UNIT 4

WEB SECURITY CONSIDERATIONS

Websites are always to prone to security risks. **Cyber crime** impacts your business by hacking your website. Your website is then used for hacking assaults that install malicious software or malware on your visitor's computer.

Security Considerations

Updated Software

It is mandatory to keep you software updated. It plays vital role in keeping your website secure.

SQL Injection

It is an attempt by the hackers to manipulate your database. It is easy to insert rogue code into your query that can be used to manipulate your database such as change tables, get information or delete data.

Cross Site Scripting (XSS)

It allows the attackers to inject client side script into web pages. Therefore, while creating a form It is good to endure that you check the data being submitted and encode or strip out any HTML.

Error Messages

You need to be careful about how much information to be given in the error messages. For example, if the user fails to

log in the error message should not let the user know which field is incorrect: username or password.

Validation of Data

The validation should be performed on both server side and client side.

Passwords

It is good to enforce password requirements such as of minimum of eight characters, including upper case, lower case and special character. It will help to protect user's information in long run.

Upload files

The file uploaded by the user may contain a script that when executed on the server opens up your website.

SSL

It is good practice to use SSL protocol while passing personal information between website and web server or database.

(OR) We can write as below

What is Web Security?

Web Security is an online security solution that will restrict access to harmful websites, stop web-based risks, and manage staff internet usage. Web Security is very important nowadays. Websites are always prone to security threats/risks. For example- when you are

transferring data between client and server and you have to protect that data that security of data is your web security.

What is a Security Threat?

A threat is nothing but a possible event that can damage and harm an information system. A security Threat is defined as a risk that, can potentially harm Computer systems & organizations. Whenever an individual or an organization creates a website, they are vulnerable to security attacks. Security attacks are mainly aimed at stealing altering or destroying a piece of personal and confidential information, stealing the hard drive space, and illegally accessing passwords. So whenever the website you created is vulnerable to security attacks then the attacks are going to steal your data alter your data destroy your personal information see your confidential information and also it accessing your password.

Top Web Security Threats

- Cross-site scripting (XSS)
- SQL Injection
- Phishing
- Ransomware
- Code Injection
- Viruses and worms
- Spyware
- Denial of Service

Security Consideration

- **Updated Software:** You need to always update your software. Hackers may be aware of vulnerabilities in certain software, which are sometimes caused by bugs and can be used to damage your computer system and

steal personal data. Older versions of software can become a gateway for hackers to enter your network. Software makers soon become aware of these vulnerabilities and will fix vulnerable or exposed areas. That's why It is mandatory to keep your software updated, It plays an important role in keeping your personal data secure.

- **Beware of SQL Injection:** SQL Injection is an attempt to manipulate your data or your database by inserting a rough code into your query. For e.g. somebody can send a query to your website and this query can be a rough code while it gets executed it can be used to manipulate your database such as change tables, modify or delete data or it can retrieve important information also so, one should be aware of the SQL injection attack.
- **Cross-Site Scripting (XSS):** XSS allows the attackers to insert client-side script into web pages. E.g. Submission of forms. It is a term used to describe a class of attacks that allow an attacker to inject client-side scripts into other users' browsers through a website. As the injected code enters the browser from the site, the code is reliable and can do things like sending the user's site authorization cookie to the attacker.
- **Error Messages:** You need to be very careful about error messages which are generated to give the information to the users while users access the website and some error messages are generated due to one or another reason and you should be very careful while providing the information to the users. For e.g. login attempt – If the user fails to login the error message should not let the

user know which field is incorrect: Username or Password.

- **Data Validation:** Data validation is the proper testing of any input supplied by the user or application. It prevents improperly created data from entering the information system. Validation of data should be performed on both server-side and client-side. If we perform data validation on both sides that will give us the authentication. Data validation should occur when data is received from an outside party, especially if the data is from untrusted sources.
- **Password:** Password provides the first line of defense against unauthorized access to your device and personal information. It is necessary to use a strong password. Hackers in many cases use complex software that uses brute force to crack passwords. Passwords must be complex to protect against brute force. It is good to enforce password requirements such as a minimum of eight characters long must including uppercase letters, lowercase letters, special characters, and numerals.

