

Apache HTTP poslužitelj

CERT.hr-PUBDOC-2019-12-396

Sadržaj

1	UVOD	3
2	INSTALACIJA APACHE HTTP POSLUŽITELJA	4
3	KORIŠTENJE APACHE HTTP POSLUŽITELJA.....	7
3.1	KONFIGURACIJSKE DATOTEKE.....	7
3.2	OSNOVNE NAREDBE.....	9
3.3	PRIMJER KORIŠTENJA APACHE HTTP POSLUŽITELJA	9
3.4	VIRTUALNI DOMAĆINI.....	12
3.5	SIGURNA KONFIGURACIJA APACHE HTTP POSLUŽITELJA.....	14
4	ZAKLJUČAK	20

Ovaj dokument izradio je Laboratorij za sustave i signale Zavoda za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Ovaj dokument vlasništvo je Nacionalnog CERT-a. Namijenjen je javnoj objavi te se svatko smije njime koristiti i na njega se pozivati, ali isključivo u izvornom obliku, bez izmjena, uz obvezno navođenje izvora podataka. Korištenje ovog dokumenta protivno gornjim navodima povreda je autorskih prava CARNET-a, a sve navedeno u skladu je sa zakonskim odredbama Republike Hrvatske.

1 Uvod

Web poslužitelji, tj. HTTP poslužiteljski softver (engl. *web server/HTTP server*), omogućuje postavljanje sadržaja na web kako bi mu mogli pristupiti korisnici diljem svijeta. Jedan od najčešće korištenih je **Apache HTTP poslužitelj** koji poslužuje preko 40% aktivnih web stranica.

Objavljen je 1995. godine i danas je jedan od najkorištenijih web poslužitelja u svijetu. Za njega se često može čuti i izraz „Apache httpd“, ili samo „Apache“ (iako je „Apache“ zapravo naziv organizacije koja održava i brojne druge programske pakete).

Osim pouzdanosti, brzine i sigurnosti, njegovoj popularnosti doprinijelo je i to što je besplatan, dobro dokumentiran i otvorenog kôda. Moguće ga je koristiti na raznim operacijskim sustavima poput Windowsa i macOS-a, ali ipak ga je uobičajeno koristiti na nekoj Linux distribuciji, gdje je u kombinaciji zvanom LAMP vrlo popularno i snažno slobodno (engl. *free and open source*) okruženje otvorenog kôda za razvoj web aplikacija. Ostale komponente LAMP-a čine operacijski sustav **Linux**, sustav za upravljanje bazom podataka **MySQL** te **PHP**, dinamički jezik kojim je moguće dohvaćati podatke iz baze podataka (umjesto njega mogu se koristiti i jezici Perl ili Python). Svaka od tih komponenti je slobodna, otvorenog kôda i besplatna za preuzimanje, a međusobno su kompatibilne i čine cjelokupni sustav za razvoj web aplikacija. **WordPress**, jedan od najpopularnijih alata za izradu web stranica, upravo se oslanja na LAMP okruženje.

Karakterističnosti Apache HTTP poslužitelja su korištenje modularne arhitekture koja će se dodatno pojasniti i prokomentirati u poglavlju u konfiguraciji. Modularna arhitektura u ovom slučaju označava da se instalacijom Apachea instaliraju osnovne funkcionalnosti potrebne za rad poslužitelja, a za dodatne mogućnosti poput rada s bazama podataka ili određenim poslužiteljskim programskim jezikom (npr. PHP) potrebno je instalirati i uključiti dodatne module.

Također, Apache podržava i posluživanje više web stranica (s različitim domenskim imenima) na istom poslužitelju, s jednom IP adresom (tzv. *virtual hosting*).

2 Instalacija Apache HTTP poslužitelja

Apache HTTP poslužitelj uobičajeno se koristi na Linux distribucijama, pri čemu se dugo vremena kao jedna od popularnijih arhitektura za posluživanje web aplikacija ističe LAMP (Linux – Apache – MySQL/MariaDB - PHP). No, iako je uobičajena instalacija i korištenje Apachea na operacijskom sustavu Linux, moguće ga je koristiti i na operacijskim sustavima Windows, macOS te na brojnim drugim operacijskim sustavima iz obitelji Unix-a (npr. FreeBSD).

U ovom dokumentu opisana je instalacija, uređivanje osnovnih postavki i korištenje **Apachea** na distribuciji Linuxa **Debian**, ali procedura je slična i za ostale distribucije, uz napomenu da je Debian specifičan po organizaciji konfiguracijskih datoteka – konfiguracija se pohranjuje u više manjih datoteka, te postoje i određene dodatne skripte za upravljanje (npr. skripte za uključivanje i isključivanje pojedinih modula, web stranica i sl.) čime se olakšava administracija i održavanje web poslužitelja.

Instalacija Apachea provest će se iz službenog repozitorija distribucije Debian. Kad god se nešto instalira na Linux distribucije, obično bi se trebala dati prednost instalaciji iz službenog repozitorija jer se na taj način osigurava kompatibilnost s ostalim softverom i jednostavnije ažuriranje instaliranog softvera. Ažuriranje je bitno za sustav kako bi se pravovremeno „zakrpale“ moguće ranjivosti i izbjegli napadi na sustav. Naravno da korisnik može i ručno provjeravati postoji li dostupna novija inačica ili zakrpa, ali kako bi se izbjegla ljudska pogreška i potreba da se neprestano provjeravaju inačice softvera, preporučljivo je to ipak prepustiti softveru za upravljanje paketima.

