

During the 2023 run of track.easterbunny.cc, we have retroactively determined that the tracker had an approximate 20-30% load failure rate in production. This in turn led to decreased traffic on the tracker throughout the entirety of April 8, resulting in abnormally high viewership on YouTube of the tracker.

This post-mortem attempts to explain why this issue occurred, how this issue occurred, and what steps we took to resolve the issue.

Background:

track.easterbunny.cc is a React-based tracker proxied behind Cloudflare. We do this as it takes an enormous load off our origin servers and lets Cloudflare do most of the heavy lifting. We also use Cloudflare as a source for analytics so we can look further into where and how the tracker is being requested.

Cloudflare offers a suite of web analytics, where on a select number of devices, it sends down a beacon. This beacon then reports back to us on data like how fast the site loads which is helpful to ensure that the tracker is loading fast.

During testing of Version 6 of the tracker, we occasionally saw that the tracker would sometimes not recover from the end of one test run to the next test run, displaying a white screen like numerous users saw. Upon further investigation when this issue occurred, we found out a 404 call to a cloudflareanalytics.com URL was 404ing, preventing the rest of the tracker from loading (more information in issue manifestation and description)

However, we generally found that closing out of the tab and reopening it usually fixed the issue. Nonetheless, our development environment did not mirror the actual production environment, as Cloudflare caching was disabled. We did this so that any changes we pushed to development were instantly visible.

Additionally, at some point before tracking, we modified the page rules so that any path on track.easterbunny.cc would have a browser & edge case TTL (time to live) of 5 hours. While we cannot say that this directly caused the issues we had during the 2023 run, it is a possibility.

Lastly, throughout the course of action, we have access to an internal insights page. This page was updated for 2023 to show reference data on 2022's tracker run, including Year-over-Year growth plotted on graphs.

Summary:

track.easterbunny.cc failed due to bad infrastructure configurations, the result of us not conducting enough testing as we moved to a React based tracker and not catching errors fast enough.

Issues during production:

The issues that manifested during production was a two-fold issue that we believe were the part of Cloudflare Analytics and excessive JS bundle purging, resulting in errors when attempting to load the tracker.

With Cloudflare Analytics, generally speaking we believe this was the primary cause of tracker failure in production. The issue pipeline generally is as follows:

- Your browser loads the tracker HTML. On failing devices, we generally believe this was a previously cached HTML file.
- Your browser's cached HTML file has a URL for a beacon.min.js file from Cloudflare Analytics on a random basis, as Web Analytics uses a sampling system.
- The beacon.min.js file is invalid, and returns a 404.
- Because the beacon is JavaScript code, the tracker cannot load the remainder of tracker scripts. Users are then left with a blank white page because the JavaScript cannot render the DOM (and there is no content in the HTML file).

We successfully ran Version 5.6 of the tracker with web analytics enabled and a small rate of failure in production. Therefore, it has led us to believe that incorrectly configured page rules were potentially the culprit of web analytics failing to load.

Additionally, we ran into issues with excessive JavaScript bundle purging throughout the course of tracking, resulting in additional tracker instances failing to load as realized through this post-mortem.

During development, we follow this development pipeline:

- We run a build process which builds the tracker's JavaScript code. Any change to the code creates a new main.<sha hash>.js file, the SHA hash changing on any change to tracker code.
- We deploy files to the server using a reverse deletion method, i.e., if there are any files on the remote server that do not match what we have locally built, they are deleted from the remote server.

We initially had this reverse deletion due to the build-up of JavaScript files. However, this excessive purging ultimately resulted in additional tracker failures due to cached HTML files wanting to request older JS bundles that did not exist.

Generally, the process in which a tracker instance requests an old JS file to failure is as follows:

- Your browser loads in the tracker HTML, which contains a URL to an older JS bundle
- Your browser requests a older JS bundle from Cloudflare
- The Cloudflare edge node may not have this older JS file (due to edge cache expiry), and tries to hit the origin server to request the older JS file
- Because the older JS file does not exist, our server returns a 404, which is forwarded back to the client
- Because the JavaScript fails to load, the tracker fails to load.

It should be noted that sometimes when loading the tracker, your browser keeps a “very old” (5+ hour old) HTML file cached from an earlier request even if the browser is intended to reach out to the server as the HTML file has expired. Sometimes you can get around this by refreshing immediately after to force the browser to get a newer HTML file with the proper links to the JS files.

