**DENNIS GITUHU MWANGI SCT212-0513/2021**

**INTERNET TECHNOLOGIES**

**SEVER SIDE PROGRAMMING**

Server-side scripting, the back-end web development technology, can be defined as a method used in web development in order to produce a customized response for each request from the user to the website. Google uses these scripts to fill in your search term, to place Ads, or to find out the thing you are searching for and so on, while e-commerce sites like [Amazon](https://www.amazon.in/) use the scripts to list products and record what you have bought.

Uses can be generalized to Process user input, Display pages, Structure web applications, Interact with permanent storage ([**SQL**](http://www.ndimensionz.com/kb/sql-injection-a-method-to-gain-control-over-vulnerable-database/), files), etc.

This involves running of user/process request on the originating server by employing scripts on the server. Many languages including ActiveVFP, ASP, C, PHP, [Python](https://www.python.org/), etc. can be used to create these scripts and are generally used to deliver dynamic content.

Many server-side scripting languages are cross-platform and are open for anyone to use. This is accomplished with the help of a Common Gateway Interface ([**CGI**](http://www.ndimensionz.com/kb/dual-php-with-apache-cgi-in-plain-server/)).

An alternative for the server is to deliver static web pages. The **static page** file has a **.htm or .html extension** whereas **dynamic pages** have a **.cgi or .php** file extension. Once a request with one of these extensions are received, the [**web server**](http://apachebooster.com/kb/types-of-open-source-web-server/) passes the request to CGI, which correctly interprets it and scripting is executed in the particular scripting language.

Then the output is passed back to the server which is to be delivered in response to the user request. Server-side scripts written in Python enables us to reuse most of the knowledge from writing Python programs on the desktop. [MySQL](https://www.mysql.com/) is utilized for storing information.

**ADVANTAGES OF SERVER-SIDE PROGRAMMING.**

1. Server-side scripting prevents increasing of the load as it **does not require plugins or browser scripting technology** (such as Javascript). Overloading leads to [**problems like slow loading**](http://www.apachebooster.com/#features), high CPU usage, and even freezing.
2. It is used to create pages dynamically on the fly. Based on the user interaction, new pages can instantly be created.
3. Server-side scripting is necessary to run dynamic pages on browsers that do not fully support [Javascript](https://www.javascript.com/" \t "_blank).
4. Server-side scripting does not depend on [**browser**](http://apachebooster.com/kb/basics-of-web-browser-web-server-big-data-and-hadoop/) processing as all the processing is performed on the server side.
5. Website owners can create their own applications and make use of CMS (content management system) to easily create and update content on the web without coding as the Server-side scripting languages like PHP can be configured to run [**CMS applications, like WordPress and Joomla**](http://apachebooster.com/kb/compare-top-cms-sites-wordpress-vs-drupal-vs-joomla/).
6. As the scripting is done on the server, it is not sent back to the browser, which prevents it from being cloned, copied, or scrutinized for [**hacking vulnerabilities**](http://www.ndimensionz.com/vulnerability-assessment-penetration-testing.php).
7. Loading time of the web pages is often reduced with Server-side scripting which helps to improve your site’s Google ranking.
8. An increased security is ensured for user privacy and is the preferred choice for e-commerce, membership and social media sites.

**DISADVANTAGES OF SERVER-SIDE PROGRAMMING**

* Server-side scripting is slow at times like: 1.) If the user disconnects from the internet, or 2.) The web hosting is down, Then the script may take a long time to execute.
* The scripting software has to be installed on the server.
* New security concerns are associated with the dynamic scripts as in some cases hackers exploit the code flaws to gain access to servers.
* A [**database**](http://www.ndimensionz.com/kb/importance-of-database-security-and-requirements/) is required to store the dynamic data that needs a regular backing up and has to be kept secure.
* Here the **pages have to be refreshed often** in order to show the dynamic content. Ajax, an innovative method of exchanging data with a server is used by the developers which resolve the problem.
* Websites using large applications and with heavy traffic have to make use of more powerful hosting methods like dedicated servers or cloud hosting to cope with demand

**SECURITY OF THE INTERNET.**

**COMMON THREATS TO USERS**

Internet security threats impact the network ,data security and other internet connected systems .Cyber criminals have evolved several techniques to threat privacy and intergrity of banksaccount business, and organization

* Malware. Malware is malicious software such as spyware, ransomware, viruses and worms. ...
* Emotet. ...
* Denial of Service. ...
* Man in the Middle. ...
* Phishing. ...
* SQL Injection. ...
* Password Attacks. ...
* The Internet of Thing

SECURE HTTP

The security of HTTPS is that of the underlying TLS, which typically uses long-term [public](https://en.wikipedia.org/wiki/Public-key_cryptography) and private keys to generate a short-term [session key](https://en.wikipedia.org/wiki/Session_key), which is then used to encrypt the data flow between the client and the server. [X.509](https://en.wikipedia.org/wiki/X.509) certificates are used to authenticate the server (and sometimes the client as well). As a consequence, [certificate authorities](https://en.wikipedia.org/wiki/Certificate_authority) and [public key certificates](https://en.wikipedia.org/wiki/Public_key_certificate) are necessary to verify the relation between the certificate and its owner, as well as to generate, sign, and administer the validity of certificates. While this can be more beneficial than verifying the identities via a [web of trust](https://en.wikipedia.org/wiki/Web_of_trust), the [2013 mass surveillance disclosures](https://en.wikipedia.org/wiki/2013_mass_surveillance_disclosures) drew attention to certificate authorities as a potential weak point allowing [man-in-the-middle attacks](https://en.wikipedia.org/wiki/Man-in-the-middle_attack).[[21]](https://en.wikipedia.org/wiki/HTTPS#cite_note-21)[[22]](https://en.wikipedia.org/wiki/HTTPS#cite_note-22) An important property in this context is [forward secrecy](https://en.wikipedia.org/wiki/Forward_secrecy), which ensures that encrypted communications recorded in the past cannot be retrieved and decrypted should long-term secret keys or passwords be compromised in the future. Not all web servers provide forward secrecy.[[23]](https://en.wikipedia.org/wiki/HTTPS#cite_note-ecdhe-23)[[*needs update*](https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Dates_and_numbers#Chronological_items)]

For HTTPS to be effective, a site must be completely hosted over HTTPS. If some of the site's contents are loaded over HTTP (scripts or images, for example), or if only a certain page that contains sensitive information, such as a log-in page, is loaded over HTTPS while the rest of the site is loaded over plain HTTP, the user will be vulnerable to attacks and surveillance. Additionally, [cookies](https://en.wikipedia.org/wiki/HTTP_cookie) on a site served through HTTPS must have the [secure attribute](https://en.wikipedia.org/wiki/Secure_cookie) enabled. On a site that has sensitive information on it, the user and the session will get exposed every time that site is accessed with HTTP instead of HTTPS.

**OVERVIEW MEASURES TO MITIGATE**

Attack mitigation is **a detection and protection strategy used to safeguard networks, servers and applications by IT administrators** in order to minimize the effect of malicious traffic and intrusion attempts while maintaining functionality for users.