Threat	Vector
Injection	Injection flaws occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
Broken authentication	Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.
Cross-site scripting	XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
Broken access control	Restrictions on what authenticated users are allowed to do are not properly enforced. Attackers can exploit these flaws to access unauthorised functionality and/or data.
Security misconfiguration	Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform
Sensitive data exposure	Many web applications and APIs do not properly protect sensitive data. Sensitive data deserves extra protection such as encryption, as well as special precautions when exchanged with the browser.
Insufficient attack protection	The majority of applications and APIs lack the basic ability to detect, prevent, and respond to both manual and automated attacks. Attack protection goes far beyond basic input validation.
Cross-site request forgery	A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application.
Using components with known vulnerabilities	Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover.
Unprotected APIs	Modern applications often involve rich client applications and APIs, such as JavaScript in the browser and mobile apps, that connect to an API of some kind. These APIs are often unprotected and contain numerous vulnerabilities.

Source: OWASP Top 10 Application Security Risks - OWASP (2017)