

**priručnik za polaznike**

**Uspostava i prilagodba mrežnih servisa**

Debian

L202

Ovu su inačicu priručnika izradili:

Autor: Ivan Rako, Ante Jurjević

Recenzent: Darko Culej

Urednik: Dominik Kenđel

Lektor: dr. sc. Jasna Novak Milić



Sveučilište u Zagrebu

Sveučilišni računski centar

Josipa Marohnića 5, 10000 Zagreb

edu@srce.hr

ISBN 978-953-382-001-9 (meki uvez)

ISBN 978-953-382-002-6 (PDF)

Verzija priručnika L202-20221117



|  |  |
| --- | --- |
|  | Ovo djelo dano je na korištenje pod licencom Creative Commons Imenovanje-Dijeli pod istim uvjetima 4.0 međunarodna (CC BY-SA 4.0). Licenca je dostupna na stranici:  https://creativecommons.org/licenses/by-sa/4.0/deed.hr. |

**Sadržaj**

[Uvod 1](#_Toc117860262)

[1. Mrežni servisi 3](#_Toc117860263)

[1.1. DNS 3](#_Toc117860264)

[1.1.1. Blok logging 6](#_Toc117860265)

[1.1.2. Blok options 7](#_Toc117860266)

[1.1.3. Blok zone 8](#_Toc117860267)

[1.1.4. BIND na Debianu 9](#_Toc117860268)

[1.2. DHCP 14](#_Toc117860269)

[1.3. Vježba 1. Osnovna konfiguracija servisa BIND 19](#_Toc117860270)

[1.4. Vježba 2. DHCP 27](#_Toc117860271)

[2. Servis za udaljeni rad 31](#_Toc117860272)

[2.1. Secure Shell (SSH) 32](#_Toc117860273)

[2.1.1. Protokol SSH 32](#_Toc117860274)

[2.1.2. Arhitektura protokola SSH 33](#_Toc117860275)

[2.1.3. Servis SSH 34](#_Toc117860276)

[2.1.4. Servis SSH 37](#_Toc117860277)

[2.2. Vježba 3. SSH 40](#_Toc117860278)

[3. Autentikacijski servisi 41](#_Toc117860279)

[3.1. LDAP 41](#_Toc117860280)

[3.1.1. Protokol LDAP 41](#_Toc117860281)

[3.1.2. Protokol LDAP 41](#_Toc117860282)

[3.1.3. Upravljanje LDAP-om 44](#_Toc117860283)

[3.1.4. Sheme 45](#_Toc117860284)

[3.1.5. Servis OpenLDAP 45](#_Toc117860285)

[3.2. RADIUS 53](#_Toc117860286)

[3.2.1. Protokol RADIUS 54](#_Toc117860287)

[3.2.2. Servis FreeRADIUS 57](#_Toc117860288)

[3.2.3. Autentikacija osnovnih servisa korištenjem protokola LDAP i RADIUS 59](#_Toc117860289)

[3.3. Vježba 4. LDAP 61](#_Toc117860290)

[4. Elektronička pošta 63](#_Toc117860291)

[4.1. SMTP 63](#_Toc117860292)

[4.2. Upravljanje mailing listama 68](#_Toc117860293)

[4.3. Filtriranje elektroničke pošte 74](#_Toc117860294)

[4.4. POP3 i IMAP 77](#_Toc117860295)

[4.5. Vježba 5. *Postfix* 79](#_Toc117860296)

[4.6. Vježba 6. Upravljanje *mailing* listama 82](#_Toc117860297)

[4.7. Vježba 7. *Dovecot* 83](#_Toc117860298)

[5. Web-servisi 85](#_Toc117860299)

[5.1. HTTP – Apache2 85](#_Toc117860300)

[5.2. Osnove MariaDB-a 100](#_Toc117860301)

[5.3. HTTP – Nginx 110](#_Toc117860302)

[5.4. Usporedba servisa Apache i Nginx 112](#_Toc117860303)

[5.5. Vježba 8. HTTP – *Apache* i *MariaDB* 113](#_Toc117860304)

[5.6. Vježba 9. HTTP - Nginx: 114](#_Toc117860305)

[6. Dijeljenje datoteka 115](#_Toc117860306)

[6.1. SMB/CIFS 115](#_Toc117860307)

[6.2. NFS 121](#_Toc117860308)

[6.3. Sinkronizacija datoteka 124](#_Toc117860309)

[6.4. Vježba 10. SMB/CIFS 129](#_Toc117860310)

[6.5. Vježba 11. NFS 129](#_Toc117860311)

# Uvod

**Trajanje poglavlja:**

**10 min**



Ovaj je tečaj koji polaznike uvodi u napredno korištenje Linuxa. Tečaj služi kao prirodni nastavak tečaja L201 te zajedno s njim predstavlja osnovu za napredno korištenje operacijskog sustava Linux. Operacijski sustav koji se koristi je Debian, konkretno Debian 11 za koji su izrađene vježbe.

Ovaj se tečaj sastoji od šest poglavlja.

Nakon pohađanja ovog tečaja moći ćete uspostaviti:

* DNS za osnovne potrebe domene
* mrežni sustav za dodjeljivanje IP adresa
* raspodijeljeni imenik za autentikaciju
* mail servis
* servis za upravljanje mail listama
* web servis
* servise za dijeljenje datoteka.

1. Mrežni servisi

**Trajanje poglavlja:**

**180 min**



Po završetku ovoga poglavlja moći ćete:

* konfigurirati DNS poslužitelj BIND
* prepoznati razliku između primarnog i sekundarnog DNS poslužitelja
* stvarati zonske datoteke
* transferirati zone
* konfigurirati DHCP poslužitelj
* automatski osvježavati DNS zone iz DHCP poslužitelja.

Ova cjelina obrađuje protokole DNS i DHCP.

Domenski sustav imena ili skraćeno DNS (engl. Domain Name System) je hijerarhijsko raspoređeni sustav imenovanja za računala, servise ili bilo koje sredstvo spojeno na mrežu. DNS služi za prevođenje lako pamtljivih imena u numeričke IP adrese koje su potrebne za lociranje računalnih servisa i uređaja širom svijeta. Kao takav, DNS je osnovni element funkcionalnosti Interneta.

DHCP (Dynamic Host Configuration Protocol) mrežni je protokol korišten za dodjeljivanje IP adresa i ostalih mrežnih postavki kao što su zadani pristupnik (engl. default gateway), mrežna maska i IP adrese DNS poslužitelja.

* 1. DNS

Dom Domenski sustav imena ili skraćeno DNS (engl. Domain Name System) je hijerarhijsko raspoređeni sustav imenovanja za računala, servise ili bilo koje sredstvo spojeno na mrežu. On povezuje različite informacije s domenskim imenima pripisanim svakom od subjekata u domeni. Ukratko rečeno, DNS služi za prevođenje lako pamtljivih imena u numeričke IP adrese koje su potrebne za lociranje računalnih servisa i uređaja širom svijeta. Kao takav, DNS je osnovni element funkcionalnosti Interneta.

DNS raspoređuje odgovornost za pripisivanje domenskih imena i pridruživanje tih imena IP adresama određivanjem autoritativnih imeničkih poslužitelja za svaku domenu. Autoritativnim imeničkim poslužiteljima pripisana je odgovornost za svoju točno određenu domenu i po redu se mogu pripisati drugi imenički poslužitelji za njihove poddomene.

Glavna zadaća jest prevođenje domenskog imena u IP adresu i obrnuto. Na primjer, domensko ime regoc.srce.hr prevodi u 161.53.2.69 (IPv4) te 2001:b68:c:2::69:0 (IPv6). To se može vidjeti i sljedećom naredbom:

$ host regoc.srce.hr  
regoc.srce.hr has address 161.53.2.69  
regoc.srce.hr has IPv6 address 2001:b68:c:2::69:0

Ovo je takozvani engl. forward lookup. DNS može prevoditi i u obrnutom smjeru (engl. reverse lookup), iz adrese dobiti naziv poslužitelja:

$ host 161.53.2.69  
69.2.53.161.in-addr.arpa domain name pointer regoc.srce.hr

Praksa korištenja imena umjesto brojčane adrese na mreži datira još iz ARPANET ere, s početka osamdesetih godina prošloga stoljeća. Prije no što je 1982. izmišljen DNS, svako mrežno računalo je imalo HOSTS.TXT datoteku koja je mapirala imena u brojčane vrijednosti. Slična datoteka još i danas postoji na većini modernih operativnih sustava i obično sadrži zadani zapis „localhost“ s pridodanom IP adresom 127.0.0.1 (sjetite se datoteke /etc/hosts iz prijašnjih stupnjeva edu4IT).

Ubrzan rast mreže učinio je održavanje HOSTS.TXT datoteke neodrživim te se ukazala potreba za uvođenjem skalabilnijega sustava imenovanja.

DNS su 1983. godine izmislili Paul Mockapetris i Jon Postel, a Radno tijelo za razvoj Interneta (engl. Internet Engineering Task Force) je iste godine objavilo izvorne specifikacije DNS-a u elektronskim dokumentima RFC 882 i RFC 883. Navedeni dokumenti su 1987. godine zamijenjeni s nadopunjenim inačicama RFC 1034 i RFC 1035.

Za prvu implementaciju DNS poslužitelja zaslužna su četvorica studenata sa Sveučilišta Berkeley (Kalifornija, SAD) koji su 1984. godine napisali softver pod nazivom Berkeley Internet Name Domain (BIND).

BIND je danas najčešće korišten softver za DNS na Internetu, a od listopada 2000. je aktualna inačica 9 koja je iz sigurnosnih razloga ponovno napisana od početka. S vremenom su razvijeni i drugi imenički poslužitelji poput PowerDNS-a i Microsoft DNS-a, no BIND se smatra najkompletnijim rješenjem jer je u potpunosti u skladu s IETF standardima. Više o usporedbi DNS poslužitelja pročitajte na sljedećoj poveznici:

<https://en.wikipedia.org/wiki/Comparison_of_DNS_server_software>

S obzirom na to da je BIND standard za DNS na unixoidnim operacijskim sustavima, obrađen je kao tema ovoga tečaja.

Za dobro razumijevanje DNS-a potrebno je shvatiti određene pojmove:

* FQDN – puno ime poslužitelja (engl. Fully Qualified Domain Name)
* Zona – administrativna funkcija pod granularnom kontrolom autoritativnoga DNS poslužitelja
* Resolver – dio DNS poslužitelja koji pretražuje informacije o imenima
* Autoritativni DNS – poslužitelj za određenu zonu
* Primarni DNS – poslužitelj za zone za koje je on autoritativan
* Sekundarni DNS – pričuvni poslužitelj
* Caching-only DNS – poslužitelj koji obrađuje rekurzivne zahtjeve klijenata
* Zone transfer – prijenos informacija o zoni s primarnog na sekundarni DNS poslužitelj
* Delegacija zone – proces kada se kontrola nad zonom prebacuje na drugi imenski poslužitelj.

Osnovna prilagodba

Konfiguracija BIND-a podijeljena je u nekoliko blokova, prikazanih sljedećom tablicom:

|  |  |
| --- | --- |
| **Osnovni blokovi konfiguracije BIND-a** | |
| logging | Postavke dnevničkih zapisa (razine zapisa, putanja do datoteka s zapisima itd.). |
| options | Definicija globalnih postavki (putanja do zonskih datoteka, sučelje za primnje upita itd.). |
| zone | Definicija pojedine zone (ime zone, zonska datoteka, tip poslužitelja) |
| acl | Ime pristupa (engl. Access Control List) |

Primjer sintakse osnovne konfiguracije BIND-a s pripadajućim blokovima:

Primjer named.conf

options {

directory "/var/named";

datasize 100M;

};

zone "." IN {

type hint;

file "named.ca";

};

zone "localhost" IN {

type master;

file "localhost.zone";

allow-update { none; };

};

zone "0.0.127.in-addr.arpa" IN {

type master;

file "named.local";

allow-update { none; };

};

* + 1. Blok logging

Sintaksa bloka logging je sljedeća:

logging {

channel "channel\_name" {

file "file\_name";

versions number\_of\_files;

size log\_size;

syslog < daemon | auth | syslog | authpriv | local0 -to- local7 | null >;

severity ;

print-category yes\_or\_no;

print-severity yes\_or\_no;

print-time yes\_or\_no;

};

category "category\_name" {

"channel\_name";

};

};

Direktiva channel definira gdje će dnevnički zapisi biti poslani (datoteka, syslog ili null). Ako je odabran syslog, moraju biti postavljene vrijednosti log level i facility dnevničkoga zapisa. Direktiva category definira koji tip informacije će se slati u određeni kanal. Bit će lakše objasniti na primjeru.

logging {

channel foo\_channel {

file "LOG";

print-time yes;

print-category yes;

print-severity yes;

};

category "queries" {

"foo\_channel";

};

};

U ovom primjeru je prikazano slanje dnevničkih zapisa u datoteku LOG koja će se kreirati na istoj lokaciji gdje se nalaze i zonske datoteke (na Debianu je zadani radni direktorij /var/cache/bind). Unutar direktive channel definiran je kanal imena foo\_channel, dok je u direktivi category definiran sadržaj namijenjen slanju na određeni kanal. U gornjem primjeru šaljemo kategoriju queries u kanal foo\_channel. Ukratko, ovo znači da će svi klijentski upiti (kategorija queries) biti spremljeni u datoteku LOG (kanal foo\_channel). Popis kategorija je već predifiniran i prikazan je sljedećom tablicom:

|  |  |
| --- | --- |
| **Kategorija** | **Opis** |
| Default | Koristi se ako nije odabran niti jedan kanal |
| General | Sve neklasificirane poruke. |
| database | Poruke vezane za interne zonske datoteke. |
| security | Sigurnosne poruke |
| config | Obrada konfiguracijskih datoteka |
| resolver | Informacije o operacijama izvedenih od strane klijenata |
| xfer-in or xfer-out | Primljene ili poslane zonske datoteke |
| notify | Poruke tipa NOTIFY |
| client | Klijentske aktivnosti |
| update | Ažuriranja zona |
| queries | Klijentski upiti |
| dnssec | Transakcije protokola DNSEC |
| lame-servers | Transakcije poslane s poslužitelja koji su označeni kao lame poslužitelji |

* + 1. Blok options

Globalne postavke DNS poslužitelja navode se u bloku options. Sintaksa toga bloka je sljedeća:

options{

option1;

option2;

....

};

Tih opcija je puno, no sljedećim tablicama pokrivene su najćešće korištene opcije.

|  |  |
| --- | --- |
| **Version** | |
| Informacija o inačici BIND-a koji se koristi. Ako se ne specificira ova opcija, poslužitelj će vratiti pravu brojčanu verziju. Radi sigurnosti dobro je sakriti verziju softvera.  U sljedećem primjeru vratit će poruku “sigurno se šališ”. ☺ | version "(surely you must be joking)"; |

|  |  |
| --- | --- |
| **Directory** | |
| Radni direktorij DNS poslužitelja. Na Debianu je to podrazumno u /var/cache/bind | directory "/var/cache/bind"; |

|  |
| --- |
| **notify (zadano: yes)** |
| Pošalji DNS NOTIFY poruku sekundarnom poslužitelju prilikom promjene zona |

|  |
| --- |
| **recursion (default: yes)** |
| Poslužitelj dopušta rekurzivne upite. Uglavnom se ovo dopušta samo za lokalnu mrežu. Na primjer, ako je ova opcija postavljena na no, DNS poslužitelj neće odgovarati na upite postavljene za zone koje ne poslužuje. |

|  |
| --- |
| **forward (only ili first)** |
| Defaultna vrijednost je first i znači da BIND prvo pita forwardere prije nego što pokuša odgovoriti na upit. Ako je opcija postavljena na only, to znači da će BIND uvijek pitati forwardere za upitom. Ova opcija se koristi zajedno s opcijom forwarders. |

|  |  |
| --- | --- |
| **forwarders (list)** | |
| Popis poslužitelja koji se koriste za proslijeđene upite. Zadana vrijednost je prazan popis. | forwarders { 10.0.0.1; 10.0.0.10;}; |
| **datasize** | |
| Ograničenje veličine predmemorije. | datasize 512M; |

|  |
| --- |
| **allow-query (list)** |
| Popis računala ili mreža koje mogu slati upite BIND poslužitelju. |

|  |
| --- |
| **allow-recursion (list)** |
| Popis računala ili mreža koje mogu slati rekurzivne upite. |

|  |
| --- |
| **allow-transfer (list)** |
| Popis računala (obično su to sekundarni poslužitelji) kojima je dozvoljen transfer zona. |

* + 1. Blok zone

Sintaksa bloka zone je sljedeća:

zone domain\_name {

type zone\_type;

file zone\_file;

local\_options;

};

Umjesto domain\_name potrebno je napisati ime zone koju BIND poslužuje. Za svaku zonu koju poslužuje postoji po jedan ovakav blok.

Zatim se definira tip zone (master, slave, hint):

* master: DNS poslužitelj je primarni za tu zonu
* slave: DNS poslužitelj je sekundarni za tu zonu i transferira je s primarnog poslužitelja
* hint: predefinirane zone u kojima se nalazi popis vršnih DNS poslužitelja.

Zatim se definiraju opcije te zone. Sintaksa je ista kao u bloku options, no ovdje se uglavnom najčešće koriste sljedeće opcije:

* allow-transfer: popis računala kojima se dopušta transfer zone
* allow-query: popis računala kojima se dopušta da postavlja upite za tu zonu
* masters: popis primarnih poslužitelja nadležnih za tu zonu.

Unutar opcije file definira se datoteka u kojoj se nalazi određena zona. Tu se može postaviti apsolutna putanja do datoteka, ili relativna (pa će se koristiti direktorij definiran unutar opcije directory unutar bloka options).

Slijedi primjer dviju zona definiranih na primarnom poslužitelju, s dopuštanjem transfera zone na sekundarni poslužitelj na adresi 10.1.2.3:

zone seafront.bar {

type master;

file "seafront.zone";

allow-transfer{10.1.2.3;);

};

zone 2.1.10.in-addr.arpa {

type master;

file "10.1.2.zone"

allow-transfer{10.1.2.3;);

};

Sljedeći primjer prikazuje definiciju dviju zona na sekundarnom poslužitelju, s time da se transfer zone radi s primarnog poslužitelja na adresi 10.1.2.1.:

zone "seafront.bar" IN {

type slave;

masters {10.1.2.1;};

file "slave/seafront.zone";

};

zone "2.1.10.in-addr.arpa" IN {

type slave;

masters {10.1.2.1;};

file "slave/10.1.2.local";

};

* + 1. BIND na Debianu

DNS poslužitelj BIND instalira se sljedećom naredbom (za instalaciju su potrebne *root* ovlasti):

# apt-get install bind9

Konfiguracijske datoteke nalaze se u direktoriju **/etc/bind**. Glavna konfiguracijska datoteka je **named.conf** te je njezin sadržaj na *Debianu* sljedeći:

# cat /etc/bind/named.conf

// This is the primary configuration file for the BIND DNS server named.

//

// Please read /usr/share/doc/bind9/README.Debian.gz for information on the

// structure of BIND configuration files in Debian, \*BEFORE\* you customize

// this configuration file.

//

// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";

include "/etc/bind/named.conf.local";

include "/etc/bind/named.conf.default-zones";

Konfiguracija BIND-a na Debian OS-u podijeljena je na nekoliko manjih datoteka (slično kao npr. Apache2) koje su uključene u datoteci named.conf. Blokovi options, logging te acl se postavljaju u datoteci named.conf.options, dok se zone postavljaju u named.conf.local. U datoteci named.conf.default-zones definirani su vršni poslužitelji i oni se uglavnom ne mijenjaju.

Primarni i sekundarni DNS servis

Kako bi se optimalno raspodijelio globalni DNS promet, danas postoji 13 vršnih DNS poslužitelja. Pored toga je važno i da se DNS podjelom na tzv. zone omogućuje hijerarhijsko raspodjeljivanje administrativnih ovlasti potrebnih za održavanje informacija. Svaki pružatelj internetskih usluga ima DNS poslužitelje (najmanje 2) za svoje korisnike. DNS stablo je podijeljeno u zone. Zona je skup čvorova u DNS stablu upravljanih od strane jednoga DNS poslužitelja. Jedan DNS poslužitelj može administrirati više zona, tj. može biti autoritativni imenski poslužitelj za više domena.

Zbog toga dolazimo do potrebe da svaku zonu poslužuju minimalno dva DNS poslužitelja. Jedan je primarni poslužitelji (ili engl. master), a svi ostali su sekundarni (engl. slave). Zona se ažurira na primarnom poslužitelju, a te se izmjene automatski prenose na jedan ili više sekundarnih poslužitelja.

Zone i kontrole pristupa

DNS poslužitelj podatke o zonama može čuvati u datotekama ili u bazi podataka. Obično se s BIND-om koriste datoteke pa će one biti obrađene u ovom poglavlju.

Zapis izvora (engl. resource record, RR) je osnovni podatkovni element u sustavu domenskih imena. Svaki zapis ima svoj tip (A, MX, NS, PTR...), vrijeme isteka vrijednosti, klasu te neki tip specifičnih podataka za taj tip podatka. Svaka zona sadrži više zapisa izvora. Sljedeća tablica prikazuje najčešće korištene zapise izvora.

|  |  |
| --- | --- |
| **RR** | **Opis** |
| A | Dohvaća se 32-bitna (IPv4) adresa iz imena poslužitelja. |
| AAAA | Dohvaća se 128-bitna (IPv6) adresa iz imena poslužitelja. |
| CNAME | Alias na drugo ime poslužitelja. Na primjer, www.srce.hr pokazuje na marun.srce.hr (zapis tipa CNAME), dok marun.srce.hr pokazuje na 161.53.2.88 (zapis tipa A). |
| PTR | Dohvaća se ime poslužitelja iz dane IP adrese. |
| MX | Dohvaća se poslužitelj za elektroničku poštu zadužen za tu domenu. Može biti jedan ili više takvih poslužitelja s različitim prioritetima. MX je kratica od Mail eXchanger. |
| NS | Delegira određenu zonu drugim DNS poslužiteljima. |

**Izrada i održavanje zonskih datoteka**

Format zonskih datoteka definiran je u dokumentu RFC 1035 i sadrži zapise izvora (RR) za održavanu domenu ili poddomenu.

**1 – Start Of Authority (SOA)**

Na početku svake zonske datoteke nalazi se zapis SOA (kratica od Start of Authority). Sintaksa toga zapisa je sljedeća:

root-name TTL IN SOA name-server email-address (

serial number;

refresh;

retry;

expire;

minimum; )

Zapis SOA sadrži sljedeće informacije:

* primarni poslužitelj za tu zonu
* odgovorna osoba za tu zonu
* serijski broj zone (obično u obliku YYYYMMDDXX gdje YYYY predstavlja godinu, MM mjesec, DD dan, a XX broj izmjene u tom danu); taj zapis mora biti povećan kod svake promjene zone
* broj sekundi prije nego zona treba biti osvježena
* broj sekundi prije ponovnog pokušaja neuspjelog osvježavanja
* gornja granica u sekundama prije nego se smatra da zona više nije autoritativna

Primjer jednoga zapisa SOA:

$TTL 86400

@ 1D IN SOA ns.seafront.bar. root.seafront.bar. (

2018101001 ; serial (d. adams)

1H ; refresh

15M ; retry

1W ; expiry

1D ) ; minimum

**2 – Zapisi koji definiraju DNS poslužitelje za tu zonu**

*domain-name IN NS name-server*

Primjer:

IN NS ns1.seafront.bar.

IN NS ns2.seafront.bar.

Na kraju punog imena računala potrebno je pisati točku. Ako se koristi kratko ime (npr. samo ns1 umjesto ns1.seafront.bar), BIND će automatski dodati naziv te zone (seafront.bar). Znači, ako se zapiše ns1.seafront.bar, BIND će smatrati da je ime DNS poslužitelja ns1.seafront.bar.seafront.bar.

**3 – Zapisi koji definiraju nadležne mail poslužitelje za tu domenu**

domain-name IN MX PRI mail-server

PRI znači prioritet, jer može biti više MX zapisa. Ako je definirano više mail poslužitelja za jednu domenu, prvo će se koristiti onaj s manjom vrijednosti broja PRI.

Primjer:

IN MX 5 mail.seafront.bar.

**4 – Autoritativne informacije o računalima unutar zone**

*host-name* IN A IP-address

Zapis tipa A vraća IP adresu za zadano ime. Znači, u sljedećem primjeru BIND će vratiti IP adresu 192.168.21.254 za upit ns1.seafront.bar:

ns1 IN A 192.168.21.254

Ako je zona reverzna, onda se koriste zapisi tipa PTR koji vraćaju ime iz zadane IP adrese.

n IN PTR host-name

U sljedećem primjeru se radi o zoni 21.168.192.in-addr.arpa i zbog sljedećeg zapisa će BIND vratiti vrijednost ns1.seafront.bar za zadanu IP adresu 192.168.21.254.

254 IN PTR ns1.seafront.bar.

DNSSEC

Domain Name System Security Extensions ili kraće DNSSEC je protokol za sigurnost na Internetu koji je osmišljen radi otkrivanja i zaustavljanja presretanja i promjene podataka na autoritativnim poslužiteljima. DNSSEC osigurava korisniku Interneta vjerodostojnost da web-stranica koju posjećuje uistinu odgovara onoj koja se nalazi na adresi koju je upisao u web-preglednik.

Uvođenje protokola DNSSEC je dugoročna strategija osiguranja više razine internetske sigurnosti, a njegovim uvođenjem ne štite se samo korisnici i njihovi podaci već se pomaže u izgradnji globalno sigurnijega sustava.

DNS dodaje komponentu sigurnosti DNS poslužiteljima i trenutačno obavlja tri funkcije:

* siguran prijenos zone
* sigurno ažuriranje zapisa zone
* očuvanje integriteta zone.

DNSSEC pokušava riješiti ranjivosti koje se pojavljuju tijekom neovlaštenih dinamičkih ažuriranja zona, kao i lažno predstavljanje. To uključuje provjeru autentičnosti između primarnog i sekundarnog poslužitelja.

S poslužiteljem BIND dolazi naredba dnssec-keygen koja se koristi za generiranje ključa na primarnom poslužitelju koji se sigurnim kanalom premjesti na sekundarni poslužitelj. Putem toga ključa obavlja se provjera autentičnosti. Taj autentikacijski mehanizam se zove TSIG (Transaction Signature).

**Konfiguracija primarnoga poslužitelja**

1. Prvo se sljedećom naredbom generira ključ na primarnom poslužitelju koji se zove seafront.bar:

dnssec-keygen –a HMAC-MD5 -b 256 -n host seafront.bar

Ta će naredba kreirati par ključeva (privatni i javni):

Kseafront.bar.+157+49196.key

Kseafront.bar.+157+49196.private

Sadržaj datoteke Kseafront.bar.+157+49196.key izgleda kao u sljedećem primjeru:

seafront.bar. IN KEY 512 3 157 QN3vIApnV76WS+a2Hr3qj+AqZjpuPjQgVWeeMMGSBC4=

2. U direktoriju /etc/bind/ kreira se datoteka slave.key sa sljedećim sadržajem:

key "seafront.bar." {

algorithm hmac-md5;

secret "QN3vIApnV76WS+a2Hr3qj+AqZjpuPjQgVWeeMMGSBC4=";

};

3. U konfiguracijsku datoteku named.conf potrebno je dodati:

include "/etc/slave.key";

zone "seafront.bar" IN {

type master;

file "seafront.zone";

allow-transfer { key seafront.bar.; };

};

zone 2.1.10.in-addr.arpa {

type master;

file "10.1.2.zone"

allow-transfer { key seafront.bar.; );

};

Time smo osigurali da je transfer zone dopušten samo računalima koja imaju ključ seafront.bar.

**Konfiguracija sekundarnoga poslužitelja**

Kopira se sadržaj datoteke slave.key na sekundarni poslužitelj u direktorij /etc/bind. Zatim se u named.conf doda sljedeći blok:

server *10.1.2.1* {

keys {seafront.bar.;};

};

include "/etc/bind/slave.key";

Time smo osigurali da sekundarni poslužitelj radi transfer zone kroz siguran kanal, korištenjem para ključeva i protokola DNSSEC.

DNSSEC također osigurava integritet podataka i njihovu autentičnost. Ako želimo potpisati određenu zonu, to se može sljedećim naredbama.

1. Prvo je potrebno kreirati par ključeva za potpis zone s naredbom:

dnssec-keygen -a DSA -b 1024 -n zone seafront.bar.

To će izgenerirati dvije datoteke:

Kseafront.bar.+003+31173.key

Kseafront.bar.+003+31173.private

2. Zatim je potrebno unijeti javni ključ na kraj nepotpisane zonske datoteke naredbom:

cat Kseafront.bar.+003+31173.key >> seafront.bar

3. I na kraju se ta zonska datoteka može potpisati korištenjem naredbe:

dnssec-signzone -o seafront.bar Kseafront.bar.+003+31173

Ta će naredba izgenerirati potpisanu zonsku datoteku naziva seafront.bar.signed. Ako želimo koristiti tu potpisanu zonsku datoteku, potrebno je u konfiguracijskoj datoteci ***named.conf*** promijeniti putanju do datoteke ***seafront.bar.signed***.

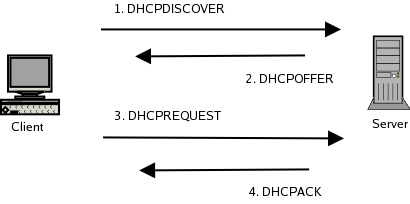
* 1. DHCP

Protokol DHCP

DHCP (Dynamic Host Configuration Protocol) mrežni je protokol korišten za dodjeljivanje IP adresa i ostalih mrežnih postavki kao što su zadani pristupnik (engl. default gateway), mrežna maska i IP adrese DNS poslužitelja. Olakšava konfiguraciju mreže jer eliminira ručno dodavanje osnovnih postavki za jednu računalnu mrežu. DHCP klijent se brine da su dodijeljene IP adrese ispravne i da u mreži nema dupliciranih adresa.

DHCP je prihvaćen kao standardni protokol u listopadu 1993. godine (RFC 1531), a posljednja definicija datira iz ožujka 1997. godine (RFC 2131).

Osnovna komunikacija procesa dobivanja IP adrese od DHCP poslužitelja prikazana je sljedećom slikom:



Kada se neko računalo poveže na mrežu, DHCP klijent šalje zahtjev za potrebnim mrežnim parametrima. DHCP poslužitelj ima određeni raspon adresa koje može dodijeliti te šalje dodatne mrežne parametre (maska, DNS poslužitelj, NTP poslužitelj i sl.). Na velikim mrežama jedan DHCP server može posluživati više podmreža uz pomoć DHCP agenata koji se nalaze na usmjernicima te komuniciraju s poslužiteljem i klijentima. Klijent može zatražiti od poslužitelja da mu uvijek dodjeljuje iste parametre, ali ne znači da će ih uvijek i dobiti nazad.

DHCP protokol se sastoji od 4 osnovne i 2 dodatne poruke koje klijent i poslužitelj izmjenjuju:

* DHCPDISCOVER (otkriće)
* DHCPOFFER (ponuda)
* DHCPREQUEST (zahtjev)
* DHCPACK (potvrda)
* DHCPINFORM (informacije)
* DHCPRELEASE (oslobađanje)

U prvom koraku klijentsko računalo šalje zahtjev DHCPDISCOVER na adresu razašiljanja (engl. broadcast) 255.255.255.255 i očekuje odgovor.

DHCP poslužitelj može dodijeliti IP adresu na dva načina:

* dinamička adresa dobivena iz određenog raspona IP adresa definiranih na DHCP poslužitelju
* fiksna adresa za određeno računalo (koje se utvrđuje putem MAC adrese).

Autoritativni DHCP poslužitelj takav zahtjev prihvaća ako klijent nije promijenio mrežu na kojoj se nalazi, u protivnom ju odbija. Neautoritativni DHCP poslužitelj takav zahtjev ignorira i dolazi do zastarijevanja zahtjeva pa je klijent primoran poslati novi zahtjev. Kada DHCP poslužitelj zaprimi poruku DHCPDISCOVER zahtjev za IP adresom, on rezervira jednu IP adresu za klijenta i šalje poruku DHCPOFFER koja sadrži: klijentovu MAC adresu, IP adresu koju poslužitelj nudi, oznaku maske, vrijeme valjanosti IP adrese i IP adresu DHCP poslužitelja koji je dao ponudu.

Nakon primitka povratne poruke klijent šalje novu poruku DHCPREQUEST kojom zahtjeva da mu se dodijeli ponuđena adresa. Klijent može primiti poruke DHCPOFFER od više DHCP poslužitelja odjednom, ali adresu smije zatražiti samo od jednog poslužitelja. Ako poslužitelj ne dobije poruku DHCPREQUEST, on ponuđenu adresu vraća u skup raspoloživih adresa.

Nakon primanja poruke DHCPREQUEST, poslužitelj šalje poruku DHCPPACK koja sadrži vrijeme valjanosti IP adrese i druge konfiguracijske parametre. Time je faza potvrde završila. Kada klijent primi poruku s konfiguracijskim paketom te izvrši konfiguraciju mrežnih postavki, on šalje poruku ARP (Address Resolution Protocol) cijeloj mreži kako bi svi mogli osvježiti svoje ARP tablice.

Klijent može zatražiti više informacija nego što ih dobije putem poruka DHCPOFFER i DHCPPACK. Na primjer, mogu se zatražiti postavke za različite aplikacije i servise (npr. web-preglednici koriste DHCPINFORM kako bi saznali postavke proxy poslužitelja). Kada je klijent gotov s korištenjem IP adrese koju je zaprimio od DHCP poslužitelja, on šalje poruku DHCPRELEASE i time vraća IP adresu

Servis dhcpd

DHCP poslužitelj na Debian OS-u nalazi se u paketu isc-dhcp-server. Da bi ga mogli koristiti, potrebno je instalirati navedeni paket korištenjem sljedeće naredbe:

# apt-get install isc-dhcp-server

Konfiguracijska datoteka DHCP poslužitelja nalazi se u datoteci /etc/dhcp/dhcpd.conf.

Jedan DHCP poslužitelj može posluživati adrese u više različitih mrežnih segmenata. Zato je potrebno definirati zasebne blokove u konfiguracijskoj datoteci za pojedinu mrežu. Na početku konfiguracijske datoteke nalaze se globalne opcije, a zatim slijede blokovi u kojima su definirani mrežni parametri za svaku pojedinu mrežu.

