[This blog post was originally posted at 11:03am PST March 8, 2023. Updated 3/8/23 3:11pm PST with detail from a technical incident report by Vinay Hiremath, Co-founder & CTO. Updated 3/10/23 7:35am PST with updated scope of impact.]

We care deeply about privacy and security at Loom. We know millions of you rely on our platform to record and share information, and protection of that information is our highest priority. Despite our numerous systems and processes in place to ensure the safety of your information, yesterday, **on March 7th 2023, some of our users had their information exposed to other user accounts**. **We will be in contact with all potentially impacted customers with more information.**

Summary of information we know so far:

The cause of this incident is detailed below.

- **Cause**: a configuration change to our Content Delivery Network (CDN) caused incorrect session cookies to be sent back to our users. Our authenticated sessions are stored in cookies, and each time a client hits our API, we extend the expiration of this session and send back a new cookie. The new configuration changes sent these cookie sessions to two routes on our CDN that cache responses. The combination of these side effects caused cached session extensions to be returned to the wrong users. We were alerted of the incident internally at 11:03 am (PST), identified the cause and attempted to mitigate at 11:10 am, and then decided to fully disable the app at 11:30 am until we could be certain of no further data leaks.
- **Result**: We know this configuration change caused incorrect sessions to be given to the wrong users, thus exposing a number of users' information to other accounts. We are actively evaluating the full, detailed impact and will have more to share. One hour and nine minutes after the configuration change was made (27 minutes after we were alerted) we disabled our app and rolled back our databases to remediate as quickly as possible. This action prioritized security; it caused a secondary impact of interrupting user recordings. **[Update 3/10/23]:** Due to the complexity of

identifying all users involved in an incident that changes user identity, we have done further analysis by working directly with our infrastructure data to identify that 0.18% of total workspaces *may* have been impacted. Our list of impacted workspaces may contain false positives.

Incident Timeline

The timetable for the incident is as follows (all times in PST). The period of the incident was from 10:21-11:30 am. The period of the downtime was 11:30 am-2:45 pm.

- *10:21 am* - Infra team makes a configuration change to update our CDN configuration.
- *11:03 am* - Incident is declared. Internally teammates begin to get logged out of loom.com and some teammates are reporting being logged into other accounts. Support becomes aware of similar issues affecting external users.
- *11:10 am* - Initial mitigation attempt. We revert the CDN configuration change and the team verifies there are no other reasonable causes outside of that configuration change that would cause this issue.
- *11:21 am* - Additional data provided. Support provides additional data reported by users seeing escalated account login issues.
- *11:30 am* - Manual downtime initiated. We fully disable the app at 11:30 am until we could be certain of no further information exposure. We determine the best course of action is to not go back up until we can ensure there is no further issue.
- *2:45 pm* - Service restored. We understand how the issue came to be, and we restore our service to a copy of our database and codebase as it existed before 10:15 am. All caches with user information (i.e., videos, user data, session information) are fully cleared.

Root Cause

Summary: During an update to our CDN configuration, we began sending session cookie headers to JS and CSS static endpoints that were served by our application behind our CDN. The application deserialized the session cookie, bumped it, and returned a `set-cookie` *response*

header which was cached by our CDN for 1 second. The users who requested that same asset within the 1 second time window would be served the initial user's session (which could be a different user) until the cache was cleared and a new user hit the cache.

Detailed explanation

Once we determined the CDN configuration change was the only update that correlated with the incident, we knew the CDN had changed its behavior and was serving incorrect session tokens based on support tickets.

At the time, what we knew was:

- The configuration changes against the CDN had been reviewed and validated by other infrastructure engineers.
- We tested these configuration changes on our dev, test, and staging environments for 10 days and had not seen any anomalous behavior.
- All caching behavior was the same. No caching on our API requests and only caching on JS and CSS assets.
- No code changes had gone out related to our authentication or session utilities.
- Our cookie-based sessions were rolling. That is, each time they are seen by our application, the expiration date would be moved into the future and a `set-cookie` response header will be sent.
- Our CDN was returning incorrect session tokens to the wrong users.
- There were a number of changes made to the CDN via the configuration change, namely:
    - We moved from deprecated AWS policies.
    - We upgraded to some new Terraform directives that were 1-for-1 with the old directives.
    - We passed on more headers.

Looking through the request logs, none of the expected API routes had their responses cached. It was still only the javascript and CSS static endpoints. We were able to trace through our code and deduce that:

- We had begun sending our session cookie headers to our JS and CSS static endpoints that were served by our application behind our CDN.
- When this happened, the application deserialized the session cookie, bumped it, and then returned a `set-cookie` response header which was then cached by our CDN for 1 second.
- The users who requested that same asset within that 1 second time window would be served the initial user's session (which could be a different user) until the cache was cleared again and a new user hit the cache.

Remediation

Going forward, we will be doing the following to remediate any potential of a similar leak:

- We will be ensuring our CDN always strips out the session cookie in response headers. We will also ensure it never passes on this cookie for non-API requests that could be cached.
- We will ensure the application does not return session cookies for any static assets it serves.
- We are updating our review policies accordingly to ensure we catch this type of issue with internal and staging test policies in the future. This includes testing load against CDN and API changes from multiple user accounts.
- We will be looking into enhancing our monitoring and alerting to help us catch abnormal session usage across accounts and services.

As the CTO, I want to reiterate our original update and apologize for this incident. This damages trust and I understand that. Trust is built over a long period of time between the customer and the company. We know we need to work to rebuild trust and we take that responsibility extremely seriously.

POSTED:

Mar 8, 2022

FEATURED IN:

Company News

SHARE THIS ARTICLE:

The Loom Team
The Loom Team

Ready to improve how your team communicates?

Built for every team no matter where they're located, Loom Enterprise boosts productivity with upgraded security and account support.

Unlock Loom Enterprise