Transport Layer Security | Secure Socket Layer (SSL) and SSL Architecture

Transport Layer Security

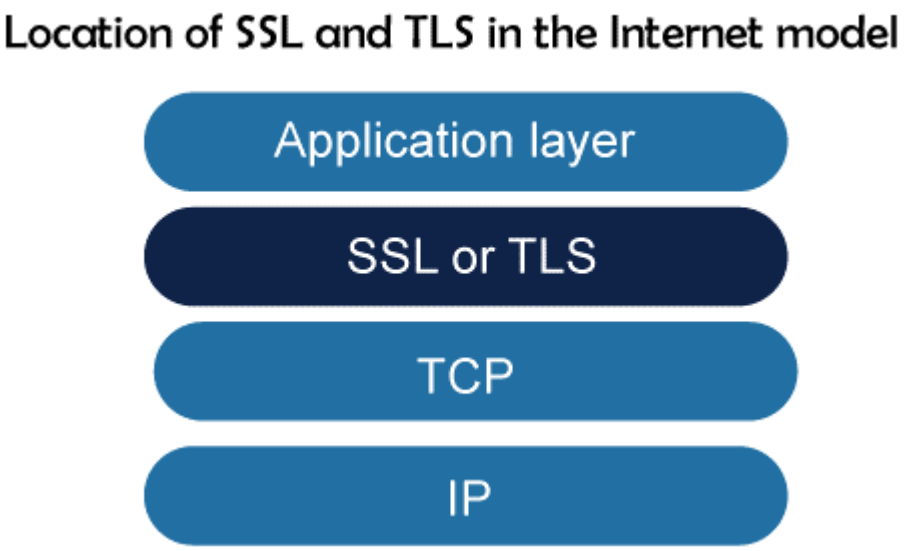
The application layer, which makes use of TCP (or SCTP) as a connection-oriented protocol, is actually secured by the transport layer, which also offers security for that layer. These apps messages are first enclosed in security protocol packets before being contained in TCP.

Due to the nature of security, which necessitates connection formation between the two entities, apps that employ UDP cannot take advantage of these security services.

The transport-layer security does not apply to electronic mail (e-mail), which is another application.

We need a particular security provision for this application because it only allows one-way communication between the sender and the receiver. This is covered in the below section.

The Secure Sockets Layer (SSL) and the Transport Layer Security (TLS) protocols are currently the two most used ones for delivering security at the transport layer.



Services Offered by SSL:

On data that has been received from the application layer, SSL offers a number of services:

- ***Fragmentation*** - The data is first divided into blocks by SSL that are 214 bytes or shorter.
- ***Compression*** - One of the lossless compression techniques agreed upon by the client and server is used to compress every data fragment. This service is not required.
- ***Message Integrity*** - A keyed-hash function is used by SSL to generate a MAC in order to protect the integrity of the data.
- ***Confidentiality*** - Strictly speaking, symmetric-key cryptography is used to encrypt both the original data and the MAC in order to maintain confidentiality.

- ***Framing*** - The encrypted payload receives a header. After that, a dependable transport-layer protocol receives the payload.

TRANSPORT LAYER SECURITY

Transport layer security protocol is one of the security protocols which are designed to facilitate privacy and data security for communications over the Internet. The main use of TLS is to encrypt the communication between web applications and servers, like web browsers loading a website.

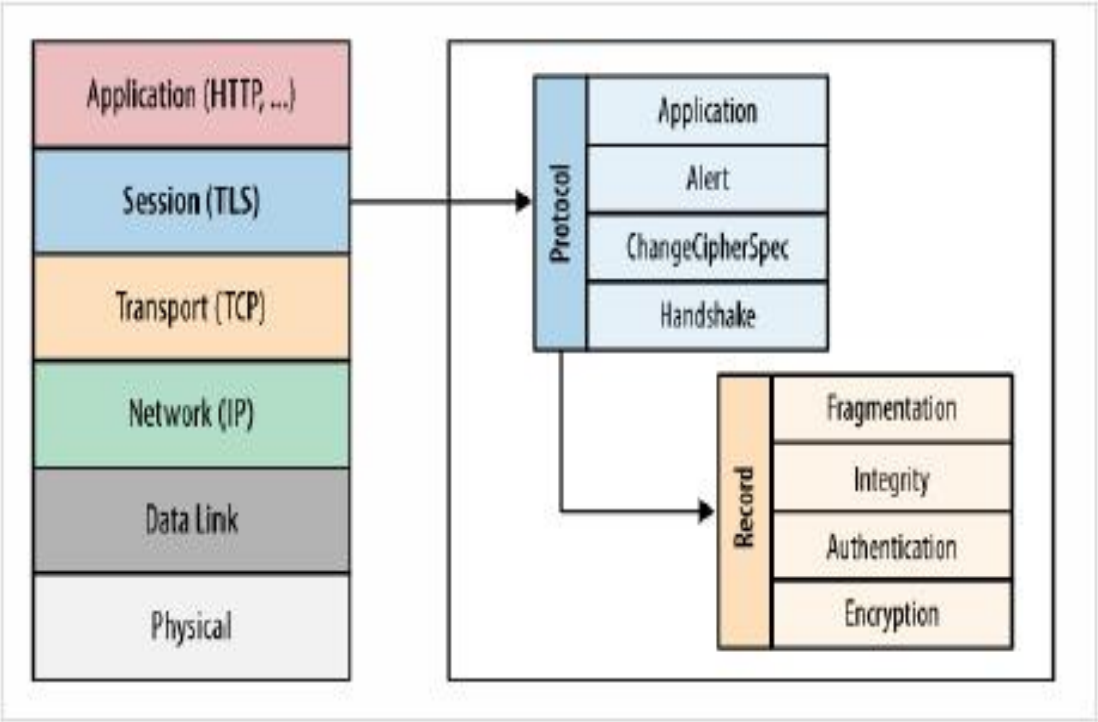
TLS is used to encrypt other communications like email, messaging, and voice over IP (VoIP). TLS was proposed by the Internet Engineering Task Force (IETF), which is an international standards organization.

