UDC: 004.774.056/.057
*Professional paper*

# SECURITY STANDARDS FOR WEB APPLICATIONS

**Grela AJVAZI[*], Florim IDRIZI, Agon MEMETI, Bleran VESELI**

*Department of Informatics, Faculty of Natural Sciences and Mathematics*
*[*]Corresponding author e-mail: grela.ajvazi@unite.edu.mk*

**Abstract**

Application security refers to security measures used at the application level to protect against stealing or hacking of data or program code. It includes security considerations that take place throughout application development and design, as well as systems and methods to protect apps after they are put into use. Like any software, web applications inherently have issues. Some of these issues represent genuine vulnerabilities that can be used against organizations. Security for web applications guards against these defects. It entails utilizing secure development methodologies and putting security controls in place at every stage of the software development life cycle (SDLC), making sure that both implementation- and design-level bugs are fixed. Development teams must follow web application security standards to defend software organizations from attack, as online applications are currently the number one target of proven security breaches.
In this article, I'll attempt to explain how web application security works and what developers truly need to do to create secure applications that allow users to enter any data. We will also highlight certain standards that have been developed by various security organizations that have attempted to develop a safe online application in order to make it as simple as possible to comprehend the security of web apps.

*Keywords:* OWASP, CISQ, Web Applications, Security Standards, Web Application Security.

## 1. Introduction

The web has recently been used by organizations not just to market their names, products, and services but also to carry out routine duties, including handling sensitive data and complex workflows. Additionally, several apps are switching from their traditional desktop-based versions to web-based ones in order to target a larger number of devices with low-cost portability due to the popularity and spread of smart hand-held devices [6]. On the other hand, as attackers multiply and develop increasingly risky and sophisticated attack methods, corporations face significant security issues when trying to protect their web applications. As a result, many ways have been suggested to protect online applications, which has made it an essential research topic.

The security and dependability of Web applications have grown to be a major concern as the use of the World Wide Web spreads to more B2B (business-to-business), B2C (business-to-client), healthcare, and e-government services. Eight of the top ten attacks in a Symantec analysis study of network-based assaults, vulnerabilities, and malicious code from 2003 [1] were related to Web applications, and the research also noted that port 80 was the most often attacked TCP port. The authors of the paper said that Web application vulnerabilities were by far the easiest to exploit and that Web applications were to blame for the dramatic increase in moderately severe vulnerabilities discovered in 2003.

The most important communication channels between various clients and service providers are web applications. The harm caused by their security weaknesses has expanded along with their importance. These online applications handle sensitive data while doing crucial tasks. Continuously disclosed vulnerabilities could result in the compromise of sensitive data. The developers' lack of security awareness is the primary cause of this issue [2]. The cost to fix security flaws discovered later in the software development cycle is higher than security flaws discovered earlier [3]. Therefore, it is the responsibility of developers to make every effort to identify problems as soon as feasible. The breadth and complexity of the code bases as well as the difficulty in finding just one developer may make it more difficult to find software flaws. Code auditing (code inspection or reviews), static analysis, dynamic analysis, and security testing can all be used to find vulnerabilities in online applications [4], [5].

Tools, methods, and regulations that detect or reduce security flaws may be included in application security. Routers have a feature called hardware application security that prevents Internet users from viewing a computer's IP address. However, the program typically also includes controls and safeguards at the application level. An example of this is an application firewall, which specifies what actions are permitted and not permitted. Procedures can include things like a procedure for application security that includes regular testing, as well as other protocols.

## 2. What are security standards?

A set of guidelines that ensures consistency, accountability, and efficiency for products or procedures is known as a security standard. Standards are created to offer a repeatable method of doing things, much like policies govern people's behavior. Utilizing written standards may be based on best practices and compliance. This makes it possible for businesses to decide on the use of security devices in an objective manner.

