# A Survey on Different Vulnerabilities in Security Systems and use of OWASP

Aditya Palshikar
ABMTC (American Business Management & Technology College) Zug, Switzerland

***Abstract— OWASP stands for online web application security project. It is a non-profit organisation established in 2004 in the US where anyone can contribute, donate, or join towards securing the web. It is an open-source platform where there are hundreds of local chapters worldwide, tens of thousands of community members, leading educational and training conferences, OWASP is a source for developers and technologists to secure the web. Hackers use the vulnerabilities in the web when an application is running and use a part of this network. There it is necessary to stop this even from a business perspective. It provides freely available technologies, tools, documentation, methodologies, and articles in the online community. OWASP has received the Haymarket media group editor choice award in 2014.***

## 1. Introduction

Owasp prepares a list of top ten vulnerabilities every three to four years. The list includes the possible impact of these vulnerabilities and how to avoid these threats are also published. These theories are used by a variety of people from different security organisations of companies and universities. The list of top vulnerabilities published were as follows: Sensitive data exposure: All sensitive data at rest or in transit should be encrypted to stop unauthorised access. These accesses of non-encrypted data are the main target of hackers who steal data of passwords, pin number or credit card details. Cross site scripting: It happens when scripts of java or html are used to inject into websites on cloud and then used as a medium to control user session or direct the user to some other websites which is under control of intruder. Injection: Injection means when a command prompt is used to insert data into the application that alters the way the application was designed to function. It can be done by LDAP or SQL injection. It can give access to the data without proper authentications. Insufficient monitoring and logging: It takes almost 200 days on average to detect the online security breach. It gives ample amount of time to steal important data, tamper the servers and implant malicious code. Using components with known vulnerabilities: Frameworks and libraries are open sourced components in many website applications. Any sort of known vulnerability become a medium through which entire website security is at risk. Security misconfigurations: Most of the cloud application breaches occur due to human error in setting the configurations. Default settings are used which aren't secure. Others that are common here are incomplete configuration, high accessibility of cloud and incorrect headers of HTTP. Broken access control: It specifies that the amount of content, databases or pages accessible to a user should be only as much required. Entire access shouldn't be given to all users by default. It can cause browsers to redirect to targeted URLs. Broken authentication: Broken authentication means when an application doesn't run the function of authentication and session management properly or incorrectly. It can allow hackers to use the sessions of some users permanently or temporarily. It can be done by compromising on passwords, security pins or session tokens. It is dangerous for the data in the user's account and other applications in the same network. Insecure deserialization : In this method the intruder remotely executes the code and assumes control of the admin by deserializing the object. An illustration of this method is where a super cookie. XML external entities : When incorrectly configured parser XML is used it can send data to unauthorised locations like a hard drive. It can be used by intruders to scan secret files, internal port scanning and service denial attacks.

## 2. Methods to remove Vulnerabilities

Command prompt should be kept separate from data. It should not be allowed to access or modify data to prevent unintended alterations. The application's coding should be parameterized rather than taking input from the user. The interpreter can be bypassed by using some safe API. There should be some sort of validation system of code to detect harmful activities at the user end. Applications should make use of more than one authentication. A session time for each login should be enforced for limiting the using time. Strong passwords must be made compulsory. All failed login attempts should be carefully monitored to check if they are from an intruder. Session IDs should not appear in URLs. Sensitive data exposure: When data is in transit between the host server and the web browser SSL certificate I.e. Secure Socket Layer must be implemented. While for data at rest user caching response should be disabled. All the sensitive data should be encrypted. XML external entities: Highly sensitive data shouldn't be

serialised. Libraries and XML processors must be patched and less complex data formats ex- JSON must be used. Broken access control: For performing a task the lowest amount of access must be given. Inactive accounts must be deleted. All the actions of each user must be audited. Unnecessary services on the servers must be discontinued. Security misconfigurations: Previously designed templates must be used for dev, test and manufacturing environments that suit the organisation's security needs. An application architecture can be useful for minimising the misconfigured settings and a library must be maintained of correctly configured images. Settings in cloud applications must be monitored continuously and any problem detected must be rectified in real time by automatic workflows. The developer must have the education of the best possible coding of JavaScript or HTML. The websites must be developed by keeping in mind - zero user trust. Suitable security policy keeps being formulated against defences in cross site scripting. Serialised objects must not be accepted by unauthorised sources. Digital signatures must be used to keep safe serial listed objects. Unexpected classes can be detected by enabling type constraints. Using components with known vulnerabilities as We must check all components for known vulnerabilities and do the remedy immediately without delay. Within the configuration management all the company's framework components must be present. An SOP must be prepared including training, testing, and deploying activities of all the patching work. In Insufficient monitoring and logging is already software available to automatically logging and auditing the application to detect any unauthorised activity. Even if the attack has failed to crack in, preventive measures can be taken to strengthen the security.