Components

The three main components that TLS accomplishes are as follows −

- **Encryption** − It is used to hide the data being transferred from third parties.
- **Authentication** − It always ensures that the parties exchanging information are who they claim to be.
- **Integrity** − Integrity verifies that the data has not been tampered with.

Given below is the pictorial representation of the **Transport layer security protocol**



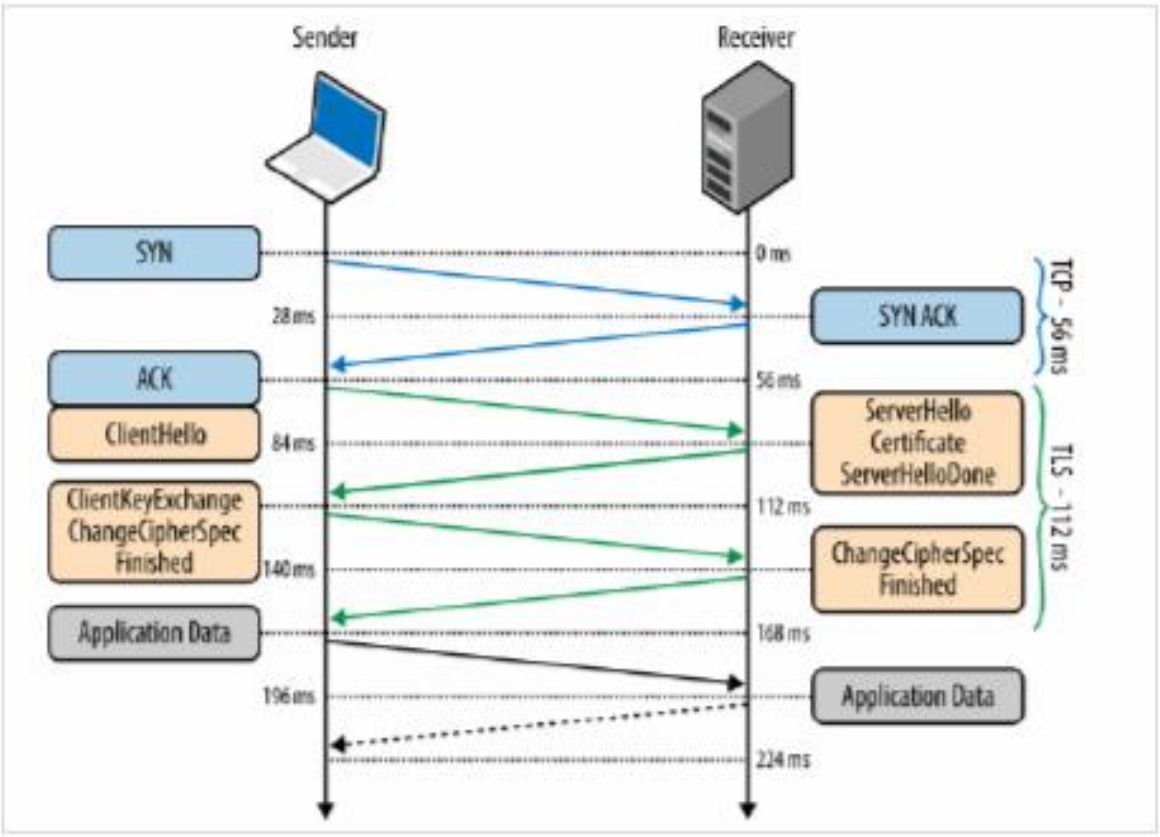
Advantages

The advantages of TLS are as follows−

- Encryption
- Interoperability
- Flexibility
- Easy of deployment
- Easy to use.

