Leaked Stack Traces

Security Incident Report – May 23, 2024

Summary

On May 15th, 2024 Sentry discovered we had published stack traces related to Sentry issue details of 15 customers to a public GitHub repository. These stack traces were available publicly from April 29th, 2024 to May 15th, 2024. In some instances, it was obvious these stack traces contained sensitive information, such as API keys.

We notified owners of the affected customer organizations via email on May 16th, 2024 with specific details about the issue data leaked to the public GitHub repository. We offered several ways to contact us directly, including email, Slack, and Discord in order to provide timely responses to any downstream incidents our customers may have experienced.

There was no compromise of Sentry infrastructure.

Incident Details

What was leaked?

The latest event details of particular issues were leaked to the public GitHub repository, `getsentry/seer`.

This included full stack trace details as well as local variable data that can sometimes include user IDs, request details including but not limited to HTTP headers, URIs, and query parameters, and other context-specific information to the application instrumented with the Sentry SDK.

The data was available in a JSON structure embedded in the output section of a Jupyter notebook. Impacted customers were provided with the specific event details exposed.

What are the impacts of the leaked data?

The impacts of this leaked data are highly context-specific to the customer's application.

There's increased risk of recording sensitive data when Sentry's data scrubbing features are not enabled or are not sufficient in their default state to detect and scrub sensitive values. For example, where a sensitive token is stored in a non-standard key name.

An analysis of all leaked stack traces uncovered strings appearing to be API keys and potentially other sensitive information. As a result, Sentry made the specific issues and stack trace information available to affected customers. This allowed customers to assess the risk of the leaked information under the context of their application and business.

How did we detect it?

The Sentry Security team performs regular reviews of our repository configurations to ensure security tooling is enabled, configured, and working as expected. We utilize GitHub's built-in security features including Dependabot, CodeQL, and Secret Scanning.

During a regular review, we discovered the Secret Scanning feature was disabled on several of our public repositories. This was remediated and as a result, GitHub Secrets Scanning occurred on all historical commits, issues, pull request comments, and so on for misconfigured repositories.

The scan detected a valid Mailgun API key belonging to Sentry in a commit on a feature branch. Our automated alerting issued an alert to the Security team to investigate and respond.

During the investigation customer stack traces were discovered in addition to our API key. We treated these stack traces with the highest priority and immediately escalated our internal incident response processes.

How likely is it that my data was copied?

The problematic commit (hash: 72d389c1d0f3017a6df9cb0a81e71dcd504edc92) containing customer data was pushed to the remote repository on April 29th, 2024 21:16 UTC.

On April 29th, 2024 21:45 UTC another commit was force-pushed to the remote repository. This effectively overwrote the commit history, making the problematic commit more difficult to discover.

In order to copy the exposed data, an actor must have had to clone the repository during the likely exposure window of April 29, 2024 from 21:16 to 21:45 UTC. Alternatively, the full commit hash (72d389c1d0f3017a6df9cb0a81e71dcd504edc92) would be required for a direct reference to the data exposed in the commit.

Sentry, however, has no way to know who may have cloned, viewed, forked, or redistributed the repository for the period of April 29, 2024 21:16 UTC to May 15, 2024 23:20 UTC.

Why were stack traces from my application in the leaked data?

Issue details, including stack traces, were incorrectly used from your application as part of a testing and validation data set. That testing and validation data set was subsequently mistakenly included in the Jupyter notebook committed to the repository. This happened because we failed to follow our internal data handling processes and procedures.

Response and Mitigation

Incident Response Steps

We're providing a summarized, long-form version below of the incident. You can find a detailed timeline of events at the end of this document.

Sentry has a robust Incident Management process, and the Sentry Security team declared an incident on May 15, 2024 and gathered the necessary engineering resources to respond to our leaked Mailgun API key committed to the repository.

It was during this response that we discovered customer stack traces also committed to the repository. Regardless of whether the stack traces contain sensitive data (ie. API tokens, session IDs, email addresses) – this was an unacceptable failure. The security incident captain escalated this incident, notifying our leadership and legal teams.

Our teams worked together to identify all affected customers and exactly what data was exposed. We made sure to gather all issue and event details necessary for customer notifications. We also put in some immediate mitigation steps to contain and control the incident (detailed in the next section).

On May 16th with confidence and verification of impacted customers, we convened our Security, Legal, and Engineering Engineering teams to validate a shared understanding of the incident impact. We also took this time to write customer notification communications with our Co-Founder and Chief Product Officer, David Cramer.

Customer notifications were our priority and David Cramer and our Head of Security made themselves readily available to all impacted customers once they were notified.