Za instalaciju Apachea iz Debianovog repozitorija potrebno je u naredbenu ljsku upisati sljedeću naredbu i nakon nje pritisnuti tipku 'Enter':

```
$ sudo apt install apache2
```

Originalna verzija poslužitelja (Apache 1) više nije podržana i ne nalazi se u službenim repozitorijima, već se koristi nadograđena i proširena inačica Apache 2 i to je razlog zbog kojeg se koristi naredba za instalaciju `sudo apt install apache2`, a ne `sudo apt install apache`.

Kako bi započela instalacija, korisnik mora unijeti lozinku, što može izgledati zbunjujuće za nove korisnike Linuxa jer konzola ne ispisuje znakove lozinke dok se unose, ali ispravna zaporka vodi na daljnje korake instalacije. Pojavljuje se pitanje „*Do you want to continue?*“ (hrv. „Želite li nastaviti?“), kao što je prikazano na slici 1, na što se unosi 'Y' kao potvrđan odgovor te započinje instalacija.

```

debian login: user
Password:
Linux debian 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u1 (2019-09-20) 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.
user@debian:~$ sudo apt install apache2

Vjerujemo da vam je administrator lokalnog sustava održao uobičajeno
predavanje. To se obično svodi na sljedeće tri stvari:

#1) Poštujte tuđu privatnost.
#2) Mislite prije tipkanja.
#3) S velikim moćima dolazi velika odgovornost.

[sudo] lozinka za user:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libbrotli1 libcurl4 libjansson4 liblua5.2-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libbrotli1 libcurl4 libjansson4 liblua5.2-0 ssl-cert
0 upgraded, 13 newly installed, 0 to remove and 0 not upgraded.
Need to get 2.959 kB of archives.
After this operation, 9.663 kB of additional disk space will be used.
Do you want to continue? [Y/n] _

```

Slika 1. Prikaz konzole nakon unosa naredbe za instalaciju

Konfiguracija Apachea instalirat će se u direktorij `/etc/apache2`.

Ako je Apache uspješno pokrenut, nakon unosa naredbe:

```
$ systemctl status apache2
```

očekuje se status *active*, kao na slici 2.

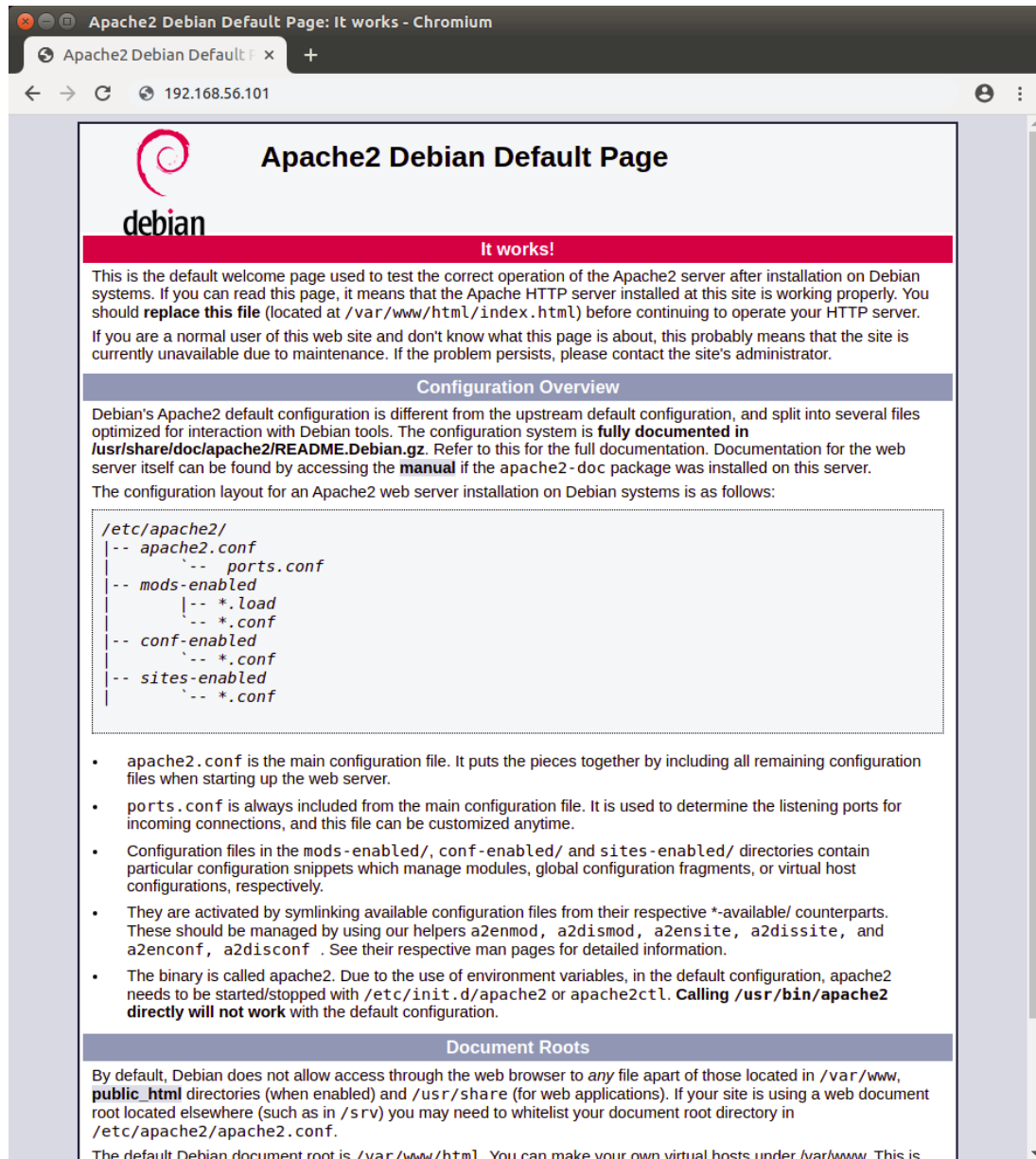
```

user@debian:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2019-11-27 09:50:51 CET; 40min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 424 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 465 (apache2)
    Tasks: 55 (limit: 1150)
   Memory: 14.9M
   CGroup: /system.slice/apache2.service
           └─465 /usr/sbin/apache2 -k start
             └─467 /usr/sbin/apache2 -k start
               └─468 /usr/sbin/apache2 -k start
user@debian:~$ _

