Module-2

System Security

Web Security

- In general, web security refers to the protective measures and protocols that organizations adopt to protect the organization from, cyber criminals and threats that use the web channel.
- Web security is critical to business continuity and to protecting data, users and companies from risk

Web Security

- Website security requires vigilance in all aspects of website design and usage.
- The Internet is a dangerous place! With great regularity, we hear about websites becoming unavailable due to denial of service attacks, or displaying modified (and often damaging) information on their homepages.
- In other high-profile cases, millions of passwords, email addresses, and credit card details have been leaked into the public domain, exposing website users to both personal embarrassment and financial risk.

Web Security

- The purpose of website security is to prevent these (or any) sorts of attacks.
- The more formal definition of website security is the act/practice of protecting websites from unauthorized access, use, modification, destruction, or disruption.

Web Security

- Effective website security requires design effort across the whole of the website:
  - in your web application, the configuration of the web server, your policies for creating and renewing passwords, and the client-side code.
  - While all that sounds very ominous, the good news is that if you're using a server-side web framework, it will almost certainly enable "by default" robust and well-thought-out defense mechanisms against a number of the more common attacks.

Web Security

- Effective website security requires design effort across the whole of the website:
  - Finally, there are publicly available vulnerability scanner tools that can help you find out if you've made any obvious mistakes.

Web security Threats

1. Cross-Site Scripting (XSS)
2. SQL injection
3. Cross-Site Request Forgery (CSRF)
4. Other threats

Web security Threats

1. Cross-Site Scripting (XSS)
   - XSS is a term used to describe a class of attacks that allow an attacker to inject client-side scripts *through* the website into the browsers of other users.
   - Because the injected code comes to the browser from the site, the code is *trusted* and can do things like send the user's site authorization cookie to the attacker.
   - When the attacker has the cookie, they can log into a site as though they were the user and do anything the user can, such as access their credit card details, see contact details, or change passwords.

Web security Threats

2. SQL injection
   - SQL injection vulnerabilities enable malicious users to execute arbitrary SQL code on a database, allowing data to be accessed, modified, or deleted irrespective of the user's permissions.
   - A successful injection attack might spoof identities, create new identities with administration rights, access all data on the server, or destroy/modify the data to make it unusable.

Web security Threats

3. Cross-Site Request Forgery (CSRF)
   - CSRF attacks allow a malicious user to execute actions using the credentials of another user without that user's knowledge or consent.
   - For Example, John is a malicious user who knows that a particular site allows logged-in users to send money to a specified account using an HTTP POST request that includes the account name and an amount of money.
   - John constructs a form that includes his bank details and an amount of money as hidden fields, and emails it to other site users (with the *Submit* button disguised as a link to a "get rich quick" site).

Web security Threats

3. Cross-Site Request Forgery (CSRF)
   - If a user clicks the submit button, an HTTP POST request will be sent to the server containing the transaction details and any client-side cookies that the browser associated with the site (adding associated site cookies to requests is normal browser behavior).
   - The server will check the cookies, and use them to determine whether or not the user is logged in and has permission to make the transaction.
   - The result is that any user who clicks the *Submit* button while they are logged in to the trading site will make the transaction. John gets rich.

Web security Threats

4. Other threats
   - Other common attacks/vulnerabilities include:
     a) Click jacking
     b) Denial of Service
     c) Directory Traversal
     d) File Inclusion
     e) Command Injection

Web security Threats

4. Other threats

- Other common attacks/vulnerabilities include:

a) Click jacking

- In this attack, a malicious user hijacks clicks meant for a visible top-level site and routes them to a hidden page beneath.
- This technique might be used, for example, to display a legitimate bank site but capture the login credentials into an invisible <iframe> controlled by the attacker. Clickjacking could also be used to get the user to click a button on a visible site, but in doing so actually unwittingly click a completely different button.
- As a defense, your site can prevent itself from being embedded in an iframe in another site by setting the appropriate HTTP headers.

