More about OWASP security principles

Previously, you learned that cybersecurity analysts help keep data safe and reduce risk for an organization by using a variety of security frameworks, controls, and security principles. In this reading, you will learn about more Open Web Application Security Project, recently renamed Open Worldwide Application Security Project® (OWASP), security principles and how entry-level analysts use them.

Security principles

In the workplace, security principles are embedded in your daily tasks. Whether you are analyzing logs, monitoring a security information and event management (SIEM) dashboard, or using a vulnerability scanner, you will use these principles in some way.

Previously, you were introduced to several OWASP security principles. These included:

- **Minimize attack surface area**: Attack surface refers to all the potential vulnerabilities a threat actor could exploit.
- **Principle of least privilege**: Users have the least amount of access required to perform their everyday tasks.
- **Defense in depth**: Organizations should have varying security controls that mitigate risks and threats.
- **Separation of duties**: Critical actions should rely on multiple people, each of whom follow the principle of least privilege.
- **Keep security simple**: Avoid unnecessarily complicated solutions. Complexity makes security difficult.
- **Fix security issues correctly**: When security incidents occur, identify the root cause, contain the impact, identify vulnerabilities, and conduct tests to ensure that remediation is successful.

Additional OWASP security principles

Next, you'll learn about four additional OWASP security principles that cybersecurity analysts and their teams use to keep organizational operations and people safe.

Establish secure defaults

This principle means that the optimal security state of an application is also its default state for users; it should take extra work to make the application insecure.

Fail securely

Fail securely means that when a control fails or stops, it should do so by defaulting to its most secure option. For example, when a firewall fails it should simply close all connections and block all new ones, rather than start accepting everything.

Don't trust services

Many organizations work with third-party partners. These outside partners often have different security policies than the organization does. And the organization shouldn't explicitly trust that their partners' systems are secure. For example, if a third-party vendor tracks reward points for airline customers, the airline should ensure that the balance is accurate before sharing that information with their customers.

Avoid security by obscurity

The security of key systems should not rely on keeping details hidden. Consider the following example from OWASP (2016):

The security of an application should not rely on keeping the source code secret. Its security should rely upon many other factors, including reasonable password policies, defense in depth, business transaction limits, solid network architecture, and fraud and audit controls.