvišeg nivoa mogu biti:

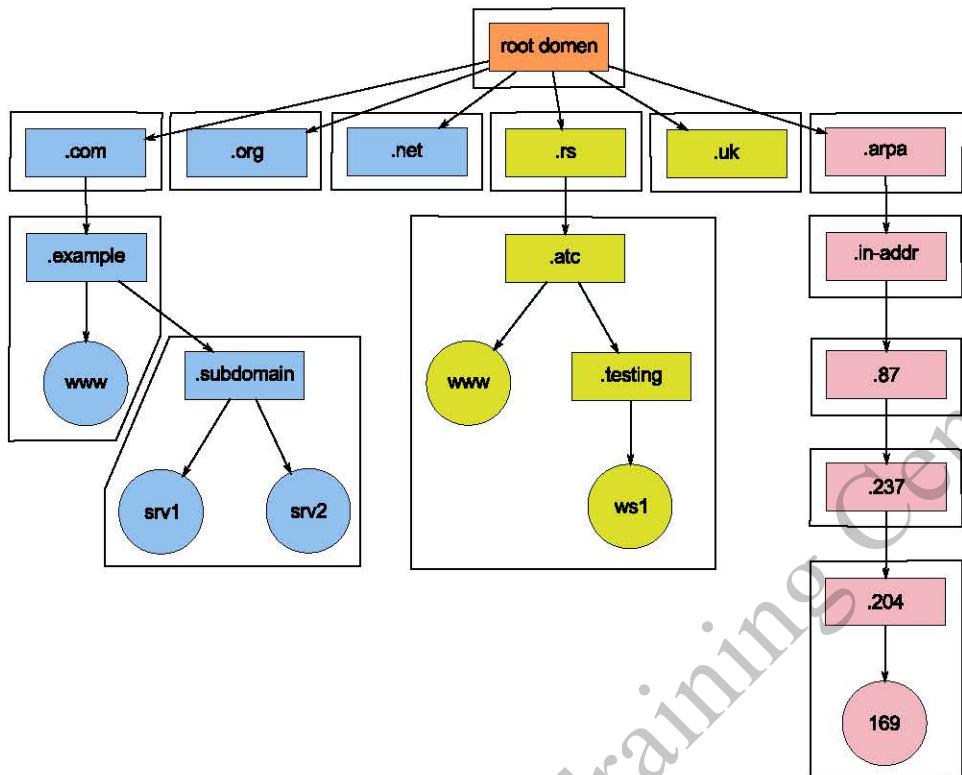
- generički top-level domeni (.com, .net, .org) koji su komercijalni, tj. čiji se poddomeni mogu slobodno registrovati za odgovarajuću cenu.
- generički top-level domeni vezani za ustanove USA (.mil, .gov, .edu). Ovi generički domeni su namenjeni različitim ustanovama unutar USA i entiteti van USA ne mogu registrovati svoj domen ispod ovih gTLD-ova.
- top-level domeni vezani za pojedine države, tzv. *country-code top-level domains - ccTLD*. U pitanju su dvoslovni domeni vezani za zemlje čija zvanična dvoslovna oznaka je navedeni domen. Same države definišu pravila i način registrovanja domena ispod određenog ccTLD-a.
- .arpa domen. Ovaj domen je specifičan jer se koristi za reverzno mapiranje IP adresa u domenska imena.

2.1.1 DNS zona

DNS zona je jedinstveni, kontinualni deo prostora domenskih imena unutar DNS za koji je administrativna odgovornost delegirana jednom menadžeru. U zavisnosti od toga da li neki domen ima svoje podomene ili ne, on može biti ceo u jednoj zoni ili administrator domena može odlučiti da svoje podomene izdvoji u zasebne zone, tako da se onda domen prostire unutar više zona. Dakle, jedna zona počinje domenom i proteže se do listova stabla domena ili do top-level čvorova poddomena.

Internet Corporation for Assigned Names and Numbers (ICANN) je neprofitna organizacija odgovorna za održavanje i koordinaciju Domain Name Sistema. Ova organizacija između ostalog nadgleda dodelu IP adresa i domena i odgovorna je za jedinstvenost i validnost svake adrese na Internetu.

U srcu DNS sistema nalazi se 13 računara zvanih *root serveri* koje održava ICANN i koji su rašireni svuda po svetu. Svi oni sadrže iste važne informacije: IP adrese svih TLD (top level



Slika 2.1: Stablo DNS sa zonama

domain) registara, kako globalnih domena (.com, .net, .org ...) tako i registrare domena zemalja (.rs, .uk, .de ...). Na Internetu se nalaze hiljade kompjutera koji se nazivaju Domain Name Resolvers (tj. name serveri), čiji je zadatak da kopiraju informacije koje se nalaze u root serverima. Name serveri se najčešće nalaze u Internet provajderima (ISP-ovima) ili različitim institucijama. Oni su ti kojima korisnici šalju zahteve za rezolvovanje imena domena kako bi dobili IP adresu mašine. Takav zahtev se najpre šalje lokalnom name serveru, koji iz svoje baze (ukoliko održava pod-domene) ili iz top-level registra (čije IP adrese je kopirao sa root servera) pribavlja odgovarajuću IP adresu i vraća je nazad korisniku koji ju je tražio. Ovaj postupak se naziva rezolvovanje IP adrese (IP address resolving). Obzirom da je DNS hijerarhijski sistem, jedan domen može imati više pod-domena. Za rezolvovanje IP adresa koje pripadaju nekom pod-domenu, odgovorni su lokalni name serveri, tako da se rezolvovanje nastavlja rekursivno, jedan po jedan pod-domen sve dok se ne pronađe adresa koja je tražena.

Ovako organizovan DNS sistem efikasno razrešava veliki broj problema na koje se naišlo tokom razvoja Interneta:

- Potreba za hijerarhijom imena
- Potreba za podelom opterećenja Name servera na veći broj mašina
- Potreba za delegacijom administracije Name servera na veći broj autoriteta

2.2 Komponente DNS sistema

RFC 1034 definiše DNS sistem u tri sloja:

- Podaci koji opisuju domene

- Program koji izvršava usluge Name servera
- Program ili biblioteka koja vrši rezolvovanje

Podaci o domenu koje DNS serveri održavaju su definisani u obliku zapisa zvanih "Resource Records" (RR) i čuvaju se u zonskim fajlovima. Format zonskog fajla je definisan u RFC 1035 i pridržava ga se većina Name servera.

Program koji pruža usluge Name servera na osnovu sadržaja fajlova Zona daje odgovore na upite koji stižu od hostova iz lokalne ili javne mreže, pribavlja podatke od drugih Name servera, vrši keširanje informacija i sinhronizovanje fajlova Zona drugih Name servera, root i lokalnih.

Treći deo DNS sistema je biblioteka ili program koji vrši rezolvovanje IP adresa pričaći TCP ili UDP protokolom sa nekim Name serverom. Standardni je deo svih mrežno orijentisanih operativnih sistema.

2.3 Zapisi resursa - Resource Records

Fajl zone se sastoji od sledećih zapisa resursa u IN klasi (Internet)

A 32-bitna IPv4 adresa pridružena imenu domena (RFC 1035)

AAAA 128-bitna Ipv6 adresa pridružena imenu domena (RFC 3596)

CNAME Kanoničko ime za DNS alias pojedinačnog hosta. Ne sme pokazivati na domen koji ima bilo koji drugi tip zapisa, niti sam biti CNAME zapis (RFC 1035)

HINFO Opcionie informacije o tipu procesora, operativnom sistemu hosta i sl. (RFC 1035)

MX Mail Exchanger – host koji upravlja elektronском poštom zone. Zapis se sastoji od imena hosta (A zapis) i prioriteta tog MX hosta (RFC 1035)

NS Authoritativni Name Server domena (RFC 1035)

PTR IP adresa hosta. Koristi se u reverznim DNS upitima (RFC 1035)

SOA Start Of Authority – zapis definiše ime zone, e-mail kontakt, vremenske periode osvezavanja i validnosti informacija (RFC 1035)

SRV specifikacija servisa dostupnih u zoni (RFC 2872)

TXT slobodan opis domena (RFC 1035)

2.4 DNS upiti

Glavni zadatak koji DNS server izvršava je odgovaranje na upite koje mu šalju lokalni ili udaljeni rezolveri ili drugi DNS serveri. Jedan DNS server ne prima samo upite o zonama koje on održava (za koje je autoritativan), već o bilo kom domenu, čak je i velika većina upita upravo o domenima i zonama koje on ne održava.

DNS server se može podestiti da, nezavisno za svaku zonu, bude:

autoritativan Autoritativen server je server koji upravlja datom zonom i može biti:

master Server koji sadrži originalne zonske zapise

slave Server koji od master servera za datu zonu preuzima zonski fajl

neautoritativan Neautoritativan server nema direktne informacije o zoni već ih dobija na neki drugi način (upitom, iz keša):

caching Server koji odgovore upita čuva u svojoj memoriji

forwarding Server koji prosleđuje upite drugom serveru i čuva odgovor

rekurzivni Server koji u ime svojih klijenata obavlja rekurzivne upite ka autoritarnim serverima.

Postoji tri vrste upita koje DNS serveri mogu da opslužuju:

- Rekurzivni upit gde se vraća kompletan odgovor, dakle IP adresa na osnovu imena hosta. DNS serveri ne moraju da podržavaju rekurzivne upite.
- Iterativni upit (tj. ne-rekurzivni) gde se vraća delimičan odgovor, tj. IP adresa hosta koji je bliži traženom hostu i može da pruži dalje informacije.
- Inverzni upit gde se na osnovu IP adrese vraća ime domena tog hosta.

Svi DNS upiti se obavljaju UDP protokolom na portu 53.

2.4.1 Rekurzivni upiti

Prilikom rekurzivnih upita DNS server vraća korisniku kompletan odgovor na zadato pitanje (ili javlja grešku). DNS server ne mora obavezno da ima podršku za rekurzivne upite. DNS server na rekurzivni upit može odgovoriti na tri načina:

1. Poslaće odgovor na upit u obliku bilo kog CNAME zapisa koji odgovara IP adresi i pri tome indicirati da li je podatak autoritativan ili keširan.
2. Javiće grešku NXDOMAIN da domen ili host ne postoji.
3. Javiće da se dogodila privremena greška, npr. da je mreža nedostupna i sl.

U rekurzivnom upitu DNS server će umesto klijenta (rezolvera) pretraživati jedan po jedan nivo u hijerarhiji sve dok ne pronađe IP adresu traženog hosta. Tok rekurzivnog rezolvovanja tipa: "Koja je IP adresa mašine www.google.com" bi mogao da teče na sledeći način:

1. Rezolver na klijentu šalje upit lokalno konfigurisanom DNS serveru
2. DNS server potraži da li je www.google.com u kešu, i ne pronađe ga
3. DNS server šalje upit root serveru
4. Root server odgovara informacijom o TLD serveru za .com
5. DNS server šalje .com TLD serveru upit "koja je IP adresa hosta www.google.com?"
6. TLD server odgovara informacijom o Name serveru koji je odgovornom za domen google.com
7. DNS server šalje Name serveru odgovornom za google.com upit "koja je IP adresa hosta item www.google.com?"

2.4.2 Iterativni (re-rekurzivni) upiti

Iterativnim upitom klijent može dobiti ili konačan ili samo delimičan odgovor koji mu može pomoći u daljem rezolvovanju IP adrese. DNS serveri moraju da imaju podršku za iterativne upite.

Na iterativni upit DNS server može odgovoriti na 4 načina:

- Vratiće kompletan odgovor zajedno sa CNAME zapisom, tj. aliasom traženog hosta i pri tome nagovestiti da li je podatak autoritativan ili keširan.
- Javiće grešku NXDOMAIN da host ili domen ne postoje.
- Javiće privremenu grešku, npr. nemogućnost pristupanja DNS-u zbog konekcije i sl.
- Vratiće referentni host ili IP adresu koji su bliži traženom hostu, npr. autoritativni Name server traženog domena.

Tok iterativnog rezolvovanja tipa: "Koja je IP adresa mašine www.google.com" bi mogao da teče na sledeći način:

1. Rezolver na klijentu šalje upit lokalno konfigurisanim DNS serveru
2. DNS server traži www.google.com u lokalnom kešu i ne nade ga.
3. DNS odgovori adresom root servera
4. Rezolver šalje upit root serveru
5. Root server odgovara adresom TLD servera za .com
6. Rezolver šalje upit .com TLD serveru
7. TLD server odgovara adresom Name servera odgovornog za google.com
8. Rezolver šalje upit Name serveru za google.com
9. Zonski fajl na google.com Name serveru sadrži zapis da je www alias za www1
10. DNS klijentu šalje IP adresu, CNAME i A zapise.

2.5 Reverzno mapiranje

Inverzni upiti se ne koriste da bi se pronašlo ime hosta neke IP adrese. Ovaj proces se naziva Reverzno mapiranje (Look-up) i koristi rekurzivne i iterativne upite unutar specijalnog domena IN-ADDR.ARPA.

Potpuno kvalifikovano ime hosta se piše s leva na desno, iako je hijerarhijska struktura obrnuta: na vrhu hijerarhije je domen najvišeg nivoa, koji se piše na kraju adrese. U IP adresama (npr. 192.168.1.2) najviši čvor u hijerarhiji (192) je naveden na početku, a oznaka hosta je na kraju. Da bi reverzno mapiranje bilo moguće, potrebno je da se obrne redosled okteta i tako smesti u domen IN-ADDR.ARPA.

Na taj način se www.google.com (host čija IP je adresa 66.249.85.104) smešta kao PTR zapis za 104.85.249.66.in-addr.arpa i rezolvovanje imena hosta na osnovu IP adrese se vrši na isti način kao i rezolvovanje IP adrese na osnovu imena hosta.

Napomena:

Zone za reverzno mapiranje i zone domenskih imena u koji se IP adrese hostova reverzno preslikavaju ne moraju uopšte pripadati istom entitetu. Naime, vlasnik zone je vlasnik domena, dok je vlasnik reverznog domena (*X.Y.in-addr.arpa*) onaj entitet ko je vlasnik odgovarajućeg IP subneta (u našem primeru, subnet *Y.X.0.0/16*).

2.6 Transfer zona

Kada je DNS bio dizajniran promene u zonama su vršene transferom zone, komandom AXFR. Internet se od u zadnjih 15 godina mnogo promenio i proširio tako da je transfer cele zone postao neefikasan i zahtevan. Da bi se smanjili potrebni resursi izmišljen je Incremental Zone Transfer (IXFR) koji predstavlja osvežavanje samo onih delova zone koji su promenjeni.

2.6.1 Transfer cele zone - AXFR

RFC-ovima 1034 i 1035 predviđeno je da sekundarni DNS serveri (slave) povremeno proveravaju da li imaju najsvežije podatke koje primarni server pruža. Vreme između dva proveravanja je definisano vrednošću REFRESH u SOA zapisu.

Proveravanje da li sekundarni DNS server ima najsvežiju verziju zone vrši se poređenjem vrednosti SERIAL definisane u SOA zapisu. Svaki put kada se promena u zoni izvrši, vrednost SERIAL se poveća tako da sekundarni DNS serveri znaju da treba da izvrše transfer zone.

Transfer zone se uvek vrši TCP protokolom preko porta 53.

2.6.2 Transfer izmenjenih delova zone - IXFR

RFC-om 1995 uveden je delimični transfer zone komandom IXFR čime se osvežavaju samo oni zapisi u zoni koji su promenjeni. Sekundarni DNS server povremeno proverava vrednost promenljive SERIAL iz SOA zapisu i ukoliko ima promena zahteva transfer zone, po mogućству delimični. Ukoliko primarni ili sekundarni DNS server ne podržava delimični transfer zone, obavlja se transfer cele zone.

2.6.3 Obaveštavanje o izmenama - NOTIFY

Da ne bi dolazilo do dugog perioda propagiranja promena između primarnog (master) i sekundarnog (slave) DNS servera, čekanjem da prođe vreme definisano promenljivom REFRESH iz SOA zapisu uvedeno je obaveštavanje sekundarnih servera da se promena možda dogodila. Kada sekundarni server primi NOTIFY poruku on proverava vrednost promenljive SERIAL i ukoliko je do promene zaista došlo vrši potpuni ili delimični transfer zone.

2.7 BIND server

Bind (skraćenica od Berkeley Internet Name Domain) je najzastupljeniji DNS server otvorenog izvornog koda i predstavlja de-fakto standard na Unix-olikim sistemima. Nalazi se u sklopu svih Linux distribucija. Razvija ga Internet Systems Consortium, Inc. (ISC), organizacija koja upravlja F-ROOT serverom i glavni je registrar za .org gTLD.

Aktuelna verzija (oktobar 2014. godine) servera BIND je 9.10.1. Iako je BIND u prošlosti često imao sigurnosnih propusta, od verzije 9 koja je napisana od početka pruža veliku sigurnost zahvaljujući ekstenzijama DNSSEC. Takođe, podrška za IPv6, podrška za više procesora i dobra portabilnost su svakako argumenti koji potvrđuju uspešnu budućnost BIND servera.

2.7.1 BIND konfiguracije

Bind može biti podešen da obavlja sledeće tipove DNS usluga:

Master (primarni) server, koji je odgovoran za određene zone

Slave (sekundarni) server, koji kopira zone od svog master DNS servera

Caching only server koji obavlja rezolvovanje umesto klijenata

Forwarding (proxy) server koji prosleđuje zahteve drugom DNS serveru

Stealth (DMZ ili split-view) server, koji radi unutar privatnih mreža

Authoritative Only server, koji je master neke zone i koji ne kešira druge

Osnovna konfiguracija BIND servera se nalazi u fajlu `/etc/bind/named.conf`, mada podrazumevana lokacija može biti promenjena parametrom `-c` putanjom prilikom startovanja. U njemu se navodi lokacija PID fajla i lokacija fajlova Zona. Fajlovi zona se obično smeštaju u direktorijum `/var/bind/`.

Fajлом `named.conf` se podešava ponašanje i funkcionalnosti BIND servera. Sastoji se od sledećih klauzula:

acl Access Control Lists – definišu liste pristupa

controls Kontrolisanje udaljene administracije pomoću alatke `rndc`

include Uključivanje sadržaja nekog drugog fajla na ovom mestu

key Deljeni ključevi kojim se vrši autentifikacija

logging Konfiguriše lokaciju, nivo i tip logovanja koje BIND vrši.

lwres Opcije kojima BIND radi kao lightweight resolver

options Globalne opcije zajedničke svim zonama

server Način pristupanja nekom drugom serveru

trusted-keys Ključevi za autentifikaciju

view Funkcionalnost BIND-a koja zavisi od adrese klijenta

zone Definiše specifične zone koje će name server održavati.

Svaka od ovih klauzula ima dosta neobaveznih parametara, koji ovde neće biti objašnjavani, već će biti prikazani u primeru. Parametri se pišu iza imena klauzule i to unutar vitičastih zagrada, a komentari mogu biti označeni na tri načina:

```
/* kao u C-u */  
// kao u C++  
# kao u PERL-u
```

Svaka klauzula, kao i parametri unutar nje se zajednički nazivaju direktive. Svaka direktiva mora da se završi sa znakom tačka-zarez `'.'`. Klauzule i parametri koji mogu imati svoje parametre su složene direktive. Složene direktive se sadrže od ključne reči (komande), opcionog imena i tela. Telo se sastoji od drugih direktiva u kom slučaju telo počinje znakom `'{'` i završava znakom `'}'`.

Upozorenje!

Mada tela složenih direktiva liče na izraze C jezika, postoji bitna sintaksna razlika — iza zatvorene vitičaste zagrade koja označava kraj tela obavezno mora da stoji delimiter ';' , kao i kod prostih direktiva.

2.7.2 Obavezne zone u named.conf fajlu

U zavisnosti od tipa usluga koje je BIND podešen da pruža, u `named.conf` fajlu je potrebno navesti različite tipove zona, ali postoje i zone koje moraju biti podešene uvek.

root.servers zona

root.servers zona se pretražuje kada ime hosta koje se traži ne pripada ni jednoj drugoj lokalno definisanoj zoni i kada odgovor nije u kešu. Tip zone je `hint`, a fajl `root.servers` sadrži spisak root servera od kojih BIND može dobiti spisak TLD servera pojedinih domena najvišeg nivoa. Ovaj spisak se osvežava svaki put kada se BIND startuje. Primer kako se zona definiše u `named.conf` fajlu:

```
zone "." {
    type hint;
    file "root.servers";
};
```

localhost zona

localhost zona je zona kojom se definiše rezolvovanje loopback adrese 127.0.0.1. Definiše se u `named.conf` fajlu pomoću:

```
zone "localhost" {
    type master;
    file "master.localhost";
};
```

Zona 0.0.127.IN-ADDR.ARPA

Zona **0.0.127.IN-ADDR.ARPA** je zona koja služi za reverzno mapiranje loopback adrese.

```
zone "0.0.127.in-addr.arpa" {
    type master;
    file "named.local";
};
```

2.7.3 Ostali tipovi zona

Zone za reverzni DNS se definišu isto kao i zone za forward DNS. Domen koji koristi reverzna zona mora biti poddomen od '.in-addr.arpa', gde su poddomeni definisani kao A, B ili C klase IP adresa. Primer kako je definisana za domen 192.168.0.0/24:

```
zone "0.168.192.in-addr.arpa" {
    type master;
    file "rev-192.168.0";
};
```

Ukoliko je server tipa 'slave' za datu zonu, onda je potrebno da server u definiciji zone definiše koji su mu master serveri od kojih povlači zonske fajlove. Takođe, master server mora dozvoliti transfer zone za slave servere:

Tako, na masteru, u okviru 'options' klauzule ili unutar definicije zone mora stajati:

```
allow-transfer { ip-adresa-slave-servera; };
```

Na slave serveru zona mora sadržati 'masters' direktivu:

```
zone "example.com" in {
    type slave;
    file "slave.example.com";
    masters { 192.168.2.7; 10.2.3.15 port 1127; 2001:db8:0:1::15; };
};
```

U gornjem primeru navode se tri master servera, jedan sa IPv4 adresom, drugi sa IPv4 adresom i navedenim portom i treći sa IPv6 adresom.