Primjer jednoga mrežnog bloka prikazan je sljedećim primjerom:

subnet 192.168.1.0 netmask 255.255.255.0 {

range 192.168.1.10 192.168.1.200;

host server1.tecaj.hr {

hardware ethernet 00:80:C6:30:0A:7E;

Fixed-address 192.168.1.2;

}

}

Mrežni blok prikazan ovim primjerom nalazi se unutar mreže 192.168.1.0/24, a DHCP poslužitelj će dinamički dodijeljivati klijentskim računalima IP adrese u rasponu od 192.168.1.10 do 192.168.1.100. Računalo server1.tecaj.hr će imati fiksnu adresu 192.168.1.2 temeljem svoje MAC adrese. U svakom mrežnom bloku mogu biti informacije kao na primjer: • zadani pristupnik (engl. default gateway) • DNS poslužitelji • domena. To se može dodati na sljedeći način u svakom bloku:

option routers 192.168.1.1;

option nis-domain "nisdomain";

option domain-name "seafront.bar";

option domain-name-servers 161.53.2.69;

Baza podataka zauzetih dinamičkih adresa nalazi se u datoteci /var/lib/dhcp/dhcpd.leases. Ta datoteka ne bi trebala biti ručno mijenjana. Tu se nalaze informacije o dužini trajanja dodijeljene adrese, koje su adrese dodijeljene, početak i kraj zauzeća adrese, MAC adresa.

Ako DHCP poslužitelj ima više mrežnih sučelja, potrebno je u datoteku /etc/default/isc-dhcp-server dodati imena mrežnih sučelja na kojima dhcpd osluškuje zahtjeve. To se radi u varijabli INTERFACESv4, kao što se vidi u sljedećem primjeru:

INTERFACESv4=eth0

Dinamični DNS

Moguće je postaviti da DHCP poslužitelj automatski obnovi DNS s adresom klijentskoga računala koje se spojilo na mrežu. To se radi korištenjem para ključeva objašnjenih u prethodnom poglavlju o DNS-u.

Na DHCP poslužitelju je potrebno u datoteci dhcpd.conf dodati podatke o ključu i zonama koje DHCP poslužitelj može osvježiti. Za naš slučaj s domenom seafront.bar to se može na sljedeći način:

ddns-update-style interim;

ignore client-updates;

key seafront.bar. {

algorithm hmac-md5;

secret QN3vIApnV76WS+a2Hr3qj+AqZjpuPjQgVWeeMMGSBC4=;

};

zone seafront.bar. {

primary 192.168.3.100;

key seafront.bar.;

}

zone 3.168.192.in-addr.arpa. {

primary 192.168.3.100;

key seafront.bar.;

}

Na DNS poslužitelju je potrebno napraviti sljedeće: 1. Ako se koriste potpisane zonske datoteke putem DNSSEC-a, potrebno je koristiti zonske datotekeke koje nisu potpisane. 2. Treba dodati opciju allow-update da se dopusti osvježavanje zone putem ključa seafront.bar:

zone "seafront.bar" IN {

type master;

file "seafront.zone";

allow-update { key seafront.bar.; };

allow-transfer { key seafront.bar.; };

};

Isto se može napraviti s reverznom zonom:

zone "3.168.192.in-addr.arpa" IN {

type master;

file "192.168.3.local";

allow-update { key seafront.bar.; };

allow-transfer { key seafront.bar.;};

};

Potrebno je ponovno pokrenuti DHCP i DNS servise i kod sljedeće dodjele IP adresa, DHCP poslužitelj će osvježiti ime klijentskoga računala i pripadajuću adresu u DNS-u.

DHCP relej

Poruka DHCPDISCOVER od klijenta do poslužitelja dolazi kroz adresu razašiljanja 255.255.255.255, no često je ta adresa razašiljanja blokirana na usmjernicima.

Ako postoji više mreža i jedan DHCP poslužitelj, svaki usmjernik treba biti u mogućnosti proslijediti zahtjeve DHCPDISCOVER s adrese razašiljanja (engl. broadcast) u toj mreži na odgovarajući DHCP poslužitelj. Ako je usmjernik računalo s Linuxom, to je moguće napraviti s alatom dhcrelay koji se nalazi u paketu isc-dhcp-relay.

* 1. Vježba 1. Osnovna konfiguracija servisa BIND

1. Na sustavu **server1** instalirajte pakete bind9 i dnsutils koristeći naredbu apt-get:

apt-get install bind9 dnsutils

Unutar paketa **bind9** nalazi se servis **named**, dok se unutar paketa **dnsutils** nalaze naredbe poput **nslookup**, **dig** ili **nsupdate**.

1. Pogledajte sadržaj direktorija **/etc/bind/**. Primijetite da je datoteka **named.conf** razlomljena na više datoteka po sekcijama.

Datoteka **named.conf** je glavna datoteka i uključuje sve ostale.

Datoteka **named.conf.options** sadrži sekciju **options** u kojoj se definiraju glavne postavke servisa named.

Datoteka **named.conf.default-zones** sadrži popis podrazumnih zona koje svaki DNS poslužitelja mora imati, kao npr. zona s popisom svih korijenskih poslužitelja.

Datoteka **named.conf.local** sadrži popis zona koje servira taj DNS poslužitelj.

1. U datoteci **/etc/bind/named.conf.options** unutar sekcije options upišite:

forward only;

forwarders {

161.53.2.69;

161.53.2.70;

};

1. Zakomentirajte redak listen-on-v6 tako da bind ne osluškuje zahtjeve na IPv6 sučelju.
2. Pokrenite naredbu named-checkconf s kojom provjeravate da li postoje greške unutar konfiguracijskih datoteka.
3. Restarajte servis bind9 koristeći naredbu:

service bind9 restart

1. Provjerite radi li servis named ispravno koristeći naredbe iz paketa dnsutils:

dig @127.0.0.1 www.srce.hr

nslookup www.srce.hr 127.0.0.1

**Vježba 1: BIND**

**Definiranje zona**

1. Dodajte sljedeću konfiguracijuna kraj datoteke **/etc/bind/named.conf.local**:

zone “tecaj.hr” IN {

type master;

file “zones/tecaj.hr”;

};

zone “1.168.192.in-addr.arpa” IN {

type master;

file “zones/1.168.192.in-addr.arpa”;

};

zone “2.168.192.in-addr.arpa” IN {

type master;

file “zones/2.168.192.in-addr.arpa”;

};

1. Promijenite forward only; u forward first; u datoteci **/etc/bind/named.conf.options**.
2. Stvorite direktorij u koji će se upisivati zonske datoteke:

mkdir /var/cache/bind/zones

chown bind:bind /var/cache/bind/zones

chmod 770 /var/cache/bind/zones

1. U direktoriju **/var/cache/bind/zones** stvorite datoteke **tecaj.hr**, **1.168.192.in-addr.arpa i 2.168.192.in-addr.arpa**.

Serijski broj u datotekama izgleda kao datum i redni broj izmjene u određenom danu. Serijski broj zone je obično u obliku YYYYMMDDXX gdje YYYY predstavlja godinu, MM mjesec, DD dan, a XX broj izmjene u tom danu. Taj zapis MORA biti povećan kod svake promjene zone. Slobodno promijenite u današnji datum.

Neka datoteka **tecaj.hr** ima sljedeći sadržaj:

$TTL 1D

@ IN SOA tecaj.hr. root.tecaj.hr. (

2020011701 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

NS server1.tecaj.hr.

localhost A 127.0.0.1

server1 A 192.168.1.100

A 192.168.2.201

server2 A 192.168.2.202

Neka datoteka **1.168.192.in-addr.arpa** ima sljedeći sadržaj:

$TTL 1D

@ IN SOA tecaj.hr. root.tecaj.hr. (

2020011701 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

NS server1.tecaj.hr.

100 PTR server1.tecaj.hr.

Neka datoteka **2.168.192.in-addr.arpa** ima sljedeći sadržaj:

$TTL 1D

@ IN SOA tecaj.hr. root.tecaj.hr. (

2020011701 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

NS server1.tecaj.hr.

201 PTR server1.tecaj.hr.

202 PTR server2.tecaj.hr.

Provjerite s naredbom **named-checkconf** postoje li greške u konfiguracijskim datotekama.

Zatim, provjerite s naredbom **named-checkzone** postoje li greške u zonama:

named-checkzone tecaj.hr \ /var/cache/bind/zones/tecaj.hr

named-checkzone 1.168.192.in-addr.arpa \ /var/cache/bind/zones/1.168.192.in-addr.arpa

named-checkzone 2.168.192.in-addr.arpa \ /var/cache/bind/zones/2.168.192.in-addr.arpa

1. Ponovno učitajte konfiguraciju servisa named koristeći naredbu:

service bind9 reload

Provjerite je li sve u redu:

host server1.tecaj.hr 127.0.0.1

1. U datoteci **named.conf.options** unutar sekcije options izmijenite postojeći ili dodajte novi (ako ne postoji) redak s direktivom allow-query tako da piše:

allow-query { 192.168.1.0/24; 192.168.2.0/24; localhost; };

**Vježba: Postavke na klijentima**

1. Na sustavu **server1** izmijenite datoteku **/etc/resolv.conf** tako da njezin sadržaj bude:

search tecaj.hr

nameserver 192.168.1.100

nameserver 192.168.2.201

1. Na sustavu **server2** izmijenite datoteku **/etc/resolv.conf** tako da njezin sadržaj bude:

search tecaj.hr

nameserver 192.168.2.201

Provjerite radi li naredba nslookup server1.

**Vježba: Sekundarni DNS**

1. Na sustavu **server1** izmijenite konfiguracijsku datoteku named.conf.options tako da u sekciju options dodate:

allow-transfer { 192.168.2.202; };

U definicije svih prethodno kreiranih zona u direktoriju /var/cache/bind/zones/ dodajte **server2.tecaj.hr** kao dodatni DNS poslužitelj, a serijski broj povećajte za jedan.

Na primjer, zaglavlje zone tecaj.hr izgledat će ovako:

@ IN SOA tecaj.hr. root.tecaj.hr. (

20200117012 ; serial

1D ; refresh

1H ; retry

1W ; expire

3H ) ; minimum

NS server1.tecaj.hr.

NS server2.tecaj.hr.

Ponovno pokrenite named koristeći naredbu:

service bind9 restart

1. Na sustavu **server2** instalirajte paket bind9.
2. Na sustavu **server2** izmijenite datoteku **/etc/bind/named.conf.options** tako da u sekciji **options** dodate:

listen-on port 53 { any; };

forward first;

forwarders {

192.168.2.201;

};

allow-query {

192.168.1.0/24;

192.168.2.0/24;

localhost;

};

1. Na kraj datoteke **/etc/bind/named.conf.local** dodajte podatke o zonama:

zone "tecaj.hr" IN {

type slave;

masters { 192.168.2.201; };

file "slaves/tecaj.hr";

};

zone "1.168.192.in-addr.arpa" IN {

type slave;

masters { 192.168.2.201; };

file "slaves/1.168.192.in-addr.arpa";

};

zone "2.168.192.in-addr.arpa" IN {

type slave;

masters { 192.168.2.201; };

file "slaves/2.168.192.in-addr.arpa";

};

1. Provjerite postoji li direktorij **/var/cache/bind/slaves**. Ako ne postoji napravite ga i promijenite vlasnika i grupu na bind:bind. Ako postoji i nije prazan, obrišite njegov sadržaj.

Naredbom named-checkconf provjerite postoje li greške u konfiguracijskim datotekama i nakon toga ponovno pokrenite named:

service bind9 restart

1. Provjerite s **nslookup server1.tecaj.hr localhost** je li sve u redu.
2. Proučite sadržaj direktorija **/var/cache/bind/slaves**. Što se dogodilo?
3. Na sustavu **server2** izmijenite sadržaj datoteke **/etc/resolv.conf** tako da bude:

search tecaj.hr

nameserver 192.168.2.201

nameserver 192.168.2.202

**Vježba: DNSSEC**

1. Na sustavu **server1** (u nekom direktoriju koji nije dostupan ostalim korisnicima, na primjer u home-direktoriju ili direktoriju **/var/cache/bind/zones**) pokrenite naredbu:

dnssec-keygen -a HMAC-MD5 -b 256 -n host tecajhr

U radnom direktoriju stvorit će se dvije nove datoteke, na primjer:

Ktecajhr.+157+19014.key

Ktecajhr.+157+19014.private

Primjer sadržaja datoteke key-datoteke (Ktecajhr.+157+19014.key):

tecaj.hr. tecajhr. IN KEY 512 3 157 gruXti7lOtAfqRA7DKCj6P73f8tLS7NQTBRRhOae3n0=

Primjer sadržaja druge datoteke (Ktecajhr.+157+19014.private):

Private-key-format: v1.3

Algorithm: 157 (HMAC\_MD5)

Key: gruXti7lOtAfqRA7DKCj6P73f8tLS7NQTBRRhOae3n0=

Bits: AAA=

Created: 20200120213445

Publish: 20200120213445

Activate: 20200120213445

U ovom primjeru konkretna vrijednost generiranog ključa je: gruXti7lOtAfqRA7DKCj6P73f8tLS7NQTBRRhOae3n0=

**Napomena**

U obje je datoteke vrijednost ključa ista jer je riječ o simetričnom algoritmu za kriptiranje. Vrijednost ključa je tajni podatak koji ne bi trebao biti dostupan ostalim korisnicima na sustavu.

1. Na sustavu **server1** stvorite datoteku **/etc/bind/slave.key** sljedećeg sadržaja, pri tome pripazite na vrijednost ključa koja mora biti ista kao ovaj koji ste upravo stvorili:

key "tecajhr" {

algorithm hmac-md5;

secret "gruXti7lOtAfqRA7DKCj6P73f8tLS7NQTBRRhOae3n0=";

};

Postavite odgovarajuće vlasništvo i prava pristupa datoteci:

chown root:bind /etc/bind/slave.key

chmod 640 /etc/bind/slave.key

1. Na kraj datoteke **/etc/bind/named.conf** dodajte redak:

include "/etc/bind/slave.key";

1. U sekciju koja opisuje zonu dodajte:

allow-transfer { key tecajhr; };

1. Provjerite da u sekciji **options** piše:

dnssec-enable yes;

dnssec-validation no;

1. Provjerite s naredbom named-checkconf postoje li greške u **datoteci named.conf**.
2. Ponovno pokrenite named: service bind9 restart
3. Na sustavu **server2** zaustavite named (service bind9 stop) i obrišite podatke u direktoriju **/var/cache/bind/slaves** (na primjer: **rm /var/cache/bind/slaves/\***).
4. Ponovno pokrenite named: service bind9 start
5. Provjerite sadržaj direktorija **/var/cache/bind/slaves**. Koje su se zone kopirale? Nedostaje li koja?

**Napomena**

U produkcijskoj okolini bi sve tri zone prebacili na DNSSEC, no zbog potreba ove vježbe prebačena je samo zona tecaj.hr.

1. Kopirajte datoteku **/etc/bind/slave.key** sa sustava **server1** na sustav **server2** (na primjer, naredbom **scp** ili neposredno kopirajući tekst (copy&paste) iz jednog u drugi terminalski prozor).
2. Postavite datoteci **/etc/bind/slave.key** (na sustavu **server2**) odgovarajuća prava pristupa:

chown root:bind /etc/bind/slave.key

chmod 640 /etc/bind/slave.key

1. U datoteku **/etc/bind/named.conf** na sustavu **server2** dodajte sljedeće retke:

include "/etc/bind/slave.key";

server 192.168.2.201 {

keys tecajhr;

};

1. Ponovno pokrenite named.
2. Pogledajte sadržaj direktorija **/var/cache/bind/slaves**. Što se promijenilo? Da li se pojavila zona koja se prenosi DNSSEC-om?
3. Potražite redak s ključnom riječju TSIG u datoteci **/var/log/daemon.log**.
   1. Vježba 2. DHCP

**Osnovne postavke DHCP-poslužitelja**

1. Na sustavu **server1** instalirajte paket isc-dhcp-server:

apt-get install isc-dhcp-server

1. Izmijenite konfiguracijsku datoteku za DHCP-poslužitelj (**/etc/dhcp/dhcpd.conf**) tako da ima sljedeći sadržaj:

subnet 192.168.2.0 netmask 255.255.255.0 {

option routers 192.168.2.201;

option subnet-mask 255.255.255.0;

option domain-name "tecaj.hr";

option domain-name-servers 192.168.2.201;

range 192.168.2.220 192.168.2.240;

}

1. U datoteci **/etc/default/isc-dhcp-server** u retku koji počinje s **INTERFACEv4** unutar navodnika dodajte mrežno sučelje koje je u mreži 192.168.2.0/24 (to je u našem slučaju eth1).
2. Provjerite postoje li greške u konfiguracijskoj datoteci DHCP-poslužitelja:

dhcpd configtest

1. Ponovno pokrenite dhcpd:

service dhcpd restart

1. Proučite sadržaj datoteke **/var/lib/dhcp/dhcpd.leases**.

**Osnovne postavke DHCP-klijenta**

1. Na sustavu **server2** preuzmite adresu mrežnog sučelja eth0 putem DHCP-a koristeći naredbu dhclient:

dhclient eth0

1. Ponovno pogledajte sadržaj datoteke **/var/lib/dhcpd/dhcpd.leases** na sustavu **server1**.
2. Pokrenite naredbu ping server2 (na bilo kojem od sustava). Što se dogodilo i zašto?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Vratite prethodne postavke mrežnog sučelja sučelja eth0:

ifdown eth0

ifup eth0

**Dynamic DNS (DDNS)**

1. Na sustavu **server1** stvorite HMAC-MD5-keyfile s imenom dhcpupdate:

dnssec-keygen -a HMAC-MD5 -b 256 -n USER \ dhcpupdate

U radnom direktoriju pojavit će se dvije nove datoteke, na primjer: **Kdhcpupdate.+157+15013.key** i **Kdhcpupdate.+157+15013.private**.

Na primjer, neka je sadržaj datoteke **dhcpupdate.+157+15013.key**:

dhcpupdate. IN KEY 0 3 157 u05IlARIf5Dv0YsxjbAB04O+ZDvnEUi+DUgY6npl7GM=

U gornjem je primjeru „u05IlARIf5Dv0YsxjbAB04O+ZDvnEUi+DUgY6npl7GM=“ vrijednost novog ključa.

1. Stvorite datoteku **/etc/dhcpupdate.key** sa sljedećim sadržajem:

key "dhcpupdate" {

algorithm hmac-md5;

secret "u05IlARIf5Dv0YsxjbAB04O+ZDvnEUi+DUgY6npl7GM=";

};

(Umjesto „u05IlARIf5Dv0YsxjbAB04O+ZDvnEUi+DUgY6npl7GM=“ upišite vrijednost stvorenog ključa.)

1. Postavite odgovarajuće postavke u datoteci **/etc/dhcpupdate.key** te je kopirajte kao **/etc/dhcp/dhcpupdate.key** (potrebno je napraviti dvije datoteke s istim sadržajem zbog različitih prava pristupa):

chown root:bind /etc/dhcpupdate.key

chmod 640 /etc/dhcpupdate.key

cp /etc/dhcpupdate.key /etc/dhcp/dhcpupdate.key

chown root:root /etc/dhcp/dhcpupdate.key

chmod 640 /etc/dhcp/dhcpupdate.key

1. Na sustavu **server1** u datoteku **/etc/bind/named.conf** dodajte redak:

include "/etc/dhcpupdate.key";

1. U zone za koje želite da se dinamički mijenjaju dodajte:

allow-update { key dhcpupdate; };

U našem slučaju to su zone **tecaj.hr** i **2.168.192.in-addr.arpa**.

Iz zona **tecaj.hr** i **2.168.192.in-addr.arpa** (datoteke **tecaj.hr** i **2.168.192.in-addr.arpa** u direktoriju **/var/cache/bind/zones**) obrišite sve podatke za **server2** (uključivši i one koje ga definiraju kao sekundarnog DNS poslužitelja).

**Napomena**

Obzirom da sustav server2 neće imati stalnu IP-adresu, nije prikladno da bude sekundarni DNS poslužitelj. Također, njegove smo podatke u potpunosti izbrisali iz DNS tablica, jer će ubuduće te podatke upisivati DHCP-poslužitelj.

1. Ponovno pokrenite named.
2. Pomoću naredbe **nsupdate** provjerite je li sve u redu:

nsupdate

> server 192.168.1.100

> key dhcpupdate u05IlARIf5Dv0YsxjbAB04O+ZDvnEUi+DUgY6npl7GM=

> zone tecaj.hr.

> update add server5.tecaj.hr. 600 IN A 192.168.2.205

> send

> zone 2.168.192.in-addr.arpa.

> update add 205.2.168.192.in-addr.arpa. 600 IN PTR server5.tecaj.hr.

> send

> quit

Ukoliko je sve u redu, tada se nakon pokretanja naredbe send neće javiti poruka o greški.

1. Služeći se naredbom nslookup dodatno provjerite radi li sve kako treba:

nslookup server5

nslookup 192.168.2.205

1. Na sustavu **server1** na početak datoteke **/etc/dhcp/dhcpd.conf** dodajte:

ddns-updates on;

ddns-update-style interim;

update-static-leases on;

ignore client-updates;

include "/etc/dhcp/dhcpupdate.key";

zone tecaj.hr. {

primary 192.168.1.100;

key dhcpupdate;

}

zone 2.168.192.in-addr.arpa. {

primary 192.168.1.100;

key dhcpupdate;

}

1. Ponovno pokrenite dhcpd.
2. Na sustavu **server2** izmijenite datoteku **/etc/resolv.conf** tako da je njezin sadržaj bude:

search tecaj.hr

nameserver 192.168.2.201

1. Provjerite naredbama nslookup i host da li ispravno radi DNS poslužitelj koji ste zapisali u /etc/resolv.conf.
2. Servis za udaljeni rad

**Trajanje poglavlja:**

**60 min**

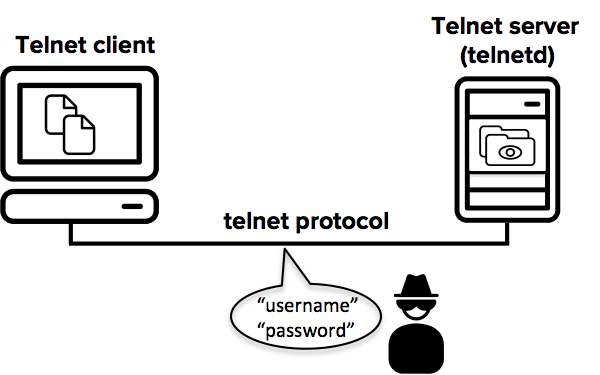


Po završetku ovoga poglavlja moći ćete:

* koristiti protokol SSH
* koristiti naredbe ssh, scp i sftp

Ova cjelina obrađuje protokol SSH (kratica od Secure Shell), mrežni protokol koji korisnicima omogućuje uspostavu sigurne veze s udaljenim poslužiteljem. Kao takav, SSH je osnovna komponenta mrežnih servisa koja omogućuje siguran i interaktivan prijenos datoteka, daljinsko nadgledanje, izvršavanje naredbi, tuneliranje i još mnogo toga. Tradicionalni mrežni protokoli kao što su TELNET, FTP, RSH i drugi, iako su vrlo praktični i jednostavni za korištenje, ujedno sadrže i velik sigurnosni nedostatak koji predstavlja ograničenje za korištenje istih.

Za izvođenje naredbi na udaljenom računalu potrebno je uspostaviti komunikaciju između klijentskog računala i udaljenog poslužitelja. Za tu svrhu dugo se vremena koristio protokol TELNET. Na klijentskom računalu pokrenuo bi se program telnet koji bi omogućavao spajanje na udaljeni poslužitelj koristeći korisničko ime i lozinku koju imamo na tom udaljenom poslužitelju. Spajanjem bi se pokrenula korisnička ljuska na udaljenom računalu i u njoj bi se mogle izvoditi naredbe potpuno jednako kao i na lokalnom računalu.



Glavni problem protokola TELNET je u tome što promet između klijenta i poslužitelja nije kriptiran te zlonamjerni napadač može prisluškivanjem mrežnoga prometa doći do povjerljivih podataka i time ugroziti sigurnost udaljenoga poslužitelja.

Iz tog razloga je protokol TELNET praktično isčezao s današnjih poslužitelja te je zamijenjen protokolom SSH koji je postao de facto standard za spajanje na udaljena računala.

2.1. Secure Shell (SSH)

SSH (kratica od Secure Shell) je mrežni protokol koji korisnicima omogućuje uspostavu sigurne veze s udaljenim poslužiteljem. Kao takav, SSH je osnovna komponenta mrežnih servisa koja omogućuje siguran i interaktivan prijenos datoteka, daljinsko nadgledanje, izvršavanje naredbi, tuneliranje i još mnogo toga. Tradicionalni mrežni protokoli kao što su TELNET, FTP, RSH i drugi, iako su vrlo praktični i jednostavni za korištenje, ujedno sadrže i velik sigurnosni nedostatak koji predstavlja ograničenje na korištenje istih.

SSH ne zamjenjuje samo protokol TELNET, već zamijenjuje cijeli set protokola poput FTP, RSH, itd.

Radi se o vrlo niskom nivou sigurnosti koji je implementiran kod većine takvih "starijih" mrežnih protokola, budući da se u vrijeme njihova nastanka nije previše pažnje obraćalo na sigurnost računalnih sustava, tj. ne postoji enkripcija pri prijenosu podataka.

U današnje vrijeme kada sigurnosni aspekt predstavlja jedan od najvažnijih elemenata svake računalne mreže taj je problem adresiran na svim razinama, od terminalske komunikacije do internetskih preglednika gdje se koriste sigurni protokoli, a nesigurni protokoli su gotov izbačeni iz upotrebe.

2.1.1. Protokol SSH

SSH protokol svoj rad temelji na uporabi kombinacije simetrične i asimetrične kriptografije, metode enkripcije koja omogućuje sigurniji prijenos podataka računalnom mrežom. SSH obično koristi TCP port 22, što je na Unix računalima definirano u datoteci /etc/services.

Treba reći da postoje dvije nekompatibilne verzije protokola SSH – SSH-1 i SSH-2. Prva verzija protokola pati od sigurnosnih propusta pa je brzo nakon nastanka zamijenjena nadograđenom inačicom SSH-2. S obzirom na to da je SSH-1 protokol zastario i sve se manje koristi, u ovom tečaju obradit ćemo samo protokol SSH-2 i gdje god se spominje protokol SSH to se zapravo odnosi na SSH-2.

Sljedećom tablicom prikazane su osnovne razlike između protokola SSH-1 i SSH-2.

|  |  |
| --- | --- |
| **SSH-1** | **SSH-2** |
| Sve u jednom protokolu | Odvojeni protokoli. |
| Slaba provjera integriteta | Jaka provjera integriteta. |
| Jedna sjednica po vezi (konekciji) | Više sjednica po vezi (konekciji). |
| Ne podržava promjenu lozinki | Podržava promjenu lozinki. |
| Ne postoji autentikacija putem javnog ključa | Postoji autentikacija putem javnog ključa. |

SSH protokol je u svojoj inačici SSH-1 nastao 1995. godine, a razvio ga je Finac Tatu Ylönen, koji je tada bio istraživač na tehnološkom sveučilištu u Helsinkiju (University of Technology, Helsinki).

Godine 1998. tvrtka SSH Communications izdaje program naziva "SSH Secure Shell" koji se temelji na inačici protokola SSH-2. Međutim, zbog toga što je program bio komercijalan, tj. nije bio besplatan kao prva inačica, on ne biva dobro prihvaćen i SSH-1 inačica protokola još par godina ostaje popularnija i korištenija nego SSH-2 verzija.

To se promijenilo tek pojavom besplatne implementacije koja se naziva OpenSSH. Nju i dandanas razvija zajednica okupljena pod projektom OpenBSD te je potpuno besplatna i otvorenog izvornog kôda. Danas je sastavni dio svih distribucija Linuxa te ostalih Unixa.

Popularnost OpenSSH implementacije je brzo rasla zbog svoje otvorenosti, dostupnosti za razne operacijske sustave i podršci za obje verzije SSH protokola. Godine 2006. protokol SSH-2 postaje predloženi Internet standard definiran u nizu RFC dokumenata.

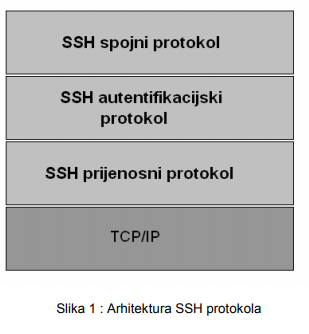
Kao što je već navedeno u uvodu ovoga poglavlja, SSH ne zamjenjuje samo protokol TELNET i ne koristi se samo za rad u ljusci na udaljenom računalu, već i zamjenjuje tzv. R-naredbe (rsh – Remote Shell, rlogin – Remote Login te rcp – Remote Copy, kao i protokol FTP – File Transfer protocol). R-naredbe, kao i TELNET ne kriptiraju promet i postoje nasljednici tih naredbi unutar protokola SSH (to su naredbe ssh, slogin, scp te sftp).

2.1.2. Arhitektura protokola SSH

Kao što je već navedeno, protokol SSH je protokol za sigurnu udaljenu prijavu i druge mrežne usluge preko nesigurne mreže kao što je Internet i koristi model klijent-poslužitelj. Protokol je definiran u nizu RFC dokumenata te se sastoji od tri glavne komponente/sloja:

* **Prijenosni protokol** (engl. **Transport Layer Protocol**) pruža autentikaciju poslužitelja, tajnost prijenosa, i integritet prenošenih podataka. Također, opcionalno omogućuje i uslugu kompresije podataka. Obično radi kao aplikacijski protokol iznad protokolnog stoga TCP/IP , ali nije ovisan o njemu, već može raditi iznad svakog protokola koji pruža pouzdan prijenos podataka.
* **Autentikacijski protokol** (engl. **User Authentication Protocol**) je zadužen za autentikaciju korisnika poslužitelju na koji se pokušava spojiti. Za njegov rad potrebno je da prijenosni sloj protokola već bude u funkciji.
* **Spojni protokol** (engl. **Connection Protocol**) je zadužen za multipleksiranje više logičkih kanala u jedan kriptirani tunel. Drugim riječima, spojni protokol iz jedne veze klijent/poslužitelj stvara različite tokove podataka, odnosno logičke kanale. Također, upravlja zahtjevima korisnika za uslugama kao što su to zahtjev za pokretanjem pseudoterminala (pty) i ljuske operacijskoga sustava.

Nakon uspostave sigurnoga prijenosnog kanala klijent obično pošalje zahtjev za pokretanjem autentifikacijskoga servisa (ssh-userauth). Također, nakon uspješne autentifikacije klijent će poslati zahtjev za pokretanjem spojnoga protokola (ssh-connection). To omogućuje da novi protokoli budu definirani i da koegzistiraju s postojećim protokolima.



Prijenosni sloj protokola SSH je najniži sloj od kojeg zavise ostala dva sloja. Najčešće se protokol vrti iznad TCP/IP protokola, ali moguće je rad i iznad drugih protokola. On pruža jaku enkripciju, autentikaciju poslužitelja, i čuva integritet prenesenih podataka. Autentikacija korisnika se ne obavlja u ovoj fazi, nego samo autentikacija poslužitelja. Ovaj protokol je dizajniran da bude jednostavan i fleksibilan, i da omogući pregovaranje o algoritmima i parametrima konekcije, i minimizira broj poruka kod pregovaranja. Algoritmi o kojima se pregovara su metode razmjene ključeva, algoritmi simetrične enkripcije, algoritam javnoga ključa za autentifikaciju poslužitelja, algoritmi za računanje sažetka poruke (hash) i MAC algoritmi (Message Authentication Code).

Poslužitelj osluškuje zahtjeve na TCP portu 22 (što je službeni broj pristupa registriran od strane IANA-e), ali to može biti i bilo koji drugi pristup. Vezu inicira klijent.

Nakon uspostavljanja sigurnoga kanala, korisnik zahtijeva pokretanje usluge autentikacije (ssh-userauth). Nakon što poslužitelj odobri zahtjev počinje proces autentikacije. Dvije najčešće korištene metode autentikacije:

* autentikacija pomoću javnoga ključa
* autentikacija pomoću lozinke.

Metoda autentikacije pomoću javnoga ključa funkcionira tako da klijent pošalje poslužitelju digitalni potpis koji on stvara svojim privatnim ključem. Također, pošalje i svoj javni ključ. Poslužitelj provjerava da taj javni ključ pripada tom korisniku. Najčešće se to radi tako da svaki korisnik u svom direktoriju na poslužitelju ima stvorenu datoteku u kojoj se nalaze njegovi javni ključevi. To je obično datoteka ~/.ssh/authorized\_keys. Sadržaj te datoteke može izgledati otprilike ovako (zbog prikaza ključ je skraćen pa se umjesto tri točkice nalazi nešto duži niz znakova):

[irako@kosjenka ~]$ cat .ssh/authorized\_keys

ssh-rsa AAAAB3NzaC1ycc...dnJrKqfUxZvQ== irako@ico

Korisnik može imati više javnih ključeva u ovoj datoteci.

Metoda autentikacije lozinkom je najkorištenija metoda autentikacije. Autentikacijski protokol koristi prijenosni protokol SSH, koji nudi tajnost i integritet poslanih podataka. Sve što se prenosi je kriptirano pa se zbog toga lozinke koje se prenose tim kanalom ne mogu doznati prisluškivanjem kanala.

**Spojni protoko**l je najviši sloj SSH protokola i radi povrh prijenosnog i autentikacijskog sloja. Ta dva protokola i sigurnost koju oni nude nužni su za rad spojnoga protokola. Spojni protokol nudi usluge kao što su interaktivne sjednice, udaljeno izvršavanje naredbi, proslijeđivanje TCP/IP prometa, i tuneliranje X11 konekcija. Osnovni pojam u SSH spojnom protokolu je kanal. Sve interaktivne sjednice i prosljeđivanja vrše se kroz kanale. Oni su doduše samo virtualni koncept jer se svi otvoreni kanali multipleksiraju kroz jednu vezu, uspostavljenu još u prijenosnom sloju.

