Injection Attack in malware

What are Injection Attacks?

Injection attacks refer to a broad class of attack vectors. In an injection attack, an attacker supplies untrusted input to a program. This input gets processed by an interpreter as part of a command or query. In turn, this alters the execution of that program.

Injections are amongst the oldest and most dangerous attacks aimed at web applications. They can lead to data theft, data loss, loss of data integrity, denial of service, as well as full system compromise. The primary reason for injection vulnerabilities is usually insufficient user input validation.

This attack type is considered a major problem in web security. It is listed as the number one web application security risk in the OWASP Top 10 – and for a good reason. Injection attacks, particularly SQL Injections (SQLi attacks) and Cross-site Scripting (XSS), are not only very dangerous but also widespread, especially in legacy applications.

▼ Types of Injection Attacks

SQL injection (SQLi) and Cross-site Scripting (XSS) are the most common injection attacks but they are not the only ones. The following is a list of common injection attack types.

Injection attack	Description	Potential impact
Code injection	The attacker injects application code written in the application language. This code may be used to execute operating system commands with the privileges of the user who is running the web application. In advanced cases, the attacker may exploit additional privilege escalation vulnerabilities, which may lead to full web server compromise.	Full system compromise
CRLF injection	The attacker injects an unexpected CRLF (Carriage Return and Line Feed) character sequence. This sequence is used to split an HTTP response header and write arbitrary contents to the response body. This attack may be combined with Cross-site Scripting (XSS).	Cross-site Scripting (XSS)
Cross-site Scripting (XSS)	The attacker injects an arbitrary script (usually in JavaScript) into a legitimate website or web	• Account impersonation •

	application. This script is then executed inside the victim’s browser.	Defacement • Run arbitrary JavaScript in the victim’s browser
Email Header Injection	This attack is very similar to CRLF injections. The attacker sends IMAP/SMTP commands to a mail server that is not directly available via a web application.	• Spam relay • Information disclosure
Host Header Injection	The attacker abuses the implicit trust of the HTTP Host header to poison password-reset functionality and web caches.	• Password-reset poisoning • Cache poisoning
LDAP Injection	The attacker injects LDAP (Lightweight Directory Access Protocol) statements to execute arbitrary LDAP commands. They can gain permissions and modify the contents of the LDAP tree.	• Authentication bypass • Privilege escalation • Information disclosure
OS Command Injection	The attacker injects operating system commands with the privileges of the user who is running the web application. In advanced cases, the attacker may exploit additional privilege escalation vulnerabilities, which may lead to full system compromise.	Full system compromise
SQL Injection (SQLi)	The attacker injects SQL statements that can read or modify database data. In the case of advanced SQL Injection attacks, the attacker can use SQL commands to write arbitrary files to the server and even execute OS commands. This may lead to full system compromise.	• Authentication bypass • Information disclosure • Data loss • Sensitive data theft • Loss of data integrity • Denial of service • Full system compromise.

XPath injection	The attacker injects data into an application to execute crafted XPath queries. They can use them to access unauthorized data and bypass authentication.	• Information disclosure • Authentication bypass

▼ Detection of Injection attacks

The most efficient way to detect injection vulnerabilities, which make injection attacks possible, is by using an automated web vulnerability scanner. You can detect them manually through penetration testing but this takes a lot more time and resources.

▼ Methods to avoid injection attacks

To avoid injection attacks you must code your web applications in a secure way to avoid injection vulnerabilities. The most important part is: never trust user input. The more you restrict, control, and monitor any form of user input, the more you can avoid your application being hacked.

▼ Statistics

- XSS and SQLi – accounted for over 40% of all injection-related CVEs analyzed in Q2-2023, or over 12.5% of ***all*** vulnerabilities examined. And even if we ignore XSS, server-side injections remain the most dangerous threat for APIs (based on Risk X Likelihood analysis), as discussed in our 2022 Year-End API ThreatStats™ Report.
- the biggest hack of 2023, it began May 27, 2023 when CL0P Ransomware Gang began exploiting a previously unknown SQL injection vulnerability in Progress Software’s managed file transfer solution known as MOVEit Transfer.

References

What Are Injection Attacks | Acunetix