```

Slika 2 Apache poslužitelj je aktivan

Sada je moguće preko IP adrese računala pristupiti web stranici, npr. na URL-u „`http://192.168.56.101/`“.



Slika 3 Prikaz početne stranice Apache HTTP poslužitelja nakon uspješne instalacije

Sada je Apache uspješno instaliran i korisniku su dostupne njegove **osnovne značajke**, ali za ostvarivanje **dodatnih funkcionalnosti** web stranica potrebno je instalirati odgovarajuće **module**, kao što su modul za PHP koji omogućuje izvođenje PHP kôda na stranici ili PHP upravljački program za baze podataka.

3 Korištenje Apache HTTP poslužitelja

Konfiguracija Apache HTTP poslužitelja nalazi se u mapi `/etc/apache2`. Definiranje i konfiguriranje datoteka u tom direktoriju objašnjeno je u nastavku. Upute su napisane za korisnike koji upravljaju računalom preko ljuške operacijskog sustava (engl. *shell*), tj. preko sučelja naredbene linije. To je uobičajeni način upravljanja poslužiteljskim računalima koja koriste operacijski sustav iz obitelji Unixa, primjerice Linux računala na kojima su postavljene web stranice, poslužiteljski servisi za e-poštu i slično.

3.1 Konfiguracijske datoteke

Prvo je potrebno pozicionirati se u direktorij u kojemu se nalaze konfiguracijske datoteke Apache web poslužitelja:

```
$ cd /etc/apache2
```

Naredbom `ls` ispisat će se postojeće datoteke unutar tog direktorija.

```
user@debian:/etc/network$ cd /etc/apache2
user@debian:/etc/apache2$ ls
apache2.conf  conf-enabled  magic          mods-enabled  sites-available
conf-available  envvars      mods-available  ports.conf    sites-enabled
```

Slika 4 Datoteke u mapi `/etc/apache2`

- **apache2.conf**: glavna konfiguracijska datoteka Apachea – iz nje je moguće mijenjati postavke gotovo cijele konfiguracije. Zbog preglednije organizacije konfiguracijskih naredbi, preporuča se u toj datoteci konfigurirati samo općenite postavke poslužitelja, a za ostale pojedinosti (konfiguracija pojedinih web stranica, modula i sl.) koristiti zasebne konfiguracijske datoteke (npr. one u direktorijima *sites-available*, *conf-available* itd).

Neki zanimljivi primjeri općenitih postavki poslužitelja navedeni su u nastavku:

Timeout: zadana vrijednost ovog parametra je 300 i tada poslužitelj ima 300 sekundi za obradu zahtjeva. Vrijednost je moguće izmijeniti, a preporučeno ju je smanjiti jer je uglavnom za prosječnu obradu dovoljno između 30 i 60 sekundi.

KeepAlive: uz uključenu opciju `KeepAlive` pojedina veza ostaje otvorena za posluživanje višestrukih zahtjeva istog klijenta, dok je u suprotnom za svaki zahtjev potrebno otvoriti novu vezu. Otvaranje nove veze za svaki zahtjev istog klijenta produljuje očekivano vrijeme posluživanja, ovisno o drugim postavkama i prometnom opterećenju.

MaxKeepAliveRequests: ovaj parametar određuje koliko zahtjeva podnosi svaka pojedina veza prije zatvaranja. Što je taj broj veći, posluživanje je učinkovitije. U slučaju da korisnik ne želi ograničiti broj zahtjeva u vezi, ovdje upisuje 0.

KeepAliveTimeout: ova postavka tiče se vremenskog ograničenja prije zatvaranja veze, odnosno koliko će dugo poslužitelj čekati nakon posljednjeg zahtjeva prije nego što prekine vezu.

- **ports.conf**: Ova datoteka sadrži informaciju o tome na kojim pretpostavljenim priključcima (engl. *default ports*) Apache čeka mrežne veze. Zadan je priključak 80 (HTTP) i dodatno priključak 443 (HTTPS) ako je uključen modul za protokol SSL/TLS.

