**19141230**

Oxford Brookes University  *Faculty of Technology, Design and Environment*

Tech7004 – Cyber security and the web

Table of Contents

[DELIVERABLE 1 – RESEARCH 2](#_Toc170383528)

[OWASP TOP 10 2](#_Toc170383529)

[HOW DOES THE PROJECT WORK? 2](#_Toc170383530)

[How do they achieve OWASP goals? 3](#_Toc170383531)

[Discussion 4](#_Toc170383532)

[OWASP Application Security Verification Standard 5](#_Toc170383533)

[How does the project work? 5](#_Toc170383534)

[How do they achieve OWASP goals? 7](#_Toc170383535)

[Discussion 7](#_Toc170383536)

[OWASP Web Security Testing Guide 8](#_Toc170383537)

[How does the project work? 8](#_Toc170383538)

[How do they achieve OWASP goals? 9](#_Toc170383539)

[Discussion 9](#_Toc170383540)

[References 10](#_Toc170383541)

[Appendixes 11](#_Toc170383542)

[Appendix 1 11](#_Toc170383543)

DELIVERABLE 1 – RESEARCH

## OWASP TOP 10

### HOW DOES THE PROJECT WORK?

The Open Web Application Security Project (OWASP) TOP 10 is a non-profit collaborative effort of the OWASP community. It is a standard awareness document which is regarded as the first step towards secure coding, as it lists the top 10 most prevalent, and severe web application security vulnerabilities in the dynamic cybersecurity landscape. The project aims to provide stakeholders a starting point for secure software application development, maintenance, and purchase. Volunteers collaborate to identify and rank the vulnerabilities in web applications. Readers can freely access descriptions of the listed security risks and implement the recommended mitigation strategies with discretion. Also, the list is revised every couple of years as new changes in risk and industry emerge.

The procedure of ranking and updating OWASP Top 10 involves five phases:

**1. Initial Planning/Data Call:** The ‘core team’ proposes a schedule and a data call is issued requesting verified and pseudo-anonymous (unverified) contributions. The contribution period lasts approximately six months. For example, for OWASP’s Top 10 2024, contributions last from January to June 2024.

**2. Industry Survey:** A survey strictly focused on web application security topics is released to solicit feedback from verified and unverified industry participants, on the *Common Weakness Enumeration* (CWEs) list. Respondents share insights about their best practices’ and security measures’ effectiveness, data on emerging threats and attack patterns, statistics on the likelihood of various kinds of vulnerabilities, and incident reports on web applications’ security breaches/vulnerabilities. The two highest-voted risks from the survey, not represented in the numeric data, are compared with the data analysis (OWASP TOP 10, no date).

**3. Data Analysis:** Using the OWASP Azure Cloud Infrastructure, analysts examine, review, and normalise the data collected. The preliminary analysis identifies links between CWEs and particular risks, and observes how they are grouped, considering potential overlaps in CWEs categories (OWASP Top Ten, no date).

**4. Draft Top Ten:** A draft list of categories pertaining to the application security risks and methods is constructed, with eight categories derived from numeric data and two from the survey. Drafts are shared publicly for review; and considerations like exploitability, detectability/likelihood, and technical impact are incorporated into the ranking process.

**5. Release:** The core team, having considered all issues raised, reaches a consensus on the final version of the OWASP Top 10 and releases it (OWASP TOP 10, no date).

### How do they achieve OWASP goals?

Using social media platforms, it raises awareness of security risks, providing developers, software architects and business owners guidance on navigating the most critical security risks currently. Documents detailing decisions, methods, and strategies implemented in the process are made publicly available on OWASP, GitHub and OneDrive repositories. Thereby, allowing users make informed decisions about applying the security recommendations.

The project promotes best practices, as it offers guidance on remediation, for enhanced secure coding, application development and vulnerability management. Thereby, reducing incident rates and vulnerabilities and enhancing protection against cyber threats.

By nourishing the community with tangible knowledge and resources, the project facilitates effective addressing of web application security vulnerabilities in real-world scenarios. Consequently, optimising the security of web applications across diverse domains and industries.

