**Web Application Security**

Web application security refers to a variety of processes, technologies, or methods for protecting web servers, web applications, and web services such as APIs from attack by Internet-based threats. Web application security is crucial to protecting data, customers, and organizations from data theft, interruptions in business continuity, or other harmful results of cybercrime.

**What Is Web Application Security?**

By most estimates, more than three-quarters of all cybercrime targets applications and their vulnerabilities. Web application security products and policies strive to protect applications through measures such as web application firewalls (WAFs), multi-factor authentication (MFA) for users, the use, protection, and validation of cookies to maintain user state and privacy status, and various methods for validating user input to ensure it is not malicious before that input is processed by an application.

**Why Is Web Application Security Important?**

The world today runs on apps, from online banking and remote work apps to personal entertainment delivery and e-commerce. It’s no wonder that applications are a primary target for attackers, who exploit vulnerabilities such as design flaws as well as weaknesses in APIs, open-source code, third-party widgets, and access control.

Common attacks against web applications include:

* Brute force
* Credential stuffing
* [SQL injection](https://www.f5.com/services/resources/glossary/sql-injection) and formjacking injections
* [Cross-site scripting](https://www.f5.com/services/resources/glossary/cross-site-scripting)
* [Cookie poisoning](https://www.f5.com/services/resources/glossary/cookie-poisoning)
* Man-in-the-middle (MITM) and man-in-the-browser attacks
* Sensitive data disclosure
* Insecure deserialization
* Session hijacking

One recent study[1](https://www.f5.com/services/resources/glossary/web-application-security#footnote1) estimated that cybercrime will cost $5.2 trillion in lost value across all industries by 2024. Another estimated the losses will reach $6 trillion annually before then[2](https://www.f5.com/services/resources/glossary/web-application-security#footnote2). Security devices and technologies are crucial for limiting, if not eliminating, such costs. In addition to direct financial and data theft, web application threats can destroy assets, customer goodwill, and business reputations. That makes web application security imperative for organizations of all sizes.

**How Does Web Application Security Work?**

Different approaches to web application security address different vulnerabilities. [Web application firewalls (WAFs)](https://www.f5.com/services/resources/glossary/web-application-firewall), among the more comprehensive, defend against many types of attack by monitoring and filtering traffic between the web application and any user. Configured with policies that help determine what traffic is safe and what isn’t, a WAF can block malicious traffic, preventing it from reaching the web application and preventing the app from releasing any unauthorized data.

Other web application security methods focus on user authentication and access management, app vulnerability scanners, cookie management, traffic visibility, and IP denylists, for instance..

**Common web app vulnerabilities**

the top 10 most common application vulnerabilities include:

* **Injection.** An injection happens when a bad actor sends invalid data to the web app to make it operate differently from the intended purpose of the application.
* **Broken Authentication.**  A broken authentication vulnerability allows a bad actor to gain control over an account within a system or the entire system.
* **Sensitive Data Exposure.**Sensitive data exposure means data is vulnerable to being exploited by a bad actor when it should have been protected.
* **XML External Entities (XXE).**A type of attack against an application that parses XML input and occurs when XML input containing a reference to an external entity is processed by a weakly configured XML parser.
* **Broken Access Control.**When components of a web application are accessible instead of being protected like they should be, leaving them vulnerable to data breaches.
* **Security Misconfigurations.** Incorrectly misconfiguring a web application provides bad actors with an easy way in to exploit sensitive information.
* **Cross Site Scripting (XSS).** An XSS attack means a bad actor injects malicious client-side scripts into a web application.
* **Insecure Deserialization**. Bad actors will exploit anything that interacts with a web application—from URLs to serialized objects—to gain access.
* **Using Components with Known Vulnerabilities.**Instances such as missed software and update change logs can serve as big tip-offs for bad actors looking for ins into a web application. Disregarding updates can allow a known vulnerability to survive within a system.
* **Insufficient Logging and Monitoring.** Lack of efficient logging and monitoring processes increases the chances of a web app being compromised