Naredbom

```
$ sudo nano/etc/apache2/ports.conf
```

otvara se datoteka kao na slici 7. Moguće je u ovoj datoteci izmijeniti zadane priključke, no uobičajeno je da se priključci mijenjaju u konfiguraciji pojedinih web stranica/virtualnih domaćina (engl. *virtual hosts*), što će biti pokazano u nastavku dokumenta.



```
GNU nano 3.2 /etc/apache2/ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Slika 5 Sadržaj datoteke *ports.conf*

- **conf-available/**: U ovom direktoriju postavljaju se različite stavke Apache konfiguracije; npr. definira se SSL/TLS konfiguracija i zadane sigurnosne postavke.
- **conf-enabled/**: Direktorij sadrži poveznice na omogućene konfiguracijske datoteke iz *conf-available*. Umjesto ručne izmjene ovog direktorija, preporučeno je koristiti skripte „a2enconf“ i „a2disconf“ za uključivanje odnosno isključivanje ovih konfiguracijskih datoteka.
- **sites-available/**: Mapa koja sadrži konfiguracijske datoteke virtualnih domaćina za različite internetske stranice.
- **sites-enabled/**: Slično kao *conf-enabled*, direktorij sadrži poveznice na omogućene konfiguracijske datoteke iz direktorija *sites-available*, te je preporučeno koristiti skripte „a2ensite“ i „a2dissite“ za uključivanje odnosno isključivanje ovih konfiguracijskih datoteka.
- **envvars**: Varijable okoline Apachea postavljene su u ovoj datoteci, npr.: `APACHE_LOG_DIR`, `APACHE_PID_FILE`, `APACHE_RUN_USERS`.

- **mods-available/**: Ovaj direktorij sadrži konfiguracijske datoteke za učitavanje i postavljanje modula.
- **mods-enabled/**: Slično kao `conf-enabled` i `sites-enabled`, direktorij sadrži poveznice na omogućene konfiguracijske datoteke iz direktorija `mods-available`, te je preporučeno koristiti skripte „`a2enmod`“ i „`a2dismod`“ za uključivanje odnosno isključivanje ovih konfiguracijskih datoteka.
- **magic**: Ovdje se nalaze upute o određivanju MIME tipa datoteke temeljene na prvih nekoliko bajtova.

3.2 Osnovne naredbe

U nastavku su navedene neke osnovne naredbe za upravljanje Apache poslužiteljem na operacijskom sustavu Debian Linux:

- `$ sudo systemctl stop apache2` - zaustavljanje Apachea,
- `$ sudo systemctl start apache2` - pokretanje Apachea,
- `$ sudo systemctl restart apache2` - zaustavljanje i ponovno pokretanje Apachea,
- `$ sudo systemctl reload apache2` - ponovno učitavanje Apachea, ali uspostavljene veze neće se prekinuti; korisno za primjenu izmijenjenih postavki,
- `$ sudo apache2 -l` - ovom naredbom moguće je provjeriti koji su moduli učitani u Apache, kao što je prikazano na slici 6.

```
user@debian:/etc/apache2$ sudo apache2 -l
Compiled in modules:
  core.c
  mod_so.c
  mod_watchdog.c
  http_core.c
  mod_log_config.c
  mod_logio.c
  mod_version.c
  mod_unixd.c
```

Slika 6 Prikaz učitanih modula

3.3 Primjer korištenja Apache HTTP poslužitelja

Stranice koje će Apache posluživati podrazumijevano (engl. *Default*) se nalaze u direktoriju `/var/www`. Moguće je uključiti i druge lokacije izmjenom konfiguracijskih datoteka. Neposredno nakon instalacije tamo se, u mapi `html` nalazi samo datoteka `index.html` koja prikazuje zadanu početnu stranicu Apache HTTP poslužitelja na operacijskom sustavu Debian Linux. Tu datoteku ćemo obrisati, napisati novu i idući put kad pristupimo Apache poslužitelju prikazat će nam se naša nova stranica.

1. Pozicioniramo se u mapu `/var/www/html` gdje se nalazi početna datoteka `index.html`.

```
$ cd /var/www/html
```

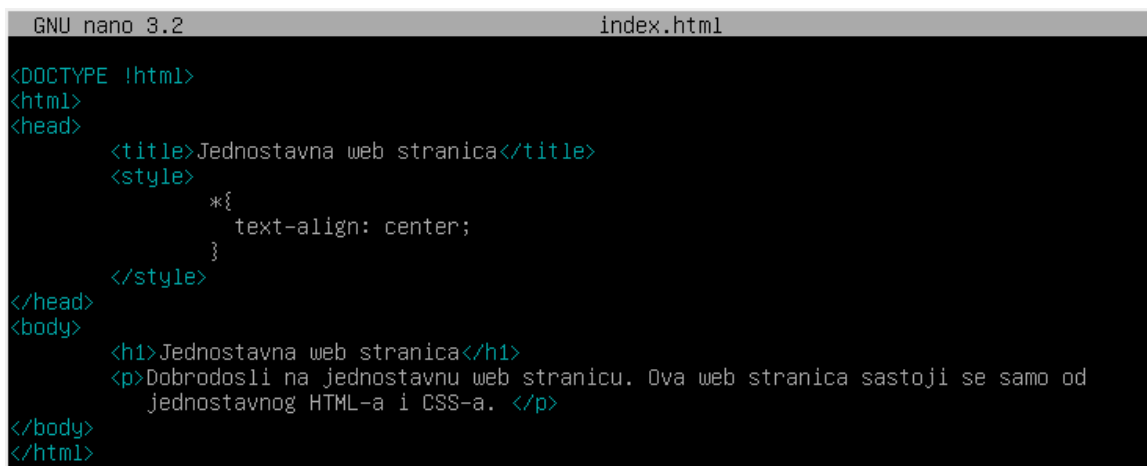
2. Obrišemo postojeću datoteku. Za brisanje ili izmjenu sadržaja ove mape potrebne su administratorske ovlasti, tako da prije naredbe za brisanje moramo unijeti i naredbu `sudo`

```
$ sudo rm index.html
```

3. Stvaramo novu datoteku istog naziva `index.html` i uređujemo je u nekom alatu za uređivanje teksta, npr. Nano. Također je potrebno koristiti naredbu `sudo` jer su i za ovu radnju potrebne administratorske ovlasti:

```
$ sudo nano index.html
```

4. Napišemo jednostavan HTML kao što je prikazano na slici 7:



```
GNU nano 3.2 index.html
<!DOCTYPE html>
<html>
<head>
  <title>Jednostavna web stranica</title>
  <style>
    *{
      text-align: center;
    }
  </style>
</head>
<body>
  <h1>Jednostavna web stranica</h1>
  <p>Dobrodošli na jednostavnu web stranicu. Ova web stranica sastoji se samo od
  jednostavnog HTML-a i CSS-a. </p>
</body>
</html>
```

Slika 7 Jednostavna HTML stranica

5. Sad kad ponovno posjetimo adresu poslužitelja, prikazat će nam se naša jednostavna web stranica:



Slika 8 Apache sad poslužuje našu web stranicu

Proširimo sad ovu datoteku dodavanjem PHP programskog kôda u novu skriptu, `welcome.php`.

