“Critical Web Vulnerabilities”....

Report on OWASP TOP 10(2017-2021)

OWASP

“The Ten Most Critical Web Application Security Risks“

“GLOBALLY RECOGNIZED BY DEVELOPERS AS
THE FIRST STEP TOWARDS MORE SECURE CODING”!!

BY:RUQUIYA FATIMA

Table of Contents

INTRODUCTION:

OWASP(the open world wide application security project) is nothing but a nonprofit foundation dedicated to improving software security.It is an online community that produces free application resources for web application security.It will help our organisations to mitigate risk.It is an open community model which means that anyone can practise in and contribute to OWASP related online chats projects and more.The OWASP security knowledge is incredibly relevant to current application security and should be required in any organisation for training developers, security researchers and even gathering requirements.OWASP provides guidance on how to develop, purchase, and maintain secure software applications.One of OWASP core principles is that all of their materials be freely available and easily accessible on their website, making it possible for anyone to improve their own web application security.

OBJECTIVES:

The objective of the OWASP is to improves software security and it aims
To track the most common tactics used by hackers and identify the necessary protection against them.And its ammbition to educate developers, designers, architects, and business owners about the risk associated with the most common web application security vulnerabilities.

- The main objective of the **OWASP** is to be aware about the importance of Web application security and it also provides educational resources to help developers.
- It also offers guidance and practices for securing our code and development processes.
- Its main priorities are security risk to help organisations and addressing the issues that pose the greatest threats.
- It establishes a common language and terminologies for discussing web application security risk.

OWASP Top Security Risks & Vulnerabilities 2021 Released:

DATE:DD/MM/YYYY

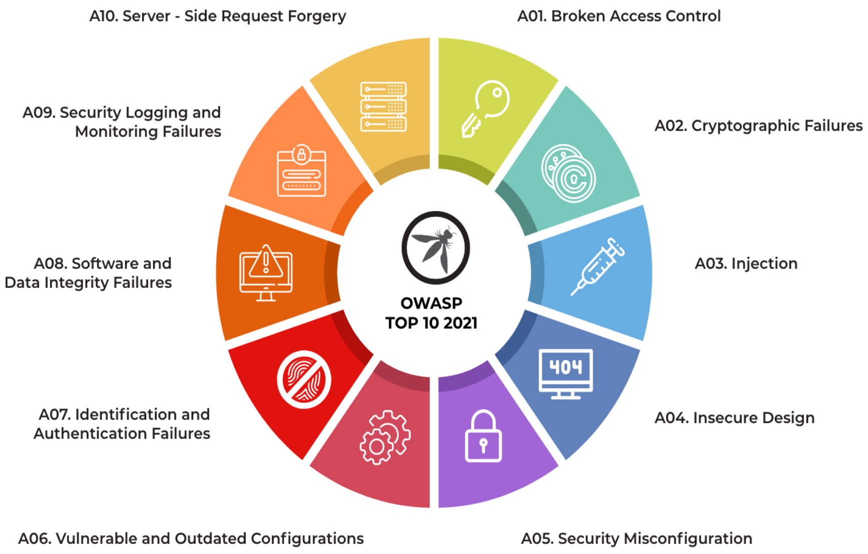
OBJECTIVE:-

-It is to inform and educate our team members about the OWASP TOP 10 was released yesterday for 2021, and it provides new security risk changes for this year.

INTRODUCTION:-

- The OWASP Top 10 is a widely recognized standard for identifying and addressing the most critical web application security risks.
- The 2021 update emphasises key security areas crucial for safeguarding web applications against evolving threats.

The Top 10 OWASP vulnerabilities in 2021 are:


A10. Server - Side Request Forgery
A01. Broken Access Control
A09. Security Logging and Monitoring Failures
A02. Cryptographic Failures
A08. Software and Data Integrity Failures
OWASP TOP 10 2021
A03. Injection
A07. Identification and Authentication Failures
A04. Insecure Design
A06. Vulnerable and Outdated Configurations
A05. Security Misconfiguration
404

- ➔ Broken Access Control
- ➔ Cryptographic Failures
- ➔ Injection
- ➔ Insecure Design
- ➔ Security Misconfiguration
- ➔ Vulnerable and Outdated Components
- ➔ Identification and Authentication Failures
- ➔ Software and Data Integrity Failures
- ➔ Security Logging and Monitoring Failures
- ➔ Server-Side Request Forgery

1.Broken Access Control :-

It is about making sure that users can only access the parts of the application they are allowed to, and not sneak into restricted areas.