As the initial notifications were being delivered we continued our investigation, re-verifying the data exposure details and expanding our scope to other systems where this data may have been duplicated. Any systems related to the affected repository were investigated including, but not limited to, build systems, data processing applications, container repositories, and storage buckets. We found no evidence of the stack traces leaked elsewhere.

Containment and Mitigation Steps

During the investigation, our initial response included two measures:

1. We set the visibility of the repository to private.
2. We set up pre-commit hooks to remove any Jupyter notebook output

Setting the repository's visibility to *private* prevented clones and additional forks of the repository from that point forward. That said, we understand once something is on the internet – it is there forever. We continued to operate under the assumption the data had been copied.

We also configured pre-commit hooks to remove any outputs from Jupyter notebook files to mitigate future exposure of data while the investigation and response was continuing.

Lessons Learned and Future Measures

Lessons Learned

1. Our data handling processes and procedures are not as well understood or practiced as we had assumed.
2. We lacked technical controls in place to prevent leaking of customer data in our GitHub repositories.

Future Measures

1. Additional data handling training for Engineering, Product, and Design teams to enhance our top-down and bottom-up employee practice in safe data processing.
2. Implement stronger technical controls used to authorize access to customer data.
3. Develop additional technical controls to identify potential customer data on systems that shouldn't have customer data.
4. A full audit of our data processing pipelines and their adherence to our data handling and retention policies.

Timeline

All times listed are in UTC.

April 25, 2024

- **17:20** - A Sentry developer unintentionally committed customer stack traces in a Jupyter notebook on a locally cloned version of our `getsentry/seer` repository.

April 29, 2024

- **21:16** - The developer pushed their local changes containing the customer stack trace events to the remote repository on GitHub and opened a pull request (PR). This repository was set to public visibility.
- **21:45** - After an initial PR review, the author force-pushed additional commits, overwriting the commit containing the customer stack traces. This new commit did not contain any customer data.

- **22:03** - The feature branch of the PR was approved and merged into the default branch.

May 15, 2024

- **20:54** - During a routine check, the Sentry Security Team noticed the GitHub Secret Scanning feature was turned off on the `getsentry/seer` repository and enabled it.
- **21:28** - Sentry Security was notified via automated alerts of a leaked Mailgun API key belonging to Sentry in the `getsentry/seer` repository.
- **21:37** - An internal incident was reported to revoke and rotate the compromised API key.
- **22:32** - During the incident and the analysis of how/where the Mailgun API key was leaked, Sentry product-specific stack traces were discovered in the output sections of the Jupyter notebook.
- **22:55** - It was confirmed the Jupyter notebook also contained customer stack traces, not just Sentry product-specific ones.
- **23:20** - The `getsentry/seer` repository was moved from public visibility to private visibility to limit any further distribution of the repository (ie. via forks).
- **23:27** - The Sentry Legal team was notified of the ongoing incident to prepare for upcoming notifications to customers.
- **23:56** - We identified directly impacted customers including specific organization IDs, project IDs, and issue IDs related to the leaked stack trace data.

May 16, 2024

- **01:42** - After additional investigation and collection of customer information necessary for notifications the team agrees to continue the investigation in the morning and sets up a checkpoint call.
- **03:00** - Historical analysis and ongoing monitoring for potential abuse of the exposed Mailgun API key.
- **13:00**–**18:00** - Security and Engineering teams continued investigation and data verification
- **18:30** - Checkpoint call is held to discuss what we know and the necessary next steps, including collection of owner emails to send the notification to and validating all stack traces exposed and their associated issue link for each customer were correct.
- **20:00** - A second checkpoint call is held with the CPO, David Cramer, to fully brief him on the incident. The initial notification was drafted during the meeting.
- **22:16** - The first notifications are sent out to customers via email with link(s) to the issue(s) where stack trace information was exposed. The email included instructions on how to contact us if assistance was needed.

- **22:16** - The Security team and David Cramer defined alerts and continued to regularly monitor for customer replies to provide timely responses at any time of day.

May 17, 2024

- **20:12** - Additional due diligence is done to ensure the customer stack trace data is not leaked elsewhere (ie. container images created during CI builds, etc)

May 21, 2024

- **20:00** - An internal post-mortem is held to discuss the incident, timeline, what went well, what did not go well, and collect action items to prevent a future incident

From May 17, 2024 to May 22, 2024, we worked through customer replies to provide the necessary information to individually evaluate the risk of the leaked stack traces.

We also took the time to collect all notes of the various teams and people involved in the investigation to ensure our timeline was as detailed and accurate as possible in order to write to complete a full incident report.