2.1.3. Servis SSH

U operativnom sustavu Debian, sve se potrebno za korištenje protokola SSH nalazi u paketima:

* **openssh-client** – klijentski programi (naredbe scp , ssh, slogin, sftp te naredbe za rukovanje ključevima poput ssh-keygen...)
* **openssh-server** – poslužiteljski servisni process sshd (daemon).

Na klijentskom računalu je potrebno instalirati SSH klijent koristeći naredbu (za instalaciju su potrebne root ovlasti):

[root@client ~]# apt-get install openssh-client

Reading package lists... Done

Building dependency tree

Reading state information... Done

0 upgraded, 0 newly installed, 1 reinstalled, 0 to remove and 81 not upgraded.

Need to get 779 kB of archives.

After this operation, 0 B of additional disk space will be used.

Get:1 http://ftp.hr.debian.org/debian stretch/main amd64 openssh-client amd64 1:7.4p1-10+deb9u3 [779 kB]

Fetched 779 kB in 0s (10.5 MB/s)

(Reading database ... 174808 files and directories currently installed.)

Preparing to unpack .../openssh-client\_1%3a7.4p1-10+deb9u3\_amd64.deb ...

Unpacking openssh-client (1:7.4p1-10+deb9u3) over (1:7.4p1-10+deb9u3) ...

Processing triggers for man-db (2.7.6.1-2) ...

Setting up openssh-client (1:7.4p1-10+deb9u3) ...

Na poslužitelju je potrebno instalirati SSH poslužitelj koristeći naredbu:

[root@server ~]# apt-get install openssh-server

Reading package lists... Done

Building dependency tree

Reading state information... Done

0 upgraded, 0 newly installed, 1 reinstalled, 0 to remove and 81 not upgraded.

Need to get 332 kB of archives.

After this operation, 0 B of additional disk space will be used.

Get:1 http://ftp.hr.debian.org/debian stretch/main amd64 openssh-server amd64 1:7.4p1-10+deb9u3 [332 kB]

Fetched 332 kB in 0s (6,460 kB/s)

Preconfiguring packages ...

(Reading database ... 174808 files and directories currently installed.)

Preparing to unpack .../openssh-server\_1%3a7.4p1-10+deb9u3\_amd64.deb ...

Unpacking openssh-server (1:7.4p1-10+deb9u3) over (1:7.4p1-10+deb9u3) ...

Setting up openssh-server (1:7.4p1-10+deb9u3) ...

Processing triggers for systemd (232-25+deb9u3) ...

Processing triggers for man-db (2.7.6.1-2) ...

Instalacijom paketa openssh-server pokrenut je servis sshd, što možemo provjeriti naredbom ps:

[irako@server ~]$ ps -ef | grep ssh

root 522 1 0 Aug05 ? 00:00:00 /usr/sbin/sshd -D

irako 29258 1366 0 10:35 pts/2 00:00:00 grep sshd

Ako želimo zaustaviti servis sshd, to možemo koristeći naredbu systemctl: [root@server ~]$ systemctl stop ssh

Konfiguracijske datoteke za SSH nalaze se u direktoriju **/etc/ssh**. Koristeći naredbu ls provjerit ćemo sadržaj toga direktorija:

[irako@client ~]$ ls -al /etc/ssh

total 592

drwxr-xr-x 2 root root 4096 May 9 07:48 .

drwxr-xr-x 132 root root 12288 Aug 9 00:02 ..

-rw-r--r-- 1 root root 553122 Mar 1 16:17 moduli

-rw-r--r-- 1 root root 1723 Mar 1 16:17 ssh\_config

-rw-r--r-- 1 root root 3298 Mar 1 16:17 sshd\_config

-rw------- 1 root root 227 May 9 07:48 ssh\_host\_ecdsa\_key

-rw-r--r-- 1 root root 175 May 9 07:48 ssh\_host\_ecdsa\_key.pub

-rw------- 1 root root 399 May 9 07:48 ssh\_host\_ed25519\_key

-rw-r--r-- 1 root root 95 May 9 07:48 ssh\_host\_ed25519\_key.pub

-rw------- 1 root root 1679 May 9 07:48 ssh\_host\_rsa\_key

-rw-r--r-- 1 root root 395 May 9 07:48 ssh\_host\_rsa\_key.pub

Datoteka ssh\_config služi za konfiguraciju SSH klijenta, dok datoteka sshd\_config služi za konfiguraciju SSH poslužitelja. Ostale datoteke su poslužiteljski ključevi generirani prilikom instalacije paketa openssh-server.

Neke od osnovnih opcija koje je moguće konfigurirati u datoteci sshd\_config prikazane su u ovoj tablici:

|  |  |
| --- | --- |
| Port 22 | Definira port na kojem se pokreće servis sshd. Može se postaviti da istovremeno radi na više portova. |
| Protocol 2,1 | Definira koja se inačica protokola koristi. Može se postaviti podrška za obje inačice, pri čemu će podrazumijevana biti ona koja je prva definirana. |
| DenyUsers [USER]@HOST | Definira korisnike koji se ne smiju spojiti na SSH poslužitelj. |
| PermitEmptyPasswords yes/no | Ako je postavljeno na yes, dopušteno je spajanje korisnika bez lozinke. Iz sigurnosnih razloga, podrazumijevana vrijednost je no. |
| PermitRootLogin yes/no | Ovom opcijom se dopušta spajanje na SSH poslužitelj kao administratorski korisnik root. Predlaže se da je opcija postavljena na no. |
| X11Forwarding yes/no | Opcija se koristi kada se kroz SSH žele izvoditi grafički programi na poslužitelju unutar okoline X11 te ih prikazati na klijentskom računalu. |

Ako mijenjamo opcije unutar konfiguracijske datoteke **sshd\_config**, servis sshd je potrebno ponovno pokrenuti kako bi se učitala nova konfiguracija. To možemo koristeći naredbu systemctl: [root@server ~]$ systemctl restart ssh.

2.1.4. Servis SSH

Nakon instalacije i konfiguracije servisa SSH, moguće se spojiti na udaljeno računalo koristeći naredbu ssh:

[irako@client ~]$ ssh server

The authenticity of host 'server (161.53.100.100)' can't be established.

ECDSA key fingerprint is SHA256:R8EUYL0HrkIRzuxH8+TYeH96+g2DSb30tQRHTdL2oeA.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.

irako@server password:

Linux server 4.9.0-11-amd64 #1 SMP Debian 4.9.189-3+deb9u2 (2019-11-11) x86\_64

The programs included with the Debian GNU/Linux system are free software;

the exact distribution terms for each program are described in the

individual files in /usr/share/doc/\*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent

permitted by applicable law.

Last login: Wed Nov 13 09:14:04 2019 from 192.168.1.108

[irako@server ~]$

Kako još nije postavljen privatni i javni ključ, metoda autentikacije je pomoću lozinke. Najprije je klijent pitao vjerujemo li poslužitelju i prihvaćamo li poslužiteljski ključ (pri sljedećem spajanju to nas više neće pitati, jer će na klijentu biti zapisan ključ ssh poslužitelja u datoteci ~/.ssh/known\_hosts) te nas je pitao lozinku. Upisivanjem ispravne lozinke dobivena je ljuska pokrenuta na udaljenom poslužitelju. Ako se želi koristiti autentikacija pomoću javnog ključa, potrebno je najprije generirati par ključeva. Na klijentskom računalu potrebno je pokrenuti naredbu ssh-keygen kojom generiramo privatni i javni ključ:

[irako@client ~]$ ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/home/irako/.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/irako/.ssh/id\_rsa.

Your public key has been saved in /home/irako/.ssh/id\_rsa.pub.

The key fingerprint is:

SHA256:yOxNHMIgWCHJcbpZodXqSsOk0JY4CMVm6vYTcN/yOaA irako@tydirium

The key's randomart image is:

+---[RSA 2048]----+

|oB\*=o |

|+oO..+ |

|oO o. o . |

|\*oBo o + . |

|\*=+ . = S |

|.\* o = + |

|o + o = o |

| . E + |

| . . |

+----[SHA256]-----+

Sadržaj datoteke .ssh/id\_rsa.pub potrebno je dodati na kraj datoteka .ssh/authorized\_keys na udaljenom poslužitelju. Prilikom sljedećeg spajanja na poslužitelj, autentikacija će se provoditi metodom javnoga ključa. Idemo probati:

[irako@client ~]$ ssh server

Enter passphrase for key '/home/irako/.ssh/id\_rsa':

Linux bagan 4.9.0-6-amd64 #1 SMP Debian 4.9.88-1+deb9u1 (2018-05-07) x86\_64

The programs included with the Debian GNU/Linux system are free software;

the exact distribution terms for each program are described in the

individual files in /usr/share/doc/\*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent

permitted by applicable law.

Last login: Fri Aug 10 09:58:59 2018 from 161.53.2.79

[irako@server ~]$

Program ssh-agent ima namjenu da lozinku privatnoga ključa spremi u radnu memoriju kako ga ne bismo morali upisivati pri svakom spajanju. Zatim se naredbom ssh-add pridodaje ključ ssh-agentu:

[irako@client ~]$ eval "$(ssh-agent -s)"; ssh-add ~/.ssh/id\_rsa

Agent pid 28851

Enter passphrase for /home/irako/.ssh/id\_rsa:

Identity added: /home/irako/.ssh/id\_rsa (/home/irako/.ssh/id\_rsa)

Kod sljedećeg spajanja na udaljeni poslužitelj, više nije potrebno upisivati lozinku privatnoga ključa:

[irako@client ~]$ ssh server

The programs included with the Debian GNU/Linux system are free software;

the exact distribution terms for each program are described in the

individual files in /usr/share/doc/\*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent

permitted by applicable law.

Last login: Fri Aug 10 10:01:40 2018 from c41.srce.vpn

[irako@server ~]$

Ako se korisničko ime na udaljenom poslužitelju razlikuje od onog na klijentskom računalu, korisničko ime možete upisati na dva načina:

ssh username@server

ssh -l username server

Već smo napomenuli da SSH zamijenjuje cijeli skup alata koji nisu koristili enkripciju. Naredba rsh služi za izvođenje naredbe na udaljenom poslužitelju. Sada se umjesto naredbe rsh može to isto napraviti koristeći naredbu ssh. U sljedećem primjeru ćemo na udaljenom poslužitelju pokrenuti naredbu hostname koja će ispisati naziv poslužitelja:

[irako@client ~]$ ssh server hostname

server

Naredba rcp služi za kopiranje datoteka s klijentskog računala na udaljeni poslužitelj. Naredba scp zamjenjuje naredbu rcp, s tom razlikom da za transportni protokol koristi SSH pa je time veza kriptirana. U sljedećem primjeru naredbom touch izradit će se datoteka test.txt i zatim poslati na udaljeni poslužitelj u home direktorij korisnika koji je pokrenuo naredbu:

[irako@client ~]$ touch test.txt

[irako@client ~]$ scp test.txt server:

test.txt 100% 0 0.0KB/s 00:00

[irako@client ~]$

Sada će se na klijentskom računalu obrisati datoteka test.txt i s udaljenog poslužitelja će se kopirati inačica datoteke nazad na klijentsko računalo:

[irako@client ~]$ rm test.txt

[irako@client ~]$ scp server:test.txt .

test.txt 100% 0 0.0KB/s 00:00

[irako@client ~]$ ls -al test.txt

-rw-r--r-- 1 irako irako 0 Aug 10 10:08 test.txt

Naredba ftp služi za prijenos datoteka između računala na mreži. Naredba sftp zamjenjuje naredbu ftp, s tom razlikom da se za prijenosni protokol koristi SSH tako da promet između klijenta i poslužitelja bude kriptiran. Naredbe unutar klijentskog programa su identične kao kod protokola FTP (cd, lcd, get, put…). Evo i primjera:

[irako@client ~]$ sftp server

Connected to server.

sftp> cd tmp

sftp> ls

test.txt

sftp> get test.txt

Fetching /home/irako/tmp/test.txt to test.txt

sftp> quit

2.2. Vježba 3. SSH

1. **Instalacija programske podrške za SSH**
2. Na sustavu **server1** instalirajte paket **openssh-client** koristeći naredbu apt-get.
3. Na sustavu **server2** instalirajte paket **openssh-server** koristeći naredbu apt-get.
4. Pogledajte datoteke unutar direktorija /**etc/ssh/.**
5. Čemu služi datoteka **sshd\_config**?
6. Provjerite u datoteci /**etc/ssh/sshd\_config** vrijednost varijable PermitRootLogin. Postavite je na vrijednost **no** i ponovno pokrenite SSH poslužitelj.
7. Provjerite možete li se na poslužitelj spojiti direktno kao korisnik *root*?
8. **Autentikacija putem ključeva**
9. U ovom zadatku omogućit će se sigurno prijavljivanje korisnika *root* s klijentskog računala na korisnički račun *tux* na poslužitelju i to bez uporabe lozinke.
10. Prethodno je potrebno stvoriti korisnika tux na poslužitelju.

Na klijentskom računalu (kao korisnik *root*) pokrenite sljedeću naredbu:  
  
ssh-keygen -q -t rsa -f ~/.ssh/id\_rsa -C '' -N ''

1. Pogledajte sadržaj direktorija **~/.ssh**. Koje se dvije datoteke nalaze u ovom direktoriju?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Sadržaj datoteke **id\_rsa.pub** kopirajte na kraj datoteke **~tux/.ssh/authorized\_keys** koja se nalazi na poslužitelju.
2. Pokušajte se sa sustava **server1** prijaviti na sustav **server2** kao korisnik *tux*:  
   ssh tux@server2

Jeste li se uspjeli spojiti? \_\_\_\_\_\_\_\_\_\_\_\_

**Napomena**

Pripazite da vlasnik direktorija **.ssh** i datoteke **authorized\_keys** bude korisnik *tux* (i da pripadaju odgovarajućoj skupini) te da prava pristupa na direktorij **.ssh** budu 700, a na datoteke **authorized\_keys** budu 640:  
  
chown -R tux:tux ~tux/.ssh  
chmod 700 ~tux/.ssh  
chmod 640 ~tux/.ssh/authorized\_keys

1. Autentikacijski servisi

**Trajanje poglavlja:**

**120 min**



Po završetku ovoga poglavlja moći ćete:

* koristiti protokole LDAP i RADIUS
* koristiti osnove alata OpenLDAP
* koristiti osnove alata FreeRADIUS
* upoznati se sa osnovnim servisima infrastrukture AAI@EduHR

Ova cjelina obrađuje najčešće korištene protokole za imenički servis i autentikaciju korisnika. LDAP je skraćenica složenice na engleskom jeziku Lightweight Directory Access Protocol i naziv je aplikacijskoga protokola za upravljanje imenikom putem TCP/IP mreže. RADIUS (Remote Authentication Dial In User Service) je mrežni protokol koji omogućuje centralizirano upravljanje autentikacijom, autorizacijom i administracijom (engl. Authentication, Authorization and Accounting, AAA) korisnika prilikom spajanja i korištenja mrežnih usluga.

Na kraju poglavlja bit će govora o sustavu AAI@EduHr te kako autenticirati korisnike korištenjem protokola RADIUS i LDAP.

3.1. LDAP

LDAP je skraćenica složenice na engleskom jeziku Lightweight Directory Access Protocol i naziv je aplikacijskoga protokola za upravljanje (spremanje i preuzimanje podataka, traženje podataka koji odgovaraju zadanom skupu kriterija, provjeru autentičnosti klijenata i još mnogo toga) imenikom putem TCP/IP mreže. Imenik je u LDAP-u datoteka ili skupina podataka koji su organizirani slično kao telefonski imenik, koji sadrže podatke o korisnicima, datotekama i aplikacijama, kao i njihove sigurnosne postavke. Zadnja inačica LDAP-a jest 3. Ovaj protokol je detaljno opisan u dokumentu IETF RFC 4510.

3.1.1. Protokol LDAP

Podaci u LDAP poslužitelju organizirani su u hijerarhijsko-relacijskom formatu. Hijerarhijski je zato što svaki zapis, osim korijenskog, ima jedan "roditeljski" zapis, a relacijski je jer se više zapisa može grupirati zajedno. Najviša razina hijerarhije u LDAP poslužitelju naziva se domenom. U jednom takvom poslužitelju može postojati više domena, jer je LDAP osmišljen tako da pruža globalnu uslugu direktorija što ponekad nije moguće ostvariti jednom vršnom domenom. Ispod domene su grane koje predstavljaju organizacijske jedinice koje su najčešće odjeli neke organizacije. Svaki zapis koji nije domena ili organizacijska jedinica naziva se list.

3.1.2. Protokol LDAP

LDAP se zasniva na četirima modelima:

* informacijski model
* model imenovanja
* model funkcionalnosti
* sigurnosni model.

3.1.2.1. Informacijski model

Opisuje strukturu informacijskoga stabla direktorija; izveden je iz standarda X.500. Važni pojmovi su:

* razred – označava grupu objekata koji imaju zajednička svojstva. Razredi se mogu nasljeđivati i postoje tri vrste:
  + apstraktni razredi (engl. abstract classes) – služe kao predlošci za strukturne razrede
  + strukturni razredi (engl. structural classes) – opisuju zapise u direktoriju
  + pomoćni razredi (engl. auxiliary classes) – definiraju skup atributa za nasljeđivanje
* atributi – jedinice podataka na temelju kojih se definiraju razredi. U shemi se definiraju zasebno od razreda, te je na taj način omogućeno korištenje iste definicije atributa u više različitih razreda
* sintaksa atributa – definira koju vrstu podataka i koje vrijednosti može sadržavati pojedini atribut
* zapisi – opisuju objekte iz stvarnog svijeta; svaki zapis je pojava (engl. instance) jednoga strukturnog razreda. To znači da sadrži vrijednost i poštuje ograničenja atributa definiranih u razredu
* shema – sadrži listu razreda i atributa koji se mogu koristiti i nasljeđivati. Kako bi zapis pripadao informacijskom stablu direktorija, mora odgovarati formatu definiranom u shemi.

U tablici 1. prikazani su neki od mogućih LDAP sintaksi atributa, što je ekvivalent za vrstu podataka. Svaka sintaksa atributa je povezana (bilo izričito ili implicitno) s atributom, a sve vrijednosti za atribute te vrste moraju se pridržavati ograničenja te sintakse. U tablici 2. dani su neki od važnijih LDAP atributa. Atributi mogu imati i pseudonime (engl. alias) koji se mogu upotrijebiti umjesto da se koristi njihovo puno ime. Primjerice cn se može upotrijebiti kao referenca na atribut commonName.

|  |  |
| --- | --- |
| **Sintaksa atributa** | **Opis** |
| bin | Binarna informacija. |
| ces | case exact string; koristi se i naziv directory string; slučaj (engl. case) je važan tijekom uspoređivanja. |
| cis | case ignore string; slučaj nije važan tijekom uspoređivanja. |
| tel | telefonski broj; brojevi se promatraju kao tekst, a praznine se ignoriraju. |
| dn | distinguished name. |

Tablica 1. LDAP sintakse atributa

|  |  |  |  |
| --- | --- | --- | --- |
| **Atribut, alias** | **Sintaksa** | **Opis** | **Primjer** |
| CommonName, cn | cis | Uobičajeno ime zapisa | Ivan Horvat |
| surname, sn | cis | Prezime osobe | Horvat |
| telephoneNumber | te | Telefonski broj | 123-456-789 |
| OrganizationalUnitName, ou | cis | Ime organizacijske jedinice | Sys |
| owner | dn | Ime osobe koja posjeduje zapis | cn=Ivan  Horvat,ou=sys,dc=srce,dc=hr |
| organization, o | cis | Ime organizacije | Srce |
| jpegPhoto | bin | Fotografija u formatu JPEG | Slika Ivana Horvata |

Tablica 2. LDAP atributi

Svaki zapis direktorija ima poseban atribut koji se zove objectClass. Vrijednost tog atributa sastoji se od liste dvije ili više naziva shema. Te sheme definiraju tipove objekata koje zapis predstavlja. objectClass definira koje atribute zapis treba imati. Tablica 3. prikazuje dio općenite sheme (objektni razredi i pripadni atributi). U mnogim slučajevima, zapis može sadržavati više od jednog objektnog razreda.

|  |  |  |
| --- | --- | --- |
| **Objektni razredi** | **Opis** | **Potrebni atributi** |
| inetOrgPerson | Definira zapise za osobu | commonName (cn); surname (sn); objectClass |
| organizationalUnit | Definira zapise za organizacijske jedinice | ou; objectClass |
| organization | Definira zapise za organizacije | o; objectClass |

3.1.2.1. Informacijski model

Modelom funkcionalnosti definiraju se operacije nad podacima u direktoriju. Postoji devet operacija koje su podijeljene u tri skupine:

* autentikacija – omogućuje korisničkom programu da dokaže svoj identitet kroz nekoliko operacija:
* Open – otvara vezu prema sustavskoj komponenti direktorija LDAP poslužitelja.
* Bind – otvara sjednicu između korisničkog programa i LDAP poslužitelja koja omogućuje razmjenu podataka potrebnih za autentikaciju.
* Unbind – prekida sjednicu između korisničkog programa i LDAP poslužitelja

izmjena nad podacima, a koriste se sljedeće operacije:

* Add – stvara objekt u informacijskom stablu direktorija koji mora zadovoljavati uvjete definirane u shemi.
* Modify – mijenja vrijednost određenog atributa zapisa, a obuhvaća dodavanje, izmjenu i brisanje vrijednosti atributa.
* Modify RDN – omogućuje micanje zapisa unutar stabla direktorija.
* Delete – omogućuje brisanje zapisa iz stabla direktorija.

3.1.2.4. Sigurnosni model

Definira mogućnosti sigurnoga pristupa podacima unutar informacijskog stabla direktorija. Standard definira korištenje postojećih SASL mehanizama (Simple Authentication and Security Layer) za osiguravanje pristupa podacima. SASL sigurnosni mehanizmi koriste se za sigurnu autentikaciju, a po potrebi je moguće zaštititi i cjelokupnu komunikaciju između korisničkog programa i poslužiteljske komponente direktorija.

3.1.3. Upravljanje LDAP-om

Protokol LDAP prometuje TCP portovima 389 (nije kriptirano, čisti tekst) ili 686 (kriptirano, korištenjem standardnih protokola poput SSL-a ili TLS-a).

Klijenti obavljaju protokolske radnje nad poslužiteljem tako da klijent šalje zahtjev opisujući operaciju koju treba izvršiti poslužitelj. Nakon što primi zahtjev, poslužitelj je zadužen obaviti potrebnu operaciju u direktoriju. Nakon što je radnja izvršena, poslužitelj vraća natrag klijentu odgovor koji sadrži rezultate operacije ili informaciju o pogrešci. Cilj protokola LDAP jest smanjivanje kompleksnosti klijenta kako bi se povećala brzina i efikasnost uporabe usluge direktorija.

Kada klijent komunicira s LDAP poslužiteljem, prolazi kroz tri osnovne faze:

1. uspostavljanje veze s poslužiteljem
2. obavljanje određene operacije nad direktorijem
3. prekidom veze s poslužiteljem.

Proces uspostave i raskida veze obavlja se putem standardnih TCP/IP mehanizama. LDAP definira nekoliko operacija koje se mogu izvršavati nad poslužiteljem:

* povezivanje s poslužiteljem
* pretraživanje sadržaja direktorija
* usporedba zapisa
* dodavanje zapisa u direktorij
* modificiranje postojećih zapisa
* brisanje zapisa iz direktorija.

3.1.4. Sheme

Shema sadrži listu razreda i atributa koji se mogu koristiti i nasljeđivati. Kako bi zapis pripadao informacijskom stablu direktorija mora odgovarati formatu definiranom u shemi. Postoje standardne sheme u LDAP-u, poput sheme core ili inetOrgPerson kojima su definirani standardni atributi. No organizacije mogu za svoje potrebe napraviti i svoje vlastite sheme.

Autentikacijska i autorizacijska infrastruktura znanosti i visokog obrazovanja u Republici Hrvatskoj (AAI@EduHr) jest infrastrukturni, posrednički sustav čija je temeljna zadaća omogućiti sigurno, pouzdano i učinkovito upravljanje elektroničkim identitetima te njihovu uporabu za pristup mrežnim i mrežom dostupnim resursima.

Poslove koordinacije, razvoja i održavanja sustava AAI@EduHr obavlja Srce – Sveučilišni računski centar Sveučilišta u Zagrebu.

AAI@EduHr svoje polazište koncepcijski ima u distribuiranom sustavu imenika utemeljenih na LDAP standardu. Izgrađena hijerarhija RADIUS poslužitelja vezanih za LDAP imenike na ustanovama, uz odgovarajuće središnje tzv. proxy RADIUS poslužitelje, standardno se koristi kao temelj autenticiranog i autoriziranog pristupa mreži. Usporedno s RADIUS infrastrukturom, sustav jedinstvene autentikacije korisnika temeljen na SAML 2.0 (Security Assertion Markup Language) standardu koristi se za autentikaciju i autorizaciju korisnika prilikom pristupanja web–aplikacijama.

3.1.4.1. hrEdu imeničke sheme

Sustav AAI@EduHr temelji se na distribuiranom sustavu imenika utemeljenih na LDAP tehnologiji. Kako bi taj distribuirani sustav imenika bio jasno definiran, pouzdan i učinkovit, nužno je definirati odgovarajuću imeničku shemu odnosno precizan popis atributa s jasnim opisom, semantikom i sintaksom.

Za potrebe AAI@EduHr infrastrukture definirane su dvije imeničke sheme, hrEduPerson (za podatke o osobama) i hrEduOrg (za podatke o ustanovama). Prve inačice tih imeničkih shema objavljene su 2005. godine, a aktualna specifikacija hrEdu imeničkih shema (inačica 1.3.1) objavljena je u srpnju 2010. godine.

Najnovije verzije imeničkih shema hrEduPerson i hrEduOrg izrađene su na temelju aktualnih standarda za imeničke sheme person, orgPerson, inetOrgPerson, eduPerson i SCHAC, a prilikom izrade nastojalo se uvažiti i sugestije korisnika te iskoristiti dosadašnja iskustva u razvoju i održavanju sustava AAI@EduHr i njegovoga povezivanja sa srodnim sustavima u zemlji i svijetu.

Na stranici Pregled atributa možete vidjeti detaljne informacije o sadržaju imeničkih shema hrEduPerson i hrEduOrg, a na stranici Pregled šifrarnika možete vidjeti popis dozvoljenih vrijednosti za pojedine atribute.

3.1.5. Servis OpenLDAP

**OpenLDAP** je popularno programsko rješenje otvorenoga kôda za implementaciju protokola LDAP.

Do inačice 2.3 OpenLDAP programskoga rješenja konfiguracijske postavke su se nalazile u datoteci **/etc/ldap/slapd.conf** u tekstualnom zapisu te su one bile učitavane u radnu memoriju prilikom pokretanja servisa LDAP. Loša strana toga je bila ta što se takve informacije nisu mogle dinamički mijenjati nego je za svaku promjenu bilo potrebno ponovno pokrenuti servis OpenLDAP.

Prilikom prelaska na verziju 2.3 predstavljen je novi način pohranjivanja konfiguracijskih postavki. One se sada nalaze u bazi podataka koja je implementirana putem tekstualnoga zapisa na disku. Ovakav način korištenja konfiguracijskih postavki omogućuje djelomično mijenjanje istih tijekom rada samoga servisa.

Sam zapis tekstualne baze podataka na disku nalazi se kao struktura direktorija u stablu podirektorija **/etc/ldap/slapd.d**. Prelaskom na novi način pohranjivanja konfiguracije došlo je do promjena načina rada s istom te se umjesto ažuriranja podataka u tekstualnoj datoteci promjene implementiraju koristeći standardne LDAP funkcije, a ručno ažuriranje LDIF datoteka u sustavu podirektorija **slapd.d** nikako se ne preporučuje.

Na Debianu se OpenLDAP može instalirati korištenjem naredbe apt-get install slapd:

# apt-get install slapd

Reading package lists... Done

Building dependency tree

Reading state information... Done

The following extra packages will be installed:

libodbc1 libslp1

Suggested packages:

libmyodbc odbc-postgresql tdsodbc unixodbc-bin slpd openslp-doc libsasl2-modules-gssapi-mit libsasl2-modules-gssapi-heimdal

The following NEW packages will be installed:

libodbc1 libslp1 slapd

0 upgraded, 3 newly installed, 0 to remove and 6 not upgraded.

Need to get 1,671 kB of archives.

After this operation, 4,973 kB of additional disk space will be used.

Do you want to continue? [Y/n]

Get:1 http://ftp.hr.debian.org/debian/ jessie/main slapd amd64 2.4.40+dfsg-1+deb8u4 [1,419 kB]

Fetched 1,419 kB in 0s (9,164 kB/s)

Preconfiguring packages …

Selecting previously unselected package slapd.

Preparing to unpack .../slapd\_2.4.40+dfsg-1+deb8u4\_amd64.deb ...

Unpacking slapd (2.4.40+dfsg-1+deb8u4) ...

Processing triggers for systemd (215-17+deb8u13) ...

Processing triggers for man-db (2.7.0.2-5) ...

Setting up slapd (2.4.40+dfsg-1+deb8u4) ...

Moving old database directory to /var/backups:

- directory unknown... done.

Creating initial configuration... done.

Creating LDAP directory... done.

Processing triggers for libc-bin (2.19-18+deb8u10) ...

Processing triggers for systemd (215-17+deb8u13) …

Srce je pripremilo programski paket openldap-aai u kojem se nalaze dodatne sheme hrEduPerson i hrEduOrg te je LDAP prilagođen okruženju AAI@EduHr. Taj paket može se instalirati korištenjem naredbe apt-get install openldap-aai:

# apt-get install openldap-aai

Reading package lists... Done

Building dependency tree

Reading state information... Done

Suggested packages:

freeradius-aai aosi-aai aosi-www-aai

The following NEW packages will be installed:

openldap-aai

0 upgraded, 1 newly installed, 0 to remove and 6 not upgraded.

Need to get 23.7 kB of archives.

After this operation, 182 kB of additional disk space will be used.

Get:1 http://ftp.srce.hr/srce-debian/ srce-jessie/main openldap-aai all 2.4.40~srce2 [23.7 kB]

Fetched 23.7 kB in 0s (2,010 kB/s)

Preconfiguring packages …

Selecting previously unselected package openldap-aai.

(Reading database ... 179126 files and directories currently installed.)

Preparing to unpack .../openldap-aai\_2.4.40~srce2\_all.deb ...

Unpacking openldap-aai (2.4.40~srce2) ...

Processing triggers for man-db (2.7.0.2-5) ...

Setting up openldap-aai (2.4.40~srce2) ...

AAI: OpenLDAP in standalone mode.

AAI: Creating initial configuration...

AAI: Creating new configuration...

AAI: Reindexing LDAP database...

AAI: No hrEduOrg record!

AAI: Creating new database...

AAI: Adding hreduadmin user...

Enter LDAP Password:

adding new entry "cn=hreduadmin,dc=srce,dc=hr"

AAI: Enabling ppolicy module...

modifying entry "cn=module{0},cn=config"

AAI: Restarting slapd...

AAI: Enabling overlay module...

adding new entry "olcOverlay=ppolicy,olcDatabase={1}mdb,cn=config"

AAI: Adding ppolicy entry ou=policies,dc=srce,dc=hr...

Enter LDAP Password:

adding new entry "ou=policies, dc=srce,dc=hr"

AAI: Adding ppolicy entry cn=newUser,ou=policies,dc=srce,dc=hr...

Enter LDAP Password:

adding new entry "cn=newUser, ou=policies, dc=srce,dc=hr"

AAI: Adding ppolicy entry cn=passwordDefault,ou=policies,dc=srce,dc=hr...