Without standards, it is challenging to define the procedures for installing security devices and why they should be installed in specific locations. As a result, many decisions about how to use and implement security technology are based solely on finances or in response to an incident. A "standard-of-care" defense of a negligent security tort is essentially impossible in light of this reactive response. There are two conditions that must be fulfilled. The first is "Do we consistently use electronic security equipment?" "Can we articulate our position for use?" is the second. Sharing of information and best practices is also made easier by security standards. They aid in ensuring that concepts, terms, and definitions are understood by all, which prevents errors [17].

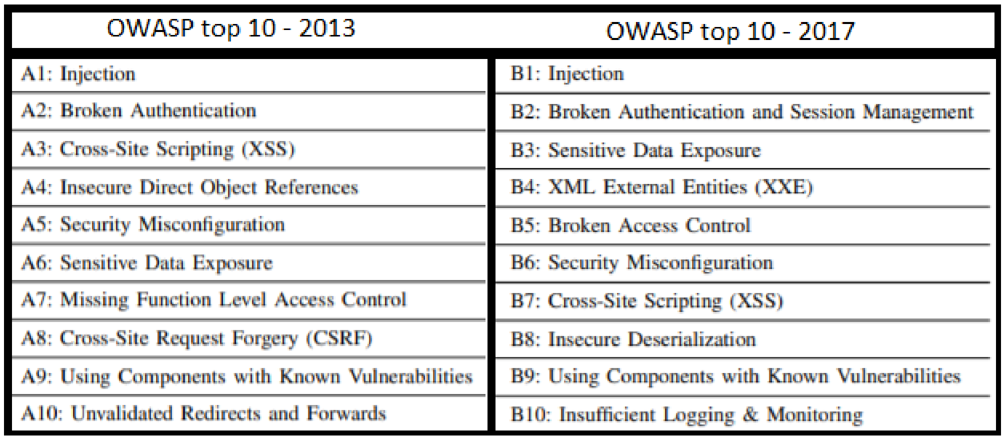
It is possible to write standards so that there are minimum requirements that can be increased as needed to ensure quality. End users can increase the minimum standards in accordance with size and budget by putting in place minimum or baseline security standards. Because of these factors, minimum standards ought to be created in a progressive manner. This format may make it possible to address variations in a facility's size or use in a more efficient manner. For instance, places of similar size and aim—let's call them Level One places—might adhere to the minimal requirements. A Level Two site will have all of the Level One standards in addition to additional minimum requirements because of its larger size or different type of operation.

## 3. OWASP- top 10 web application security risks

A standard asset for developers and web application security is the OWASP Top 10. It shows a more general comprehension of the most important security threats to web applications. Businesses should adopt this document and begin the process of ensuring that the risks associated with their web applications are minimized. The OWASP Top 10 is probably the best place to start if you want to change your organization's software development culture and produce more secure code.

The latest version that received consensus by the start of this project is the OWASP Top 10 2013 [7]. The list was developed by analyzing 500,000 vulnerabilities found in thousands of applications. The vulnerabilities are rated based on several occurrences, exploitability, detectability, and impact. The 2013 ranking is listed in Table 1.
The spring 2017 publication of the OWASP Top 10 was eventually withdrawn due to community disagreements regarding its validity. The final version was finally published on October 20, unfortunately too late to be part of this project. There has been quite a lot happening on the application development front in the last 4 years. Microservices, RESTful APIs, and Single Page Applications have completely changed the architecture of web applications and come with their own set of security challenges. The fundamental technologies have changed and are now dominated by new web frameworks such as Angular and React [8]. Many of the vulnerabilities are rearranged. Fortunately, all of the flaws from the OWASP top 10 2013 are still applicable; just two were removed from the list due to their low frequency.