This issue would have not occurred if we were not patching the tracker in production. However, because we were patching and releasing new JS bundles, there was potential for mismatched bundle files.

We cannot determine these variables for these failures:

- If the issue occurred on first load or after the first load
- If the issue occurred after a certain period of time
 - In a community poll with 31 votes, 32% of users said this issue occurred 5+ hours after loading the tracker, so we cannot say this with certainty.
- If the incident rate generally followed the Cloudflare beacon sampling rate, which is not 100% (generally the sampling rate is 25%)

Timeline:

During testing in the weeks prior to launch, we ran into issues with Cloudflare Analytics not loading. In particular, we experienced these issues as the tracker went from the end of a development run to the next.

We were testing the tracker on a near daily basis, however, this incidence rate only occurred at about 2%. We were able to catch this issue and debug it on a laptop, showing a 404 error code for Cloudflare Analytics. However, we determined the incident rate to be insignificant and did not

When the tracker launched at 2 AM EDT, we detected no issues and initially saw a positive net growth of ~10%. At about 3:28 AM EDT, the tracker had reached the inflection point of growth - meaning that the YoY growth was 0%.

Year-over-Year Growth from 2022 Graph



(Figure 1: YoY growth on our insights page, from 2 AM EDT 4/9 to ~6 AM EDT 4/9)

As the morning rolled on, our YoY growth count continued to slip as presumably the tracker failed to load. We passed -10% YoY at 4:05 AM EDT, -20% at 6:30 AM EDT, and -30% at 2:30 PM EDT.

At about 10 AM EDT, we noticed that the stream we were running had abnormally high viewership count - 170 watching at the time. All streams on YouTube combined had about 300-400 watching, which was about 200-300% higher than at that time compared to the year prior.

Around this time, we began to try and theorize why we were seeing significantly lower traffic. The first working theory was the fact that COVID-related restrictions have decreased from 2022 to 2023, so users were no longer tracking the Easter Bunny and returning to regular Easter activities. We also theorized that YouTube usage had increased this year as potentially people enjoyed tracking on YouTube rather than on the website. In hindsight, none of these theories were correct.

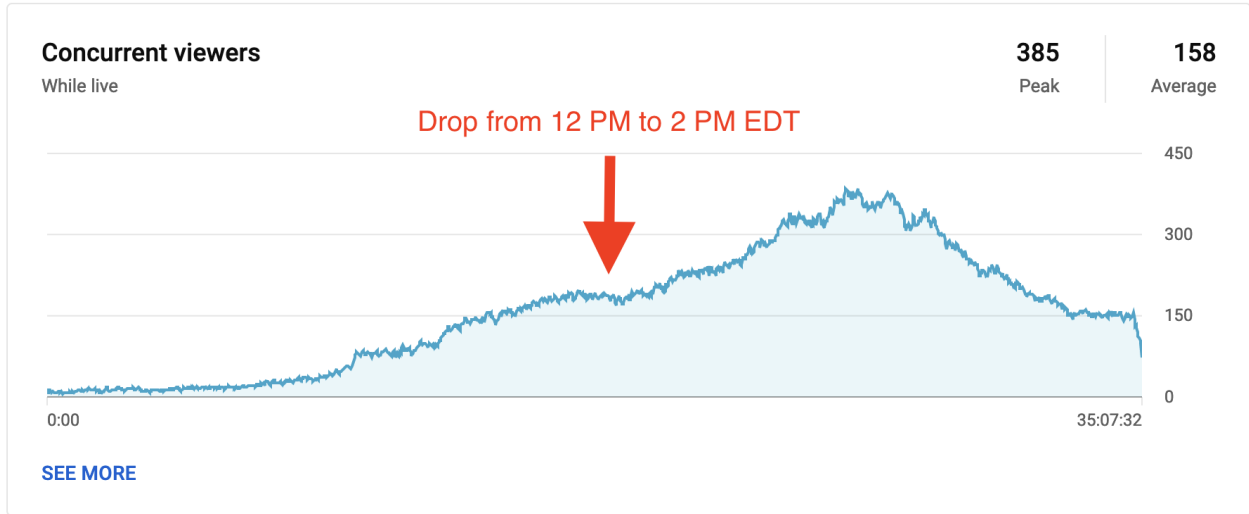
At about 12:10 PM EDT, a user on our Discord server reported they were having issues with the tracker not loading - hitting a white screen. We remembered that this was likely an issue with

Cloudflare Web Analytics as we had seen in testing. Five minutes later, we took the action to disable Cloudflare Web Analytics from our website but did not purge the tracker cache. When we took this action, the stream we were running started dropping in viewership by a small but non-negligible amount. This drop ended at about 2:15 PM EDT as viewership began to increase.

