UNIT 3.
WEB SERVER ADMINISTRATION

Web Applications Deployment
CFGS DAW

Author: Carlos Cacho López

Reviewed by: Lionel Tarazón Alcocer

lionel.tarazon@ceedcv.es

2019/2020

License

Nomenclature

During this unit we are going to use special symbols to distinct some important elements. The symbols are:

 Interesting

 Important

 Attention

Index

UT03. WEB SERVER ADMINISTRATION

1. INTRODUCTION

As we said in unit 1, a **web server** is a program or a set of them which offers web services over a network. Its aim is to offer the means and necessary resources for two or more programs to communicate independently of how they have been created and their characteristics.

In this unit we will learn how to manage these servers and their most important features.

2. HTTP PROTOCOL

Hypertext Transfer Protocol (**HTTP**) is an application protocol for transmitting hypermedia documents, such as HTML. It was designed for communication between web browsers and web servers, though it can be used for other purposes as well. It follows a classical client-server model, with a client opening a connection (port **80**), making a request, and waiting until it receives a response.

The main disadvantage is that all the informations sent in that connection is not encrypted, so a third party can read passwords by sniffing the network.

3. HTTPS PROTOCOL

The **HTTPS** protocol (Hypertext Transfer Protocol Secure) is the secure version of the HTTP protocol. It is not a protocol in itself, but the union of two: HTTP protocol plus a layer with the **SSL** or **TLS** protocols. It uses the port **443**.

SSL (Secure Socket Layer) is the secure protocol that is added to HTTP. Nowadays version 3.0 is used, but there's also a derived protocol: the TLS (Transport Layer Security). These protocols assure integrity, genuineness, confidentiality and not rejection between client and server.

There are two ways to configure it: the server shows its identity or the client and server use digital certificates.

4. DIGITAL CERTIFICATE

A digital certificate is an electronic "passport" that allows a person, computer or organization to exchange information securely over the Internet using the **public key infrastructure (PKI)**. A digital certificate may also be referred to as a public key certificate.

Just like a passport, a digital certificate provides identifying information, is forgery resistant and can be verified because it was issued by an official, trusted agency. The certificate contains the name of the certificate holder, a serial number, expiration dates, a copy of the certificate holder's public key (used for encrypting messages and digital signatures) and the digital signature of the **certificate-issuing authority** (CA) so that a recipient can verify that the certificate is real.

To provide evidence that a certificate is genuine and valid, it is digitally signed by a **root certificate** belonging to a trusted certificate authority. Operating systems and browsers maintain lists of trusted CA root certificates so they can easily verify certificates issued and signed by CA's. When PKI is deployed internally, digital certificates can be **self-signed**.

5. MIME TYPE

The Internet standard **MIME** (Multipurpose Internet Mail Extensions) extends the format of email to support:

- Text in character sets other than ASCII.
- Non-text attachments: audio, video, images, application programs, etc.
- Message bodies with multiple parts (multi-part).
- Header information in non-ASCII character sets

When a browser tries to open a file, the MIME standard helps it knowing with what type of file it is working and the program that can open it correctly. If the file does not have a MIME type specified, the program supposes that the type is the file extension. For instance, in the case of a file with a .txt extension, the program supposes it is a text file.

The structure of a MIME type is very simple; it consists of a type and a subtype, two strings, separated by a `'/'`. No space is allowed.

The *type* represents the category and can be a *discrete* or a *multipart* type. The *subtype* is specific to each type.

Some examples of discrete types are:

Type	Description	Example of typical subtypes
text	Represents any document that contains text and is theoretically human readable	text/plain, text/html, text/css, text/javascript
image	Represents any kind of images. Videos are not included, though animated images (like animated gif) are describes with an image type.	image/gif, image/png, image/jpeg, image/bmp, image/webp
audio	Represents any kind of audio files	audio/midi, audio/mpeg, audio/webm, audio/ogg, audio/wav
video	Represents any kind of video files	video/webm, video/ogg
application	Represents any kind of binary data.	application/octet-stream, application/pkcs12, application/vnd.mspowerpoint, application/xhtml+xml, application/xml, application/pdf

For text documents without specific subtype, text/plain should be used. Similarly, for binary documents without a specific or known subtype the application/octet-stream should be used.

Multipart types indicates a category of document that are broken in distinct parts, often with different MIME types. It is a way to represent composite document. With the exception of multipart/form-data, that are used in relation of HTML Forms and POST method.

6. WEB SERVER ADMINISTRATION

6.1 Modules

All web servers should give us the possibility to get new functionalities (or remove them) in an easy way and according to our needs. Some module examples are: SSL, PHP, LDAP, etc.

6.2 Virtual hosts/Server blocks

The term Virtual host is used in Apache and Server block is used in Nginx.

Both refer to the practice of running more than one web site (such as course1.daw.net and course2.daw.net) on a single machine. Although they are running on the same physical machine, is not apparent to the end user.

Virtual hosts and Server blocks can be "IP-based", meaning that you have a different IP address for every web site, or "name-based", meaning that you have multiple names running on each IP address.

IP-based virtual hosts or server blocks use the IP address of the connection to determine the correct virtual host to serve. Therefore you need to have **a separate IP address for each host**.

In name-based virtual hosts or server blocks, the server relies on the client to report the hostname as part of the HTTP headers. Using this technique, many **different hosts can share the same IP address**.

Name-based virtual hosts or server bocks are usually simpler, since you only need to configure your DNS server to map each hostname to the correct IP address, and then configure the Web Server to recognize the different hostnames.

6.3 Authentication and authorization

First of all, it is important to know that **authentication** is any process by which you verify that someone is who they claim to be, and **authorization** is any process by which someone is allowed to be where they want to go, or to have information that they want to have.

To do so there are some methods such as **Basic** or **Digest**.

In both of them the user has an id and a password to access resources. The main differences are that Basic does not encrypt the password (Digest does) and Digest works with domains (Basic doesn’t).

6.4 Logs

The registry files or logs allow to see the performance of the web server as well as to perform statistics or analysis.

There are logs that store the errors that happen in the server or who has accessed to any resource.

7. BIBLIOGRAPHY

[1] Álvarez Villanueva, Cristina (2015): “Web servers administration”, CEEDCV

[2] Cuesta, Sergio (2014): *Despliegue de aplicaciones web*

[3] https://httpd.apache.org/docs

[4] https://developer.mozilla.org/es/docs