-EXAMPLE: It means being able to access private files on a web application without proper permission.

2. Cryptographic failures:-

It means making sure our secret codes are strong and not easily cracked.

- Cryptographic Failures are like using a secret code to lock a treasure chest, but someone figures out how to easily break the code.
- It's a security weakness in how we keep information safe with secret keys or passwords.

-Example:A web application uses weak encryption to store user passwords. If an attacker can easily figure out the passwords because of this weak encryption, it's a cryptographic failure.

3. Injections:-

Injection is like sneaking an unexpected message into a conversation, making the system misunderstand and do things it shouldn't.

-Example: Writing a wrong code into a website's search bar that crashes the entire site .

4. Insecure design:-

It is nothing but a vulnerability that is caused by mistakes in how the websites are built.

-Example: building a website without considering safety measures from the start.

5. Security misconfiguration:-

a web server has default settings enabled that provide too much information about the technology it's using. This information can be useful for attackers.

-Example:Security Misconfiguration is like leaving the back door of your house unlocked without realising it. It's a mistake in setting up the security of a system, allowing attackers to find weaknesses easily.

6. Vulnerable and Outdated Component:-

A web application that uses an older version of a software library. If a new and secure version is available, but the application continues to use the old one, it might have known vulnerabilities that attackers could exploit.

-Example:Having Vulnerable and Outdated Components is like having old, worn-out parts in a car. They might break easily, causing the whole system to fail.

7. Identification and Authentication Failures:-

A website that allows users to log in with just a username and no password. This is an identification and authentication failure because there's no proper way to confirm the user's identity.

-Example:Identification and Authentication Failure is like someone claiming to be a trusted friend without showing any ID.

8. Software and Data Integrity Failures:-

In software and data integrity failure we are assuming that things are safe there is no need to worry without any double-checking.

-Example:Accessing someone's account without a password.

9. Security logging and monitoring failures:-

Security Logging and Monitoring is like having surveillance cameras and alarms in your house. It involves keeping a close eye on activities, logging important events, and setting up alarms for suspicious actions

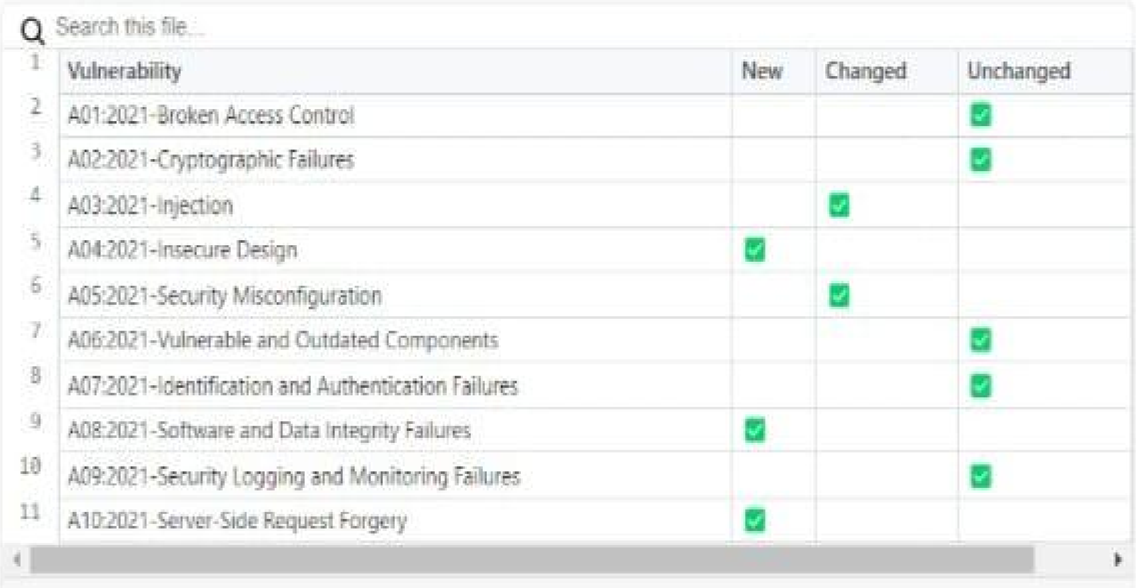
-Example:if the system detects multiple failed login attempts from different locations in a short period, it triggers an alert