TLS handshake Protocol

- The working condition of the TLS Handshake protocol is shown below –



- A client sends a synchronous message “client hello” requesting a connection and presents a list of supported cipher suites and a random string of bytes.
- The server responds with a “server hello” message containing a server certificate.

- The server is sending its SSL certificate to the client for the purpose of authentication. The client then authenticates the server by verifying the server's SSL certificate, and also sends a certificate for authentication if requested by the server.
- The client sends the client key exchange, change Cipher specification finished message to the server.
- The server decrypts the message sent by client secret with the private key.
- Both client and server generate session keys from the client random, the server random, and the secret message.
- The client sends a “finished” message that has been encrypted with a session key.
- The server responds with a finished message which was encrypted with a session key.
- The client and server have successfully achieved secure symmetric encryption, meaning the handshake is complete and communication can continue with the established session keys.
- Finally transfer the application data.

What is HTTPS?

Hypertext transfer protocol secure (HTTPS) is the secure version of HTTP, which is the primary protocol used to send data between a web browser and a website. HTTPS is encrypted in order to increase security of data transfer. This is particularly important when users transmit sensitive data, such as by logging into a bank account, email service, or health insurance provider.

Any website, especially those that require login credentials, should use HTTPS. In modern web browsers such as Chrome, websites that do not use HTTPS are marked differently than those that are. Look for a padlock in the URL bar to signify the webpage is secure.

How does HTTPS work?

HTTPS uses an encryption protocol to encrypt communications. The protocol is called Transport Layer Security (TLS), although formerly it was known as Secure Sockets Layer (SSL). This protocol secures communications by using what's known as an asymmetric public key infrastructure. This type of security system uses two different keys to encrypt communications between two parties:

1. The private key - this key is controlled by the owner of a website and it's kept, as the reader may have speculated, private. This key lives on a web server and is used to decrypt information encrypted by the public key.
2. The public key - this key is available to everyone who wants to interact with the server in a way that's secure.

Information that's encrypted by the public key can only be decrypted by the private key.

Why is HTTPS important? What happens if a website doesn't have HTTPS?

HTTPS prevents websites from having their information broadcast in a way that's easily viewed by anyone snooping on the network.

When information is sent over regular HTTP, the information is broken into packets of data that can be easily "sniffed" using free software.

This makes communication over the an unsecure medium, such as public Wi-Fi, highly vulnerable to interception.

In fact, all communications that occur over HTTP occur in plain text, making them highly accessible to anyone with the correct tools, and vulnerable to on-path attacks.

With HTTPS, traffic is encrypted such that even if the packets are sniffed or otherwise intercepted, they will come across as nonsensical characters. Let's look at an example:

SSH stands for **Secure Shell or Secure Socket Shell**. It is a cryptographic network protocol that allows two computers to communicate and share the data over an insecure network

such as the internet. It is used to login to a remote server to execute commands and data transfer from one machine to another machine.

Secure Shell:

The SSH protocol was developed by **SSH communication security Ltd** to safely communicate with the remote machine.

