

4.uzdevums

LAMP + SSL

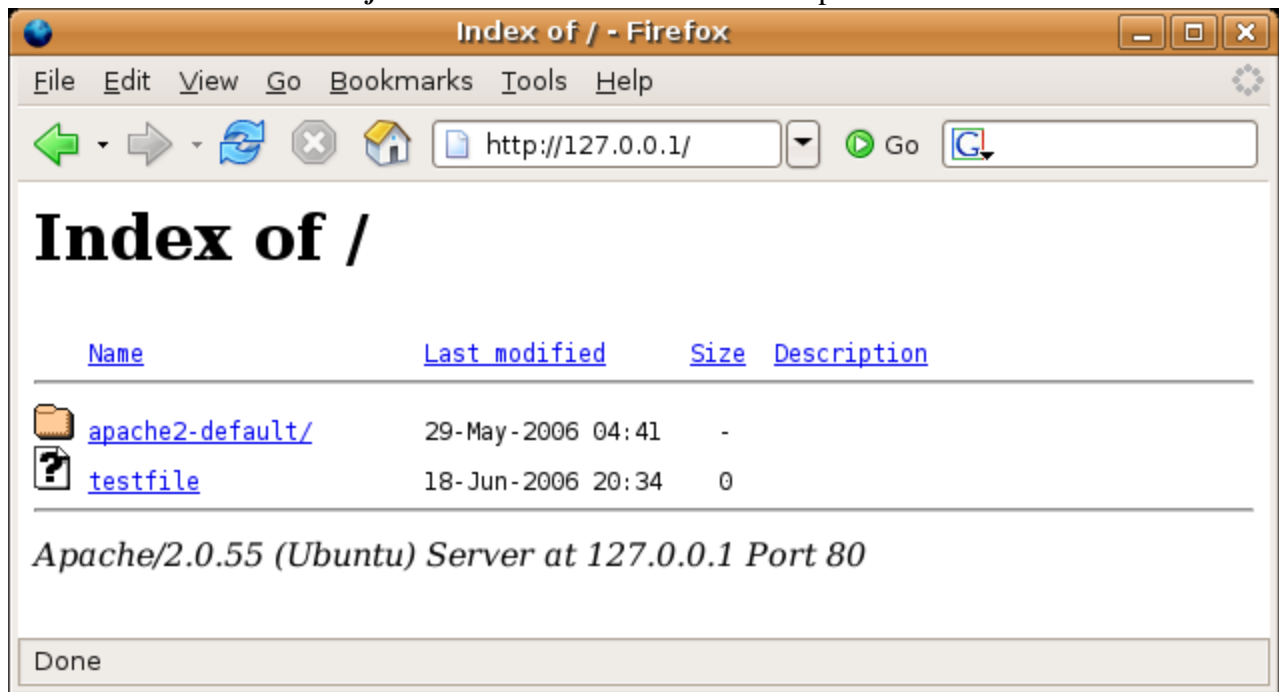
Linux + Apache + MySQL + PHP ir bieži izmantojama konfigurācija, uz kuras bāzes tiek būvēts web-serveris. SSL nodrošina aizsargāto savienojumu ar to. Mūsu uzdevums ir tikt galā ar šīs programmatūras uzstādīšanu.

LAMP

Instalējam Apache serveri:

```
sudo apt-get install apache2
```

Tagad mapē `/var/www` tiks novietots viss, kas būs musū web-serverā. Lai pārbaudītu, ka tas strādā, izveidosim failu `/var/www/testfile` un brauzerā Firefox atversim lapu 127.0.0.1



Pievienojam tām PHP:

```
sudo apt-get install php4
```

restartējam:

```
sudo /etc/init.d/apache2 restart
```

Lai mēs varētu pārbaudīt, ka PHP strādā, veidojam testa failu:

```
sudo gedit /var/www/testphp.php
```

Tajā ierakstam:

```
<?php phpinfo(); ?>
```

Pārlukprogrammā tagad mēs varēsīm redzēt arī *testphp.php* failu. Kad atversīm to, varēsīm redzēt

informāciju par mūsu PHP.

System	Linux nikita-home 2.6.15-25-686 #1 SMP PREEMPT Wed Jun 14 11:34:19 UTC 2006 i686
Build Date	Apr 6 2006 09:35:57
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php4/apache2/php.ini
PHP API	20020918
PHP Extension	20020429
Zend Extension	20050606
Debug Build	no
Zend Memory Manager	enabled
Thread Safety	disabled
Registered PHP Streams	php, http, ftp, https, ftps, compress.bzip2, compress.zlib

This program makes use of the Zend Scripting Language Engine:
Zend Engine v1.3.0, Copyright (c) 1998-2004 Zend Technologies

Tagad mums ir jāuzinstalē MySQL serveris

```
sudo apt-get install mysql-server  
mysqladmin -u root password db_user_password
```

MySQL konfigurācija:

Jāuzstāda jauna parole MySQL root lietotājam (drošības nolūkos galvenais lietotājs nedrīkst būt galvenais servera lietotājs). Taču jāņem vērā, ka šis root lietotājs nav linux sistēmas root lietotājs un, acīmredzamu iemeslu dēļ, parolei nevajadzētu būt tādai pašai kā sistēmas root lietotāja kontam:

```
mysqladmin -u root password jauna-parole
```

un jāsavieno to ar Apache:

```
sudo apt-get install libapache2-mod-auth-mysql
```

```
sudo apt-get install php4-mysql
```

Pēc restārta

```
sudo /etc/init.d/apache2 restart
```

varām redzēt, ka ar MySQL mums arī viss ir forši:

MySQL Support	enabled
Active Persistent Links	0
Active Links	0
Client API version	5.0.22
MYSQL_MODULE_TYPE	external
MYSQL_SOCKET	/var/run/mysqld/mysqld.sock
MYSQL_INCLUDE	-I/usr/include/mysql
MYSQL_LIBS	-L/usr/lib -lmysqlclient

Taisam SSL

SSL, Secure Socket Layer ir aizsargata savienojuma protokols, kuru izstrādāja Netscape. Tās izmanto asimetrisko atslēgu shēmu. Kad servers atsūta klientam informāciju, tas atsūta to iekodētā ar privāto atslēgu (atslēgu, kas ir tikai serveram) veidā. Pirms savienojuma laikā klients prasa serveram sertifikātu, kas ir iedota vienā no Certificate Authority organizācijām. Sertifikātā ir informācija par serveru un publiska atslēga (atslēga, ar kuru var dekodēt informāciju). Klients uzzinā, ka informāciju sutā tieši vajadzīgais serveris ar sekojošo paņemienu. Serveris pēc savienojuma atsūta kādu ziņojumu divas reizes – iekodētā un atklātā. Klients dekodē ziņojumu un salīdzinā ar atklāto. Ja tie ir vienādi – servers ir īstais. Servers arī var paprasīt sertifikātu no klientu.

Mums ir jāizveido sertifikātu (protāms, neautorizēto :)) un jānodrošina SSL-savienojumu.

SSL ir iekļauts Apache2. Taisam sertifikātu:

```
sudo apache2-ssl-certificate
```

```
nikita@nikita-home: ~
File Edit View Terminal Tabs Help
enter your ServerName at the Common Name prompt

If you want your certificate to expire after x days call this programm
with -days x
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to '/etc/apache2/ssl/apache.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:LV
State or Province Name (full name) [Some-State]:Vidzeme
Locality Name (eg, city) []:Riga
Organization Name (eg, company; recommended) []:LU
Organizational Unit Name (eg, section) []:LC
server name (eg. ssl.domain.tld; required!!!) []:ssl.localhost
Email Address []:ndukhovny@inbox.lv
nikita@nikita-home:~$ █
```

Tagad mapē */etc/apache2/ssl/* parādījas divi faili.

Sertifikāts *apache.pem* un privāta atslēga *9f46e9f2.0* (vārds tiek generēts).

Protams, ja mums būtu autorizēts sertifikāts, mēs nedarītu augsā aprakstītas darbības.

Komplektā ar Apache jau ir iekļauts SSL konfigurācijas faila piemērs - */usr/share/doc/apache2/examples/ssl.conf.gz*

Ir jāatver to, un jānokopē sekciju *<VirtualHost *:443>...</VirtualHost>*. Pēc tam to ir jāpievieno failam */etc/apache2/sites-available/default*. Tajā failā ir jāuzlabo bijušo *VirtualHost* ierakstu uz *<VirtualHost *:80>*. Ir arī jānozdeš pirmo rindu *VirtualHost Name **.

Tadā veidā mēs izdarījam divus virtualus hostus – uz viena datora mums ir divi saiti. Viens ir paredzēts HTTP protokolam (80 ports), otrs – HTTPS (443 ports).

Ir vēl pāris lietas, kas mums ir jāizdara. *<VirtualHost *:443>...</VirtualHost>* sekcijā ir jārediģē *DocumentRoot*. Ja gribam, lai SSL mums dublētu parasto HTTP, liekam tur tadu pašu *"/var/www"*. Protams, ir iespēja definēt jebkuru citu mapi.

Pedejā lieta – mums ir jāatļauj 443 portu Apačam. Failā */etc/apache2/ports.conf* ierakstam **Listen 443**.

Tagad, ievēdot Firefox nevis <http://127.0.0.1>, bet <https://127.0.0.1>, varām (pēc tam, kad akceptējam protokolu) redzēt:

Index of / - Firefox

File Edit View Go Bookmarks Tools Help

https://127.0.0.1/ Go

Index of /

Name	Last modified	Size	Description
 apache2-default/	18-Jun-2006 20:32	-	
 testfile	18-Jun-2006 20:34	0	
 testphp.php	18-Jun-2006 20:40	21	

Find: Find Next Find Previous Highlight all Match case

Done 127.0.0.1