**VISION INSTITUTE OF MANAGEMENT**

**COMPUTER NETWORK SECURITY**

**BCA 3rd YEAR/6th SEM**

**UNIT 3(Web Security)**

# What is Web Security?

Web security is also known as “Cybersecurity”. It basically means protecting a website or web application by detecting, preventing and responding to cyber threats.

Websites and web applications are just as prone to security breaches as physical homes, stores, and government locations. Unfortunately, cybercrime happens every day, and great web security measures are needed to protect websites and web applications from becoming compromised.

That’s exactly what web security does – it is a system of protection measures and protocols that can protect your website or web application from being hacked or entered by unauthorized personnel. This integral division of Information Security is vital to the protection of websites, web applications, and web services. Anything that is applied over the Internet should have some form of web security to protect it.

### **Details of Web Security**

There are a lot of factors that go into web security and web protection. Any website or application that is secure is surely backed by different types of checkpoints and techniques for keeping it safe.

There are a variety of security standards that must be followed at all times, and these standards are implemented and highlighted by the OWASP. Most experienced web developers from [**top cybersecurity companies**](https://www.goodfirms.co/it-services/cybersecurity-companies) will follow the standards of the OWASP as well as keep a close eye on the Web Hacking Incident Database to see when, how, and why different people are hacking different websites and services.

Essential steps in protecting web apps from attacks include applying up-to-date encryption, setting proper authentication, continuously patching discovered vulnerabilities, avoiding data theft by having secure software development practices. The reality is that clever attackers may be competent enough to find flaws even in a fairly robust secured environment, and so a holistic security strategy is advised.

### **Available Technology**

There are different types of technologies available for maintaining the best security standards. Some popular technical solutions for testing, building, and preventing threats include:

* Black box testing tools
* Fuzzing tools
* White box testing tools
* Web application firewalls (WAF)
* Security or vulnerability scanners
* Password cracking tools

### **The Best Strategies**

There are two big strategies that a developer can use to protect their website or web application. The two main methods are as follows:

* **Resource assignment** – By assigning all necessary resources to causes that are dedicated to alerting the developer about new web security issues and threats, the developer can receive a constant and updated alert system that will help them detect and eradicate any threats before security is officially breached.
* **Web scanning** – There are several web scanning solutions already in existence that are available for purchase or download. These solutions, however, are only good for known vulnerability threats – seeking unknown threats can be much more complicated. This method can protect against many breaches, however, and is proven to keep websites safe in the long run.

**Web Security also protects the visitors from the below-mentioned points –**

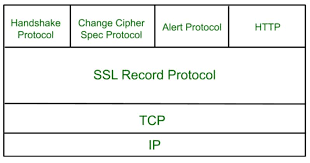
* **Stolen Data:**Cyber-criminals frequently hacks visitor’s data that is stored on a website like email addresses, payment information, and a few other details.
* **Phishing schemes:**This is not just related to email, but through phishing, hackers design a layout that looks exactly like the website to trick the user by compelling them to give their sensitive details.
* **Session hijacking:**Certain cyber attackers can take over a user’s session and compel them to take undesired actions on a site.
* **Malicious redirects:** Sometimes the attacks can redirect visitors from the site they visited to a malicious website.
* **SEO Spam:**Unusual links, pages, and comments can be displayed on a site by the hackers to distract your visitors and drive traffic to malicious websites.

Thus, web security is easy to install and it also helps the business people to make their website safe and secure. A web application firewall prevents automated attacks that usually target small or lesser-known websites. These attacks are borne out by malicious bots or malware that automatically scan for vulnerabilities they can misuse, or cause DDoS attacks that slow down or crash your website.

Thus, Web security is extremely important, especially for websites or web applications that deal with confidential, private, or protected information. Security methods are evolving to match the different types of vulnerabilities that come into existence.

# Secure Socket Layer (SSL)

[**Secure Socket Layer (SSL)**](https://practice.geeksforgeeks.org/problems/what-is-ssl) provide security to the data that is transferred between web browser and server. SSL encrypt the link between a web server and a browser which ensures that all data passed between them remain private and free from attack.



**Secure Socket Layer Protocols:**

* SSL record protocol
* Handshake protocol
* Change-cipher spec protocol
* Alert protocol

**SSL Record Protocol:**

SSL Record provide two services to SSL connection-

* Confidentiality
* Message Integrity

In SSL Record Protocol application data is divided into fragments. The fragment is compressed and then encrypted MAC (Message Authentication Code) generated by algorithms like SHA (Secure Hash Protocol) and MD5 (Message Digest) is appended. After that encryption of the data is done and in last SSL header is appended to the data.

**Handshake Protocol:**

Handshake Protocol is used to establish sessions. This protocol allow client and server to authenticate each other by sending a series of messages to each other. Handshake protocol uses four phases to complete its cycle.

* **Phase-1:** In Phase-1 both Client and Server send hello-packets to each other. In this IP session, cipher suite and protocol version are exchanged for security purpose.
* **Phase-2:** Server send his certificate and Server-key-exchange. Server end the phase-2 by sending Server-hello-end packet.
* **Phase-3:** In this phase Client reply to the server by sending his certificate and Client-exchange-key.
* **Phase-4:** In Phase-4 Change-cipher suite occurred and after this Handshake Protocol ends.

**Change-cipher Protocol:**

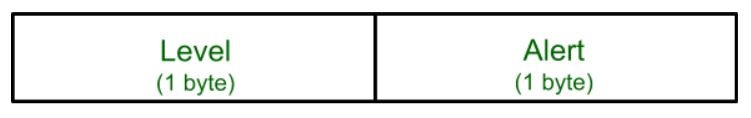
This protocol uses SSL record protocol. Unless Handshake Protocol is completed, the SSL record Output will be in pending state. After handshake protocol the Pending state is converted into Current state.