Enter LDAP Password:

adding new entry "cn=passwordDefault, ou=policies, dc=srce,dc=hr"

3.1.5.1. Naredba ldapadd

Za rad s imenikom, na raspolaganju je nekoliko naredbi: ldapadd, ldapsearch, ldapmodify i ldapdelte. Naredbom ldapadd može se dodati podatak u imenik. Da bismo dodali ili promijenili podatak u imeniku, potrebno je izraditi LDIF datoteku u kojoj će biti navedene promjene. Na primjer, za dodavanje korisnika *testko* u imenik, izradimo LDIF datoteku sljedećega sadržaja (npr. /tmp/testko.ldif):

dn: uid=testko,dc=srce,dc=hr

objectClass: person

objectClass: organizationalPerson

objectClass: inetOrgPerson

objectClass: hrEduPerson

objectClass: schacContactLocation

o: Srce

hrEduPersonHomeOrg: srce.hr

postalAddress: Josipa Marohnica 5

l: Zagreb

uid: testko

hrEduPersonUniqueID: testko@srce.hr

hrEduPersonPersistentID: 271624687c4e7fe817706913cdbded64

userPassword: {SSHA}WbUZmban6bx7N1xA7SIKpHx99uAow84s

givenName: Testko

sn: Testic

cn: Testko Testic

hrEduPersonOIB: 81690555286

hrEduPersonUniqueNumber: OIB: 81690555286

mail: testko@srce.hr

hrEduPersonAffiliation: djelatnik

hrEduPersonPrimaryAffiliation: djelatnik

hrEduPersonRole: administrator imenika

hrEduPersonExpireDate: NONE

Korištenjem naredbe ldapadd moguće je dodati ovoga korisnika u LDAP imenik:

# cat /tmp/testko.ldif | ldapadd -h localhost -p 389 -x -D "cn=admin,dc=srce,dc=hr" -W

Enter LDAP Password:

adding new entry "uid=testko,dc=srce,dc=hr"

3.1.5.2. Naredba ldapsearch

Naredba ldapsearch koristi se za pretraživanje LDAP imenika. Sljedećom naredbom prikazat ćemo podatke o korisniku **testko** u imeniku:

# ldapsearch -x -H ldap://localhost:389/ -b dc=srce,dc=hr uid=testko

# extended LDIF

#

# LDAPv3

# base <dc=srce,dc=hr> with scope subtree

# filter: uid=testko

# requesting: ALL

#

# testko, srce.hr

dn: uid=testko,dc=srce,dc=hr

objectClass: person

objectClass: organizationalPerson

objectClass: inetOrgPerson

objectClass: hrEduPerson

objectClass: schacContactLocation

o: Srce

hrEduPersonHomeOrg: srce.hr

postalAddress: Josipa Marohnica 5

l: Zagreb

uid: testko

hrEduPersonUniqueID: testko@srce.hr

hrEduPersonPersistentID: 271624687c4e7fe817706913cdbded64

givenName: Testko

sn: Testic

cn: Testko Testic

mail: testko@srce.hr

hrEduPersonAffiliation: djelatnik

hrEduPersonPrimaryAffiliation: djelatnik

hrEduPersonExpireDate: NONE

# search result

search: 2

result: 0 Success

# numResponses: 2

# numEntries:

3.1.5.3. Naredba ldapmodify

Ako se želi promijeniti vrijednost određenog atributa, to se može na sljedeći način. Najprije je potrebno napraviti LDIF datoteku sljedećega sadržaja (npr. /tmp/promijeni-testko.ldif):

dn: uid=testko,dc=srce,dc=hr

changetype: modify

replace: postalAddress

postalAddress: Marohniceva 5

-

replace: o

o: Sveucilisni racunski centar

Iz te datoteke je vidljivo da mijenjamo attribute postalAddress i o s novim vrijednostima, te da se ta promjena odnosi na korisnika testko. Korištenjem naredbe ldapmodify moguće je promijeniti vrijednosti tih atributa:

# cat /tmp/promjena-testko.ldif | ldapmodify -h localhost -p 389 -x -D "cn=admin,dc=srce,dc=hr" -W

Enter LDAP Password:

modifying entry "uid=testko,dc=srce,dc=hr"

Uporabom naredbe ldapsearch moguće je provjeriti je li promjena aktivna u imeniku:

# ldapsearch -x -H ldap://localhost:389/ -b dc=srce,dc=hr uid=testko

# extended LDIF

#

# LDAPv3

# base <dc=srce,dc=hr> with scope subtree

# filter: uid=testko

# requesting: ALL

#

# testko, srce.hr

dn: uid=testko,dc=srce,dc=hr

objectClass: person

objectClass: organizationalPerson

objectClass: inetOrgPerson

objectClass: hrEduPerson

objectClass: schacContactLocation

hrEduPersonHomeOrg: srce.hr

l: Zagreb

uid: testko

hrEduPersonUniqueID: testko@srce.hr

hrEduPersonPersistentID: 271624687c4e7fe817706913cdbded64

givenName: Testko

sn: Testic

cn: Testko Testic

mail: testko@srce.hr

hrEduPersonAffiliation: djelatnik

hrEduPersonPrimaryAffiliation: djelatnik

hrEduPersonExpireDate: NONE

postalAddress: Marohniceva 5

o: Sveucilisni racunski centar

# search result

search: 2

result: 0 Success

# numResponses: 2

# numEntries: 1

3.1.5.4. Naredba ldapdelete

Naredba ldapdelete služi za brisanje zapisa iz LDAP imenika. U sljedećem će se primjeru obrisati korisnik **testko**:

# ldapdelete -h localhost -p 389 -x -D "cn=admin,dc=srce,dc=hr" -W "uid=testko,dc=srce,dc=hr"

Enter LDAP Password:

3.1.5.5. AOSI web–sučelje

Da bi se lakše pretraživalo i održavalo LDAP imenik, Srce je u tu svrhu napravilo *web*-sučelje pod nazivom AOSI (Aplikacija za održavanje sadržaja imenika). To sučelje moguće je instalirati naredbom apt-get install aosi-www-aai

# apt-get install aosi-www-aai

Reading package lists... Done

Building dependency tree

Reading state information... Done

The following extra packages will be installed:

aosi-aai

Suggested packages:

apache2-cn php5-dev

The following NEW packages will be installed:

aosi-aai aosi-www-aai

0 upgraded, 5 newly installed, 0 to remove and 6 not upgraded.

Need to get 237 kB/554 kB of archives.

After this operation, 3,882 kB of additional disk space will be used.

Do you want to continue? [Y/n]

Get:1 http://ftp.srce.hr/srce-debian/ srce-jessie/main aosi-aai all 3.3.6 [82.2 kB]

Get:2 http://ftp.srce.hr/srce-debian/ srce-jessie/main aosi-www-aai all 1:2.0.2 [155 kB]

Fetched 237 kB in 0s (3,256 kB/s)

Selecting previously unselected package aosi-aai.

(Reading database ... 178846 files and directories currently installed.)

Preparing to unpack .../aosi-aai\_3.3.6\_all.deb ...

Unpacking aosi-aai (3.3.6) ...

Selecting previously unselected package aosi-www-aai.

Preparing to unpack .../aosi-www-aai\_1%3a2.0.2\_all.deb ...

Unpacking aosi-www-aai (1:2.0.2) ...

Processing triggers for systemd (215-17+deb8u13) ...

Processing triggers for man-db (2.7.0.2-5) ...

Processing triggers for libapache2-mod-php5 (5.6.40+dfsg-0+deb8u4) ...

Setting up aosi-aai (3.3.6) ...

AAI: AOSI in standalone mode.

Installing certificate key from /etc/ssl/private/ssl-cert-snakeoil.key to /etc/aosi/certs/aosi\_key.pem...

Installing certificate from /etc/ssl/certs/ssl-cert-snakeoil.pem to /etc/aosi/certs/aosi\_cert.pem...

Synchronizing state for aosi-aai.service with sysvinit using update-rc.d...

Executing /usr/sbin/update-rc.d aosi-aai defaults

Executing /usr/sbin/update-rc.d aosi-aai enable

Created symlink from /etc/systemd/system/aosi.service to /lib/systemd/system/aosi-aai.service.

Setting up aosi-www-aai (1:2.0.2) ...

AAI: Backing up old configuration file /etc/aosi-www/config.php to /var/backups/aai/

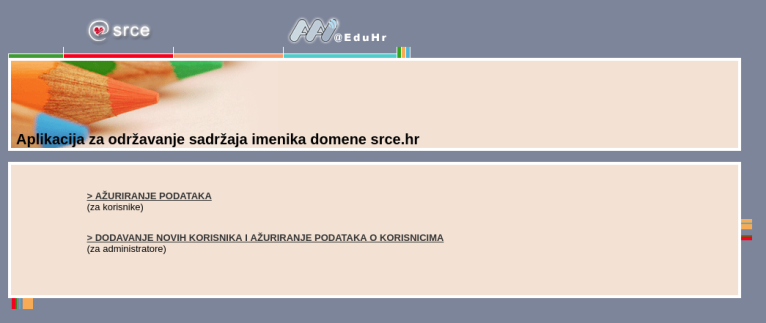
Enabling conf aosi-www-aai.

Adding user `aosi' to group `www-data' ...

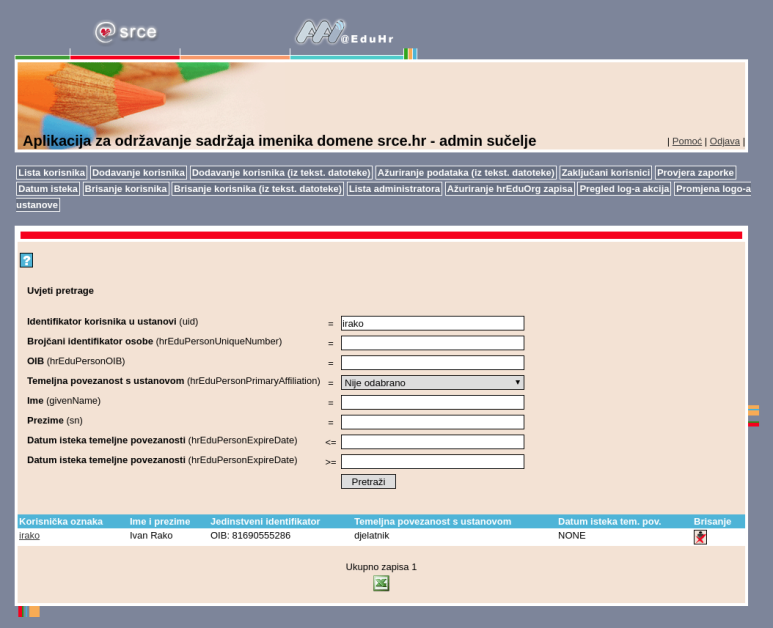
Adding user aosi to group www-data

Done.

Nakon instalacije, web–sučelju je moguće pristupiti na adresi <http://localhost/ldap/>.



Svaki korisnik može ažurirati podatke o sebi, dok administrator može administrirati cijeli imenik i sve podatke u njemu. Na sljedećoj slici nalazi se primjer pretraživanja korisnika:



3.1.5.6. Prebacivanje podataka LDAP imenika s poslužitelja na poslužitelj

Podaci LDAP imenika nalaze se u binarnom obliku u direktoriju **/var/lib/ldap/.** Naredbom ls -al /var/lib/ldap prikazat će se sadržaj toga direktorija:

# ls -al /var/lib/ldap

total 144

drwx------ 2 openldap openldap 4096 Aug 23 13:10 .

drwxr-xr-x 64 root root 4096 Aug 23 13:34 ..

-rw------- 1 openldap openldap 131072 Aug 23 14:40 data.mdb

-rw-r--r-- 1 root root 96 Aug 23 13:08 DB\_CONFIG

-rw------- 1 openldap openldap 8192 Aug 23 14:40 lock.mdb

Ako želimo premjestiti cijeli imenik, potrebno je napraviti tekstualnu kopiju (engl. dump) tih podataka u LDIF datoteku. To je moguće naredbom slapcat: Na novom poslužitelju potrebno je instalirati programsku podršku (instalacija paketa openldap-aai koja će instalirati sve potrebne pakete). Nakon toga potrebno je zaustaviti servis slapd korištenjem naredbe systemctl: # systemctl stop slapd

Sada je potrebno obrisati datoteke iz /var/lib/ldap/\*.mdb koje su nastale tijekom instalacije, korištenjem naredbe rm

# rm/var/lib/ldap/\*.mdb

Nakon što su napravljene radnje potrebne za uvoz LDIF datoteke, sljedećom slapadd naredbom uvozimo željenu datoteku.

# slapadd -l /tmp/srce.ldif

\_#################### 100.00% eta none elapsed none fast!

Closing DB...

Za kraj, potrebno je postaviti ispravno vlasništvo nad datotekama i zatim možemo pokrenuti servis slapd:

# chown -R openldap:openldap /var/lib/ldap/\*.mdb

# systemctl start slapd

Time je završeno premještanje OpenLDAP imenika na novi poslužitelj.

3.2. RADIUS

RADIUS (Remote Authentication Dial In User Service) je mrežni protokol, na portu 1812, koji omogućuje centralizirano upravljanje autentikacijom, autorizacijom i administracijom (engl. Authentication, Authorization and Accounting, AAA) korisnika prilikom spajanja i korištenja mrežnih usluga. Pojavio se 1991. godine kao protokol za autentikaciju i administraciju korisnika na mrežnim pristupnim uređajima. Zbog široko dostupne podrške i sveprisutnosti protokola RADIUS često koriste davatelji internetskih usluga kao i mnoge organizacije za upravljanje pristupom Internetu, privatnim i bežičnim mrežama ili integriranim e-mail uslugama. Te mreže mogu sadržavati modeme, DSL-ove, bežićne pristupne točke (engl. wireless access points), virtualne privatne mreže (VPN) i sl. RADIUS je protokol koji radi na principu komunikacije klijent-poslužitelj, spada u skupinu protokola aplikacijskoga sloja i može koristi TCP ili UDP (User Datagram Protocol) transportni protokol. Poslužitelji za udaljeni pristup (engl. Remote Access Server), poslužitelji virtualne privatne mreže (engl. Virtual Private Network server), mrežni preklopnici (engl. switch) i mrežni pristupni poslužitelji (engl. Network Access Server) su sučelja koja nadziru pristup mreži i sadrže komponentu RADIUS klijenta koja komunicira s RADIUS poslužiteljem. RADIUS poslužitelj se obično izvršava kao pozadinski proces na računalu s UNIX/Linux ili Windows operacijskim sustavom. Poslužitelj ima tri funkcije:

* autenticirati korisnike ili uređaje prije odobravanja pristupa mreži,
* autorizirati korisnike ili uređaje za određene mrežne usluge i
* pratiti aktivnosti korisnika tih usluga.

3.2.1. Protokol RADIUS

RADIUS klijent je obično uređaj koji pristupa mreži ili NAS-u (Network Access Server), a RADIUS poslužitelj je servisni proces (engl. daemon) koji se izvodi neovisno o operacijskom sustavu. Prilikom prijave u mrežu, korisnik šalje svoje podatke RADIUS klijentu koji potom izmjenjuje RADIUS poruke specifičnoga formata s RADIUS poslužiteljem. Svrha tih poruka jest ostvarivanje triju funkcija „AAA“ koncepta: autentikacije, autorizacije i administracije korisnika.

Autentikacija je proces kojim se potvrđuje korisnikov digitalni identitet, obično putem neke vrste identifikatora i pripadnih podataka. Primjeri tih podataka su lozinke, tokeni, digitalni certifikati i brojevi telefona.

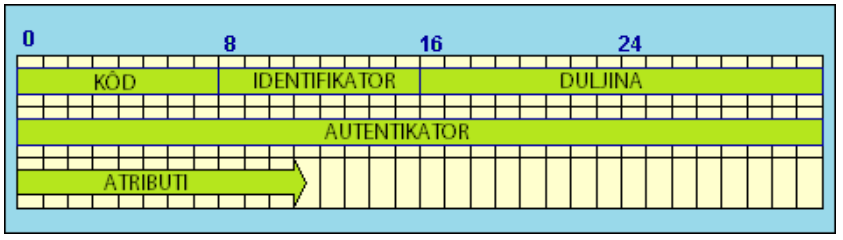
Autorizacijom se utvrđuje je li određeni entitet ovlašten izvoditi neku aktivnost (što se najčešće provodi prijavom pomoću lozinke). Autorizacija se može provoditi nizom ograničenja poput vremenskog, ograničenja fizičke lokacije ili ograničenja protiv višestrukih prijava istog entiteta ili korisnika. Primjeri tipova usluga jesu filtriranje IP adrese, dodjeljivanje adrese, dodjeljivanje puta usmjeravanja, kvaliteta usluge (QoS), diferencijalne usluge, kontrola pojasne širine, upravljanje prometom, obavezno tuneliranje do određene krajnje točke i enkripcija.

Accounting ili administracija korisnika jest proces praćenja korištenja mrežnih resursa. Ti podaci se mogu koristiti za upravljanje, planiranje, naplaćivanje usluge te u druge (specifične) svrhe. Accounting se u stvarnom vremenu odnosi na podatke koji se dostavljaju za vrijeme korištenja resursa. Skupni accounting (engl. batch accounting) se odnosi na podatke koji se čuvaju i kasnije dostavljaju pružatelju mrežne usluge. Podaci koji se obično prikupljaju su identitet korisnika, vrsta pružene usluge, kad je usluga počela i kad je završila.

RADIUS poruke se izmjenjuje kad korisnik želi pristupiti mreži čijim pristupom upravlja RADIUS klijent. Klijent izmjenjuje poruke s RADIUS poslužiteljem, čime se odobravaju ili odbijaju zahtjevi korisnika. RADIUS poruka sadrži 8-bitni kôd poruke, 8-bitni identifikator, 16-bitnu duljinu poruke, 32-bitni autentikator i, opcionalno, atribute. Format poruke je prikazan na slici 1. Kôd određuje tip poruke, a moguće vrijednosti su:

* zahtjev za pristupom (engl. Acces-Request),
* odobren pristup (engl. Access-Accept),
* odbijen pristup (engl. Access-Reject),
* zahtjev za praćenje korištenja mrežnih resursa (engl. Accounting-Request),
* odgovor za praćenje korištenja mrežnih resursa (engl. Accounting-Response),
* (11) osporavanje pristupa (engl. Access-Challenge),
* (12) status poslužitelja (engl. Status-Server),
* (13) status klijenta (engl. Status-Client),
* (255) rezervirano (engl. Reserved).

Identifikator je vrijednost koja omogućuje RADIUS klijentu da poveže RADIUS odgovor s ispravnim neispunjenim zahtjevom, dok „duljina“ označava duljinu poruke uključivo sa zaglavljem. Polje autentikator koristi se za provjeru autentičnosti odgovora od RADIUS poslužitelja, a zaštićeno je lozinkom pomoću algoritma za šifriranje. U polje atributa upisuje se proizvoljni broj atributa kao što su na primjer korisničko ime (engl. User-Name) i lozinka (engl. User-Password). RADIUS nudi 256 mogućih atributa, od kojih su 49 još nedodijeljena.



3.2.1.1. Razmjena RADIUS poruka

U nastavku će se prikazati najčešći oblik komunikacije RADIUS porukama, zahtjev za pristupom na temelju korisničkog imena i lozinke. Dvije strane u komunikaciji su klijent koji želi ovjeriti pristupne podatke koje je dobio od korisnika i poslužitelj koji ima pristup bazi podataka s podacima o korisnicima. Baza podataka može biti tekstualna datoteka, SQL relacijska baza podataka, LDAP, Active Directory itd.

Slijede koraci koji se poduzimaju tijekom komunikacije.

1. Klijent stvara poruku Access-Request koja od atributa mora sadržavati barem korisničko ime i lozinku. Polje identifikatora poruke nije specificirano protokolom RADIUS već je riječ o jednostavnom brojaču koji je stvoren na klijentu i koji se povećava sa svakim zahtjevom. Poruka sadrži autentikator zahtjeva (engl. Request Authenticator), što je zapravo slučajno odabran niz bitova duljine 256. Osim atributa s lozinkom, cijela poruka je nezaštićena. Lozinka je zaštićena pomoću zajedničkog ključa klijenta i poslužitelja. Zajednički ključ je niz znakova koji služi kao lozinka između RADIUS klijenta i poslužitelja, klijenta i posrednika (engl. proxy) ili posrednika i poslužitelja. Može se dobiti programom za generiranje i mora biti poznat objema stranama prije početka komunikacije.
2. Poslužitelj prima poruku Access-Request i provjerava posjeduje li zajednički ključ toga klijenta, o čemu ovisi obrađuje li se zahtjev. Ako poslužitelj ima isti ključ kao i klijent, može dešifrirati korisničku lozinku. Nakon toga traži u bazi podataka dobiveno korisničko ime i lozinku kako bi ih ovjerio. Ako su podaci valjani, poslužitelj šalje klijentu poruku Access-Accept. Ako nisu valjani, poslužitelj šalje poruku Access-Reject.
3. Obje poruke koriste vrijednost identifikatora iz dobivene Access-Request poruke, a polje autentikatora postavljaju u vrijednost autentikatora odgovora (engl. Response Authenticator).
4. Nakon što klijent primi odgovor, koristi polje identifikatora da ga poveže s neispunjenim zahtjevom. U slučaju kad nema zahtjeva s istim identifikatorom, poruka se zanemaruje. U suprotnom, klijent provjerava vrijednost polja Response Authenticator, računajući ga na isti način na koji je to učinio poslužitelj. Ako zaprimljena vrijednost nije ista izračunatoj, poruka se zanemaruje.
5. Ako je klijent primio ovjereni Access-Accept paket, smatra se da su korisničko ime i lozinka ispravni te je korisnik ovjeren. Ako je klijent primio ovjereni Access-Reject paket, smatra se da su korisničko ime i/ili lozinka neispravni te korisnik nije ovjeren.

Opisana komunikacija se može prikazati na sljedećoj slici.



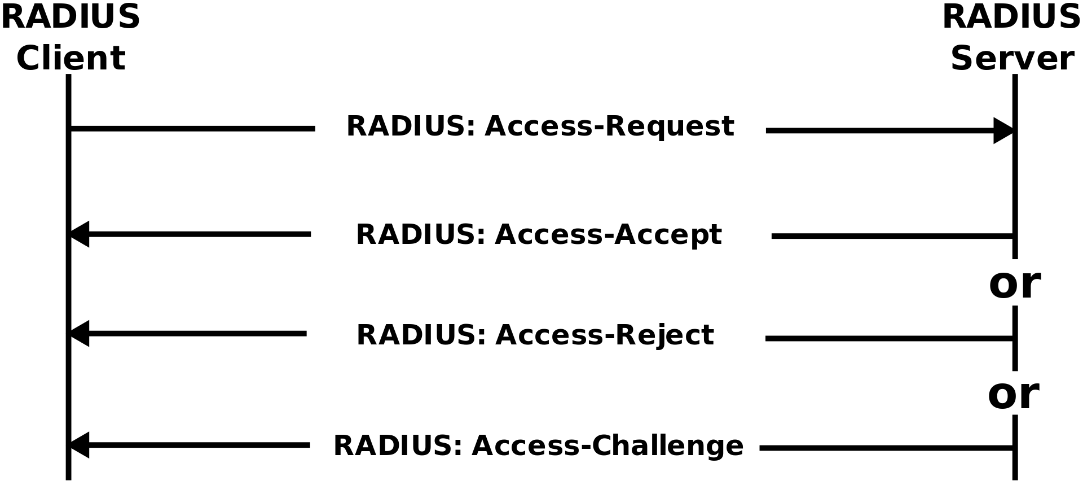
3.2.1.2. Autentikacija i autorizacija

Korisničko računalo šalje poslužitelju zahtjev za pristup određenim mrežnim resursima (npr. web–stranici, FTP arhivi, bazi podataka ili privatnoj mreži) koristeći svoje identifikacijske podatke specifične za tu mrežu. RADIUS klijent šalje poruku Access-Request RADIUS poslužitelju tražeći autorizaciju za pristup putem protokola RADIUS. Zahtjev za autorizacijom sadrži pristupne podatke, obično u obliku korisničkog imena i lozinke ili sigurnosnoga certifikata korisnika. Podaci mogu uključivati i druge podatke koje klijent ima o korisniku poput IP adrese, broja telefona i detalja o korisnikovu fizičkom mjestu priključivanja.

RADIUS poslužitelj vraća jedan od tri moguća odgovora na zahtjev (prikazano na slici):

* Access-Reject – korisniku je bezuvjetno osporen pristup svim traženim mrežnim resursima. Razlog tomu mogu biti nemogućnost dokazivanja identiteta, nepoznat ili neaktivan korisnički račun.
* Access-Challenge – zahtjevaju se dodatni podaci, poput naknadne lozinke, PIN-a i sl. Ova poruka koristi se u složenijim autentikacijskim dijalozima gdje se uspostavlja sigurni tunel između korisničkog računala i RADIUS poslužitelja tako da se pristupni podaci skrivaju od NAS-a.
* Access-Accept – korisniku je odobren pristup. Kad je korisnik autenticiran, RADIUS poslužitelj će provjeriti je li korisnik autoriziran za korištenje tražene mrežne usluge. Korisniku tako može, na primjer, biti dozvoljen pristup poslovnoj bežičnoj mreži, ali ne i VPN-u.

Sva tri odgovora mogu sadržavati atribut Reply-Message koji daje razlog odbijanja, zahtjev za dodatnim podacima ili poruku dobrodošlice.



3.2.2. Servis FreeRADIUS

Postoji nekoliko implementacija RADIUS poslužitelja otvorenoga kôda. Jedan od najčešće korištenih je FreeRADIUS, koji je izdan pod licencom GPL. FreeRADIUS je brz, bogat mogućnostima, modularan i skalabilan.

Sveučilišni računski centar (Srce) održava programski paket freeradius-aai koji donosi potrebnu konfiguraciju da biste svoj RADIUS poslužitelj uskladili s infrastrukturom AAI@EduHr. Sljedećom narebom instalirat ćemo taj paket:

# apt-get install freeradius-aai

Reading package lists... Done

Building dependency tree

Reading state information... Done

The following extra packages will be installed:

freeradius freeradius-common freeradius-ldap freeradius-utils libfreeradius2

Suggested packages:

freeradius-postgresql freeradius-mysql freeradius-krb5

The following NEW packages will be installed:

freeradius freeradius-aai freeradius-common freeradius-ldap freeradius-utils libfreeradius2

0 upgraded, 6 newly installed, 0 to remove and 6 not upgraded.

Need to get 415 kB/1,044 kB of archives.

After this operation, 3,945 kB of additional disk space will be used.

Do you want to continue? [Y/n]

Get:1 http://ftp.hr.debian.org/debian/ jessie/main libfreeradius2 amd64 2.2.5+dfsg-0.2+deb8u1 [107 kB]

Get:2 http://ftp.hr.debian.org/debian/ jessie/main freeradius-common all 2.2.5+dfsg-0.2+deb8u1 [229 kB]

Get:3 http://ftp.hr.debian.org/debian/ jessie/main freeradius-utils amd64 2.2.5+dfsg-0.2+deb8u1 [79.6 kB]

Fetched 415 kB in 0s (6,522 kB/s)

Preconfiguring packages ...

Selecting previously unselected package libfreeradius2.

(Reading database ... 178996 files and directories currently installed.)

Preparing to unpack .../libfreeradius2\_2.2.5+dfsg-0.2+deb8u1\_amd64.deb ...

Unpacking libfreeradius2 (2.2.5+dfsg-0.2+deb8u1) ...

Selecting previously unselected package freeradius-common.

Preparing to unpack .../freeradius-common\_2.2.5+dfsg-0.2+deb8u1\_all.deb ...

Unpacking freeradius-common (2.2.5+dfsg-0.2+deb8u1) ...

Selecting previously unselected package freeradius.

Preparing to unpack .../freeradius\_2.2.5+dfsg-0.2+deb8u1\_amd64.deb ...

Unpacking freeradius (2.2.5+dfsg-0.2+deb8u1) ...

Selecting previously unselected package freeradius-ldap.

Preparing to unpack .../freeradius-ldap\_2.2.5+dfsg-0.2+deb8u1\_amd64.deb ...

Unpacking freeradius-ldap (2.2.5+dfsg-0.2+deb8u1) ...

Processing triggers for man-db (2.7.0.2-5) ...

Processing triggers for systemd (215-17+deb8u13) ...

Setting up libfreeradius2 (2.2.5+dfsg-0.2+deb8u1) ...

Setting up freeradius-common (2.2.5+dfsg-0.2+deb8u1) ...

Adding user freerad to group shadow

Setting up freeradius (2.2.5+dfsg-0.2+deb8u1) ...

update-rc.d: warning: start and stop actions are no longer supported; falling back to defaults

Updating default SSL certificate settings, if any...

Adding user freerad to group ssl-cert

Generating DH parameters, 1024 bit long safe prime, generator 2

This is going to take a long time

Setting up freeradius-ldap (2.2.5+dfsg-0.2+deb8u1) ...

Processing triggers for systemd (215-17+deb8u13) ...

Selecting previously unselected package freeradius-aai.

(Reading database ... 179542 files and directories currently installed.)

Preparing to unpack .../freeradius-aai\_2.2.5~srce4\_all.deb ...

Unpacking freeradius-aai (2.2.5~srce4) ...

Selecting previously unselected package freeradius-utils.

Preparing to unpack .../freeradius-utils\_2.2.5+dfsg-0.2+deb8u1\_amd64.deb ...

Unpacking freeradius-utils (2.2.5+dfsg-0.2+deb8u1) ...

Setting up freeradius-aai (2.2.5~srce4) ...

AAI: Backing up old /etc/freeradius/modules/eap-aai to /var/backups/freeradius/...

AAI: Backing up old /etc/freeradius/modules/ldap-aai to /var/backups/freeradius/...

AAI: Backing up old /etc/freeradius/attrs.pre-proxy to /var/backups/freeradius/...

AAI: Backing up old /etc/freeradius/sites-available/aai to /var/backups/freeradius/...

AAI: Creating CA root key and request...

AAI: Self sign request and include certificate...

AAI: Self sign CA root certificate DER format create...

AAI: Create server key and request for certificate...

AAI: Sign server certificate...

Setting up freeradius-utils (2.2.5+dfsg-0.2+deb8u1) ...

U datoteci /etc/freeradius/clients.conf dodaje se popis RADIUS klijenata. Ako ne postoji klijent localhost, dodajte na kraj datoteke sljedeći blok (zajednički ključ za klijenta i poslužitelja u ovom primjeru jest testing123):

client 127.0.0.1 {

secret = testing123

shortname = localhost

}

Nakon svake promjene konfiguracijskih parametara u direktoriju /etc/freeradius potrebno je ponovno pokrenuti servis freeradius:

# systemctl restart freeradius

Sada je naredbom radtest moguće isprobati radi li autentikacija. Sintaksa naredbe radtest je:

radtest <korisničko ime> <zajednički ključ>

# radtest testko ispravnalozinka localhost:1812 1 testing123

Sending Access-Request of id 232 to 127.0.0.1 port 1812

User-Name = "testko"

User-Password = "ispravnalozinka"

NAS-IP-Address = 161.53.0.183

NAS-Port = 1

Message-Authenticator = 0x00000000000000000000000000000000

rad\_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=232, length=53

Callback-Number = "testko@srce.hr"

Configuration-Token = "djelatnik"

Connect-Info = "NONE"

RADIUS klijent je poslao poruku Access-Request i proslijedio je atribute poput User-Name, User-Password, NAS-IP-Address, NAS-Port. RADIUS poslužitelj je obradio taj zahtjev, provjerio postoji li korisnik u LDAP-u i je li lozinka ispravna, te vratio poruku Access-Accept. Uz tu poruku, vratio je i nekoliko atributa poput Callback-Number, Configuration-Token i Connect-Info. Isprobat ćemo sada pokretanje naredbe radtest s neispravnom lozinkom:

# radtest testko neispravnalozinka localhost:1812 1 testing123

Sending Access-Request of id 151 to 127.0.0.1 port 1812

User-Name = "testko"

User-Password = "neispravnalozinka"

NAS-IP-Address = 161.53.0.183

NAS-Port = 1

Message-Authenticator = 0x00000000000000000000000000000000

rad\_recv: Access-Reject packet from host 127.0.0.1 port 1812, id=151, length=20

RADIUS poslužitelj je ustanovio da je lozinka neispravna, i vratio je poruku Access-Reject. Time nije omogućeno povezivanje korisnika na sustav.

3.2.3. Autentikacija osnovnih servisa korištenjem protokola LDAP i RADIUS

Za autentikaciju različitih servisa putem protokola LDAP mogu se koristiti dva modula: **pam\_ldap** i **pam\_radius**. Preporuka AAI@EduHr službe jest uporaba modula **pam\_radius**.

Način rada u ovom slučaju je posredan, jer **pam\_radius** kontaktira RADIUS poslužitelj, koji zatim komunicira s LDAP poslužiteljem.

Uz pretpostavku da se radi o *Debianovoj* distribuciji, potrebno je instalirati paket **libpam-radius-auth** koji donosi potreban PAM modul. Instalacija je standardna, korištenjem naredbe: # apt-get install libpam-radius-auth

U konfiguraciji FreeRADIUS-a prijavite poslužitelj kao klijenta, u našem slučaju, klijent na lokalnom računalu (engl. localhost):

client 127.0.0.1 {

secret = tajna\_lozinka

shortname = localhost

}

Paket freeradius-aai donosi klijenta localhost s lozinkom (engl. secret) testing123, nalazi se u datoteci /etc/freeradius/clients.conf. Lozinka se treba prenijeti i u konfiguraciju pam\_radius-a u datoteci /etc/pam\_radius\_auth.conf:

# server[:port] shared\_secret timeout (s)

127.0.0.1:1812 neki\_secret 3

Time smo obavili sve predradnje za autentikaciju servisa preko protokola RADIUS. Konfiguracijske datoteke PAM-a nalaze se u direktoriju /etc/pam.d/. Ako želite sve servise autenticirati preko RADIUS-a, u datoteci /etc/pam.d/common-auth, bez izmjene ostalih datoteka, zakomentirajte sve retke i na kraj datoteke dodajte:

auth sufficient pam\_radius\_auth.so

auth required pam\_unix.so try\_first\_pass

Time smo postigli da se autentikacija korisnika obavlja preko RADIUS poslužitelja, a tek u slučaju neuspješne autentikacije pita se pam\_unix (odnosno traži unos zaporke navedene u datoteci /etc/shadow). Svaki servis ima svoju konfiguracijsku datoteku i može se zasebno podešavati. Na primjer, za Secure shell u /etc/pam.d/ssh zakomentirajte redak: #@include common-auth i dodajte dva nova:

auth sufficient pam\_radius\_auth.so

auth required pam\_unix.so try\_first\_pass

U /etc/pam.d nalazi se konfiguracija i za druge servise. Na isti način možete unijeti i konfiguraciju, na primjer, za ftp, imap, pop itd.

3.3. Vježba 4. LDAP

1. Na sustavu **server1** instalirajte programski paket **openldap-aai**.
2. Dodati korisnika **testko** u imenik korištenjem naredbe **ldapadd**.
3. Pretražiti LDAP imenik te provjeriti da li postoji korisnik **testko**.
4. Promijeniti neki atribut korisnika **testko**.
5. Obrisati korisnika **testko** iz LDAP imenika.
6. Napraviti dump cijele LDAP baze korištenjem naredbe **slapcat**.

Napraviti migraciju LDAP imenika, tj. importirati imenik korištenjem naredbe **slapadd**. Pripazite da su vlasništva datoteka ispravna

**Napomena**

Prilikom rješavanja ove vježbe služite se uputama iz teorijskog dijela priručnika.

1. Elektronička pošta

**Trajanje poglavlja:**

**180 min**



Po završetku ovoga poglavlja moći ćete:

* opisati protokol SMTP
* koristiti servis Postfix za slanje elektroničkih poruka
* koristiti servis Mailman za upravljanje mailing listama
* koristiti alat procmail za filtriranje elektroničke pošte
* opisati protokole POP3 i IMAP
* koristiti servis Dovecot za primanje elektroničke pošte.

Ova cjelina obrađuje načine primanja i slanja elektroničke pošte protokolima SMTP (Simple Mail Transfer Protocol), POP3 (Post Office Protocol 3) i IMAP (Internet Message Access Protocol) koristeći servise Postfix i Dovecot. Dodatno je opisan i Mailman, softver za upravljanje mailing listama koji ima i opcije arhiviranja, grupiranja i sl.

Na kraju poglavlja bit će govora o načinima filtriranja elektroničke pošte, a detaljnije će biti objašnjen Procmail alat koji služi za filtriranje, sortiranje i pohranjivanje pošte na temelju sadržaja njihovoga zaglavlja ili samoga tijela poruke.

4.1. SMTP

Protokol SMTP

Poslužitelji elektroničke pošte dijele se na dvije kategorije, na one koji šalju i one koji primaju elektroničku poštu. Najčešći protokol za slanje elektroničke pošte je SMTP, a za primanje se koriste protokoli POP3 i IMAP.

