Web Analytics (A2) Report

Avery Lee
Miguel Fernandez
Matthew Groover
Garrett Turner

September 18, 2023

Analytics Engine

Our team used Cloudflare Analytics as our official user-tracking software. Our website was already hosted through Cloudflare, so we added the domain to our 'Web Analytics' tab. We then added a script to our index.html page that contains a link to the Cloudflare Insights Beacon along with our personal access token. This allowed the software to track unique visitors to our website.

Our Visitor Counter

How we hosted/architected our visitor counter. What data we stored and in what format.

We hosted our visitor counter on Amazon Web Services (AWS) using their Lambda platform. We also store the visitor data locally within the Lambda functions. This data is stored as an integer, and every time we detect a unique visitor using local storage, we increment this integer. To ensure that visitors are unique, we utilize an API for browser fingerprinting called "fingerprints". This API derives a unique fingerprint for users based on various attributes such as browser version, User Agent, fonts, and several other unique pieces of information. If a user is determined not to be unique, we neither update the local storage nor add to the count.

Comparison

How our visitor counter performed compared to the analytics engine.

Our visitor counter recorded more visitors to our page than the analytics engine did on Cloudflare. This was in large part due to an error in an early version of our counter making it difficult to compare the counts directly as the Cloudflare count has never been reset. Both Cloudflare and our count appear to register users as unique when using incognito or private mode in the browser. Cloudflare analytics tend to be slow to update when directly compared to our internal measurement.

- Try testing with different devices, from different networks, and with incognito windows
- When does the external one work better?

  The external analytics work better when using google chrome. This is because google chrome will wipe local storage and our unique counter relies on local storage.
- When does yours look better?

When local storage is used we can check that it is the same user whether or not they have cookies enabled. This applies to the vast majority of browsers as most browsers do not wipe local storage even in extreme cases.

Appendix

The code we wrote for this project (include both client and server-side portions)

Client Slide

```
<!-- Importing jQuery -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>

<!-- AJAX call to get the unique users count and display it -->
<script>
    $.ajax({
        url: "https://aq1kk2fsu8.execute-api.us-east-2.amazonaws.com/default/test-lambda",
        dataType: "json",
        success:function(unique){
            populate(unique);
        }
    });
    function populate(unique) {
        document.getElementById('unique').innerHTML = unique;
    }
</script>
```

```
        if (localStorage.getItem('recordedVisit') != 1) {
            const visitorID = navigator.userAgent + Math.random(); // Simple method for generating a unique visitor ID.


            $.ajax({
                url: "https://aq1kk2fsu8.execute-api.us-east-2.amazonaws.com/default/test-lambda",
                method: "POST",
                dataType: "json",
                success:function(response){
                    localStorage.setItem('recordedVisit', 1);
                    console.log("")
                },
                error: function(err) {
                    console.error("Failed to record the visitor:", err);
                }
            });

            localStorage.setItem('recordedVisit', 1);

        }
    </script>
</body>
</html>
```

Server Side

```
let count = 0;

export const handler = async (event) => {
    if (event.httpMethod === 'POST') {
        // Increment the static variable on a POST request.
        count++;
    }
    else {
        const response = {
            statusCode: 200,
            body: JSON.stringify(count),
        };
        return response;
    }
}
```