V3 JBX Deployment Bug Postmortem

Updated Feb 8, 2023 12:30 PM UTC-3

The deployment of the V3 JBX token contained a bug that caused excess reserved token issuance. This bug was caught before it could have been exploited, and mitigated by following the JBX Emergency Procedure.

Severity: High

Status: Mitigated

Summary

On January 26, 2023 at 1:19 AM UTC-3, Jango realized there were 2,086,888,439.481608634942732443 reserved JBX mintable from the JuiceboxDAO V3 treasury. This was following the deployment of V3 JBX and its attachment to the JuiceboxDAO V3 treasury several hours prior.

The suggested course of action was for the JuiceboxDAO multisig to exercise its JBX Emergency Procedure duties to set the reserved token allocation to go entirely to the dao.jbx.eth wallet, distribute the excess reserved tokens, then burn the excess tokens.

The transaction to set the reserved token distribution can be found here: https://etherscan.io/tx/0x4a0e8e84cce8674a28255b038cf1f82e6de235543e3e90c6f950bbfdd2d27f18

The transaction to burn the excess reserved tokens distributed to dao.jbx.eth can be found here: https://etherscan.io/tx/0x57ba0b44d57de2a4a19a45917a8245c8d02e999f1e843087665039186ac89460.

How you'll be effected

If you're a part of a Juicebox project, you are unaffected. If your project is running on V3, you'll be recommended to migrate your project's `controller` to an updated version that makes reserved token distributions safer.

If you're a DAO contributor, you won't get reserved tokens for at least another 2 weeks, and likely 4 weeks (two governance cycles).

If you are a JBX holder, the timeline to have open redemptions against the V3 treasury will likely be extended up to 4 weeks (two governance cycles).

If you're a DAO multisig member, you'll likely be responsible for queueing, reviewing, and executing transactions to burn excess reserved tokens that accumulate to the V3 treasury in consequence to V1 and V2 issuance.

Description

V3 JBX takes into account the V1 and V2 token balances when calculating its total supply. The V3 JBController (0xFFdD70C318915879d5192e8a0dcbFcB0285b3C98) uses a project token's total supply, along with a tracking property, to calculate the current outstanding reserved token issuance. The tracking property was not updated before the V3 token was issued, as it is an internal property only updatable by distributing the currently outstanding reserved token supply. As such, the V3 JBX token's deployment design was flawed. This was not caught in the test suites, internal reviews, or external audits.

Detection

The bug was discovered and reported by Jango to the JuiceboxDAO telegram chat at January 26, 2023 at 1:19 AM UTC-3, 14 hours and 11 minutes after the JBX V3 token was deployed by the JuiceboxDAO multisig and attached to its V3 treasury in this transaction. The bug was discovered by looking at the JuiceboxDAO V3 treasury page at juicebox.money/@juicebox and noticing a suspicious amount of reserved tokens distributable. At the time the bug was discovered, the reserved token distribution was routed to the regular list of several accounts as determined by DAO proposals. If the excess reserved tokens would have been distributed at that time, each recipient would have received JBX unintended for them.

Jango immediately realized the cause of the problem and made note of it in the original message reporting the bug.

The bug would have had a greater likelihood of detection prior to the deployment the V3 JBX token if JuiceboxDAO had an organized checklist of possible functions that could be effected by newly introduced components.

Response

Unless explicitly locked, the reserved token distribution within a funding cycle's reserved rate is changeable by the project's owner. Since the recipients of the original distribution were not locked, the mitigation strategy proposed by Jango and agreed upon through the JBX Emergency Procedure was to set the reserved token allocation to be routed entirely to the multisig at dao.jbx.eth. This had to be done before anyone sent the public transaction to distribute outstanding reserved tokens to the current list of recipients. The multisig would then burn this excess supply in a subsequent transaction.

Over the next hour, members of the multisig were contacted to review the circumstance and proposal, and admins of the juicebox.money website to disable its facilitation of the transaction to distribute reserved tokens to the JuiceboxDAO reserved token allocations.