SMTP je razvijen 1980. godine kao protokol za prijenos elektroničke pošte, a temelji se na ASCII tekstualnim naredbama. Kako bi se proširile njegove mogućnosti, 1995. je definiran ESMTP ( Extended SMTP) protokol, danas najkorišteniji protokol za slanje elektroničke pošte.

Korisničko računalo, na kojem je instaliran neki od klijenata elektroničke pošte (engl. Mail User Agent – MUA, mail client) za slanje i primanje elektroničke pošte (npr. Microsoft Outlook, Mozilla Thunderbird, Mutt itd.) šalje elektroničku poruku prema SMTP poslužitelju (npr. smtp.srce.hr) koji osluškuje zahtjeve na TCP portu 25. Svaka elektronička poruka mora imati adresu primatelja i pošiljatelja, dok su sadržaj poruke i privitak opcionalni. SMTP poslužitelj prvo provjerava domenu primatelja. Ako je ista kao i pošiljatelj, elektronička pošta se prosljeđuje direktno na POP3 ili IMAP poslužitelj. Ako je domena pošiljatelja različita, SMTP poslužitelj mora provjeriti DNS (Domain Name System) zapise te domene koji specificiraju IP adresu poslužitelja zaduženoga za primanje elektroničke pošte. Elektronička poruka na svome putu može promijeniti nekoliko poslužitelja prije nego što dođe do svojeg odredišta (engl. email relaying). Kada poruka dođe do domenskog SMTP poslužitelja, provjeravaju se adresa i domena primatelja, te se poruka prosljeđuje POP3 ili IMAP poslužitelju. Primatelj u svome klijentu elektroničke pošte provjerava stanje svoga sandučića i čita pristiglu elektroničku poštu. Osim programa za čitanje i slanje elektroničke pošte koji se instaliraju na operacijski sustav, poštu je također moguće čitati i pomoću web-sučelja (Roundcube, RainLoop i sl.).

Neke od SMTP naredbi koje se koriste prilikom komunikacije između SMTP klijenta i poslužitelja:

HELO – prva naredba kojom se klijent predstavlja SMTP poslužitelju nazivom svoje domene (npr. klijent.srce.hr)

MAIL FROM – definira se pošiljatelj elektroničke poruke i na tu adresu elektroničke pošte stižu eventualne poruke o greškama

RCPT TO – definira jednog ili više primatelja poruke

DATA – označava početak poruke (ovaj podatak, za razliku od prijašnjih, nije dio zaglavlja).

Svaka elektronička poruka sastoji se od zaglavlja poruke, koje sadrži polja From, To, Cc, Date, Subject itd., te tijela poruke koje je odvojeno od zaglavlja praznim redom.

Primjer komunikacije između SMTP klijenta i poslužitelja:

poslužitelj: 220 smtp.srce.hr ESMTP Postfix

klijent: HELO klijent.srce.hr

poslužitelj: 250 smtp.srce.hr

klijent: MAIL FROM:<marko@srce.hr>

poslužitelj: 250 Ok

klijent: RCPT TO:<mario@srce.hr>

poslužitelj: 250 Ok

klijent: RCPT TO:<maja@srce.hr>

poslužitelj: 250 Ok

klijent: DATA

poslužitelj: 354 End data with .

klijent: From: "Marko" <marko@srce.hr>

klijent: To: Mario <mario@srce.hr>

klijent: Cc: maja@srce.hr

klijent: Date: Tue, 30 May 2019 12:12:12 +0200

klijent: Subject: Testna poruka

klijent:

klijent: Pozdrav Mario!

klijent: Sve najbolje želim ti na novom radnom mjestu,

klijent: Marko

klijent: .

poslužitelj: 250 Ok: queued as 58971

klijent: QUIT

poslužitelj: 221 Bye

Kada je početkom 70-ih godina razvijen SMTP, nije se vodilo računa o autentikaciji korisnika prilikom slanja elektroničke pošte. To znači da svatko može poslati e-poštu sa SMTP poslužitelja bez prethodne autentikacije. Iako ovo u početku nije predstavljalo problem, e-mail poruke neželjenoga sadržaja su krajem 90-ih postale jedna od najvećih pošasti na Internetu. SMTP i ESMTP ne definiraju metode autentikacije, već je potrebno koristiti razne dodatke za tu svrhu.

Najčešće korišteni mehanizmi SMTP autentikacije su:

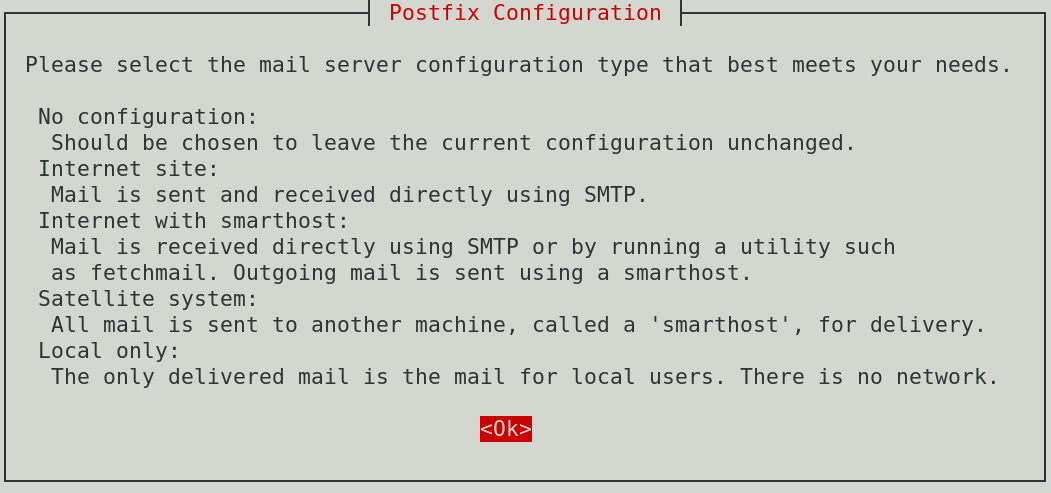
* PLAIN
* LOGIN
* CRAM MD5
* DIGEST MD5
* NTLM
* GSSAPI Servis Postfix

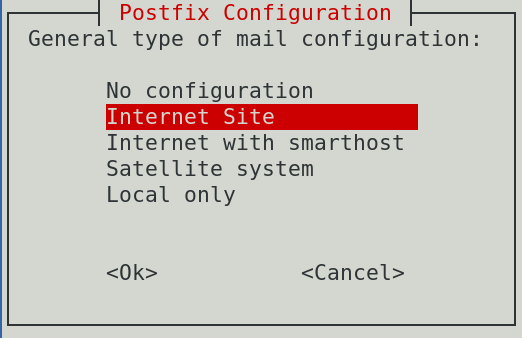
Postfix je besplatni MTA (Mail Transfer Agent) otvorenoga kôda koji se koristi za usmjeravanje (engl. routing) i slanje elektroničke pošte. Protokol SMTP može se koristiti na lokalnoj mreži ili može slati elektroničku poštu između poslužitelja na Internetu. Ne koristi se za protokole POP3 i IMAP, tj. za dostavljanje elektroničke pošte krajnjem korisniku. Postfix se koristi na oko 34 % javno dostupnih poslužitelja elektroničke pošte, a može se instalirati na većini operacijskih sustava baziranih na Unix arhitekturi. Postfix je po dizajnu modularan, čime se postiže fleksibilnost i brzina. Modularnost se koristi i pri implementiranju sigurnosti postfixa, pa tako svaki modul ima svoj userid i najmanja moguća prava koja su mu potrebna za ispravan rad. Ako napadač kompromitira jedan modul, koji je izoliran od ostalih, neće moći kompromitirati druge module, a niti poslužitelj. Također, svaki od modula se može isključiti s čime se pojednostavnjuje instalacija, održavanje i smanjuje vektor napada. Pouzdanost servisa postiže se detektiranjem preopterećenosti sustava praćenjem iskorištenja memorije i diska te prilagođavanjem rada. Tehnikom ograničenja broja procesa za korištenje datotečnoga sustava, servis Postfix postiže bolje rezultate, a da pri tome ne opterećuje sustav. Uz Postfix mogu se implementirati i drugi servisi koji tvore jednu cjelinu, poslužitelj elektroničke pošte. Za protokole POP3 i IMAP koristi se Dovecot, a za zaštitu od neželjenih poruka i zlonamjernoga softvera koriste se ClamAV, Amavis i SpamAssassin. Mailing listama može se upravljati programom Mailman. Instalacija servisa Postfix vrši se sljedećim naredbama:

# apt update

# apt install postfix

Tijekom instalacije prikazuju se sljedeće opcije konfiguracija:





Konfiguracija servisa Postfix

Dvije su glavne konfiguracijske datoteke, main.cf i master.cf, te se nalaze u direktoriju /etc/postfix. Konfiguracijske datoteke i direktorij moraju biti pod vlasništvom korisnika root, kako bi se spriječilo čitanje i mijenjanje postavki od strane drugih korisnika. Konfiguracije u datoteci main.cf koriste se na globalnoj razini osim ako nisu nadjačane u datoteci master.cf sa specifičnim postavkama za servisne procese (daemons) postfix. Ovisno o odabiru tijekom instalacijskog izbornika, konfiguracijske datoteke bit će ispunjene zadanim vrijednostima.

Konfiguracijski parametri definiraju se na sljedeći način: parametar = vrijednost

Vrijednost parametara iz gornjeg primjera može se pozivati dodavanjem znaka '$' ispred imena parametra: drugi\_parametar = $parametar

Na poslužitelj elektroničke pošte spajaju se klijenti, tj. korisnici kako bi poslali elektroničku poštu. Poslužitelj šalje elektroničku poštu za korisnike "korisnik@domena.hr" i prima elektroničku poštu za "korisnik@ime\_posluzitelja.domena.hr" i "korisnik@domena.hr".

Slijedi primjer konfiguracije za datoteku main.cf:

mydomain = domena.hr

myhostname = mail.domena.hr

myorigin = $mydomain

mydestination = $myhostname localhost.$mydomain localhost $mydomain

mynetworks = 127.0.0.1/8 192.168.100.1/24

inet\_interfaces = all

Slijedi objašnjenje konfiguracija:

|  |  |
| --- | --- |
| **Parametar** | **Objašnjenje** |
| myhostname | Puno ime poslužitelja (FQDN), npr. mail.domena.hr. |
| mydomain | Ime domene. Ako nije definirano, ime domene će biti vrijednost varijable $myhostname bez imena poslužitelja. |
| myorigin | Definira domenu izvorišta elektroničke poruke, tj. korisnik@domena.hr. Prema zadanim postavkama, myorigin parametar preuzima ime poslužitelja ako se eksplicitno ne definira. |
| mydestination | Definira za koje sve domene poslužitelj prihvaća elektroničku poštu, tj. ako je elektronička pošta naslovljena na domene koje su ovdje postavljene, one će biti i dostavljene u lokalni sandučić elektroničke pošte. |
| mynetworks | Definiraju se podmreže kojima je dozvoljeno usmjeravavnje (engl. relay) elektroničke pošte. |
| inet\_interfaces | Definiranje mrežnih sučelja koje će servis postfix koristiti. |

Nakon promjena u konfiguracijskim datotekama potrebno je pokrenuti sljedeću naredbu da bi se konfiguracije primijenile na servisu *postfix*: $ postfix reload

**Napomena**

1. Kako bi svako računalo u lokalnoj mreži moglo slati elektroničku poštu, moraju biti podešene DNS postavke poslužitelja elektroničke pošte, na primjer:
2. domena.hr IN MX 10 mail.domena.hr..

Dodavanje korisnika i *aliasa*

Za svaku adresu elektroničke pošte potrebno je imati i korisnika na operacijskom sustavu *Debian*, a servis *postfix* će se pobrinuti da korisnik prima elektroničku poštu. Za dodavanje korisnika potrebna je sljedeća naredba i upisivanje osnovnih informacija:

# adduser marko

Adding user `marko' ...

Adding new group `marko' (1001) ...

Adding new user `marko' (1001) with group `marko' ...

Creating home directory `/home/marko' ...

Copying files from `/etc/skel' ...

New password:

Retype new password:

passwd: password updated successfully

Changing the user information for marko

Enter the new value, or press ENTER for the default

Full Name []: Marko Srecic

Room Number []: 101

Work Phone []: 555

Home Phone []:

Other []:

Is the information correct? [Y/n] y

Aliasi se koriste za preusmjeravanje elektroničke poruke za lokalne primatelje. Ime novostvorenoga korisnika koristi se u formi elektroničke adrese, tj. marko@srce.hr, a ako je potrebno dodavanje i prezimena, koriste se aliasi. Tako će sva elektronička pošta naslovljena na marko.srecic@srce.hr biti dostavljena u lokalni sandučić za marko@srce.hr. Konfiguracija aliasa nalazi se u datoteci /etc/aliases u sljedećoj formi:

marko.srecic: marko

Elektronička pošta može biti dostavljena i na vanjsku domenu, samo je potrebno upisati punu adresu elektroničke pošte željenoga korisnika:

marko: marko.srecic@gmail.com

Moguće je napraviti i jedan alias za grupu korisnika, što će rezultirati time da svi korisnici na aliasu prime sadržaj poslan na adresu aliasa. U ovom slučaju radi se o grupnoj adresi svi@srce.hr:

svi: marko, ante, ivan@gmail.com, mario

Nakon svake izmjene u datoteci /etc/aliases potrebno je učitati promjene s naredbom: # newaliases

**Zanimljivi izvori**

* <http://www.postfix.org/>
* <https://en.wikipedia.org/wiki/Postfix_(software)>
* <https://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol>

4.2. Upravljanje mailing listama

Servis *mailman3*

Upravljanje mailing listama moguće je pomoću aliasa, ali kod dinamičnih okruženja, tj. kod mailing lista koje se često mijenjaju preporučeno je koristiti program za upravljanje mailing listama. Koristeći programe za upravljanje mailing listama korisnici mogu bez pomoći administratora uređivati liste i mijenjati njihove postavke. Program također može arhivirati elektroničke poruke te pregledavati i odobravati prije slanja.

Mailman3 je besplatan softver otvorenoga kôda za upravljanje mailing lista kojim se upravlja pomoću web-sučelja. Uz upravljanje lista, ugrađene su opcije arhiviranja, automatskoga procesiranja bounceova, filtriranja sadržaja, filtriranje neželjene elektroničke pošte itd.

**Instalacija i konfiguracija**

Prvo je potrebno instalirati servis postfix, apache i php, sa potrebnim ekstenzijama:

# apt install postfix mailutils apache2 php7.4-{bcmath,common,curl,fpm,gd,intl,mbstring,mysql,soap,xml,xsl,zip,cli}

Tijekom instalacije *postfixa* potrebno je odabrati opciju *Internet Site*. Slijedi primjer konfiguracije za datoteku main.cf nakon koje je potrebno ponovno pokrenuti servis *postfix*:

mydomain = domena.hr

myhostname = mail.domena.hr

myorigin = $mydomain

mydestination = $myhostname localhost.$mydomain localhost $mydomain

mynetworks = 127.0.0.1/8 192.168.100.1/24

inet\_interfaces = all

Meta-paket *mailman3-full* ima sve potrebne ovisnosti koje su potrebne za instalaciju servisa *mailman3*:  # apt install mailman3-full. Tijekom instalacije servisa *mailman3* potrebno je odgovoriti potvrdno na pitanje o kreiranju baze podataka za servis mailman3 i za mailman3-web i potrebno je odabrati bazu podataka *sqlite3*.

Za aktivaciju virtualnog hosta mailman3 potrebno je napraviti simbolički link na konfiguraciju, omogućiti ekstenziju *proxy\_uwsgi* i ponovno pokrenuti servis *apache2*:

# ln -s /etc/mailman3/apache.conf /etc/apache2/conf-enabled/mailman3.conf

# a2enmod proxy\_uwsgi

# systemctl restart apache2

Konfiguriranje Mailman3 za komunikaciju s mail servisom koji je na istom poslužetelju je jednostavno. U svoju datoteku mailman.cfg dodajte (ili uredite) sljedeće linije:

site\_owner: root@domena.hr

[mta]

incoming: mailman.mta.postfix.LMTP

outgoing: mailman.mta.deliver.deliver

lmtp\_host: 127.0.0.1

lmtp\_port: 8024

smtp\_host: localhost

smtp\_port: 25

configuration: python:mailman.config.postfix

Za spajanje servisa mailman sa servisom postfix potrebno je dodati sljedeće linije u datoteku /*etc/postfix/main.cf* i ponovno pokrenuti servis postfix:

recipient\_delimiter = +

unknown\_local\_recipient\_reject\_code = 550

owner\_request\_special = no

transport\_maps =

hash:/var/lib/mailman3/data/postfix\_lmtp

local\_recipient\_maps =

proxy:unix:passwd.byname $alias\_maps hash:/var/lib/mailman3/data/postfix\_lmtp

relay\_domains =

hash:/var/lib/mailman3/data/postfix\_domains

Za konfiguraciju servisa mailman-web potrebno je promijeniti adresu elektroničke pošte za administratora u datoteci: /etc/mailman3/mailman-web.py:

ADMINS = (

('Mailman Suite Admin', 'root@domena.hr'),

)

Ponovno pokrenite servise mailman3, mailman3-web i postfix.

Za kreiranje administratora za mailman potrebno je pokrenuti sljedeću naredbu i ispuniti tražena pitanja.

# mailman-web createsuperuser

Username (leave blank to use 'www-data'): admin

Email address: root@domena.hr

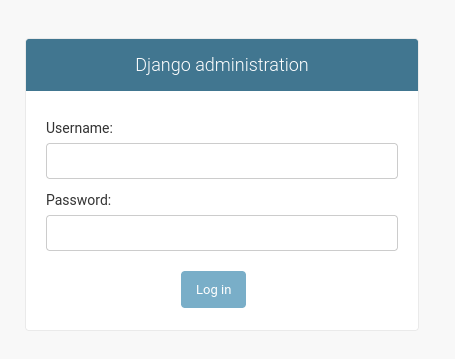
Password: linux1

Password (again): linux1

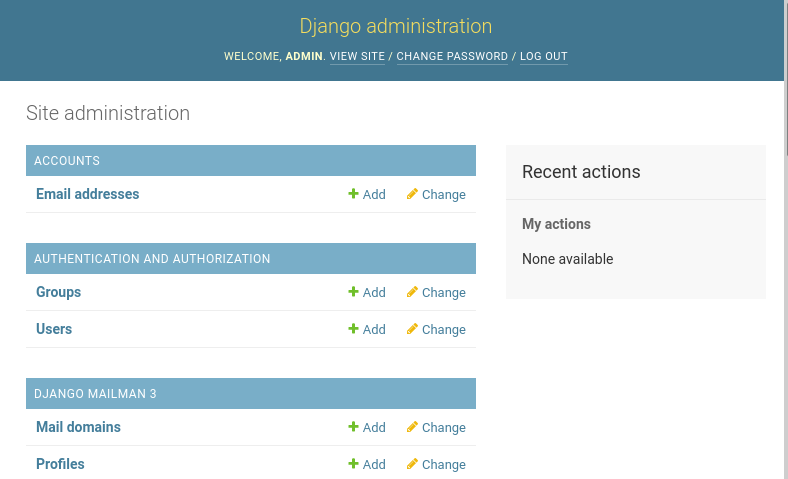
Superuser created successfully.

U web pregledniku upišite URL http://localhost/mailman3/admin i spojite se kao administrator.

localhost/mailman3/admin i spojite se kao administrator.



Nakon prijave pokazati će se sljedeće sučelje:



4.3. Filtriranje elektroničke pošte

**Filtriranje sadržaja elektroničke pošte**

Filtriranje je moguće prema sadržaju zaglavlja, tijela ili privitka elektroničke pošte. Koristi se za prosljeđivanje ili spremanje elektroničkih poruka u željene direktorije, za antivirusnu zaštitu i zaštitu protiv neželjene elektroničke pošte. Filteri mogu mijenjati, brisati, prosljeđivati, odgovarati ili označiti elektroničke poruke, ovisno u koju svrhu se filtriranje koristi.

Filtriranje je moguće na nekoliko načina:

**Filtriranje po tijelu i zaglavlju koristeći servis Postfix (Before-Queue Content FIlter, After-Queue Content Filter)**

Ovo filtriranje je najjednostavnije za implementaciju i najčešće se koristi za antivirusnu zaštitu, zaštitu od neželjene pošte i globalno filtriranje na razini cijele domene. Važno je napomenuti da ovakvo filtriranje ne može koristiti običan korisnik.

Može se koristiti za ponovno slanje elektroničke poruke nakon promjene sadržaja ili odredišne elektroničke adrese, brisanje ili zadržavanje elektroničke poruke ili odbijanje elektroničke poruke uz vraćanje pošiljatelju. Također, unutar postfix konfiguracijske datoteke master.cf moguće je kroz parametre smtpd\_sender\_restrictions i smtpd\_recipient\_restrictions napraviti osnovnu zaštitu od neželjene pošte koristeći samo rješenja otvorenoga kôda.

**Mail delivery agent**

Mail delivery agent (MDA) je program koji prima elektroničku poštu od MTA, u ovom slučaju od servisa postfix. Nakon što sortira elektroničke poruke, dostavlja ih u poštanski sandučić primatelja, tj. do klijenta elektroničke pošte. Korisnici koristeći alate kao što su sieve ili procmail mogu upravljati svojom konfiguracijom, ali moraju imati korisnički račun na operacijskom sustavu ili omogućen pristup web-sučelju.

**Klijenti elektroničke pošte**

Svaki klijent elektroničke pošte ima opcije filtriranja koje korisnik može postaviti prema svojim potrebama. Ovisno o mogućnostima klijenta, to može biti jednostavno filtriranje prema pošiljatelju ili napredno filtriranje za zaštitu od neželjene pošte ili zlonamjernoga softvera.

Za antivirusnu zaštitu, zaštitu od neželjene pošte i globalno filtriranje, na razini organizacije preporučuje se koristiti filtriranje pomoću servisa Postfix. Za filtriranje za korisnike gdje svatko može konfigurirati svoje parametre preporučuju se klijenti elektroničke pošte ili mail delivery agenti.

**Alat procmail**

Procmail je alat za filtriranje elektroničkih poruka koji se može koristiti na poslužitelju elektroničke pošte ili na klijentskim računalima. Alat procmail filtrira sve poruke prije nego što su dostavljene u poštanski sandučić korisnika. Elektroničke poruke se filtriraju na osnovi zaglavlja ili tijela poruke koristeći regularne izraze, zatim poruke prosljeđuju u odgovarajuće direktorije. Također je moguće prosljeđivati i dodavati retke u zaglavlje elektroničkih poruka.

Alat procmail instalira se sljedećom naredbom: # apt install procmail

**Korištenje na klijentima operacijskoga sustava Linux**

Za korištenje alata procmail na klijentima operacijskoga sustava Linux potrebno je stvoriti datoteku .*forward* u korisničkom home direktoriju sa sljedećim sadržajem: *|/usr/bin/procmail*

Ovakvom konfiguracijom datoteka .*forward* prosljeđuje sve elektroničke poruke alatu procmail, a u datoteci .*procmailrc* u direktoriju *home* definira se putanja do poštanskog sandučića i pravila za filtriranje elektroničkih poruka.

**Napomena**

Moguće je i globalno definirati procmail u main.cf datoteci pa nisu potrebne pojedinačne .*forward* konfiguracije.

**Datoteka .procmailrc**

Slijedi primjer konfiguracijskoga bloka koji je potrebno definirati na vrhu .procmailrc:

PATH=$HOME/bin:/usr/local/bin:/usr/bin:/bin

SHELL=/bin/sh

MAILDIR=$HOME/Maildir

DEFAULT=$MAILDIR/

LOGFILE=$MAILDIR/procmail.`date +%Y.%m`.log

Slijedi objašnjenje konfiguracija:

|  |  |
| --- | --- |
| **Mogućnost** | **Objašnjenje** |
| PATH | Putanje do ostalih programa, ako su potrebne alatu procmail. |
| SHELL | Putanja do ljuske. |
| MAILDIR | Putanja do poštanskog sandučića gdje će filtrirane elektroničke poruke biti spremljene. |
| DEFAULT | Putanja do poštanskog sandučića gdje će nefiltrirane elektroničke poruke biti spremljene. |
| LOGFILE | Putanja do datoteke dnevničkoga zapisa. |

Pravila za filtriranje

Tri su dijela svakoga procmail pravila (engl. procmail recipe) za filtriranje:

* ':0 Flags': Označava početak pravila. Zastavice (engl. flags) definiraju kako se pravilo izvršava. Ako zastavice nisu definirane, po zadanome se koriste zastavice "Hhb".
* '\*': Označava početak uvjeta za filtriranje, tj. definiranje regularnog izraza za pretraživanje.
* Redak koji definira ime datoteke gdje će se spremiti elektronička poruka.

Neke od zastavica su:

|  |  |
| --- | --- |
| **Mogućnost** | **Objašnjenje** |
| H | Pretražuje se zaglavlje (uključeno po zadanome). |
| B | Pretražuje se tijelo. |
| h | Zaglavlje se prosljeđuje u pipe, direktorij ili na neku drugu adresu elektroničke pošte (uključeno po zadanome). |
| b | Tijelo se prosljeđuje u pipe, u neki direktorij ili na drugu adresu elektroničke pošte (uključeno po zadanome). |
| D | Razlikuje mala i velika slova. |

Slijedi nekoliko primjera (komentari su označeni znakom ljestve „#”):

# elektroničku poštu s domenom @srce.hr ili @srce.com proslijedi   
u direktorij srce

:0

\* ^From:.\*(\@srce\.hr|\@srce\.com)

$MAILDIR/srce

# elektroničku poštu koja je poslana na svi@srce.hr proslijedi   
u direktorij „svi“

:0

\* ^To:.\*svi@srce.hr

$MAILDIR/svi

# elektroničku poštu koja je poslana s domene @srce.hr ili @srce.com te   
ima riječ „alert“ u naslovu proslijedi u direktorij „alert“

:0

\* ^From:.\*(\@srce\.hr|\@srce\.com)

\* ^Subject:.\*alert

$MAILDIR/alert

# svu ostalu elektroničku poštu koja ima riječ „alert“ u naslovu   
proslijedi u direktorij „alert“

:0

\* ^Subject:.\*alert

$MAILDIR/alert

# zadano pravilo koje svu ostalu elektroničku poštu prosljeđuje   
u direktorij $MAILDIR nije potrebno navoditi

:0

\* .\*

$MAILDIR

Dovoljno je samo spremiti promjene u datoteku .***procmailrc*** kako bi se nova pravila aktivirala.

**Zanimljivi izvori**

* <https://www.linux.com/news/process-your-email-procmail>
* <http://www.postfix.org/FILTER_README.html>
* <http://www.postfix.org/SMTPD_PROXY_README.html>
* <https://www.linuxbabe.com/mail-server/block-email-spam-postfix>

4.4. POP3 i IMAP

**Protokol POP3**

Protokol POP3, nakon korisničke autentikacije, uspostavlja vezu između korisničkog računala i poslužitelja elektroničke pošte. Sve elektroničke poruke preuzimaju se na korisničko računalo i ne ostaju na poslužitelju. Na ovaj način jedina kopija elektroničke poruke nalazi se na korisničkom računalu, te ako ne postoji pouzdana sigurnosna pohrana i ako se nešto dogodi s računalom, gubi se elektronička poruka. Također, nije moguće preuzimanje elektroničke poruke s nekim drugim uređajem. U današnje vrijeme neki klijenti elektroničke pošte imaju mogućnost da se poruka koja je preuzeta protokolom POP3 ne briše s poslužitelja ili da se obriše tek nakon određenog vremena.

Prilikom slanja elektronička poruka se čuva samo na računalu s kojeg je poslana, a s ostalih uređaja neće biti moguće vidjeti tu poruku. Ako se elektronička poruka obriše, ona će biti obrisana samo na računalu s kojeg se briše, tj. bit će prisutna na svim ostalim računalima koji su tu elektroničku poruku preuzeli.

Za preuzimanje elektroničke pošte protokolom POP3 koristi se port 110, dok se šifrirani protokol POP3S (POP3 Secure) koristi na portu 995.

Protokol POP3 je brz i robustan, ali s obzirom na njegove nedostatke, preporučuje se koristiti samo ako se poruke pregledavaju na jednom računalu.

**Protokol IMAP**

Koristeći protokol IMAP, klijentom elektroničke pošte pristupa se poslužitelju elektroničke pošte i pregledavaju se poruke direktno na poslužitelju. Poruke se ne brišu s poslužitelja pa im korisnik može pristupiti s više uređaja, a izvorna kopija ostaje na samom poslužitelju. U usporedbi s protokolom POP3, ovakav način više odgovara današnjem načinu pregledavanja elektroničke pošte.

Kada se elektronička poruka obriše na računalu, također se obriše i s poslužitelja, a svaka poslana poruka se spremi na poslužitelju. Zbog spremanja svake elektroničke poruke s privitcima na poslužitelju može doći do manjka diskovnoga prostora.

Za preuzimanje elektroničke pošte protokolom IMAP s kriptiranjem koristi se port 993, a bez kriptiranja port 143.

**Servis Dovecot**

Dovecot je program otvorenoga kôda za posluživanje elektroničke pošte protokolima IMAP i POP3. Koristi se na operacijskim sustavima Linux i razvijen je da bude siguran. Lagano se njime upravlja, brz je i koristi malo radne memorije.

Za instalaciju se koristi naredba: # apt install dovecot-imapd dovecot-pop3d

Glavna konfiguracijska datoteka je /etc/dovecot/dovecot.conf, a ostale konfiguracijske datoteke su u direktoriju /etc/dovecot/conf.d.

Nakon svake promjene u konfiguraciji potrebno je ponovno pokrenuti servis Dovecot sljedećom naredbom: *# systemctl restart dovecot*

Po zadanome, servis Dovecot će posluživati protokole POP3 i IMAP. To se može izmijeniti u konfiguracijskoj datoteci /etc/dovecot/dovecot.conf tako da se obriše željeni protokol u sljedećoj liniji: *protocols = pop3 imap*

**Lokacija poštanskoga sandučića**

Postoje dvije opcije formata poštanskoga sandučića elektroničke pošte, maildir i mbox, a razlikuju se po putanji i načinu rada. Kod formata maildir elektronička pošta se nalazi u home direktoriju korisnika, tj. u $HOME/Maildir/, a svaka elektronička poruka se sprema u zasebnu datoteku. Također sadrži direktorije new, cur i tmp. Kod formata mbox se elektronička pošta nalazi u direktoriju /var/mail i jedna datoteka sadrži sve elektroničke poruke za jednog korisnika. Maildir je noviji način upravljanja elektroničkom poštom i ima mnogo unaprjeđenja u usporedbi s mboxom te je preporučeno njegovo korištenje. Maildir je skalabilan, otkriva duplikate, ima opciju pretraživanja, brži je pri radu s velikom količinom elektroničkih poruka itd.

Konfiguracijska datoteka /etc/dovecot/conf.d/10-mail.conf sadrži parametre za posluživanje i spremanje elektroničkih poruka, a za promjenu formata poštanskoga sandučića potrebno je dodati maildir u sljedećem retku: mail\_location = maildir:~/Maildir

**Autentikacija**

Kako bi se korisnici povezali s poslužiteljem, moraju potvrditi svoj identitet nekim od podržanih oblika autentikacije. Po zadanome, servis Dovecot korisnicima omogućuje spajanje na poslužitelj koristeći korisničko ime i lozinku. Autentikacija se izvodi koristeći čisti tekst (engl. plain text), a za kriptiranu komunikaciju mogu se koristiti protokoli IMAPS (IMAP Secure) i POP3S (POP3 Secure). Korištenje nešifriranih protokola IMAP i POP3 preporučuje se jedino u svrhu testiranja, a to se može postaviti u datoteci /etc/dovecot/conf.d/10-auth.conf sa sljedećim retkom: disable\_plaintext\_auth = no

**Zanimljivi izvori**

* <https://www.howtogeek.com/99423/>
* <https://en.wikipedia.org/wiki/Post_Office_Protocol>
* <https://en.wikipedia.org/wiki/Internet_Message_Access_Protocol>

4.5. Vježba 5. *Postfix*

U ovoj vježbi instalirat ćemo mail poslužitelje (MTA) na sustavima *server1* i *server2*. Sustav ***server1*** će biti *relayhost* za sustav *server2*.

1. Instalirajte paket *Postfix* na sustavu *server1*. Kod instalacije odaberite „Internet Site“.
2. Pošaljite poruku korisniku *root* (mail root@server1.tecaj.hr). Provjerite je li isporučena. Ako nema naredbe mail, instalirajte paket *mailutils*.

Proučite sadržaj direktorija **/var/spool/mail**.

1. Ako je potrebno, instalirajte paket telnet (apt-get install telnet).

Pokrenite telnet:

telnet localhost 25

i upišite sljedeće:

EHLO pogodi-tko-sam.hr

MAIL From: [opasan\_haker@tajna-lokacija.hr](mailto:opasan_haker@tajna-lokacija.hr) RCPT To: [root@server1.tecaj.hr](mailto:root@server1.tecaj.hr)

DATA

Subject: Opasna prijetnja Pozdrav od hakera.

. QUIT

Je li isporučena nova poruka za korisnika *root*? Proučite zapise u *log* datoteci ***/var/log/mail.log***.

1. U zonu tecaj.hr (datoteka ***/var/cache/bind/zones/tecaj.hr***) dodajte sljedeći MX-zapis:

tecaj.hr. IN MX 10 server1.tecaj.hr.

Pokrenite rndc reload.

Provjerite ispravnost nove konfiguracije: nslookup -query=mx tecaj.hr

1. U datoteku ***/etc/postfix/main.cf*** upišite tecaj.hr kod parametra mydestionation.

Ponovno pokrenite *Postfix*: service postfix restart. Pošaljite poruku na adresu [root@tecaj.hr.](mailto:root@tecaj.hr) Provjerite je li isporučena.

1. U datoteku ***/etc/aliases*** dodajte redak:

ime.prezime: root

Pokrenite newaliases.

