WAF Testing Framework
Benchmark Results

Table of Contents

Methodology

This report details the results of running the Web Application Firewall Testing Framework by Imperva against a Web Application Firewall (WAF) of your choice. These results can be used to improve the effectiveness of your WAF by tuning its policies to reduce the number of false positive and false negatives documented in this report. Unlike other WAF testing tools that focus exclusively on generating attack traffic, the Web Application Firewall Testing Framework generates both attack traffic and legitimate traffic. This approach makes it possible to test the ability of a WAF to detect malicious traffic and also to DISTINGUISH malicious traffic from good traffic. It provides a REAL WORLD testing scenario in which the WAF must block attack traffic and avoid blocking good traffic (i.e., generating false positives).

Attacks and Evasion Techniques

The following attack and evasion techniques are included in testing:

- **SQL Injection**: a technique that takes advantage of non-validated input vulnerabilities to pass SQL commands through a Web application for execution by a back-end database. Attackers exploit the fact that programmers often chain together SQL commands with user-provided parameters. When applications are developed this way, attackers can embed SQL commands inside these parameters in order to run SQL queries and/or commands on the back-end database server, giving them access to sensitive data.
- **Cross Site Scripting (XSS)**: an attack that takes advantage of a Web site vulnerability in which the site displays content that includes un-sanitized (i.e., potentially malicious) user-provided data. For example, an attacker might place a hyperlink with an embedded malicious script into an online discussion forum. When the hyperlink is selected, the malicious script launches an attack. For example, the script could copy user cookies containing sensitive, personal or other important data and then send those cookies to the attacker.
- **Remote File Inclusion (RFI)**: an attack that targets the computer servers running Web sites and their applications. RFI exploits are most often attributed to the PHP programming language, which is used by many large firms including Facebook and SugarCRM. RFI works by exploiting applications that dynamically retrieve external scripts. Attackers cause these applications to include a malicious script hosted on a remote server, thereby giving the attacker control over server and data resources. The executed scripts can be used for temporary data theft or manipulation, or for a long term takeover of the vulnerable server.
- **HTTP Parameter Pollution (HPP)**: an evasion technique in which an attack vector in an HTTP request is split between multiple instances of a parameter with the same name. None of the relevant RFCs define the semantics of such manipulation, and therefore each web application delivery platform may deal with it differently. In particular, some environments process such requests by concatenating the values taken from all instances of the same parameter name within the request. This behavior is abused by attackers to bypass pattern-based security mechanisms.