2.7.4 Kompletan sadržaj named.conf fajla

```
// named.conf
acl "mynet" { 192.168.1.0/24; };

controls {
    inet 127.0.0.1 allow { localhost; };
    keys { "mykey"; };
};

key "mykey" {
    algorithm hmac-md5;
    secret "";
};

options {
    directory "/var/named";
    allow-recursion { "mynet"; };
    pid-file "/var/run/named.pid";
    statistics-file "/var/named/named.stats";
    allow-update { none; };
};

// zone
zone "localdomain" {
    type master;
    file "forward/localhost";
    allow-transfer { none; };
};

zone "test.com" {
    type master;
    file "forward/test.com";
    allow-transfer { 192.168.1.2; };
};

zone "1.168.192.in-addr.arpa" {
    type master;
    file "reverse/1.168.192";
    allow-transfer { 192.168.1.2; };
};

zone "slave.org" {
    type slave;
    masters { 192.168.1.2; };
    file "slave/slave.org";
};
```

```
zone "." {  
    type hint;  
    file "root.hints";  
};  
// kraj fajla named.conf
```

Sadržaj slave.org zonskog fajla:

```
// zone fajl za slave.org  
$TTL 2d      // 172800 sekundi je podrazumevan TTL za zone  
@ IN SOA ns1.slave.com. hostmaster.slave.com. (  
        2003080800 // se = serijski broj  
        12h      // ref = refresh  
        15m      // ret = ponovni pokusaj osvezavanja  
        3w      // ex = validnost podataka  
        3h      // min = minimum vremena su podaci validni  
        )  
  
        IN NS ns1.example.com.  
        IN MX 10 mail.example.net.  
joe   IN A 192.168.254.3  
www   IN CNAME joe
```

Modul 3: DHCP - Dinamičko konfigurisanje hostova

Dynamic Host Configuration Protocol (DHCP) pruža mogućnost automatizovanja mrežne konfiguracije računara i mrežnih uređaja koji koriste TCP/IP protokol.

U jednoj mreži ne sme biti više od jednog DHCP servera da ne bi dolazilo do konfikata prilikom dodeljivanja IP adresa.

3.1 DHCP server

Standardni DHCP server je `dhcpd` koji je proizvod ISC-a (iste organizacije koja je zadužena za održavanje BIND DNS servera). Server `dhcpd` implementira kompletan DHCP i BOOTP protokol,

Po instalaciji, potrebno je konfigurisati server, a to se radi editovanjem fajla `/etc/dhcp/dhcpd.conf`. Ovaj fajl sadrži komentare (počinju znakom '#'), prazne linije i direktive, Direktive mogu biti:

parametri - definišu kako se neka aktivnost izvršava, da li je treba izvršiti ili koje mrežne opcije treba poslati klijentu;

deklaracije - opisuju topologiju mreže, opisuju klijente, obezbeđuju adrese klijentima ili primenjuju grupu parametara na grupu deklaracija.

Parametri koji počinju ključnom reči **option** nazivaju se opcijama. Ove opcije kontrolišu *DHCP opcije*. Ostali parametri konfigurišu vrednosti koji nisu opcione ili kontrolišu ponašanje DHCP servera.

Parametri (uključujući i opcije) koji su deklarisani pre sekcije ogradiene vitičastim zagrada ({{}}) se smatraju globalnim parametrima. Globalni parametri se primenjuju na sve sekcije definisane u nastavku fajla.

3.1.1 Parametri

Navođenje parametara u `dhcpd.conf` fajlu je dato u sledećem primeru:

```
default-lease-time 600;
max-lease-time 7200;

option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
```

```
option domain-search "example.com";
```

U zavisnosti gde su, unutar konfiguracionog fajla, definisani parametri njihova oblast važenja se menja.

- Parametri koji su definisani van svih deklaracija su globalni parametri i važe od trenutka deklarisanja za sve deklaracije koje slede deklaraciju parametra.
- Parametri unutar neke deklaracije važe od trenutka deklarisanja do kraja same deklaracije. Pošto deklaracija može biti unutar druge deklaracije, to istovetni parametri koji su definisani kao globalni ili unutar deklaracije koja sadrži datu deklaraciju gube svoju vrednost i preuzimaju vrednost definisanu unutar date deklaracije.

Za detaljan spisak mogućih opcija i parametara pogledajte man stranu za *dhcp-options*.

3.1.2 Deklaracije

Deklaracije se sastoje od više direktiva koje su grupisane u 'telu' deklaracije, ograničenom otvorenom i zatvorenom vitičastom zagradom ({}). Direktive unutar deklaracije mogu biti parametri ili druge deklaracije.

3.1.3 Deklaracija subnet

Deklaracija **subnet** je najvažnija deklaracija unutar konfiguracionog fajla *dhcpd.conf*. Ona definiše sve podmreže na koje je zakačen DHCP server.

Upozorenje!

U konfiguracionom fajlu morate deklarisati sve podmreže na koje je zakačen DHCP server, uključujući i one za koje DHCP server neće alocirati adrese!

Format deklaracije je:

```
subnet mrežna-adresa netmask mrežna-maska {
    iskazi;
}
```

Ukoliko je predviđeno da DHCP server dinamički alocira adrese za datu podmrežu, onda deklaracija te podmreže mora sadržati **range** direktivu:

```
range početna-adresa krajnja-adresa;
```

Direktiva 'range' definiše kontinualan blok adresa koji počinje adresom 'početna-adresa' a završava adresom 'krajnja-adresa'. Podrazumeva se da su obe adrese iz opsega podmreže u kojoj su definisani.

Sa definisanim 'range' opsegom, DHCP server će dodeliti prvu slobodnu adresu hostu koji bude zahtevao da mu se adresa dodeli. Ovo znači da će hostovi iz jedne podmreže vremenom menjati svoje adrese. Ukoliko to nije željeno ponašanje, već želimo da neki hostovi imaju tačno definisane adrese iz podmreže celo vreme, onda možemo koristiti deklaraciju **host**:

```
host ime-hosta {
    hardware ethernet ethernet-MAC-adresa;
    fixed-address ip-adresa;
}
```

Direktiva 'hardware ethernet' definiše MAC adresu po kojoj možemo prepoznati koji host je u pitanju, a direktiva 'fixed-address' dodeljuje fiksnu adresu datom hostu. Ova adresa

bi trebala da bude iz istog opsega kao i podmreža u kojoj se nalazi host direktiva ali da istovremeno ne pripada 'range' bloku.

Ukoliko na jednom fizičkom interfejsu imamo više podmreža, onda sve njihove deklaracije moramo spakovati u **shared-network** deklaraciju:

```
shared-network ime {  
    subnet ...  
    subnet ...  
}
```

Ime deljene mreže je identifikator te mreže i ne koristi se na drugim mestima.

Takođe, više deklaracija koje imaju iste opcije, a koje se razlikuju od globalnih ili opcija definisanih unutar okružujuće deklaracije mogu biti grupisane **group** deklaracijom:

```
group {  
    subnet-ili-host...  
    subnet-ili-host...  
}
```

Primer jednog /etc/dhcp/dhcpd.conf fajla je prikazan u nastavku.

```
default-lease-time 600;  
max-lease-time 7200;  
  
option domain-name-servers 192.168.1.200 192.168.2.200;  
option domain-search "example.com";  
  
shared-network interna-mreza {  
    option routers 192.168.2.254;  
    option broadcast-address 192.168.3.255;  
  
    subnet 192.168.1.0 netmask 255.255.252.0 {  
        range 192.168.1.65 192.168.1.128;  
    }  
  
    subnet 192.168.2.0 netmask 255.255.252.0 {  
        range 192.168.2.65 192.168.2.128;  
  
        option domain-search "lab.example.com";  
        default-lease-time 300;  
        max-lease-time 3600;  
    }  
}  
  
group {  
    option domain-search "infra.example.com";  
  
    host mon1 {  
        hardware ethernet AA:BB:CC:DD:EE:FF;  
        fixed-address 192.168.1.10;  
    }  
  
    host mon2 {  
        hardware ethernet 0A:0B:0C:0D:0E:0F;  
        fixed-address 192.168.2.10;  
    }  
}
```

3.2 Najam adresa

DHCP, na zahtev klijenta dodeljuje IP adresu. Ta adresa se smatra 'iznajmljenom' (engl. *leased*) na određeno vreme. Podrazumevano vreme iznajmljivanja adrese je definisano direktivom 'default-lease-time' čiji je argument vreme u sekundama. No, klijent može da zatraži da dobije adresu na neko drugo, duže vreme. Maksimalno vreme na koje se adresa može iznajmiti je definisano direktivom 'max-lease-time' čiji je argument takođe vreme u sekundama.

Po isteku najma, klijent može da zatraži novu adresu ili, češće, da obnovi najam već dodeljene adrese.

Informacije o tome koje adrese su trenutno iznajmljene, u koje vreme i ostali parametri vezani za najam adresa se čuvaju u fajlu `/var/lib/dhcpd/dhcpd.leases`. Ovaj fajl ne bi trebalo menjati niti dirati.

3.3 DHCP relaying

Ukoliko postoji razgranata mreža sa dosta podmreža na kojima je potrebno dinamički dodeljivati adrese, onda je za svaku podmrežu potreban DHCP server. No, u slučaju velikog broja podmreža, nije isplativo imati zaseban DHCP server za svaku od njih. U tim slučajevima moguće je koristiti jednu od karakteristika DHCP protokola a to je **relaying**. Naime, moguće je umesto DHCP servera postaviti **dhcrelay** servis koji će zahteve sa nekog interfejsa slati odgovarajućem DHCP serveru. Na taj način jedan DHCP server može da opslužuje više podmreža čak i ako nije fizički nakačen na svaku od njih.

3.4 Podešavanje DHCP klijenta

Na Linux distribucijama nije potrebno zasebno podešavati DHCP klijenta koji je zadužen za komunikaciju sa DHCP serverom, već je to inkorporirano u sistem na koji podešavamo mrežne parametre servera.

Jedan od klijenata za DHCP je program **dhclient**. Njega najjednostavnije možete pozvati iz komandne linije sa:

```
# dhclient [opcije] interfejs
```

Naravno, ovu komandu treba koristiti isključivo u procesu otklanjanja mrežnih problema, jer se ona najčešće poziva iz samih sistemskih skriptova koji služe za konfigurisanje mrežnih parametara.

Za opcije komande 'dhclient' pogledajte njenu man stranu.

Modul 4: LDAP

Lightweight Directory Access Protocol je protokol koji definiše način pristupa, prikaza i uvoza/izvoza podataka koji sačinjavaju globalni katalog podataka. LDAP definiše četiri modela:

informacioni model - način na koji su podaci organizovani u globalnom katalogu

model imenovanja - način na koji se podaci referenciraju

funkcionalni model - način na koji se pristupa LDAP-u, čitaju, pretražuju, dodaju i modifikuju podaci

bezbednosni model - način kontrole pristupa pojedinim podacima.

4.1 Organizacija LDAP-a

LDAP, sam po sebi, predstavlja bazu podataka koja je optimizovana za mali broj upisa/izmena i veliki broj čitanja podataka. Takođe, organizacija podataka je hijerarhijska, koja se može najjednostavnije prikazati stablom objekata. Svaki objekat se naziva **stavka** (engl. *entry*). Svaka stavka ima tačno jednu stavku-roditelja i nula ili više stavki-dece. Sve stavke-deca koje imaju istog roditelja su 'rođaci'. Kompletna struktura se naziva **stablo informacije podataka** (engl. *Data Information Tree*). Vrh stabla (jedina stavka koja nema svog roditelja) se naziva **koren, baza** ili **sufiks**.

Svaka stavka je instanca jedne ili više 'klasa objekata' (engl. *objectClass*). Svaka klasa objekata sadrži nula ili više **atributa**. Atributi imaju imena (ponekad i aliase) i sadrže podatke. Karakteristike klase objekata i njihovih atributa su definisane po ASN.1 protokolu.

4.1.1 Atributi LDAP i klase objekata

Atributi imaju sledeće osobine:

- svi atributi su članovi jedne ili više klase objekata
- svaki atribut definiše **tip podataka** koji može sadržati
- atributi mogu biti opcioni ili obavezni. Jedan atribut može biti opcion za jednu klasu objekata a obavezan za drugu klasu objekata.
- atributi mogu imati pojedinačnu ili višestruku vrednost. Pojedinačna vrednost označava da samo jedna vrednost može biti prisutna unutar jedne stavke. Višestruka vrednost označava da unutar jedne stavke može biti više vrednosti vezanih za jedan atribut.

- atributi imaju imena i, ponekad, aliase koji su obično skraćena imena atributa (npr. atribut *commonName* je član klase *person* i ima svoj alias *cn*; oba imena mogu biti upotrebljena kada se referencira ovaj atribut).
- svaka stavka se jednoznačno može identifikovati prema nekom atributu ili skupu atributa. Takvi atributi se nazivaju **imenski atributi** (engl. *naming attribute*) ili **RDN** - *Relative Distinguished Name*.

Klase objekata takođe mogu biti hijerarhijski organizovane i imati svoje roditeljske klase od kojih nasleđuju attribute.

4.2 LDIF fajl

Popunjavanje DIT se može obaviti na više načina od kojih je najčešći korišćenjem tzv. LDIF (Lightweight Data Interchange Format) fajlova. Svako stablo se popunjava od korenog objekta nagore, tj. uvek mora biti prvo dodata roditeljska stavka pre stave-potomka. LDIF fajl je tekstualni fajl koji ima svoj format i koji služi za razmenu podataka između LDAP servera.

Primer jednog LDIF fajla koji definiše DIT čiji je koren stavka 'dc=example,dc=com' a sadrži dve druge stavke: kontejner 'people' i jednu stavku unutar 'people' kontejnera:

```
version: 1

## ROOT elemenat
dn: dc=example, dc=com
objectClass: dcObject
objectClass: organization
dc: example
description: Example kompanija
o: Example, Inc.

# Prvi nivo - stavka-kontejner 'people'
dn: ou=people, dc=example, dc=com
objectClass: organizationalUnit
ou: people
description: Svi zaposleni u firmi

# Drugi nivo - stavke unutar kontejnera 'people'
dn: cn=Joe Average, ou=people, dc=example, dc=com
objectClass: inetOrgPerson
cn: Joe Average
sn: Average
uid: javerage
mail: joe@example.com
mail: j.average@example.com
ou: sales
```

Atribut 'dn' (DistinguishedName) definiše 'poziciju' stavke unutar DIT. Sastoje se od svih RDN-ova kada se krećemo od korena stabla ka stavci. Tako npr. 'dn' osobe 'Joe Average' je: 'cn=Joe Average, ou=people, dc=example, dc=com'. Svaki od navedenih atributa, razdvojenih zarezom, predstavlja RDN stavke kad se krećemo od korena ('dc=example, dc=com'), preko kontejnera 'ou=people' do osobe za čiji smo RND odabrali atribut 'cn' ('cn=Joe Average'). Odabir RDN nije unapred definisan već zavisi od toga kako će administrator LDAP stabla organizovati stablo - npr. administrator je mogao umesto 'cn' atributa koristiti 'uid' atribut, sve dok je odabrani atribut ima jedinstvenu vrednost za svaku stavku

unutar kontejnera 'ou=people'. Odabir atributa koji će biti RDN najviše zavisi od toga kako će se LDAP podaci koristiti.

4.3 LDAP šeme

Klase objekata i atributi su definisani unutar **šeme**. LDAP administrator može da bira koje šeme, a sledstveno tome i klase objekata i atribute želi u svom stablu prilikom konfigurisanja LDAP servera. Naravno, pri tome se mora voditi računa da su neki atributi definisani unutar jedne šeme ali se koriste istovremeno i unutar druge šeme (atribut neće biti redefinisan unutar svake šeme koja ga koristi, već se definicija povlači iz šeme koja je prva definisala dati atribut — ovo znači da je potrebno u konfiguraciju servera uključiti i one šeme čije klase objekata možda nećemo koristiti ali koje sadrže atribute koje koriste klase objekata koje hoćemo koristiti).

Šema fajlovi su takođe tekstualni fajlovi koji sadrže definicije klasa objekata i atributa. Definicija sadrži osnovne parametre klase kao što su ime i OID (Object ID), koje atribute sadrži, koji atributi su obavezni a koji su opcioni i koja je roditeljska klasa.

4.3.1 OID

OID je bitan podatak jer on jednoznačno određuje i klase i atribute. OID nije specifika LDAP-a već se koristi za različite stvari (npr. u SNMP-u definiše takođe strukturu). OID se sastoji od niza brojeva razdvojenih tačkama koji sačinjavaju hijerarhijsku strukturu. Npr. OID klase 'inetOrgPerson' je '2.16.840.1.113730.3.2.2', a brojevi unutar označavaju, s leva na desno:

2 - zajednički ISO/ITU prostor OID-a

2.16 - prostor OID-a alociran pojedinim državama

2.16.840 - prostor OID-a alociran Sjedinjenim Američkim Državama

2.16.840.1 - prostor OID-a koje su SAD alocirale za kompanije

2.16.840.1.113730 - prostor OID-a alociran Netscape-u

2.16.840.1.113730.3 - prostor OID-a koji je Netscape odvojio za LDAP

2.16.840.1.113730.3.2 - Netscape-ove klase objekata

2.16.840.1.113730.3.2.2 - inetOrgPerson klasa koju je definisao Netscape

OID-e dodeljuje IANA (*Internet Assigned Numbers Authority*), organizacija koja je zadužena za upravljanje IP adresama i definisanje 'well-known' portova. Dodeljivanje OID-a je besplatno i isti se dodeljuje nekom entitetu (kompaniji, državi, obrazovnoj ustanovi i sl.). Entitet koji dobije svoj OID koristi ga kao početni OID za svoje potrebe i organizacija OID strukture ispod početnog broja je diskreciono pravo vlasnika OID-a. Na taj način se postiže jednoznačnost OID-a. Entitet koji je vlasnik nekog OID-a može, ali ne mora, objaviti strukturu pod-OID-a koji su pod njegovom kontrolom.

Atributi su takođe jednoznačno određeni svojim OID-om.

Kao administrator LDAP baze, sve dok koristite predefinisane šeme, nećete morati zahtevati svoje OID-e. Oni vam trebaju jedino ako želite definisati svoje posebne atribute ili klase objekata.

4.3.2 Primeri definicija klasa i atributa

Osnovna klasa je 'top' i njena definicija unutar šeme je:

```
objectclass ( 2.5.6.0 NAME 'top' ABSTRACT
MUST objectClass )
```

Definicija klase 'top' je jednostavna: njen OID je '2.5.6.0', ime joj je 'top', u pitaju je abstraktna klasa i ima jedan atribut: 'objectClass' koji je obavezan (MUST).

Definicija 'dcObject' klase je ovakva:

```
objectclass ( 1.3.6.1.4.1.1466.344 NAME 'dcObject'
DESC 'RFC2247: domain component object'
SUP top AUXILIARY MUST dc )
```

Klasa 'dcObject' je potomak klase 'top' (SUP top), ima OID '1.3.6.1.4.1.1466.344', ima ime 'dcObject', opis 'RFC2247: domain component object' (DESC), tipa AUXILIARY, što znači da je u pitanju tzv. pomoćna klasa, tj. nije moguće kreirati stavku koja bi bila samo klase 'dcObject', već je potrebna i još neka klasa (tipa STRUCTURAL). Postoji jedan atribut, pored onog nasleđenog od klase 'top', a to je 'dc' koji je obavezan (MUST).

Strukturalna klasa, tj. klasa od koje se mogu praviti stavke prikazana je u sledećem primeru:

```
objectclass ( 2.5.6.2 NAME 'country' SUP top STRUCTURAL
DESC '2 character iso 3611 assigned country code'
MUST c
MAY ( searchGuide $ description ) )
```

U pitanju je klasa 'country' koja je izvedena iz klase 'top', strukturalnog tipa, sa OID-om '2.5.6.2', datim opisom ('2 character iso 3611 assigned country code') i tri atributa: 'c' koji je obavezan i 'searchGuide' i 'description' koji su opcioni. Iz klase 'top' nasleđen je obavezni atribut 'objectClass'.

Sami atributi su takođe definisani na sličan način unutar šema. Npr. atribut 'name' je definisan kao:

```
attributetype ( 2.5.4.41 NAME 'name'
EQUALITY caseIgnoreMatch
SUBSTR caseIgnoreSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32768} )
```

OID ovog atributa je 2.5.4.41, a tip podataka je takođe određen OID-om (DirectoryString, UTF8 string), čija maksimalna dužina može biti 32768 znakova. Parametar EQUALITY određuje način na koji se poređi ovaj parametar ako je uslov pretrage potpuno zadat (name='vrednost') — u ovom slučaju poređenje ne pravi razliku između malih i velikih slova (caseIgnoreMatch - ovo, kao i ostala poređenja su takođe definisani unutar šeme i imaju svoj OID!). Parametar SUBSTR definiše kako se vrši poređenje ako je dat deo vrednosti sa 'wildcard' znakom '*' — u ovom slučaju se takođe ne pravi razlika između malih i velikih slova. Pošto nije naveden parametar SINGLE-VALUE, podrazumevana vrednost za ovaj atribut je da može unutar jedne stavke imati više vrednosti.

Šeme su takođe sastavni deo DIT.

4.4 OpenLDAP

OpenLDAP je podrazumevana implementacija LDAP servera na Linuxu, mada ne i jedina. OpenLDAP paket aplikacija se sastoji od serverskih i klijentskih aplikacija.

Paketi koje treba instalirati da bi OpenLDAP server bio funkcionalan su:

- openldap
- openldap-servers
- openldap-clients
- compat-openldap

Osnovna aplikacija je **slapd** koja implementira LDAP server. OpenLDAP je baziran na plug-in arhitekturi, što se posebno odnosi na način na koji se smeštaju podaci unutar LDAP servera. Na raspolaganju je nekoliko 'back-end' plug-inova, a standardni je Berkli DB.

Pored **slapd** programa, serverski paket sadrži i:

slapacl Omogućava proveravanje pristupa atributima.

slapadd Dodaje podatke preko LDIF fajlova.

slapauth Omogućava proveravanje liste ID-eva radi autentifikacije i autorizacije.

slapcat Izvlači stavke iz LDAP back-end-a i snima ih u LDIF formatu.

slapdn Proverava listu DN-ova baziranu da dostupnim šemama.

slapindex Omogućava reindeksiranje podataka unutar LDAP baze.

slappasswd Omogućava kreiranje kriptovane lozinke za korisnika.

slapschema Omogućava da se proveri da li je baža kompatibilna sa zadatom šemom.

slaptest Testira konfiguraciju LDAP servera.