Change-cipher protocol consists of single message which is 1 byte in length and can have only one value. This protocol purpose is to cause the pending state to be copied into current state



**Alert Protocol:**

This protocol is used to convey SSL-related alerts to the peer entity. Each message in this protocol contain 2 bytes.



Level is further classified into two parts:

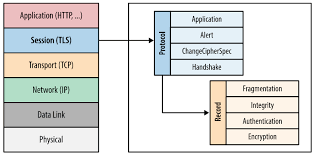
* **Warning:**  
  This Alert have no impact on the connection between sender and receiver.
* **Fatal Error:**  
  This Alert breaks the connection between sender and receiver.

**Silent Features of Secure Socket Layer:**

* Advantage of this approach is that the service can be tailored to the specific needs of the given application.
* Secure Socket Layer was originated by Netscape.
* SSL is designed to make use of TCP to provide reliable end-to-end secure service.
* This is two-layered protocol.

# Transport Layer Security (TLS)

Transport Layer Securities (TLS) are designed to provide security at the transport layer. TLS was derived from a security protocol called [Secure Service Layer (SSL)](https://www.geeksforgeeks.org/secure-socket-layer-ssl/). TLS ensures that no third party may eavesdrop or tamper with any message.



There are several benefits of TLS:

* **Encryption:**  
  TLS/SSL can help to secure transmitted data using encryption.
* **Interoperability:**  
  TLS/SSL works with most web browsers, including Microsoft Internet Explorer and on most operating systems and web servers.
* **Algorithm flexibility:**  
  TLS/SSL provides operations for authentication mechanism, encryption algorithms and hashing algorithm that are used during the secure session.
* **Ease of Deployment:**  
  Many applications TLS/SSL temporarily on a windows server 2003 operating systems.
* **Ease of Use:**  
  Because we implement TLS/SSL beneath the application layer, most of its operations are completely invisible to client.

**Working of TLS:**

The client connect to server (using [TCP](https://www.geeksforgeeks.org/tcp-ip-model/)), the client will be something. The client sends number of specification:

1. Version of SSL/TLS.
2. Which cipher suites, compression method it wants to use.

The server checks what the highest SSL/TLS version is that is supported by them both, picks a cipher suite from one of the clients option (if it supports one) and optionally picks a compression method. After this the basic setup is done, the server provides its certificate. This certificate must be trusted either by the client itself or a party that the client trusts. Having verified the certificate and being certain this server really is who he claims to be (and not a man in the middle), a key is exchanged. This can be a public key, “Premaster Secret” or simply nothing depending upon cipher suite.

Both the server and client can now compute the key for symmetric encryption. The handshake is finished and the two hosts can communicate securely. To close a connection by finishing. TCP connection both sides will know the connection was improperly terminated. The connection cannot be compromised by this through, merely interrupted

# Secure Electronic Transaction (SET)

Secure Electronic Transaction (SET) is a system for ensuring the security of financial transactions on the Internet. It was supported initially by MasterCard, Visa, Microsoft, Netscape, and others. With SET, a user is given an *electronic wallet* ([digital certificate](https://searchsecurity.techtarget.com/definition/digital-certificate)) and a transaction is conducted and verified using a combination of digital certificates and [digital signature](https://searchsecurity.techtarget.com/definition/digital-signature)s among the purchaser, a merchant, and the purchaser's bank in a way that ensures privacy and confidentiality. SET makes use of Netscape's Secure Sockets Layer ([SSL](https://searchsecurity.techtarget.com/definition/Secure-Sockets-Layer-SSL)), Microsoft's Secure Transaction Technology (STT), and Teresa System's Secure Hypertext Transfer Protocol ([S-HTTP](https://searchsoftwarequality.techtarget.com/definition/S-HTTP)). SET uses some but not all aspects of a public key infrastructure ([PKI](https://searchsecurity.techtarget.com/definition/PKI)).

Here's how SET works:

Assume that a customer has a SET-enabled browser such as Netscape or Microsoft's Internet Explorer and that the transaction provider (bank, store, etc.) has a SET-enabled server.

1. The customer opens a MasterCard or Visa bank account. Any issuer of a credit card is some kind of bank.
2. The customer receives a [digital certificate](https://searchsecurity.techtarget.com/definition/digital-certificate). This electronic file functions as a credit card for online purchases or other transactions. It includes a [public key](https://searchsecurity.techtarget.com/definition/public-key) with an expiration date. It has been through a [digital switch](https://searchnetworking.techtarget.com/definition/digital-switch) to the bank to ensure its validity.
3. Third-party merchants also receive certificates from the bank. These certificates include the merchant's public key and the bank's public key.
4. The customer places an order over a Web page, by phone, or some other means.
5. The customer's browser receives and confirms from the merchant's certificate that the merchant is valid.
6. The browser sends the order information. This message is encrypted with the merchant's public key, the payment information, which is encrypted with the bank's public key (which can't be read by the merchant), and information that ensures the payment can only be used with this particular order.
7. The merchant verifies the customer by checking the digital signature on the customer's certificate. This may be done by referring the certificate to the bank or to a third-party verifier.
8. The merchant sends the order message along to the bank. This includes the bank's public key, the customer's payment information (which the merchant can't decode), and the merchant's certificate.
9. The bank verifies the merchant and the message. The bank uses the digital signature on the certificate with the message and verifies the payment part of the message.
10. The bank digitally signs and sends authorization to the merchant, who can then fill the order.