heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Technologies Internet (TEI)
Laboratoire HTTP Infrastructure 1

Olivier Liechti
April 15, 2013

Contents

1 Guidelines

- The lab can be done either individually or in teams of two students. Every student is expected to do the complete setup and testing procedure, every student is expected to have a working system at the end of the lab.
- Questions have to be answered in english.
- The answers should be packaged in a zip file named `lastname1_lastname2.zip`, with the following structure:
  - `doc`: folder containing your answers and detailed explanations, in one file named `lastname1_lastname2.pdf`. Please make sure to use lab task numbers as section headings in your text.
  - `src`: folder containing the source files related to the lab tasks; the files should be organized in sub-folders, where the name of the sub-folder corresponds to the lab task (e.g. `src/task2.2`).
  - `tests`: folder containing the test scripts (shell scripts, jmeter scripts, etc.)
  - `screenshots`: folder containing the raw screenshots; please also include the screenshots directly in the main document.
- Please send your files to `olivier.liechti@heig-vd.ch` until April 29th (07h00) at the latest.
- On April 29th, be ready to do a demo of your setup on your machine. You might be asked to reproduce the procedures and the tests, as well as to answer additional questions during the lecture and/or the lab session.

2 Setting up a web environment

You work in the IT department of a company, which develops and operates on-line project management tools. As a system engineer, you are part of the team that designs, builds and maintains the technical infrastructure for the company. You are responsible for the technology stack from the hardware level up to the middleware level. You collaborate with the software developers, who design and build the application-level components.

Most of the components are written in Java and are deployed in a Java EE application server. For a new project, however, the developers would like to create a simple web front-end in PHP. To support this activity, you have been requested to setup a web environment. You have been asked to design both the development environment and the production environment. The development environment is what all developers will use on their local machine on a daily basis. The production environment is what customers will use on a daily basis. This week, you will focus on the development environment.

2.1 Lab Task 1: setup of development environment

The web environment consists of three main components. The first component is the apache web server, the second one is the PHP engine and the third one is the MySQL relational database management system. Your first task is to propose one method for installing and configuring this web environment.

There are several methods to install an apache web server on a developer's machine, with their respective pros and cons. Your first task is to think about what these methods are, to think about their relative advantages and disadvantages and finally to recommend one the methods. There is no right or wrong choice, but it is very important for you to explain clearly why you have the choice and to provide clear and convincing arguments. It is also important for you to highlight the possible limitations of your proposed setup.

Lab Task 1.1 List and describe at least 3 different methods installing the web environment on a developer's machine. We expect at least 500 words.

Lab Task 1.2 List at least 3 different criteria to consider when selecting one of these installation methods. Explain how you weight these criteria and justify your decision. Again, remember that there is no right or wrong method but that you have to provide clear motivations for your choice. We expect at least 300 words.

Lab Task 1.3 Give the reference(s) that you have followed in order to perform your installation.

Lab Task 1.4 Describe the steps of the installation procedure and illustrate them with appropriate screenshots and/or shell commands. What you write here is what you would provide to the software developers in your company, so it must be clear and comprehensive.

Lab Task 1.5 Describe at least 2 limitations or constraints associated with the installation method that you have selected. We expect at least 200 words.

2.2 Lab Task 2: create a simple PHP application

Once you have installed and configured the web server, you will want to validate your setup. Obviously, you will want to make sure that developers can create and deploy PHP applications. You will also want to validate that PHP sessions work as expected (this will be an issue to address when designing the production environment).

Lab Task 2.1 Specify what the application will do and explain how it will illustrate the use of PHP sessions. The application should be very simple. *Hint 1: think about a page displaying a page hit counter*. *Hint 2: at some point, you will have more than one web server behind a load balancer.*

Lab Task 2.2 Provide the source code for your application. Do not forget that the source code should have a proper header, have comments, etc.

Lab Task 2.3 Provide screenshots and clear explanations, to show evidence that sessions are properly managed by the server.

Lab Task 2.4 If sessions were not managed properly, what would you observe?

Lab Task 2.5 As a user (not a developer), how can you start a new session?

Lab Task 2.6 At the HTTP protocol level, explain clearly how sessions are managed. We expect at least 300 words.

2.3 Lab Task 3: deploy an existing open source PHP application

To do a final validation of your setup (and to prepare follow-up tasks), you now have to deploy an existing PHP application. There are many open source applications that you can use for this purpose, such as the wordpress blog system (`http://wordpress.org/`) or the elgg social networking engine (`http://www.elgg.org/`).

Lab Task 3.1 Select one open source PHP application and provide the reference. Check the technical requirements and installation procedure.

Lab Task 3.2 Perform and document the installation and highlight issues that you might have encountered.

