What is the OWASP Top 10?

OWASP has developed a number of resources that describe the most common vulnerabilities that exist in various systems, including web applications, APIs, mobile devices, and more. The most famous of these is the OWASP Top Ten, which describes the ten most common and impactful vulnerabilities that appear in production web applications. This list is updated every few years based on a combination of security testing data and surveys of professionals within the industry.

The most recent version of the OWASP Top 10 list was released in 2021. This resource provides information on the most common vulnerabilities, examples of each type, best practices for preventing them, and descriptions of how the vulnerability can be exploited. Additionally, each vulnerability includes references to related Common Weakness Enumeration (CWE) specifications, which describe a particular instance of a vulnerability. For example, the use of hard-coded passwords (CWE-259) falls under the Identification and Authentication Failures vulnerability within the OWASP Top Ten List.

Methodology

The OWASP Top Ten list is based on a combination of analysis of user-provided data and a survey of professionals within the industry. Based on data submitted by the community, the OWASP team determines the top eight vulnerabilities on its list, providing visibility into the vulnerabilities that are most common in production code today. Organizations were asked to submit the CWEs that they saw in testing and the number of applications tested that contained at least one instance of a CWE. The resulting 400 CWEs were then analyzed based on impact and exploitability and classified to produce eight of the top ten categories.

However, with the 2021 update to the list, the OWASP team reserved the bottom two slots on the list for input from a community survey. The goal of this was to reflect more recent and emerging trends in vulnerabilities where a lack of data or inability to test for a vulnerability might cause it to be underrated by a process that relied solely on vulnerabilities discovered during testing.

OWASP Top Vulnerabilities

The latest version of the OWASP Top Ten contained several changes from the previous version. The 2021 list includes the following vulnerabilities:

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 Vulnerable and Outdated Components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

Of these, four vulnerabilities (4, 8, and 10) are brand new, four are unchanged other than ranking, and the remainder consolidates or rename categories from the previous version of the list.

#1. Broken Access Control

Access control systems are intended to ensure that only legitimate users have access to data or functionality. Vulnerabilities in the broken access control category include any issue that allows an attacker to bypass access controls or that fails to implement the principle of least privilege. For example, a web application might allow a user to access another user's account by modifying the provided URL.

#2. Cryptographic Failures

Cryptographic algorithms are invaluable for protecting data privacy and security; however, these algorithms can be very sensitive to implementation or configuration errors. Cryptographic failures include a failure to use encryption at all, misconfigurations of cryptographic algorithms, and insecure key management. For example, an organization might use an insecure hash algorithm for password storage, fail to salt passwords, or use the same salt for all stored user passwords.

#3. Injection

Injection vulnerabilities are made possible by a failure to properly sanitize user input before processing it. This can be especially problematic in languages such as SQL where data and commands are intermingled so that maliciously malformed user-provided data may be interpreted as part of a command. For example, SQL commonly uses single (‘) or double (“) quotation marks to delineate user data within a query, so user input containing these characters might be capable of changing the command being processed.

#4. Insecure Design

Vulnerabilities can be introduced into software during the development process in a couple of different ways. While many of the vulnerabilities on the OWASP Top Ten list deal with implementation errors, this vulnerability describes failures in design that undermine the security of the system. For example, if the design for an application that stores and processes sensitive data does not include an authentication system, then a perfect implementation of the software as designed will still be insecure and fail to properly protect this sensitive data.

#5. Security Misconfiguration

In addition to its design and implementation, the security of an application is also determined by how it is configured. A software manufacturer will have default configurations for their applications, and the users may also enable or disable various settings, which can improve or impair the security of the system. Examples of security misconfigurations could include enabling unnecessary applications or ports, leaving default accounts and passwords active and unchanged, or configuring error messages to expose too much information to a user.

#6 Vulnerable and Outdated Components

Supply chain vulnerabilities have emerged as a major concern in recent years, especially as threat actors have attempted to insert malicious or vulnerable code into commonly used libraries and third-party dependencies. If an organization lacks visibility into the external code that is used within its applications — including nested dependencies — and fails to scan it for dependencies, then it may be vulnerable to exploitation. Also, a failure to promptly apply security updates to these dependencies could leave exploitable vulnerabilities open to attack. For example, an application may import a third-party library that has its own dependencies that could contain known exploitable vulnerabilities.

#7. Identification and Authentication Failures

Many applications and systems require some form of identification and authentication, such as a user proving their identity to an application or a server providing a digital certificate verifying its identity to a user when setting up a TLS-encrypted connection. Identification and authentication failures occur when an application relies upon weak authentication processes or fails to properly validate authentication information. For example, an application that lacks multi-factor authentication (MFA) (/cyber-hub/network-security/what-is-multi-factor-authentication-mfa/)might be vulnerable to a credential stuffing attack in which an attacker automatically tries username and password combinations from a list of weak, common, default, or compromised credentials.

#8. Software and Data Integrity Failures

The Software and Data Integrity Failures vulnerability in the OWASP Top 10 list addresses weaknesses in the security of an organization's DevOps pipeline and software update processes similar to those that made the SolarWinds hack possible. This vulnerability class includes relying on third-party code from untrusted sources or repositories, failing to secure access to the CI/CD pipeline (/cyber-hub/cloud-security/devsecops/what-is-a-ci-cd-pipeline/), and not properly validating the integrity of automatically applied updates. For example, if an attacker can replace a trusted module or dependency with a modified or malicious version, then applications that are built with that dependency could run malicious code or be vulnerable to exploitation.

#9. Security Logging and Monitoring Failures

Security Logging and Monitoring Failures is the first of the vulnerabilities that are derived from survey responses and has moved up from the tenth spot in the previous iteration of the list. Many security incidents are enabled or exacerbated by the fact that an application fails to log significant security events or that these log files are not properly monitored and handled. For example, an application may not generate log files, may generate security logs that lack critical information, or these log files may only be available locally on a computer, making them only useful for investigation after an incident has been detected. All of these failures degrade an organization's ability to rapidly detect a potential security incident and to respond in real-time.

#10. Server-Side Request Forgery

Server-side request forgery (SSRF) is unusual among the vulnerabilities listed in the OWASP Top Ten list because it describes a very specific vulnerability or attack rather than a general category. SSRF vulnerabilities are relatively rare; however, they have a significant impact if they are identified and exploited by an attacker. The Capital One hack is an example of a recent, high-impact security incident that took advantage of an SSRF vulnerability.

SSRF vulnerabilities can exist when a web application does not properly validate a URL provided by a user when fetching a remote resource located at that URL. If this is the case, then an attacker exploiting the vulnerability can use the vulnerable web application to send a request crafted by the attacker to the indicated URL. This allows the attacker to bypass access controls, such as a firewall, which would block direct connections from the attacker to the target URL but is configured to provide access to the vulnerable web application.