**Table 1**. OWASP top 10



| OWASP top 10 - 2013 | OWASP top 10 - 2017 |
|---|---|
| A1: Injection | B1: Injection |
| A2: Broken Authentication | B2: Broken Authentication and Session Management |
| A3: Cross-Site Scripting (XSS) | B3: Sensitive Data Exposure |
| A4: Insecure Direct Object References | B4: XML External Entities (XXE) |
| A5: Security Misconfiguration | B5: Broken Access Control |
| A6: Sensitive Data Exposure | B6: Security Misconfiguration |
| A7: Missing Function Level Access Control | B7: Cross-Site Scripting (XSS) |
| A8: Cross-Site Request Forgery (CSRF) | B8: Insecure Deserialization |
| A9: Using Components with Known Vulnerabilities | B9: Using Components with Known Vulnerabilities |
| A10: Unvalidated Redirects and Forwards | B10: Insufficient Logging & Monitoring |

*3.1. Comparison of OWSAP top 10 from 2017 to 2021:* Based on various research and interviews conducted by OWASP itself, we have a relatively large change in the standards set by OWASP during the period from 2017 to 2021, which are best illustrated in Figure 1.

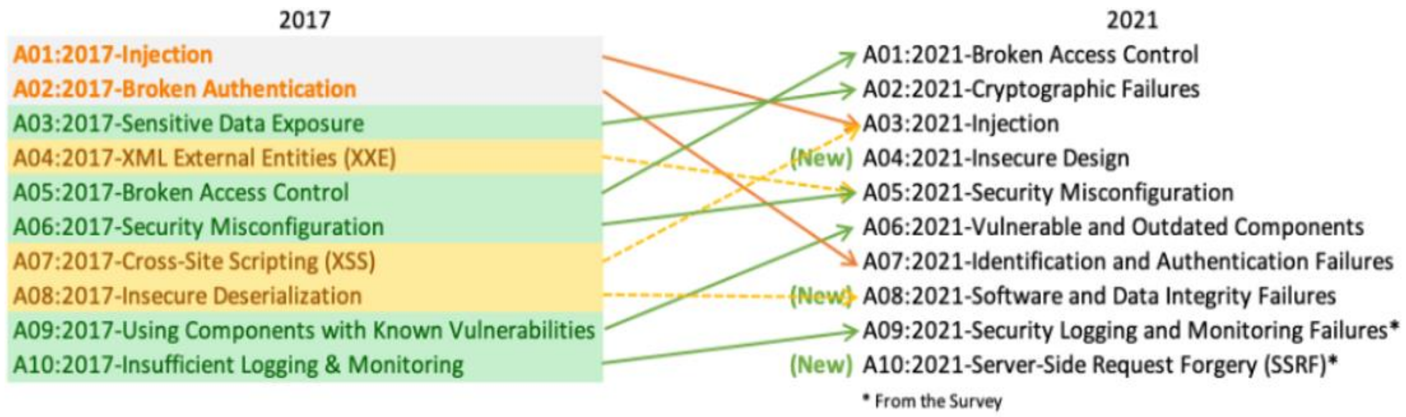

2017
A01:2017-Injection
A02:2017-Broken Authentication
A03:2017-Sensitive Data Exposure
A04:2017-XML External Entities (XXE)
A05:2017-Broken Access Control
A06:2017-Security Misconfiguration
A07:2017-Cross-Site Scripting (XSS)
A08:2017-Insecure Deserialization
A09:2017-Using Components with Known Vulnerabilities
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