Upozorenje!

Sve slap programe je potrebno pokretati samo kada LDAP server nije pokrenut!*

Klijentske aplikacije mogu biti instalirane i na računarama koji ne izvršavaju OpenLDAP server, pošto navedene aplikacije omogućavaju udaljeni pristup serveru:

ldapadd Omogućava dodavanje stavki u LDAP stablo.

ldapcompare Omogućava upoređivanje datog atributa sa stavkom.

ldapdelete Omogućava brisanje stavki iz LDAP stabla.

ldapmodify Omogućava izmenu stavki u LDAP stablu.

ldapmodrdn Omogućava izmenu RDN-a stavke u LDAP stablu.

ldappasswd Omogućava postavljanje i izmenu korisničke lozinke u LDAP stablu.

ldapsearch Omogućava pretragu LDAP stabla.

ldapwhoami Omogućava da izvršite 'whoami' operaciju na LDAP stablu.

Izuzev komande 'ldapsearch', sve ostale 'ldap*' komande najčešće se koriste LDIF fajlovi kao argumenti.

4.5 Konfigurisanje OpenLDAP servera

Konfigurisanje OpenLDAP servera se sastoji iz dva dela: konfigurisanja klijentskih aplikacija kako bi se nakačile na server i konfigurisanje samog servera.

4.5.1 Konfigurisanje OpenLDAP klijenata

Konfiguracioni fajl za klijentske aplikacije je `/etc/openldap/ldap.conf`. Ovaj fajl se sastoji iz direktiva, praznih redova i komentara (počinju znakom '#').

Direktive se sastoje iz ključne reči (imena direktive), jednog ili više blanko znakova i vrednosti. Vrednost se NE STAVLJA UNUTAR ZNAKOVA NAVODA.

Najčešće korišćene direktive su:

URI `ldap URL` definiše server na koji se kače klijenti, port i vrstu konekcije (obična (`ldap://...`) ili kriptovana (`ldaps://...`)). Umesto URL mogu se koristiti direktive HOST i PORT ali su one zastarele.

BASE DN definiše početni DN za sve operacije. Ovaj DN ne mora biti root DN servera već može pokazivati na neko podstablo, u kom slučaju klijentima nisu dostupne stavke van tog podstabla.

BINDDN DN DN LDAP korisnika preko kojeg se pristupa LDAP serveru.

TLS_CACERT fajl definiše CA sertifikat koji će se koristiti za proveru serverskog sertifikata kada se koristi kriptovana konekcija.

TLS_CACERTDIR direktorijum direktorijum u kojem se nalaze CA sertifikati koji će se koristiti za proveru serverskog sertifikata kada se koristi kriptovana konekcija.

TLS_CERT fajl korisnički sertifikat kojim se korisnik može autentifikovati na LDAP server.

TLS_KEY fajl korisnikov tajni ključ koji odgovara sertifikatu.

4.5.2 Konfiguracija slapd servera

Prethodne verzije OpenLDAP servera su koristile konfiguracioni fajl `/etc/openldap/slapd.conf`, čiji je format, kao i kod `ldap.conf`, baziran na direktivama. No, tekuća verzija OpenLDAP servera koristi fajlove u LDIF formatu koji su smešteni u `/etc/openldap/slap.d/` direktorijumu. Od fajlova koji su nam zanimljivi, osnovni je `/etc/openldap/slap.d/cn=config.ldif`. Bitniji atributi su:

olcAllows: opcija... definiše koje opcije treba uključiti. Opcije su:

bind_v2 - omogućava prihvatanje zahteva po protokolu verzije 2.

bind_anon_cred - omogućava anonimno kačenje (bez autentifikacije) kada je DN prazan.

bind_anon_dn - omogućava anonimno kačenje kada DN nije prazan.

update_anon - omogućava procesiranje anonimnih operacija ažuriranja.

proxy_authz_anon - omogućava procesiranje proksiranih anonimnih zahteva za autorizaciju.

olcConnMaxPending: broj - definiše maksimalan broj konekcija na čekanju kod anonimne sesije. Podrazumevana vrednost je 100.

olcConnMaxPendingAuth: *broj* - definiše maksimalan broj konekcija na čekanju kod autentifikovane sesije. Podrazumevana vrednost je 1000.

olcDisallows: *opcija...* - definše koje osobine treba isključiti:

bind_anon - onemogućava anonimno kačenje na LDAP server

bind_simple - onemogućava 'simple' autentifikaciju

tls_2_anon - onemogućava anonimnu sersiju posle STARTTLS komande.

tls_authc - onemogućava STARTTLS komandu posle autentifikacije sesije.

olcIdleTimeout: *broj* - definiše koliko sekundi treba čekati da bi se zatvorila 'idle' konekcija. Podrazumevana vrednost je 0.

olcLogFile: *fajl* - definiše log fajl.

olcReferral: *URL* - definiše URL drugog LDAP servera kojeg treba propitati ako postojeći server ne može da odgovori na upit.

olcWriteTimeout: *broj* - definiše vreme u sekundama koliko treba čekati pre nego što se zatvori konekcija koja ima zahtev za upis podataka. Podrazumevana vrednost je 0.

Drugi fajl koji je od interesa je fajl koji definiše back-end bazu, a to je podrazumevano Berkli DB (*Berkeley DB*). Konfiguracioni fajl ima malo čudno ime:

/etc/openldap/slap.d/cn=config/olcDatabase={2}bdb.ldif. Atributi od interesa su:

olcReadOnly: *boolean* - definiše da li se baza koristi u read-only modu. Podrazumevana vrednost je FALSE.

olcRootDN: *DN* - DN korisnika koji ima sva prava nad LDAP bazom i ne može biti ograničen pravima pristupa. Podrazumevana vrednost je 'cn=Manager,dc=my-domain,dc=com'.

olcRootPW: *string* - definiše lozinku korisnika defenisanog 'olcRootDN' atributom. Lozinka može biti nekriptovana i kriptovana (heš).

olcSuffix: *DN* - definiše domen koji je root stabla. Parametar treba da bude DN koji označava FQDN. Podrazumevana vrednost je 'dc=my-domain,dc=com'.

Heš lozinka se može kreirati 'slappasswd' komandom:

```
$ slappasswd
New password:
Re-enter new password:
{SSHA}WczWsyPEnMchFf1GRTweq2q7XJcvmSxD
```

Dobijeni tekst treba kompletno prekopirati kao vrednost 'olcRootPW' atributa.

Definicija šeme je data u /etc/openldap/slap.d/cn=config/cn=schema/ direktorijumu.

4.6 Konfigurisanje sistema da koristi OpenLDAP za autentifikaciju

Da bismo jedan server u mreži podesili da se autentificuje na zajednički LDAP server, potrebno je prvo instalirati potreban softver:

```
# yum install openldap openldap-clients nss-pam-ldapd migrationtools
```

Paket 'migrationtools' obezbeđuje skup bash i Perl skriptova koji mogu da pomognu da se migrira lokalni skup korisnika u LDAP bazu. Instalirani skriptovi su u `/usr/share/migrationtools/`.

Prvo je potrebno podesiti `/etc/openldap/ldap.conf` i proveriti (npr. komandom `ldapsearch`) da li možemo pristupiti LDAP serveru.

Zatim je potrebno podesiti parametre koji definišu domen i bazu LDAP stabla u fajlu `/usr/share/migrationtools/migrate_common.ph`:

```
# Default DNS domain  
$DEFAULT_MAIL_DOMAIN = "padl.com";  
  
# Default base  
$DEFAULT_BASE = "dc=padl,dc=com";
```

Na raspolaganju su različiti skriptovi, zavisno od toga šta želimo da migriramo:

migrate_all_online.sh - fajlovi iz `/etc/` direktorijuma kada je LDAP server aktivan.

migrate_all_offline.sh - fajlovi iz `/etc/` direktorijuma kada je LDAP server neaktivan/ne-dostupan.

Modul 5: NFS

Network File System (NFS) omogućava udaljenim računarima da zakače fajl sisteme preko mreže i da interaguju sa njima kao da su u pitanju lokalni fajl sistemi. NFS je razvio Sun Microsystems, koristeći tehnologiju RPC (*Remote Procedure Calls*).

Trenutno postoje tri verzije NFS protokola. Najstarija i najpodržanija verzija je NFSv2. NFSv3 podržava sigurne asinhronne upise i robustnija je i otpornija od NFSv2. NFSv3 takođe podržava 64-bitne ofsete tako da omogućava klijentima da pristupe fajlovima koji su veći od 2GB.

NFSv4 je tekuća verzija koja radi dobro ukoliko saobraćaj prolazi kroz fajervolove i više ne zahteva **rpcbind** servis. Takođe, ova verzija podržava ACL.

Linux podržava sve tri verzije protokola, s tim što se verzija NFSv4 smatra podrazumevnom ako konfiguracijom nije drugačije određeno.

Sve verzije NFS mogu koristiti TCP mrežni protokol, s tim što je kod NFSv4 on obavezan. NFSv2 i NFSv3 mogu koristiti i UDP protokol, koji može dati bolje rezultate kod manjih mreža kod kojih ne postoji zagrušenje. U suprotnom slučaju, TCP protokol je bolji jer ne zahteva, kao UDP, da se zbog jednog izgubljenog paketa ponavlja kompletna RPC operacija, već samo retransmisiju izgubljenog paketa.

5.0.1 RPC

Remote Procedure Call metod omogućava da se na udaljenom serveru izvršavaju funkcije i procedure kao da su na lokalnom sistemu. Sun RPC protokol omogućava da udaljena strana dinamički dobija informacije koje funkcije i procedure su implementirane. Sam saobraćaj preko mreže je transparentan za aplikaciju i bazira se na tzv. 'portmap' servisu. Ovaj servis služi da registruje lokalne izvršioce funkcija i procedura i da im dodeli slobodan port. Druga glavna funkcija portmap servisa je da omogući udaljenim klijentima da dobiju informacije koji RPC servisi su aktivni na lokalnom serveru i na kojim portovima oni slušaju.

Ova osobina dinamičkog dodeljivanja portova (jedini poznati port je port 111/tcp samog portmap servisa) čini da je skoro nemoguće napraviti odgovarajuću konfiguraciju fajervola jer je unapred nepoznato koji port će neki od servisa dobiti. Kasnija proširenja portmap protokola su omogućavala da pojedini servis od portmapa zahteva odgovarajući, unapred zadati, port.

U verziji 6 RedHat Enterprise Linux-a i njegovih klonova, servis 'portmap' je zamjenjen servisom 'rpcbind'.

Verzija NFSv4 ne zahteva mnoge dodatne RPC servise koji su potrebni za NFSv2 i NFSv3 jer ih sam NFSv4 protokol implementira, kao što su 'rpcbind', 'lockd' i 'rpc.statd' servisi. Servis 'rpc.mountd' je i dalje potreban kao nezavistan servis. NFSv4 koristi port 2049/tcp.

Komandom 'rpcinfo' moguće je videti koji RPC servisi su pokrenuti i koji portovi se koriste:

```
# rpcinfo -p
```

5.1 NFS servisi

NFS sistem je delimično implementiran u samom kernelu Linuxa, a dodatni servisi su implementirani kao nezvisni RPC servisi:

nfs NFS servis se pokreće komandom 'service nfs start'.

nfslock Ovaj servis se zasebno startuje i služi da klijenti mogu zaključavati fajlove na serveru.

rpcbind Ovaj servis je zamena za stariji 'portmap' servis i služi da rezerviše portove za lokalne RPC servise i oglasi ih udaljenim klijentima. Takođe, ovaj servis uspostavlja konekcije između RPC servera i klijenta. Ne koristi se u NFSv4 verziji NFS protokola.

rpc.nfssd Ovaj servis implementira odgovarajuću NFS serversku instancu.

NFS servisi koriste sledeće RPC servise za potpunu implementaciju NFS protokola:

rpm.mountd Ovaj servis se pokreće na NFS serveru i zadužen je za procesiranje MOUNT zahteva od strane NFSv2 i NFSv3 klijenata.

lockd Ovaj servis je implementiran kompletno u kernelu (kao zasebana nit) i implementira *Network Lock Manager* protokol, koji omogućava NFSv2 NFSv3 klijentima da zaključavaju fajlove na serveru. Ovaj servis mora da bude aktivan i na serverskoj i na klijentskoj strani. Automatski se startuje pokretanjem NFS servisa i kačenjem udaljenog fajl sistema.

rpc.statd Ovaj servis implementira *Network Status Monitor* i bazično služi da obaveštava NFS klijente kada se NFS servis restartuje na serveru. NFSv4 ga ne koristi.

rpc.quotad Ovaj RPC servis implementira kvote za udaljene korisnike.

rpc.idmapd Ovaj servis koristi isključivo NFSv4 radi mapiranja NFSv4 imena (oblika *korisnik@domen*) u lokalne UID-e i GID-ove. Moran je zasebno konfigurisati kroz konfiguracioni fajl /etc/idmapd.conf ali nije potreban ako svi klijenti i serveri imaju isti DNS domen.

5.2 Konfigurišanje NFS servera

NFS server se konfiguriše na dva načina:

- editovanjem fajla /etc(exports
- preko komande exportfs

Fajl /etc(exports je tekstualni fajl koji se sastoji iz praznih linija, linija sa komentarima (linija počinje znakom '#') i linijama koje sadrže definicije eksportovanih fajl sistema.

Linije koje definišu eksportovane fajl sisteme imaju format:

```
/eksportovani/fajl/sistem host{opcije} ...
```

Linija se sastoji iz specifikacije direktorijuma koji se eksportuje i jednog ili više specifikacija hostova, koji mogu imati specificirane opcije eksportovanja. Ova linija mora da bude napisana tako da zadovolji sledeća pravila:

- dugačke linije mogu se prelomiti u više linija tako što se, kao zadnji znak u liniji navede '\';
- svaki eksportovani fajl sistem treba biti zapisan u jednoj (logičkoj) liniji;
- lista autorizovanih hostova mora biti rastavljena blanko znakovima;
- opcije koje se odnose na neki host moraju biti zapisane unutar običnih zagrada '()' ;
- između specifikacije autorizovanog hosta i otvorene zgrade koja obuhvata opcije ne sme da bude razmaka;
- opcije unutar zagrada su razdvojene zarezom ',', bez belina.

Parametar '/eksportovani/fajl/sistem' predstavlja putanju do direktorijuma od kojeg se eksportuje deo lokalnog fajl sistema. Idealno, ovaj direktorijum je ujedno i tačka kačenja lokalnog fajl sistema, pošto NFS najbolje radi kada se eksportuju kompletni lokalni fajl sistemi. Postoje opcije koje su specifično namenjene kada se ne izvoze kompletni lokalni fajl sistemi već samo neki njegov deo.

Parametar 'host' je specifikacija jednog ili više hostova koji su autorizovani da mogu lokalno zakačiti eksportovani fajl sistem. Ovaj parametar može se navesti na više načina:

Pojedinačna mašina - ako se autorizuje pojedinačna mašina, ona se može navesti ili preko FQDN-a, imena hosta (definisanog u fajlu /etc/hosts na lokalnom serveru) ili, najbolje, preko IP adrese.

IP podmreža - autorizacija cele podmreže se može zadati u obliku $a.b.c.d/z$ gde je $a.b.c.d$ mrežna adresa a z broj bitova mrežne maske, ili u obliku $a.b.c.d/m.n.o.p$, gde je $a.b.c.d$ mrežna adresa a $m.n.o.p$ mrežna maska.

NIS mrežna grupa - NIS mrežna grupa može biti autorizovana ako se navede u obliku $@ime-grupe$.

Grupa mašina iz istog domена - ako želimo da autorizujemo grupu mašina koja pripada istom domenu ali koja nije unutar iste IP podmreže ili NIS grupe, onda možemo koristiti wildcard znak '*', u obliku *.domen.tld. U ovom slučaju znak '*' menja bilo koje ime, ali ne i poddomen — drugim rečima specifikacija *.domen.tld' obuhvata host 'www.domen.tld' ali ne i 'www.poddomen.domen.tld'.

Parametar 'opcije' je neobavezani parametar koji definiše opcije pod kojima se zadatom autorizovanom hostu eksportuje dati fajl sistem. Opcije su:

secure Ova opcija zahteva da se klijentski zahtevi šalju sa porta manjeg od 1024. Ovo je podrazumevana opcija. Ukoliko želimo da je isključimo, trebamo zadati opciju **insecure**.

rw Ova opcija omogućava da klijent može i da čita i da piše po eksportovanom fajl sistemu. Podrazumevano stanje je da klijent može samo da čita eksportovani fajl sistem (opcija **ro**).

async Ova opcija označava da će server potvrditi izmene na fajl sistemu pre nego što se one stvarno zapisuju. Time se mogu poboljšati performanse NFS servera ali opcija može biti opasna u slučaju kraha servera gde je moguće da se izgube podaci koji nisu zapisani na disk. Ova opcija je u ranijim verzijama bila podrazumevana, ali više nije.

sync Ova opcija označava da NFS server neće potvrditi operaciju izmene sadržaja sve dok se ta izmena ne zapiše na disk. Na ovaj način se dobija sistem koji je sigurniji i otporniji na kraj NFS servera uz cenu nižih performansi. Ovo je sada podrazumevana opcija.

no_wdelay Ukoliko je u upotrebi 'sync' opcija, NFS server će odložiti zahtev za upis na disk ukoliko smatra da će uskoro doći novi zahtev za upis koji se može izvesti jednom komandom upisa na disk. Time se mogu povećati performanse sistema. No, ukoliko NFS serveru stižu zahtevi za upis koji nisu u međusobnoj korelaciji, onda ovo odlaganje upisa može smanjiti performanse. Ovom opcijom se to odlaganje isključuje. Suprotna opcija je **wdelay** koja je i podrazumevana. Obe ove opcije nemaju značenje ako je zadata opcija 'async'.

nohide Ova opcija utiče na slučaj da su eksportovana dva fajl sistema od kojih je jedan poddirektorijum drugog sistema. Klijent, da bi mogao pristupiti oba fajl sistema mora zakačiti oba. Ukoliko zakači samo onaj nadređeni fajl sistem (koji obuhvata drugi eksportovani sistem), onda će direktorijum od kojeg počinje nezakačeni eksportovani fajl sistem biti prazan. Opcijom 'nohide' se omogućava klijentu da pristupa i sadržaju drugog, nezakačenog fajl sistema koji je poddirektorijum zakačenog sistema. Suprotna opcija je **hide**. Ovu opciju je potrebno upotrebljavati sa pažnjom, a inače je primenljiva samo kod hostova koji su pojedinačne mašine.

no_subtree_check Ova opcija isključuje proveru podstabla. Provera podstabla se izvodi kada je eksportovano podstablo lokalnog fajl sistema a ne ceo fajl sistem. U tom slučaju za svaku aktivnost na fajlu, NFS server mora da detektuje da li fajl pripada odgovarajućem lokalnom fajl sistemu, kao i da li pripada eksportovanom ili neeksportovanom delu stabla. Ova opcija je podrazumevana u novijim implementacijama. Suprotna opcija je **subtree_check**.

insecure_locks, no_auth_nlm Ovo su dva sinonima za istu opciju koja instruira NFS server da ne zahteva autentifikaciju kod zahteva za zaključavanje fajlova (NLM protokol). Podrazumevana opcija je suprotna: **secure_locks**, odnosno **auth_nlm**.

mountpoint=putanja, mp=putanja Opet su dve opcije sinonimi. Ove opcije obezbeđuju da će se fajl sistem eksportovati samo ukoliko se lokalni fajl sistem uspešno zakači. Ako se parametar 'putanja' izostavi, pretpostavlja se da je tačka kačenja lokalnog fajl sistema ista kao eksportovana putanja. Ako je putanja zadata onda se ona smatra tačkom kačenja lokalnog fajl sistema.

root_squash Ova opcija služi za mapiranje korisnika na klijentskoj mašini u korisnike na NFS serveru. Sa opcijom 'root_squash' svi zahtevi root korisnika na klijentskoj mašini se mapiraju u anonimnog korisnika na lokalnoj mašini. Ostali korisnici se mapiraju prema identičnim UID-ima i GID-ovima. Suprotna opcija je **no_root_squash**.

all_squash Ova opcija mapira sve korisnike na klijentskoj mašini u anonimne korisnike na lokalnoj mašini. Podrazumevana je suprotna opcija, **no_all_squash**.

anonuid, anongid Ove dve opcije definišu UID i GID anonimnog korisnika (podrazumevane vrednosti su 65534 za obe opcije).

Primer jednog /etc(exports fajla je:

```
# Primer /etc/exports fajla
/projects      proj*.local.domain(rw)
/usr          *.local.domain(ro) @trusted(rw)
/home/joe     pc001(rw,all_squash,anonuid=150,anongid=100)
/pub          *(ro,insecure,all_squash)
```

5.2.1 Komanda `exportfs`

Prilikom startovanja NFS servisa izvršava se i komanda `exportfs`, koja učitava sadržaj `/etc/exports` fajla. Ova komanda kasnije može biti korišćena da eksportuje ili deeksportuje fajl sisteme bez restartovanja NFS servera, što je potrebno ako se samo menja fajl `/etc/exports`.

Sledeće opcije su dostupne za komandu `exportfs`:

- r** Reeksportuje sve eksportovane fajl sisteme iz fajla `/etc/exports`.
- a** Označava da se druga opcija (export ili deeksport) primenjuje na sve eksportovane fajl sisteme. Bez dodatnih opcija ekvivalentna opciji **-r**.
- o fajl-sistemi** Eksportuje fajl sisteme koji su navedeni kao argumenti opcije (u istom formatu kao i u fajlu `/etc/exports`). Zadati fajl sistemi ne moraju biti navedeni u fajlu `/etc/exports`.
- u** Deeksportuje sve prethodno eksportovane fajl sisteme.

Bez opcija, komanda `exportfs` prikazuje eksportovane fajl sisteme.

5.3 Prikaz eksportovanih fajl sistema na klijentskoj mašini

Da biste videli koji fajl sistemi su dostupni sa NFS servera možete koristiti komandu 'showmount':

```
$ showmount -e nfs-server
Export list for nfs-server
/exportfs/foo
/exportfs/bar
```

5.4 Kačenje eksportovanog fajl sistema

Za kačenje NFS eksportovanog fajl sistema koristi se standardna 'mount' komanda:

```
# mount -t nfs -o opcije server:/export
/lokalni/dir
```

Gornja komanda kači `/export` direktorijum sa servera 'server' na lokalni direktorijum `/local/dir` koristeći zadate `opcije`. Ove opcije su specifične za klijentsku stranu i ne treba ih mešati sa opcijama iz `/etc/exports` fajla.