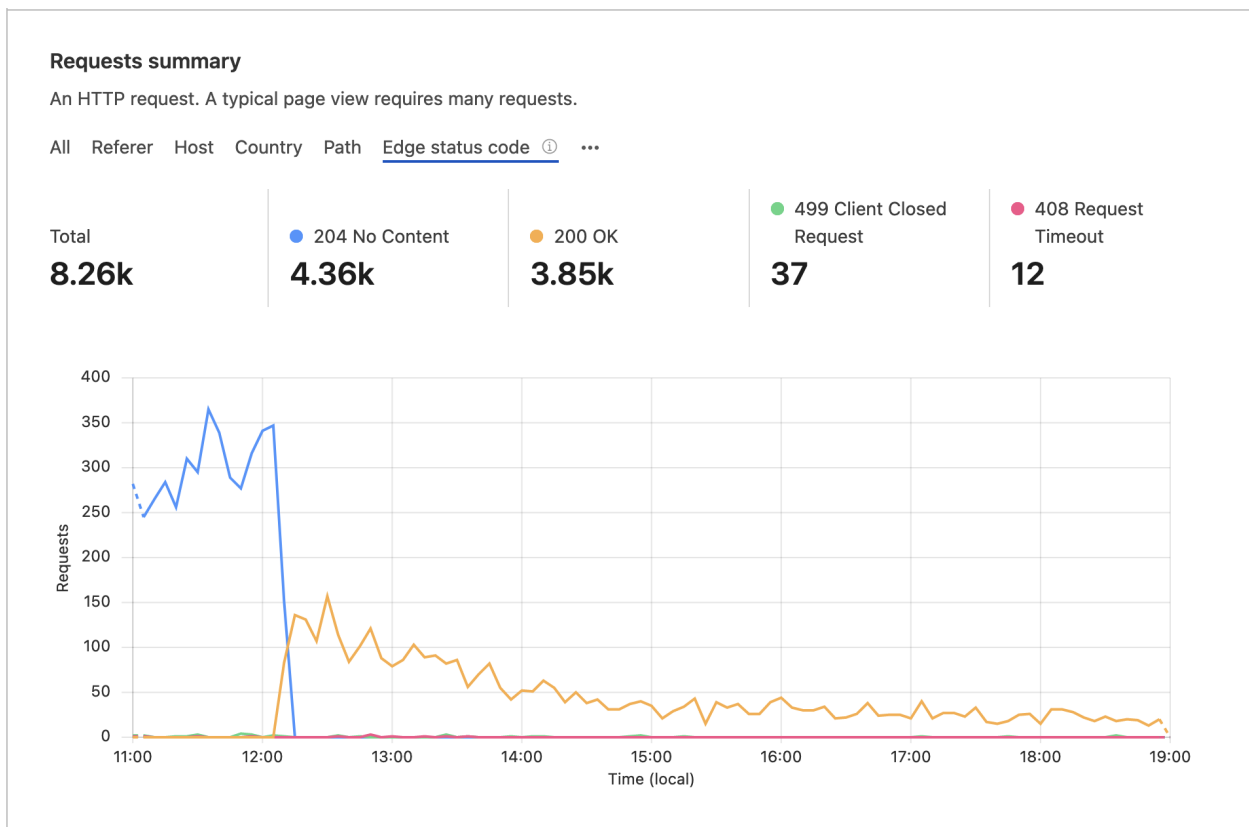
As we did earlier in testing, we generally took this as a one-time incident report as something just going wrong in production. Therefore, no further action was taken to fully resolve this issue in a timely manner.



(Figure 2: Graph of visits calculated by Cloudflare Analytics, showing the drop off at 12:15 PM EDT when we turned off analytics)



(Figure 3: Viewership drop on the eastmountainfilms Easter Bunny Tracker live stream from 12 PM to 2 PM EDT)



(Figure 4: Graph of the edge status code of the /cdn-cgi/rum endpoint. When web analytics was turned off, the status code went from 204 to 200. You can see a decreasing number of 200 OKs due to cached pages hitting real user monitoring)

Requests summary

An HTTP request. A typical page view requires many requests.

All Referrer Host Country Path Edge status code ⓘ ...