```
GNU nano 3.2 welcome.php
<?php
    echo '<p>Pozdrav! Ovaj tekst ispisala je PHP skripta welcome.php</p>';
?>
```

Slika 9 PHP skripta `welcome.php`

Kako bismo mogli koristiti skriptu, uključujemo je u početnu stranicu i stranici mijenjamo nastavak iz `.html` u `.php`:

```
GNU nano 3.2 index.html
<!DOCTYPE !html>
<html>
<head>
    <title>Jednostavna web stranica</title>
    <style>
        *{
            text-align: center;
        }
    </style>
</head>
<body>
    <h1>Jednostavna web stranica</h1>
    <p>Dobrodošli na jednostavnu web stranicu. Ova web stranica sad se, umjesto samo od
    HTML-a i CSS-a, sastoji i od jednostavnog PHP programskog koda. </p>
    <?php
        include 'welcome.php';
    ?>
</body>
</html>
```

Slika 10 Uključivanje PHP skripte u našu novu stranicu

Ponovno unesimo IP adresu Apache poslužitelja i pogledajmo je li sad prikazan i ispis koji bi trebao generirati PHP programski kôd.



Slika 11 Ispis koji bi trebao generirati PHP nije vidljiv

No, rezultat izvođenja PHP kôda nam se nije prikazao. Razlog tome je to što je potrebno eksplicitno instalirati i dodati modul za korištenje PHP programskog jezika. Ako nije

instaliran modul za PHP, neće biti moguće izvođenje PHP kôda na web stranici. Apache modul za rad s PHP-om, na operacijskom sustavu Debian Linux, instalira se naredbom:

```
$ sudo apt install libapache2-mod-php
```

Ako već nije omogućen, Apache modul za PHP može se omogućiti naredbom:

```
$ sudo a2enmod php7.3 (za inačicu PHP-a 7.3)
```

Nakon toga Apache poslužitelj se ponovno pokreće naredbom

```
$ sudo systemctl restart apache2
```

Sad, kad ponovno posjetimo stranicu, vidimo da je prikazan i ispis kojeg generira PHP kôd.



Slika 12 Modul za rad s PHP-om je uspješno dodan

Kao i za korištenje PHP-a, i za ostale dodatne funkcionalnosti (npr. rad s bazama podataka) potrebno je instalirati odgovarajuće softverske pakete.

3.4 Virtualni domaćini

Virtual hosting je metoda posluživanja više web stranica za više odvojenih domena koje se nalaze na jednom, istom poslužitelju koji među njima dijeli svoje resurse poput memorije i procesorskog vremena.

Virtual hosting može biti temeljen na domenskom imenu, IP adresi ili priključku, ali u kontekstu ovog dokumenta koncentrirat ćemo se na *virtual hosting* temeljen na domenskom imenu, tj. *name-based virtual hosting*. *Name-based virtual hosting* s jedne IP adrese poslužuje stranice za više različitih domena. Npr. na IP adresi 192.168.56.101 poslužitelj može primiti zahtjeve za web stranice na dva različita domenska imena, npr. *carnet.hr* i *cert.hr*. Budući da se resursi za obje web stranice nalaze na istoj IP adresi, u HTTP zahtjevu u zaglavlju „Host“ mora biti specificirano na koju se točno domenu zahtjev odnosi, ali to zaglavlje je ionako obavezno u inačici HTTP protokola 1.1.

Osnovna jedinica koja opisuje web stranicu koja se dohvaća naziva se **virtualni domaćin (engl. *virtual host*)**. Ovakav dizajn omogućuje administratoru pohranu i uređivanje

brojnih web stranica na jednostavan način. Klijent se preusmjerava u direktorij sa sadržajem željene stranice i nije mu vidljivo da u stvari komunicira s istim poslužiteljem.

Za postavljanje novog virtualnog domaćina dovoljno je kopirati postojeću datoteku naredbom:

```
$ sudo cp /etc/apache2/sites-available/000-default.conf
    /etc/apache2/sites-available/nazivnovestranice.conf
```

a do uređivanja datoteke zadanog virtualnog domaćina (engl. *default virtual host*) dolazi se sljedećom naredbom:

```
$ sudo nano /etc/apache2/sites-available/000-default.conf
```

Time se dobiva sljedeći prikaz:

```
GNU nano 3.2 /etc/apache2/sites-available/000-default.conf
<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>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Slika 13 Deklaracija zadanog virtualnog domaćina

Iz zaglavlja je vidljivo da će se posluživati zahtjev s bilo kojeg sučelja, sa standardnog HTTP priključka 80 (ako je potrebno posluživanje s nekog drugog priključka, ovdje se upisuje novi broj).

Uz „Server Admin“ stoji adresa e-pošte za slučaj problema s poslužiteljem. Ako u datoteci `/etc/apache2/conf.d/security` postavimo „ServerSignature“ na „Email“, ta adresa će se prikazati na stranici s greškom.