Top 10 fosters inclusivity and accessibility. By ensuring compatibility with both open-source and commercial security tools, software, and solutions. Users and contributors have access to resources, learning, skill improvement and collaboration opportunities. Lastly, the project’s collective effort in maintaining a consistent flow of data for identifying the latest prevalent and critical web application security risks, contributes to the improvement of security for all.

### Discussion

A diagram of a number of text

Description automatically generated with medium confidence

*Figure 1 OWASP Top Ten Mapping (2021 Release)*

**Secure Design**

In the category *A04:2021 – Insecure Design* (see Appendix 1 for a break-down of Top Ten category identifiers), the project emphasises the importance of enhancing security controls by employing threat modelling, secure design patterns, reference architectures and principles for every ‘new’ application added to the network infrastructure. This safeguards against anticipated attacks, subsequently reducing the risks related to design flaws (OWASP, 2021a). Such measures help prevent incidents like the SolarWinds attack.

**Security Risk**

The project addresses the most critical web application security risks presently, by exploiting the voluntary contribution of OWASP community members, - including security experts, researchers, industry experts, application security practitioners, and the ‘core team’ of reviewers and validators -, to construct a comprehensive guide that identifies and helps in remediating crucial security flaws. Industries can then allocate their resources accordingly.

**Social Engineering**

OWASP fosters a best practice culture for secure coding and vulnerability management. Which inadvertently helps in mitigating social engineering attacks. Users learn to recognise and respond to suspicious activities.

**Legal, Ethical, and Social Issues**

OWASP Top 10 is not affiliated with any technology company, instead, the expertise and insights of diverse community contributors allow for unbiased and pragmatic recommendations; They show efforts in providing the latest information by re-updating the Top 10 list every couple of years. Also, Top 10 maintains its integrity by remaining transparent throughout the process and encouraging contributors to disclose demographic information. They ensure data privacy, confidentiality and promote ethical practices by holding themselves and members accountable for malpractice. And advice that developers should only apply the security standards in a form that aligns with their jurisdiction’s legal standards.

## OWASP Application Security Verification Standard

### How does the project work?

The Application Security Verification Standard (ASVS) project is a standard testing guide for application technical security controls, adapted to any environment. It is intended for ensuring the security of web applications by protecting against vulnerabilities such as Cross-Site Scripting (XSS) and SQL injection (OWASP, no date a). ASVS recommends it be used as a Secure Coding Checklist in conjunction with DAST and SATS tools (Lewis, 2021), to addresses security vulnerabilities of applications, platforms or organisations.

Organisations can implement ASVS by firstly defining their unique security requirements and refer to the appropriate levels and categories (Figure 2) most relevant to them, based on the nature of their business (Lewis, 2021). Next, applications are designed and developed according to the security requirements stated on the ASVS guide, ensuring the integration of secure coding practices from the beginning. Later, code reviews, penetration testing, and automated security testing tools are employed to ensure compliance with ASVS. Finally, the security posture of the application continuously improves by using the feedback from the verification process. To ensure a comprehensive verification, organisation must report excluded key components, passed and failed tests with solutions and clearly justify why certain security requirements do not apply to a specific application. Keeping detailed records of the testing process will facilitate resolving potential disputes (Lewis, 2021a).

Moreover, there are 14 categories (v1, v2, v3, … , v14.) with their own specific security requirements and verification tasks, each addressing different aspects of application security such as authentication, session management, access control, stored cryptography, error handling and logging, and more.

A screen shot of a computer

Description automatically generated

*Figure 2 Levels of the ASVS Version 4.0.3*

The framework is organised into three levels (Figure 2) tailored to address specific security needs:

**Level 1 – Opportunistic:** The elementary security requirement for all applications, which focuses on basic security hygiene with essential security controls to counteract easily discoverable security issues listed in the OWASP Top 10 and similar checklists. It is the only level that can be tested entirely through penetration testing by humans without requiring documentation and source code. The ASVS discourages this form of testing level alone as it is not completely effective for ensuring security, instead it should be used as a preliminary step in a multi-phase security process (OWASP, 2021a, p.12).

**Level 2 – Standard:** Adequate for data-sensitive applications managing confidential information. This level sets stringent security requirements to protect against advanced threats. However, it is a more complex and resource-intensive process requiring access to documentation, source code, configuration and interaction with the development team for enhanced testing (OWASP, 2021a, p.12).

