

## The Ten Most Critical Web Application Security Vulnerabilities 2004

<b>Unvalidated Input</b>	Information from web requests is not validated before being used by a web application. Attackers can use these flaws to attack backend components through a web application.
<b>Broken Access Control</b>	Restrictions on what authenticated users are allowed to do are not properly enforced. Attackers can exploit these flaws to access other users' accounts, view sensitive files, or use unauthorised functions.
<b>Broken Authentication and Session Management</b>	Account credentials and session tokens are not properly protected. Attackers that can compromise passwords, keys, session cookies, or other tokens can defeat authentication and assume other users' identities.
<b>Cross-Site Scripting (XSS) Flaws</b>	The web application can be used as a mechanism to transport an attack to an end users' browser. A successful attack can disclose the end users' session token, attack the local machine, or spoof content to fool the user.
<b>Buffer Overflows</b>	Web application components in some languages that do not properly validate input can be crashed and, in some cases, used to take control of a process. These components can include CGI, libraries, drivers, and web application server components.
<b>Injection Flaws</b>	Web applications pass parameters when they access external systems or the local operating system. If an attacker can embed malicious commands in these parameters, the external system may execute those commands on behalf of the web application.
<b>Improper Error Handling</b>	Error conditions that occur during normal operation are not handled properly. If an attacker can cause errors to occur that the web application does not handle, they can gain detailed system information, deny service, cause security mechanisms to fail, or crash the server.
<b>Insecure Storage</b>	Web applications frequently use cryptographic functions to protect information and credentials. These functions and the code to integrate them have proven difficult to code properly, frequently resulting in weak protection.
<b>Denial Of Service</b>	Attackers can consume web application resources to a point where other legitimate users can no longer access or use the application. Attackers can also lock users out of their accounts or even cause the entire application to fail.
<b>Insecure Configuration Management</b>	Having a strong server configuration standard is critical to a secure web application. These servers have many configuration options that affect security and are not secure out of the box.