Najčešće korišćene opcije su:

intr Omogućava da NFS zahtevi budu prekinuti ako NFS server nije dostupan.

nfsvers=broj Definiše koja verzija protokola (2, 3 ili 4) se koristi. Podrazumevana vrednost je najnovija verzija koju server podržava.

noacl Isključuje procesiranje ACL-ova.

nolock Isključuje lokovanje fajlova.

noexec Blokira izvršavanje binarnih programa sa deljenog diska.

nosuid Blokira SUID i SGID bitove.

port=*broj* Specificira port na kojem sluša NFS server. Ako nije definisana (broj je 0) komanda 'mount' će pitati udaljeni servis 'rpcbind' ili, ako on nije pokrenut, koristiti podrazumevani port 2049/tcp.

rsize=*broj* Definiše veličinu bloka za operacije čitanja. Veličina treba da bude deljiva sa 4096 i predstavlja veličinu u bajtovima.

wsize=*broj* Definiše veličinu bloka za operacije pisanja. Veličina treba da bude deljiva sa 4096 i predstavlja veličinu u bajtovima.

tcp Definiše da se za komunikaciju koristi TCP protokol.

udp Definiše da se komunikaciju koristi UDP protokol.

Dodatne opcije možete pogledati u man strani za komandu 'mount'.

Permanentno kačenje fajl sistema se izvodi tako što se doda zapis u fajl /etc/fstab kao i za bilo koji drugi lokalni fajl sistem. Poslednja dva parametra u zapisu za /etc/fstab treba da budu '0' (fajl sistem se ne može dampovati) i '0' (fajl sistem ne treba čekirati komandom 'fsck').

Indeks

D

DHCP.....	23
client.....	26
dhcpd.lease.....	26
dhcrelay.....	26
relaying	26
DNS.....	13
.arpa	13
AXFR.....	18
BIND	18
ccTLD	13
gTLD	13
ICANN	13
IXFR	18
NOTIFY	18
organizacija	13
resource record.....	15
RR	15
transfer.....	18
zapisi resursa.....	15
zona.....	13

N

NFS	35
/etc(exports.....	36
exportfs	39
mount opcije.....	39
RPC	35
servisi	36
showmount.....	39
NTP.....	7
ntp.conf	9
stratum	7

L

LDAP	27
šeme	29
atributi	27
LDIF.....	28
OID	29
OpenLDAP	30
konfiguracija, 32	
organizacija	27

Property of Admin Training Center



Admin Training
Center

L2-4 Administracija osnovnih korisničkih servisa

Veselin Mijušković, Marko Uskoković, Ljubiša Radivojević

Copyright © 2014 Veselin Mijušković, Marko Uskoković, Ljubiša Radivojević

OBJAVIO ADMIN TRAINING CENTER

www.atc.rs

Licencirano po Creative Commons Attribution-NonCommercial 3.0 Unported License (the "License"). You may not use this file except in compliance with the License. You may obtain a copy of the License at <http://creativecommons.org/licenses/by-nc/3.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Prvo izdanje, 2014

Sadržaj

Uvod	7
Tipografske konvencije	7
Modul 1: Web servis	9
1.1 HTTP protokol	9
1.2 Apache web server	10
1.2.1 Instalacija	10
1.2.2 Konfiguracija	10
1.2.3 Osnovne direktive	10
1.2.4 Kontrola web sadržaja	12
1.2.5 Dodatni Moduli	14
1.2.6 Logovi	15
1.2.7 .htaccess fajlovi	16
1.2.8 Osnovna HTTP autentifikacija	16
1.2.9 Virtuelni hostovi	16
1.3 SSL/TLS Enkripcija	17
1.3.1 SSL sertifikati	17
1.3.2 Konfigurisanje Apache servera za SSL/TLS	18
1.4 PHP	19
1.5 CGI	19
Modul 2: MySQL DBMS	21
2.1 Uvod	21
2.2 Instalacija i inicijalizacija	21
2.2.1 Binarni paketi	21
2.2.2 Instalacija u različitim distribucijama	21
2.2.3 Konfiguracija	22
2.2.4 Inicijalizacija baza	22
2.3 Rad sa MySQL bazama	23
2.3.1 Kreiranje baze	23
2.3.2 Dropovanje baze	23
2.3.3 Pravljenje tabele unutar baze	23

2.3.4	Unošenje podataka u tabelu	23
2.3.5	Dobijanje specifičnih podataka iz tabele	24
2.3.6	Dropovanje tabela unutar baze	24
2.3.7	Dodavanje korisnika i podešavanje prava	24
2.4	Bekap i restore podataka	24
2.5	Pregled programa MySQL distribucije	24
2.6	Pregled MySQL naredbi	25
2.6.1	Naredbe za opisivanje podataka (Data Definition Statements)	25
2.6.2	Naredbe za upravljanje podacima (Data Manipulation Statements)	25
2.6.3	Naredbe za upravljanje nalozima (Account Management Statements)	26
2.7	Pregled MySQL privilegija	27
2.8	Primeri najčešćih administratorskih aktivnosti	27
Modul 3:	Mail servis	29
3.1	Komponente sistema elektronske pošte	29
3.2	Struktura elektronskih poruka	30
3.2.1	Koverta (envelope)	30
3.2.2	Zaglavlja (headers)	30
3.2.3	Telo poruke (body)	31
3.3	Struktura skladišta za e-mail poruke	31
3.3.1	Mbox	31
3.3.2	Maildir	31
3.4	Protokoli za slanje i primanje pošte	31
3.4.1	SMTP	31
3.4.2	POP3	32
3.4.3	IMAPv4	32
3.5	Podešavanje DNS zapisa za e-mail	33
3.6	Postfix SMTP server	34
3.6.1	Komponente Postfixa	34
3.6.2	Postfix komande	34
3.6.3	Konfiguracioni fajlovi	35
3.7	Dovecot POP i IMAP	39
3.7.1	Instalacija	39
3.7.2	Konfiguracioni fajlovi	39
3.7.3	Primer konfiguracije: virtuelni korisnici definisani u MySQL bazi	41
Modul 4:	Squid proxy servis	45
4.1	Uvod	45
4.2	Osnovna konfiguracija	45
4.3	Transparent proxy server	47

Modul 5: Samba fajl server	49
5.1 Uvod	49
5.2 Instalacija	49
5.2.1 Samba kao klijent	49
5.2.2 Samba alatke	49
5.2.3 Korlšćenje cifs	50
5.3 Samba kao server	50
5.4 Načini autentifikacije	50
5.4.1 Korisnički nivo sigurnosti (User level security)	50
5.4.2 Sigurnost na nivou šera (Share level security)	50
5.4.3 NT4 domen mod sigurnosti (Domain security mode)	50
5.4.4 Aktivni direktorijum mod sigurnosti (ADS security mode)	50
5.5 Deljenje fajlova i direktorijuma	51
5.6 Samba kao PDC	51
5.6.1 Uvod	51
5.7 Konfigurisanje Sambe	51
5.8 Podešavanje Windows klijenata	53
5.9 Podešavanje Linux klijenata	53
5.10 Implementacija korisničkih profila	54

Property of Admin Training Center

Uvod

Skripta je podeljena na poglavlja, a poglavlja na sekcije. Unapred skrećemo pažnju polaznicima da je ova skripta samo deo dokumentacije koju oni treba da koriste. Polaznicima se savetuje da pročitaju man i help strane za svaku komandu, kao i da potraže na Internetu dodatne informacije i načine kako da iste koristite.

Tipografske konvencije

Radi lakšeg snalaženja u tekstu, koristili smo neke tipografske konvencije na koje vam ovde skrećemo pažnju:

- ukoliko se uvodi neki značajan pojam, on će u prvom pomenu biti isписан **proporcionalnim bold tekstom**;
- boldovano su prikazane i neke značajne tvrdnje na koje treba obratiti pažnju u tekstu;
- *proporcionalnim italicom* su napisane reči na stranom jeziku, najčešće engleskom;
- nazivi fajlova u tekstu su ispisani *neproporcionalnim fontom*

Oblik neke komande prikazan je na sledeći način:

```
# komanda [opcije] argument...
```

Deo koji je napisan uspravnim fontom se unosi kako je napisan. Opcioni deo, koji se može izostaviti je naveden uvek u uglastim zagradama (same zgrade se ne unose). Ukoliko je tekst isписан *italicom* to znači da je u pitanju neki opšti naziv i umesto njega treba uneti neku stvarnu vrednost. Ukoliko iza teksta stoje tri tačke '...' to znači da se taj deo može ponavljati.

Deo teksta koji se odnosi na direktni unos korisnika i ispis računara, kao i sadržaji fajlova i sl. će biti prikazani u zasebnom bloku:

```
$ ls -F
myscript.sh*  Vezbe/
```

Boldovanim tekstom je prikazano ono što polaznik treba da unese onako kako je napisano u skripti. Regularnim tekstom je prikazan ispis programa koji ne treba unositi.

Gde god smo mislili da nešto posebno treba naglasiti, to smo naveli na kao napomenu ili upozorenje na sledeći način:

Napomena:

Hard linkovi ne mogu biti kreirani za direktorijume, već samo za regularne fajlove!

Upozorenje!

Uvek koristite 'modprobe' za učitavanje modula jer ćete tako učitati i zavisne module!

Na kraju, ukoliko se zadaje deo koda (npr. skripte komandnog interpretera bash), onda će to izgledati ovako:

```
1 #!/bin/bash
2 #
3 # helloworld.sh - standardni Hello, World skript
4 #
5
6 echo "Hello, World!"
```

Modul 1: Web servis

1.1 HTTP protokol

HTTP (*hypertext transfer protocol*) predstavlja metod za prenos informacija na WWW-u. Verzija 1.1 HTTP protokola specificirana je RFC-om 2616 iz 1999. godine.

```
http://user:pass@example.com:8081/app/page?var=value#anchor  
protokol://login@host:port/putanja?upit#interni_link
```

Informacije se u HTTP protokolu prenose u obliku HTML strana koje klijent zatraži od servera specificiranjem određenog URI-a (Uniform Resource Identifier). URI se sastoji od više komponenti, predstavljenih na sledećem dijagramu:

HTTP protokol se izvodi uspostavljanjem TCP konekcije između klijenta i servera na portu na kome sluša HTTP server (podrazumevano port 80). Klijent šalje serveru zahtev za stranom koji se sastoji od metoda (GET, POST...), putanje do resursa koji se zahteva (/index.php), verzije protokola (HTTP/1.1) i različitih opcionih zaglavlja (npr. Host ili Accept-Language). Primer komunikacije:

```
turncoat@bot:~$ telnet www.example.com 80  
Trying 10.0.0.1...  
Connected to www.example.com.  
Escape character is '^]'.  
GET /index.php HTTP/1.1  
Host: www.example.com  
  
HTTP/1.1 200 OK  
Date: Mon, 26 Mar 2007 16:37:06 GMT  
Server: Apache/2.0.59 (Gentoo) mod_ssl/2.0.59 OpenSSL/0.9.7d  
Transfer-Encoding: chunked  
Content-Type: text/html; charset=iso-8859-2  
  
<!doctype html public "-//w3c//dtd html 4.0 transitional//en``>  
<html>  
...
```

Klijent se u ovom primeru povezao na server www.example.com na port 80 i zatražio stranicu http://www.example.com/index.php putem HTTP protokola verzije 1.1. Server je na takav zahtev odgovorio linijom sa statusnim kodom 200 kojim je naznačio da stranica postoji i da će biti prosleđena u nastavku odgovora. Kodovi koji počinju cifrom 3 označavaju redirektovanje na drugu lokaciju zbog nekog određenog razloga, kodovi sa cifrom 4 na početku označavaju različite klijentske greške (npr. pristup zabranjenoj ili nepostojećoj stranici), dok kodovi sa 5 na početku označavaju neku serversku grešku (npr. u konfiguraciji ili resursima).

1.2 Apache web server

Apache HTTP je web server koga razvija Apache Software Foundation i koji predstavlja najzastupljeniji web server na Internetu još od 1996. godine, sa trenutno više od 58% udela na tržištu.

Apache HTTP server je licenciran pod Apache License, Version 2.0, Open Source licencom. Sajt projekta se nalazi na adresi <http://httpd.apache.org/>

1.2.1 Instalacija

Apache HTTP server se može instalirati kompajliranjem izvornog koda sa sajta projekta, ili instaliranjem binarne verzije koja dolazi uz najveći broj distribucija. Ubuntu distribucija, na primer, u svojoj serverskoj varijanti daje mogućnost instaliranja LAMP (Linux, Apache, MySQL, PHP) sistema kao dodatak standardnoj instalaciji, čime se jednostavno dobija inicijalno podešen Apache HTTP server. Na standardnim instalacijama Ubuntu-a, dovoljno je instalirati paket apache2, dok se na RedHat/CentOS distribucijama ovaj paket zove httpd.

1.2.2 Konfiguracija

Konfiguracioni fajlovi Apache HTTP servera se nalaze u direktoriju /etc/httpd ili /etc/apache2. Podrazumevano ime konfiguracionog fajla zavisi od distribucije, ali najčešće je to httpd.conf ili apache2.conf, kao i svi ostali fajlovi koje ovi fajlovi mogu uključiti (vidi direktivu *Include*).

Apache HTTP server se može startovati alatkom apache2ctl sa parametrom 'start', stopirati sa parametrom 'stop', a sa parametrom 'configtest' se može proveriti sintaksa konfiguracionih fajlova. Distribucije često uključuju i inicijalizacionu skriptu /etc/init.d/apache2 ili /etc/init.d/httpd kojima se Apache HTTP server može automatski startovati prilikom startovanja sistema.

Sledi primer jednostavnog, ali funkcionalnog konfiguracionog fajla za Apache HTTP server na mašini www.primer.com koji služi sadržaj iz direktorijuma /var/www/html:

```
ServerRoot "/etc/httpd"
PidFile /var/run/httpd.pid
Listen 80
User apache
Group apache
AddDefaultCharset UTF-8
DirectoryIndex index.html index.cgi index.pl index.php index.xhtml
UseCanonicalName Off
TypesConfig /etc/httpd/mime.types
DefaultType text/plain
HostnameLookups Off
ServerName www.primer.com:80
DocumentRoot "/var/www/html"
ErrorLog /var/log/httpd/error.log

<Directory "/var/www/html">
    Order allow,deny
    Allow from all
</Directory>
```

1.2.3 Osnovne direktive

Apache HTTP server ima veliki broj konfiguracionih opcija i direktiva kojima se može fino podesiti ponašanje i mogućnosti koje pruža. Sledi pregled osnovnih direktiva:

ServerName Ime mašine i port koje server koristi da identifikuje sebe

```
ServerName www.primer.com:80
```

ServerRoot Osnovni direktorijum unutar kog se traže svi fajlovi sa relativnim putanjama.

```
ServerRoot /etc/apache2/
```

DocumentRoot Direktorijum koji predstavlja koren podstabla fajl sistema vidljivog sa weba

```
DocumentRoot /var/www
```

Include Uključuje sadržaj drugih konfiguracionih fajlova u glavni fajl.

```
Include /usr/local/apache2/conf/vhosts/*.conf
```

```
LoadModule -- Učitava objektni fajl ili biblioteku i aktivira modul tog imena
```

```
LoadModule ime_modula ime_fajla
```

ServerAdmin - E-mail adresa koja se koristi u porukama o greškama koje se daju klijentima.

```
ServerAdmin webmaster@primer.com
```

AddDefaultCharset Specificira podrazumevanu vrednost parametra "media-type" (tj. ime enkodingu karaktera) koja se koristi u slučaju kada je "content-type" vrednost fajla ili text/plain ili text/html

```
AddDefaultCharset utf-8
```

Listen - IP adresa i/ili port na kojima sluša server

```
Listen 192.168.1.10:80
```

```
Listen 8080
```

PidFile Ime fajla u koji se zapisuje proces ID Apache HTTP daemona.

```
PidFile /var/run/apache.pid
```

User Unix korisničko ime ili korisnički ID pod čijim privilegijama se izvršava server

```
User www-data
```

```
User #81
```

Group Grupa ili ID grupe pod čijim privilegijama se izvršava server

```
Group unix-groupname
```

```
Group #unix-groupid
```

AddLanguage Kontroliše automatsko učitavanje određene verzije fajla klijentske promenljive "Accept-Language"

```
AddLanguage en .en
```

```
AddLanguage rs .rs
```

AddType Kontroliše tip fajla na osnovu njegove ekstenzije

```
AddType image/gif .gif
```

DefaultLanguage Podrazumevani jezik za sve fajlove u tom opsegu

```
DefaultLanguage en
```

1.2.4 Kontrola web sadržaja

Sekcija **Files** omogućava da se navedu direktive specifične samo za fajlove određenog imena (bez putanje), omogućavajući finu kontrolu pristupa i opcije na nivou pojedinačnih fajlova. Sekcija se može navesti unutar osnovnog konfiguracionog fajla, virtualnog hosta, direktorijuma ili **.htaccess** fajla. Ukoliko je potrebno koristiti regularne izraze za specificiranje imena fajlova treba koristiti **FilesMatch** sekciju.

```
<Files private.html>
...
</Files>
```

```
<FilesMatch "\.(gif|jpe?g|png)$">
...
</FilesMatch>
```

Sekcija **Directory** somogućava da se navedu direktive specifične samo za određeni direktorijum i njegove poddirektorijume na fajl sistemu. Za specificiranje imena direktorijuma mogu se koristiti džoker karakter '*' koji predstavlja više karaktera različitih od karaktera '/', karakter '?' koji zamenjuje bilo koji znak osim kose crte i karakteri '[' i ']' kojima se navodi opseg karaktera koji dolaze u obzir. Ukoliko je potrebno koristiti regularne izraze za specificiranje direktorijuma za to treba koristiti **DirectoryMatch** sekciju.

```
<Files private.html>
...
</Files>
```

```
<FilesMatch "\.(gif|jpe?g|png)$">
...
</FilesMatch>
```

Sekcija **Location** omogućava navođenje direktiva specifičnih samo za određene URL-ove kojim klijent pristupa. Mogu se koristiti džoker znaci '?' (jedan karakter) i '*' (više karaktera).

```
<Location /status>
...
</Location>
```

```
<LocationMatch "/(me|you)/data">
...
</LocationMatch>
```

Allow i **Deny** su direktive dostupne ukoliko je učitan modul **mod_access** tj. **mod_authz_host** u verziji 2.2, što je podrazumevana situacija u Ubuntu-u. Koriste se da bi se određenim hostovima dozvolio (**allow**) ili zabranio (**deny**) pristup nad određenim fajlovima, direktorijumima ili lokacijama (pogledaj direktive iznad). Hostove je moguće specificirati po IP adresi, imenu hosta, imenu domena ili mreži kojoj pripada (u network/netmask ili CIDR notaciji). Moguće je navesti više hostova u jednoj liniji ili navesti ključnu reč 'all' za sve

hostove.

```
Allow from tux.rs .example.com 172.16.0.0/255.240.0.0
Deny from 192.168.1.104 192.168.2. 192.168.3.0/24
```

Obzirom da se direktive odnose i na poddirektorijume, treba voditi računa da se za root direktorijum ('/') postave restriktivne dozvole ("Deny from all"), a zatim za sve pojedinačne direktorijume kojima treba da se pristupa eksplisitno dozvoliti pristup. Redosled Allow i Deny direktiva u fajlu nije važan; važan je parametar Order direktive.

Direktiva Order kontroliše rad Allow i Deny direktiva kada se zajedno koriste. Sledi primeri:

```
Order Deny,Allow
Deny from all
Allow from example.com
```

1. Prvo se obrađuje Deny direktiva pa onda Allow
2. Pristup je zabranjen svima, osim hostovima iz mreže example.com.

```
Order Deny,Allow
Allow from all
Deny from bla.atc.rs
```

1. Prvo se obrađuje Deny direktiva, pa zatim Allow
2. Pristup je dozvoljen svima (redosled Allow i Deny direktiva unutar sekcije nije važan)

```
Order Allow,Deny
Allow from atc.rs
Deny from srv.atc.rs
```

1. Prvo se obrađuje Allow direktiva, pa zatim Deny.
2. Pristup je dozvoljen svima iz mreže atc.rs, osim mašini srv.atc.rs. Hostovima van mreže ATC nije dozvoljen pristup.

DirectoryIndex je direktiva koju pruža modul `mod_dir`. Ova direktiva kontroliše koji će fajl biti automatski prikazan kada se kao URL navede direktorijum, a ne određeni fajl. Direktiva se može navesti unutar konfiguracije servera ili podešavanja pojedinih direktorijuma. Podrazumevana vrednost je `index.html`.

```
DirectoryIndex index.html index.cgi index.pl index.php index.xhtml
```

Options direktiva kontroliše mogućnosti koje su dostupne u pojedinim direktorijumima. Podrazumevana vrednost je 'All', a može se postaviti na 'None' ili neku kombinaciju vrednosti:

Indexes označava da će modul 'mod_autoindex' generisati listing fajlova unutar tog direktorijuma ukoliko nema fajla specificiranog direktivom `DirectoryIndex`.

ExecCGI dozvoljeno je izvršavanje CGI skripti ukoliko je učitan modul `mod_cgi`

FollowSymLinks server će pratiti simboličke linkove ako postoje u ovom direktorijumu. Dozvole nad lokacijom na koji link pokazuje se nasleđuju od trenutnog direktorijuma.

SymLinksIfOwnerMatch server će pratiti simboličke linkove ako postoje u ovom direktorijumu ukoliko je vlasnik linka takođe i vlasnik fajla ili direktorijuma na koji link pokazuje. Dozvole nad lokacijom na koji link pokazuje se nasleđuju.

Opcije se prenose i na poddirektorijume, osim ukoliko se i u njima specificiraju opcije. Moguće je i kombinovanje pojedinih opcija jednog poddirektorijuma sa opcijama nadređenih direktorijuma korišćenjem znaka '+' ili '-' ispred imena opcije. Primer:

```
<Directory /srv/www>
    Options Indexes FollowSymLinks
</Directory>

<Directory /srv/www/code>
    Options Includes
</Directory>
```

Samo je opcija 'Includes' postavljena za direktorijum /srv/www/code

```
<Directory /srv/www>
    Options Indexes FollowSymLinks
</Directory>

<Directory /srv/www/code>
    Options +Includes --Indexes
</Directory>
```

Za direktorijum /srv/www/code su postavljene opcije 'FollowSymLinks' i 'Includes'.