Total

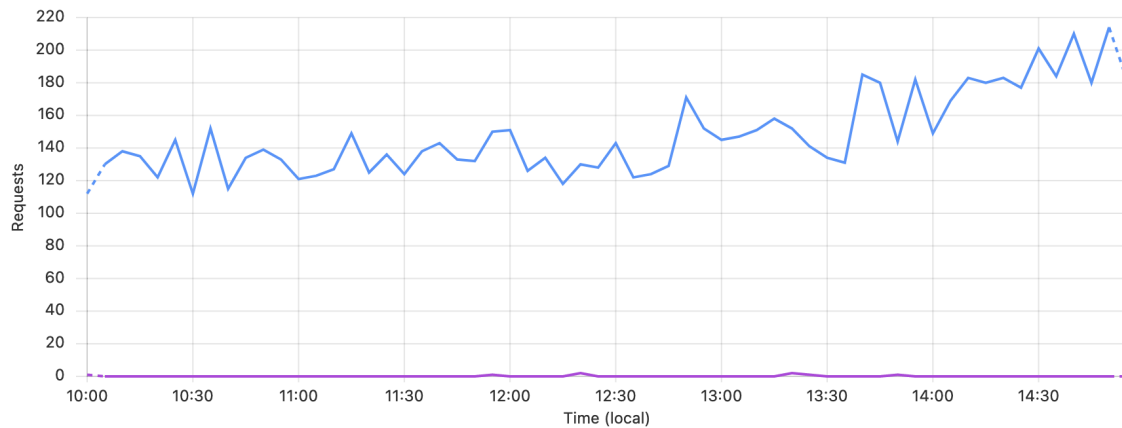
8.87k

● 200 OK

8.86k

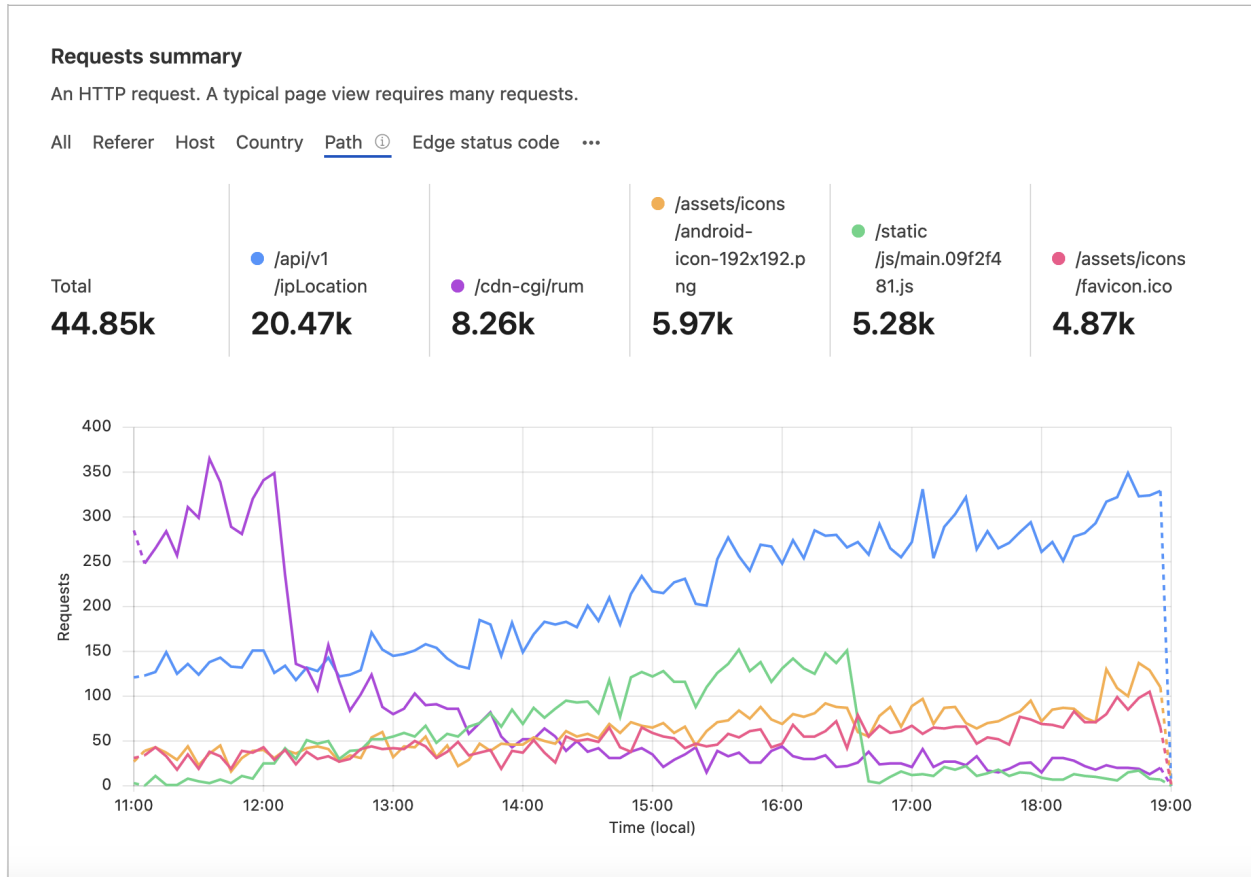
● 499 Client Closed Request

8



(Figure 5: An increase of 200 OK requests to the `/api/v1/ipLocation` endpoint to the Geo API, the best representation of tracker health in production from 10 AM EDT - 3 PM EDT on 4/8.)

At about 2 PM EDT, we noted that our stream was doing viewership numbers not at all expected, and across YouTube the combined total viewership of TEBCC streams was exceeding 600, about 3x that of the year before.