**Level 3 – Advanced:** The highest verification level intended for critical applications requiring optimum level of security assurance. It encompasses comprehensive verification of security controls, including those in Level 1 and 2, necessitating detailed analysis of architecture and coding; testing is more detailed and thorough. L3 demands extensive collaboration with the development team incurring higher costs and resource demands, in terms of expertise and effort (OWASP, 2021a, p.12).

### How do they achieve OWASP goals?

ASVS raises awareness about security threats among developers, architects and security professionals, through its provision of detailed security requirements and guidelines, freely available to the public, encouraging the integration of security considerations throughout the software development lifecycle (OWASP, no date).

The standard facilitates secure development practices by incorporating security requirements into the development process, which enables early identification of vulnerabilities. Also, ASVS contributes to secure development training by providing a proactive alternative (OWASP, 2021a, p.15) promoting the positive characteristics that define secure software, consequently enhancing the ability of trainees to develop secure applications. Besides, ASVS can be utilised in *agile development processes*, thereby facilitating the identification of missing security controls, the rigorous management of missing security elements, and helps teams reflect/manage security in their agile project process (OWASP, 2021a, p.15). Consequently, ASVS contributes to the secure design and development of web-applications and its dissemination.

Furthermore, ASVS enhances the security posture of applications by ensuring the maintenance of basic security hygiene, the protection of sensitive data and helping organisations reach optimal level of security assurance for their critical applications. The framework’s standardised security verification process permits the practice of consistent and repeatable assessments across various applications and organisations, aiding regulatory compliance(OWASP, 2021a, p.14).

The project fosters collaboration and community engagement by engaging security experts in amending the standard. It ensures transparency and informed application of security recommendations by sharing public documentation of methods, strategies and processes on public repositories within GitHub and OWASP.

### Discussion

ASVS promotes practical and scalable security practices aligned with recognised standards like NIST 800-63 (OWASP, 2021a, p.22), minimising conflicting requirements and compliance costs. It incorporates automated testing tools and integrates security into development pipelines to ensure continuous verification. Detailed architectural guidance and secure coding practices help developers build secure applications, addressing both current and emerging threats. The V1 – Architecture, Design and Threat Modelling Verification Requirement provide guidance on secure design through security architecture reviews, threat modelling, and secure coding practices (OWASP, 2021a, pp.11-17). The framework facilitates identifying and mitigating security risks, with V5 – Validation, Sanitation, and Encoding Verification Requirement addressing injection attacks. V2 – Authentication Verification Requirement mandate multi-factor authentication, reducing the effectiveness of social engineering attacks (OWASP, 2021a, p.22). ASVS also encourages legal and ethical compliance with V13 – API and Web Service Verification Requirement reminding developers to protect sensitive data and respect user privacy.

## OWASP Web Security Testing Guide

### How does the project work?

OWASP Web Security Testing Guide (WSTG) offers a complete methodology to *test the security of web services and applications* by providing clear instructions on how to implement the framework in practice (OWASP, no date b). It does this by outlining best practices and procedures for security testing across multiple aspects of web application security, from planning to execution. Moreover, it provides a comprehensive introduction clearly defining and explaining the significance of each aspect discussed in every chapter of the guide and clarifies the pre-requisites necessary to execute them. This ensures that potential users of the guide are equipped for the implementation of the framework.

Initially, users are required to define the scope and objectives of the security testing, while gathering critical details about the target application and understanding its architecture, using tools like web crawlers and publicly available data. The guide proceeds to provide testing methods and techniques for security misconfigurations and vulnerabilities, ensuring that applications adhere to best practices in configuration and deployment. It places a significant focus on identity management, recommending verifying authentication and authorisation mechanisms, and suggesting scrutinising session management practices to ensure secure session tokens and handling; The greater the number of technical issues tested, the more complete and accurate, the security posture assessment is (Lewis, 2021).

Additionally, WSTG considers social engineering risks by incorporating input validation prevention recommendations addressing SQL injection and Cross-Site Scripting.