Alias je direktiva dostupna ukoliko je učitan modul `mod_alias`. Ona pruža mogućnost da se pristupa fajlovima unutar direktorijuma koji je izvan 'DocumentRoot' direktive. I za taj direktorijum je potrebno specificirati dozvole.

```
Alias /student /home/student
<Directory /home/student>
    Order allow,deny
    Allow from all
</Directory>
```

UserDir direktiva je dostupna ukoliko je učitan modul `userdir_module`. Direktiva pruža mogućnost da sistemski korisnici sami mogu unutar navedenog direktorijuma unutar svog home direktorijuma postaviti svoj web sadržaj kome se može pristupati preko URL-a `http://server/~username`

```
UserDir public_html
UserDir disabled root
<Directory /home/*public_html>
    Order allow,deny
    Allow from all
</Directory>
```

Direktiva `ErrorDocument` naznačava fajl ili poruku koji se prikazuju u slučaju greške.

```
ErrorDocument 404 /fajl_ne_postoji.html
ErrorDocument 503 http://backupserver/preopterecenje.html
```

1.2.5 Dodatni Moduli

Moduli omogućavaju jednostavnu proširivost Apache HTTP servera dodatnim mogućnostima i funkcionalnostima. Učitavaju se direktivom `LoadModule` kao u sledećem pri-

meru:

```
LoadModule mod_mime_magic /usr/lib/apache2/modules/mod_mime_magic.so
```

U konfiguracionom fajlu se pojedine direkutive mogu bezbedno koristiti samo kada je odgovarajući modul učitan i to korišćenjem `IfModule` sekcije, kao u primeru:

```
<IfModule mod_mime_magic.c>
    MIMEMagicFile /etc/apache2/magic
</IfModule>
```

Neki moduli mogu biti ukompajlirani u Apache HTTP server, tako da su stalno prisutni i nije ih potrebno ručno učitavati. Spisak takvih modula se može videti komandom:

```
# apache2 -l
```

1.2.6 Logovi

Praćenje aktivnosti Apache HTTP servera se lako može vršiti usled log fajlova koje zapisuje. `ErrorLog` direktiva kontroliše ime fajla u koji se zapisuju sve važne poruke o greškama i druge dijagnostičke informacije. Primer jednog zapisa ovog fajla je:

```
[Sun Mar 11 14:32:52 0100] [error] [client 127.0.0.1] client denied by server
configuration: /srv/www/index.html
```

Ovim je u log fajlu zapisana informacija o tome da je u subotu 11. marta u 14 sati 32 minuta i 52 sekunde u vremenskoj zoni GMT+1 klijent sa IP adresom 127.0.0.1 pokušao da pristupi fajlu /srv/www/index.html na serveru, ali mu to nije dozvoljeno na osnovu konfiguracije servera. To znači da je u korisnik dobio statusnu poruku 403, što se može videti i u drugom log fajlu koji server piše – log fajlu pristupa.

Zapis o pristupima se zapisuju u fajlu specificiranom `CustomLog` direktivom, a se može podešiti direktivom `LogFormat`. Tipičan format (CLF - *Common Log Format*) je sledeći:

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access_log common
```

Ovakav format označava da se najpre beleži IP adresa klijenta (%h), zatim identifikacija (%l i %u) ili minusi ukoliko korisnik nije identifikovan, nakon toga datum i vreme (%l), pa HTTP zahtev koji je posao klijent (%r) koji je okružen navodnicima i zatim statusni kod (%>s) i veličina u bajtovima objekta bez zaglavlja koji je poslat klijentu (%b).

Linija `access_log` fajla koja odgovara gornjem primeru u `error_log` fajlu po gore specificiranom 'common' formatu je:

```
127.0.0.1 - - [11/Mar/2007:14:32:52 0100] "GET /index.html HTTP/1.0" 403 -
```

Osim tipičnog CLF formata log fajla pristupa, često se koristi još i kombinovan format, gde se uključuju i dodatne informacije iz zaglavlja HTTP zahteva, npr. "Referer"(odakle je klijent došao na tu stranicu) ili "User-agent"(softver koji klijeknt koristi). Sva HTTP zaglavlja mogu se zabeležiti u log fajl korišćenjem `LogFormat` direktive i forme `%{ime_zaglavlja}`i.

Osim u fajlove, logovi se mogu slati i preko pajpova bilo kom programu na sistem. Ovim se omogućava jednostavno obezbeđivanje rotiranja logova ili logovanje u syslog, kao u primeru:

```
CustomLog "| /usr/bin/logger -p daemon.info" common
```

1.2.7 .htaccess fajlovi

Apache HTTP server ima jednostavan način za kontrolisano omogućavanje promene podešavanja specifičnih samo za određene direktorijume – korišćenje modula `mod_auth`. Na ovaj način korisnici sami mogu podesiti prava pristupa pojedinim direktorijumima unutar svog home direktorijuma, podesiti autentifikaciju korisnika i drugo. Dovoljno je da u direktorijumu za koji žele da promene podešavanja kreiraju fajl sa imenom specificiranim direktivom `AccessFileName` (obično `.htaccess`) i u njemu upišu podešavanja koja žele za taj direktorijum i njegove poddirektorijume. Naravno, ovo mora biti dozvoljeno u serverskoj konfiguraciji, i to se radi direktivom `AllowOverride` kojom se za određeni direktorijum specificiraju grupe direktiva koje se mogu promeniti u `.htaccess` fajlu (`AuthConfig`, `FileInfo`, `Indexes`, `Limit`, `Options`, `All`, `None`)

Korišćenje `.htaccess` fajlova je ekvivalentno stavljanju direktiva iz tog fajla unutar `<Directory>` direktive za direktorijum u kom se `.htaccess` fajl nalazi, osim što se `.htaccess` fajl učitava i ispituje prilikom svakog pristupa dokumentima u tom direktorijumu, dok se direktive iz glavnog konfiguracionog fajla učitavaju samo prilikom startovanja servera.

1.2.8 Osnovna HTTP autentifikacija

Apache HTTP server pruža mogućnost autentifikovanja korisnika prilikom pristupanja nekom dokumentu. Osnovna autentifikacija pruža mogućnost da se definišu korisnici koji smeju da pristupe određenim dokumentima na serveru ukoliko otkucaju odgovarajuću šifru. Treba voditi računa da se korisničko ime i šifra kroz mrežu do servera šalju u čistom tekstu, tako da se ne treba osloniti samo na ovaj metod ukoliko su u pitanju osetljivi podaci (pogledaj odeljak 1.3).

Autentifikacija se podešava u glavnom konfiguracionom fajlu (unutar `<Directory>`, `<Files>` ili `<Location>`) ili u `.htaccess` fajlu za određeni direktorijum. Najpre treba kreirati fajl u kome će biti sačuvana korisnička imena i šifre. Ukoliko se ovaj fajl ne može kreirati na lokaciji koja nije dostupna preko web-a (npr. zbog prava pristupa) onda je neophodno zaštititi taj fajl `Deny from all` direktivom. Fajl se kreira komandom `htpasswd`:

```
# htpasswd -c /srv/passwords/htpasswd student
```

Parametar `-c` kreira novi fajl, pa ga treba izostaviti ako dodajemo korisnika u postojeći fajl. Direktive koje obezbeđuju autentifikovanje korisnika putem ovog fajla su:

```
AuthType Basic  
AuthName "Neophodna identifikacija"  
AuthUserFile /srv/passwords/htpasswd  
Require valid-user
```

1.2.9 Virtuelni hostovi

Virtuelni hostovi su mogućnost web servera da služi veći broj sajtova na jednoj mašini. Postoje dva načina pružanja virtualnog hostinga: *IP-bazirani virtualni hosting* gde svaki sajt ima zasebnu IP adresu i *Host-bazirani* (ili *Name-bazirani*) *virtualni hosting* gde se u DNS-u podešava da više imena sajtova pokazuju na jedan server koji na osnovu imena koje je klijent koristio da bi došao do servera odlučuje koji će sadržaj prikazati.

IP-bazirani virtualni hosting je pogodan kod servera na kojima je neophodno izvršiti razdvajanje pristupa dokumentima različitih virtualnih hostova na sistemskom nivou, po kretanjem više instanci Apache HTTP servera koji slušaju na različitim IP adresama i koji se izvršavaju kao različiti sistemski korisnici. Ovaj princip je jako zahtevan po pitanju resursa i zahteva da se za svaki sajt odvoji zasebna IP adresa. Pošto to često nije moguće ostvariti, mnogo više se koristi Host-bazirani virtualni hosting.

Podešavanje Host-baziranog virtualnog hostinga se vrši dodavanjem u konfiguracioni fajl servera direktive `NameVirtualHost` kojom se naznačava IP adresa i port na koje se odnose virtualni hostovi i definisanjem virtualnih hostova unutar za koje se mogu navesti skoro sve direktive kojima se podešava rad servera. Sledeći primer pokazuje konfiguraciju servera koji služi dva virtualna hosta: `primer1.atc.rs` i `primer2.atc.rs` čiji se sadržaji prikazuju iz odgovarajućih direktorijuma:

```
NameVirtualHost *:80

<VirtualHost *:80>
    ServerName primer1.atc.rs
    ServerAlias www.primer1.atc.rs *.primer1.atc.rs
    DocumentRoot /var/www/html/primer1.atc.rs
    ErrorLog /var/log/httpd/primer1.atc.rs-error.log
    CustomLog /var/log/httpd/primer1.atc.rs-access.log common
</VirtualHost>

<VirtualHost *:80>
    ServerName primer2.atc.rs
    DocumentRoot /var/www/html/primer2.atc.rs
    ErrorLog /var/log/httpd/primer2.atc.rs-error.log
    CustomLog /var/log/httpd/primer2.atc.rs-access.log common
</VirtualHost>
```

1.3 SSL/TLS Enkripcija

SSL/TLS enkripcija HTTP saobraćaja omogućava bezbedan način za prenos osetljivih podataka preko weba. Apache HTTP server se lako može podesiti da radi kriptovanje sadržaja.

1.3.1 SSL sertifikati

Potrebno je najpre, pribaviti SSL sertifikat koji će server koristiti. Svaki SSL sertifikat mora biti potpisani drugim sertifikatom. Time se kreira lanac sertifikata koji počinje osnovnim (root) sertifikatom, a završava se sertifikatom servera. Sami sertifikati se potpisuju tajnim ključem koji odgovara javnom ključu smeštenom u samom sertifikatu kojim se potpisuje. Na taj način strana koja proverava serverski sertifikat može utvrditi njegovu ispravnost ukoliko ima pristup svim sertifikatima koji su se koristili kao potpisi u lancu sertifikata.

Sertifikati koji služe za potpisivanje drugih sertifikata a koji sami nisu osnovni sertifikat nazivaju se posrednički (intermediate) sertifikati. Server može biti konfigurisan da umesto samo sopstvenog SSL sertifikata, šalje deo lanca koji sačinjavaju serverski SSL sertifikat i deo posredničkih sertifikata koji zajedno čine kontinualan deo lanca.

Root SSL sertifikat može kreirati bilo ko i koristiti ga za potpisivanje svojih sertifikata. Ukoliko server šalje i root sertifikat u sklopu lanca, onda se takav sertifikat naziva samopotpisani sertifikat. Pošto veb klijent ne poseduje niti osnovni sertifikat korišćen kod samopotpisanih sertifikata niti deo lanca, on će obično smatrati takav sertifikat sumnjivim i upozoriće korisnika na to. U tim slučajevima korisnik mora samostalno prihvati takav sertifikat.

Pošto SSL sertifikat može poslužiti i kao metod autentifikacije, a Apache server može biti konfigurisan ne samo da šalje svoj sertifikat klijentu već i da od njega traži klijentski sertifikat, to se često takav scenario koristi onamo gde treba ograničiti pristup samo autorizovanim korisnicima. U takvim slučajevima najčešće sam entitet (firma ili grupa koja upravlja serverom i autorizuje korisnike za pristup tom serveru) kreira svoj root SSL sertifikat (obično uz jedan ili više posredničkih sertifikata) kojim potpisuje i serverski i klijentske sertifikate. Ovo je takođe čest slučaj ukoliko takav entitet kontroliše više različitih veb servera od kojih

svaki ima svoj SSL sertifikat. U veb klijente koji pristupaju takvim serverima unapred mora biti instaliran root sertifikat (i odgovarajući posrednički sertifikati) jer će u protivnim veb klijent alarmirati korisnika da pristupa sajtu čiju autentičnost klijent ne može da proveri.

Ukoliko sajtu treba da pristupaju različiti korisnici, posebno ako je u pitanju sajt preko kojeg se obavlja elektronska trgovina, najbolje je SSL sertifikat pribaviti od kuća koje prodaju takve sertifikate. Ove kuće se nazivaju *Certification Authorities* (CA) i većinom posluju po komercijalnom principu. Te kuće takođe sklapaju ugovore sa autorima veb klijenata da njihovi root i posrednički sertifikati unapred budu instalirani. Veb klijenti neće u takvim slučajevima prikazivati nikakvo upozorenje već će obično označiti takav sajt kao bezbedan.

Svi SSL sertifikati su vremenski ograničeni i datum i vreme prestanka važenja sertifikata je zapisano u samom sertifikatu. Takođe, moguće je da onaj entitet koji je potpisao sertifikat pre vremena proglaši dati sertifikat nevažećim. Iz tog razloga sertifikaciona tela, bilo da su u pitanju CA ili entiteti koji samostalno izdaju sertifikate, moraju obezbediti način da se svakom sertifikatu proveri validnost. Ovo se obavlja ili putem objavljivanja tzv. CRL (*Certificate Revocation List*), spiska povučenih sertifikata ili korišćenjem OCSP (*Online Certificate Status Protocol*) protokola, gde sertifikaciono telo obezbeđuje servis preko kojeg može da se proveri validnost sertifikata. Ove informacije (URL CR liste ili OCSP servisa) se čuvaju u samom sertifikatu.

1.3.2 Konfigurisanje Apache servera za SSL/TLS

Apache server implementira SSL/TLS protokol preko eksternog modula `mod_ssl`, koji mora biti instaliran.

Da bi server slušao na portu 443 (HTTPS – HTTP over SSL) treba dodati odgovarajuću `Listen` direktivu u konfiguracioni fajl. Parametri koji kontrolišu ponašanje `mod_ssl`-a se obično podešavaju u zasebnom fajlu, gde se definišu i podrazumevane vrednosti za tzv. glavni sajt. Podrazumevane vrednosti za glavni sajt se mogu izmeniti u konfiguracijama za virtuelne hostove. Primer:

```
NameVirtualHost *:443

<Virtualhost *:443>
ServerAdmin webmaster@localhost

SSLEngine On
SSLCertificateFile /etc/pki/tls/certs/server.crt
SSLCertificateKey /etc/pki/tls/private/server.key

DocumentRoot /var/www-secure

<Directory />
Options FollowSymLinks
AllowOverride None
</Directory>

<Directory /var/www-secure/>
Options Indexes FollowSymLinks MultiViews
AllowOverride None
Order allow,deny
Allow from all
</Directory>
```

Poslednji korak je restartovanje apache-a.

Često je potrebno forsirati korišćenje SSL-a za određenu lokaciju na sajtu, za šta je najbolje koristiti `mod_rewrite` modul.

1.4 PHP

PHP je skripting jezik jako popularan za različite tipove web aplikacija. PHP se može krenuti na više načina:

1. kao CGI skript (php kao komandni interfejs)
2. kao FCGI skript (Fast CGI) (preko odgovarajućeg servera php-fpm)
3. kao eksterni modul Apache servera (mod_php)

Konfiguracioni fajl PHP-a se zove `php.ini` i obično se nalazi u direktorijumu `/etc` ili `/etc/php`. U njemu je potrebno otkomentarisati ili promeniti vrednosti kojima se kontrolisu mogućnosti i ponašanje PHP-a, pošto je i sam PHP modularan. Npr. nakon instaliranja `php5-mysql` paketa koji daje podršku za pristupanje MySQL bazi iz PHP skripti, potrebno je otkomentarisati liniju `extension=mysql.so`. Druga važna vrednost koju treba promeniti je `register_globals`, koju zbog sigurnosti treba postaviti na 'Off'. Još jedan primer šta se može podešiti u `php.ini` fajlu je promenljiva `memory_limit` koju je potrebno povećati ukoliko je PHP aplikacijama potrebno više memorije.

1.5 CGI

CGI je skraćeno od *Common Gateway Interface* i predstavlja tradicionalan način za obezbeđivanje dinamičkih sadržaja. Apache HTTP server u ovom slučaju služi samo kao posrednik između korisnika i nekog programa instaliranog na sistemu. Kada korisnik zatraži određenu stranicu, Apache HTTP server izvršava odgovarajući program i standardni izlaz tog programa prosleđuje korisniku.

CGI se na Ubuntu distribuciji instalira sa paketom `apache2-common`, a omogućava se komandom `a2enmod cgi`. Specificiranje u kojim se direktorijumima mogu nalaziti CGI programi se može obaviti na dva načina:

1. Korišćenjem `ScriptAlias` direkture kojom se svi fajlovi u direktorijumu koji je izvan `DocumentRoot`-a smatraju CGI programima:
2. Korišćenjem `Options +ExecCGI` i `AddHandler cgi-script .cgi` direktive kojima se omogućava izvršavanje programa sa `.cgi` ekstenzijom unutar direktorijuma u čijem su opsegu.

Property of Admin Training Center

Modul 2: MySQL DBMS

2.1 Uvod

MySQL je multitredovani, multikorisnički SQL DBMS (*database management system*) koji je jako popularan za upotrebu u web aplikacijama. Predstavlja database komponentu LAMP, MAMP i WAMP sistema (Linux/Mac/Windows – Apache – MySQL – PHP/Perl/Python). Biblioteke za pristup MySQL bazama postoje u svim glavnim programskim jezicima, a postoji i ODBC interfejs.

Korisničke aplikacije za pristup MySQL bazama postoje za Linux, Windows i MacOS, a postoji i jednostavna PHP aplikacija phpMyAdmin za uređivanje baze preko web-a.

MySQL je vlasništvo kompanije Oracle koja pruža komercijalnu podršku. Licenciran je dualnom licencom – korisnici koji žele, mogu koristiti MySQL pod GNU GPL licencem, uz izuzetak da softver licenciran pod nekom drugom OSI licencem može da bude linkovan sa MySQL klijentskim bibliotekama. Korisnici kojima ovakva licenca ne odgovara (npr. projektima zatvorenog koda koji imaju potrebu da uključe MySQL) mogu od MySQL AB kupiti MySQL pod vlasničkom komercijalnom licencem.

2.2 Instalacija i inicijalizacija

2.2.1 Binarni paketi

MySQL se isporučuje kao standardna baza na RHEL/CentOS verzijama 5 i 6. Verzija 7 je zamenila MySQL sa MariaSQL DBMS-om – open source forkom MySQL baze koju razvijaju originalni autori MySQL DBMS. MariaSQL je binarno kompatibilna sa MySQL DBMS.

2.2.2 Instalacija u različitim distribucijama

RHEL/CentOS/Fedora

Instalacija se može obaviti tokom instalacije servera ili naknadno komandom

```
# yum -y install mysql mysql-server
```

MySQL se startuje komandom:

```
# service mysqld start
```

Automatsko startovanje servera se omogućava komandom:

```
# chkconfig mysqld on
```

Konfiguracijski fajl je `/etc/my.cnf`, a baze se snimaju u `/var/lib/mysql` direktorijumu. Root nalog za bazu nije inicijalno zaštićen šifrom.

Po prvom pokretanju servisa, zgodno je startovati interaktivnu komandu:

```
# mysql_secure_install
```

Ovom komandom se postavlja root lozinka, uklanja mogućnost logovanja root korsinika sa udaljene mašine i uklanja temp baza čime se ojačava bezbednost servisa.

Debian i Ubuntu

MySQL se nalazi u `main` repozitorijumu i može se instalirati komandom:

```
# apt-get install mysql-server
```

MySQL se startuje komandom

```
# service mysql start
```

Konfiguracioni fajl je `/etc/mysql/my.cnf`, a baze se instaliraju u `/var/lib/mysql` direktorijumu. Root nalog za bazu nije inicijalno zaštićen šifrom.

2.2.3 Konfiguracija

Sve direkutive i konfiguracioni parametri se MySQL-u mogu proslediti kroz komandnu liniju u obliku `-direktiva ili -parametar=vrednost`. Konfiguracioni fajl MySQL-a se sastoji od sekcija kojima se podešava određeni deo MySQL sistema. Unutar sekcija se navode direkutive ili parametri i njihove vrednosti. Osnovni parametri koji kontrolisu rad servera su u `[mysqld]` sekciji:

```
user = mysql # sistemski korisnik
pid-file = /var/run/mysqld/mysqld.pid # lokacija PID fajla
socket = /var/run/mysqld/mysqld.sock # lokacija socket-a
port = 3306 # port na kom slusa
bind-address = 127.0.0.1 # adresa na kojoj slusa
basedir = /usr # gde je instaliran
datadir = /var/lib/mysql # gde se nalaze baze
tmpdir = /tmp # privremene tabele
language = /usr/share/mysql/english # jezik poruka o greskama
```

2.2.4 Inicijalizacija baza

Nakon instalacije MySQL-a i kada je potrebno kreirati baze od početka, potrebno je izvršiti inicijalizaciju osnovne mysql baze. To se kod većine distribucija obavlja automatski nakon instalacije.

Da bi se ručno obavilo inicijalizovanje baza, najpre treba spustiti MySQL server, zatim izbrisati `/var/lib/mysql` direktorijum i onda kreirati osnovne baze pokretanjem komande:

```
# mysql_install_db
```

Ova komanda se pokreće pod privilegijama root korisnika, pa je potrebno promeniti vlasništvo nad kreiranim fajlovima i direktorijumima tako da pripadaju korisniku `mysql` (parametar `user` iz `[mysqld]` sekcije `my.cnf` fajla):

```
# chown -R mysql:mysql /var/lib/mysql
```

Na kraju treba startovati mysql server i postaviti šifru za root korisnika MySQL servera:

```
# mysqld_safe --user=mysql &
# mysqladmin -u root password '<nova lozinka>'
```

U slučaju gubljenja šifre root korisnika MySQL-a, potrebno je privremeno dodati parametar `skip-grant-tables` u [mysqld] sekciju my.cnf fajla, zatim restartovati MySQL i povezati se kao root korisnik na bazu, pa postaviti novu šifru promenom polja `password` u `mysql.user` tabeli za root korisnika:

```
# mysql -u root
...
mysql> use mysql
mysql> update user set password=password('novasifra') where User='root';
```

Kod Ubuntu i Debian distribucija za inicijalizaciju baze potrebno je izvršiti komandu:

```
$ sudo dpkg-reconfigure mysql-server-5.0
```

Ova rekonfiguracija će kreirati osnovne baze, postaviti password root korisnika MySQL-a i kreirati "debian-sys-maint" korisnika koji služi za automatsko startovanje MySQL servera kroz inicijalizacione skripte.

Nalog "debian-sys-maint" se može koristiti i u slučaju gubljenja šifre root korisnika: potrebno je prikačiti se na bazu kao ovaj korisnik, koristeći šifru koja se nalazi u fajlu /etc/mysql/debian.cnf i promeniti šifru korisniku root:

```
# mysql -u debian-sys-maint -p
mysql$>$ set password for 'root'@'localhost'=password("sifra");
```

2.3 Rad sa MySQL bazama

2.3.1 Kreiranje baze

Pomoću mysqladmin alatke:

```
# mysqladmin -u root -p create primer1
```

Iz mysql klijenta:

```
mysql> create database primer2;
```

2.3.2 Dropovanje baze

Pomoću mysqladmin alatke:

```
# mysqladmin -u root drop primer1
```

Iz mysql klijenta:

```
mysql> drop database primer2;
```

2.3.3 Pravljenje tabele unutar baze

Iz mysql klijenta:

```
mysql> use primer1
mysql> CREATE TABLE Imenik ( Id VARCHAR(5), Ime VARCHAR(50), Telefon CHAR(11) );
```

2.3.4 Unošenje podataka u tabelu

Iz mysql klijenta:

```
mysql> INSERT INTO Imenik (Id, Ime, Telefon) VALUES (1, 'Pera Peric', '011/123-456-7');
```

2.3.5 Dobijanje specifičnih podataka iz tabele

Iz mysql klijenta:

```
mysql> SELECT Ime, Telefon FROM Imenik WHERE Id = 1;
```

2.3.6 Dropovanje tabela unutar baze

Iz mysql klijenta:

```
mysql> DROP TABLE Imenik;
```

2.3.7 Dodavanje korisnika i podešavanje prava

Za kreiranje korisnika "student" koji ima sve privilegije MySQL-u kada se povezuje sa lokalne mašine i kad unese šifru "šifra1" potrebna je sledeća grant naredba:

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'student'@'localhost'  
IDENTIFIED BY 'šifra1' WITH GRANT OPTION;
```

Za oduzimanje pojedinih privilegija koristi se naredba:

```
mysql> REVOKE SELECT ON baza.tabla FROM 'student'@'localhost';
```

Za brisanje korisnika koristi se naredba:

```
mysql> DROP USER 'student'@'localhost';
```

2.4 Bekap i restore podataka

Backup baze se može izvesti na dva načina:

1. Kreiranjem SQL dump fajla kojim se restore vrši mysql komandom

```
# mysqldump --opt db_name $>$ bekap.sql
```

2. Kopiranjem samih fajlova baze (samo MyISAM), kada se restore takođe vraća kopiranjem:

```
# mysqlhotcopy db_name /putanja/do/nekog/direktorijuma
```

2.5 Pregled programa MySQL distribucije

mysql Program za interaktivno unošenje SQL upita kroz komandnu liniju ili izvršavanje upita iz fajla.

myisamchk Vrši proveru, optimizovanje, popravljanje i opisivanje MyISAM tabela

mysqlaccess Skripta koja vrši proveru pristupa na osnovu imena baze, korisničkog imena i adrese hosta sa kog se korisnik kači na bazu

mysqladmin Administrativni klijent kojim se može obaviti kreiranje i dropovanje baza, ponovno učitavanje grant tabela, zapisivanje tabela na disk i ponovno otvaranje log fajlova.

mysqlbinlog Alatka za čitanje i procesiranje binarnih logova.

mysqlcheck Klijent za operacije nad tabelama kao što su provera, analiza, popravka i optimizacija

mysqldump Klijent za eksportovanje MySQL baze u SQL fajl ili XML.

mysqlhotcopy Klijent za brzo kreiranje online bekapa MyISAM tabele

mysqlimport Klijent koji uvozi tekstualne fajlove u tabele.

mysqlshow Klijent koji prikazuje informacije o bazama, tabelama, kolonama i indeksima

mysql_fix_extensions Alatka koja prebacuje slova iz ekstenzije fajlova u kojima se čuvaju tabele u mala slova.

mysql_setpermission Program za interaktivno postavljanje dozvola nad bazama i tabelama.

mysql_tableinfo Alatka za generisanje opisnih informacija o bazi.

mysql_waitpid Alatka koja gasi proces određenog PID-a

perror Alatka koja prikazuje shvatljive poruke umesto numeričkih kodova grešaka.

2.6 Pregled MySQL naredbi

2.6.1 Naredbe za opisivanje podataka (Data Definition Statements)

ALTER DATABASE menja opšte karakteristike baze kao što su CHARACTER SET i COLLATE. Neophodna je ALTER privilegija.

ALTER TABLE menja strukturu tabele u cilju dodavanja novih kolona, promene imena ili brisanja postojećih, kreiranje i brisanje indeksa.

CREATE DATABASE kreira bazu. Neophodna je CREATE privilegija.

CREATE TABLE kreira tabelu navedene strukture. Neophodna je CREATE privilegija.

```
| CREATE TABLE test (a INT NOT NULL AUTO_INCREMENT, PRIMARY KEY (a), b INT);  
| CREATE TABLE new_tbl SELECT * FROM orig_tbl;
```

DROP DATABASE uklanja navedenu bazu i sve tabele u njoj. Neophodna je DROP privilegija.

DROP TABLE uklanja jednu ili više tabele. Neophodno je imati DROP privilegiju na svakoj od tabele koje se brišu.

2.6.2 Naredbe za upravljanje podacima (Data Manipulation Statements)

DELETE briše redove specificirane opcionom WHERE klauzulom i vraća broj obrisanih redova. Redosled brisanja se može navesti klauzulom ORDER BY, a broj redova za brisanje klauzulom LIMIT.

```
| DELETE FROM tbl_name WHERE 1>$0;
```

INSERT unosi nove redove u postojeću tabelu.

```
| INSERT INTO tbl_name (col1,col2) VALUES(15,col1*2);  
| INSERT INTO tbl_name set col1=15, col2=col1*2;  
| INSERT INTO tbl_temp2 (fld_id)  
|     SELECT tbl_temp1.fld_order_id  
|     FROM tbl_temp1
```

```
WHERE tbl_temp1.fld_order_id > 100;
```

LOAD DATA INFILE čita vrednosti za unos u tabelu iz fajla.

REPLACE funkcioniše kao INSERT naredba, osim kada se unosi red sa postojećom vrednošću primarnog ključa (PRIMARY KEY) ili jedinstvenog indeksa (UNIQUE index), u kom slučaju se pretnodni red briše pre insertovanja novog.

SELECT prikazuje redove iz jedne ili više tabele.

```
SELECT college, region, seed
      FROM tournament
     ORDER BY region, seed;

SELECT CONCAT(last_name, ', ', first_name) AS full_name
      FROM mytable
     ORDER BY full_name;

SELECT *
      FROM t1
     WHERE column1 = (SELECT column1 FROM t2);
```

TRUNCATE TABLE briše sve zapise iz tabele.

UPDATE postavlja nove vrednosti kolona u redovima tabele koji se specificiraju sa opcionom WHERE klauzulom. Ukoliko se koristi ORDER BY klauzula, redovima se postavljaju vrednosti po tom redosledu. Ako se koristi LIMIT klauzula, onda se postavljaju nove vrednosti za najviše toliki broj redova.

```
UPDATE persondata SET age=age+1;
UPDATE items,month SET items.price=month.price WHERE items.id=month.id;
```

DESCRIBE pruža informacije o kolonama tabele. Predstavlja sinonim za SHOW COLUMNS FROM.

USE db_name - postavlja db_name kao aktivnu bazu za naredbe koje slede.

2.6.3 Naredbe za upravljanje nalozima (Account Management Statements)

DROP USER briše korisnika. Za korišćenje ove naredbe neophodna je DELETE privilegija na bazi mysql. Nalozi se navode u obliku 'nalog'@'adresa'.

GRANT dodeljuje privilegije korisnicima. Za korišćenje ove naredbe neophodna je GRANT OPTION privilegija. Ne mogu se dodeliti privilegije koje i sami nemate.

```
GRANT ALL ON test.* TO ''@'localhost' IDENTIFIED BY 'mypass';
```

REVOKE oduzima privilegije od korisnika. Za korišćenje ove naredbe neophodna je GRANT OPTION privilegija.

```
REVOKE ALL PRIVILEGES, GRANT OPTION FROM user [, user] ...
```

SET PASSWORD postavlja šifru za korisnički nalog

```
SET PASSWORD FOR 'bob'@'%loc.gov' = PASSWORD('newpass');
```

2.7 Pregled MySQL privilegija

ALL [PRIVILEGES] Sinonim za sve osnovne privilegije osim GRANT OPTION

ALTER Omogućava korišćenje ALTER TABLE naredbe

CREATE Omogućava korišćenje CREATE TABLE naredbe

CREATE TEMPORARY TABLES Omogućava korišćenje naredbe CREATE TEMPORARY TABLE

DELETE Omogućava korišćenje DELETE naredbe

DROP Omogućava korišćenje DROP TABLE naredbe

FILE Omogućava korišćenje naredbe SELECT u obliku INTO OUTFILE i naredbe LOAD DATA INFILE

GRANT OPTION Omogućava dodeljivanje privilegija drugima

INDEX Omogućava korišćenje naredbi CREATE INDEX i DROP INDEX

INSERT Omogućava korišćenje INSERT naredbe

LOCK TABLES Omogućava korišćenje LOCK TABLES naredbe na tabelama za koje je dozvoljeno izvršavati naredbu SELECT

PROCESS Omogućava korišćenje SHOW FULL PROCESSLIST naredbe

RELOAD Omogućava korišćenje FLUSH naredbe

REPLICATION CLIENT Omogućava dobijanje informacija o master i slejv serverima

REPLICATION SLAVE Dozvoljava korisniku da čita binarne logove za replikaciju

SELECT Omogućava korišćenje SELECT naredbe

SHOW DATABASES Prikazuje sve baze

SHUTDOWN Omogućava korišćenje mysqladmin alatke za gašenje servera

SUPER Omogućava korišćenje CHANGE MASTER, KILL, PURGE MASTER LOGS i SET GLOBAL naredbi i omogućava povezivanje na server kada je dostignut max_connections, maksimalni broj konekcija

UPDATE Omogućava korišćenje naredbe UPDATE

USAGE Sinonim za "bez privilegija"

2.8 Primeri najčešćih administratorskih aktivnosti

Postavljanje lozinke za 'root' korisnika koji se loguje lokalno:

```
$ mysqladmin -u root password '$<$nova_lozinka$>'
```

Logovanje na MySQL server kao 'root' korisnik:

```
$ mysql -u root -p mysql
```

Bezbednosna provera lozinki za 'root' korisnika:

```
mysql$>$ SELECT Host, User, Password FROM user;
```

Postavljanje istovetnih lozinki za 'root' korisnika koji se loguje sa svih navedenih lokacija:

```
mysql$>$ UPDATE user SET Password=PASSWORD('$<$nova lozinka$>$') WHERE User='root';
```

Kreiranje baze 'bazal':

```
$ mysqladmin -u root -p create bazal
```

Kreiranje korisnika 'user1' i dodeljivanje svih privilegija nad bazom 'bazal':

```
mysql> GRANT USAGE ON *.* TO 'user1'@'localhost' IDENTIFIED BY '<user1 lozinka>';  
mysql> GRANT ALL PRIVILEGES ON bazal.* TO 'user1'@'localhost';
```

Logovanje na MySQL server kao 'user1' korisnik:

```
$ mysql -u user1 -p bazal
```

Kreiranje tabele 'tabela1':

```
mysql$>$ CREATE TABLE tabela1 (id INTEGER PRIMARY KEY, naziv VARCHAR(32) NOT NULL);
```

Popunjavanje tabele 'tabela1' sadržajem:

```
mysql$>$ INSERT INTO tabela1 VALUES (1, 'prvi red');  
mysql$>$ INSERT INTO tabela1 VALUES (2, 'drugi red');  
mysql$>$ INSERT INTO tabela1 VALUES (3, 'treci red');
```

Kreiranje punog backupa za bazu 'bazal':

```
$ mysqldump---opt bazal > bazal-backup.sql
```

Brisanje baze 'bazal':

```
$ mysqladmin -u root -p drop bazal
```

Restore baze 'bazal':

```
$ mysqladmin -u root -p create bazal
```

```
$ mysql -u user1 -p bazal $<$ bazal-backup.sql
```

Modul 3: Mail servis

Elektronska pošta predstavlja najznačajniji i najviše korišćeni servis interneta. Nije netačno reći da je Internet i razvijen kako bi se omogućila razmena elektronske pošte između udaljenih mašina. Nastao kao servis za kopiranje tekstualnih fajlova između korisnika na istoj mašini, danas predstavlja univerzalni način za prosleđivanje teksta i multimedijalnih sadržaja. Važno je primetiti da poruke danas gotovo nikad ne idu direktno od jednog korisnika do drugog, već putem servera za transport i isporuku poruka.

3.1 Komponente sistema elektronske pošte

Komponente sistema elektronske pošte se mogu definisati kao:

MUA Mail User Agent

MSA Mail Submission Agent

MTA Mail Transport Agent

MDA Mail Delivery Agent

Mail User Agent (MUA) je korisnički program za pisanje, slanje i primanje elektronske pošte. Korisnik piše poruku koristeći običan ili formatiran (npr. HTML-om) tekst i upućuje je na e-mail adresu primaoca. MUA ne mora da zna koji server je odgovoran za prijem pošte za e-mail adresu primaoca, niti da sam kontaktira taj server (koji možda trenutno nije dostupan), već poruku prosleđuje *Mail Submission Agentu* (MSA) svog Internet provajdera. MSA je zadužen za prihvatanje poruka od autorizovanih korisnika, eventualno dodavanje ili ispravljanje različitih zaglavljiva poruke (npr. polja, ili datum) i prosleđivanje poruke *Mail Transport Agentu* (MTA). MTA kontaktira server koji prima poštu za e-mail adresu primaoca, nakon što je utvrdio koji je to server. Na tom serveru se takođe izvršava MTA, ali ovde u ulozi prijemnog agenta za određene e-mail adrese. Kada poruku prihvati, prijemni MTA kontaktira *Mail Delivery Agentu* (MDA) koji poruku isporučuje u sanduče primaoca. Primalac koristi svoj MUA za pristup i preuzimanje poruka iz svog sandučeta.

Napomena:

MTA1 može kontaktirati MTA2 direktno ili preko svog nadređenog MTA Uloge MSA, MTA i MDA komponenti često može obavljati jedan isti softver.

3.2 Struktura elektronskih poruka

Format elektronskih poruka definisan je RFC-om 5322 za osnovni format poruke sa čistim tekstom i RFC-ovima 2045-2049 kojima se reguliše slanje UTF-8 enkodovanog teksta, HTML formatiranog teksta, slika, zvuka i drugog. Sve e-mail poruke se šalju korišćenjem čistog teksta, pri čemu se podaci enkoduju BASE64 ili nekim drugim algoritmom.

3.2.1 Koverta (envelope)

Pod kovertom poruke se podrazumevaju informacije o poreklu i serverima kroz koje je prošla poruka na putu od pošiljaoca do primaoca. Te informacije dodaju na početak poruke MSA-ovi, MTA-ovi i MDA-ovi, tako da se prilikom određivanja porekla poruke čita od pozadi. Takođe, u ovom delu poruke se nalazi i **Return-Path:** adresa (envelope sender) na koju se šalje obaveštenje ako dođe do problema sa isporukom poruke primaocu (npr. nepostojeći korisnik ili nedostupan server).

```
Delivered-To: uskokovic@gmail.com
Received: by 10.210.105.20 with SMTP id
d20mr34879116ebc.84.1231805039748; Mon, 12 Jan 2009 16:03:59 -0800 (PST)
Return-Path: <uskokovic@etf.rs>
Received: from smtp.etf.bg.ac.rs (SMTP.ETF.BG.AC.RS [147.91.8.62])
by mx.google.com with ESMTP id
f3si89816667nfh.74.2009.01.12.16.03.58; Mon, 12 Jan 2009 16:03:59 -0800 (PST)
Received: from kondor.etf.bg.ac.rs (kondor.ETF.BG.AC.RS [147.91.8.8])
by smtp.etf.bg.ac.yu (Postfix) with ESMTP id 83CB21BB27D for
<uskokovic@gmail.com>; Tue, 13 Jan 2009 01:03:54 +0100 (CET)
Received: from localhost (localhost [127.0.0.1]) by
kondor.etf.bg.ac.rs (Postfix) with ESMTP id C5B0FE2B70 for
<uskokovic@gmail.com>; Tue, 13 Jan 2009 01:03:54 +0100 (CET)
Received: from 91.148.91.47 (SquirrelMail authenticated user uskokovic)
by localhost with HTTP; Tue, 13 Jan 2009 01:03:54 +0100 (CET)
```

U navedenom primeru, korisnički program (MUA) je vebmail softver *SquirrelMail*. MSA je softver koji se izvršava se na lokalnoj mašini (127.0.0.1). MTA je *Postfix* softver na serveru *kondor.etf.bg.ac.rs*. Nadređeni MTA je softver Postfix na serveru *smtp.etf.bg.ac.rs*, destinacioni MTA je softver na serveru *mx.google.com*, a MDA je softver na serveru 10.210.105.20. Poruka je isporučena korisniku *uskokovic@gmail.com*, a poslata je sa adresе *uskokovic@etf.rs*.

3.2.2 Zaglavljia (headers)

Zaglavljia poruke generiše MUA, ali ih mogu prepraviti MTA-ovi. U njima se nalaze informacije o adresi pošiljaoca (From: polje), adresi primaoca (To:), naslovu poruke (Subject:), datumu i sl.

Korisnički programi često dodaju polje User-Agent i druge informacije. Ukoliko telo poruke sadrži enkodovane podatke (slike i sl) onda se granica kojom se razdvajaju pojedini enkodovani fajlovi navodi u Content-Type polju.

```
Message-ID: x<38424.91.148.91.47.1231805034.squirrel@kondor.etf.bg.ac.rs>
Date: Tue, 13 Jan 2009 01:03:54 +0100 (CET)
Subject: testFrom: "Marko Uskokovic" <uskokovic@etf.rs>
To: uskokovic@gmail.com
User-Agent: SquirrelMail/1.4.10aMIME-Version: 1.0
Content-Type: multipart/mixed; boundary="----=_20090113010354_76450"
X-Priority: 3 (Normal)
Importance: Normal
```

3.2.3 Telo poruke (body)

Telo poruke se od koverte i zaglavlja odvaja jednim praznim redom i to je prvi takav red od početka poruke. Ukoliko se sa porukom šalju i slike ili drugo, oni se navode u obliku enkodovanog teksta između graničnika koji je naveden u zaglavlju poruke.

```
-----_20090113010354_76450
Content-Type: text/plain; charset="utf-8"
Content-Transfer-Encoding: 8bit

Pozdrav! -- Potpis

-----_20090113010354_76450
Content-Type: image/gif; name="crn_btmlft.gif"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="crn_btmlft.gif"

R0LGODlh0gAfALMAAAAAAK2t raWlpZmZmcXFxWZmZv///+Dg4A8QEnZ2drKysszMzIeHiMDAwP//_
/wAAACH5BAUUA4ALAAAAA6AB8AAATLEJBAaz0plc27/2AYSoFgnuahrmzrvnCsSmht21aul3d/
0r6gcEgEnnbIJJJoi1GCS6a0Zpxar6YqdnvTabng3jctozJQ6VifUaH224uPI6d0632uzRPV+b4
ekMSCLMDhoeIAldKg4FkEgRXiZ0IYAAxElH0pcEhFiTdAadn45QSKMADJ56fkKjl6ulkpSGUxaw
saymu7m6s61+Fb6XkJ5ryMnKa1FbxKkAC9LT1NXTBNjZ2tvcBA3f4N/PsADLAjo6err703u7hEA
0w=-
-----_20090113010354_76450 --
```

3.3 Struktura skladišta za e-mail poruke

3.3.1 Mbox

U Mbox formatu sve poruke iz jednog sandučeta su spojene i zapisane u jednom fajlu. Pojedine poruke se razdvajaju linijom koja sadrži samo znak '..'. Nakon svake poruke se dodaje još jedan prazan red kako bi programi mogli pravilno da obrađuju tekst i izvuku pojedine poruke iz sandučeta. Format Mbox sandučeta nikad nije prošao standardizaciju tako da postoje različite implementacije ovog formata od različitih programa.

3.3.2 Maildir

Maildir je noviji, široko-korišćeni format za skladištenje e-mail poruka. Superiorniji je u odnosu na Mbox jer se poruke čuvaju u zasebnim fajlovima tako da nije potrebno brinuti o zaključavanju prilikom dodavanja, brisanja ili pomeranja poruka. Fajlovi poruka imaju jedinstvena imena i smešteni su u Maildir direktorijumu sa poddirektorijumima "cur", "new" i "tmp". Maildir+ format omogućava i formiranje poddirektorijuma sandučeta. Oni se formiraju kao skriveni poddirektorijumi Maildir direktorijuma (imena im počinju tačkom), tako da bi sanduče *Inbox/Posao* u Maildir-u bilo smešteno u direktorijumu *.Inbox.Posao*.

3.4 Protokoli za slanje i primanje pošte

3.4.1 SMTP

Simple Mail Transfer Protocol (SMTP) je standardni protokol za slanje elektronske pošte. Koriste ga klijenti kada svom MSA-u prosleđuju poštu koja treba da se pošalje, kao i MTA-ovi između sebe prilikom isporuke pošte. SMTP je definisan RFC-om 821, dok RFC 5321 definiše *extended SMTP* (ESMTP).

