fastly

PRODUCT DATASHEET

# GraphQL Inspection

## Visibility and protection for your GraphQL APIs

The evolution of API development has played a key role in the explosive growth of dynamic applications. Organizations are seeking more efficient ways to speed up release cycles, and this velocity is changing the CI/CD pipeline, from how releases are organized (from waterfall to agile development, for example) all the way down to the architecture level.

While REST and SOAP have played a major role in how APIs are written, GraphQL is quickly becoming a developer favorite for its efficiency, speed, and specificity.

The Fastly Next-Gen WAF (powered by Signal Sciences) provides advanced protection for your applications, APIs, and microservices, wherever they live, from a single unified solution. With the addition of GraphQL Inspection, we can provide complete coverage over your APIs, no matter what architecture or specification your developers use.

## Setting a new standard for API Protection

The strength of GraphQL over REST is that it allows the caller to request the exact information they need without returning any extraneous data. And this is done with a single call instead of making multiple roundtrips to the backend which decreases overall server strain.

The increased adoption of the open-source query language GraphQL demonstrates how developers are moving away from REST-based APIs to a specification that is faster and more efficient. While this provides a better experience for both developers and the end-users, GraphQL is still subject to OWASP API Top 10 attacks, as well as GraphQL-specific exploits.

The Fastly Next-Gen WAF offers coverage that blocks OWASP-style injection attacks, denial of service attacks, and other vulnerabilities that can target GraphQL APIs. GraphQL Inspection gives your teams the ability to adopt newer technologies without taking on additional application security risk. With over 90% of customers in full blocking mode, we provide comprehensive API protection that doesn't break your application or block legitimate requests.

*While REST has set the standard for API development over the past two decades, organizations are quickly adopting GraphQL as a more efficient way to make calls that power their applications.*

## Key benefits

### Increased API attack visibility and coverage

As your organization increases its application footprint, the Fastly Next-Gen WAF provides scalable and comprehensive protection across all of your Layer 7 assets with full feature parity. Teams can apply their existing WAF rules to GraphQL requests and block attacks, or create custom rules to specifically handle GraphQL traffic.

### Protection where your developers are

Help your developers to deploy applications using the tools they want, without slowing down release cycles. GraphQL Inspection allows organizations to write APIs safely while giving development teams the freedom and flexibility to work within the languages that fit into their workflows.

### Operational efficiency

GraphQL Inspection enables organizations to reap the benefits of GraphQL's efficiency improvements without putting their applications at risk.

## Blocking GraphQL attacks

GraphQL Inspection for the Fastly Next-Gen WAF offers several ways for organizations to take advantage of our patented approach to security:

### Automation

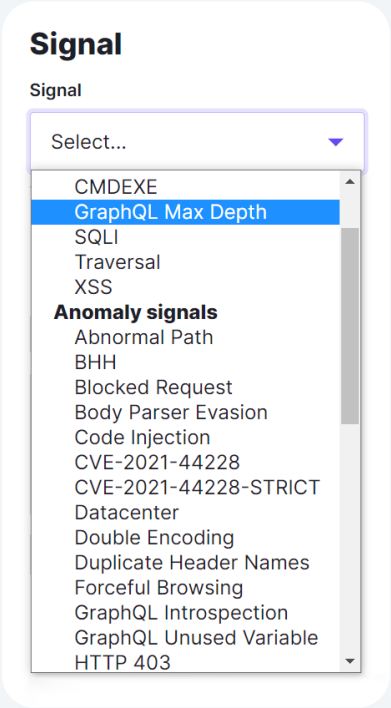
As a part of our out-of-the-box solution, GraphQL Inspection will parse GraphQL requests and inspect the contents of the request within context. If an OWASP-style attack is present, we will block it automatically without any additional setup and configuration required.

### Custom Signals

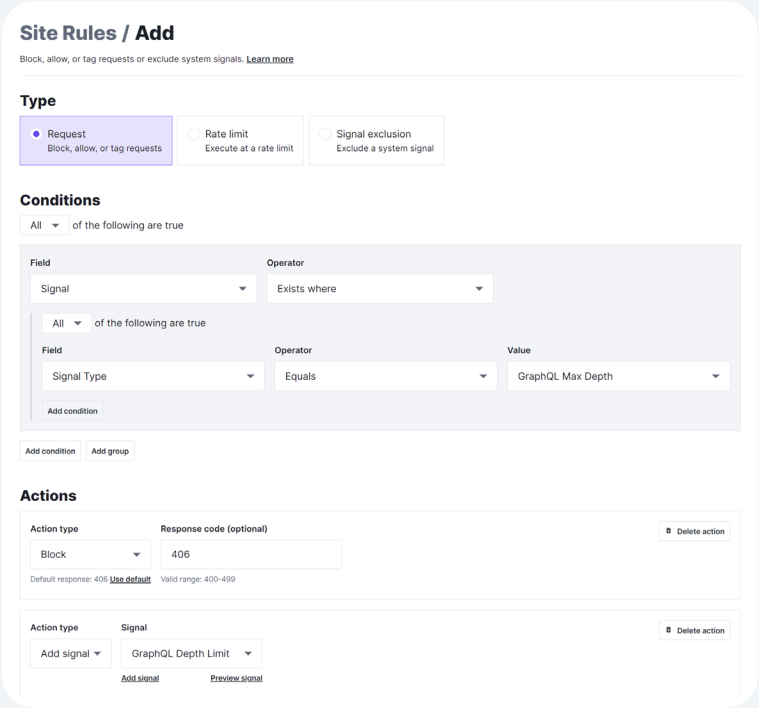
Fastly provides GraphQL-specific signals in the console for customized protection based on user configuration. With these signals, you can define rules to route requests when certain thresholds or events happen. Some of these signals are included below.

### Templated Rules

Includes a templated rule around requests for a GraphQL IDE. GraphQL attacks are now visible via the console.




Image[1]




Image[2]

1: Signals, including GraphQL-specific Signals, can be used to create a Request, Rate Limit or Exclusion rules to a Corp or Site

2: Example of creating a Request Rule using a GraphQL Max Depth Signal to block traffic to avoid misuse or an attack

## What is a signal?

A signal is a descriptive tag about a request. It makes certain characteristics of the request visible and apparent – this includes what types of attack payloads a request contains or whether the request was blocked.

## Common GraphQL Vulnerabilities and Anomaly Signal Examples

| | | | |
|---|---|---|---|
| **OWASP API Security Top 10** | Injection Attacks<br>• SQL Injection<br>• Command Injection<br>• Server-side<br>• Request Forgery | (Distributed) Denial of Service | Broken Access Controls |
| **GraphQL-specific** | Max Depth - complex queries that crash servers | Introspection - public data exposure through queries | Unused variables - sign of an attack within request |

## Getting started

API security must be part of any strategic security plan. Fastly's Next-Gen WAF is the only web application and API protection platform that defends against a wide variety of API threats, across major API security categories and across established and newer API architectures.

To learn more about our security solutions, visit us at fastly.com/secure or contact us at sales@fastly.com.