10.Server-Side Request Forgery:-

Server-Side Request Forgery is like tricking a server into making requests on your behalf, as if you were the server. It's a way for attackers to make the server fetch information from unintended sources.

-Example: A web application that lets users input a URL to fetch information.

Let's discuss about the changes in 2021:



Vulnerability	New	Changed	Unchanged
A01:2021-Broken Access Control			☑
A02:2021-Cryptographic Failures			☑
A03:2021-Injection		☑	
A04:2021-Insecure Design	☑		
A05:2021-Security Misconfiguration		☑	
A06:2021-Vulnerable and Outdated Components			☑
A07:2021-Identification and Authentication Failures			☑
A08:2021-Software and Data Integrity Failures	☑		
A09:2021-Security Logging and Monitoring Failures			☑
A10:2021-Server-Side Request Forgery	☑		

A03:2021-Injections:-

The first change relates to injections.Injections are attacks in which an attacker attempts to send data to a web application to execute something that the web application was not actually designed to do.

-Example:injection vulnerabilities such as SQL, OS or LDAP injections.

The new OWASP Top 10 Update also contains the vulnerability A07:2017-Cross Site Scripting (XSS), because this vulnerability is in principle also an injection.

A04:2021-Insecure Design:-

Insecure Design is a new category in the OWASP Top 10 and directly started in fourth place four.
It covers architectural flaws and design mistakes that result in a missing or useless control design. While an Insecure implementation could be easily fixed, fixing an insecure design is more complicated or even impossible.

-Example:One of the best-known examples for insecure design is-
password recovery based on questions and answers like
What is the name of your favourite pet?.

Many people know the name of someone else's pet. Also, the name of someone's mother or favourite TV show is easy to guess. This is especially true in the times of social media, where you can find all this information online.

A05:2021-Security Misconfiguration:-

- Modern software gets increasingly complex. We moved from simple systems with one webserver and one database to microservice architectures, where we have several services deployed on multiple servers.

- These are connected to the internet by clusters of reverse proxies and load-balancers.

Including XXE is an attack against an application that processes XML input from a client.

An XML-External-Entities-Attack occurs when untrusted XML input, containing references to external entities, is parsed and processed.

A08:2021-Software and Data Integrity Failures:-

This can occur when a web application relies on plugins, libraries, or modules from sources, repositories, and content delivery networks that are not trusted. Then a CI/CD pipeline, that does not validate external resources, can provide the potential for unauthorised access, malicious code, or system compromise.

-Example: would be an update without signing.

A10:2021-Server-Side Request Forgery (SSRF)

the last change A10:2021-Server-Side Request Forgery (SSRF).

vulnerability can occur when an attacker has full or partial control over the requests that a web application sends.

-Example:web application with three services is given, which have ACLs and authorization rules configured to establish trust between them The service with sensitive data is protected and only accepts requests from the administration service and the user service.

Q1.HOW OWASP IS GOING TO HELP CYBER ATTACKS?

In an age of cybercrime, hackers seek new ways to exploit the vulnerabilities of software systems every day. Denial-of-service attacks, broken access control and data breaches are normal and we as engineers must deal with them. To avoid these security problems, software development teams must be aware of software security. The OWASP Foundation created the OWASP Top 10. A list of the ten most critical security risks to modern web applications, sorted by their observed importance.

Q2. Why OWASP TOP10?

