CIA Lab Assignment: Web Servers*

A. Safin R. Hussain †

Feedback deadline:
October 8, 2016 00:00 MSK

Abstract

Web servers are an important way of putting information out on the Internet or on an Intranet. Well-known web servers are the Apache HTTP server, `nginx` and `lighttpd`. You will compile and install one of these web servers under Ubuntu on your experimentation machine. The goal is to learn the various configuration options of that server and to understand the concepts.

1 History

The Apache web server originated from the public domain web server of the National Center for Supercomputer Applications (NCSA). After the development of the NCSA web server stopped in 1994, system administrators started creating their own patches for it. A group of system administrators decided to create a web server based on the NCSA web server with the most useful patches included.

Since then the Apache web server has been completely rewritten, and from version 2 onwards the old NCSA code has been fully replaced and is no longer part of the source. One year after its introduction the Apache web server was already the most popular web server and it stayed that way for a long time, though since about 2006 there seems to be a shift and the popularity of Apache is declining.

Nginx (pronounced 'engine-x') originated from the need for a stable server for high-traffic sites. Unlike Apache, it uses an asynchronous event-driven approach, which makes it behave more predictably under high loads. Nginx can also be used as a reverse proxy for many different protocols and services.

lighttpd (pronounce 'lighty') is an open-source webserver aimed for speed, and was orignally developed as a proof-of-concept to solve the c10k problem: how to handle 10.000 connections in parallel on one server. The server runs as a single process with a single thread and non-blocking I/O. It has since been adopted and extended by a larger community.

*Based on earlier work by E.P. Schatborn, A. Bakker, A. van Inge. N. Sijm and C. Dumitru.

†a.safin@innopolis.ru

Question

1. Can you think of reasons for the change in popularity of Apache? Explain.

2 Installation

Which server you should install depends on your table number t:

$$\begin{cases} \text{Apache} & t \bmod 3 = 0 \\ \text{nginx} & t \bmod 3 = 1 \\ \text{lighttpd} & t \bmod 3 = 2 \end{cases}$$

Download the latest release of your designated web server. Be sure to check the signature as per usual.

Question

2. Older source trees like Apache 2.2.*, and nginx 1.6.* are still maintained. Can you think of reasons why?

To compile the server you will be using the standard sequence of `./configure`, `make` and `make install`. Be sure to set the correct options for `configure` when you start compilation. (Hint: read the requirements section carefully ...)

Modern web servers are modular. Modules can either be compiled into the binary, or they can be compiled as Dynamic Shared Objects (DSO) which can be loaded at runtime. Make sure at least the module with TLS/SSL support is enabled.

Be sure to first remove all other installations since they might interfere with yours, then compile and install the software in `/usr/local/<servername>`, where servername is `apache`, `nginx`, or `lighttpd`. Make sure that the web server starts at boot, using the method prescribed by the server.

The servers use `/usr/local/apache/conf/httpd.conf`, `/usr/local/nginx/conf/nginx.conf` and `/usr/local/lighttpd/lighttpd.conf`, respectively, as their main configuration file. Read the example files carefully and look up things that are not clear to you. Then configure the basic settings like the webmaster email address, the web server name, the root directory for the web documents and the port the web server listens on. You can check the syntax of the configuration file using `apachectl -t`, `nginx -t` and `lighttpd -t`, respectively.

3 Virtual Hosts

A much used functionality that these three web servers provide is the hosting of multiple domains from a *single* web server. This "virtual host" functionality resembles the MTA setup we worked with last week. The virtual domains exist only as resource records on the DNS server and as data on the web server.

Questions

3. Implement virtual hosting in the web server for the virtual domains `anyname.st[X].os3.su`, `anyname.st[X-1].os3.su` and `anyname.st[X+1].os3.su`. Show your configuration.
4. Create a simple, unique HTML page for each virtual host make sure that the server can correctly serve it. Please include virtualhost name to the main page.
5. Use `curl` to display the contents of a full HTTP/1.1 session served by your server. Explain the meaning of each request and reply header.

4 Encryption

It may be necessary to encrypt an HTTP session so that others cannot listen in. It is essential for online banking or financial transactions. This is what the HTTPS protocol is used for. It is a connection over port 443 using SSL or TLS encryption.

6. Configure your web server to support TLS. Make sure you disable the SSLv2+3 and TLSv1.0+1.1 protocols as they are shown to be unsafe.

   - It is not difficult to set up your server to support TLS. It *is* however difficult to do it *correctly*. Be sure you know what you are doing and read the server's documentation on TLS carefully[1].
   - Read the TLS configuration options carefully before you start. Adapt it to your needs and make sure the TLS module is loaded.

7. List the encryption standards / cipher suites that the web server supports with the standard configuration file. Select one and explain all its components.

8. Describe how you created your own certificate for your web server.

9. You can test your secure web server using a web browser, but you can also use `openssl` and `curl`. Test your web server using both these tools, and report your findings.

10. Can you enable HTTPS for all your virtual hosts? Explain how you can configure one web server to serve multiple TLS enabled domains.

5 Web Server Security

The security of a web server is largely derived from the access rights given to documents on the server. Apache has a reasonably good reputation in the area of security. It is mostly the code that users[2] add to the web server that creates security problems.

11. Investigate what configuration options there are on your assigned Web server that govern access rights. What ways are there to use these options on documents (folder access rights, IP acl, authentification methods, htaccess, robot.txt)?

12. Now create two web pages, one with a simple SSI instruction and one with a simple Perl/Python/Ruby CGI script (beware of Shellshock). Set up your web server so that only code on these pages can be executed (if possible).

6 Extra Assignments (Optional)

6.1 Web Server Performance

Web server performance is a very important issue as this directly impacts user experience and can affect operation costs.

[1] Apache HOWTO quote:

> [...] but always try to understand the stuff before you use it. Nothing is worse than using a security solution without knowing its restrictions and coherences.

[2] The users are the people using your web server to make their content available on the Internet.

13. Investigate what configuration options there are that can potentially improve the performance of the web server. Also look at how you can check the (current) load on the web server using e.g. the Apache `mod_status` module.

14. Using a standard benchmarking tool (e.g. `ab`, `siege`, etc.) evaluate the performance of your server before and after optimizations for both the static page and the dynamic page. Try to maximize the number of requests per second. Explain all the changes made.

6.2 Logging

Checking the logging information on your web server is important to discover problems and/or attacks on your web server. The web servers have different configuration options to adapt the log information to your needs. This allows you to extract useful information from the logfile.

15. Define your own log format containing information you deem important. Use *Conditional logging* to add User-Agent and Referrer information to request logs that generated an error.