(Figure 6: Graph of /cdn-cgi/rum/ hits compared to /api/v1/ipLocation requests. When analytics was turned off, we generally saw an increase of ipLocation requests from 11 AM EDT - 7 PM EDT on 4/8)

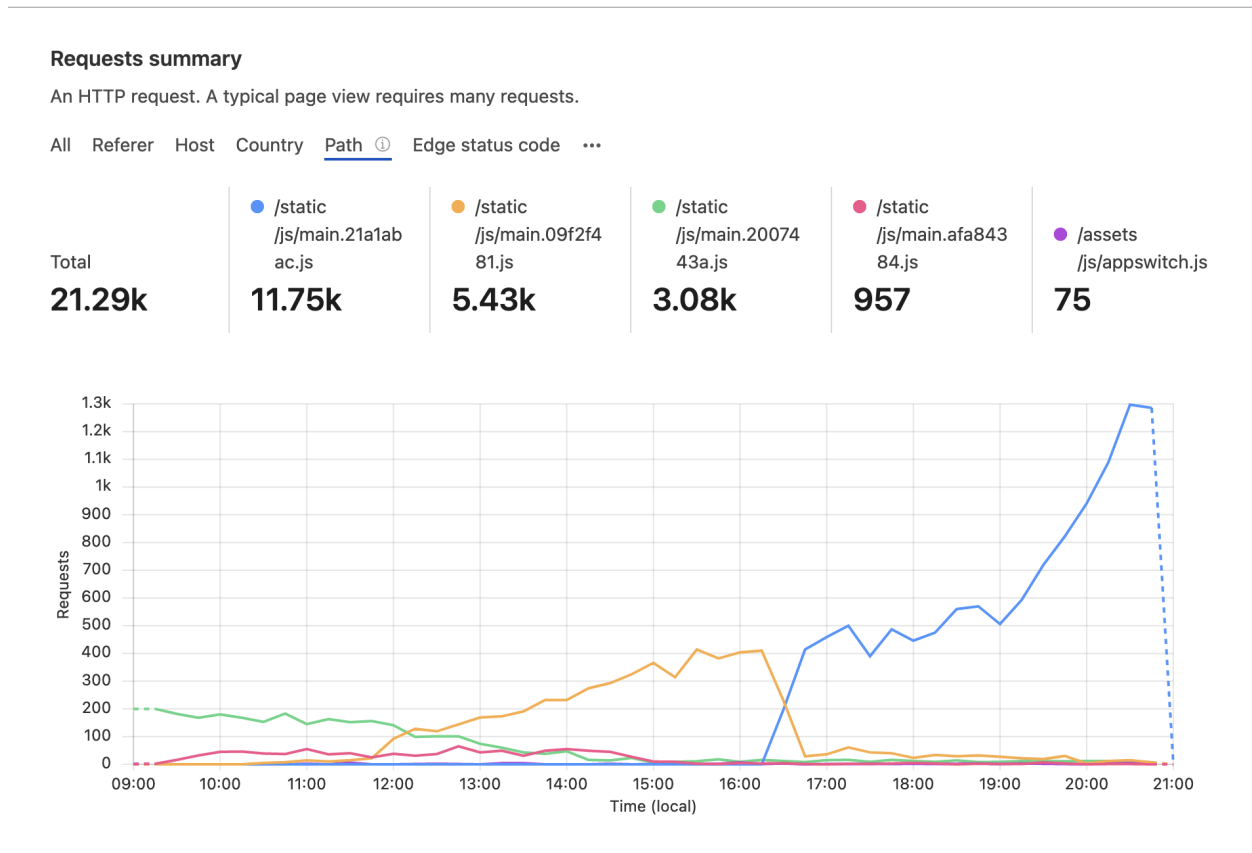
At about 4 PM EDT in our stream chat, we saw two more users report the exact same issue with the blank screen, both with the exact same resolution - incognito makes the tracker load again. We then discovered that there was a systematic issue with the tracker and began steps to further resolve things.

