The OWASP Top 10

To prepare for future risks, security professionals need to stay informed. Previously, you learned about the **CVE® list**, an openly accessible dictionary of known vulnerabilities and exposures. The CVE® list is an important source of information that the global security community uses to share information with each other.

In this reading, you'll learn about another important resource that security professionals reference, the Open Web Application Security Project, recently renamed Open Worldwide Application Security Project® (OWASP). You'll learn about OWASP's role in the global security community and how companies use this resource to focus their efforts.

What is OWASP?

OWASP is a nonprofit foundation that works to improve the security of software. OWASP is an open platform that security professionals from around the world use to share information, tools, and events that are focused on securing the web.

The OWASP Top 10

One of OWASP's most valuable resources is the OWASP Top 10. The organization has published this list since 2003 as a way to spread awareness of the web's most targeted vulnerabilities. The Top 10 mainly applies to new or custom made software. Many of the world's largest organizations reference the OWASP Top 10 during application development to help ensure their programs address common security mistakes.

Pro tip: OWASP's Top 10 is updated every few years as technologies evolve. Rankings are based on how often the vulnerabilities are discovered and the level of risk they present.

Note: Auditors also use the OWASP Top 10 as one point of reference when checking for regulatory compliance.

Common vulnerabilities

Businesses often make critical security decisions based on the vulnerabilities listed in the OWASP Top 10. This resource influences how businesses design new software that will be on their network, unlike the CVE® list, which helps them identify improvements to existing programs. These are the most regularly listed vulnerabilities that appear in their rankings to know about:

Broken access control

Access controls limit what users can do in a web application. For example, a blog might allow visitors to post comments on a recent article but restricts them from deleting the article entirely. Failures in these mechanisms can lead to unauthorized information disclosure, modification, or destruction. They can also give someone unauthorized access to other business applications.

Cryptographic failures

Information is one of the most important assets businesses need to protect. Privacy laws such as General Data Protection Regulation (GDPR) require sensitive data to be protected by effective encryption methods. Vulnerabilities can occur when businesses fail to encrypt things like

personally identifiable information (PII). For example, if a web application uses a weak hashing algorithm, like MD5, it's more at risk of suffering a data breach.

Injection

Injection occurs when malicious code is inserted into a vulnerable application. Although the app appears to work normally, it does things that it wasn't intended to do. Injection attacks can give threat actors a backdoor into an organization's information system. A common target is a website's login form. When these forms are vulnerable to injection, attackers can insert malicious code that gives them access to modify or steal user credentials.

Insecure design

Applications should be designed in such a way that makes them resilient to attack. When they aren't, they're much more vulnerable to threats like injection attacks or malware infections. Insecure design refers to a wide range of missing or poorly implemented security controls that should have been programmed into an application when it was being developed.

Security misconfiguration

Misconfigurations occur when security settings aren't properly set or maintained. Companies use a variety of different interconnected systems. Mistakes often happen when those systems aren't properly set up or audited. A common example is when businesses deploy equipment, like a network server, using default settings. This can lead businesses to use settings that fail to address the organization's security objectives.

Vulnerable and outdated components

Vulnerable and outdated components is a category that mainly relates to application development. Instead of coding everything from scratch, most developers use open-source libraries to complete their projects faster and easier. This publicly available software is maintained by communities of programmers on a volunteer basis. Applications that use vulnerable components that have not been maintained are at greater risk of being exploited by threat actors.

Identification and authentication failures

Identification is the keyword in this vulnerability category. When applications fail to recognize who should have access and what they're authorized to do, it can lead to serious problems. For example, a home Wi-Fi router normally uses a simple login form to keep unwanted guests off the network. If this defense fails, an attacker can invade the homeowner's privacy.

Software and data integrity failures

Software and data integrity failures are instances when updates or patches are inadequately reviewed before implementation. Attackers might exploit these weaknesses to deliver malicious software. When that occurs, there can be serious downstream effects. Third parties are likely to become infected if a single system is compromised, an event known as a supply chain attack.

A famous example of a supply chain attack is the SolarWinds cyber attack (2020) where hackers injected malicious code into software updates that the company unknowingly released to their customers.

Security logging and monitoring failures

In security, it's important to be able to log and trace back events. Having a record of events like user login attempts is critical to finding and fixing problems. Sufficient monitoring and incident response is equally important.

Server-side request forgery

Companies have public and private information stored on web servers. When you use a hyperlink or click a button on a website, a request is sent to a server that should validate who you are, fetch the appropriate data, and then return it to you.

Server-side request forgeries (SSRFs) are when attackers manipulate the normal operations of a server to read or update other resources on that server. These are possible when an application on the server is vulnerable. Malicious code can be carried by the vulnerable app to the host server that will fetch unauthorized data.

Key takeaways

Staying informed and maintaining awareness about the latest cybersecurity trends can be a useful way to help defend against attacks and prepare for future risks in your security career. OWASP's Top 10 is a useful resource where you can learn more about these vulnerabilities.