The OWASP (Open Web Application Security Project) Top 10 is a list of the most critical web application security risks. It is widely recognized and accepted as a standard for identifying and prioritising the most prevalent and dangerous security issues facing web applications. The OWASP Top 10 serves several important purposes:

1. Standardised Reference:

- Universal Recognition: It's a widely recognized and respected document in the field of web application security.

-Common Language: Provides a common language for discussing and addressing security risks in web applications.

2. Awareness and Education:

- Accessible Information: It's structured in a way that makes complex security risks more understandable to a broader audience, including non-cyber students.

- Educational Tool: Helps in educating individuals about prevalent security risks and their potential impact.

3. Risk Prioritisation:

- Focus on Critical Risks: Identifies the top 10 most critical security risks, allowing organisations to prioritise their efforts in addressing these vulnerabilities.

-Highlighting Consequences: Helps stakeholders understand the potential consequences of these risks on business operations, data integrity, and user privacy.

4. Guidance for Security Measures:

- Mitigation Strategies: Offers guidance on mitigating these risks, providing actionable steps and best practices to enhance application security.

- Risk Reduction: Helps organisations in implementing preventive measures against common attack vectors.

5. Industry Alignment and Compliance:

-Compliance Framework: Many compliance standards and regulations refer to or incorporate elements from the OWASP Top 10, making it essential for organisations striving to comply with industry standards.

6. Evolving Security Landscape:

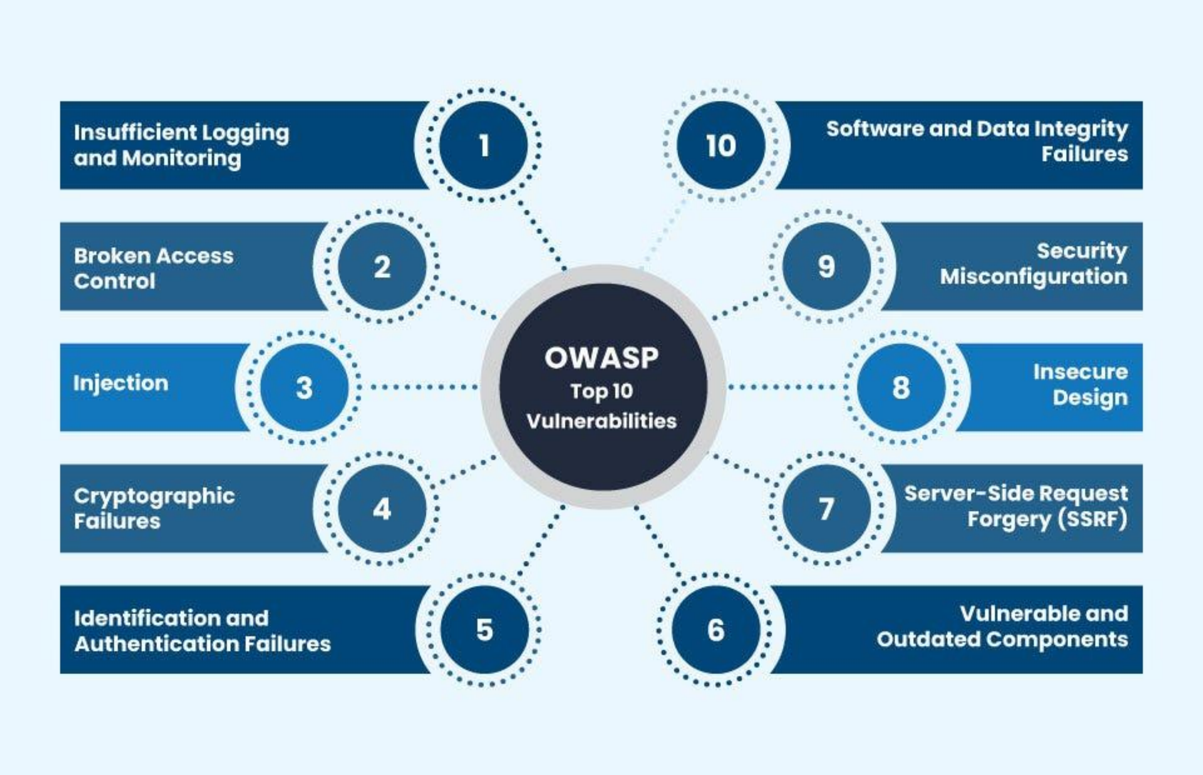
- Updated Regularly: It gets updated periodically to reflect the evolving threat landscape, ensuring it remains relevant in the face of emerging cyber threats.