Lab Task 3.3 Provide screenshots to provide evidence that the application has been successfully installed and is properly running.

3 Testing the HTTP stack

There are different methods and tools for observing and analyzing HTTP traffic, and for troubleshooting web infrastructures. One objective of the lab is to experiment with these tools and to understand when they can be applied.

3.1 Lab Task 4: testing with wireshark

Like any network protocol, HTTP lends itself very well to analysis with tools like `tcpdump` or wireshark. To start your analysis of the HTTP protocol, after reading the RFC, you will use wireshark to examine the traffic exchanged by a client and a server.

Hint: when you perform this task, make sure that you only have one browser (and one tab) opened. Otherwise, you might see requests others than the ones you explicitly generate.

Lab Task 4.1 Describe the formal structure of an HTTP request (this is information that you will find in the RFC) and illustrate this structure with an example.

Lab Task 4.2 Describe the formal structure of an HTTP response (this is information that you will find in the RFC) and illustrate this structure with an example.

Lab Task 4.3 Describe 2 important headers that are sent by the client as part of the HTTP request. Describe 2 important headers that are sent by the server as part of the HTTP response. We expect at least 100 words for each header description.

Lab Task 4.4 Start a wireshark session and visit the following site: `http://nodejs.org`. Take snapshots and describe what you observe in wireshark.

Lab Task 4.5 How many requests do you observe? Why do you see more than one?

Lab Task 4.6 You should see a request for a resource named `__utm.gif`. What is the purpose of this request? We expect you to do investigation and research on this advanced question and provide an explanation of at least 300 words.

3.2 Lab Task 5: testing with telnet

Since HTTP is an application protocol that operates on top of TCP, it is possible to use a telnet client to connect to an HTTP server. In this case, what would normally be sent by the HTTP client (e.g. by the a web browser) has to be typed in a terminal. This rudimentary testing method is useful to make initial experiments with the protocol (by typing HTTP requests yourself, you will get a good understanding of the protocol). It is also very useful to perform initial tests when troubleshooting issues with web

infrastructures (e.g. at the beginning of a troubleshooting session, the first step is often to check if the HTTP server is alive and accepting client requests). When using this method, remember that the standard HTTP port is 80 (therefore, you will need to open the telnet connection on that port).

Lab Task 5.1 Use telnet in order to send HTTP requests to public web sites; keep track of what you typed as well as the responses you got from the servers.

Lab Task 5.2 Provide screenshots and explain how you interpret the answers of the servers. We expect to see requests sent to at least three different servers, with detailed explanations about the headers being sent and returned.

3.3 Lab Task 6: testing with firebug

Firebug is a browser extension that is very useful when developing or troubleshooting web applications (also when AJAX is involved in the design). Firebug offers many features, and one of them is to expose HTTP traffic to the user directly in the browser (in other words, it gives

Lab Task 6.1 Install Firebug in your browser (Firebug was initially developed as a Firefox extension; a lite version is available for Chrome).

Lab Task 6.2 Activate the `Net` panel and visit a number of sites, to get a first feel for the tool.

Lab Task 6.3 Take screenshots and describe how the information provided by Firebug is similar to the information provided by wireshark.

Lab Task 6.4 Under the `Net` panel, what is the purpose of the `XHR` tab? Go to `http://www.google.com`, make sure that Google Instant is turned on and start typing a query in the search box. Observe what is happening in the XHR panel and describe what is happening.

Lab Task 6.5 Using a screenshot as evidence, indicate what kind of http server is used by the `http://www.qoqa.ch` web site (i.e. is is apache, IIS, etc.?). From a security point of view, what do you think about your ability to answer this question? What would be your advice to the IT engineers working on the the qoqa web infrastructure?

3.4 Lab Task 7: testing with jmeter

Lab Task 7.1 Read Sections 1 to 6, as well as Section 16.4 or the JMeter User Manual (`http://jakarta.apache.org/jmeter/usermanual`)

Lab Task 7.2 Perform and document and installation of JMeter on your machine.

Lab Task 7.3 Implement a first test plan to test the simple application created in Lab Task 2. Run the test, document and analyze the test results. We expect to see screenshot and a description of at least 300 words.

Lab Task 7.4 Implement two different test plans to test the application installed in Lab Task 3. In each case, start by defining the use case that you want to test (in an e-commerce web site, the use case might be "as a customer, I want to log into the system" or "as a customer, I want to put items in my shopping cart and check-out"). Take a screenshot for each step of the process associated to the use case. Then, proceed to implement the test plan in JMeter and include one more listeners, to visualize results. Explain why you have selected the particular listener(s).

Lab Task 7.5 Execute the test plans, collect and analyze the results.