The transaction to set the reserved list allocations as 100% to dao.jbx.eth was queued by jango.eth at Jan 26, 2023 at 1:51:43 AM UTC-3, and executed by peri.eth at Jan 26, 2023 at 2:48:11 AM here https://etherscan.io/tx/0x4a0e8e84cce8674a28255b038cf1f82e6de235543e3e90c6f950bbfdd2d27f18.

The public transaction to distribute the excess reserved tokens to the reserved token list was executed by aeolian.eth on Jan 26, 2023 at 2:58:59 AM UTC-3 here https://etherscan.io/tx/0x4a0e8e84cce8674a28255b038cf1f82e6de235543e3e90c6f950bbfdd2d27f18.

The transaction to burn the distributed supply from dao.jbx.eth was queued by jango.eth at Jan 26, 2023 at 3:11:36 AM UTC-3, and executed by jbx.filipv.eth at Jan 26, 2023 at 5:58:23 AM UTC-3 here https://etherscan.io/tx/0x57ba0b44d57de2a4a19a45917a8245c8d02e999f1e843087665039186ac89460.

Recovery

After the effective response by the multisig members, there is no imminent risk to the treasury or assets to be recovered.

There are, however, steps the DAO should take in order to restore the expected reserved list allocations to its V3 treasury. As it currently stands, newly issued JuiceboxDAO V1 and V2 treasury tokens will continue adding to V3 JBX's total supply, thus continuing to create unwarranted reserved token issuance that should be burned accordingly by the multisig until these steps are completed. These steps can be completed in full through a series of funding cycle configurations and an optional controller migration at the next available opportunity.

The principle goal is to prevent unexpected reserved token issuance from JuiceboxDAO's V3 treasury, there are 3 known ways to achieve this each with varying tradeoffs.

1. Stop issuing treasury tokens from the DAO's V1 and V2 treasuries. The tradeoff is we would no longer be managing interoperable fee-collecting treasuries.
2. Stop issuing from the DAO's V1 treasury, but keep the V2 treasury open by reconfiguring its reserved rate to 0 and reducing issuance rate. This is possible since V2 treasuries allow owners to explicitly set a custom token issuance rate whereas in V1 the only way to affect issuance is through a discount rate capped at 20%.
3. Keep both the DAO's V1 and V2 treasuries open, and instead fix the problem by moving the DAO's V3 controller to an upgraded version with an improved and sturdier accounting method for reserved tokens. The tradeoff is this would require prioritized client support to begin facilitating the new V3.1 controller, and it would be introducing another component that needs review before deployment. The v3.1 controller is something we will likely want in the near future as a learning from this postmortem, but not necessarily an immediate need.

Options 2 and 3 are mutually exclusive. If we go with option 2, we'll have to reasses V2 treasury strategy alongside a future V3.1 JBController consideration.

Below is a game plan if we were to go with option 2, which A) adjusts the Juicebox DAO V2 treasury's token issuance, and B) stops issuance from JuiceboxDAO's V1 treasury altogether:

Option 2

A.

1. A funding cycle should be scheduled to move the V2 treasury's reserved rate to 0%. The V2 treasury's issuance rate should be adjusted accordingly such that it continues issuing the same amount of tokens outwardly to payers as before. JuiceboxDAO reserved rate recipients will still receive their V2 allocations, but will get it directly on V3.

B)

1. V1 issuance cannot be set directly, only through discount rates. Because of this, V1 issuance needs to be paused altogether. In order to do this, JuiceboxDAO must migrate its V1 payment terminal to V1.1 which respects a funding cycle metadata flag instructing for payments to be paused.
2. A funding cycle should be scheduled with the metadata flag with pause payments turned on.
3. JuiceboxDAO should submit transactions to move the V1 payment terminal fee to 0% and the V1.1 payment terminal fee to 0%. This is needed in order to continue allowing V1 projects to distribute funds, since a fee incurred to a treasury with paused payments would revert.