7. Community-Driven Initiative:

-Community Involvement: Developed by a large community of security experts, practitioners, and volunteers, ensuring a diverse perspective and collective expertise.

OWASP TOP 10 2017:

The OWASP Top 10 is a regularly-updated report outlining security concerns for web application security.


OWASP
Top 10
Vulnerabilities
1
Insufficient Logging and Monitoring
2
Broken Access Control
3
Injection
4
Cryptographic Failures
5
Identification and Authentication Failures
6
Vulnerable and Outdated Components
7
Server-Side Request Forgery (SSRF)
8
Insecure Design
9
Security Misconfiguration
10
Software and Data Integrity Failures

Below are the security risks reported in the **OWASP Top 10 2017** report:

- A01: Injection
- A02: Broken Authentication
- A03: XML external entities(XXE)
- A04: Broken Access control
- A06: security misconfiguration
- A07: Cross-Site Scripting(XSS)
- A08:Insecure deserialization
- A09:Using components with known vulnerabilities
- A10: Insufficient logging and monitoring

1. INJECTION:

Injection is nothing but a vulnerability where untrusted data is sent through an interpreter as part of a command,like sneaky attacks by adding bad code into a system.

-Example: SQL injection, an attacker may input malicious SQL code into an input field of a web application

2. BROKEN AUTHENTICATION:

It is a vulnerability where an attacker can exploit weaknesses in the authentication process of a web application. Authentication is the process of verifying the identity of users, typically through usernames and passwords.

-Example:In a web application, broken authentication might occur if there are weak passwords.

3. SENSITIVE DATA EXPOSURE:

If web applications don't protect sensitive data such as financial information and passwords, attackers can gain access to that data and sell or utilise it for nefarious purposes.

- Example: avoid storing sensitive information if not necessary,and securely dispose of it when no longer needed.

4. XML EXTERNAL ENTITIES (XEE):

It disables the xml external entity and DTD processing if not required.

-example: if the web application is vulnerable to XXE and processes the XML without proper precautions, it might expose the content of the /etc/passwd file.

5. BROKEN ACCESS CONTROL:

It refers to a system that controls access to information.

-Example: It regularly reviews ant test access control configuration to ensure proper restrictions are in place.

6. SECURITY MISCONFIGURATION:

Security misconfiguration is the most common vulnerability on the list, and is often the result of using default configurations or displaying excessively verbose errors.

-Example: it follows security best practices for server and application configuration.

7. CROSS-SITE SCRIPTING (XSS):

Cross site scripting vulnerabilities occur when web applications allow users to add custom code into a url path or onto a website that will be seen by other users. This vulnerability can be exploited to run malicious JavaScript code on a victim’s browser.

-Example: It implements content security policy headers to mitigate XSS attacks.

8. INSECURE DESERIALIZATION:

This threat targets the many web applications which frequently serialise and deserialize data. Serialisation means taking objects from the application code and converting them into a format that can be used for another purpose, such as storing the data to disk or streaming it. Deserialization is just the opposite: converting serialised data back into objects the application can use.

-Example: It uses integrity checks and digital signatures to validate serialised objects.