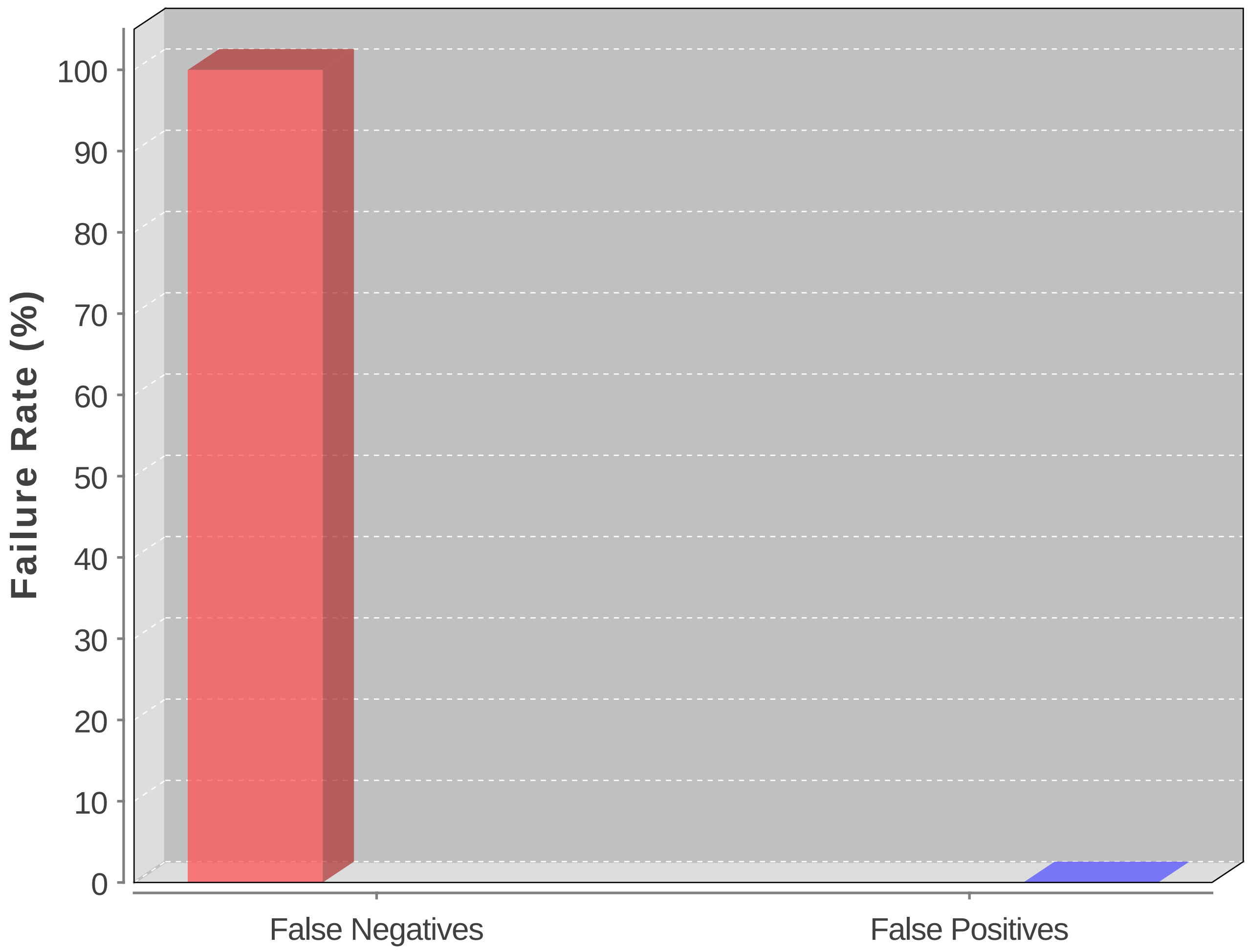
False Positives

False positives are cases where the WAF classifies legitimate traffic as an attack. This happens when the WAF is not sensitive enough to distinguish good traffic from bad. For example, if a WAF considers every request with the single quote character (‘) in it to be SQL injection, as many WAFs do, then a request containing the value “O’Henry” would trigger a false positive.

The following elements are included in the testing:

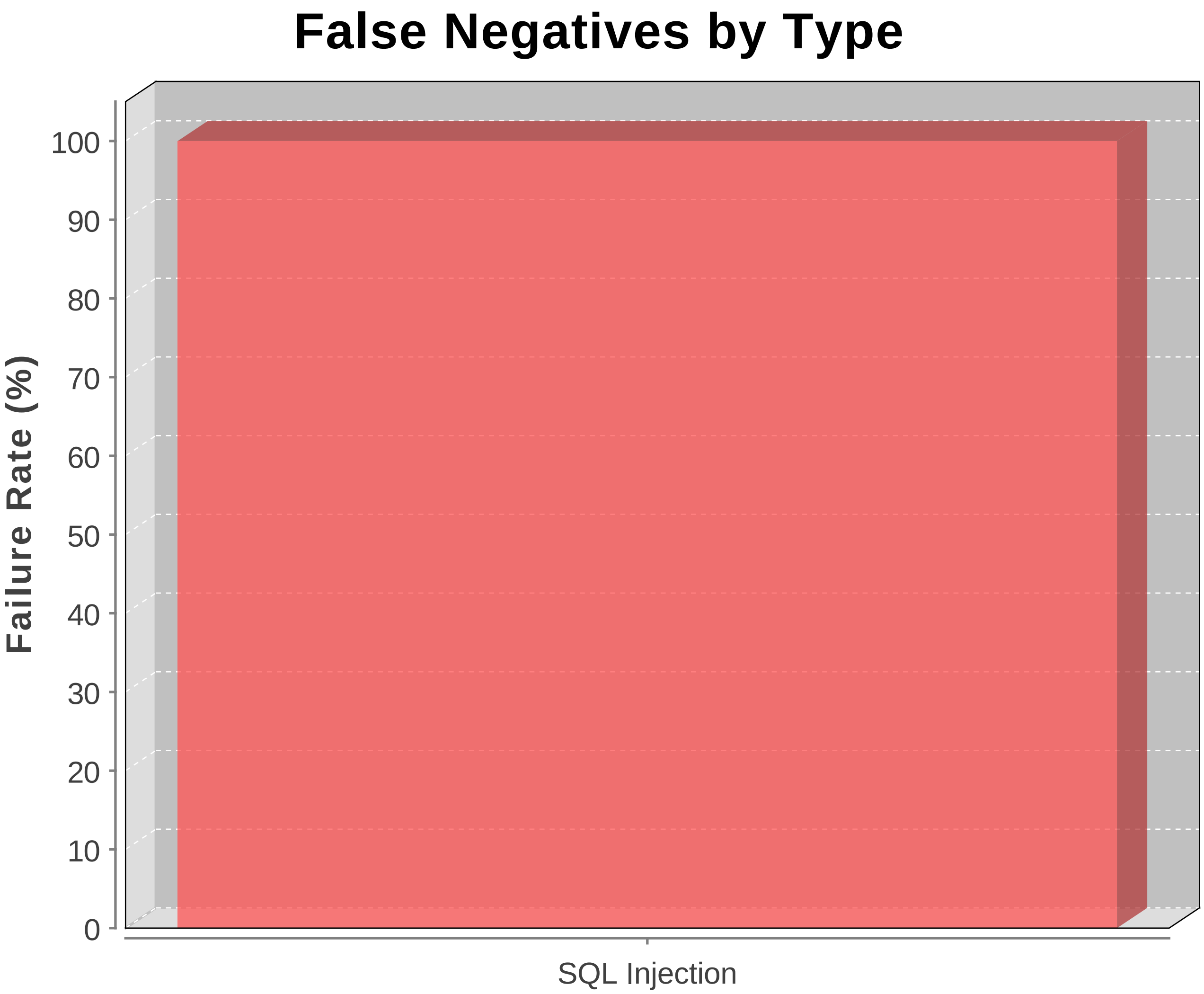
- **Tokens**: certain tokens may trigger a false positive alert. For example, the presence of the tokens ”select” and ”from” may trigger a SQL Injection alert.
- **Special Characters**: characters such as quotes and ampersands may trigger false positive alerts in WAFs that are too strictly tuned.
- **Arbitrarily Text**: sentences or paragraphs. These are common in forum and blog posts, but may be too long or complex for some WAF detection rules, especially if the text contains elements that could appear in attacks (but are in fact harmless when embedded as text in the case of a forum or blog post). In this case, the WAF may issue an alert even though no attack is taking place.
- **Multiple Lines**: the presence of multiple lines in conjunction with certain characters may trigger a false positive “response splitting” alert.
- **External Links**: external links in parameter values along with certain parameter names may trigger a false positive “remote file include” alert. An example of such a parameter is “reditect_uri” which is used by the Facebook API.

False Negatives and False Positives



Test Type	Total Requests	Misclassified	%
False Negatives	1	1	100
False Positives	118	0	0

- **False Negatives:** Attacks which the WAF should have identified and stopped, but did not
- **False Positives:** Legitimate traffic was incorrectly identified as an attack by the WAF



Attack Type	Total Attacks	Misclassified	%
SQL Injection	1	1	100

- **For each attack type:**The percentage of attacks that should have been blocked and was not

False Negatives

<table>
<tr><td><b>ID</b></td><td>2260</td><td><b>Method</b></td><td>GET</td><td><b>Attack Type</b></td><td>SQL Injection</td><td><b>Blocked</b></td><td>No</td></tr>
<tr><td colspan="8"><b>URL</b></td></tr>
<tr><td colspan="8">http://10.0.212.182/WebGoat/attack?Screen=2634&menu=900&message=%27%20drop%20/*&message=*/table%20dbo.users</td></tr>
<tr><td colspan="8"><b>Headers</b></td></tr>
<tr><td colspan="8">Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Authorization: Basic Z3Vlc3Q6Z3Vlc3Q=
Referer: http://10.0.212.182/WebGoat/attack</td></tr>
<tr><td colspan="8"><b>Cookies</b></td></tr>
<tr><td colspan="8">JSESSIONID=CEDD09F025A398786D306220E6EB12BC
session-cookie=14e4ed765d1bbde60a00442900000000dc3abcce8a3eb134a01e91c762730443d5bd55ee13664429e45e98aca2d10a49</td></tr>
<tr><td colspan="8"><b>Comment</b></td></tr>
<tr><td colspan="8">Http Parameter Pollution</td></tr>
</table>

False Positives