

Power Rules

A powerful platform with an intuitive user-interface to define, monitor, and take action on any web application or API transaction, providing protection beyond OWASP injection attacks

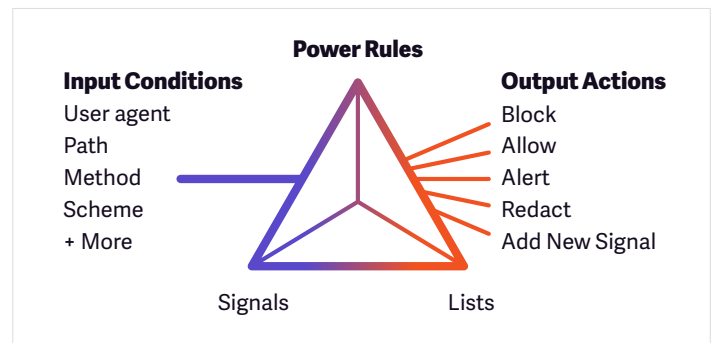
SUMMARY

Power Rules automate actions based on defined Signals and conditions, and require no regexes or knowledge of scripting languages.

Signal Sciences automates coverage against OWASP injection attacks out of the box by utilizing SmartParse, our proprietary detection technology, without any signature configuring or tuning. For advanced coverage of today's unique, complex application logic abuse and attacks, Power Rules are easy to set up and provide unparalleled flexibility and customization to protect your web applications and services. With Power Rules, there are no regexes to tune and no complicated rules or scripting language to learn or manage.

See: Start with Signal Visibility

Power Rules enable you to gain visibility into application logic attacks like feature abuse and misuse, account takeover (ATO) attempts, bad bot activity, and more. By using our intuitive user-interface in the Console (also configurable via API), you can define your own application-specific Signals with inputs selected from drop-down menus, including user agent, path, method, scheme, post or query parameter, request cookies, and more. Signal Sciences Console displays your unique Signals on time-series dashboards for easy monitoring. The Console also offers several templates for popular login and registration workflows to apply Signals to failed and successful attempts, which form the basis of account takeover detection.



Secure: Trigger Automated Actions

Define conditions and thresholds using Signals to trigger actions to alert and block malicious and anomalous application traffic specific to your web applications and services. Alert triggers automate push notifications to Slack,

PagerDuty, Datadog, Splunk, and more. For some actions, you might just want alerts, such as when you exceed a set threshold for TOR traffic; for others, you can set the Power Rule to alert and block, such as surpassing a threshold of failed login attempts.

Scale: Augment Signal Visibility with Lists

Using lists, you can augment Signal visibility and trigger conditions with your own trusted data sources. Lists allow you to parameterize Rules with business data you have collected, such as IPs, user agents, countries, wildcards, and more.

The screenshot shows a configuration form with three main sections: 'Field', 'Operator', and 'Value'. The 'Field' dropdown is set to 'Path'. The 'Operator' dropdown is set to 'Is in list'. The 'Value' dropdown is open, showing three options: 'Checkout Paths' (selected with a checkmark), 'Banned IPs', and 'Whitelisted Partner IPs'.

The screenshot shows a 'Conditions' configuration panel. At the top, it says 'All of the following are true'. Below this is a table-like structure with three rows: 'Field' (IP Address), 'Operator' (Equals), and 'Value' (1.2.3.4/8). Below the table, there is an 'Action' section with three radio buttons: 'Block' (selected), 'Allow', and 'None'. At the bottom, there is an 'Add Signal (optional)' section with a 'None' dropdown and a 'Create new signal' button.

Outright Blocking

Basic functions like whitelisting, blacklisting and virtual patching for application CVEs are also configurable using Power Rules drop-down menus in the user-interface as well as via API.

Power Rules are included with your Signal Sciences license.

USE CASES

- Protection against account takeover, brute force, and credential stuffing attacks
- Visibility and blocking of application and feature abuse for orders, transfers, queries, and more
- Request rate limiting
- Data privacy with redaction of sensitive fields
- Auto-blocking OFAC traffic
- CVE coverage for zero-day exploits



Any App

Cloud, Containers, PaaS,
and Serverless
Web Servers and Languages
Gateways and Proxies



Any Attack

OWASP Top 10
Application DoS
Brute force attacks
+ MORE



Any DevOps Toolchain

Slack Splunk
HipChat SIEM/SOC
Datadog tools via APIs
Webhooks **+ MORE**