In conclusion, with the contribution of security experts, WSTG guides developers in selecting and designing tests tailored to the specific vulnerability unique to the application’s functionality, such as business logic flaws and API security issues (OWASP, no date b).

### How do they achieve OWASP goals?

The WSTG project achieves OWASP goals by disseminating and continuously amending best practices for securing services and applications through it’s open-source project. Numerous industry experts validate the testing framework which augments the integrity of the project (OWASP, no date b). WSTG not only protects both suppliers and consumers from security threats and malicious activity but also educates developers and testers. The project delivers a comprehensive framework which helps people comprehend everything that testing applications implicates, including the economics of insecure software (OWASP, no date b). Moreover, WSTG illuminates insecure practices and vulnerabilities, however, they explain that this is a symptom of the framework being able to define secure practices.

The significance of the project attracts the corporate support from reputable organisations concerned with technology and software, furthering OWASP’s mission to create a collaborative movement for security improvement.

### Discussion

From a secure design perspective,the WSTG includes sections that verify the security of the application’s architecture and configuration, ensuring early identification and mitigation of design flaws. This proactive approach helps in building secure applications from the bottom-up. In terms of security risk, the WSTG provides methodologies for uncovering a wide range of vulnerabilities, enabling proactive risk management, while mainly focusing on technical vulnerabilities.

The WSTG addresses social engineering by testing the robustness of authentication and session management mechanisms, reducing the effectiveness of social engineering attacks like phishing. These mechanisms mitigate risks associated with social engineering attacks. Legal, ethical and social issues are addressed through guidelines for ethical testing practices, emphasising the need for proper authorisation before conducting tests, and ensures that testing activities do not disrupt normal application operations (OWASP, no date b). This focus helps organisations comply with legal standards and promotes responsible behaviour among testers.

# References

Glas, B. (no date) *OWASP/Top10*, *GitHub*. Available at: https://github.com/OWASP/Top10/tree/master/2024/Data (Accessed: 19 April 2024).

Lewis, I. (2021a) *ASVS Assessment and Certification*, *GitHub*. Available at: https://github.com/OWASP/ASVS/blob/master/4.0/en/0x04-Assessment\_and\_Certification.md (Accessed: 21 May 2024).

Lewis, I. (2021b) *Using the ASVS*, *GitHub*. Available at: https://github.com/OWASP/ASVS/blob/master/4.0/en/0x03-Using-ASVS.md (Accessed: 21 May 2024).

OWASP (no date a) *OWASP Application Security Verification Standard*, *OWASP org*. OWASP. Available at: https://owasp.org/www-project-application-security-verification-standard/ (Accessed: 10 May 2024).

OWASP (2020) *Rules of Procedure - Code of Conduct*, *OWASP org*. OWASP. Available at: https://owasp.org/www-policy/operational/code-of-conduct.html#compliance (Accessed: 6 April 2024).

OWASP (2021a) *Application Security Verification Standard 4.0.3*, *OWASP*. Available at: https://github.com/OWASP/ASVS/tree/v4.0.3#latest-stable-version---403 (Accessed: 10 May 2024).

OWASP (2021b) *OWASP Top 10:2021*, *OWASP Org*. Available at: https://owasp.org/Top10/ (Accessed: 20 April 2024).

OWASP (2021c) *OWASP Top Ten*, *Owasp.org*. OWASP. Available at: https://owasp.org/www-project-top-ten/ (Accessed: 6 April 2024).

OWASP (no date b) *WSTG - Latest*, *OWASP org*. Available at: https://owasp.org/www-project-web-security-testing-guide/latest/2-Introduction/ (Accessed: 21 May 2024).

OWASP TOP 10 (no date) *The Making of the OWASP Top Ten*, *OWASP TOP 10 Org*. Available at: https://www.owasptopten.org/themakingoftheowasptopten (Accessed: 18 April 2024).

# Appendixes

## Appendix 1

|  |  |
| --- | --- |
| **Identifier element** | **Description** |
| A | AppSec |
| 01: | Rank number followed by “:” |
| 2021- | Decision year followed by “-“ |
| Broken Access Control | Web Application Security Risk Category |

*Table 1 Breaking Down an OWASP Top 10 Category Identifier.*

*E.g., A01:2021-Broken Access Control*