9. USING COMPONENTS WITH KNOWN VULNERABILITIES:

Many modern web developers use components such as libraries and frameworks in their web applications. These components are pieces of software that help developers avoid

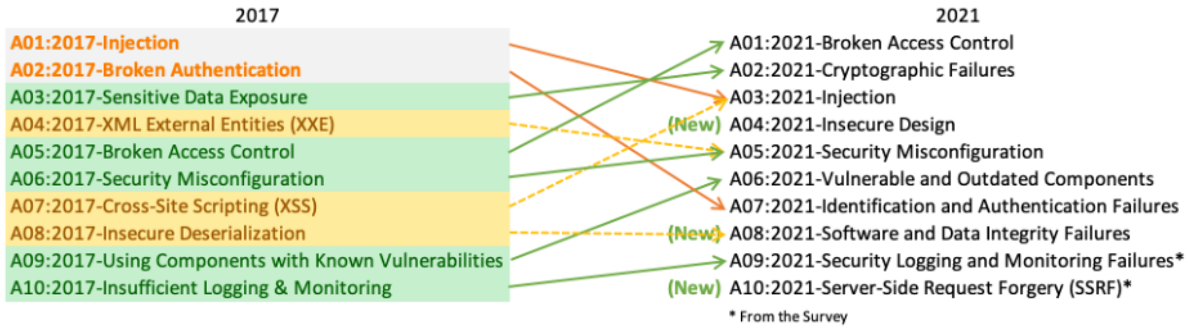
redundant work and provide needed functionality.

-Examples: Include front-end frameworks like React and smaller libraries that used to add shared icons or a/b testing.

10. INSUFFICIENT LOGGING AND MONITORING:

Many web applications are not taking enough steps to detect data breaches. The average discovery time for a breach is around 200 days after it has happened. This gives attackers a lot of time to cause damage before there is any response. OWASP recommends that web developers should implement logging and monitoring as well as incident response plans to ensure that they are made aware of attacks on their applications.

Difference Between OWASP 2017 And 2021


2017
A01:2017-Injection
A02:2017-Broken Authentication
A03:2017-Sensitive Data Exposure
A04:2017-XML External Entities (XXE)
A05:2017-Broken Access Control
A06:2017-Security Misconfiguration
A07:2017-Cross-Site Scripting (XSS)
A08:2017-Insecure Deserialization
A09:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring
2021
A01:2021-Broken Access Control
A02:2021-Cryptographic Failures
A03:2021-Injection
(New) A04:2021-Insecure Design
A05:2021-Security Misconfiguration
A06:2021-Vulnerable and Outdated Components
A07:2021-Identification and Authentication Failures
(New) A08:2021-Software and Data Integrity Failures
A09:2021-Security Logging and Monitoring Failures*
(New) A10:2021-Server-Side Request Forgery (SSRF)*
* From the Survey

Each update typically includes real-world case studies and examples to illustrate the impact of security vulnerabilities. This helps developers and security practitioners better understand the risks and how to mitigate them.The OWASP Top 10 is usually updated to reflect the current threat landscape. Emerging security risks and new attack vectors are considered in each revision.As new vulnerabilities and exploitation techniques come to light, the OWASP Top 10 may be adjusted to include or highlight these issues. The list aims to address the most critical and prevalent risks facing web applications.

CONCLUSION:-

The changes in the OWASP Top 10 are quite interesting. The new category A04:2021-Insecure Design is a clear sign that we need to focus on security even in the design phase. The next thing is that microservices bring security advantages over monoliths. But A10:2021-Server-Side Request Forgery (SSRF) shows that they are not bulletproof, and they need more configuration (A05:2021-Security Misconfiguration).

If we look at the top positions, in 2017 Injection and Broken Authentication were the two most common. With the new OWASP Top 10, this has changed, and both moved down. Injections are now on position 3, and Broken Authentication lost five places and is now on position 7. The two most common OWASP Top 10 are now Broken Access Control and Cryptographic Failures.

There are a variety of reasons for this. On the one hand, the OWASP Top 10 focused on them. On the other hand, the tools to detect them are getting better and better. Overall, the list of CWEs that the OWASP Top 10 covers is long, and many things are too big for manual testing. To write secure software, we need automation and proper tooling.

REFERENCES:-

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

https://www.horangi.com/blog/real-life-examples-of-web-vulnerabilities

https://www.infosectrain.com/blog/owasp-top-10-vulnerabilities-2021-revealed/

https://medium.com/digitalfrontiers/changes-in-owasp-top-10-2017-vs-2021-7cea41838b