Pošaljite poruke na [ime.prezime@tecaj.hr](mailto:ime.prezime@tecaj.hr) i [Ime.Prezime@tecaj.hr:](mailto:Ime.Prezime@tecaj.hr)

mail -s „Ovo je /etc/passwd“ [ime.prezime@tecaj.hr](mailto:ime.prezime@tecaj.hr)

< /etc/passwd

mail -s „Ovo je /etc/passwd“ [Ime.Prezime@tecaj.hr](mailto:Ime.Prezime@tecaj.hr)

< /etc/passwd

Provjerite jesu li poruke isporučene.

1. Popis poruka koje čekaju u redu za isporuku (engl. *mail queue*) može se ispisati naredbom **mailq**. Provjerite ima li na sustavu *server1* neisporučenih poruka koje čekaju u redu.

Pokušajte poslati sljedeću poruku:

mail -s „Oprostite na smetnji“ [user@srce.hr](mailto:user@srce.hr) <

/dev/null

1. Provjerite je li poruka ostala čekati u redu za isporuku:

mailq

Pronađite u log datoteci /var/log/mail.log zapise vezane za isporuku poruke koje ste poslali.

Pokušajte odgovoriti na sljedeća pitanja:

Kojem je poslužitelju u domeni srce.hr naš poslužitelj pokušao proslijediti poruku?

Koji je SMTP reply-kôd naš poslužitelj dobio kad je pokušao isporučiti poruku?

Zašto je poruka ostala u redu čekanja (nije isporučena primatelju niti se vratila pošiljatelju)?

**Mail relay**

S obzirom na to da *server2* za sada još uvijek ne može pristupati sadržajima na Internetu (za to je potrebno implementirati NAT na sustavu *server1*, što ćemo napraviti kasnije), postavit ćemo da sva pošta sa sustava *server2* ide preko sustava *server1*. Isto tako, postavit ćemo da sustav *server1* prihvaća i prosljeđuje svu poštu za sustav *server2*.

1. Instalirajte paket *Postfix* na sustavu *server2* te kod instalacije odaberite „Satellite system“. Za SMTP *relay host* upišite **server1.tecaj.hr**.

Upoznajte se s aktualnom konfiguracijom:

postconf mail\_version postconf inet\_interfaces

Proučite koje su standardne postavke programa *Postfix* izmijenjene zapisima u **main.cf**:

postconf -n

Također, postavit ćemo (putem datoteke ***.forward***) da se pošta za korisnika [*root@server2.tecaj.hr*](mailto:root@server2.tecaj.hr)prosljeđuje i korisniku *root* na sustavu *server1*.

1. Na sustavu *server2* pokušajte poslati poruku korisniku [*user@srce.hr*:](mailto:user@srce.hr)

mail [user@srce.hr](mailto:user@srce.hr) < /dev/null

Ako nedostaje naredba **mail**, instalirajte paket *mailutils*. Na oba sustava (server1 i server2) pokrenite naredbu **mailq** i provjerite na kojem je sustavu ova poruka završila u redu za isporuku (trebala bi biti na sustavu *server1*).

Što znači dobiveni rezultat?

1. Na sustavu *server2* u *home*-direktoriju korisnika *root* stvorite datoteku **.forward** sljedećega sadržaja:

\root (\root obavezno mora biti u prvom retku) [root@server1.tecaj.hr](mailto:root@server1.tecaj.hr)

Postavite odgovarajuća prava pristupa:

chmod 644 .forward

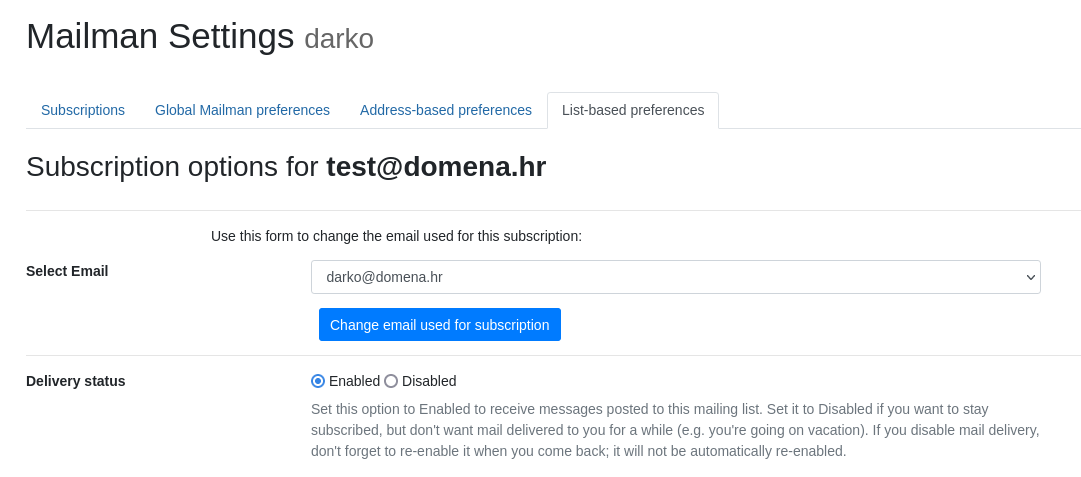
Pošaljite poruku korisniku [*root@server2.tecaj.hr*](mailto:root@server2.tecaj.hr)i provjerite je li ona isporučena na obje adrese navedene u datoteci ***.forward***.

4.6. Vježba 6. Upravljanje *mailing* listama

**Napomena**

1. Prilikom rješavanja ove vježbe služite se uputama iz teorijskog dijela priručnika.
2. Prođite sve napisane korake prema uputama iz teorijskog dijela priručnika.
3. Za dodavanje nove domene otvorite u web pregledniku http://localhost/mailman3/postorius/domains i kliknite na **Add domain**.  
   Dodajte domenu **domena.hr**. Za dodavanje nove liste otvorite u web pregledniku http://localhost/mailman3/postorius/lists i kliknite na **Add list**.   
   Dodajte domenu **Test**.
4. Dodajte korisnika **darko**:

adduser darko

1. Prijavite se kao **darko** na sučelje mailman3 na URL-u: http://localhost/mailman3/accounts/signup i spojite se kao **darko** kroz naredbenu liniju i kliknite na poveznicu u elektroničkoj poruci (u poruci će biti URL koji ima u sebi https, pa ga je potrebno prebaciti u http).   
   Spojite se kao darko na URL-u: http://localhost/mailman3/accounts/login i prijavite se na listu **Test**.
2. U web sučelju potrebno je omogućiti primanje klikom na '**Enabled**'.
3. Pošaljite testnu elektroničku poruku na novu mailing listu kao root korisnik:  
   echo 'Testna poruka' | mail -s 'Subject' test@domena.hr -r [root@domena.hr](mailto:root@domena.hr).
4. Spojite se kao korisnik **darko** i provjerite elektroničku poštu.

4.7. Vježba 7. *Dovecot*

1. Na sustavu *server1* instalirajte programske pakete za *Dovecot*:

dovecot-imapd i dovecot-pop3d

1. Provjerite je li ispravno podignut servis POP3, tako da se naredbom **telnet** spojite na port 110. Naredbe protokola POP3 označene su podebljanim slovima te ih upišite kako biste provjerili radi li POP3 ispravno.

# telnet localhost 110 Trying 127.0.0.1...

Connected to localhost. Escape character is '^]'.

+OK Dovecot ready.

**USER <korisnicko ime>**

+OK

**PASS <lozinka>**

+OK Logged in.

**LIST**

+OK 2 messages:

1 989

2 989

**. QUIT**

+OK Logging out.

Connection closed by foreign host.

Ako imate koju poruku, prikažite je POP3 naredbom RETR <indeks poruke>.

1. Provjerite je li ispravno podignut servis IMAP, tako da se naredbom **telnet** spojite na port 143.

**# telnet localhost 143 Trying 127.0.0.1...**

**Connected to localhost. Escape character is '^]'.**

OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN- REFERRALS ID ENABLE IDLE AUTH=PLAIN] Dovecot

ready.

a **login <korisnicko ime> <lozinka>**

a OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN- REFERRALS ID ENABLE IDLE SORT SORT=DISPLAY THREAD=REFERENCES THREAD=REFS THREAD=ORDEREDSUBJECT MULTIAPPEND URL-PARTIAL CATENATE UNSELECT CHILDREN NAMESPACE UIDPLUS LIST- EXTENDED I18NLEVEL=1 CONDSTORE QRESYNC ESEARCH ESORT SEARCHRES WITHIN CONTEXT=SEARCH LIST-STATUS BINARY MOVE SPECIAL-USE] Logged in

b **select inbox**

FLAGS (\Answered \Flagged \Deleted \Seen \Draft)

OK [PERMANENTFLAGS (\Answered \Flagged \Deleted

\Seen \Draft \\*)] Flags permitted.

2 EXISTS

0 RECENT

OK [UNSEEN 1] First unseen.

OK [UIDVALIDITY 1554747841] UIDs valid

OK [UIDNEXT 10681] Predicted next UID

b OK [READ-WRITE] Select completed (0.000 + 0.000 secs).

^]

telnet> quit Connection closed.

Protokol IMAP podržava daleko više opcija od protokola POP3, pa su time i same naredbe toga protokola daleko opsežnije.

Na računalu na kojem radite vježbe, pokrenite *Thunderbird* te pristupite poštanskom sandučiću na sustavu *server1*

1. Web-servisi

**Trajanje poglavlja:**

**180 min**



Po završetku ovoga poglavlja moći ćete:

* koristiti Apache2 za posluživanje web-stranica
* koristiti servis MariaDB kao sustav za upravljanje bazama podataka
* koristiti Nginx za posluživanje web-stranica
* usporediti servise za posluživanje web-stranica, Apache i Nginx.

Ova cjelina obrađuje dva najkorištenija poslužitelja web-stranica, Apache i Nginx. Također, fokus će biti i na servisu za upravljanje bazama podataka MariaDB.

**Napomena**

1. U ovom tečaju pojam „servis“ odnosi se na poslužitelje web-stranica te ne obrađuje web-usluge (engl. web-services).

5.1. HTTP – Apache2

**Apache2**

Web-poslužitelji, tj. servisi koji su instalirani na poslužiteljima, dohvaćaju i poslužuju web-stranice. U jednostavnom primjeru korisnik upisuje URL (Uniform Resource Locator), tj. web-adresu u svoj web-preglednik i web-poslužitelj prikazuje sadržaj korisniku. Također, web-adresa može pokrenuti neki program na poslužitelju i rezultat će biti prikazan korisniku.

Apache HTTP Server je besplatni softver otvorenoga kôda koji se najčešće koristi na operacijskim sustavima Linux. Najkorišteniji je softver za posluživanje web-stranica s udjelom od oko 67 %. Brz je, pouzdan i siguran, te se može prilagoditi raznim okolinama koristeći mnogobrojne dodatke.

**Instalacija**

Servis *Apache2* instalira se iz paketa sljedećom naredbom: # apt install apache2

**Direktoriji**

Servis *Apache2* ima četiri glavna direktorija:

|  |  |
| --- | --- |
| **Direktorij** | **Objašnjenje** |
| /etc/apache2/ | Sadrži konfiguracijske datoteke, a glavna datoteka je apache2.conf. |
| /var/www/ | Sadrži HTML datoteke, dokumente, slike, podatke koji se poslužuju korisniku. |
| /var/log/apache2/ | Sadrži access and error dnevničke zapise. |
| /usr/lib/cgi-bin/ | Sadrži CGI (Common Gateway Interface) skripte koje mogu biti pokrenute od servisa Apache2 za korisnike web-stranica. |

**Konfiguracijske datoteke**

Osim kod nekoliko glavnih konfiguracijskih datoteka, servis Apache2 koristi nazive direktorija koji završavaju na -available za sve dostupne konfiguracije i -enabled za sve trenutačno aktivne konfiguracijske datoteke. U direktoriju -enabled nalaze se simboličke poveznice na konfiguracije iz direktorija -available. Ovakav način omogućava administratorima onemogućavanje i omogućavanje raznih opcija bez potrebe za brisanjem konfiguracijskih datoteka.

Slijedi primjer naredbe:

# ln -s /etc/apache2/sites-available/domena\_hr.conf /etc/apache2/sites-enabled/domena\_hr.conf

Alternativno se mogu koristiti skripte za stvaranje i brisanje simboličkih poveznica:

|  |  |
| --- | --- |
| **Naredba** | **Objašnjenje** |
| a2enconf i a2disconf | Za konfiguracijske datoteke u direkotoriju conf-enabled. |
| A2enmod i a2dismod | Za module u direktoriju mods-enabled. |
| A2ensite i a2dissite | Za virtualne hostove u direktoriju sites-enabled. |

Za učitavanje konfiguracije potrebno je ponovno pokrenuti servis *Apache2*. Za uklanjanje konfiguracije potrebno je obrisati simboličku poveznicu i ponovno pokrenuti servis *Apache2*.

Slijedi tablica objašnjenja konfiguracija koje se nalaze unutar direktorija /etc/apache2:

|  |  |
| --- | --- |
| **Datoteke i direktoriji** | **Objašnjenje** |
| apache2.conf | Glavna konfiguracijska datoteka u kojoj se nalaze globalne konfiguracijske opcije. |
| conf-available | Sadrži sve ostale konfiguracijske datoteke. |
| conf-enabled | Simboličke poveznice na datoteke u direktoriju conf-available. |
| envvars | Varijable okruženja (engl. environment variables) servisa Apache2 |
| magic | Upute za određivanje tipa datoteka |
| mods-available | Sadrži sve datoteke za učitavanje i konfiguraciju modula. |
| mods-enabled | Simboličke poveznice na datoteke u direktoriju mods-available. |
| ports.conf | Konfiguracija portova na kojima će servis Apache2 osluškivati dolazne zahtjeve, po zadanom je na portu 80. |
| sites-available | Konfiguracije svih virtualnih hostova. |
| sites-enabled | Simboličke poveznice na virutalne hostove iz direktorija sites-available |

**Naredba apachectl**

Naredba apachectl koristi se za upravljanje (pokretanje, zaustavljanje i sl.) i promjenu konfiguracija servisa Apache2.

Slijedi sintaksa za upravljanje servisom: # apachectl naredba

Neke od naredbi za upravljanje su:

|  |  |
| --- | --- |
| **Naredba** | **Objašnjenje** |
| start | Pokreće servis Apache2 ili prikaže grešku ako je već pokrenut. |
| stop | Zaustavlja servis Apache2. |
| restart | Ponovno pokreće servis Apache2. Ako servis nije pokrenut, naredba će ga pokrenuti. Naredba provjerava ispravnost konfiguracijskih datoteka kako bi se servis ispravno pokrenuo. |
| fullstatus | Prikazuje potpuni izvještaj modula mod\_status, ako je modul aktiviran |
| status | Prikazuje skraćeni izvještaj |
| graceful | Ponovno učitava konfiguraciju servisa Apache2. Ako servis nije pokrenut, naredba će ga pokrenuti. Ovom naredbom se neće prekinuti već postojeće veze (engl. connections). Naredba također provjerava ispravnost konfiguracijskih datoteka kako bi se servis ispravno pokrenuo. |
| graceful-stop | Servis će se zaustaviti bez prekidanja otvorenih veza. |
| configtest | Provjerava ispravnost konfiguracijskih datoteka. |
| startssl | Pokreće servis Apache2 sa SSL podrškom. |

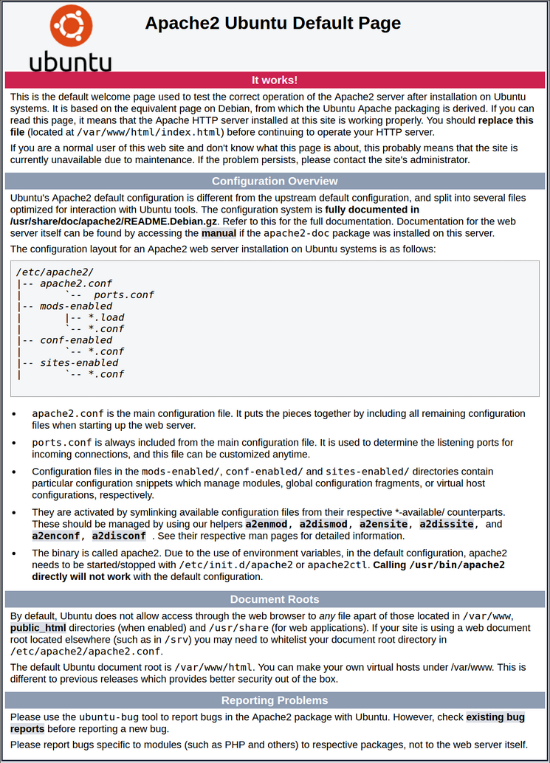
**Virtualni host**

Jedan web-poslužitelj može posluživati više web-domena, npr. www.srce.hr, www.unizg.hr, www.trecadomena.org. Ta mogućnost zove se virtualno web-sjedište, ili virtualni host. Servis Apache2 osluškuje zahtjeve korisnika i određuje za koju domenu je došao zahtjev te usmjeri promet prema ispravnom direktoriju. Na ovaj se način potpuno mogu odvojiti web-domene i njihovi direktoriji na istom poslužitelju.

Virtualni host može osluškivati zahtjeve na IP adresi poslužitelja ili može imati ime (FQDN). Virtualni host za IP adresu mora imati svoju IP adresu na poslužitelju (engl. IP based), a više virtualnih hostova s imenima (engl. name based) može dijeliti jednu IP adresu, zbog toga se i češće koristi. Za korištenje virtualnoga hosta s imenom (engl. name based) potrebno je postaviti DNS zapis koji povezuje ime virtualnoga hosta i IP adresu poslužitelja na kojem se nalazi.

Nakon instalacije servisa Apache2 omogućen je jedan virtualni host koji služi kao primjer, 000-default.

Za pristup zadanome virtualnom hostu potrebno je u web-preglednik upisati http://localhost i biti će prikazana sljedeća web-stranica:



Konfiguracija se nalazi u datoteci /etc/apache2/sites-available/000-default.conf i napravljena je simbolička poveznica u direktoriju /etc/apache2/sites-enabled:

/etc/apache2/sites-enabled/000-default.conf -> ../sites-available/000-default.conf

Slijedi primjer konfiguracije zadanoga virutalnog hosta:

<VirtualHost \*:80>  
 # The ServerName directive sets the request scheme, hostname and port that

# the server uses to identify itself. This is used when creating

# redirection URLs. In the context of virtual hosts, the ServerName

# specifies what hostname must appear in the request's Host: header to

# match this virtual host. For the default virtual host (this file) this

# value is not decisive as it is used as a last resort host regardless.

# However, you must set it for any further virtual host explicitly.

#ServerName www.example.com

ServerAdmin webmaster@localhost

DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,

# error, crit, alert, emerg.

# It is also possible to configure the loglevel for particular

# modules, e.g.

#LogLevel info ssl:warn

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

# For most configuration files from conf-available/, which are

# enabled or disabled at a global level, it is possible to

# include a line for only one particular virtual host. For example the

# following line enables the CGI configuration for this host only

# after it has been globally disabled with "a2disconf".

#Include conf-available/serve-cgi-bin.conf  
</VirtualHost>

Slijedi objašnjenje konfiguracije:

|  |  |
| --- | --- |
| **Mogućnost** | **Objašnjenje** |
| <VirtualHost \*:80> | Definira se IP adresa i port za virtualni host. Znak „\*“ označava sve IP adrese na poslužitelju. |
| ServerName | Definira se FQDN (Fully Qualified Domain Name) tj. web-domena virtualnoga hosta. Zadani virtualni host nema postavljenu direktivu ServerName i poslužuje sav promet koji dođe na web-poslužitelj, osim ako su definirane web-domene u drugim virtualnim hostovima. |
| ServerAlias | Definiraju se alternativna imena za virtualni host. Moguće je koristiti zamjenske znakove (engl. wildcards), npr. zvjezdica: \*.domena.hr |
| ServerAdmin | Definira elektroničku adresu administratora na koju se šalje elektronička poruka u slučaju problema s virtualnim hostom. |
| DocumentRoot | Definira direktorij koji sadrži sve potrebne datoteke koje će korisniku biti prikazane u njegovom pregledniku. |
| ErrorLog | Putanja do dnevničkog zapisa za greške. |
| CustomLog | Putanja do dnevničkog zapisa za bilježenje zahtjeva prema virtualnom hostu. |

Za dodavanje novoga virtualnog hosta potrebno je dodati konfiguraciju u direktorij /etc/apache2/sites-available/ i pokrenuti sljedeće naredbu:

# a2ensite ime-datoteke-vhosta

# apachectl gracefull

Prva naredba će napraviti simboličku poveznicu u direktorij /etc/apache2/sites-enable/, a druga će ponovno učitati konfiguraciju servisa Apache2 i time aktivirati virtualni host.

**Moduli**

Servis Apache2 dolazi s uključenim samo osnovnim funkcionalnostima, a brojne dodatke moguće je dodatno uključiti koristeći module, što servis čini izuzetno prilagodljivim. Postoje moduli koji služe za razne tipove autentikacije, podršku za programske jezike (Python, Perl, PHP, Ruby i sl.), podršku za protokole (SSL, FastCGI, http2 i sl.) itd. Neki od modula aktiviraju se nakon instalacije, a popis svih aktivnih modula moguće je provjeriti sljedećom naredbom:

# apachectl -M

Loaded Modules:

core\_module (static)

so\_module (static)

watchdog\_module (static)

http\_module (static)

log\_config\_module (static)

logio\_module (static)

version\_module (static)

unixd\_module (static)

access\_compat\_module (shared)

alias\_module (shared)

auth\_basic\_module (shared)

authn\_core\_module (shared)

authn\_file\_module (shared)

authz\_core\_module (shared)

authz\_host\_module (shared)

authz\_user\_module (shared)

autoindex\_module (shared)

deflate\_module (shared)

dir\_module (shared)

env\_module (shared)

filter\_module (shared)

mime\_module (shared)

mpm\_event\_module (shared)

negotiation\_module (shared)

reqtimeout\_module (shared)

setenvif\_module (shared)

status\_module (shared)

Popis svih modula može se pronaći na web-stranici: <https://httpd.apache.org/docs/2.4/mod/>

Slijedi opis nekoliko najkorištenijih modula:

|  |  |
| --- | --- |
| **Modul** | **Objašnjenje** |
| mod\_ssl | Omogućuje kriptiranje protokolima SSL (Secure Sockets Layer) i TLS (Transport Layer Security) kako bi se osigurala komunikacija. |
| mod\_status | Omogućava pregledavanje statistike posjeta virtualnim hostovima. |
| mod\_rewrite | Omogućava preusmjeravanje s jednog URL-a na drugi. Mogu se koristiti i regularni izrazi. |
| mod\_cache | Uključuje privremenu memoriju za web-stranice koje se često prikazuju kako bi web-preglednici brže učitavali sadržaj. |
| mod\_security | Radi kao vatrozid koristeći regularne izraze za osiguravanje web-sjedišta, tj. obranu od napada. |
| mod\_deflate | Sažima sadržaj prije slanja web-pregledniku kako bi se web-stranica brže učitavala. |
| mod\_proxy | Implementiraju se proxy ili reverse proxy mogućnosti koji preusmjeravaju promet prema poslužiteljima s traženim virtualnim hostom. Također, postoje mogućnosti balansiranja prometa. |

**Aktiviranje modula**

Aktiviranje modula vrši se naredbom a2enmod, nakon čega je potrebno ponovno pokrenuti servis Apache2:

# a2enmod ime\_modula

# systemctl restart apache2

Deaktiviranje modula vrši se naredbom a2enmod, nakon čega je potrebno ponovno pokrenuti servis Apache2:

# a2dismod ime\_modula

# systemctl restart apache2

Popis modula koji su instalirani uz servis Apache2 može se provjeriti u direktoriju /etc/apache2/mods-available, a ako neki modul nije instaliran, to je potrebno napraviti sljedećom naredbom:

# apt install libapache2-mod-ime\_modula

I nakon toga je potrebno aktivirati modul naredbom a2enmod i ponovno pokrenuti servis Apache2.

**Konfiguracija modula**

Konfiguracije modula nalaze se u direktoriju /etc/apache2/mods-available u datotekama koje imaju ekstenziju .conf. U datotekama se može nalaziti blok <IfModule> u kojem se nalaze konfiguracije modula, na primjer:

# cat /etc/apache2/mods-available/userdir.conf  
<IfModule mod\_userdir.c>  
 UserDir public\_html

UserDir disabled root

<Directory /home/\*/public\_html>

AllowOverride FileInfo AuthConfig Limit Indexes

Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec

Require method GET POST OPTIONS

</Directory>  
</IfModule>

**HTTPS virtualni host**

Modul mod\_ssl omogućava kriptiranje komunikacije između web-poslužitelja i web-preglednika, kako bi se zaštitile osjetljive informacije. Aktiviranje modula mod\_ssl i virtualnoga hosta na portu 443 koristi se protokol HTTPS (HyperText Transfer Protocol Secure). Kada korisnik upisuje web-adresu u web-preglednik, koristi se prefiks „https://”. Većina web-stranica preusmjerava promet s protokola HTTP na HTTPS, pa nije potrebno eksplicitno navoditi prefiks.

Modul se aktivira naredbom a2enmod nakon čega je potrebno ponovno pokrenuti servis Apache2:

# a2enmod ssl

Considering dependency setenvif for ssl:

Module setenvif already enabled

Considering dependency mime for ssl:

Module mime already enabled

Considering dependency socache\_shmcb for ssl:

Enabling module socache\_shmcb.

Enabling module ssl.

See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.

To activate the new configuration, you need to run:

systemctl restart apache2

# apachectl restart

Naredbom apachectl-M može se provjeriti je li dodani modul aktivan:

# apachectl -M | grep ssl

ssl\_module (shared)

**Virtualni host i certifikat**

Primjer konfiguracije virtualnoga hosta za korištenje protokola HTTPS stvoren je tijekom instalacije i nalazi se u datoteci: /etc/apache2/sites-available/default-ssl.conf:

<IfModule mod\_ssl.c>

<VirtualHost \_default\_:443>

ServerAdmin webmaster@localhost

DocumentRoot /var/www/html

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

SSLEngine on

SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem

SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

</VirtualHost>  
</IfModule>

Slijedi objašnjenje konfiguracija koje se razlikuju od HTTP virtualnoga hosta:

|  |  |
| --- | --- |
| **Mogućnosti** | **Objašnjenje** |
| <IfModule mod\_ssl.c> | Blok definira preduvjet za aktiviranje virtualnoga hosta, tj. ako je modul mod\_ssl aktiviran, može se aktivirati virtualni host u njegovu bloku. |
| <VirtualHost \_default\_:443> | Prihvaća promet koji dođe na port 443, osim ako nije namijenjen nekom drugom HTTPS virtualnom hostu. |
| SSLEngine | Aktiviranje protokola SSL/TLS. |
| SSLCertificateFile | Lokacija javnoga ključa certifikata. |
| SSLCertificateKeyFile | Lokacija privatnoga ključa certifikata. |

Kako bi korištenje protokola HTTPS bilo moguće, potrebni su certifikati pomoću kojih se komunikacija kriptira. Servis Apache2, po zadanome, stvori samopotpisani certifikat, tj. javni i privatni ključ koji se mogu koristiti za testiranje. Ti ključevi nalaze se u datotekama /etc/ssl/certs/ssl-cert-snakeoil.pem i /etc/ssl/private/ssl-cert-snakeoil.key. Za korištenje HTTPS virtualnoga hosta preporuča se korištenje certifikata izdanih od CA (Certificate Authority) organizacije, kao što su Comodo, DigiCert, IdenTrust, Symantec itd.

Za aktiviranje virtualnoga hosta koristi se naredba a2ensite nakon čega je potrebno ponovno pokrenuti konfiguraciju servisa Apache2:

# a2ensite default-ssl

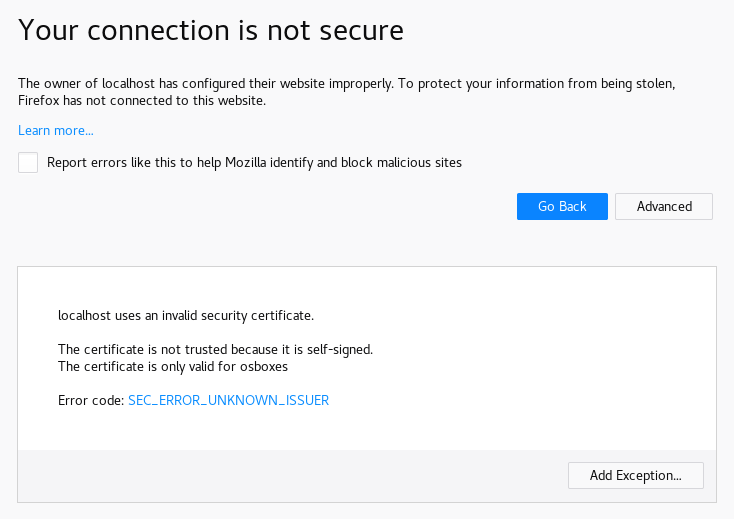
Enabling site default-ssl.

To activate the new configuration, you need to run:

systemctl reload apache2

# apachectl graceful

Upisivanje URL-a https://localhost u web-preglednik prikazati će se sljedeća web-stranica:



Web-preglednik, u ovom slučaju Firefox, prikazuje obavijest da je web-stranica nesigurna i da ne vjeruje certifikatu. Nakon što se testni certifikat zamijeni s certifikatom izdanim od CA organizacije obavijest će nestati. Virtualni host default-ssl, s testnim certifikatom može se prikazati nakon klika na gumb Add exception i potom Confirm Security Exception.

**Modul PHP**

PHP (Hypertext Preprocessor) je skriptni jezik namijenjen razvoju dinamičkoga web-sadržaja, a izvodi se na poslužiteljskoj strani na način da se ugrađuje u izvorni HTML kôd web-stranice. Programski jezik PHP je jedan od najzastupljenijih programskih jezika za izradu web-stranica te ima sličnu sintaksu kao programski jezici C, Java i skriptni programski jezik Perl.

Za stvaranje dinamičkih web-stranica koristi se grupa programa Linux, Apache, PHP i MySQL poznatija pod zajedničkim nazivom LAMP (Linux, Apache, MySQL, PHP). Servis Apache2 poslužuje dinamičke web-stranice koje generira programski jezik PHP koristeći podatke iz baze podataka MySQL. Instalacija paketa koji sadrži programski jezik PHP i aktivacija modula PHP vrši se sljedećom naredbom:

# apt install php

Reading package lists... Done

Building dependency tree

Reading state information... Done

The following additional packages will be installed:

libapache2-mod-php7.0 php-common php7.0 php7.0-cli php7.0-common php7.0-json php7.0-opcache php7.0-readline

Suggested packages:

php-pear

The following NEW packages will be installed:

libapache2-mod-php7.0 php php-common php7.0 php7.0-cli php7.0-common php7.0-json php7.0-opcache php7.0-readline

0 upgraded, 9 newly installed, 0 to remove and 6 not upgraded.

Need to get 3,570 kB of archives.

After this operation, 14.0 MB of additional disk space will be used.

Do you want to continue? [Y/n] y

...

apache2\_invoke: Enable module php7.0

Setting up php7.0 (7.0.33-0+deb9u3) ...

Setting up php (1:7.0+49) ..

Modul php7\_module aktivira se tijekom instalacije, što se može provjeriti sljedećom naredbom:

# apachectl -M | grep php

php7\_module (shared)

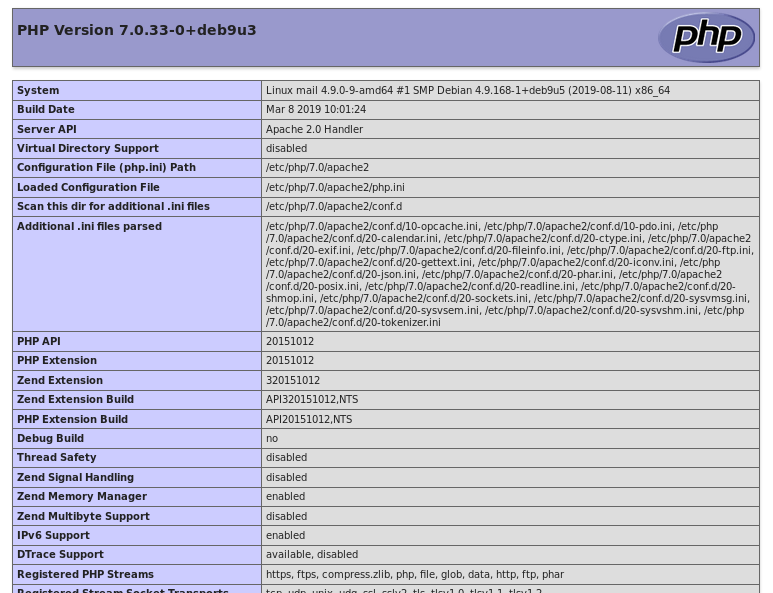
**Testiranje rada modula php7\_module**

Rad modula php7\_module može se testirati sljedećim kôdom PHP-a u datoteci /var/www/html/info.php:

# vim /var/www/html/info.php

<?php  
phpinfo();  
?>

Upisivanjem URL-a http://localhost/info.php u web-preglednik prikazat će se sljedeća web-stranica sa svim relevantnim informacijama i konfiguracijama o instaliranom modulu PHP:



**Konfiguracija**

Modul PHP ima glavu konfiguracijsku datoteku koja se koristi za servis Apache2: /etc/php/7.0/apache2/php.ini. Ostale konfiguracijske datoteke vezane za izvršavanje kôda PHP u naredbenoj liniji mogu se prikazati koristeći sljedeću naredbu:

# php --ini

Configuration File (php.ini) Path: /etc/php/7.0/cli

Loaded Configuration File: /etc/php/7.0/cli/php.ini

Scan for additional .ini files in: /etc/php/7.0/cli/conf.d

Additional .ini files parsed: /etc/php/7.0/cli/conf.d/10-opcache.ini,

/etc/php/7.0/cli/conf.d/10-pdo.ini,

/etc/php/7.0/cli/conf.d/20-calendar.ini,

/etc/php/7.0/cli/conf.d/20-ctype.ini,

/etc/php/7.0/cli/conf.d/20-exif.ini,

/etc/php/7.0/cli/conf.d/20-fileinfo.ini,

/etc/php/7.0/cli/conf.d/20-ftp.ini,

/etc/php/7.0/cli/conf.d/20-gettext.ini,

/etc/php/7.0/cli/conf.d/20-iconv.ini,

/etc/php/7.0/cli/conf.d/20-json.ini,

/etc/php/7.0/cli/conf.d/20-phar.ini,

/etc/php/7.0/cli/conf.d/20-posix.ini,

/etc/php/7.0/cli/conf.d/20-readline.ini,

/etc/php/7.0/cli/conf.d/20-shmop.ini,

/etc/php/7.0/cli/conf.d/20-sockets.ini,

/etc/php/7.0/cli/conf.d/20-sysvmsg.ini,

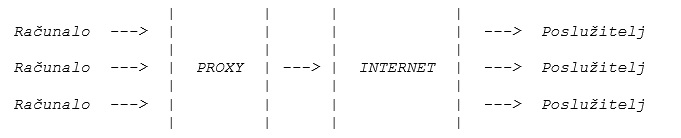
/etc/php/7.0/cli/conf.d/20-sysvsem.ini,

/etc/php/7.0/cli/conf.d/20-sysvshm.ini,

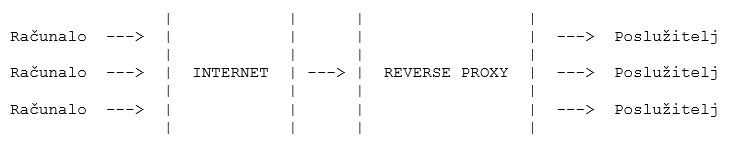
/etc/php/7.0/cli/conf.d/20-tokenizer.ini

**Moduli proxy i reverse proxy**

Proxy je modul koji posreduje između klijentskih zahtjeva (najčešće u lokalnoj mreži) i poslužitelja na Internetu. Klijentsko računalo povezuje se s modulom proxy tražeći neki resurs, kao što je web-stranica. Modul proxy procjenjuje zahtjev i, ako je odobren, zahtjev se prosljeđuje do poslužitelja. Prema tome, klijent nikada ne komunicira direktno s vanjskim poslužiteljem i svaki nedozvoljeni zahtjev se onemogućava.



