Injection

This guide aims to provide some basic knowledge of Injection attacks and how you can defend against them.

It covers the basics of injection and you can understand how injection attacks works, and how to remediate those vulnerabilities.

Content:

According to OWASP Top10 2017 (https://www.owasp.org/index.php/Top_10-2017_Top_10), **I njection flaws** are the most serious web application security risks.

Injection attacks can be devastating to your business, both from a technical aspect and from the business side. Injection can result in data loss, corruption, or disclosure to unauthorized parties, loss of accountability, or denial of access. It can sometimes lead to a complete host takeover. **Once an injection attack takes place, you can no longer trust your data. It may be corrupted or denial of access may occur.**

Eliminating any opportunities for an attacker to take advantage of injection flaws should be a top concern for your business because of the high impact an attack could have on critical business data.
Do I need to be a security specialist to prevent injection attacks?

The short answer is No. You don’t need to be an expert to protect yourself against injection attacks.

Kiuwan assists you in answering these fundamental questions:

1. Does my app have injection vulnerabilities? Which ones and where?
2. How can I remediate them?
3. After fixing my code, have those vulnerabilities been fixed? Did I introduce new ones?

But first of all, you should assess how vulnerable your application is. There are different approaches:

DAST Approach

In Dynamic Application Security Testing, you attack your application searching for injection points, blocking any malicious request addressed to those vulnerable points.

This is what we might call a symptomatic approach: detect the symptoms (the vulnerabilities) and provide a mechanism to avoid those consequences (basically, blocking requests with specific injection patterns).

SAST approach

Kiuwan, however, uses a **Static Application Security Testing** approach. This based on an etiological approach, meaning the **study of underlying causes.** Symptoms are only the surface of the problem. Root causes must be detected and fixed in their origin.

The more you know about how something works, the better you will be able to defend against it.

If you clearly understand how injection works it will not be complicated to fix it. You will see that once a door is open (i.e. there exists an injection point) it’s a matter of imagination on the part of the attacker to take advantage of it.

Kiuwan helps you by scanning your source code and searching for all those injection points.

It shows to you their root causes. It provides remediation clues and lets you assess how effective your remediation is.