Alternatively, below is a game plan if we were to go with option 3:

Option 3

1. A funding cycle should be scheduled that allows controller migration.

2. JuiceboxDAO shouls submit a transaction to migrate its controller from V3 to V3.1.

   Other projects using the V3.1 controller would also be safe to use the same token deployer that caused this bug in the first place, and future reserved rate risks would be mitigated protocol-wide due to a more-explicit storing of data.

   More abstractly, this V3.1 controller re-examines the reserved token calculation design. The current design prioritizes the cheapest possible cost to pay a project, the tradeoff being a reserved token issuance calculation derived from outstanding token supply and an internal tracking property updated only when currently reserved tokens are distributed. The cost savings from payments may not justify a more-explicitly tracked reserved token supply calculation that exposes fewer dependencies, and the savings aren't that clear to begin with. A V3.1 controller is likely in our future anyways, the question is moreso if we want to prioritize it now to also leverage it to address our immediate needs caused by the bug.

Once either option 2 or option 3 steps have been completed, the expected V3 treasury reserved rate distribution can be restored.

Actionable lessons learned

1. It's clear we need a stricter checklist with cross-protocol mechanism considerations when approving the deployment of new components. We must better reflect on how a new component might introduce conflicts to core mechanisms when OKing code. This needs to be integrated in a exhaustive and formalized "DevSecOps", part of our devops model we perhaps haven't been paying attention to.
2. We should re-evaluate the tradeoffs made in payment terminals and controllers between high-frequency transaction costs with low-probability unexpected behavior exposure. Documentation isn't always enough to communicate risks, sometimes it's better to make the tradeoff of more-expensive day-to-day operations to prevent risks altogether that could have destabilizing effects. There's no universal answer, it always depends on the circumstances – we first need to quantify these risks for our specific contexts.
3. Those thinking about V3 JBX implementation, review, and testing should not have been simultaneously encouraged to entertain other more surface ideas and needs. The V3 JBX contract and concept was simple enough, so we casually leaned on each other's reviews and external audits as we built other products and supported other happenings around the DAO's ecosystem. When pursuing multiple coding ventures displaying an apparent simplicity ("peripherals" or "extensions" contracts), we should keep in mind the highly adversarial environment we are evolving in. A flaw exposing a vulnerability, even in an extremely remote contract might, for instance, lead to an infinite mint bug and, eventually, loss of funds. We need to stay as paranoid as we were for V3 "core" contracts, without any exceptions. Being laser-focused on less tasks is therefore a prerequisite, slower beats rekt.
4. The DAO's engineering tendencies lately have devolved towards co-responsibility over code with no clear ownership, while encouraging contributors to have their hands in many things at once. There's immense value in having many reviewers to a code base, but through our actions we should re-balance the culture to create space for each person to get lost in a specific set of projects at a time, and to be clear about the individuals that have ownership over the outcomes of each prioritized project or experiment. There are times when it makes sense to encourage towards entertaining the current latest idea, but there are others where we need to encourage total focus on our outstanding commitments.
5. Interoperability of treasuries can expose risks that aren't worth the po-

tential opportunities. If we would have decided to stop issuance from the V1 and V2 treasuries entirely before the introduction of V3 JBX, we would not have had to introduce components facilitating token migration and this specific bug would not have happened. When designing the V3 JBX deployment strategy, the choice was made to dogfood interoperability despite the relatively small scale distribution of V1 and V2 in order to rehearse a potential future necessary and higher-risk need for interoperability into a new protocol version.

Update Feb 8, 2023

- JuiceboxDAO passed JBP-335, which allowed for the migration of the DAO's controller to version 3.0.1 eth:0xA139D37275d1fF7275e6F33821898934Bc8Cb7B6. This transaction was confirmed by JuiceboxDAO multisig on Feb 8, 2023 at 7:55 AM UTC in transaction with hash 0xfa8f774e572e805d788f333a1a0e670aa1d7d5bf6cc7b4484feab45