### 3. Current scenario of the system with advanced security in Owasp

OWASP top 10 are updated every year. Current most vulnerable 10 breaches are as follows:

1. Broken Access Control: Moved up from fifth position. 94% of application were tested for some form of broken access control. It has more occurrences in applications than in any other categories.

2. Cryptographic Failures: Previously known as sensitive data exposure, the renewed focus is on failures related to cryptography which often leads to sensitive data exposure.

3. Injection: 94% of the applications were tested for some sort of injection. It has second most occurrence in applications than in any other category.

4. Insecure Design: It is a new category with focus on design flaws. It calls for more use of threat modelling, secure design patterns and reference architectures.

5. Security Misconfiguration: About 90% of the applications were tested for misconfiguration. With more shifts towards configuration of software this category is not surprisingly moving up.

6. Vulnerable and Outdated Components: Similar to using components with known vulnerability.

7. Identification and Authentic Failures: It has moved down the list of top 10. Increased availability of standardised frameworks seems to be helping.

8. Software and Data Integrity Failures: A new category where software updates, critical data are assumed without verifying integrity.

9.Security Logging and Monitoring Failures: This category is expanded to include more types of failures but is really challenging to test. Failures in this category can directly impact visibility, incident alerting and forensics.

10. Server-Side Request Forgery: This category represents the scenario where the security community members are telling us this is important even though its not illustrated in the data

### 4. Conclusion

OWASP is doing a great job of integrating people with a productive goal of securing the internet. With more people using the internet each year it is equally important to update the security threats every year. More people in this domain should contribute their knowledge towards a good cause that might help thousands of users.

I strongly believe these measures are to a great extent helping people to become alert and look for threats. These top threats should be more publicised so that many more people can be alerted, and it shouldn't be just for IT professionals. I believe not 100% of the threats would be avoided, some new ways hackers might come up with to cause distortion or use part of the bandwidth network. Still OWASP is an unavoidable part of the internet security system and should be more popularised.

### REFERENCES

1. *Bach-Nutman, Matthew. "Understanding the top 10 owasp vulnerabilities." arXiv preprint arXiv:2012.09960 (2020).*
2. *Wiradarma, Anak Agung Bagus Arya, and Gusti Made Arya Sasmita. "IT Risk Management Based on ISO 31000 and OWASP Framework using OSINT at the Information Gathering Stage (Case Study: X Company)." International Journal of*

*Computer Network and Information Security 11.12 (2019): 17.*

3. *Li, Jinfeng. "Vulnerabilities mapping based on OWASP-SANS: a survey for static application security testing (SAST)." Annals of Emerging Technologies in Computing (AETiC), Print ISSN (2020): 2516-0281.*
4. *Fredj, Ouissem Ben, et al. "An OWASP top ten driven survey on web application protection methods." International Conference on Risks and Security of Internet and Systems. Springer, Cham, 2020.*
5. *Qian, Kai, Reza M. Parizi, and Dan Lo. "Owasp risk analysis driven security requirements specification for secure android mobile software development." 2018 IEEE Conference on Dependable and Secure Computing (DSC). IEEE, 2018.*
6. *Pratama, I. P. A. E., and Anak Agung Bagus Arya Wiradarma. "Open source intelligence testing using the owasp version 4 framework at the information gathering stage (case study: X company)." International Journal of Computer Network and Information Security 11.7 (2019): 8-12.*
7. *Lala, Shubham Kumar, Akshat Kumar, and T. Subbulakshmi. "Secure web development using owasp guidelines." 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE, 2021.*
8. *Poston, Howard. "Mapping the OWASP top ten to blockchain." Procedia Computer Science 177 (2020): 613-617.*
9. *Sϕhoel, Halldis, Martin Gilje Jaatun, and Colin Boyd. "OWASP Top 10-Do Startups Care?." 2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security). IEEE, 2018.*
10. *Alanda, Aide, et al. "Mobile application security penetration testing based on OWASP." IOP Conference Series: Materials Science and Engineering. Vol. 846. No. 1. IOP Publishing, 2020.*
11. 2016.