Sledi primer SMTP komunikacije između klijenta i servera prilikom slanja pošte:

```
uskokovic@kondor:~$ telnet zmaj.etf.rs 25
Trying 147.91.8.62...
Connected to zmaj.etf.rs.
Escape character is '^]'.
220 zmaj.etf.bg.ac.rs ESMTP
HELO kondor.etf.rs
250 zmaj.etf.bg.ac.rs
MAIL FROM: root@kondor.etf.rs
250 Ok
RCPT TO: uskokovic@etf.rs
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: Root korisnik
Subject: Primer SMTP komunikacije
Primer tela poruke
Tackom se zavrsava poruka
.
250 Ok: queued as 4E34A1BB278
QUIT
221 Bye Connection closed by foreign host.
```

3.4.2 POP3

Post Office Protocol verzije 3 je protokol kojim klijenti preuzimaju pristiglu poštu sa servera na svoj računar. Izvršava se na standardnom TCP portu 110, odnosno 995 za SSL enkriptovane konekcije. Namenjen je prvenstveno za korisnike koji nemaju stalnu vezu sa serverom, već prilikom povezivanja na server preuzimaju sve poruke na lokalni računar za kasnije čitanje. Poruke se mogu i čuvati na serveru, mada je za takvu upotrebu bolje koristiti IMAP. Opisan je RFC-om 1939.

Evo primera komunikacije između klijenta i servera:

```
turncoat@bot:~$ telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]' .
+OK Hello there.
USER korisnik
+OK Password required.
PASS sifra
+OK logged in.
LIST
+OK 2 messages (320 octets)
1 120
2 200.
RETR 1
+OK 120 octets
poruka
...
DELETE 1
+OK message 1 deleted
QUIT
+OK POP3 server signing off
```

3.4.3 IMAPv4

Internet Message Access Protocol verzije 4 je zajedno sa POP3 protokolom, najzastupljeniji protokol za pristup i preuzimanje pristigle pošte. Izvršava se na standardnom TCP portu 143, odnosno 993 za SSL enkriptovane konekcije. Namenjen je prvenstveno za korisnike koji

maju stalnu vezu sa serverom i koji žele da svu svoju poštu čuvaju na serveru. Za razliku od POP3 protokola IMAP4 podržava korišćenje poddirektorijuma sandučeta, paralelan pristup sandučetu od strane više klijenata, parcijalno preuzimanje poruka (npr. samo telo, a ne i attachment), označavanje poruka (pročitana, odgovorena...) i pretraživanje poruka na serveru. Definisan je RFC-om 3501.

Primer komunikacije:

```
user@machine:~$ telnet localhost 143
Trying 127.0.0.1...
Connected to imap.example.com
Escape character is '^]'
* OK Dovecot ready.
a1 LOGIN username password
a1 OK Logged in.
a2 LIST "" "*"
* LIST (\ HasNoChildren) "." "INBOX"
a2 OK List completed
a3 EXAMINE INBOX
* FLAGS (\ Answered \ Flagged \ Deleted \ Seend \ Draft)
* OK [PERMANENTFLAGS ()] Read-only mailbox.
* 1 EXISTS
* 1 RECENT
* OK [UNSEEN 1] First unseen.
* OK [UIDVALIDITY 1247842737] UIDs valid
* OK [UIDNEXT 2] Predicted next UID
a3 OK [READ-ONLY] Select completed.
a4 FETCH 1 BODY[]
* 1 FETCH (BODY[] {405}
Return-Path: sender@example.com
Recieved: from client.example.com ([192.168.2.1])
    by mx1.example.com with ESMTP
    id $<$20040120203404.CCCC18555.mx1.example.com@client.example.com$>$
    for $<$recipient@example.com$>; Tue, 20 Jan 2004 22:34:24 +0200
From: sender@example.com
Subject: Test message
To: recipient@example.com
Message-Id: $<$20040120203404.CCCC18555.mx1.example.com@client.example.com$>

This is a test message.
)
a4 OK Fetch completed.
a5 LOGOUT
* BYE Logging outa5 OK logout completed.
```

3.5 Podešavanje DNS zapisa za e-mail

E-mail adresa se sastoji iz dva dela: korisnika i domena ili imena računara na kome postoji taj korisnik. Prilikom slanja pošte, MTA konsultuje DNS tražeći MX (*Mail eXchange*) zapis za domen za koji je upućena poruka. Kao odgovor na ovaj upit, dobija se lista A zapisa servera koji primaju poštu za taj domen. Svaki od tih servera je označen nekim prioritetom, tako da se, ako konekcija sa prioritetskim serverom ne može biti ostvarena, poruka može isporučiti drugom serveru.

Ukoliko MX zapis za traženi domen ne postoji, poruka će biti isporučena serveru koji stoji kao A zapis za taj domen. Ako postoji, MX zapis mora biti A zapis, a ne CNAME.

Sledi primer zone fajla sa definisanim MX zapisom za domen lita.internal:

```
$TTL 2d ; zone default = 2 days or 172800 seconds
$ORIGIN lita.internal.

lita.internal. IN SOA      lita.internal. uskokovic.etf.rs. (
                          14           ; Serial
                          604800       ; Refresh
                          86400        ; Retry
                          2419200     ; Expire
                          604800 )    ; Negative Cache TTL
;
IN      MX  10      mail.lita.internal.

mail   IN      A       10.0.0.2
```

3.6 Postfix SMTP server

Postfix je SMTP server kompatibilan sa Sendmail SMTP serverom koji je bio prvi SMTP server i dugo vremena najupotrebljiviji takav server na Unix i Linux mašinama. No, Sendmail je poznat i po tome što nije modularan, ima kompleksan sistem konfiguracije i priličan broj bezbednosnih propusta. Iz tih razloga je nastao Postfix, koji je modularan, ima bolju bezbednost, lakše se konfiguriše i ima bolje performanse.

Prilikom instalacije Postfixa, na Ubuntu/Debian sistemima je moguće odmah definisati osnovnu konfiguraciju. Kod RedHat/CentOS sistema, postfix je prekonfigurisan da prima konekcije od lokalnih korisnika.

3.6.1 Komponente Postfixa

Glavna komponenta Postfix sistema je *master* daemon. Ovaj daemon upravlja drugim komponentama (agentima), koji obezbeđuju postfixovu modularnost.

Prijem poruka

Postfix prima poruke preko mreže ili direktno od lokalnih korisnika. U slučaju prijema poruka sa mreže koriste se agenti *smtpd* i *qmqpd* koji poruke dalje prosleđuju agentu *cleanup*.

Lokalne poruke prosleđuje program *sendmail* koja obezbeđuje kompatibilnost sa Sendmail programom. Program-agent *sendmail* dalje prosleđuje poruke agentu *postdrop*, koji ih smešta u red 'maildrop', iz kojeg ih čita agent *pickup* koji ih prosleđuje *cleanup* agentu, baš kao i kod prijema poruka sa mreže.

Agent *cleanup* prosleđuje poruke agentu *trivial-rewrite* koji, u zavisnosti od konfiguracije, pretvara lokalne adrese pošiljalaca u kanonički oblik, a zatim celu poruku vraća nazad *cleanup* agentu koji ih onda smešta u 'incoming' red.

Slanje poruka

Pri slanju poruka, glavnu ulogu ima *qmgr* (Queue Manager - menadžer redova). On povlači poruke iz 'incoming' reda i zatim ih smešta u red 'active', koji služi kao 'radni sto' *qmgr* agenta. Broj poruka u redu 'active' je mnogo manji nego u 'incoming' redu, što omogućava da Postfix izdržava velika opterećenja bez zagušivanja memorije. Agent *qmgr* dalje prosleđuje poruke ponovo ka *trivial-rewrite* agentu, koji ovog puta pokušava da prepravi adrese primaoca. Ovaj agent potom ponovo vraća poruke agentu *qmgr* koji ih onda šalje preko nekog od specijalnih agenata za slanje: *smtp*, *lsmtp*, *local*, *virtual*, *pipe*, *discard* ili *error*.

Poruke koje trenutno ne mogu biti isporučene iz različitih razloga agent *qmgr* smešta u red 'deferred'. Agent *qmgr* povereno pokušava da ponovo pošalje poruke iz 'deferred' reda.

3.6.2 Postfix komande

Postfix obezbeđuje sledeće komande:

postalias/newalias Procesiranje alias baze (podrazumevano `/etc/aliases`).

postconf Prikazuje ili menja konfiguraciju Postfixa koja je smeštena u fajlu `main.cf`.

postfix Kontroliše rad Postfix servisa, uključujući startovanje, gašenje, proveru stanja i učitavanje nove konfiguracije.

postmap Konvertuje tekstualne tabelarne fajlove u odgovarajuće binarne zapise koje Postfix učitava.

postqueue/mailq Prikazuje poruke u redovima i upravlja njima.

3.6.3 Konfiguracioni fajlovi

Postfix ima dva glavna konfiguraciona fajla, `master.cf` i `main.cf`. Pored ovih, Postfix koristi i dodatne fajlove koji obično sadrže podatke za filtriranje, prosleđivanje, virtuelne korisnike itd.

master.cf

Fajl `master.cf` je tekstualni fajl koji definiše konfiguraciju *master* daemona. Ova konfiguracija definiše koji agenti i na koji način trebaju biti startovani. Svaka linija predstavlja jedan zapis. Ako linija počinje znakom '#' onda se ona zanemaruje, pošto je u pitanju komentar. Jedna 'logička' linija može biti napisana kao više linija teksta, ali se ona i dalje tretira kao jedna linija. Svaka linija koja počinje od prve x-pozicije u fajlu je početak logičke linije. Linije koje su uvućene za dva blanko znaka u odnosu na prethodnu liniju koja počinje od početka reda predstavljaju nastavak logičke linije.

Svaki zapis ima osam polja razdvojenih jednim ili više blanko znakova. Poslednje polje može sadržati blanko znakove bez posebnog navođenja, tj. osmo polje se proteže od prvog ne-blank znaka posle sedme praznine pa do kraja 'logičke' linije.

Polja su:

service Ime servisa/agenta.

type Način transporta koji se primenjuje

private Da li je servis namenjen isključivo za Postfix.

unpriv Da li se servis izvršava kao neprivilegovani korisnik.

chroot Da li se servis izvršava chroot-ovan za odgovarajući red.

wakeup Vremenski interval (u sekundama) kada se pokreće servis.

maxproc Maksimalni broj procesa/niti koje servis može da kreira.

command+args Komanda i njeni argumenti koji pokreću servis.

Primer `master.cf` fajla:

```
#  
# Postfix master process configuration file. For details on the format  
# of the file, see the master(5) manual page (command: "man 5 master").  
#  
# Do not forget to execute "postfix reload" after editing this file.  
#  
# ======  
# service type private unpriv chroot wakeup maxproc command + args  
#           (yes)   (yes)   (yes)   (never) (100)
```

```

# =====
smtp      inet  n      -      n      -      -      smtpd
submission  inet  n      -      n      -      -      smtpd
  -o smtpd_tls_security_level=encrypt
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
  -o milter_macro_daemon_name=ORIGINATING
smtps     inet  n      -      n      -      -      smtpd
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
  -o milter_macro_daemon_name=ORIGINATING
amavisfeed unix  -      -      n      -      2      lmtp
  -o lmtp_data_done_timeout=1200
  -o lmtp_send_xforward_command=yes
  -o disable_dns_lookups=yes
  -o max_use=20
127.0.0.1:10025 inet  n      -      n      -      -      smtpd
  -o content_filter=
  -o smtpd_delay_reject=no
  -o smtpd_client_restrictions=permit_mynetworks,reject
  -o smtpd_helo_restrictions=
  -o smtpd_sender_restrictions=
  -o smtpd_recipient_restrictions=permit_mynetworks,reject
  -o smtpd_data_restrictions=reject_unauth_pipeline
  -o smtpd_end_of_data_restrictions=
  -o smtpd_restriction_classes=
  -o mynetworks=127.0.0.0/8
  -o smtpd_error_sleep_time=0
  -o smtpd_soft_error_limit=1001
  -o smtpd_hard_error_limit=1000
  -o smtpd_client_connection_count_limit=0
  -o smtpd_client_connection_rate_limit=0
  -o receive_override_options=no_header_body_checks,
no_unknown_recipient_checks,no_milters,no_address_mappings
  -o local_header_rewrite_clients=
  -o smtpd_milters=
  -o local_recipient_maps=
  -o relay_recipient_maps=
pickup    fifo  n      -      n      60      1      pickup
cleanup   unix  n      -      n      -      0      cleanup
qmgr      fifo  n      -      n      300      1      qmgr
tlsmgr    unix  -      -      n      1000?    1      tlsmgr
rewrite   unix  -      -      n      -      -      trivial-rewrite
bounce    unix  -      -      n      -      0      bounce
defer     unix  -      -      n      -      0      bounce
trace     unix  -      -      n      -      0      bounce
verify    unix  -      -      n      -      1      verify
flush     unix  n      -      n      1000?    0      flush
proxymap  unix  -      -      n      -      -      proxymap
proxywrite unix  -      -      n      -      1      proxymap
smtp      unix  -      -      n      -      -      smtp
relay     unix  -      -      n      -      -      smtp
  -o smtp fallback relay=
showq    unix  n      -      n      -      -      showq
error     unix  -      -      n      -      -      error
retry     unix  -      -      n      -      -      error
discard   unix  -      -      n      -      -      discard
local     unix  -      n      n      -      -      local
virtual   unix  -      n      n      -      -      virtual

```

```
lsmtp    unix  -      -      n      -      -      lsmtp
anvil    unix  -      -      n      -      1      anvil
scache   unix  -      -      n      -      1      scache
```

main.cf

Sadržaj main.cf fajla, pored običnog izlistavanja, može se dobiti i komandom:

```
# postconf -n
```

Komandom:

```
# postconf -d
```

se mogu videti podrazumevane vrednosti parametara koje nisu izmenjene u main.cf fajlu.

Primer main.cf fajla za sistem sa virtuelnim korisnicima je prikazan u nastavku:

```
# PUTANJE I VLASNIŠTVA NAD FAJLOVIMA
#
sendmail_path = /usr/sbin/sendmail.postfix
newaliases_path = /usr/bin/newaliases.postfix
mailq_path = /usr/bin/mailq.postfix
setgid_group = postdrop
html_directory = no
manpage_directory = /usr/share/man
sample_directory =
readme_directory =

queue_directory = /var/spool/postfix
command_directory = /usr/sbin
daemon_directory = /usr/libexec/postfix
data_directory = /var/lib/postfix

mail_owner = postfix

# INTERNET HOST I DOMENSKA IMENA
#
myhostname = mail.example.com
mydestination = example.com

# SENDING MAIL
#
myorigin = $mydestination

# RECEIVING MAIL
#
inet_interfaces = all
inet_protocols = all

# TLS parameters
smtpd_tls_cert_file = /etc/pki/tls/certs/server.crt
smtpd_tls_key_file = /etc/pki/tls/private/server.key
smtpd_use_tls = yes
smtpd_tls_session_cache_database = btree:$data_directory/smtpd_scache
smtp_tls_session_cache_database = btree:$data_directory/smtp_scache

# SASL parametri
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_local_domain = example.com
```

```
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
smtpd_tls_auth_only = no

# REJECTING MAIL FOR UNKNOWN LOCAL USERS
#
unknown_local_recipient_reject_code = 550

# TRUST AND RELAY CONTROL
#
mynetworks = 127.0.0.1/8

# ADDRESS REWRITING
#
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases

# VIRTUAL MAIL
#
virtual_mailbox_domains = mysql:/etc/postfix/db/vmd.cf
virtual_alias_maps = mysql:/etc/postfix/db/vam.cf
virtual_mailbox_maps = mysql:/etc/postfix/db/vmm.cf
virtual_mailbox_base = /srv/vmail
virtual_minimum_uid = 500
virtual_uid_maps = 500
virtual_gid_maps = 500
virtual_transport = dovecot

# RESTRIKCIJE
#
smtpd_etrn_restrictions =
    reject_unknown_client,
    permit_mynetworks,
    reject

smtpd_client_restrictions =
    check_client_access hash:/etc/postfix/maps/access-client,
    permit_mynetworks,
    reject_rbl_client sbl.spamhaus.org,
    reject_rbl_client blackholes.easynet.nl,
    permit

smtpd_helo_restrictions =
    check_helo_access hash:/etc/postfix/maps/access-helo,
    reject_invalid_hostname,
    permit

smtpd_sender_restrictions =
    check_sender_access hash:/etc/postfix/maps/access-sender,
    permit_mynetworks,
    reject_non_fqdn_sender,
    reject_unknown_sender_domain,
    permit

smtpd_recipient_restrictions =
    check_recipient_access hash:/etc/postfix/maps/access-recipient,
    reject_unknown_sender_domain,
    reject_unknown_recipient_domain,
    permit_mynetworks,
```

```
permit_sasl_authenticated,
reject_invalid_hostname,
reject_non_fqdn_sender,
reject_non_fqdn_recipient,
reject_unauth_destination,
reject_unauth_pipelining,
reject_rbl_client sbl-xbl.spamhaus.org,
reject_rbl_client bl.spamcop.net,
reject_rbl_client cbl.abuseat.org,
reject_rhsbl_sender dsn.rfc-ignorant.org,
permit
#reject_non_fqdn_hostname
#check_relay_domains

# PERFORMANCE TUNING
#
# broj konekcija
smtpd_client_connection_rate_limit = 0
#broj simultanih konekcija
smtpd_client_connection_count_limit = 0
#broj mailova po klijentu
smtpd_client_message_rate_limit = 0
# SHOW SOFTWARE VERSION OR NOT
smtpd_banner = $myhostname ESMTP $mail_name

# DEBUGGING KONTROLA
#
debug_peer_level = 2
debugger_command =
    PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin
    ddd $daemon_directory/$process_name $process_id & sleep 5
```

3.7 Dovecot POP i IMAP

Dovecot se jedan od češće korišćenih POP i IMAP servera za Linux. Popularan je zbog svoje brzine, jednostavnosti podešavanja i zbog dobre sigurnosti. Radi podjednako dobro i sa malim i sa velikim brojem korisnika zahvaljujući indeksiranju poruka koje transparentno obavlja. Podržava mbox i maildir skladišta poruka, koja bezbedno mogu biti i na NFS-u. Autentifikacija korisnika se može obaviti prema više baza korisnika i mehanizama autentifikacije (PAM, MySQL, LDAP...). Postoji i podrška za plugin-ove kao što su kvote i pristupne liste.

3.7.1 Instalacija

Ubuntu:

```
sudo apt-get install dovecot-imapd dovecot-pop3d
```

RedHat/CentOS:

```
yum install dovecot
```

3.7.2 Konfiguracioni fajlovi

Konfiguracioni fajl za dovecot je standardno `/etc/dovecot/dovecot.conf`, ali se putanja do fajla koji se koristi može videti kada se dovecot pokrene sa parametrom `-n`. U verziji 2 ovog softvera, delovi fajla koji su vezani za različite funkcionalnosti softwera su razdvojene na zasebne fajlove, koji se uključuju u osnovni konfiguracioni fajl. Ovi fajlovi se nalaze u direktorijumu `/etc/dovecot/conf.d/`:

10-auth.conf U ovom fajlu se podešavaju načini autentifikacije.

10-director.conf Podešavanje direktor procesa, koji služe za mapiranje 'korisnik→server' kada se Dovecot koristi kao proksi server.

10-logging.conf U ovom fajlu se definiše šta se i kako loguje.

10-mail.conf U ovom fajlu se definišu lokacije mailboxova.

10-master.conf Definicija internih servisa Dovecot-a.

10-ssl.conf Podešavanje SSL/TLS parametara

15-lda.conf Podešavanja za LDA (*Local Delivery Agent*) protokol, koji služi da preuzme poruku od MTA i isporuči je lokalno Dovecot-u.

20-imap.conf Podešavanja za IMAPv4 servis.

20-lmtp.conf Podešavanja za LMTP (*Local Mail Transport Protocol*), protokol čija je namena ista kao kod LDA protokola.

20-managesieve.conf Podešavanja plugin-a za *Sieve*, servis koji omogućava manipulaciju porukama prilikom isporuke, uključujući njihovo filtriranje, snimanje u različite pod-foldere i kreiranje *out-of-office* poruka.

20-pop3.conf Podešavanja za POP3 servis.

90-acl.conf Podešavanja prava pristupa za IMAP foldere.

90-plugin.conf Specifikacija plugin-ova koji se koriste u zadatoj instanci Dovecot-a.

90-quota.conf Podešavanja za kvote.

90-sieve.conf Podešavanja Sieve protokola (obratiti pažnju da se u fajlu **20-managesieve.conf** podešava samo plugin).

auth-checkpassword.conf.ext Opcioni fajl koji se uključuje u **10-auth.conf** i konfiguriše autentifikaciju preko 'checkpassword' drajvera.

auth-deny.conf.ext Podešavanje koje definiše kojim korisnicima je zabranjen pristup.

auth-ldap.conf.ext Podešavanje autentifikacije za LDAP. Kredencijali za pristup LDAP serveru i mapiranje LDAP atributa u Dovecot atributu se podešava u zasebnom fajlu.

auth-master.conf.ext Podešavanje autentifikacije za tzv. 'master' korisnike.

auth-passwdfile.conf.ext Podešavanje korisnika koji se čuvaju u zasebnom, lokalnom passwd fajlu.

auth-sql.conf.ext Podešavanja za autentifikaciju korisnika preko SQL baze podataka. Kredencijali za pristup bazi, upit koji vraća potrebne podatke je definisan u zasebnom fajlu.

auth-static.conf.ext Podešavanja za statičku autentifikaciju (npr. svi korisnici imaju istu lozinku).

auth-system.conf.ext Podešavanje za autentifikaciju korisnika definisanih na sistemu. Za autentifikaciju se koristi PAM sistem.

auth-vpopmail.conf.ext Podešavanje za VPOPmail autentifikaciju.

3.7.3 Primer konfiguracije: virtuelni korisnici definisani u MySQL bazi

U ovom primeru konfiguracije, korisnici su definisani u lokalnoj MySQL bazi. Baza se zove 'mail' a unutar nje se korisnici nalaze u tabeli 'mailbox'. Dovecot je konfigurisan da pruži POP3 i IMAPv4 servis sa i bez SSL/TLS enkripcije. Podrazumevani domen korisnika je 'domain.tld'.