Web security Threats

4. Other threats

- Other common attacks/vulnerabilities include:

  b) Denial of Service

  - DoS is usually achieved by flooding a target site with fake requests so that access to a site is disrupted for legitimate users.
  - The requests may be numerous, or they may individually consume large amounts of resource (e.g., slow reads or uploading of large files).

Web security Threats

4. Other threats

- Other common attacks/vulnerabilities include:

  c) Directory Traversal

  - In this attack, a malicious user attempts to access parts of the web server file system that they should not be able to access.
  - This vulnerability occurs when the user is able to pass filenames that include file system navigation characters (for example, ../../). The solution is to sanitize input before using it.

Web security Threats

4. Other threats

- Other common attacks/vulnerabilities include:

  d) File Inclusion

  - In this attack, a user is able to specify an "unintended" file for display or execution in data passed to the server.
  - When loaded, this file might be executed on the web server or the client-side (leading to an XSS attack).
  - The solution is to sanitize input before using it.

Web security Threats

4. Other threats

- Other common attacks/vulnerabilities include:

  e) Command Injection

  - Command injection attacks allow a malicious user to execute arbitrary system commands on the host operating system.
  - The solution is to sanitize user input before it might be used in system calls.

Application Security

- **Application security** describes security measures at the application level that aim to prevent data or code within the app from being stolen or hijacked.
- It encompasses the security considerations that happen during application development and design, but it also involves systems and approaches to protect apps after they get deployed.

Application Security

- Application security may include hardware, software, and procedures that identify or minimize security vulnerabilities.
- A router that prevents anyone from viewing a computer's IP address from the Internet is a form of hardware application security.
- But security measures at the application level are also typically built into the software, such as an application firewall that strictly defines what activities are allowed and prohibited.
- Procedures can entail things like an application security routine that includes protocols such as regular testing.

Application Security

- Application security is the process of developing, adding, and testing security features within applications to prevent security vulnerabilities against threats such as unauthorized access and modification.

Application Security

- **Why application security is important?**
  - Application security is important because today's applications are often available over various networks and connected to the cloud, increasing vulnerabilities to security threats and breaches.
  - There is increasing pressure and incentive to not only ensure security at the network level but also within applications themselves.
  - One reason for this is because hackers are going after apps with their attacks more today than in the past.
  - Application security testing can reveal weaknesses at the application level, helping to prevent these attacks.

Application Security

- **Types of application security**
  - Different types of application security features include authentication, authorization, encryption, logging, and application security testing
    1. Authentication
    2. Authorization
    3. Encryption
    4. Logging
    5. Application security testing

Application Security

- **Types of application security**
  1. Authentication
     - When software developers build procedures into an application to ensure that only authorized users gain access to it.
     - Authentication procedures ensure that a user is who they say they are. This can be accomplished by requiring the user to provide a user name and password when logging in to an application.
     - Multi-factor authentication requires more than one form of authentication—the factors might include something you know (a password), something you have (a mobile device), and something you are (a

Application Security

- **Types of application security**
  2. Authorization
     - After a user has been authenticated, the user may be authorized to access and use the application.
     - The system can validate that a user has permission to access the application by comparing the user's identity with a list of authorized users.
     - Authentication must happen before authorization so that the application matches only validated user credentials to the authorized user list.

Application Security

- **Types of application security**

  3. Encryption
     - After a user has been authenticated and is using the application, other security measures can protect sensitive data from being seen or even used by a cybercriminal.
     - In cloud-based applications, where traffic containing sensitive data travels between the end user and the cloud, that traffic can be encrypted to keep the data safe.

Application Security

- **Types of application security**

4. Logging
    - If there is a security breach in an application, logging can help identify who got access to the data and how.
    - Application log files provide a time-stamped record of which aspects of the application were accessed and by whom.
5. Application security testing
    - A necessary process to ensure that all of these security controls work properly.