At about 4:37 PM EDT, we released a patch to the tracker where if the tracker failed to load, it would include instructions in the HTML about what to do. Most importantly, we purged Cloudflare's cache, meaning if you were to refresh the tracker you would get the latest tracker which did not include the beacon files.

At about 4:50 PM EDT, we ran a community poll on the live stream asking if users had issues loading the tracker. With about 68 votes, 70% of users voted No, while 30% of users voted Yes. It had become clear at this point that there was an issue with the tracker that was potentially ongoing.

At about 5:00 PM EDT, we further modified our page rules to disable the browser cache setting and rely on Cloudflare automatically configuring browser caching, rather than globally applying it to all our endpoints.

At about 5:15 PM EDT, after further community polls, we determined that there was an issue with the tracker failing to load in production about 25% of the time.



(Figure 7: Graph of the different .js bundles of the tracker across time, April 8, 2023)

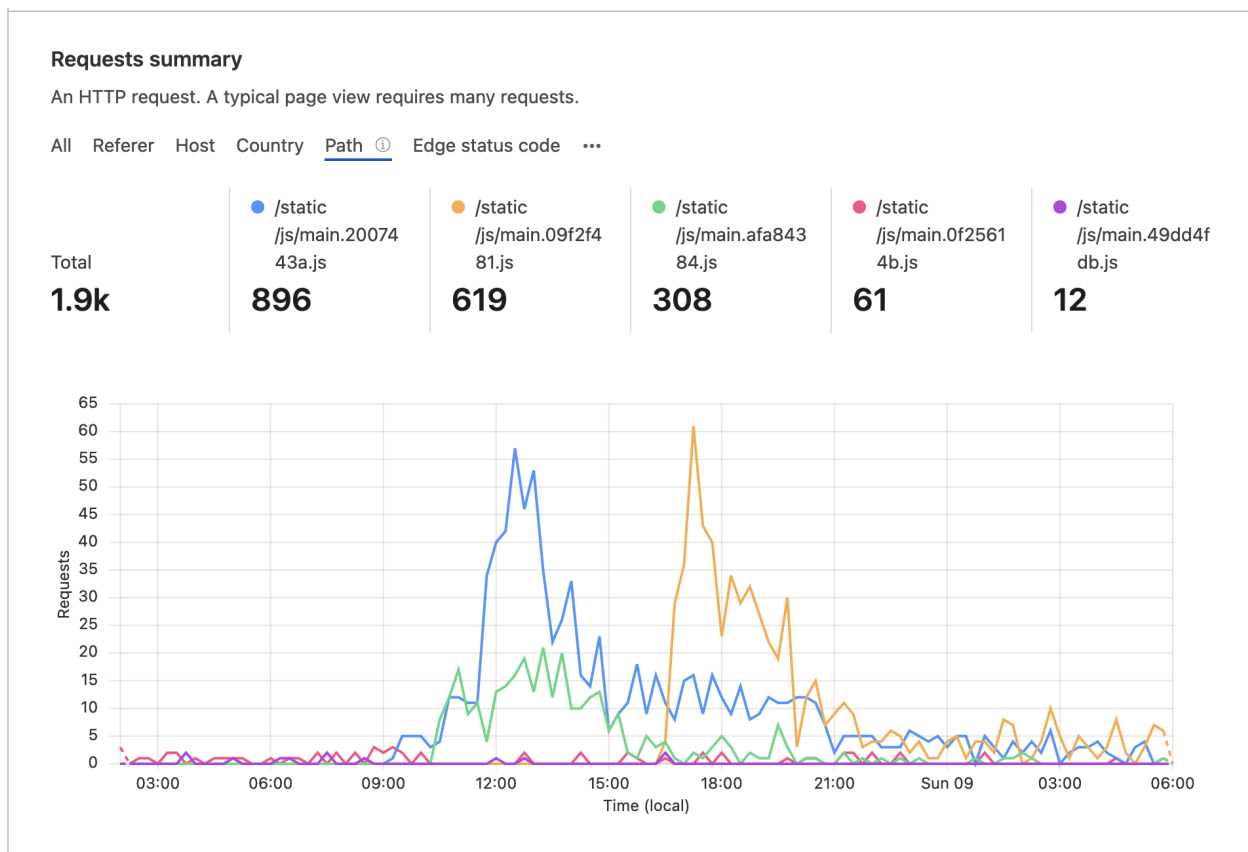
During the course of tracking, we released various different JS bundles, which can generally show how fast the caches were working. A guide to the bundles are:

- main.2007443a.js was built at 1:12 AM EDT on April 8 and was the bundle the tracker launched with.
- main.afa84384.js was built at 9:24 AM EDT on April 8 to change the frequency of the donate banner.
- main.09f2f481.js was built at 10:40 AM EDT on April 8 to change the frequency of the donate banner again.
- main.21a1abac.js was built at 4:37 PM EDT on April 8 with fixes to the HTML file. Even though we just fixed the HTML file, ultimately a new bundle was generated at that time because files were changed.

Curiously at noon, usage of the 09f2f481.js bundle increased dramatically, with the original 2007443a bundle decreasing slowly. The afa84384.js bundle saw some usage but very little usage.

This graph also shows the dramatic uptick in usage of the 21a1abac.js bundle as caches were purged and Cloudflare began to primarily serve this JS bundle with the modified HTML file.

Upon further investigation after the tracker ran, we found out that the tracker was seeing an abnormally high number of requests returning a 404 error code (not found). These requests were all for JavaScript files. Upon further analysis, we generated the graph below.



(Figure 8: Graph of 404 errors for JavaScript files throughout the course of tracking from 2 AM EDT 4/8 - 6 AM EDT 4/9.)

As you can see in Figure 7, when we had JavaScript file rollovers at 12 PM EDT and 5 PM EDT on 4/8, we generally saw an uptick in 404 errors due to the older JS files no longer existing due to the aggressive purging policy mentioned earlier in this report. Correlating with Figure 6 when JS file rollovers occurred, this meant that ~1,900 tracker instances during tracking could not load.

Additionally, the lack of 404 errors up until about 9 AM EDT when the first patches began to be released show that this issue largely began to happen because we were patching the tracker in

production. Thankfully, 404 errors began to generally reside by the time peak traffic was occurring.

After about 5:30 PM EDT, we generally considered this issue resolved. No additional patches were released during tracking. Hits were lower than expected through the rest of tracking. We believe this was the case as people were watching the tracker on YouTube, or went back to YouTube to get tracker updates (instead of visiting TEBCC due to prior load failures).

Root Causes:

The cause was determined to be the result of numerous, small issues cascading into one larger issue:

- We saw during tracker testing that Cloudflare Web Analytics would fail to load, but did not remedy the issue and did not take the issue seriously.
- We misconfigured the page rules for the tracker potentially making it so that users would get invalid Cloudflare beacons. These page rules were not thoroughly tested to ensure it would not have an impact on the tracker operation.
- We misconfigured the way Cloudflare Web Analytics was loaded into the tracker, partially the result of invalid page rules.
- We saw that the tracker had low hit counts from 9 AM EDT onward, but failed to connect this with the tracker regularly failing to load in production due to no user reports.
 - This could have been identified by installing third-party error monitoring software, but this type of issue would've likely not come up.
- We never included text in the tracker HTML file as a backup if the tracker never loaded, which should've been done.
- We ultimately caught the issue but it took nearly 15 hours from start to finish to resolve the issue fully with lingering side effects.
- We ran on a caching configuration from Version 5.6, which bundled all the JavaScript code in the HTML file rather than the JavaScript file.
 - This caching configuration ended up biting us in a bit of an unexpected way, only unearthed in this post-mortem.
- The release of patches in the tracker resulted in increased 404 errors due to our excessive JavaScript file purging policies.
- Throughout Version 6 development, we largely focused on tracker development but did not spend time testing our infrastructure, and largely left it alone from Version 5.6.
- We should have been monitoring our infrastructure beforehand, seeing if there were any issues when we released new versions of the tracker.
 - Early monitoring would have alerted us to 404 errors whenever a new tracker version was released, potentially letting us resolve this specific issue ahead of time.

