Most Common Vulnerability checks

OWASP TOP 10 | SANS TOP 25

Presented By :
Amrit Choudhary

OWASP TOP 10

The methodology uses a combination of data-driven analysis and industry surveys to establish a list of the ten most significant application security vulnerabilities

1. Broken Access Control

Access controls are critical for securing applications against unauthorized access to data and resources.Broken access controls can lead to data compromise, obtaining permissions beyond what's intended for standard users, or account takeover attacks where outsiders hijack user accounts and initiate fraudulent transactions.

Access controls are harder to implement later, so communicate the importance of implementing proper access controls, such as denying requests by default and rate limiting APIs early on in web app development.

OWASP TOP 10

2. Cryptographic Failures : Cryptographic failures refer to either a bad implementation of encryption or a complete lack of encryption Eg.

Key compromise: If the key used to encrypt or decrypt data is compromised, the security of the data may be compromised as well.

Weak or flawed algorithms: If the cryptographic algorithm being used is weak or has known flaws, it may be vulnerable to attack.

3. Injection : Injection is a risk category that refers to the ability of threat actors to provide malicious input to web applications that result in the app executing unexpected and unwanted commands

Eg. SQL injection: This occurs when an attacker is able to inject malicious SQL code into a system through the input of data, allowing them to gain access to or manipulate sensitive data stored in a database.

To prevent injection vulnerabilities, it is important to properly validate and sanitize all user input

OWASP TOP 10

4. Insecure Design :

Insecure design refers to the design of a system or product that does not adequately consider security risks and vulnerabilities. This can lead to vulnerabilities being present in the system or product, which can be exploited by attackers to gain unauthorized access, steal sensitive information, or execute other malicious actions.

To prevent insecure design, it is important to consider security risks and vulnerabilities during the design process and to test systems and products thoroughly for vulnerabilities before they are released.

5. Security Misconfiguration :

Misconfigurations are increasingly common due to the cloud being used as a development environment and web apps being built with container images.

A pivotal strategic change is to ensure you have a repeatable process for hardening configurations and a tool or process that automatically audits and verifies those configurations across on-premise and cloud environments.

6. Vulnerable and Outdated Components :

When threat actors try to compromise an application, they look at its component parts and attempt to exploit any vulnerabilities. Often, these vulnerabilities come from using out-of-date frameworks or libraries that are easy to exploit.

The overall strategic mitigation here is to ensure an effective patch management strategy is in place.

OWASP TOP 10

7. Identification and Authentication Failures: Failures in authentication and identity management make applications vulnerable to threat actors masquerading as legitimate users. Some examples of vulnerabilities include not setting validity periods for session IDs, permitting weak passwords that are easy to guess, and not rate limiting login attempts against automated attacks.

The solutions include implementing multi-factor authentication in apps and communicating the importance of complying with recommended password length, complexity, and rotation policies to developers.

8. Software and Data Integrity Failures : making faulty default assumptions within development pipelines about the integrity of software or data. Since web apps regularly rely on plugins and libraries from external sources, a lack of verification of the integrity of these sources introduces the risk of malicious code, unauthorized access, and compromise.

The main mitigation strategy is ensuring external code or data hasn't been tampered with by requiring digital signatures.

9. Security Logging and Monitoring Failures: Logging and monitoring help to provide security accountability, visibility into events, incident alerting, and forensics.

To mitigate implement monitoring and alerting, and create an incident recovery and response strategy

OWASP TOP 10

10. Server-Side Request Forgery (SSRF) :

Most web apps today require external resources for their functionality, which are usually accessed at URLs. SSRF occurs when hackers can get servers to make requests that they control. The typical vulnerability is that the web application doesn't validate the user-supplied URL, potentially allowing access to internal services or resources by bypassing access controls.

To prevent SSRF vulnerabilities, it is important to properly validate and sanitize user input, and to limit the types of requests that can be made through the web application. It is also important to implement strong access controls and to monitor for unusual activity on the server.

SANS TOP 25

SANS top 25 most dangerous software flaws is a list of the most dangerous flaws because they let attackers gain entire control of the software, steal data and information from it, or prohibit it from functioning at all.

The SANS top 25 is a versatile starting point that can be used by almost any organization, regardless of size, industry, geography or government/ commercial status.

The Common Vulnerabilities and Exposures Team generated the list using publicly available data, CWE mappings from the National Vulnerability Database (NVD), and CVSS scores for each CWE. A scoring algorithm was then used to determine the severity of each fault.

List Of SANS Top 25

1. Out-of-bounds Write
2. Improper Neutralization of Input During Web Page Generation (‘Cross-site Scripting’)
3. Out-of-bounds Read
4. Improper Input Validation
5. Improper Neutralization of Special Elements used in an OS Command (‘OS Command Injection’)

SANS TOP 25

6. Improper Neutralization of Special Elements used in an SQL Command (‘SQL Injection’)

7. Use After Free

8. Improper Limitation of a Path name to a Restricted Directory (‘Path Traversal’)

9. Cross-Site Request Forgery (CSRF)

10. Unrestricted Upload of File with Dangerous Type

11. Missing Authentication for Critical Function

12. Integer Overflow or Wraparound

13. Deserialization of Untrusted Data

14. Improper Authentication

15. NULL Pointer Dereference

16. Use of Hard-coded Credentials

17. Improper Restriction of Operations within the Bounds of a Memory Buffer

SANS TOP 25

18. Missing Authorization

19. Incorrect Default Permissions

20. Exposure of Sensitive Information to an Unauthorized Actor

21. Insufficiently Protected Credentials

22. Incorrect Permission Assignment for Critical Resource

23. Improper Restriction of XML External Entity Reference

24. Server-Side Request Forgery (SSRF)

25. Improper Neutralization of Special Elements used in a Command (‘Command Injection’)

REFERENCES :

https://owasp.org/www-project-top-ten/

https://www.sans.org/top25-software-errors/