„Server Name“ nije nužno navesti za svakog virtualnog domaćina – ako „Server name“ nije naveden, u tom slučaju će taj virtualni domaćin odgovarati na sve zahtjeve čije ime

domaćina (navedeno u HTTP zaglavlju „Host“) ne odgovara nijednom drugom virtualnom domaćinu.

„Document Root“ pokazuje s koje lokacije Apache dohvaća sadržaj početne stranice. Na Debianu je inicijalno postavljen u `/var/www/html`.

Kada je virtualni domaćin postavljen tako da odgovara željenim zahtjevima, potrebno je još omogućiti tog domaćina, odnosno tu konfiguracijsku datoteku, nakon čega će se osim u `sites-available` datoteka nalaziti i u direktoriju `sites-enabled`. To se postiže naredbom

```
$ sudo a2ensite nazivnovestranice
```

```
i
```

```
$ sudo systemctl restart apache2.service
```

za ponovno pokretanje Apachea i primjenu novih postavki.

Sličan je postupak i za onemogućavanje stranice, korisnik treba unijeti naredbe:

```
$ sudo a2dissite nazivnovestranice
```

```
i
```

```
$ sudo systemctl restart apache2.service
```

3.5 Sigurna konfiguracija Apache HTTP poslužitelja

U ovom poglavlju opisat će se neke od uobičajenih sigurnosnih mjera kojih se programeri web stranica trebaju držati i koje se mogu konfigurirati na Apache HTTP poslužiteljskom softveru kako bi se spriječili napadi na web stranice koje su na njemu pohranjene ili njihove posjetitelje.

- **Redovito ažuriranje softvera**

Iako je na dobrom glasu kad je po pitanju sigurnost, kao i svaki softver, i Apache ima neke pogreške u kôdu koje prođu neprimjetno i otkriju se tek nakon objave inačice softvera. Te pogreške predstavljaju ranjivosti i ako ih napadač otkrije prije no što ih razvojni tim softvera ispravi, postoji mogućnost da iskoristi ranjivost i napadne poslužitelj. Kako bi se minimizirao rizik od takvih napada, bitno je redovito ažurirati Apache. U prethodnim poglavljima spomenuto je kako je ispravna praksa uvijek instalirati softver sa službenog repozitorija operacijskog sustava kako bi ažuriranje svih instaliranih softvera bilo automatizirano prilikom ažuriranja operacijskog sustava naredbama:

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

Kako bi bilo moguće lakše pratiti sigurnosne zakrpe, Apache je objavio i održava Apache HTTP Server Announcements List dostupnu na [poveznici](#) na koju se korisnici mogu predbilježiti (engl. *subscribe*) i biti u toku s novim objavama i sigurnosnim ažuriranjima.

- **Sprječavanje DoS napada**

Mrežni poslužitelji lako se mogu naći na meti DoS (engl. *Denial of Service*) napada kojima je cilj spriječiti poslužitelja da odgovori zahtjevima legitimnih klijenata na način da pretrpaju poslužitelj svojim zahtjevima i zauzmu njegove resurse. Tad poslužitelj neće imati resurse da odgovori i legitimnim zahtjevima već će ih odbijati ili će klijent morati dulje čekati odgovor. Ovakve napade nije moguće u potpunosti spriječiti, ali postoji nekoliko sigurnosnih mjera u Apache HTTP poslužitelju koje se mogu poduzeti kako bi se resursi držali pod kontrolom, tj. kako ne bi došlo do toga da jedan zlonamjerni klijent uspije zauzeti sve resurse. Inače su dobra rješenja za sprječavanje DoS napada dobro konfiguriran vatrozid (engl. *Firewall*) i alati poput Fail2Bana koji je bio tema prethodnog dokumenta Nacionalnog CERT-a [Fail2Ban](#).

No, osvrnimo se i na Apacheovu glavnu konfiguracijsku datoteku **apache2.conf** i nabrojimo samo neke od njenih direktiva koje nam mogu pomoći u minimizaciji problema koje sa sobom donose DoS napadi:

- Direktivom ***RequestReadTimeout*** moguće je ograničiti vrijeme koje će poslužitelj potrošiti na primanje zahtjeva od klijenta, npr. definicija:

```
RequestReadTimeout handshake=5 header=10 body=30
```

znači da će Apache poslužitelj čekati 5 sekundi za dovršavanje TLS trostrukog rukovanja (engl. *Three-way Handshake*), 10 sekundi za primanje zaglavlja zahtjeva (engl. *Request Headers*) i 30 sekundi za primanje tijela zahtjeva (engl. *Request Body*). Ovom direktivom može se spriječiti ostavljanje otvorenih veza i prevelikih zahtjeva.

- Smanjenjem vrijednosti direktiva ***TimeOut*** i ***KeepAliveTimeout*** na svega nekoliko sekundi spriječit će se predugo čekanje poslužitelja na novi zahtjev od klijenta.
- Ispravno konfigurirane direktive ***LimitRequestBody***, ***LimitRequestField***, ***LimitRequestFieldSize***, ***LimitRequestLine*** i ***LimitXMLRequestBody*** ograničit će dodjelu resursa za klijentske zahtjeve
- Direktiva ***MaxRequestWorkers*** postavlja se na način da se definira koji najveći broj istovremenih konekcija poslužitelj može obrađivati bez da ostane bez resursa.