Ultimately, the failures during Easter 2023 were largely on our infrastructure rather than the tracker.

Action Items:

For Version 7 of track.easterbunny.cc, we're taking the following actions to ensure that an infrastructure failure does not occur in production:

- We will be doing more strenuous testing on environments that mirror production for the tracker, including mirroring of page rules & caching rules.
- We will be reconfiguring our deployment rules so that the last 72 hours of JavaScript files are saved on the server.
- We will be taking a deeper dive into how the tracker actually gets deployed so we have additional knowledge into the inner workings of deployments.
 - The tracker was scaffolded using Create React App, and while this did reduce the time to get the tracker scaffolded, ultimately it led us to not understand the inner workings of how it builds the tracker.
- We will be reconfiguring our caching rules in a few ways:
 - We will be making it so that HTML files have a lower edge/browser cache TTL as the HTML file has a negligible amount of data transfer and can benefit from a more aggressive caching policy.
 - We will attempt to configure the tracker such that the likelihood of trackers loading an expired tracker page is lower.
- We will be reconfiguring how Cloudflare Web Analytics is loaded in so that it is not automatically injected, rather we manually put the script tag inside of the tracker.
- We will be installing third-party error monitoring software to catch any tracker issues that may occur.
 - We have largely been flying blind for the past 5 years with no way to monitor tracker health in production from the standpoint of JS failures that are the result of bugs in code.
- We will be switching to automated testing so that we can perform load testing on the tracker, and long-form testing in scenarios such as this one where patches are being released throughout the course of tracking.
 - Automated testing will also help us more easily test common failure cases without the need of manual testing, making the tracker less buggy.
- We will be increasing our monitoring efforts during tracking in an attempt to get a better inside look at how the tracker is being used in production
 - We debated having a system to calculate the number of users on the tracker at once for 2022 but scrapped it. However, such a system will be a higher priority for 2023.
 - We will start logging data on YouTube usage of the tracker, including combined concurrent viewers across streams. This additional data point will allow us to potentially see any production issues occurring due to increased viewership on YouTube.
- We will be purchasing Cloudflare Pro at least a month before tracking starts.
 - Generally speaking, we purchase Cloudflare Pro (which unlocks more detailed analytics) about a week before tracking starts then cancel it once tracking is over

for cost-saving measures, as we don't need the additional page rules and analytics when the tracker is not running.

- By purchasing Cloudflare Pro earlier, we will be able to get additional insights into potential tracker issues ahead of time so they can be patched before the tracker runs.

Successes from this incident:

Despite the incident, there were numerous successes that allowed us to more effectively find and resolve the issue:

- Cloudflare Pro Analytics continues to be an incredibly helpful resource allowing us to look more into how the tracker is doing in production, ultimately unearthing the 404 errors with JS bundles once tracking was complete.
- Our continued presence on YouTube allows us to have redundancy when the tracker does not load.
 - While it is not a redundancy we do not want to rely on, the quantity and variety of streamers on YouTube allowed users to continue tracking the Easter Bunny using our tracker.
- Ultimately, these issues were largely resolved before peak tracker traffic started at 7 PM EDT. Residual issues were still causing tracker failures in production, however we believe the failure rate was ~5-10%.

Conclusion:

Running, developing, and testing an Easter Bunny Tracker is an extremely challenging process from a technical standpoint. Trackers are subjected to an immense increase of traffic in a very short period of time, compounded by having to extensively test code that will only run for 28 hours in production once a year. The benefits of certain deployment strategies aren't available to us due to the unique traffic loads of the tracker.

Testing is made only harder by the fact that you can only do so much before tracking starts, and despite the usage of the word we throughout this document, the tracker is run by a single person, and I can only do so much for testing. It is impossible for myself to emulate the large loads put on the tracker by all our users.

Version 6 was a massive refactoring of the tracker completed in just 4 months into an entirely new language. What we did not realize was the refactoring also affected our infrastructure setup, and we did not take into account what the React-based tracker needed to run successfully on our infrastructure.

Ultimately, we failed tens of thousands of people who wanted to engage in the yearly tradition of Easter Bunny tracking, and for that, we are deeply sorry. We hope to regain your trust in following years, and we will work tirelessly to ensure an issue like this does not occur again.

This post-mortem is our first (and hopefully our last for a while), but an important step in realizing the failures in the tracker and the action steps to take for next year.

Respectfully,
The track.easterbunny.cc Team