**Figure 1.** OWASP from 2017 to 2021

**Broken access control** rises from fifth place in A01:2021; 94% of apps were checked for broken access control.
A02:2021 - **Cryptographic Failures,** previously known as Sensitive Data Exposure, move up one place to #2 but are no longer a root cause. Here, flaws in cryptography - which frequently expose sensitive data or compromise systems - are the subject of renewed attention.
A03:2021 - **The injection** moves down to take up the third position. The 33 CWEs mapped into this category had the second most occurrences in apps, and 94% of the applications were checked for some type of injection.
The new type A042021 - **Insecure Design** for 2021 concentrates on threats associated with design flaws More usage of threat modeling, safe design patterns and principles, and reference architectures are required if our industry is to effectively "move left."
**Security Misconfiguration**, which was ranked #6 in the previous edition, is now at position A05:2021; 90% of applications were examined for security misconfigurations. It's not unexpected to see this category advance given the increasing changes toward highly flexible software.
A06:2021 - **Vulnerable and Outdated Components**, which was formerly known as Using Components with Known Vulnerabilities, is ranked #2 in the Top 10 community survey but also qualified for the list via data analysis because there was sufficient information. This category rises from position nine in 2017 and is a well-known problem for which we struggle to test and evaluate risk.
Broken Authentication, which became A07:2021 - **Identification and Authentication Failures**, is dropping from the second spot and now includes CWEs that are primarily focused on identification problems. As the availability of standardized frameworks has expanded, this category still ranks in the Top 10 and appears to be profiting from it.
The new category for 2021, A08:2021 - **Software and Data Integrity Failures,** focuses on establishing assumptions about software updates. One of the 10 CWEs in this category has one of the highest weighted impacts from Common Vulnerabilities and Exposures/Common Vulnerability Scoring System (CVE/CVSS) data.
Previously known as Insufficient Logging & Monitoring, A09:2021 - **Security Logging and Monitoring Failures** is added from the industry survey (#3), rising from #10 before. This category has been widened to encompass a wider range of failure types.
From the Top 10 Community Survey, A10:2021 - **Server-Side Request Forgery** is included as number one. The information reveals a comparatively low incidence rate and above-average assessments of the potential for Exploitation and Impact [10].

## 4. Related work

A secure development lifecycle (SDL) is a term used for addressing security throughout the software development process and implementing security activities through all phases of development. Another, and perhaps more precise, term used to describe this is Secure Software Development Life cycle (SSDL). Today there exist different methodologies to achieve this; Microsoft SDL, Microsoft SDL Agile, openSAMM (software assurance maturity model), and Comprehensive Lightweight Application Security Process (CLASP) [8].
A survey by Errata Security [9] showed that 81 % of the asked companies had heard about the methodologies, however only 30.4 % were using them. When asked about why they did not use them 23.9 % provided that

they were too time-consuming, 15.2 % that they required too many resources, and 4.3 % that they were too expensive. The survey also showed that smaller companies implemented secure software development life cycles at a higher rate than larger ones. This is natural as smaller companies are more flexible because they do not have as many decision levels and protocols to follow.

## 5. Consortium for Information & Software Quality - CISQ

The Consortium for Information & Software Quality (CISQ) was founded with the goals of advancing the creation and maintenance of secure, dependable, and trustworthy software as well as the establishment of global standards to automate software quality measurement. Industry-supported standards for gauging software size, structural quality, and technical debt from source code have been created due to the efforts of CISQ. For procuring, creating, testing, accepting, securing, and deploying enterprise IT and embedded systems, companies in the private sector, government agencies, and not-for-profits employ tools that are implemented with these standards [13].

CISQ was created to solve a serious issue: IT executives' lack of awareness of the structural quality and risk of the key business applications they manage. A core tenet of CISQ is that addressing this difficulty requires the development of worldwide standards for evaluating software characteristics, particularly at the source code level. To promote the use of quality attributes in benchmarking and managing software procurement, the IT sector needs standard metrics. At the moment, software measurements are far too frequently manual, costly, and based on inconsistent or even subjective concepts.

The Consortium for IT Software Quality (CISQ) has defined five structural quality attributes essential for a piece of software to have business value, they are:

- reliability: an attribute that addresses the risk of application failure and time-out and the implicit impact on users’ efficiency.
- efficiency: an attribute that addresses the performance and scalability of the code and the resources required to execute it
- security: an attribute that addresses the eventuality of security breaches arising from bad coding practices and poor system design
- maintainability: an attribute that addresses the ease with which updates or changes can be made to an application
- size: an attribute indicative of the amount of code produced, the work completed, the complexity of the solution, and hence the maintainability of the application [15].

*5.1. Software standards that work:* The world's leading software engineering specialists were gathered by CISQ to determine the smallest set of coding standards that adequately address the majority of crucial software concerns. This list now has slightly under a hundred coding standards after undergoing two evolutions. It has been empirically demonstrated that these coding standards guard against serious issues in operational systems. Why do CISQ coding standards matter so much? The ones that harm include architecture- and design-level patterns. 90% of structural outages are caused by faults at the system level, according to literature on software engineering.

