Summarize different SDLC stages in detail.

=>

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process. Different SDLC stages are described as follows -

1) Stage 1: Planning and Requirement Analysis

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage.

2) Stage 2: Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

3) Stage 3: Designing the Product Architecture

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

4) Stage 4: Building or Developing the Product

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

5) Stage 5: Testing the Product

   This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

6) Stage 6: Testing the Product

   This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Summarize OWASP Top 10 vulnerabilities.

=>

The Top 10 OWASP Vulnerabilities:
1. Injection
2. Cross-Site Scripting (XSS)
3. Broken Authentication and Session Management
4. Insecure Direct Object References
5. Cross-Site Request Forgery
6. Security Misconfiguration
7. Insecure Cryptographic Storage
8. Failure to restrict URL Access
9. Insufficient Transport Layer Protection
10. Un-validated Redirects and Forwards

1-Injection

The injection method sends invalid data/request by the attacker to the web application having intension to do something that the application was not designed to perform.

In this way, the attacker may be able to insert code that the application will then execute.

The intension is to enter the database commands as input, and the application converts it into a query that includes the attacker’s command and sends it to the database. When it gets executed, it provides an attacker with the desired information.

2- Cross-Site Scripting (XSS)

• It is a type of injection vulnerability in which the attacker uses his script code (JavaScript or HTML) and inserts it into a vulnerable web page.

3- Broken Authentication and Session Management

The broken authentication vulnerability can be minimized by avoiding to leave the login pages to access by default or non legitimate users.

4- Insecure Direct Object References

•Insecure Direct Object References (IDOR) occurs when an application provides direct access to the object based on the user-supplied input.

•By exploiting this vulnerability, an attacker can bypass authorization and access resources in the system directly. For example, database record files.

5- Cross-Site Request Forgery

•Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they’re currently authenticated.

•Cross-site request forgery is an example of a confused deputy attack against a web browser because the web browser is tricked into submitting a forged request by a less privileged attacker.

6- Security Misconfiguration

•Security misconfiguration is a web application vulnerability that exists into the application if the developer fails to implement all the security controls for the server/application, or implement with errors.

7- Insecure Cryptographic Storage

•Insecure cryptographic vulnerability comes into existence if we store sensitive information like password unencrypted on the server.

•It is not a good practice to store the password on the server, though if required, better to store a one-way cryptographic hash of a user’s password rather than the password itself.

8- Failure to restrict URL Access

•Failure to restrict URL access is a web application vulnerability that lets an attacker access pages that are meant to be restricted or hidden.

•Failure to restrict URL access occurs when an
error in access control setting results in users
being able to access pages that are meant to be
restricted or hidden.

9- Insufficient Transport Layer Protection

•A very common weakness in a web application is Insufficient Transport Layer Protection. It is
caused by not taking any measures to protect network traffic. If the data transfer between client
and server is unencrypted, the session IDs can be intercepted and re-created to make the
application vulnerable to exploitation.

10- Unvalidated Redirects and Forwards

•When an application accepts untrusted inputs, the unvalidated redirects and forward are possible.
It is a vulnerability of a web application that may redirect a request to a URL contained within
the untrusted input.
•By modifying untrusted URL input to a malicious site, an attacker may successfully launch a
phishing scam and steal user credentials.

Develop a list of different types of Cross Site Scripting (XSS).

=>

Cross Site Scripting (XSS) is a code injection attack executed on the client side of a web application.

1) Attacker injects malicious script through the web browser.
2) The malicious script is executed when the victim visits the web page or web server.
3) Steals cookies, session tokens and other sensitive information.
4) Modify the contents of the website.

Types:

1) Reflected XSS (Non-persistent)
    a) Script is executed on the victim side
    b) Script is not stored on the server

       Eg.

       As input:

       <h1>Hello </h1>

```
<script>alert(“Hi”);</script>
<script>alert(“XSS”)</script>
<script>alert(document.cookie)</script>
<scr<script>ipt>alert(“XSS”)</script>
<img src=X onMouseOver=alert(“XSS”)/>
```

2) Stored XSS (Persistent)
   a) Script is stored and executed on the server
   b) Executed every time the malicious site is requested.
   c) After refresh the same malicious code is executed again because data is stored on the database.

   Eg. <a href="hacked.php">Congratulation!!! Click here to win 5 lac of rupees.</a>

   Congratulation!!! Click here to win 5 lac of rupees.
   <script src=hacked.js></script>

   <img src=”X” alt=”Click Here” onMouseOver="mess()">

3) DOM (Document Object Model) XSS
   a) Client-side attack. Script is not sent to the server.
   b) Legitimate server script is executed followed by malicious script.
   c) It is injected in the URL (1st server code is executed and then injection code is executed)

Develop a list of different types of Malwares in detail.

=>

Malware, short for malicious software, is any software used to disrupt computer or mobile operations, gather sensitive information, gain access to private computer systems, or display unwanted advertising.
Several types of malicious code or malware exist, such as viruses, worms, Trojan horses and logic bombs.

1) Viruses
   - A virus is a small application, or string of code, that infects software.
   - The main function of a virus is to reproduce and deliver its payload, and it requires a host application to do this.
   - In other words, viruses cannot replicate on their own.
   - A virus infects a file by inserting or attaching a copy of itself to the file.

• The virus is just the “delivery mechanism.”
• It can have any type of payload (deleting system files, displaying specific messages, reconfiguring systems, stealing sensitive data, installing a sniffer or back door).

2) Worms

• A computer worm is a standalone malware computer program that replicates itself in order to spread to other computers.
• Often, it uses a computer network to spread itself, relying on security failures on the target computer to access it.
• Worms always cause at least some harm to the network, even if only by consuming bandwidth, whereas viruses always corrupt or modify files on a targeted computer.
• Many worms that have been created are designed only to spread, and do not attempt to change the systems they pass through.
• However, as the Morris worm and My doom showed, even these "payload free" worms can cause major disruption by increasing network traffic and other unintended effects.

3) Trojan horses

• A Trojan horse is a program that is disguised as another program.
•For example, a Trojan horse can be named Notepad.exe and have the same icon as the regular Notepad program.
• However, when a user executes Notepad.exe, the program can delete system files.
• Trojan horses perform a useful functionality in addition to the malicious functionality in the background.
• So, the Trojan horse named Notepad.exe may still run the Notepad program for the user, but in the background, it will manipulate files or cause other malicious acts.

4) Logic Bombs

- A logic bomb executes a program, or string of code, when a certain set of conditions are met.
- The logic bomb software can have many types of triggers that activate its payload execution, as in time and date or after a user carries out a specific action.