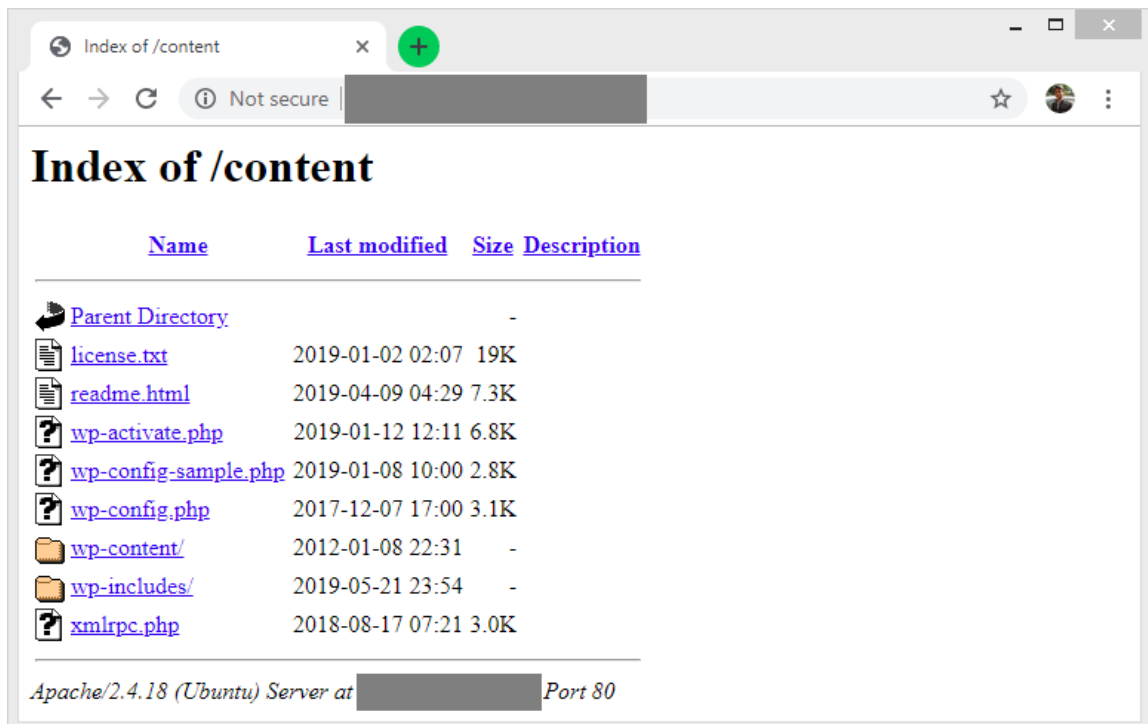
Osim direktiva, postoji i još nekoliko dodatnih modula koji su dostupni na [poveznici](#), a mogu pomoći u ograničenju klijentskog ponašanja i time ublažiti DoS probleme.

- **Zaštita postavki poslužitelja**

Postavke poslužitelja, jednom kada ih administrator postavi i testira, u principu, ne bi trebalo mijenjati. Zbog toga, a u svrhu zaštite, preporuča se ukidanje prava izmjene konfiguracijskih datoteka korisnicima bez administratorskih ovlasti.

- **Zaštita poslužiteljskih datoteka i informacija o softverskoj inačici poslužitelja**

Često je HTTP poslužiteljski softver poput Apachea konfiguriran da, kad se posjeti URL nekog direktorija, poslužitelj prvo posluži datoteku naziva index (npr. `index.html` ili `index.php`). No, ako datoteka s tim nazivom ne postoji, korisniku će se pokazati sadržaj direktorija na poslužitelju kao što je prikazano na slici 14.



Slika 14 Izlistan sadržaj direktorija /content

Loša je sigurnosna praksa dopustiti korisniku da gleda sadržaj direktorija, pogotovo jer su neke stranice namijenjene korisnicima različitih razina ovlasti ili je riječ o skriptama i programskom kôdu čiji sadržaj korisnik ne bi trebao znati jer bi uvidom u kôd možda uspio napasti stranicu ili prikupiti osjetljive podatke.

Kako bi se onemogućilo izlistavanje direktorija na Apache HTTP poslužitelju, jedno praktično rješenje je u direktoriju čije izlistavanje želimo onemogućiti stvoriti datoteku `.htaccess` iz koje će Apache iščitati konfiguraciju specifičnu za taj direktorij.

U novostvorenu datoteku `.htaccess` treba dodati sljedeću liniju kôda koja će onemogućiti prikaz sadržaja direktorija ako ne postoji datoteka naziva `index`:

```
IndexIgnore * ili
```


Options -Indexes

Umjesto sadržaja direktorija prikazat će se odgovor „403 Forbidden“.

Kako bi se uspješno čitala konfiguracija iz `.htaccess` datoteke, potrebno je unutar Apachea omogućiti njeno korištenje koje je podrazumijevano (engl. *default*) onemogućeno.

Osim za sprječavanje izlistavanja direktorija, datoteka `.htaccess` korisna je i za još neke postavke od kojih ćemo nabrojati neke od najvažnijih:

- Mogu se definirati IP adrese kojima je dopušten ili zabranjen pristup poslužitelju.
- Moguće je tražiti korisničko ime i lozinku za pristup određenim direktorijima (korisniku će se pojaviti skočni prozor u web pregledniku).
- Moguće je zabraniti pristup „botovima“. Botovi su programi koji posjećuju web stranice i skeniraju njihov sadržaj u potrazi za sigurnosnim propustima ili npr. adresama e-pošte. Svaka adresa navedena na web stranici može se naći na popisu adresa za slanje *spam* poruka e-pošte upravo zahvaljujući takvim *botovima*. U ovom se kontekstu često spominje datoteka `robots.txt`, ali ona samo obavještava *botove* da stranica na kojoj se nalaze ne želi da je skeniraju. No, zlonamjerni *botovi* to mogu uredno ignorirati. Konfiguracijom u datoteci `.htaccess` moguće je doslovce blokirati česte *botove* dodavanjem linija kôda kao u nastavku:

```
RewriteEngine On
RewriteCond %{HTTP_USER_AGENT} ^BlackWidow [OR]
RewriteCond %{HTTP_USER_AGENT} ^Bot\
mailto:craftbot@yahoo.com [OR]
RewriteCond %{HTTP_USER_AGENT} ^ChinaClaw [OR]
RewriteCond %{HTTP_USER_AGENT} ^Custo [OR]
RewriteCond %{HTTP_USER_AGENT} ^DISCO [OR]
RewriteCond %{HTTP_USER_AGENT} ^Download\ Demon [OR]
RewriteCond %{HTTP_USER_AGENT} ^eCatch [OR]
RewriteCond %{HTTP_USER_AGENT} ^EirGrabber [OR]
RewriteCond %{HTTP_USER_AGENT} ^EmailSiphon [OR]
RewriteCond %{HTTP_USER_AGENT} ^EmailWolf [OR]
RewriteCond %{HTTP_USER_AGENT} ^Express\ WebPictures
[OR]
RewriteRule ^.* - [F,L]
```

- Kada traže web poslužitelje koje mogu napasti, napadači se oslanjaju na informaciju o inačici softvera poslužitelja i/ili operacijskog sustava koja im se može javiti u zaglavlju odgovora na zahtjev ili na karakterističnim stranicama koje se pojavljuju kad se unese određen zahtjev (npr. Apache ima karakterističnu 404 stranicu).

```

▼ Response Headers view source
Accept-Ranges: bytes
Connection: Keep-Alive
Content-Length: 4897
Content-Type: text/html; charset=UTF-8
Date: Sun, 18 Feb 2018 07:01:37 GMT
ETag: "1321-5058a1e728280"
Keep-Alive: timeout=5, max=95
Last-Modified: Thu, 16 Oct 2014 13:20:58 GMT
Server: Apache/2.4.6 (CentOS)

```

Slika 15 Informacija o inačici poslužiteljskog softvera i operacijskom sustavu

Ako napadač zna inačicu softvera poslužitelja na kojem se nalazi web stranica koju želi napasti, može provjeriti postoji li koja javno dostupna ranjivost koju može napasti. Kako bi spriječili napadača (ili mu barem otežali napad), korisno je ukloniti informaciju o softverskoj inačici poslužitelja. Za uklanjanje takvih informacija dodaju se sljedeće dvije linije kôda u datoteku `/etc/apache2/conf-enabled/security.conf` i zatim ponovno pokreće Apache HTTP poslužitelj:

```
ServerTokens Prod
```

```
ServerSignature Off
```

```

▼ Response Headers view source
Accept-Ranges: bytes
Connection: Keep-Alive
Content-Length: 4897
Content-Type: text/html; charset=UTF-8
Date: Sun, 18 Feb 2018 07:05:51 GMT
ETag: "1321-5058a1e728280"
Keep-Alive: timeout=5, max=100
Last-Modified: Thu, 16 Oct 2014 13:20:58 GMT
Server: Apache

```

Slika 16 Nedostupna je informacija o inačici poslužiteljskog softvera

- **Sigurno izvršavanje dinamičkog sadržaja**

Pretpostavimo da na web stranici koja je pohranjena na Apache poslužitelju postoji obrazac za unos korisničke slike. Zlonamjerni korisnik mogao bi, umjesto slike, učitati zlonamjernu PHP skriptu.

Kako bi se spriječilo izvršavanje PHP kôda s mjesta gdje je kôd neočekivan (npr. direktorij `/slike` u koji se pohranjuju sve učitane slike sa stranice), u konfiguracijskim datotekama može se definirati da se u određenim direktorijima nikad ne izvršava kôd nego uvijek prikaže isključivo statički dio stranice (HTML, CSS).

- **Zapisivanje i pregledavanje dnevnika (engl. *logs*)**

Posljednja od navedenih mjera za postizanje sigurnosti web poslužitelja je redovito praćenje dnevnika (engl. *log*) koje poslužitelj bilježi (poslužitelji svaki zahtjev, akciju i pogrešku bilježe u posebne *log* datoteke). Iz tih informacija

zapisanih u dnevnicima iskusni administrator može uočiti pokušaje napada na poslužitelj, njihovu učestalost pa IP adrese s kojih je napad izvršen. Postoje sustavi i alati poput Fail2Bana koji automatizirano prate log datoteke te pri uočavanju bilo kakvih nepravilnosti reagiraju ili obavještavaju administratora.

Osim navedenih, još nekoliko korisnih sigurnosnih postavki i konfiguracija može se pronaći i na sljedećoj [poveznici](#).

4 Zaključak

Apache HTTP poslužitelj popularan je i pouzdan web poslužitelj koji se može koristiti na različitim operacijskim sustavima. Uz ostale komponente operacijskog sustava Linux, sustava za upravljanje bazom podataka MySQL i dinamičkog programskog jezika PHP/Python/Perl čini moćno okruženje za razvoj web aplikacija **LAMP**, na koje se oslanja i jedan od najčešće korištenih alata za izradu web stranica WordPress.

Apache je **modularni** poslužitelj: pruža osnovne funkcionalnosti od kojih su neke navedene i opisane u ovom dokumentu, dok je za napredne opcije korištenja potrebno učitati dodatne module. Otvorenog je kôda i besplatan za preuzimanje, pa njegovom daljnjem razvijanju i održavanju pridonose volonteri iz cijeloga svijeta.