Kod virtualnih korisnika, koji ne postoji na sistemu, potrebno je kreirati jednog sistemskog korisnika koji će biti vlasnik svih fajlova. Takođe, potrebno je kreirati i odgovarajući direktorijum u kojem će se smeštati mejlboksovi virtualnih korisnika. U ovom primeru, virtualni korisnik je 'vmail', njegova grupa je 'vmail', a osnovni direktorijum '/srv/vmail'. UID korisnika 'vmail' je 493, a njegov GID, tj. GID grupe 'vmail' je 494.

Fajl 10-auth.conf

U ovom fajlu ćemo definisati da se ne prihvata metod autentifikacije gde se lozinka prenosi u otvorenom tekstu (disable_plaintext_auth = no), da je podrazumevani domen 'domain.tld' (auth_realms i auth_default_realm), da je format imena 'user@domain.tld' (auth_username_format = "%n@%d"), te da prihvatamo metode autentifikacije 'plain' i 'login'.

Ostale stvari vezane za autentifikaciju, uključujući specifikacije za 'userdb' i 'passdb' su definisane u fajlu auth-sql.conf.ext:

```
disable_plaintext_auth = no
auth_realms = domain.tld
auth_default_realm = domain.tld
auth_username_format = "%n@%d"
auth_mechanisms = plain login
!include auth-sql.conf.ext
```

Fajl 10-mail.conf

U ovom fajlu ćemo definisati UID i GID za sistemskog korisnika koji je vlasnik virtualnih mejlboksova, koje plugin-ove ćemo učitati i način zaključavanja fajlova:

```
first_valid_uid = 494
first_valid_gid = 493
mail_plugins = quota expire mail_log notify
mbox_write_locks = fcntl
```

Fajl 10-master.conf

U fajlu 10-master.conf treba izmeniti i/ili dodati sledeće stavke:

```
service auth
    unix_listener auth-userdb
        mode = 0660
        user = vmail
        group = vmail

    unix_listener /var/spool/postfix/private/auth
        mode = 0660
        user = postfix
        group = postfix
```

Servis auth definiše dva UNIX soketa, 'auth-userdb' i '/var/spool/postfix/private/auth', sa zadatim vlasništvima i pravima pristupa. Ovi soketi se koriste kada neki drugi sistem, kao što je Postfix, želi da izvrši SASL autentifikaciju korisnika. SASL je mehanizam generičkog autentifikovanja, gde su način autentifikacije i načini prenosa kredencijala odvojeni od same aplikacije i smešteni u zaseban modul, kojem može da pristupi više različitih aplikacija.

Fajl 10-ssl.conf

U fajl 10-ssl.conf treba dodati putanje do privatnog ključa i SSL sertifikata (obratite pažnju na znak '<' ispred putanje fajla!):

```
ssl_cert = </etc/pki/dovecot/certs/dovecot.pem
ssl_key = </etc/pki/dovecot/private/dovecot.pem
```

Fajl auth-sql.conf.ext

Pošto smo u fajlu 10-auth.conf uključili fajl auth-sql.conf.ext, njegov sadržaj je:

```
passdb
  driver = sql
  args = /etc/dovecot/dovecot-sql.conf.ext

userdb
  driver = prefetch
```

Generalno, možemo koristiti različite upite za bazu lozinki (password_query) i bazu korisničkih informacija(user_query). No, ako se oba ova upita prave nad istom tabelom u bazi, onda je optimalnije da pribavljanje parametara svedemo na jedan upit, password_query. Iz tog razloga, 'userdb' sekcija koristi 'prefetch' drajver koji označava da je neka druga sekcija (passdb) već povukla sve potrebne informacije.

Upit 'user_query' se izvršava za svakog korisnika i on mora da vrati sledeće parametre:

home Početni direktorijum virtuelnog korisnika

mail Maildir virtuelnog korisnika (apsolutna putanja)

uid UID virtuelnog korisnika (uvek će biti isti za sve korisnike i to je UID 'vmail' korisnika)

gid GID virtuelnog korisnika (uvek će biti isti za sve korisnike i to je GID 'vmail' korisnika)

quota Kvota parametar, ako se koristi kvota.

Upit 'password_query' se takođe izvršava za svakog korisnika i on mora da vrati sledeće parametre:

user Korisničko ime virtuelnog korisnika

password Kriptovana lozinka virtuelnog korisnika

Pored ovih parametara, upit 'password_query', može vratiti i parametre koje vraća upit 'user_query' tako da možemo izbeći da šaljemo dva upita za jednog korisnika. U tom slučaju imena parametara su nešto izmenjena:

user_query	password_query
home	userdb_home
mail	userdb_mail
uid	userdb_uid
gid	userdb_gid
quota	userdb_quota

Tabela 3.1: Nazivi identičnih parametara kod 'user_query' i 'password_query'

Fajl dovecot-sql.conf.ext

U fajlu /etc/dovecot/dovecot-sql.conf.ext se nalaze kredencijali i upiti za bazu. Sadržaj ovog fajla je:

```
default_pass_scheme=MD5-CRYPT
driver = mysql
connect = host=127.0.0.1 dbname=mail user=useradm password=userpass
password_query = SELECT
    username as user,
    password,
    concat('/srv/vmail/', maildir, 'Maildir/') as userdb_home,
    concat('maildir:/srv/vmail/', maildir, 'Maildir/') as userdb_mail,
    494 as userdb_uid,
    493 as userdb_gid
FROM brlmailbox
WHERE username = '%u' AND active = '1'
```

Property of Admin Training Center

Modul 4: Squid proxy servis

4.1 Uvod

Squid server može na radi kao:

proxy server — šalje zahteve u ime svojih klijenata, uz mogućnost filtriranja ko, kada, i koje adrese može da poseti.

cache server — ubrzava veb brauzovanje čuvanjem lokalnih kopija fajlova koji se češće dovlače i kojima nije naznačeno da ne treba da budu keširani.

transparentni proxy/cache server — transparentan rad, gde klijent ne zna da je saobraćaj proksiran ili keširan.

httpd akcelerator — tzv. reverzni proksi, koristi se da se ubrzaju određeni sajtovi keširanjem njihovih sadržaja.

Squid server može biti povezan u mrežu proksi servera, koristeći ICP protokol (radi preko UDP-a) ili HTTP protokol (radi preko TCP-a). Odnos između različitih proksi servera može biti:

sibling — lokalni squid može da potraži stranu u kešu sibling servera, ali ukoliko ni tamo ne nađe datu stranu, sam će da je dovuče od originalnog servera;

parent — lokalni squid uvek dobija strane koje nema u svom kešu od parent servera, bilo da ih ovaj ima u svom kešu bilo da ih on dovlači od originalnog servera u ime svog child-a.

4.2 Osnovna konfiguracija

```
# OPŠTE OPCIJE
append_domain .etf.bg.ac.rs
coredump_dir /tmp
http_port 8080
pid_filename /var/log/squid/squid.pid
visible_hostname proxy.etf.bg.ac.rs

# ACL DEFINICIJE
acl QUERY urlpath_regex cgi-bin \?
acl all src 0.0.0.0/0.0.0.0
acl localhost src 127.0.0.1/255.255.255.255
acl SSL_ports port 443 563 7443 10000 20000 19638
acl Safe_ports port 80 21 443 563 70 210 1025-65535
acl CONNECT method CONNECT
acl http proto HTTP
```

```

acl ftp proto FTP
acl ftptime time 0:30-06:30
acl lita src 147.91.13.128/25
acl blocked src 147.91.13.161/32

# CACHE DEFINICIJE
cache_mem 24 MB
cache_swap_low 90
cache_swap_high 95
maximum_object_size 8192 KB
cache_dir aufs /var/cache/squid 8200 24 256

# LOGFILE PUTANJE
cache_access_log /var/log/squid/access.log
cache_log /var/log/squid/cache.log
cache_store_log none

# KONTROLA PRISTUPA
no_cache deny QUERY all
http_access deny blocked
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow ftp ftptime
http_access deny ftp !ftptime
http_access allow lita http
http_access allow all password
http_access deny all

```

Osnovna konfiguracija Squid servera se svodi na definisanje ACL-ova i kontrole pristupa. ACL-ovi definišu izraze za kontrolu pristupa i uvek su oblika:

acl *ime tip argumenti*

Ime je ime koje koristimo da pozivamo datu ACL. Tip označava na šta se ACL odnosi i najčešći tipovi su:

urlpath_regex Regularni izraz koji opisuje relativni deo URL-a

src Specifikacija IP adrese mreže ili pojedinih klijenata

srcdomain Specifikacija domenskih imena klijenata

dst Specifikacija IP adresa ciljeva (targeta)

dstdomain Specifikacija domenskih imena ciljena (host deo URL-a)

port Specifikacija TCP portova

method Specifikacija metoda koji klijent izvršava (GET, POST, CONNECT, itd.)

proto Naziv protokola

time Specifikacija vremenskog intervala

Ukoliko želimo da Squid radi kao kešing server onda je potrebno specificirati kešing parametre (deo # CACHE DEFINICIJE). Osnovni parametri su:

cache_mem Količina RAM-a koja se odvaja za keširanje

cache_swap_low, cache_swap_high Parametri koji definišu kolika popunjenoš RAM-a treba da bude kada se objekti iz keš RAM-a prebacuju na disk.

maximum_object_size Najveća dužina fajla koji će biti keširan.

cache_dir Specifikacija direktorijuma gde se smeštaju keširani objekti na disku. Prvi argument je tip fajl sistema, drugi argument je početni direktorijum, treći argument maksimalna veličina celog podstabla u megabajtima, četvrti argument je broj direktorijuma direktno ispod početnog direktorijuma, a poslednji argument je broj direktorijuma drugog nivoa. U primeru sa slike, ispod /var/cache/squid će biti kreirana 24 direktorijuma čija imena će biti velika slova engleske abecede. Unutar svakog od ovih direktorijuma biće kreirano po 256 poddirektorijuma, čija imena će biti heksadeci malni brojevi od 00 do FF. Svaki objekat koji se bude čuvao u kešu će imati ime oblika '<slovo-abecede><niz-heksa-cifara>', koje se dobija hešing funkcijom URL-a samog objekta. Takav fajl će biti smešten u poddirektorijum čije ime odgovara prve dve heksa cifre iz imena fajla, a koji se nalazi u direktorijumu '<slovo-abecede>'. Ukupan prostor koji ovo slabo direktorijuma i fajlova zauzima ne sme da pređe 8200MB.

Kada Squid služi kao proksi server, potrebno je definisati pravila kontrole pristupa (deo # KONTROLA PRISTUPA). Pri specificiranju prava kontrole pristupa se koriste ACL-ovi, tako da se konačan izraz dobija zamenom imena ACL-a njegovom definicijom. Princip određivanja prava pristupa je da se primenjuje prvo pravilo koje odgovara upitu klijenta.

4.3 Transparent proxy server

Obavezni parametri za transparentni proxy u squid.conf fajlu:

```
httpd_accel_host     virtual
httpd_accel_port     80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

Podešavanje netfilter-a:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT \
--to-port 3128
```

Property of Admin Training Center

Modul 5: Samba fajl server

5.1 Uvod

Samba je mrežna alatka za Unix-olike sisteme koja omogućava deljenje fajlova i štampača, autentifikaciju korisnika i autorizaciju aktivnosti, rezolvovanje imena i oglašavanje servisa između Windows i Unix-olikih sistema i to putem SMB (*Server Message Block*) protokola. Pomoću Samba softvera, Unix-oliki sistem kakav je Linux može efikasno obavljati funkciju *Primarnog Domen Kontrolera* (PDC) za Windows i Linux klijente. Takođe, Samba daje mogućnost Linux klijentima da budu članovi NT4 domena i Aktivnog Direktorijuma Windows Servera.

Trenutno aktuelna verzija Samba paketa je 3.x, softver je licenciran GNU GPL licencom i izvorni kod se može skinuti sa matičnog sajta <http://www.samba.org>.

5.2 Instalacija

Samba se na Debian i Ubuntu sistemima može instalirati komandom:

```
# sudo apt-get install samba
```

Paket samba je na Ubuntu distribuciji povezan sa `samba-common` i `smbclient` paketima, a pored njih od koristi su još i paketi `samba-doc` (dokumentacija), `cifs-utils` (za montiranja samba šerova), `swat` (*Samba Web Administration Tool*) i `winbind` (za autentifikaciju na Windows NT domen).

Na RedHat/CentOS sistemima, Samba se instalira komandom:

```
# yum install samba
```

Ova komanda će, pored paketa `samba` instalirati i pakete `samba-client`, `samba-common`, `samba-winbind` i `samba-winbind-clients`, ukoliko oni već nisu instalirani.

5.2.1 Samba kao klijent

5.2.2 Samba alatke

findsmb Služi za pronalaženje SMB hostova u mreži i daje izveštaj o imenu mašine, IP adresi, domenu ili radnoj grupi kojoj pripada i operativnom sistemu koji je na hostu.

smbtree Daje kompletan spisak deljenih resursa koji su dostupni u lokalnoj mreži

nmblookup Služi za pronalaženje IP adrese hosta koji se odaziva na navedeno NetBIOS ime

net Alatka za izvršavanje RPC komandi na udaljenim mašinama

smbclient Klijent kojim se može pristupati SMB resursima na udaljenim mašinama

5.2.3 Korišćenje cifs

Nakon instalacije `cifs-utils` paketa može se učitati `cifs` modul i time dobiti mogućnost montiranja smb šerova u stablo lokalnog fajl sistema.

```
student@la-p:~$ sudo modprobe smbfs
student@la-p:~$ grep cifs /proc/filesystems
nodev   cifs
```

Montiranje se može izvesti pomoću mount komande sa parametrom '`-t cifs`':

```
# mount -t cifs -o username=winuser,password=winnpassword
winbox
share /mnt/winshare
```

5.3 Samba kao server

Samba ima tri mrežna daemona:

smbd Daemon koji pruža fajl i printer servise, autentifikaciju i autorizaciju, zaključavanje resursa i drugo. Sluša na portovima 139/tcp i/ili 445/tcp.

nmbd Daemon koji razgovara sa drugim SMB/CIFS hostovima putem NetBIOS protokola i učestvuje u oglašavanju resursa. Sluša na portovima 137/udp i/ili 138/udp.

winbindd Daemon koji služi za mapiranje korisnika i grupa iz Windows NT domenu na Unix sistemske korisnike i grupe radi obezbeđivanja single sign-on okruženja.

Konfiguracioni fajl svih sambinih daemona je `/etc/samba/smb.conf`.

5.4 Načini autentifikacije

Samba autentificira zahteve za resursima putem dva nivoa sigurnosti i dva moda sigurnosti.

5.4.1 Korisnički nivo sigurnosti (User level security)

Autentifikacija klijenta i autorizacija zahteva se obavlaju prilikom povezivanja na server, i to na osnovu hosta sa kog zahtev dolazi i korisničkog imena i šifre koji se prosleđuju kao deo zahteva. Klijent koristi sve dostupne resurse servera (šerove i štampače) bez ponovnog unošenja šifre. Moguća je i situacija gde klijent dobija više autentifikacionih konteksta na osnovu više poslatih zahteva sa različitim kombinacijama korisničkog imena i šifre.

5.4.2 Sigurnost na nivou šera (Share level security)

Autorizacija zahteva se obavlja prilikom povezivanja na deljeni resurs, specificiranjem jedinstvene šifre tog resursa za sve korisnike.

5.4.3 NT4 domen mod sigurnosti (Domain security mode)

Koristi se kada Samba ima ulogu člana NT4 domena i autentifikacija korisnika se ne obavlja na ovom hostu, već se prosleđuje domen kontroleru.

5.4.4 Aktivni direktorijum mod sigurnosti (ADS security mode)

Koristi se kada Samba ima ulogu člana Aktivnog direktorijuma i autentifikacija korisnika se ne obavlja na ovom hostu, već se prosleđuje password serveru aktivnog direktorijuma.

5.5 Deljenje fajlova i direktorijuma

Primer podešavanja `smb.conf` fajla za šer pod imenom data

```
[data]
comment = Podaci
path = /srv/podaci
admin users = @podaci
valid users = @podaci
write list = @podaci
public = no
writeable = yes
printable = no
browseable = yes
create mask = 0770
directory mask = 0770
force group = +tp
hosts allow = 147.91.13.
veto files = /lost+found/
```

Primer podešavanja `smb.conf` fajla za deljenje home direktorijuma korisnika:

```
[homes]
comment = RC ETF users
browsable = no
writable = yes
```

5.6 Samba kao PDC

5.6.1 Uvod

Domen je oblik centralizovanja mrežnog okruženja u kome se podaci o korisnicima i grupama korisnika, korisnički fajlovi i korisničke privilegije nalaze na serveru koji se naziva domen kontroler i koji je zadužen za obavljanje autentifikacije korisnika i korisničkih zahteva. Uloge koje mogu uzimati mašine u domenu su sledeće:

Domen kontroler

- Primarni domen kontroler
- Sekundarni domen kontroler
- ADS domen kontroler
- WINS server

Član domena

- član NT4 domena
- član Aktivnog direktorijuma
- Lokalni master browser

5.7 Konfigurisanje Sambe

Fajl `/etc/samba/smb.conf`:

```
[global]
workgroup = LITA
server string = Samba PDC
netbios name = la-p
wins support = yes
dns proxy = no
log file = /var/log/samba.log
log level = 1
max log size = 1000
syslog = 0
admin users = admin
security = user
guest account = nobody
encrypt passwords = true
passdb backend = tdbsam
invalid users =
map to guest = Bad Password
password level = 0
add machine script = /usr/sbin/useradd -s /bin/false '%u' -g masine
logon path =
logon home =
domain logons = Yes
os level = 64
preferred master = Yes
domain master = Yes
socket options = TCP_NODELAY
time server = yes

[netlogon]
path = /srv/netlogon
public = no
writeable = no
browsable = no
valid users = @korisnici
```

Potrebno je kreirati direktorijum `/srv/netlogon`, kreirati grupe: `masine`, `ntadmini`, `ntkorisnici`, korisnika `ntadmin` koji će pripadati grupi `ntadmini` i korisnika `ntkorisnik` koji će pripadati grupi `ntkorisnici`:

```
# sudo mkdir -p /srv/netlogon
# sudo groupadd masine
# sudo groupadd ntadmini
# sudo groupadd ntkorisnici
# sudo useradd ntadmin -g ntadmini
# sudo useradd ntkorisnik -g ntkorisnici
```

Zatim je potrebno korisnicima `ntadmin` i `ntkorisnik` kreirati samba naloge i postaviti šifre:

```
# sudo smbpasswd -a ntadmin
# sudo smbpasswd -a ntkorisnik
```

I na kraju kreirati mapiranje između Windows i Unix grupa:

```
#net groupmap add rid=512 unixgroup=ntadmini ntgroup='`Domain Admins``'
#net groupmap add rid=513 unixgroup=ntkorisnici ntgroup='`Domain Users``'
#net groupmap add rid=514 unixgroup=nogroup ntgroup='`Domain Guests``'
```

Nakon restartovanja samba servera može se preći na podešavanje klijenata.

5.8 Podešavanje Windows klijenata

Potrebno je kao korisnik sa administratorskim privilegijama potrebno je ubaciti mašinu u domen. To se može uraditi u prozoru Computer Name Changes koji se dobije nakon desnog klika na My Computer, zatim stavka Properties, pa kartica Computer Name.

U polje Domain treba uneti parametar WORKGROUP iz `smb.conf` fajla, kliknuti na ok i uneti šifru ntadmin korisnika definisanog u Sambi.

Ukoliko je sve prošlo kako treba, javiće se poruka "Welcome to domain" i računar će biti potrebnno restartovati.

Nakon restartovanja će se u login prozoru pojaviti stavka za logovanje na domen.

5.9 Podešavanje Linux klijenata

Da bi se Linux klijent uključio u domen, potrebno je instalirati pakete samba i winbind. U fajlu `/etc/samba/smb.conf` treba uneti sledeća podešavanja u globalnoj sekciji:

```
workgroup = LITA
idmap uid = 10000-20000
idmap gid = 10000-20000
template shell = /bin/bash
template homedir = /home/%D/%U
winbind enum users = yes
winbind enum groups = yes
winbind cache time = 10
winbind separator = +
security = domain
password server = *
winbind use default domain = yes
```

Dalje, treba editovati fajl `/etc/nsswitch.conf` i postaviti:

```
passwd: compat winbind
group: compat winbind
```

U direktorijumu `/etc/pam.d` editovati fajlove: common-account:

```
#Commented for winbind to work
account sufficient pam_winbind.so
account required pam_unix.so

common-auth:
auth sufficient pam_winbind.so
auth required pam_unix.so nullok_secure use_first_pass

common-session:
session required pam_unix.so
session required pam_mkhomedir.so umask=0022 skel=/etc/skel/
```

sudo:

```
auth sufficient pam_winbind.so
auth required pam_unix.so use_first_pass
```

Zatim treba pridružiti mašinu u domen:

```
# net rpc join -D LITA -U ntadmin
```

Kreirati direktorijum `/home/LITA` i restartovati mašinu. Nakon restartovanja možete se ulogovati bilo sa lokalnim, bilo sa domenskim korisničkim imenom i šifrom.

5.10 Implementacija korisničkih profila

Korisnički profili Windows klijenata se mogu čuvati na Samba serveru. Dovoljno je kreirati šer pod imenom `profiles` sa podešavanjima `smb.conf` fajla:

```
[profiles]
path = /var/lib/samba/profiles
browseable = no
writeable = yes
default case = lower
preserve case = no
short preserve case = no
case sensitive = no
hide files = /desktop.ini/ntuser.ini/NTUSER./*
write list = @ntkorisnici @root
create mask = 0600
directory mask = 0700
csc policy = disable
```

Specificiranje korišćenja roaming profila i mapiranja home direktorijuma se obavlja sledećim stavkama globalne sekcije `smb.conf` fajla:

```
logon path = \\%L\ profiles\ %U
logon drive = H:
```

Ukoliko je potrebno izvršavati određene komande na klijentima nakon logovanja, moguće je parametrom `logon script` definisati windows (.bat) skriptu u kojoj se nalaze te komande:

```
logon script = login.bat OR %U.bat
```