Technical Analysis
Automating Data Privacy Requests

Table of Contents

Intro

I started looking into this thinking it would be a data project - moving data from one place to another using a script and the JIRA API.

But it's a more complex flow than I anticipated, and I think this work might belong with the Chorus team who are already familiar with the form and the request pipeline. At the very least, someone from the Chorus team will need to be involved - who and how much will depending on the route we decide on.

To clarify, this doc covers automating the *beginning* of the process, the step that will get incoming requests from users into JIRA without all the current manual effort. Automating the *entire* deletion process is a discussion for later.

What we're trying to solve for by automating the beginning of the deletion process includes:

- Automating the input into JIRA to eliminate the time spent on the manual task
- Standardizing/templatizing the requests in JIRA so that they're consistent and easier to follow
- It's possible that we could also reduce our Zendesk use/costs

Current Flow

1. We receive data privacy requests:
    a. through the same form that's used for other contact requests (GDPR/CCPA are just separate options) - https://www.sbnation.com/contact#donotsell
    b. unformatted requests also come in via email (vmprivacy@voxmedia.com) and go through a slightly different automated confirmation cycle
2. For all GDPR/CCPA requests:

a. the request payload is stored in redis with a one-hour expiry and a key that is shared with the user and can only be accessed once
b. a confirmation link is sent to the user's email address
c. on confirmation, the request payload is accessed in redis and posted to Zendesk (where a community support person confirms the info and then enters into JIRA manually)

Fields we're collecting in the form:

- Reason for contacting (either GDPR or CCPA)
- Request options: (Opt out, Erase, Export, Other)
- Additional details (freeform text)
- First and last name
- Email address
- Email address confirmation
- Chorus username (optional)

Fields we want in the JIRA template:

- Name
- Email address
- Chorus user id
- Chorus username
- Authored stories in Chorus (yes/no)
- Member of Chorus customer communities (yes/no)
- Data records (yes/no)
- Data deletion (yes/no)
- Report details ("Additional details" field from the form)
- Additional details (another freeform text field, this one entered by the community team)
- Deadline (calculated based on request receipt date - is there a different date range for GDPR vs. CCPA?)

Options

1. Give an external process access to the redis store? An hourly (or semi-hourly) collector job could ingest the entire redis store and post to the JIRA API to create tickets
    - Expiry on the redis store wouldn't have to change, but whatever mechanism is in place that removes each key after the user accesses it (through the confirmation step) would have to allow this process to access each key also

    OR

    - The process could make a copy of the redis store (on the same instance), parse keys and request payloads from the copy, post those payloads to JIRA, then destroy the redis copy once the process is complete.
2. Have an external job pull records from Zendesk and post them to JIRA? Setting up authentication on both sides would be a PITA, but both services have APIs that are able to manage these requests. We'd have to be careful about logging so that this process doesn't create additional user data.

3. Bypass Zendesk and have the confirmation step send to an email address instead, using email handler to post into JIRA
4. Bypass Zendesk and have the confirmation step post directly to JIRA instead, using the API to post into JIRA
    - I'm not sure what value we get from using Zendesk to route these requests - couldn't the requests just be managed directly in JIRA? On the other hand, since privacy requests and other contact uses share the same form, it might be too difficult to bypass Zendesk for privacy requests only. On the *other* other hand, this could be a model for discontinuing the use of Zendesk altogether. (I understand that it's a very expensive service - maybe we could ultimately route all of our contact requests directly into JIRA to be managed by the community team there?)
5. Automate the step between Zendesk and JIRA
    - I'm not even sure this can be done, but I'd be remiss if I didn't at least bring up the possibility. They both have APIs - can they be made to talk to each other? What would the authentication look like? What levels of permissions would the accounts on each side need?
6. Create a separate privacy request form, with more specific fields, and have that form submission post to JIRA
    - I'm not sure what level of effort would be required on the Chorus side - I'm guessing it would be trivial to move the CCPA/GDPR stuff into a new form, but then there would have to be a new link to this new form somewhere on the site ... and I'm not sure what else is involved. This is something for a Chorus developer to chime in on.

Reference

Our internal docs
Data privacy request workflow
User Deletion Request planning doc

Contact form
https://www.sbnation.com/contact#donotsell
https://github.com/voxmedia/sbn/pull/9863

Zendesk
API

JIRA
API
email handlers

auth/permissions
Privacy request board: https://vmproduct.atlassian.net/jira/software/projects/DPR/boards/176

Redis notes
https://redis.io/commands/keys
https://stackoverflow.com/questions/50533841/fastest-way-to-copy-35gb-of-redis-from-one-instance-to-another-in-azure-standar
https://www.digitalocean.com/community/tutorials/how-to-back-up-and-restore-your-redis-data-on-ubuntu-14-04