The limitations of legacy detection and blocking techniques

Legacy web application firewalls (WAF) that rely on regular-expression pattern matching rules require an extensive tuning period to ensure that default rule sets do not generate false positives and block legitimate traffic for their applications.

The legacy approach of using default rules leaves you with subpar options: you can enable only a small number of rules you’ve validated as “safe” yet leave your applications exposed. Or you can enable more rules that risk blocking legitimate traffic in exchange for catching more potential attacks.

Legacy regex-based WAF approach

Legacy WAF solutions use a set of regex rules to distinguish between normal requests and malicious requests. This often results in false positives, preventing legitimate traffic from reaching the application.
A superior approach to detection

The Fastly Next-Gen WAF (powered by Signal Sciences) was designed by security practitioners that have lived the pain of constant rules tuning and have seen where pattern matching and signature-based rule sets fall short.

Compared to legacy WAFs that rely on regex matching and are rarely used in blocking mode, 90+% of Fastly’s customers enable full blocking mode across all default attack types without any tuning. The key to our reliable, accurate decisions lies in our patented architecture and proprietary detection technology, SmartParse, developed by Signal Sciences. SmartParse makes instantaneous decisions in line to determine if malicious or anomalous payloads are present. By evaluating the context of the request and how it would actually execute, SmartParse is able to make highly accurate detections.

For advanced coverage of today’s unique, complex application logic abuse and attacks, our Next-Gen WAF offers rules which are easy to set up and provide unparalleled flexibility and customization to protect your web applications and services. With our intuitive rule builder, there are no regexes to tune and no complicated rules or scripting language to learn or manage. Through a combination of SmartParse and customizable rules, Fastly delivers automated defense against OWASP Top 10 threats—and beyond.

Injection attacks

With injection attacks via SQLi or XSS, untrusted data is sent to an interpreter as part of a command or query. SmartParse analyzes request parameters to determine whether the code is actually executable and tokenizes the results. The tokenized representation of the request is analyzed, at runtime, to detect attacks such as SQLi, XSS, and other OWASP Top 10 injection attacks. This approach has a much lower false-positive rate and is much faster than signature-based detection approaches.
**SmartParse detection method**

**WEB REQUEST 2**

```
POST /input.html HTTP/1.1

Content-Type: application/x-www-form-urlencoded

msg=Great! Just select users from Jess' list where it makes sense
```

SmartParse tokenizes web requests to provide more accurate detections.

**Coverage beyond injection attacks**

Beyond detection and blocking of SQLi and XSS attacks, Fastly enables you to automatically detect other OWASP Top 10 attacks through rules. With rules, you can set up thresholds, rate limiting, automatic blocking, and alert triggers specific to your web applications and business logic within the Next-Gen WAF console. Rules can also apply vulnerability patches to address outdated or otherwise compromised components, such as libraries, frameworks, and other software modules that run with the same privileges as an application. Rules can be created on individual sites (site rules) as well as globally (corp rules) to be easily used across multiple sites.
The Fastly Next-Gen WAF rule builder provides an intuitive interface for quickly setting up or modifying rules.

**Virtual patching with rules**

A virtual patch prevents the exploitation of a known vulnerability in either a module or framework. A virtual patch analyzes transactions and intercepts attacks in transit, so malicious traffic never reaches the web application. The resulting outcome from applying a virtual patch is that, while the actual source code of the application itself has not been modified, the exploitation attempt does not succeed. This buys time in the development process to fix the underlying vulnerability while the patch is protecting the application at runtime.

Fastly provides and enables you to apply virtual patches that address various Common Vulnerability and Exposures (CVEs) and immediately block requests containing the CVE exploit. Within the console, you can use templated rules that cover various CVEs in a default list.
In this templated rule example, the Fastly Next-Gen WAF can block requests that attempt to exploit the Apache Log4j JNDI vulnerability that leads to remote code execution.

A superior approach to blocking

The Fastly Next-Gen WAF vastly improves detection accuracy by separating blocking decisions from initial detections using a threshold-based approach. Instead of the legacy approach of blocking any incoming request that matches a regular expression (regex) immediately, our Next-Gen WAF uses SmartParse detections coupled with time-based thresholds and anomaly data around the request and response to make informed blocking decisions.

When incoming requests contain attacks, a snippet of that request is sent to our Cloud Engine (see the Data Redactions FAQ to learn how this is done in a safe and private manner). The Cloud Engine aggregates attacks from across all deployed agents—including other customers’ agents through our proprietary Network Learning Exchange (NLX). When enough malicious activity is seen from a potential attacker based on pre-defined yet customizable thresholds built using big data analytics, the engine flags that user for blocking. This method results in highly accurate detections and provides broader context around various attacks.

Using a combination of default detections plus rules functionality, users of the Fastly Next-Gen WAF are able to gain more accurate and far greater blocking coverage across the OWASP Top 10 and advanced threats.
**Getting started**

Learn more about our patented solution to securing web applications:

- Fastly Next-Gen WAF Architecture and Deployment Overview
- Fastly Next-Gen WAF datasheet

[Request a Demo](#)