

PREVENTING BAD ACTORS FROM SPOILING THE SHOW AT **carwow**



ABOUT

carwow connects auto dealers and buyers in a transparent marketplace, creating better buyer experiences and incremental sales for dealers.

ENVIRONMENT

Heroku

Rails

Microservices-lite, one app per region, multiple apps per country

Kafka

GOALS

carwow's engineering team faced an urgent need to scale their services seamlessly and quickly as their successful online car review and selling business expanded into larger markets in new regions.

To continue to maintain a high quality of service, they needed to be able to catch performance problems and identify affected (internal and external) customers, but the tools they had been using could not show them the actual source of slowness and issues. As their infrastructure grew in scale and complexity to support their expanding business landscape, it became harder to pick out the real reason for slowness when all they had to work with was aggregated/averaged data.

"We run on Heroku, so we used New Relic because it came with the platform—but it's not enough." —**Mike Wagg, Technical Director**

Because the service is hosted on Heroku, all endpoints use the same request queue, and identifying which requests are slow is important to prevent queuing across all endpoints.

WHAT THEY NEEDED:

- An observability service that allowed them to maintain deep visibility into their infrastructure as they scale it to meet their flexibility requirements
- A single interface to trace request performance and the interaction of systems across their architecture from beginning to end

PREVENTING BAD ACTORS FROM SPOILING THE SHOW AT **carwow**

HONEYCOMB @ CARWOW

"Our monitoring reported that request responses were slow or timing out, and all New Relic could tell us was "some endpoints are slow and triggering errors". " **–Isaac Seymour, Senior Developer**

The engineers at carwow looked at endpoint performance information in New Relic and could see there was a big spike in error rates across many endpoints in one of their apps but there was no way to tell which requests were causing the slowness.

"In New Relic, it just looked like normality, with lots of traffic." **–Mike Wagg, Technical Director**

Because this was happening in a part of the service that did not require users to log in, they had no way to dig in further with New Relic. When they turned to Honeycomb, they were able to break down by the IP each request was coming from—a high-cardinality field.

"Answering the question, "Is this affecting all our users or just the subset?" is absolutely the thing that Honeycomb does that no one gets near to." **–Mike Wagg, Technical Director**

And through the lens of Honeycomb, it all became clearer—a series of malicious requests, attempts to perform SQL injection attacks were coming from a single IP, which they were able to immediately block—but there would have been no way of knowing or addressing this with their previous tools. Figuring this out required being able to isolate the high-cardinality field of each endpoint over a particular time.

"Honeycomb is the only place we have to go to really find this stuff out. It required being able to dig in and look at the raw data—that's one of the differentiators in Honeycomb—being able to isolate some specific time period and drop into the raw data." **–Mike Wagg, Technical Director**

Once they found the culprit, they were able to quickly mitigate and address the attack, and resolve the performance issue for their customers. The ability to break down by the highest-cardinality fields — fast — saves the day again.