# Protocols Task - Theoretical

# Question 1

**Name three services provided by SSH.**  
-Secure Remote Logins  
-Secure File Transfer  
-Secure Remote Command Execution  
-Keys and Agents  
-Access Control  
-Port Forwarding

# Question 2

**SSH operates on three layers simultaneously: the transport layer, the user authentication layer, and the connection layer. Explain the function of each layer:&nbsp;  
  
  
Transport layer &nbsp;  
User authentication layer &nbsp;  
Connection layer**  
-Transport layer: It uses the Transmission Control Protocol (TCP) of TCP/IP, reserving port number 22 as a server listening port. This layer handles initial key exchange as well as server authentication, and sets up encryption, compression, and integrity verification.  
  
-User authentication layer: It handles client authentication, and provides a suite of authentication algorithms. Authentication is client-driven: when one is prompted for a password, it may be the SSH client prompting, not the server. The server merely responds to the client's authentication requests.   
  
-Connection layer: It defines the concept of channels, channel requests, and global requests, which define the SSH services provided. A single SSH connection can be multiplexed into multiple logical channels simultaneously, each transferring data bidirectionally.

# Question 3

**What is PuTTy?**  
PuTTy is a text-based user interface that may be used to access remote computers that are running SSH, Telnet, or any other supported protocols. PuTTy is a software terminal emulator that is available for Windows and Linux.

# Question 4

**Enter the differences between HTTP and HTTPS in the textbox below, according to the following table:**

|  |  |  |
| --- | --- | --- |
|  | HTTP | HTTPS |
| Segurity | Not secure | Secure protocol |
| Port number | TCP port 80 | TCP port 443 |
| OSI Layer | Application Layer | Application Layer |
| SSL certificate | No SSL certificate is required. | Signed and implemented by a Certification Authority(CA). |
| Encryption | Has no data encryption. | Data is encrypted right before it is transmitted. |
| Domain validation | Does not always require domain validation. | Requires domain validation. Certain certifications requiring a legal process. |
| Control over data | Plaintext without encryption | Plaintext with encryption |

# Question 5

**What** **makes HTTPS more secure than HTTP? Explain your answer.**  
Encrypt communications between the browser and website.  
HTTPS encrypts the data received by HTTP and ensures that all data exchanged over the Internet between computers and servers is protected by making the data unreadable. It accomplishes this by the use of encryption algorithms.

# Question 6

**Fill in the following table to define the five key differences between HTTP and FTP.**

|  |  |  |
| --- | --- | --- |
|  | HTTP | FTP |
| Purpose | HTTP is utilized to access websites in the network. | FTP transfers files from one host to another. |
| Connection type | Persistent Connections. A client maintains a single connection to the server and uses that for all data transfer. | A new connection must be established for each data transfer. |
| Authentication | HTTP does not expect authentication. | FTP expects a password. |
| Transfer quantities | HTTP is useful in transferring smaller files like web pages. | FTP is useful in transferring larger files. |
| Data Saving | No data storage takes place (e.g browser just retrieves and displays data). | Data transferred (files) are stored on both ends of the connection (client and server). |

# Question 7

**There are nine types of HTTP request methods: GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, TRACE, and PATCH.&nbsp;  
  
Describe and explain four of these methods.**  
GET:It requests a representation of a specified resource, only retrieve data.  
HEAD: It asks for a response identical to that of a GET request, but without a response body.  
POST: It is used to submit an entity to a specified resource, often causing a change in state or side effects on the server.  
PUT: It replaces all current representations of a target resource with the requested payload.  
DELETE: It deletes a specified resource.  
CONNECT: It establishes a tunnel to the server identified by a target resource.  
OPTIONS: It describe the communication options for a target resource.  
TRACE: It performs a message loop-back test along the path to a target resource.  
PATCH: It applies partial modifications to a resource.

**Fill in the following table to show the differences between the GET and POST methods:**

[**https://www.w3schools.com/tags/ref\_httpmethods.asp**](https://www.w3schools.com/tags/ref_httpmethods.asp)

|  |  |  |
| --- | --- | --- |
|  | GET | POST |
| Back button/Reload | Harmless | Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted) |
| Bookmark | Can be bookmarked | Cannot be bookmarked |
| Cached | Can be cached | Not cached |
| Encryption | application/x-www-form-urlencoded | application/x-www-form-urlencoded or multipart/form-data. Use multipart encoding for binary data |
| History | Parameters remain in browser history | Parameters are not saved in browser history |
| Length restriction | Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters) | No restrictions |
| Type restriction | Only ASCII characters allowed | No restrictions. Binary data is also allowed |
| Security | GET is less secure compared to POST because data sent is part of the URL  Never use GET when sending passwords or other sensitive information! | POST is a little safer than GET because the parameters are not stored in browser history or in web server logs |
| Visibility | Data is visible to everyone in the URL | Data is not displayed in the URL |

# Question 8

**1xx**  
Informational - 1XX: This means the request was received, and the process is continuing.

**2xx**  
Success - 2XX: This means the action was successfully received, understood, and accepted.

**3xx**  
Redirection - 3XX: This means further action must be taken to complete the request.

**4xx**  
Client Error - 4XX: This means the request contains incorrect syntax or cannot be fulfilled.

**5xx**  
Server Error - 5XX: This means the server failed to fulfill an apparently valid request.

# Question 9

**What are directory listings?**  
Directory listings are web pages containing lists of files and folders that can be found on a web server. Server directories are relatively easy to look for and usually contain valuable information.  
They can help us discover hidden files and directories and find resources not intended to be publicy accessible. Directories also contain helpful information about the server itself, such as its version, login credentials, and more.  
A directory listing is inappropriately exposed, yielding potentially sensitive information to attackers.  
A directory listing provides an attacker with the complete index of all the resources located inside of the directory. The specific risks and consequences vary depending on which files are listed and accessible.

# Question 10

**How would you execute Google searches for the following information?  
  
  
A directory listing for the Spanish government's website  
An Excel sheet called “phone\_numbers” in a directory listing  
The parent directory of the Wikipedia site**  
-site:gob.es directory listing  
-intitle:index.of phone\_numbers.xls  
- site:wikipedia.org

# Question 11

**When would you, as a cybersecurity professional, use directory listings?**  
In the next tasks:  
  
**Information Gathering:** During the reconnaissance phase of an assessment or investigation, we may utilize directory listings to gather information about the structure, organization, and content of a website or web server. This information can provide insights into the system's architecture and assist in identifying potential vulnerabilities or misconfigurations.  
  
**Vulnerability Assessment:** Directory listings can help identify hidden or sensitive files that are inadvertently exposed on a web server. By reviewing the directory listing, cybersecurity professionals can discover files or directories that were not intended to be publicly accessible, which may contain sensitive information or pose security risks. This can help in conducting comprehensive vulnerability assessments and ensuring proper access controls.  
  
**Incident Response:** In the aftermath of a security incident or breach, cybersecurity professionals may analyze directory listings to identify any unauthorized files or changes made to a system. By comparing the directory listing before and after the incident, we can detect any anomalies, identify malicious activities, and determine the scope of the compromise.  
  
**Secure Configuration Reviews:** Directory listings can assist in verifying the security configurations of web servers or applications. By analyzing the listing, we can check for secure configurations, such as disabling directory browsing, preventing the exposure of sensitive files, or implementing proper access controls.  
  
**Forensic Investigations:** In digital forensic investigations, directory listings can play a crucial role in reconstructing events and understanding the timeline of an incident. Analyzing the timestamps and metadata of files listed in directories can help establish a timeline of file creation, modification, or deletion, aiding in the investigation process.