Reverse proxy radi u obrnutom smjeru, zahtjeve koji dolaze od klijenata koji se nalaze na Internetu prosljeđuje do poslužitelja koji je smješten u lokalnoj mreži. Između klijenta i poslužitelja smješten je poslužitelj reverse proxy koji distribuira zahtjeve na jedan ili više pozadinskih (engl. backend) poslužitelja.



Na ovaj način klijent nikad ne komunicira direktno s pozadinskim poslužiteljima, a postoji i mogućnost balansiranja zahtjeva na više poslužitelja, što osigurava neprekidnost usluge i rasterećenje pojedinih poslužitelja. Poslužitelj reverse proxy ima i opciju privremene memorije ako se neki sadržaj često dohvaća pa s time dodatno smanjuje opterećenje poslužitelja. Također je moguće kontrolirati pristup poslužitelju, tj. uspostaviti vatrozid.

Modul proxy se može omogućiti na poslužitelju na kojem se nalaze aplikacije koje poslužuje, tako da zahtjeve koji dođu na port 80 ili 443 posreduje (proxy) na neki drugi port na kojem radi željena aplikacija.

Servis Apache2 koristi nekoliko modula za sve navedene mogućnosti:

* mod\_proxy: glavni modul za posredovanje (proxy) prometa raznim aplikacijama.
* mod\_proxy\_http: omogućuje posredovanje (proxy) HTTP prometa.
* mod\_proxy\_balancer i mod\_lbmethod\_byrequests: omogućuje balansiranje prometa na više poslužitelja.

Moduli se aktiviraju naredbom a2enmod, nakon čega je potrebno ponovno pokrenuti servis Apache2:

# a2enmod proxy proxy\_http proxy\_balancer lbmethod\_byrequests

# apachectl restart

# apachectl -M | grep 'proxy\|proxy\_http\|proxy\_balancer\|lbmethod\_byrequests'

lbmethod\_byrequests\_module (shared)

proxy\_module (shared)

proxy\_balancer\_module (shared)

proxy\_http\_module (shared)

Konfiguracija za reverse proxy definira se u virtualnom hostu na koji će dolaziti klijentski zahtjevi, u ovom slučaju to je zadani SSL virtualni host:

# vim /etc/apache2/sites-enabled/default-ssl.conf  
<VirtualHost \_default\_:443>

ProxyPreserveHost On

ProxyPass / http://127.0.0.1:8080/

ProxyPassReverse / http://127.0.0.1:8080/  
</VirtualHost>

# apachectl restart

**Objašnjenje konfiguracije:**

|  |  |
| --- | --- |
| **Mogućnosti** | **Objašnjenje** |
| ProxyPreserveHost | Servis Apache2 prosljeđuje originalno zaglavlje zahtjeva tako da pozadinski poslužitelji ili aplikacije znaju njegovu IP adresu. |
| ProxyPass | Definira gdje se prosljeđuje zahtjev. Na primjer, sav promet koji dođe na https://127.0.0.1 proslijedit će se na http://127.0.0.1:8080 i vratit će željenu web-stranicu klijentu. |
| ProxyPassReverse | Može biti isti kao i ProxyPass, a korigirat će zaglavlje koje dođe s pozadinskog poslužitelja. Tako će web-preglednik od klijenta dobiti IP adresu od proxy poslužitelja, a ne od pozadinskog poslužitelja. |

Da proxy dobro prosljeđuje promet može se provjeriti u dnevničkom zapisu error.log nakon upisivanja https://127.0.0.1 u web-preglednik:

[Wed Sep 26 17:42:44.779313 2018] [proxy:error] [pid 7552:tid 139906918418176] (111)  
Connection refused: AH00957: HTTPS: attempt to connect to 127.0.0.1:4433 (127.0.0.1) failed

Servis Apache2 pokuša doći do porta 4433, što je i konfigurirano, ali zbog nepostojanja aplikacije koja osluškuje na tom portu, ispisuje se greška.

Za korištenje balansera prometa, koji prosljeđuje promet prema nekoliko poslužitelja, koristi se sljedeća konfiguracija:

# vim /etc/apache2/sites-enabled/default-ssl.conf  
<VirtualHost \_default\_:443>

ProxyPreserveHost On

ProxyPass / balancer://backend/

ProxyPassReverse / balancer://backend/

<Proxy balancer://backend>

BalancerMember https://192.168.0.50:443

BalancerMember https://192.168.0.51:443

</Proxy>  
</VirtualHost>

# apachectl restart

Atributi ProxyPass i ProxyPassReverse sada sadrže direktivu (balancer://backend) koja se referencira na posebnom bloku konfiguracija (<Proxy></Proxy>). U ovom bloku se definira više lokacija prema kojima će se promet balansirati. Iz error.log zapisa može se provjeriti prosljeđuje li proxy dobro promet, ali zbog nepostojanja aplikacije koja sluša na tom portu, ispisuje se greška.

[Wed Sep 26 17:50:30.687771 2018] [proxy:error] [pid 7635:tid 140571102627584]   
(110)Connection timed out: AH00957: HTTP: attempt to connect to 192.168.0.51:443 (192.168.0.51)   
failed

**Nadzor opterećenja servisa Apache2**

Za nadzor opterećenja servisa Apache2 koristi se modul status\_module. Modul se aktivira tijekom instalacije, a za prikaz opterećenja potrebno je postaviti sljedeći blok unutar željenog virtualnog hosta i ponovno učitati konfiguraciju servisa Apache2:

# vim /etc/apache2/sites-enabled/000-default.conf

<Location /server-status>

SetHandler server-status

Require host localhost  
</Location>

# apachectl graceful

Na web-adresi http://localhost/server-status bit će prikazane statistike u tekstualnom obliku. Web-stranica prikazuje trenutačne statistike i te ju je potrebno osvježiti kako bi se dobili aktualni podatci ili je potrebno koristiti sljedeći URL za automatsko osvježavanje: *http://localhost/server-status?refresh=N*. Slijedi primjer web-stranice:

Apache Server Status for localhost (via ::1)

Server Version: Apache/2.4.25 (Debian) OpenSSL/1.0.2s

Server MPM: prefork

Server Built: 2019-04-02T19:05:13

Current Time: Tuesday, 27-Aug-2019 04:10:45 EDT

Restart Time: Monday, 26-Aug-2019 07:47:57 EDT

Parent Server Config. Generation: 4

Parent Server MPM Generation: 3

Server uptime: 20 hours 22 minutes 48 seconds

Server load: 0.00 0.00 0.00

Total accesses: 8 - Total Traffic: 42 kB

CPU Usage: u0 s.03 cu0 cs0 - 4.09e-5% CPU load

.000109 requests/sec - 0 B/second - 5.3 kB/request

1 requests currently being processed, 5 idle workers

\_\_\_W\_\_.......................................................... ................................................................ .....................

Scoreboard Key:

"\_" Waiting for Connection, "S" Starting up, "R" Reading Request,

"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,

"C" Closing connection, "L" Logging, "G" Gracefully finishing,

"I" Idle cleanup of worker, "." Open slot with no current process

Srv PID Acc M CPU SS Req Conn Child Slot Client Protocol VHost Request

0-3 12618 0/0/1 \_ 0.03 516 182 0.0 0.00 0.02 ::1 http/1.1 fe80::40ee:7811:3dc2:6a:443   
GET /info.php HTTP/1.1

1-3 12619 0/0/1 \_ 0.00 516 13 0.0 0.00 0.00 ::1 http/1.1 fe80::40ee:7811:3dc2:6a:443   
GET /server-status HTTP/1.1

2-3 12620 0/6/6 \_ 0.00 501 0 0.0 0.01 0.01 ::1 http/1.1 fe80::40ee:7811:3dc2:6a:443   
GET /server-status HTTP/1.1

3-3 12621 0/0/0 W 0.00 0 0 0.0 0.00 0.00 ::1 http/1.1 fe80::40ee:7811:3dc2:6a:443   
GET /server-status HTTP/1.1

Srv Child Server number - generation

PID OS process ID

Acc Number of accesses this connection / this child / this slot

M Mode of operation

CPU CPU usage, number of seconds

SS Seconds since beginning of most recent request

Req Milliseconds required to process most recent request

Conn Kilobytes transferred this connection

Child Megabytes transferred this child

Slot Total megabytes transferred this slot

SSL/TLS Session Cache Status:

cache type: SHMCB, shared memory: 512000 bytes, current entries: 0

subcaches: 32, indexes per subcache: 88

index usage: 0%, cache usage: 0%

total entries stored since starting: 0

total entries replaced since starting: 0

total entries expired since starting: 0

total (pre-expiry) entries scrolled out of the cache: 0

total retrieves since starting: 0 hit, 2 miss

total removes since starting: 0 hit, 0 miss

Apache/2.4.25 (Debian) Server at localhost Port 443

Opis statistika:

* Broj procesa (engl. worker) koji obrađuju zahtjeve.
* Broj procesa koji čekaju zahtjeve.
* Status svakoga procesa, broj zahtjeva koji je proces obradio i ukupni broj bajtova koji je proces poslužio.
* Ukupni broj bajtova.
* Vrijeme aktivnosti servisa Apache2 od zadnjeg pokretanja.
* Prosječan broj zahtjeva po sekundi, broj bajtova po sekundi i broj bajtova po zahtjevu.
* Postotak korištenja procesora za sve procese.
* Trenutačni broj poslužitelja i zahtjeva koji se obrađuju.

**Zanimljivi izvori**

* <https://httpd.apache.org/>
* <https://www.php.net/>
* <https://www.w3schools.com/php/>

5.2. Osnove MariaDB-a

**Instalacija i konfiguracija**

Servis MariaDB je sustav za upravljanje relacijskim bazama podataka. Po zadanome se instalira na operacijskom sustavu Debian i koristi se za organizaciju i osiguravanje pristupa informacijama koji se nalaze u bazama podataka, a prikazuju se na web-stranicama. Servis MariaDB je fork servisa MySQL koji se koristi na starijim verzijama operacijskoga sustava Debian. Sustavi za upravljanje bazama podataka MariaDB i MySQL su u mnogočemu kompatibilne, te je migracija bazi podataka, te njihovih tablica, podataka, korisnika i njihovih prava iz jedne baze u drugu jednostavna.

Naredbe za instalaciju i provjeru stanja servisa jesu:

# apt install mariadb-server

# systemctl status mariadb

● mariadb.service - MariaDB 10.1.38 database server

Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)

Active: active (running) since Mon 2019-08-26 04:40:46 EDT; 24h ago

Docs: man:mysqld(8)

https://mariadb.com/kb/en/library/systemd/

Main PID: 592 (mysqld)

Status: "Taking your SQL requests now..."

Tasks: 27 (limit: 4915)

CGroup: /system.slice/mariadb.service

└─592 /usr/sbin/mysqld

Glavna konfiguracijska datoteka je /etc/mysql/mariadb.cnf:

# The MariaDB configuration file

#

# The MariaDB/MySQL tools read configuration files in the following order:

# 1. "/etc/mysql/mariadb.cnf" (this file) to set global defaults,

# 2. "/etc/mysql/conf.d/\*.cnf" to set global options.

# 3. "/etc/mysql/mariadb.conf.d/\*.cnf" to set MariaDB-only options.

# 4. "~/.my.cnf" to set user-specific options.

#

# If the same option is defined multiple times, the last one will apply.

#

# One can use all long options that the program supports.

# Run program with --help to get a list of available options and with

# --print-defaults to see which it would actually understand and use.

#

# This group is read both both by the client and the server

# use it for options that affect everything

#

[client-server]

# Import all .cnf files from configuration directory

!includedir /etc/mysql/conf.d/

!includedir /etc/mysql/mariadb.conf.d/

Ona aktivira i konfiguracijske datoteke koje se nalaze u direktorijima /etc/mysql/conf.d/ i /etc/mysql/mariadb.conf.d/:

# ls /etc/mysql/conf.d/

mysql.cnf mysqldump.cnf

# ls /etc/mysql/mariadb.conf.d/

50-client.cnf 50-mysql-clients.cnf 50-mysqld\_safe.cnf 50-server.cnf

**Osiguravanje baze podataka**

Po zadanome, konfiguracije servisa MariaDB nisu sigurne. Sljedećom naredbom pokreće se interaktivna skripta koja služi za: osiguravanje konfiguracije koja postavlja lozinku za root korisnika, brisanje anonimnih korisnika, brisanje testnih baza itd.:

# mysql\_secure\_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB

SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current

password for the root user. If you've just installed MariaDB, and

you haven't set the root password yet, the password will be blank,

so you should just press enter here.

Enter current password for root (enter for none): test123

OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB

root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n] Y

New password: Akd6#5$ga14s!vf

Re-enter new password: Akd6#5$ga14s!vf

Password updated successfully!

Reloading privilege tables..

... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone

to log into MariaDB without having to have a user account created for

them. This is intended only for testing, and to make the installation

go a bit smoother. You should remove them before moving into a

production environment.

Remove anonymous users? [Y/n] Y

... Success!

Normally, root should only be allowed to connect from 'localhost'. This

ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y

... Success!

By default, MariaDB comes with a database named 'test' that anyone can

access. This is also intended only for testing, and should be removed

before moving into a production environment.

Remove test database and access to it? [Y/n] Y

- Dropping test database...

... Success!

- Removing privileges on test database...

... Success!

Reloading the privilege tables will ensure that all changes made so far

will take effect immediately.

Reload privilege tables now? [Y/n] Y

... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB

installation should now be secure.

Thanks for using MariaDB!

**Osnove upravljanja bazama podataka u MariaDB-u**

Naredba za ulaz u sustav za upravljanje bazama podataka jest:

# mariadb

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 15

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

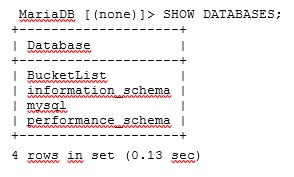
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>

Sve naredbe koje se upisuju u naredbenu liniju moraju završavati znakom točka-zarez ( ; ) kako bi se izvršile. MariaDB naredbe se pišu kombinacijom velikih i malih izraza radi lakšeg raspoznavanja, ali to nije potrebno jer naredbe ne razlikuju mala i velika slova.

**Uređivanje baze podataka**

Naredba za prikaz svih bazi podataka je:



Za dodavanje baze podataka koristi se naredba CREATE DATABASE, a za brisanje DROP DATABASE:

MariaDB [(none)]> CREATE DATABASE test;

Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> DROP DATABASE BucketList;

Query OK, 2 rows affected (1.11 sec)

MariaDB [(none)]> SHOW DATABASES;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| mysql |

| performance\_schema |

| test |

+--------------------+

4 rows in set (0.00 sec)

Za ulaz u pojedinu bazu podataka koristi se naredba USE, a popis tablica u bazi podataka prikazuje se naredbom SHOW tables;

MariaDB [information\_schema]> USE performance\_schema;

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

MariaDB [performance\_schema]> SHOW tables;

+----------------------------------------------------+

| Tables\_in\_performance\_schema |

+----------------------------------------------------+

| accounts |

| cond\_instances |

| events\_stages\_current |

| events\_stages\_history |

| events\_stages\_history\_long |

| events\_stages\_summary\_by\_account\_by\_event\_name |

| events\_stages\_summary\_by\_host\_by\_event\_name |

| events\_stages\_summary\_by\_thread\_by\_event\_name |

| events\_stages\_summary\_by\_user\_by\_event\_name |

| events\_stages\_summary\_global\_by\_event\_name |

| events\_statements\_current |

| events\_statements\_history |

| events\_statements\_history\_long |

| events\_statements\_summary\_by\_account\_by\_event\_name |

| events\_statements\_summary\_by\_digest |

| events\_statements\_summary\_by\_host\_by\_event\_name |

| events\_statements\_summary\_by\_thread\_by\_event\_name |

| events\_statements\_summary\_by\_user\_by\_event\_name |

| events\_statements\_summary\_global\_by\_event\_name |

| events\_waits\_current |

| events\_waits\_history |

| events\_waits\_history\_long |

| events\_waits\_summary\_by\_account\_by\_event\_name |

| events\_waits\_summary\_by\_host\_by\_event\_name |

| events\_waits\_summary\_by\_instance |

| events\_waits\_summary\_by\_thread\_by\_event\_name |

| events\_waits\_summary\_by\_user\_by\_event\_name |

| events\_waits\_summary\_global\_by\_event\_name |

| file\_instances |

| file\_summary\_by\_event\_name |

| file\_summary\_by\_instance |

| host\_cache |

| hosts |

| mutex\_instances |

| objects\_summary\_global\_by\_type |

| performance\_timers |

| rwlock\_instances |

| session\_account\_connect\_attrs |

| session\_connect\_attrs |

| setup\_actors |

| setup\_consumers |

| setup\_instruments |

| setup\_objects |

| setup\_timers |

| socket\_instances |

| socket\_summary\_by\_event\_name |

| socket\_summary\_by\_instance |

| table\_io\_waits\_summary\_by\_index\_usage |

| table\_io\_waits\_summary\_by\_table |

| table\_lock\_waits\_summary\_by\_table |

| threads |

| users |

+----------------------------------------------------+

52 rows in set (0.01 sec)

**Upravljanje korisnicima**

Korisnik, uz korisničko ime koje koristi za prijavu, mora imati definirano ime poslužitelja s kojeg se povezuje i lozinku. Za prikaz svih korisnika u bazi podataka i njihovih informacija koristi se sljedeća naredba:

MariaDB [(none)]> SELECT host, user, password FROM mysql.user;

+-----------+-------+-------------------------------------------+

| host | user | password |

+-----------+-------+-------------------------------------------+

| localhost | root | \*3D3B92F242033365AE5BC6A8E6FC3E1679F4140A |

| localhost | marko | \*995C7E05D796CF837BF774A0FDF7C6AAF604D684 |

+-----------+-------+-------------------------------------------+

2 rows in set (0.00 sec)

Kao što se vidi u primjeru, korisnici root i marko mogu se spojiti na bazu podataka samo s lokalnog poslužitelja, tj. s poslužitelja na kojem se nalazi baza podataka koristeći lozinku.

Ako je korisniku *marko* potrebno dodijeliti pristup na bazu podataka i s drugih poslužitelja, localhost vrijednost je potrebno zamijeniti s IP adresom ili imenom poslužitelja. Moguće je koristiti i zamjenski znak „%“ koji definira bilo koju vrijednost i na taj način je moguće postaviti pristup nizu različitih poslužitelja.

MariaDB [(none)]> UPDATE mysql.user SET Host='%' WHERE Host='localhost' AND User='marko';

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [(none)]> select host, user, password from mysql.user;

+-----------+-------+-------------------------------------------+

| host | user | password |

+-----------+-------+-------------------------------------------+

| localhost | root | \*3D3B92F242033365AE5BC6A8E6FC3E1679F4140A |

| % | marko | \*995C7E05D796CF837BF774A0FDF7C6AAF604D684 |

+-----------+-------+-------------------------------------------+

2 rows in set (0.00 sec)

Za stvaranje novoga korisnika koristi se naredba CREATE USER, a za brisanje DROP USER:

MariaDB [(none)]> CREATE USER 'mario'@'web.srce.hr' IDENTIFIED BY 'sa23#$@af1!';

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> DROP USER 'marko'@'%';

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> select host, user, password from mysql.user;

+-------------+-------+-------------------------------------------+

| host | user | password |

+-------------+-------+-------------------------------------------+

| localhost | root | \*3D3B92F242033365AE5BC6A8E6FC3E1679F4140A |

| web.srce.hr | mario | \*995C7E05D796CF837BF774A0FDF7C6AAF604D684 |

+-------------+-------+-------------------------------------------+

2 rows in set (0.00 sec)

Sljedeća naredba koristi se za promjenu lozinke, a u ispisu korisnika može se primijetiti promjena kriptografskoga sažetka lozinke:

MariaDB [(none)]> select host, user, password from mysql.user;

+-------------+-------+-------------------------------------------+

| host | user | password |

+-------------+-------+-------------------------------------------+

| localhost | root | \*3D3B92F242033365AE5BC6A8E6FC3E1679F4140A |

| web.srce.hr | mario | \*995C7E05D796CF837BF774A0FDF7C6AAF604D684 |

+-------------+-------+-------------------------------------------+

2 rows in set (0.00 sec)

MariaDB [(none)]> SET PASSWORD FOR 'mario'@'web.srce.hr' = PASSWORD('n0v4\_l0z!nk4');

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> select host, user, password from mysql.user;

+-------------+-------+-------------------------------------------+

| host | user | password |

+-------------+-------+-------------------------------------------+

| localhost | root | \*3D3B92F242033365AE5BC6A8E6FC3E1679F4140A |

| web.srce.hr | mario | \*9E1F803D727AA3FC0E2EBF0551F50640EF886684 |

+-------------+-------+-------------------------------------------+

2 rows in set (0.00 sec)

**Upravljanje pravima**

Svaki korisnik može imati različita prava za pojedine baze podataka i tablice unutar njih. Detaljan popis korisničkih prava može se pronaći na poveznici <https://mariadb.com/kb/en/library/grant/> .

Za prikaz svih prava za pojedinog korisnika koristi se sljedeća naredba:

MariaDB [(none)]> SHOW GRANTS FOR 'root'@'localhost';

+--------------------------------------------------------------------------------------------------------------------------------------------------+

| Grants for root@localhost

|

+--------------------------------------------------------------------------------------------------------------------------------------------------+

| GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'localhost' IDENTIFIED VIA unix\_socket USING '\*3D3B92F242033365AE5BC6A8E6FC3E1679F4140A' WITH GRANT OPTION |

| GRANT PROXY ON ''@'%' TO 'root'@'localhost' WITH GRANT OPTION |

+--------------------------------------------------------------------------------------------------------------------------------------------------+

2 rows in set (0.00 sec)

MariaDB [(none)]> SHOW GRANTS FOR 'mario'@'web.srce.hr';

+----------------------------------------------------------------------------------------------------------------+

| Grants for mario@web.srce.hr

|

+----------------------------------------------------------------------------------------------------------------+

| GRANT USAGE ON \*.\* TO 'mario'@'web.srce.hr' IDENTIFIED BY PASSWORD '\*9E1F803D727AA3FC0E2EBF0551F50640EF886684' |

+----------------------------------------------------------------------------------------------------------------+

1 row in set (0.00 sec)

Korisnik root ima sva prava (ALL PRIVILEGES) za sve baze podataka (\*.\*), a korisniku mario nisu dodana nikakva korisnička prava (USAGE).

**Napomena**

“\*” se koristi kao zamjenski znak za bilo koji drugi znak.

Za dodavanje svih prava na bazu podataka test korisniku mario koristi se sljedeća naredba:

MariaDB [(none)]> GRANT ALL PRIVILEGES ON test.\* TO 'mario'@'web.srce.hr';

Query OK, 0 rows affected (0.04 sec)

MariaDB [(none)]> SHOW GRANTS FOR 'mario'@'web.srce.hr';

+----------------------------------------------------------------------------------------------------------------+

| Grants for mario@web.srce.hr |

+----------------------------------------------------------------------------------------------------------------+

| GRANT USAGE ON \*.\* TO 'mario'@'web.srce.hr' IDENTIFIED BY PASSWORD '\*9E1F803D727AA3FC0E2EBF0551F50640EF886684' |

| GRANT ALL PRIVILEGES ON `test`.\* TO 'mario'@'web.srce.hr' |

+----------------------------------------------------------------------------------------------------------------+

2 rows in set (0.01 sec)

Za dodavanje svih korisničkih prava na sve baze podataka korisniku mario koristi se sljedeća naredba:

MariaDB [(none)]> GRANT ALL PRIVILEGES ON \*.\* TO 'mario'@'web.srce.hr';

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> SHOW GRANTS FOR 'mario'@'web.srce.hr';

+-------------------------------------------------------------------------------------------------------------------------+

| Grants for mario@web.srce.hr |

+-------------------------------------------------------------------------------------------------------------------------+

| GRANT ALL PRIVILEGES ON \*.\* TO 'mario'@'web.srce.hr' IDENTIFIED BY PASSWORD '\*9E1F803D727AA3FC0E2EBF0551F50640EF886684' |

| GRANT ALL PRIVILEGES ON `test`.\* TO 'mario'@'web.srce.hr' |

+-------------------------------------------------------------------------------------------------------------------------+

2 rows in set (0.00 sec)

Za oduzimanje svih korisničkih prava na sve baze korisniku mario koristi se sljedeća naredba:

MariaDB [(none)]> REVOKE ALL PRIVILEGES ON \*.\* FROM 'mario'@'web.srce.hr';

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> SHOW GRANTS FOR 'mario'@'web.srce.hr';

+----------------------------------------------------------------------------------------------------------------+

| Grants for mario@web.srce.hr |

+----------------------------------------------------------------------------------------------------------------+

| GRANT USAGE ON \*.\* TO 'mario'@'web.srce.hr' IDENTIFIED BY PASSWORD '\*9E1F803D727AA3FC0E2EBF0551F50640EF886684' |

| GRANT ALL PRIVILEGES ON `test`.\* TO 'mario'@'web.srce.hr' |

+----------------------------------------------------------------------------------------------------------------+

2 rows in set (0.00 sec)

Potrebno je sljedećom naredbom očistiti internu privremenu memoriju kako bi se učitala promijenjena korisnička prava.

MariaDB [(none)]> FLUSH PRIVILEGES;

Query OK, 0 rows affected (0.02 sec)

**Izvoz i uvoz baze podataka**

Izvoz i uvoz baze podataka koristi se u slučaju migracije ili sigurnosne pohrane. Izvoz i uvoz baze podataka može se izvršiti naredbom mysqldump, koja dolazi s instalacijom MySQL ili MariaDB baze podataka.

**Izvoz baze podataka**

Osnovna sintaksa naredbe za izvoz baze podataka je:

mysqldump -u <korisnicko\_ime> -p <ime\_baze\_podataka> > <ime\_pohrane.sql>

Slijedi primjer izvoza baze podataka:

# mysqldump -u root baza1 > /backup/ime\_baze.sql

# ls -al /backup/

total 12

drwxr-xr-x 2 root root 4096 Oct 18 14:20 .

drwxr-xr-x 26 root root 4096 Oct 18 08:47 ..

-rw-r--r-- 1 root root 1290 Oct 18 14:46 baza1.sql

Za izvoz više baza podataka koristi se opcija --databases, a za izvoz svih baza koristi se opcija --all-databases:

# mysqldump -u root --databases baza1 baza2 > /backup/baza1\_baza2.sql

# mysqldump -u root --all-databases > /backup/sve\_baze.sql

**Uvoz baze podataka**

Osnovna sintaksa naredbe za uvoz baze podataka:

mysqldump -u <korisnicko\_ime> -p <ime\_baze\_podataka> < <ime\_pohrane.sql>

Za uvoz baze podataka potrebno je spojiti se na bazu podataka naredbom mariadb i stvoriti novu bazu s naredbom CREATE DATABASE restore\_baze;:

# mariadb

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 6

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE restore\_baze;

Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> exit

Bye

Naredbom mysql -u uveze se pohranjena baza podataka: # mysql -u root restore\_baze < /backup/ime\_baze.sql

**Zanimljivi izvori**

* <https://blog.panoply.io/a-comparative-vmariadb-vs-mysql>
* <https://mariadb.com/kb/en/library/documentation/>

5.3. HTTP – Nginx

**Osnovna prilagodba**

Nginx je besplatni web-poslužitelj otvorenoga kôda koji se može koristiti kao reverse proxy s mogućnosti privremene memorije, balansiranja prometa, posredovanja (proxy) elektroničke pošte, HTTP spremnik privremene memorije itd. Koristi se kao alternativa servisu Apache2.

**Instalacija**

Servis se instalira sljedećom naredbom:

# apt install nginx

# systemctl status nginx

● nginx.service - A high performance web server and a reverse proxy server

Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)

Active: active (running) since Wed 2019-08-28 07:27:11 EDT; 1s ago

Docs: man:nginx(8)

Process: 18507 ExecStop=/sbin/start-stop-daemon --quiet --stop --retry QUIT/5 --pidfile /run/nginx.pid (code=exited, status=0/SUCCESS)

Process: 18512 ExecStart=/usr/sbin/nginx -g daemon on; master\_process on; (code=exited, status=0/SUCCESS)

Process: 18510 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master\_process on; (code=exited, status=0/SUCCESS)

Main PID: 18515 (nginx)

Tasks: 2 (limit: 4915)

CGroup: /system.slice/nginx.service

├─18515 nginx: master process /usr/sbin/nginx -g daemon on; master\_process on;

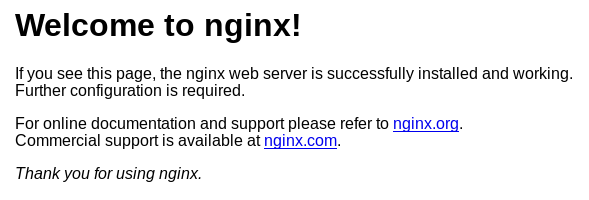
└─18516 nginx: worker process

Aug 28 07:27:11 mail systemd[1]: Starting A high performance web server and a reverse proxy server...

Aug 28 07:27:11 mail systemd[1]: nginx.service: Failed to read PID from file /run/nginx.pid: Invalid argument

Aug 28 07:27:11 mail systemd[1]: Started A high performance web server and a reverse proxy server.

Nakon upisivanja URL-a *http://localhost* u web-preglednik prikazat će se sljedeća web-stranica:



**Konfiguracije**

Slijede putanje i objašnjenja direktorija i datoteka koji se koriste u radu servisa Nginx:

|  |  |
| --- | --- |
| **Datoteke i direktoriji** | **Značenje** |
| /var/www/html | Direktorij sa sadržajem koji će se prikazivati u web-pregledniku. |
| /etc/nginx | Direktorij s konfiguracijskim datotekama. |
| /etc/nginx/nginx.conf | Glavna konfiguracijska datoteka s globalnim konfiguracijama. |
| /etc/nginx/sites-available/ | Konfiguracije svih virtualnih hostova. |
| /etc/nginx/sites-enabled/ | Simboličke poveznice na virutalne hostove iz direktorija sites-available. tj. aktivni virutalni hostovi. |
| /var/log/nginx/access.log | Dnevnički zapisi zahtjeva. |
| /var/log/nginx/access.log | Dnevnički zapisi pogrešaka. |

**Konfiguracija virtualnih hostova (server blocks)**

Nakon instalacije aktivan je jedan virtalni host, default:

# cat /etc/nginx/sites-enabled/default

server {

listen 80 default\_server;

listen [::]:80 default\_server;

root /var/www/html;

index index.html index.htm index.nginx-debian.html;

server\_name \_;

location / {

try\_files $uri $uri/ =404;

}

}

|  |  |
| --- | --- |
| **Mogućnosti** | **Objašnjenje** |
| listen 80 | Definira se port na kojem servis osluškuje zahtjeve. |
| default\_server | Definira zadani virtualni host koji će posluživati sav promet koji dođe na web-poslužitelj, osim za one web-domene koje su definirane u drugim virtualnom hostovima. |
| root /var/www/html | Definira direktorij koji sadrži sve potrebne datoteke za virtualni host. |
| index index.html index.htm index.nginx-debian.html | Definira poredak prioriteta datoteka koje se učitavaju za virtualnom hostu. |
| server\_name | Definira se FQDN, tj. web-domena virtualnoga hosta. |
| location / | Blok kojim se definiraju odgovori za zahtjeve za URL-ove i resurse koji su različiti od onih u virutalnom hostu. |

**Dodavanje novoga virtualnog hosta**

Nakon stvaranja novoga virtualnog bloka u direktoriju /etc/nginx/sites-available/ potrebno je napraviti simboličku poveznicu u direktorij /etc/nginx/sites-enabled/ i ponovno učitati konfiguraciju servisa Nginx:

# ln -s /etc/nginx/sites-available/domena.hr /etc/nginx/sites-enabled/

# systemctl reload nginx

**Zanimljivi izvori**

* <https://nginx.org/en/docs/>

5.4. Usporedba servisa Apache i Nginx

Apache2 i Nginx su dva najpopularnija poslužitelja otvorenoga kôda za posluživanje web-stranica. Osnovna razlika između servisa jest u dizajnu njihove arhitekture, tj. razlikuju se u načinu obrade zahtjeva i web-prometa te načinu reagiranja na različite mrežne zahtjeve.

