

Agent Performance Benchmarks

Summary

The Signal Sciences engineering team developed an agent performance testing framework to determine CPU and memory utilization in various host configurations. The resulting benchmarks represent what our customers routinely see in their production deployments: that our lightweight agent requires few host resources, will not create significant latency yet remains performant.

Introduction

From inception, Signal Sciences' agent was designed to be as lightweight and efficient as possible wherever our customers run their apps and APIs. We provide the agent performance metrics in this document based on internal performance testing to prove out that our agent is lightweight, requires few host resources, and will not create significant latency once installed¹.

When evaluating web application and API protection solutions, we know performance is top of mind for development and operations teams: business-critical applications must be performant and available to drive your business. Additionally, legacy WAF and other application and API protection solutions typically lead to significant performance and reliability issues that do not occur with Signal Sciences.

¹ Performance can vary by environment so you may not see similar results.

Example Performance Benchmark Scenario

To report on the benchmarks presented in this document, our engineering team analyzed our agent usage metrics across multiple typical customer configurations. Below we focus on one test scenario our team chose because it reflects a common host configuration we see across our customer base.

Most acronyms used in this document will be known to development, operations and security professionals, but to be clear “RPS” stands for “Requests Per Second”: 1,000 web requests per second per server is well over the average request volume of even high scale web apps and APIs experience in production.

As the data below proves, even with significant request traffic load our agent decision remains fast and performant, which is why it’s trusted by some of the largest scale companies on the Internet.

Before moving onto the benchmark results, there are two factors that influence the resulting agent performance metrics in our test scenario::

- Request rate (RPS) and size have the most impact on performance results
- The larger the request volume, the bigger the request size and thus the more CPU and memory resources required.

Typical Test Scenario

1,000 RPS on compute instance with four cores running at 3.4GHz: running the benchmark test on a host with this configuration, the following findings result:

- **Median agent decision time required:** under one millisecond¹. Larger requests take longer, but the agent still decides to block or allow the request in under a millisecond for requests over 7KB in size.
- **CPU utilization on average:** less than 10% CPU²
- **Agent memory utilization:** less than 40MB of available host memory³ — memory utilization is dependent on volume.

RPS Simulated	Request Size (bytes)	Median Agent Decision Time (milliseconds)	Avg. Agent CPU Utilization % (out of 100%)	Avg. Memory Usage (bytes)
1,000	none (0)	0.17	6.29	31MB
1,000	small (560)	0.25	9.65	32MB
1,000	med (7320)	0.98 ¹	8.16 ²	39MB ³

NOTE: the resulting metrics do not include overhead potentially generated by the module. The Signal Sciences’ module resource consumption is highly dependent on the application itself. However, typical applications add very little overhead.

Learn more about agent performance with a demo

To learn how the Signal Sciences agent performs while protecting apps [request a demo](#).