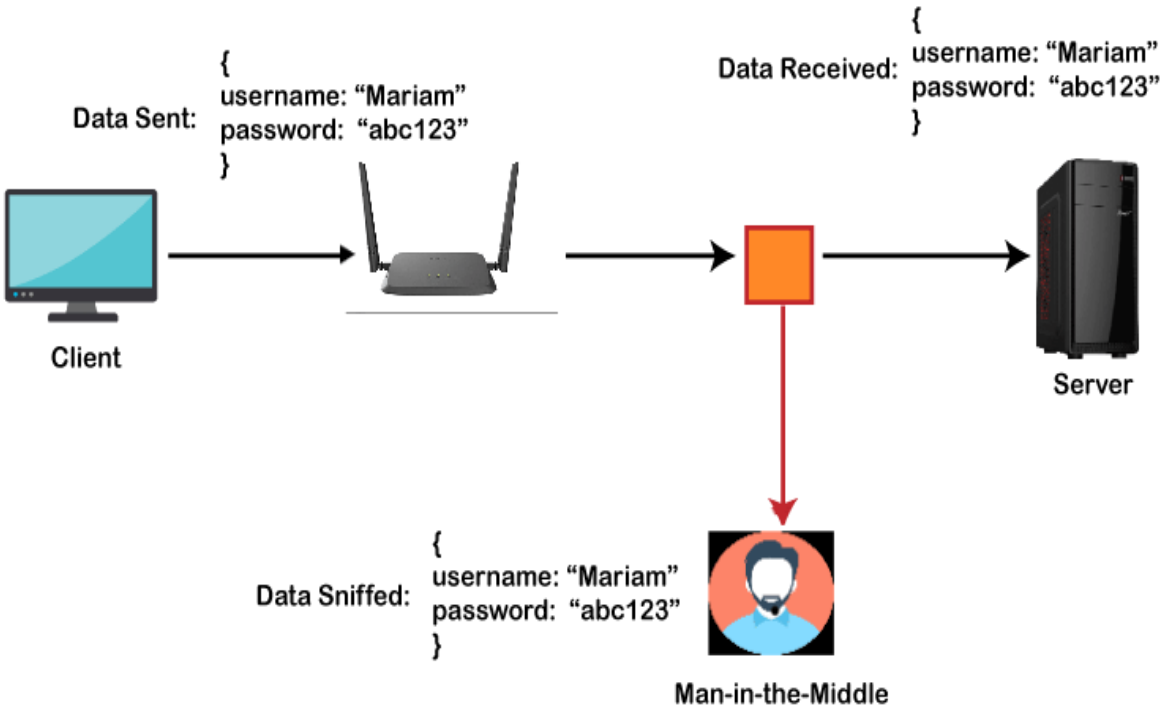
Secure communication provides a strong password authentication and encrypted communication with a public key over an insecure channel. It is used to replace unprotected remote login protocols such as **Telnet, rlogin, rsh, etc**., and insecure file transfer protocol **FTP**.

Its security features are widely used by network administrators for managing systems and applications remotely.

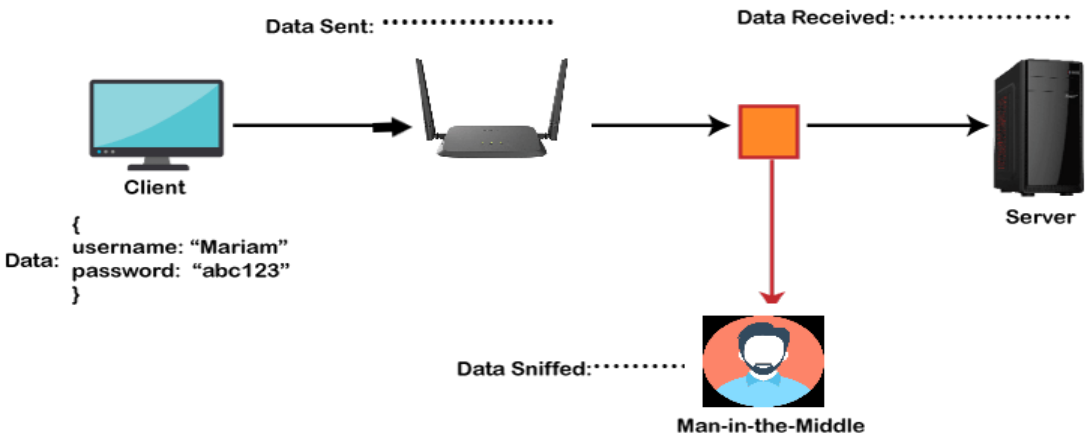
The SSH protocol protects the network from various attacks such as **DNS spoofing, IP source routing, and IP spoofing.**

A simple example can be understood, such as suppose you want to transfer a package to one of your friends. Without SSH protocol, it can be opened and read by anyone. But if you will send it using SSH protocol, it will be encrypted and secured with the public keys, and only the receiver can open it.

Before SSH:


Data Sent:
{
username: “Mariam”
password: “abc123”
}
Data Received:
{
username: “Mariam”
password: “abc123”
}
Client
Server
Data Sniffed:
{
username: “Mariam”
password: “abc123”
}
Man-in-the-Middle

After SSH:


Data Sent: ••••••••••••••••••
Data Received: ••••••••••••••••••
Client
Server
Data:
{
username: “Mariam”
password: “abc123”
}
Data Sniffed: ••••••••••
Man-in-the-Middle