## 6. Conclusions

Security standards improve an organization's overall risk management and physical security in a variety of ways. By fostering a shared understanding of concepts, phrases, and definitions that can help avoid expensive mistakes, security standards also facilitate the exchange of knowledge and best practices. Written standards provide a mechanism to compare installation procedures and services to objective standards, which can enhance the quality of an installation.

One must first assess the existing security features at the various locations before implementing an enterprise-wide security program. As previously indicated, minimum standards can be increased as needed to ensure quality and effectiveness. It is possible to more effectively address facility disparities by creating progressive security standards.

The standards and guides for conformity assessment published by standardization organizations and institutes reflect an international, regional, or national consensus on best practices. Their use contributes to the consistency of conformity assessment worldwide.

In this paper, I have examined in more detail the standards set by OWASP. They have depending on the requirements and experiences of users, who have been interviewed by the company themselves to give their opinions on how those standards affect their websites. Based on these findings, the OWASP list has changed three times since 2013, with the final list appearing in Figure 1 presented in advance.

CINQ is also mentioned as a part where the standards of security and quality of a website are set, which helps organizations in particular to make faster and better decisions to have profits and be competitive in the market.

## References

[1]. Higgins, M., Ahmad, D., Arnold, C. L., Dunphy, B., Prosser, M., and Weafer, V., “Symantec Internet Security Threat Report—Attack Trends for Q3 and Q4 2002,” Symantec, Feb 2003.

[2]. Jovanovic, N., Kruegel, C., & Kirda, E. (2006, May). Pixy: A static analysis tool for detecting web application vulnerabilities. In IEEE Symposium on Security and Privacy, 2006 (pp. 6-pp).

[3]. Imran, Asif, Shadi Aljawarneh, and Kazi Sakib. "Web Data Amalgamation for Security Engineering: Digital Forensic Investigation of Open Source Cloud." Journal of Universal Computer Science 22.4 (2016): 494-520.

[4]. Livshits, V. Benjamin, and Monica S. Lam. "Finding Security Vulnerabilities in Java Applications with Static Analysis." Usenix Security. Vol. 2013. 2005.

[5]. Lee, Taeseung, et al. "Detection and Mitigation of Web Application Vulnerabilities Based on Security Testing." Network and Parallel Computing. Springer Berlin Heidelberg, 2012. 138-144.

[6]. Jemal, I., Cheikhrouhou, O., Hamam, H., Mahfoudhi, A.: Sql injection attack detection and prevention techniques using machine learning. International Journal of Applied Engineering Research 15(6), 569–580 (2020)

[7]. owasp.org, “Top 10 2013,” 2013, available online at: https://www.owasp. org/index.php/Top 10 2013-Top 10, Last accessed: 2022-02-07.

[8]. Colin Boyd, Martin Gilje Jaatun, Halldis Søhoel, “OWASP Top 10 - Do Startups Care?”, 2018.

[9]. D. Geer, “Are companies actually using secure development life cycles?” IEEE Computer, vol. 43, no. 6, pp. 12–16, 2010. [Online]. Available: https://doi.org/10.1109/MC.2010.159

[10]. https://owasp.org/www-project-top-ten/

[11]. Richard Mark Soley, Bill Curtis, “The Consortium for IT Software Quality (CISQ)”, 2013.

[12]. Aggeliki Tsohou, Spyros Kokolakis, Costas Lambrinoudakis, Stefanos Gritzalis, “A security standards’ framework to facilitate best practices’ awareness and conformity”, November 2010.

[13]. https://www.it-cisq.org/standards/code-quality-standards/

[14]. Dan Constantin Tofan, “Information Security Standards”, 2011.

[15]. https://www.cybersecurityintelligence.com/consortium-for-information-and-software-quality-cisq-5593.html

[16]. https://www.veracode.com/security/web-application-security-standards