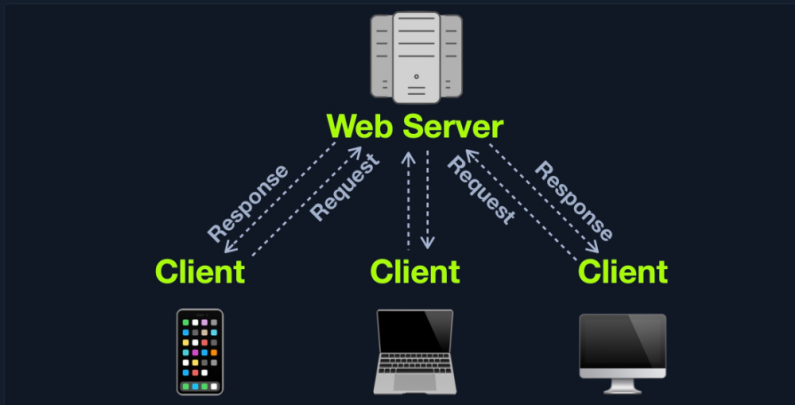


Web Servers

A **web server** is an application that runs on the back end server, which handles all of the HTTP traffic from the client-side browser, routes it to the requested pages, and finally responds to the client-side browser. Web servers usually run on TCP **ports 80 or 443**, and are responsible for connecting end-users to various parts of the web application, in addition to handling their various responses.

Workflow

A typical web server accepts HTTP requests from the client-side, and responds with different HTTP responses and codes, like a code **200 OK** response for a successful request, a code **404 NOT FOUND** when requesting pages that do not exist, code **403 FORBIDDEN** for requesting access to restricted pages, and so on.



The following are some of the most common **HTTP response codes**:

| Code | Description |
|----------------------------------|---|
| Successful responses | |
| 200 OK | The request has succeeded |
| Redirection messages | |
| 301 Moved Permanently | The URL of the requested resource has been changed permanently |
| 302 Found | The URL of the requested resource has been changed temporarily |
| Client error responses | |
| 400 Bad Request | The server could not understand the request due to invalid syntax |
| 401 Unauthorized | Unauthenticated attempt to access page |
| 403 Forbidden | The client does not have access rights to the content |
| 404 Not Found | The server can not find the requested resource |
| 405 Method Not Allowed | The request method is known by the server but has been disabled and cannot be used |
| 408 Request Timeout | This response is sent on an idle connection by some servers, even without any previous request by the client |
| Server error responses | |
| 500 Internal Server Error | The server has encountered a situation it doesn't know how to handle |
| 502 Bad Gateway | The server, while working as a gateway to get a response needed to handle the request, received an invalid response |
| 504 Gateway Timeout | The server is acting as a gateway and cannot get a response in time |

Web servers also accept various types of user input within HTTP requests, including text, **JSON**, and even binary data (i.e., for file uploads). Once a web server receives a web request, it is then responsible for routing it to its destination, run any processes needed for that request, and return the response to the user on the client-side. The pages and files that the webserver processes and routes traffic to are the web application core files.

The following shows an example of requesting a page in a Linux terminal using the **cURL** utility, and receiving the server response while using the **-I** flag, which displays the headers:

```
Web Servers
MisaelMacias@htb[/htb]$ curl -I https://academy.hackthebox.com

HTTP/2 200
date: Tue, 15 Dec 2020 19:54:29 GMT
content-type: text/html; charset=UTF-8
...SNIP...
```

While this **cURL** command example shows us the source code of the webpage:

```
Web Servers
MisaelMacias@htb[/htb]$ curl https://academy.hackthebox.com
```

[? Go to Questions](#)

Table of Contents

Introduction to Web Applications

| | |
|------------------------|---|
| Introduction | ✓ |
| Web Application Layout | ✓ |
| Front End vs. Back End | ✓ |

Front End Components

| | |
|------------------------------|---|
| HTML | ✓ |
| Cascading Style Sheets (CSS) | ✓ |
| JavaScript | ✓ |

Front End Vulnerabilities

| | |
|-----------------------------------|---|
| Sensitive Data Exposure | ✓ |
| HTML Injection | ✓ |
| Cross-Site Scripting (XSS) | ✓ |
| Cross-Site Request Forgery (CSRF) | ✓ |

