Security Assessment by Trail of Bits
June 2020 - Actions taken

This document outlines the actions taken by Basecamp to address Trail of Bits team's manual findings in HEY encryption approach.

Summary

The main goal we had in mind when hiring Trail of Bits was to assess our at-work encryption approach for HEY before the launch, so we addressed the three high-severity findings uncovered by manual code review that concerned encryption and data validation. You can find these outlined in page 11 of the Trail of Bit report.

The mitigation for these three issues was put in place before launch. Later on, about nine months after HEY launch, we improved the mitigation for one of the issues and open-sourced our encryption approach, integrating it with Active Record in Rails (https://github.com/rails/rails/pull/41659).

Details on the actions taken

1. Deterministic encryption is flawed - 🔴 high-severity
We switched to a new approach where each encrypted value gets its own IV based on its contents. The IV is still generated in a deterministic way (HMAC-SHA256). We also removed any options in the public API to use fixed IVs.

2. Database storage is vulnerable to deserialization attacks - 🔴 high-severity
This issue was related to our usage of Marshal when serializing an encrypted payload, which we addressed before HEY launch for our particular usage and our fully encrypted data. However, for other apps that still had unencrypted data and started using this encryption approach, there was a possible exploit. We addressed this by switching to serialization with JSON before open-sourcing this, then re-encrypted all our data to use the new serialization method.

3 . Lack of guardrails to protect against the misuse of the encryption library - 🔴 high-severity
This was intended towards the future integration with Rails, rather than our own usage. We removed all options that prevented encryption or decryption errors from bubbling up. For example, the option to set a fixed initialization vector for a given attribute. The only option we kept was `support_unencrypted_data`, defaulting to false, to support transitioning to encryption for an existing app that already has unencrypted data.