Servis Nginx koristi znatno manje radne memorije od servisa Apache2 i može obrađivati oko 4 puta više zahtjeva po sekundi. Servis Apache2 je fleksibilniji, na primjer, može nadjačati generalne postavke pristupa za pojedine datoteke. Servis Nginx je teže konfigurirati i instalirati, kao i njegove module.

Oba servisa imaju velik broj korisnika što doprinosi aktivnom i sigurnom razvoju poslužitelja. Ali servis Nginx ima puno manje programskoga kôda, a time i manje mogućnosti za ranjivosti u budućnosti, što ga čini puno perspektivnijim u smislu sigurnosti.

Oba servisa su podjednaka u većini usporednih testova. Servis Nginx je puno brži za statični sadržaj, a za dinamički su podjednako dobri. Servis Apache2 ima bolju podršku što se tiče naprednijih mogućnosti, kao što je reprodukcija video sadržaja, reverse proxy i protokoli koji nisu bazirani na HTTP-u.

**Zanimljivi izvori**

* <https://www.digitalocean.com/community/tutorials/apache-vs-nginx-practical-considerations>
* <https://serverguy.com/comparison/apache-vs-nginx/>
* <https://www.plesk.com/blog/various/nginx-vs-apache-which-is-the-best-web-server/>
* <https://www.maketecheasier.com/nginx-vs-apache/>
  1. Vježba 8. HTTP – *Apache* i *MariaDB*

Ova vježba se radi na sustavu *server1*.

1. Instalirajte servis *Apache* (potrebno je instalirati paket Apache2).
2. Instalirajte podršku za PHP (potrebno je instalirati pakete php i php- mysql).
3. Stvorite virtualni host naziva **server1.tecaj.hr**. Slijedi primjer osnovne konfiguracije (datoteka ***/etc/apache2/sites- available/server1.tecaj.hr.conf***):

<VirtualHost \*:80>

ServerName server1.tecaj.hr ServerAdmin [root@server1.tecaj.hr](mailto:root@server1.tecaj.hr)

DocumentRoot "/var/www/server1.tecaj.hr"

<Directory "/var/www/server1.tecaj.hr"> Options +FollowSymLinks

AllowOverride All Require all granted

</Directory>

ErrorLog "/var/log/apache2/server1.tecaj.hr- error.log"

CustomLog "/var/log/apache2/server1.tecaj.hr- access.log" combined

</VirtualHost>

1. Stvorite direktorij **/var/www/server1.tecaj.hr**
2. Uključite virtualni host korištenjem naredbe **a2ensite**.
3. Stvorite datoteku ***/var/www/server1.tecaj.hr/phpinfo.php*** sljedećega sadržaja:

<?php phpinfo();

?>

1. Provjerite radi li PHP ispravno tako da se *web*-preglednik usmjeri na adresu: <http://server1.tecaj.hr/phpinfo.php>.
2. Instalirajte *MariaDB* (potrebno je instalirati pakete mariadb-server i mariadb-client).
3. Stvorite bazu podataka naziva *wordpress*.
4. Stvorite korisnika *wordpress* u sustavu *MariaDB*, te mu dajte sve ovlasti nad bazom podataka *wordpress*.
5. Skinite zadnju verziju *Wordpressa* sljedećom naredbom:

# cd /tmp; wget https://wordpress.org/latest.tar.gz

1. Otpakirajte preuzetu skinutu arhivu naredbom **tar**.

# tar xfvz latest.tar.gz

1. Kopirajte datoteke iz direktorija **/tmp/latest/** u **/var/www/server1.tecaj.hr/**.
2. U *web*-pregledniku otvorite adresu <http://server1.tecaj.hr/> i dovršite instalaciju *Wordpressa* kroz grafičko sučelje.
   1. Vježba 9. HTTP - Nginx:

Ova se vježba radi na sustavu *server2*.

1. Instalirajte servis *Nginx* (potrebno je instalirati paket nginx).
2. Stvorite virtualni host naziva **server2.tecaj.hr** i postavite da poslužuje podatke koji se nalaze u direktoriju **/var/www/server2.tecaj.hr**.
3. Instalirajte programski paket mutt.
4. Iskopirajte datoteke iz direktorija **/usr/share/doc/mutt/html/** u direktorij **/var/www/server2.tecaj.hr/**.

Usmjerite *web*-preglednik na adresu: <http://server2.tecaj.hr/>.

1. Dijeljenje datoteka

**Trajanje poglavlja:**

**180 min**



Po završetku ovoga poglavlja moći ćete:

* opisati protokol SMB/CIFS
* koristiti servis Samba za dijeljenje datotečnog sustava
* opisati protokol NFS
* koristiti servis NFS za dijeljenje datotečnog sustava
* koristiti alate rsync i inosync za kopiranje i sinkorizaciju podataka

Ova cjelina obrađuje načine dijeljenja datoteka na operacijskom sustavu, a zatim su opisani alati za lokalnu i udaljenu sinkronizaciju.

6.1. SMB/CIFS

**Protokol SMB/CIFS i servis Samba**

Protokol SMB (Server Message Block), u nekim verzijama poznat kao CIFS (Common Internet File System), je mrežni protokol za pristup datotekama, printerima, serijskim portovima između dvaju ili više računala na mreži, a služi i kao mehanizam za autenticiranje korisnika. Protokol SMB je razvio IBM sredinom osamdesetih, a protokol CIFS je specifična implementacija protokola SMB na operacijskim sustavima Windows. Zato što je protokol CIFS jako sličan protokolu SMB, klijenti koji koriste protokol SMB mogu komunicirati s poslužiteljem koji koristi protokol CIFS i obratno.

Danas se preporučuje koristiti protokol SMB, tj. njegove kasnije verzije 2 i 3, koji su standardi još od operacijskog sustava Microsoft Windows Vista 2006. godine pa do danas aktualnih verzija operacijskih sustava Windows. Unatoč tome što se najčešće koristi za Windows operacijske sustave, postoje implementacije za operacijske sustave Unix, Linux, macOS, OS/2, DOS.

Glavna funkcija protokola SMB jest dijeljenje datoteka, tj. dijeljenje datotečnoga sustava klijentima putem mreže, ali ima još mogućnosti:

* povezivanje i odspajanje s dijeljenih aplikacija
* pretraživanje direktorija
* ispis preko mreže
* pregled i promjena konfiguracija datoteka
* zaključavanje i otključavanje veličine podataka u datotekama

itd.

Servis Samba je besplatni program otvorenoga kôda koji se koristi za dijeljenje datoteka i usluga ispisa koristeći protokole SMB/CIFS za sve verzije operacijskih sustava Windows i Linux. Najčešće se koristi za integraciju s Active Directoryjem (baza podataka koja vodi evidenciju o svim resursima u sustavu – korisnicima, računalima, sigurnosnim grupama, servisima itd.) na operacijskim sustavima Linux i Unix. Prema tome, korisnici operacijskih sustava Windows mogu pristupati direktorijima, printerima itd. na operacijskom sustavu Linux. Dijeljeni direktoriji na operacijskom sustavu Linux, korisnicima operacijskoga sustava Windows djeluju kao normalni direktoriji dostupni preko mreže. Važan dio je autentikacija, pa servis Samba omogućava operacijskom sustavu Linux mogućnosti Domain Controllera (poslužitelj na kojem se nalazi Active Directory), tj. vjerodajnice operacijskoga sustava Windows mogu se koristiti na operacijskom sustavu Linux bez potrebe za dodatnim sustavima autentikacije.

**Prilagodba servisa Samba**

Servis Samba instalira se sljedećom naredbom: # apt install samba smbclient

Servis Samba u operacijskom sustavu Debian zove se smbd, a detaljnije informacije mogu se provjeriti sljedećom naredbom:

# systemctl status smbd

● smbd.service - Samba SMB Daemon

Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset: enabled)

Active: active (running) since Mon 2019-09-02 05:29:15 EDT; 12min ago

Docs: man:smbd(8)

man:samba(7)

man:smb.conf(5)

Main PID: 23049 (smbd)

Status: "smbd: ready to serve connections..."

Tasks: 4 (limit: 4915)

CGroup: /system.slice/smbd.service

├─23049 /usr/sbin/smbd

├─23050 /usr/sbin/smbd

├─23051 /usr/sbin/smbd

└─23054 /usr/sbin/smbd

**Napomena**

servis za svoj rad koristi dva pozadinska procesa:

1. smbd – dijeli datoteke i printere

nmbd – prevodi NetBIOS računalna imena u IP adrese.

IP adresa se zamjenjuje simboličkim imenom i obratno koristeći protokol NetBIOS. Naredbom servisa Smbclinet provjerava se primijenjena konfiguracija servisa Samba:

# smbclient -L localhost

Enter root's password:

Domain=[WORKGROUP] OS=[Windows 6.1] Server=[Samba 4.5.16-Debian]

Sharename Type Comment

--------- ---- -------

print$ Disk Printer Drivers

IPC$ IPC IPC Service (Samba 4.5.16-Debian)

Domain=[WORKGROUP] OS=[Windows 6.1] Server=[Samba 4.5.16-Debian]

Server Comment

--------- -------

MAIL Samba 4.5.16-Debian

Workgroup Master

--------- -------

WORKGROUP MAIL

**Konfiguracija servisa Samba**

Glavna konfiguracijska datoteka servisa *Samba* nalazi se u datoteci */etc/samba/smb.conf*:

[global]

workgroup = WORKGROUP

dns proxy = no

log file = /var/log/samba/log.%m

max log size = 1000

syslog = 0

panic action = /usr/share/samba/panic-action %d

server role = standalone server

passdb backend = tdbsam

obey pam restrictions = yes

unix password sync = yes

passwd program = /usr/bin/passwd %u

passwd chat = \*Enter\snew\s\*\spassword:\* %n\n \*Retype\snew\s\*\spassword:\* %n\n \*password\supdated\ssuccessfully\* .

pam password change = yes

map to guest = bad user

usershare allow guests = yes

[homes]

comment = Home Directories

browseable = no

read only = yes

create mask = 0700

directory mask = 0700

valid users = %S

[printers]

comment = All Printers

browseable = no

path = /var/spool/samba

printable = yes

guest ok = no

read only = yes

create mask = 0700

[print$]

comment = Printer Drivers

path = /var/lib/samba/printers

browseable = yes

read only = yes

guest ok = no

Konfiguracijska datoteka je podijeljena na četiri sekcije:

|  |  |
| --- | --- |
| **Sekcije** | **Objašnjenje** |
| [global] | Sadrži generalne konfiguracije servisa. |
| [homes] | Konfiguracije klijentskih home direktorija. |
| [printers] | Konfiguracije printera koje Windows klijenti mogu koristiti. |
| [print$] | Definira lokaciju drivera za printer za Windows klijente. |

Objašnjenja nekih konfiguracija global sekcije:

|  |  |
| --- | --- |
| **Mogućnosti** | **Objašnjenje** |
| workgroup = WORKGROUP | Ime workgrupe/NT-domene servisa Samba. |
| dns proxy = no | Sprječavanje pretrage NetBIOS imena s DNS-om. |
| log file = /var/log/samba/log.%m | Lokacija dnevničkoga zapisa. |
| syslog = 0 | Najniža razina dnevničkoga zapisa. |
| server role = standalone server | Servis će imati ulogu jedinoga poslužitelja u mreži. |

Objašnjenja nekih konfiguracija sekcije homes:

|  |  |
| --- | --- |
| **Mogućnosti** | **Objašnjenje** |
| comment = Home Directories | omentar koji se pojavljuje kada klijent zatraži direktorij. |
| browseable = no | Onemogućava pregled direktorija prilikom prikazivanja popisa dostupnih direktorija dijeljenih preko mreže. |
| read only = yes | Po zadanome dijeljeni direktoriji i datoteke neće se moći mijenjati, za suprotno potrebno je promijeniti konfiguraciju na no. |
| create mask = 0700 | Zbog sigurnosnih razloga stvorene datoteke imaju prava 0700. Preporučene postavke su 0775. |
| directory mask = 0700 | Zbog sigurnosnih razloga stvaranji direktoriji imaju prava 0700. Preporučene postavke su 0775. |
| valid users = %S | Definira popis korisnika koji se mogu povezati. U ovom slučaju %S definira da korisnik može vidjeti samo svoj home direktorij. |

Po zadanome, svaki novi korisnik će moći pristupiti svom home direktoriju, ali neće ga moći mijenjati, pa je potrebno primijeniti postavke kako su navedene u prethodnom paragrafu.

**Dodavanje dodatnih direktorija za dijeljenje**

Za dodavanje direktorija za dijeljenje, koji nije po zadanome home potrebno je dodati sljedeći blok u datoteku /etc/samba/smb.conf:

[srce\_projekt]

comment = Dijeljene datoteke za Srce projekt

read only = no

locking = no

path = /var/srce\_projekt

guest ok = no

Da bi se promjene primijenile, potrebno je ponovno pokrenuti servis Samba: # systemctl restart smbd

Sljedećom naredbom provjerava se dostupnost novoga dijeljenog direktorija:

# smbclient -L localhost

Enter root's password:

Domain=[WORKGROUP] OS=[Windows 6.1] Server=[Samba 4.5.16-Debian]

Sharename Type Comment

--------- ---- -------

homes Disk Home Directories

print$ Disk Printer Drivers

srce\_projekt Disk Dijeljene datoteke za Srce projekt

IPC$ IPC IPC Service (Samba 4.5.16-Debian)

Domain=[WORKGROUP] OS=[Windows 6.1] Server=[Samba 4.5.16-Debian]

Server Comment

--------- -------

MAIL Samba 4.5.16-Debian

Workgroup Master

--------- -------

WORKGROUP MAIL

I sljedećom naredbom se može pristupiti dijeljenom direktoriju srce\_projekt:

# smbclient '\\localhost\srce\_projekt'

Enter root's password:

Domain=[WORKGROUP] OS=[Windows 6.1] Server=[Samba 4.5.16-Debian]

smb: \>

smb: \>

smb: \> ls

. D 0 Mon Sep 2 09:15:49 2019

.. D 0 Mon Sep 2 09:15:21 2019

test N 5 Mon Sep 2 09:15:49 2019

226664964 blocks of size 1024. 209657528 blocks available

smb: \>

**Dodavanje korisnika**

Korisnik mora postojati u operacijskom sustavu Linux, tj. mora biti naveden u datoteci /etc/passwd, prije nego što se doda u servisu Samba. Cijela procedura dodavanja korisnika:

# adduser marko

Adding user `marko' ...

Adding new group `marko' (1003) ...

Adding new user `marko' (1003) with group `marko' ...

Creating home directory `/home/marko' ...

Copying files from `/etc/skel' ...

Enter new UNIX password:

Retype new UNIX password:

passwd: password updated successfully

Changing the user information for marko

Enter the new value, or press ENTER for the default

Full Name []:

Room Number []:

Work Phone []:

Home Phone []:

Other []:

Is the information correct? [Y/n] Y

# smbpasswd -a marko

New SMB password:

Retype new SMB password:

Added user marko.

Naredba za pregled svih korisnika je:

# pdbedit -w -L

marko:1003:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX:3B1B47E42E0463276E3DED6CEF349F93:[U ]:LCT-5D6D25F6:

Spajanje kao korisnik marko na svoj home direktorij radi se sljedećom naredbom:

# smbclient -U marko '\\localhost\marko'

Enter marko's password:

Domain=[WORKGROUP] OS=[Windows 6.1] Server=[Samba 4.5.16-Debian]

smb: \> ls

. D 0 Mon Sep 2 10:23:12 2019

.. D 0 Mon Sep 2 10:23:12 2019

.profile H 675 Mon Sep 2 10:23:12 2019

.bashrc H 3526 Mon Sep 2 10:23:12 2019

.bash\_logout H 220 Mon Sep 2 10:23:12 2019

277680196 blocks of size 1024. 263372568 blocks available

Korisnik marko se može spojiti i na direktorij srce\_projekt, jer direktorij ima postavke da ga svi mogu vidjeti i mijenjati:

# smbclient -U marko '\\localhost\srce\_projekt'

Enter marko's password:

Domain=[WORKGROUP] OS=[Windows 6.1] Server=[Samba 4.5.16-Debian]

smb: \> ls

. D 0 Mon Sep 2 09:15:49 2019

.. D 0 Mon Sep 2 09:15:21 2019

test N 5 Mon Sep 2 09:15:49 2019

226664964 blocks of size 1024. 209657524 blocks available

/pre>

**Zanimljivi izvori**

* <https://www.samba.org/>
* <http://wiki.open.hr/wiki/SAMBA>
* <https://www.varonis.com/blog/cifs-vs-smb/>

6.2. NFS

**Protokol NFS**

Protokol NFS je razvila tvrtka Sun Microsystems 1984. za mrežno dijeljenje datoteka i direktorija između operacijskih sustava Linux/Unix, ali se može koristiti i na operacijskim sustavima Windows i OS X. Klijent može koristiti mrežno dijeljene direktorije kao da su lokalni direktoriji.

Protokol NFS, uz nekoliko drugih protokola za mrežno dijeljenje, jest standard za NAS (Network Attached Storage), tj. centralizirani sustav za mrežno dijeljenje podataka. Takav sustav koristi poslužitelj koji upravlja autentikacijom, autorizacijom i upravljanjem klijentima, tj. drugim računalima koji se na njega povezuju i potražuju podatke.

Neke od prednosti korištenja mrežno dijeljenih direktorija su:

* datoteke i direktoriji se mogu dijeliti preko mreže na veći broj korisnika
* klijentska računala koriste manje diskovnoga prostora
* korištenje podataka neovisno je o lokaciji
* s protokolom NFS nije potrebno da sva računala koriste isti operacijski sustav
* home direktoriji korisnika mogu se dohvaćati preko mreže, pa nije potrebno da se na svakom računalu, na kojem korisnik ima račun, nalazi zasebni home direktoriji
* protokolom NFS može se konfigurirati centralizirani sustav za mrežno dijeljenje podataka
* nije potrebno ručno osvježavanje pri dohvaćanju novih podataka.

**Prilagodba servisa NFS**

Implementacija protokola NFS uključuje NFS poslužitelj, tj. računalo koje poslužuje druga računala (klijente). Na NFS poslužitelju instalira se paket nfs-kernel-server, a na klijentskim računalima paket nfs-common:

posluzitelj:~# apt install nfs-kernel-server

klijent:~# apt install nfs-common

Potrebno je stvoriti direktorij koji će se dijeliti na NFS poslužitelju s postavljenim pravima tako da svi mogu pregledavati i dodavati dokumente:

posluzitelj:~# mkdir /projekti

posluzitelj:~# chown nobody:nogroup /projekti

posluzitelj:~# ls -al /projekti

total 8

drwxr-xr-x 2 nobody nogroup 4096 Sep 3 07:19 .

**Datoteka /etc/exports**

Sljedeći korak je dodavanje direktorija koji će se dijeliti u konfiguracijsku datoteku /etc/exports. Datoteka sadrži popis direktorija koji će se dijeliti i način na koji će se dijeliti. Sintaksa konfiguracije je: direktorij klijent1(opcija1,opcija2) klijent2(opcija3,opcija4) Slijedi objašnjenje dijelova konfiguracije:

* direktorij: dijeljeni direktorij.
* klijentX: klijentska računala koja imaju pristup direktoriju. To može biti ime računala, njihove IP adrese, podmreža ili grupa računala.
* opcijaX: definira prava svakoga klijenta.

Ta prava mogu biti:

|  |  |
| --- | --- |
| **Mogućnosti** | **Objašnjenje** |
| ro | Direktorij se može samo čitati, tj. klijent ne može dodavati ili mijenjati dokumente na njemu. Ovo je zadano stanje. |
| rw | Klijent ima pravo čitanja i pisanja u direktoriju. |
| no\_root\_squash | Po zadanome, svaki zahtjev za datotekom od korisnika root na klijentskom računalu tretira se kao da je napravio korisnik nobody na poslužitelju. Ako no\_root\_squash nije aktiviran, korisnik root na klijentu imat će ista prava kao i korisnik root na poslužitelju, što je sigurnosni problem i ova opcija se mora oprezno koristiti. |
| no\_subtree\_check | Ako se samo jedan dio diska dijeli, ova rutina provjerava je li datoteka koju klijent zahtijeva u točnom dijelu diska. Ako se dijeli cijeli disk, onemogućavanje ove opcije ubrzat će kopiranje podataka. |
| sync | Podaci se mijenjaju trenutno, što može smanjiti performanse, ali je manja mogućnost da se izgube podaci. Ovo je zadana postavka. |
| async | Obratno od opcije sync, dozvoljava NFS poslužitelju da prekrši pravila protokola NFS i odgovori na zahtjev prije nego su promjene napravljene, što može dovesti do unaprijeđenja performansi, ali i do gubitka ili kvara na podacima. |

Slijedi primjer datoteke /etc/exports:

/projekti 192.168.1.1(rw,sync,no\_subtree\_check)

Da bi dijeljeni direktorij /projekti bio dostupan klijentu koji ima IP adresu 192.168.1.1, potrebno je ponovno pokrenuti servis nfs-kernel-server: posluzitelj:~# systemctl restart nfs-kernel-server

**Prilagodba klijenta**

Kako bi klijent pristupio dijeljenom direktoriju protokolom NFS, potrebno je instalirati paket nfs-common te stvoriti direktorij za dijeljenje:

klijent:~# apt install nfs-common

klijent:~# mkdir -p /projekti

Naredbom mount povezuje se s NFS poslužiteljem na IP adresi 192.168.1.2 i montira (engl. mount) se željeni direktorij:

klijent:~# mount 192.168.1.2:/projekti /projekti

Naredbom mount provjerava se ispravnost montiranoga direktorija i može se testirati stvaranje nove datoteke:

klijent:~# cd /projekti

klijent:/# mount | grep nfs

192.168.1.2:/projekti on /projekti type nfs4 (rw,relatime,vers=4.2,rsize=131072,wsize=131072,namlen=255,hard,proto=tcp,port=0,timeo=600,retrans=2,sec=sys,clientaddr=192.168.1.1,local\_lock=none,addr=192.168.1.2)

klijent:/projekti# touch test

klijent:/projekti# ls -al

total 8

drwxr-xr-x 2 nobody nogroup 4096 Sep 3 08:52 .

drwxr-xr-x 25 root root 4096 Sep 3 08:51 ..

-rw-r--r-- 1 nobody nogroup 0 Sep 3 08:52 test

Na NFS poslužitelju provjerava se dodana datoteka:

posluzitelj:~# ls -al /projekti/

total 8

drwxr-xr-x 2 nobody nogroup 4096 Sep 3 08:52 .

drwxr-xr-x 26 root root 4096 Sep 3 07:22 ..

-rw-r--r-- 1 nobody nogroup 0 Sep 3 08:52 test

Za odmontiranje dijeljenoga direktorija koristi se naredba umount: # umount /projekti/ Nakon ponovnog pokretanja klijenta potrebno je ponovno montirati dijeljeni direktorij naredbom mount ili se može postaviti automatsko montiranje dodavanjem sljedećega retka u konfiguracijsku datoteku */etc/fstab*:

192.168.1.2:/projekti /projekti nfs auto,nofail,noatime,nolock,intr,tcp,actimeo=1800 0 0

**Zanimljivi izvori**

* <https://www.dummies.com/computers/operating-systems/linux/how-to-share-files-with-nfs-on-linux-systems/>
* <https://vitux.com/install-nfs-server-and-client-on-ubuntu/>

6.3. Sinkronizacija datoteka

**Alat rsync**

Alat rsync (Remote Sync) efikasno kopira i sinkronizira podatke između računala, vanjskih tvrdih diskova ili preko mreže, uspoređujući njihovo vrijeme promjene i veličinu datoteka. Zbog svoje fleksibilnosti, brzine i mogućnosti skriptiranja, alat rsync je postao standardni alat operacijskoga sustava Linux. Koristi algoritam koji smanjuje količinu prometa preko mreže, kompresijom zlib se mogu dodatno smanjiti podaci, a korištenjem protokola ssh protok podataka je sigurniji. Alat rsync se koristi za sigurosne pohrane, sinkroniziranje sadržaja i kao napredna naredba za kopiranje datoteka.

Neke od mogućnosti alata rsync su:

* može ažurirati cijela stabla direktorija i datotečnih sustava
* kopiranje poveznica, uređaja, vlasništva i korisničkih prava
* opcije izostavljanja (engl. exclude) podataka
* može koristiti ljuske za udaljeno upravljanje, kao što su ssh i rsh
* ne zahtjeva sudo ovlasti.

**Sintaksa**

**Napomena**

Dijelovi sintakse koji se nalaze u zagradama < > su opcionalni.

Prilikom korištenja samo jednog argumenta za IZVOR, a bez argumenta ODREDIŠTE, naredba će prikazati popis datoteka umjesto kopiranja.

Sintaksa izvođenja naredbi razlikuje se ovisno o tome gdje se kopiraju podatci.

Za lokalnu upotrebu sintaksa je: rsync IZVOR <ODREDIŠTE>

Za kopiranje podataka korištenjem ljuske za udaljeno upravljanje:

rsync <KORISNIK@>IME\_RAČUNALA:IZVOR <ODREDIŠTE>

rsync IZVOR <KORISNIK@>IME\_RAČUNALA:ODREDIŠTE

Za kopiranje podataka korištenjem pozadinskoga procesa rsync:

rsync <KORISNIK@>IME\_RAČUNALA::IZVOR <ODREDIŠTE>

rsync rsync://<KORISNIK@>IME\_RAČUNALA<:port>/IZVOR <ODREDIŠTE>

rsync IZVOR <KORISNIK@>IME\_RAČUNALA::ODREDIŠTE

rsync IZVOR rsync://<KORISNIK@>IME\_RAČUNALA<:port>/ODREDIŠTE

**Opcije**

Neke od opcija alata rsync su:

|  |  |
| --- | --- |
| **Opcije** | **Objašnjenje** |
| -a, --archive | Uključuje sve potrebne opcije za arhiviranje, kao što je kopiranje datoteka rekurzivno, s očuvanjem simboličkih poveznica, prava, korisnika, grupa i vremena. Evkivalentno opcijama -rlptgoD. |
| -v, --verbose | lat rsync po zadanome ne prikazuje detaljne informacije nakon pokretanja naredbe. Opcija -v će prikazati informacije o datotekama koje se kopiraju i kratki sažetak o kopiranim podacima. Opcija -vv prikazat će informacije o statusu algoritma delta-transmission i koje su datoteke ažurirane kako bi se preskočile, te još više informacija o kopiranim podacima. Opcija -vvv se koristi za rješavanje problema koji se događaju tijekom pokretanja naredbe. |
| -h, --human-readable | Brojčane vrijednosti se prikazuju u formatu čitljivom za ljude. |
| -z, --compress | Koristi se opcija sažimanja radi smanjivanja mrežnoga prometa. |
| -d, --delete | Brišu se datoteke na odredištu ako se ne nalaze na izvorištu. |
| -g, --group | Zadržava grupno vlasništvo datoteka i direktorija. |
| -l, --links | Kopira simboličke poveznice. |
| -o, --owner | Zadržava vlasništvo datoteka i direktorija |
| -p, --perms | Zadržava prava nad datotekama i direktorijima. |
| -t, --times | Zadržava vremena promjena datoteka i direktorija.. |
| -r, --recursive | Rekurzivno kopira direktorije i sve što je u njima. |
| -u, --update | Ne kopira podatke do odredišta, ako su na odredištu noviji podaci. |
| -n, --dry-run | Vrši se simulacija sinkronizacije, bez kopiranja podataka. |
| --progress | Prikazuje napredak sinkronizacije. |
| -D, --devices --specials | BSprema uređaje i posebne datoteke.. |
| --remove-source-files | Briše izvorišne podatke nakon kopiranja |
| -e | Definira udaljenu ljusku |

**Primjeri upotrebe alata rsync**

Slijedi primjer lokalnoga kopiranja datoteke i cijeloga direktorija:

# rsync -avh /data/miner /backup/single/

sending incremental file list

created directory /backup/single

miner

sent 419.53M bytes received 72 bytes 119.87M bytes/sec

total size is 419.43M speedup is 1.00

# rsync -avh /data/ /backup/data1/

sending incremental file list

./

nautilus

nm-applet

pulseaudio.desktop

sent 367.09M bytes received 76 bytes 146.84M bytes/sec

total size is 367.00M speedup is 1.00

Ako na odredištu ne postoji direktorij, rsync će ga stvoriti.

**Napomena**

Putanje direktorija moraju završavati kosom crtom („/”).

Primjer kopiranja na udaljeno računalo pod korisnikom linux1:

# rsync -azvh /data/miner linux1@192.168.2.3:/home/linux1

linux1@192.168.2.3's password:

sending incremental file list

miner

sent 419.67M bytes received 35 bytes 10.62M bytes/sec

total size is 419.43M speedup is 1.00

Primjer kopiranja s udaljenog na lokalno računalo:

# rsync -azvh linux1@192.168.2.3:/home/linux1/extent /data/extent

linux1@192.168.2.3's password:

receiving incremental file list

extent

sent 43 bytes received 419.67M bytes 8.84M bytes/sec

total size is 419.43M speedup is 1.00

Za kriptirano slanje podataka može se koristiti protokol SSH dodavanjem opcije -e ssh:

# rsync -azvhe ssh /data/qlen linux1@192.168.2.3:/home/linux1

linux1@192.168.2.3's password:

sending incremental file list

qlen

sent 10.49M bytes received 35 bytes 2.33M bytes/sec

total size is 10.49M speedup is 1.00

**Alat inosync**

Alat inosync se koristi za sinkroniziranje podataka u realnom vremenu lokalno ili na udaljenu lokaciju. Alternativno se može koristi kombinacija crona i alata rsync, ali bez opcije sinkronizacije u realnom vremenu. Funkcija inotify je dio jezgre operacijskoga sustava Linux koja prati aktivnosti diska. Jedna od tih aktivnosti su zastavice (engl. flags), koje se koriste kako bi označavale promjenu podataka na disku. Alat inosync kombinira praćenje dodanih, promijenjenih ili obrisanih podataka na disku alatom rsync.

Alat inosync instalira se sljedećom naredbom: # apt install inosync

Primjer konfiguracijske datoteke nalazi se u datoteci: /usr/share/doc/inosync/examples/sample\_config.py:

cat /usr/share/doc/inosync/examples/sample\_config.py

# directory that should be watched for changes

wpath = "/var/www/"

# exclude list for rsync

rexcludes = [

"/localhost",

]

# common remote path

rpath = "/var/www/"

# remote locations in rsync syntax

rnodes = [

"a.mirror.com:" + rpath,

"b.mirror.com:" + rpath,

"c.mirror.com:" + rpath,

]

# extra, raw parameters to rsync

#extra = "--rsh=ssh -a"

# limit remote sync speed (in KB/s, 0 = no limit)

#rspeed = 0

# event mask (only sync on these events)

#emask = [

# "IN\_CLOSE\_WRITE",

# "IN\_CREATE",

# "IN\_DELETE",

# "IN\_MOVED\_FROM",

# "IN\_MOVED\_TO",

# "IN\_MODIFY",

#]

# event delay in seconds (prevents huge amounts of syncs, but dicreases the

# realtime side of things)

#edelay = 10

# rsync log file for updates

#logfile = /var/log/inosync.log

# rsync binary path

#rsync = "/usr/bin/rsync"

Konfiguracijske linije koje je potrebno podesiti su:

* wpath: izvorišni direktorij koji se sinkronizira.
* rexcludes: lista datoteka i direktorija koji se isključuju od sinkronizacija.
* rpath: odredišni direktorij.
* rnodes: udaljena računala gdje se nalaze odredišni direktoriji.

Za pokretanje alata koristi se naredba inosync s definiranom opcijom -c, tj. s konfiguracijom datotekom: inosync -c /etc/inosync/conf.py

Moguće je napraviti konfiguracijsku datoteku servisa inosync za systemd te omogućiti da se automatski pokreće nakon ponovnog pokretanja operacijskoga sustava. Slijedi primjer konfiguracijske datoteke:

cat /etc/systemd/system/inosync.service

[Unit]

Description=Inosync

After=network.target

[Service]

Type=simple

ExecStart=/usr/bin/inosync -c /etc/inosync/conf.py

User=inosync

Group=inosync

[Install]

WantedBy=multi-user.target

**Zanimljivi izvori**

* <https://en.wikipedia.org/wiki/Rsync>
* <https://www.digitalocean.com/community/tutorials/how-to-use-rsync-to-sync-local-and-remote-directories-on-a-vps>
* <https://github.com/hollow/inosync>

6.4. Vježba 10. SMB/CIFS

* + - 1. Na sustavu *server1* instalirajte servis SMB (potrebno je instalirati paket samba).
      2. Naredbama **ps** i **systemctl** provjerite jesu li aktivni servisi *nmbd* i *smbd*.
      3. U konfiguracijskoj datoteci **Sambe** podesite direktorij za dijeljenje (blok [srce\_projekt] u teorijskom dijelu priručnika). Nakon promjena konfiguracije, ne zaboravite ponovno pokrenuti servis.
      4. Ako ne postoji, dodajte korisnika ***tux*** na sustav *server1* i u *Sambu*.
      5. Instalirajte paket smbclient.
      6. Provjerite je li se moguće spojiti na **\\localhost\srce\_projekt** te [**\\localhost\\tux**](file:///D:\tux).

6.5. Vježba 11. NFS

1. Na sustavu *server1* instalirajte potrebnu programsku podršku za NFS poslužitelj (paket nfs- kernel-server).
2. Napravite direktorij /projekti koji će biti dijeljen protokolom NFS. Prema uputama u teorijskom dijelu priručnika, postavite potrebna vlasništva i dozvole nad tim direktorijem.
3. Na sustavu *server1* u datoteku /**etc/exports** dodajte sljedeći redak:

/projekti \*(rw,sync,no\_subtree\_check)

Nakon izmjena u datoteci **/etc/exports** ponovno pokrenite NFS poslužitelj naredbom:

service nfs-kernel-server restart

1. Na sustavu *server2* instalirajte klijentsku podršku (paket nfs-common).
2. Na sustavu *server2* napravite direktorij **/projekti** i u njega montirajte udaljeni direktorij.

mount -t nfs 192.168.100.201:/projekti /projekti

1. Koristeći naredbu **df** provjerite je li ispravno montirano.
2. Kreirajte proizvoljne datoteke i direktorije u direktoriju **/projekti** i provjerite jesu li datoteke vidljive na obama sustavima *server1* i *server2*.

***Bilješke:***