Incident Aug. 31, 2021: App identifier in place of customer email at checkout

Availability: **Sales partially affected**

% of clients affected: **VTEX IO Stores using Minicart v1**

Duration of incident: **2 hours and 15 minutes**

Symptoms

Customers who accessed IO stores from 19h19 to 21h34 UTC had their shopping carts corrupted. They saw an application identifier on their email address field in the profile form when heading to checkout. Customers who placed orders with the incorrect email address didn't receive any confirmation and could not access such orders.

Summary

A new version of a VTEX IO application introduced a bug while authenticating customers with the checkout service on stores that used IO Minicart v1. Stores with front-end customizations may have also been affected. Customers that accessed stores during that time would see the email field on the checkout profile form filled by an application identifier; a value prefixed by `vrn--vtexsphinx--aws-us-east-1`.

It is important to highlight that no sensitive customer information was leaked during this incident due to the way our Smart Checkout is designed.

Customers could still place orders normally by removing the identifier and adding their email address. We could not correlate this incident with any decrease in placed orders.

A small fraction of customers placed orders with the wrong email address. These users did not receive any order confirmation and could not access these orders through "My Orders", even though they were charged.

We will provide each client with a list of affected orders and whether these were canceled or not.

Timeline

[2021-08-31 19:19 UTC] We deployed a new version of the `store-graphql` service.

[2021-08-31 21:12 UTC] We were notified about an issue with Checkout filling an incorrect email address on the user profile form and started investigating the issue.

[2021-08-31 21:34 UTC] We discovered that the `store-graphql` deployment caused the problem and rolled back its version.

[2021-09-01 02:40 UTC] We implemented a solution to clean up corrupted shopping carts during the incident.

Mitigation strategy

Once we identified the incident's root cause, we rolled back `store-graphql` to the previous well-functioning version. Gradually, the service returned to the correct behavior across all stores.

After the rollback, we cleaned shopping carts that were corrupted during the incident, preventing further damage. We also canceled refundable orders that were not handled yet.

Follow-up actions: preventing future failures

As a follow-up to this incident, we will improve our alarms to detect similar problems faster in the future and preventing them from happening in the first place.