Back End Components

| | |
|-------------------------------|---|
| Back End Servers | ✓ |
| Web Servers | ✓ |
| Databases | ✓ |
| Development Frameworks & APIs | ✓ |

Back End Vulnerabilities

| | |
|----------------------------|---|
| Common Web Vulnerabilities | ✓ |
| Public Vulnerabilities | ✓ |

Next Steps

| | |
|------------|---|
| Next Steps | ✓ |
|------------|---|

My Workstation

OFFLINE

Start Instance

00 / 1 spawns left

```
</doctype html>
<html lang="en">
<head>
<meta charset="utf-8" />
<title>Cyber Security Training : HTB Academy</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Many web server types can be utilized to run web applications. Most of these can handle all types of complex HTTP requests, and they are usually free of charge. We can even develop our own basic web server using languages such as [Python](#), [JavaScript](#), and [PHP](#). However, for each language, there's a popular web application that is optimized for handling large amounts of web traffic, which saves us time in creating our own web server.

Apache



[Apache](#) or [httpd](#) is the most common web server on the internet, hosting more than [48%](#) of all internet websites. [Apache](#) usually comes pre-installed in most [Linux](#) distributions and can also be installed on Windows and macOS servers.

[Apache](#) is usually used with [PHP](#) for web application development, but it also supports other languages like [.Net](#), [Python](#), [Perl](#), and even OS languages like [Bash](#) through [CGI](#). Users can install a wide variety of [Apache](#) modules to extend its functionality and support more languages. For example, to support serving [PHP](#) files, users must install [PHP](#) on the back end server, in addition to installing the `mod_php` module for [Apache](#).

[Apache](#) is an open-source project, and community users can access its source code to fix issues and look for vulnerabilities. It is well-maintained and regularly patched against vulnerabilities to keep it safe against exploitation. Furthermore, it is very well documented, making using and configuring different parts of the webserver relatively easy. [Apache](#) is commonly used by startups and smaller companies, as it is straightforward to develop for. Still, some big companies utilize Apache, including:

[Apple](#)

[Adobe](#)

[Baidu](#)

NGINX



[NGINX](#) is the second most common web server on the Internet, hosting roughly [39%](#) of all internet websites. [NGINX](#) focuses on serving many concurrent web requests with relatively low memory and CPU load by utilizing an async architecture to do so. This makes [NGINX](#) a very reliable web server for popular web applications and top businesses worldwide, which is why it is the most popular web server among high traffic websites, with around 60% of the top 100,000 websites using [NGINX](#).

[NGINX](#) is also free and open-source, which gives all the same benefits previously mentioned, like security and reliability. Some popular websites that utilize [NGINX](#) include:

[Google](#)

[Facebook](#)

[Twitter](#)

[Cisco](#)

[Intel](#)

[Netflix](#)

[HackTheBox](#)

IIS



[IIS](#) ([Internet Information Services](#)) is the third most common web server on the internet, hosting around [15%](#) of all internet web sites. [IIS](#) is developed and maintained by Microsoft and mainly runs on Microsoft Windows Servers. [IIS](#) is usually used to host web applications developed for the Microsoft .NET framework, but can also be used to host web applications developed in other languages like [PHP](#), or host other types of services like [FTP](#). Furthermore, [IIS](#) is very well optimized for Active Directory integration and includes features like [Windows Auth](#) for authenticating users using Active Directory, allowing them to automatically sign in to web applications.

Though not the most popular web server, many big organizations use [IIS](#) as their web server. Many of them use Windows Server on their back

end or rely heavily on Active Directory within their organization. Some popular websites that utilize IIS include:

Microsoft



Office365

Skype

Stack Overflow


Dell

Aside from these 3 web servers, there are many other commonly used web servers, like [Apache Tomcat](#) for [Java](#) web applications, and [Node.JS](#) for web applications developed using [JavaScript](#) on the back end.

☐ Enable step-by-step solutions for all questions  

Questions

Answer the question(s) below to complete this Section and earn cubes!

+1  If a web server returns an HTTP code 201, what does it stand for?

Created

 Submit

 Hint

 Previous

Next 

 Mark Complete & Next

