Liquity ChickenBond

Smart Contract Audit

coinspect

Liquity

ChickenBond - 3rd Audit

Smart Contract Audit

V220929 Prepared for Liquity • September 2022

1. Executive Summary

In **September 2022, Liquity** engaged Coinspect to perform a source code review of **ChickenBond**, their *autonomous self-bootstrapping treasury system*. The objective of this **third** audit was to evaluate the security of the changes performed to the project's smart contracts since Coinspect's second audit.

Most modifications reviewed during this engagement are related to the NFT data and generation feature and do not affect the economic model or security features of the protocol.

Coinspect concluded that the security of the project was not affected by the changes reviewed during this third engagement.

Coinspect performed a total of three audits of the ChickenBonds project, which include all code that is intended to be deployed. As a result, all issues reported have been addressed.

The following issues were identified during the initial assessment:

High Risk	Medium Risk	Low Risk
0	0	1
Fixed 0	Fixed 0	Fixed 1

The only low-risk issue identified during this assessment allowed attackers to block users from creating bonds when relying on the new permit based interface. Coinspect verified the Liquity's fix for this issue is correct.

2. Assessment and Scope

The audit started on **Sep 12 2022** and was conducted on the **main** branch of the git repository at https://github.com/liquity/ChickenBond. The last commit that was reviewed was commit **`1aec0deea30233c116200e4dcb295b019f3ecb3e`** of **Sep 26 2022**.

The audited files have the following sha256sum hash:

```
817caf78a240c96212dde377442fe03233b03d1d87acc359c394581296d6d474  ./ChickenBondManager.sol
1d76a3ff36d05c1471fbaec949104e6ae721c2c5c4e35d6fb624f0339e8d1fb3  ./BondNFT.sol
6ade1a46b823ce6694814690bed72bb34d17ea375baea3b807799b539d58d7b6  ./utils/ChickenMath.sol
21d2c25d5296d74543316bdc6d58112c4e63ed29c09c8271430259bf75ef9c62  ./utils/BaseMath.sol
1edd3b6e7fa7c3d4a379ac10f0e09f60722380317bb0001a998e29e6595b66f0  ./Proxy/ChickenBondOperationsScript.sol
7048079ec375f3f754997df6219aab3ae8db909d0d3590f4c4fc4a3be74c70fc  ./Interfaces/IBondNFT.sol
61d3ac03318e4a2b229bdf1ebea44726fc6ee3953bd083b707f6766186dc93ac  ./Interfaces/IChickenBondManager.sol
a46199de329976691b02dcecd3ace2de7c1a00705fb2ae96043626ca9a7c89be  ./Interfaces/IYearnVault.sol
3d6f0d974c7e2a677fb4eac8b73ef97d9885af7c9f0a2c1bee87003830d1d678  ./Interfaces/ICurveGaugeController.sol
74af7edb11581c9db3bdcea9ad26daf0df30b1c9f1d9c4aaaf42a6a3f4f560c7  ./Interfaces/IBLUSDToken.sol
a161f869c8f6127d401498a94cfb7dda44383ecefc14a3d6a574e16f641de889  ./Interfaces/StrategyAPI.sol
1259edc75b57d2795e09aa3d7f57c4aa10110b219db411235544972f9b3f8940  ./Interfaces/IBondNFTArtwork.sol
dc7dbaa5ed2d12d652f9fd292d8780210a4e9b4ab6b80a63b96d6216d50675f3  ./Interfaces/ITroveManager.sol
82838b15e39fa5a18ad5f50ce6f1c97d2c84a41faa52242901f6599f0d4e4617  ./Interfaces/ILQTYStaking.sol
11e67ace43558e198a640fa751f08eda20acd80223bc335d39b1f5a778d7d005  ./Interfaces/ILUSDToken.sol
d558a107f29ee21e92442bc670041a6d1815ccb9bd9cfcb26707c1aff49034f1  ./Interfaces/ICurveLiquidityGaugeV5.sol
ed0149693659bcc489b84afbd285c2ec955b4555ace335cb9ac6a32c749983ae  ./Interfaces/IPickleJar.sol
6fbc774158264ae2100bff94baa628554a1c243bed2cca74a831a5bce2d4d0b6  ./Interfaces/ICurvePool.sol
fd668d98f76008276d71245f0d3956f94fcf0b96e82e639383afb197fa77f7d6  ./Interfaces/IBAMM.sol
a8e89966466bdb949bc3ffd7368edfb901cbefdb6dd19ced57c1d414e6ec2cd1  ./Interfaces/IYearnRegistry.sol
a39415e85a140eb3650f2fc38a247f8d9efd6446be3bb89a573f7bd664498ff5  ./BLUSDToken.sol
79e55dfd16c24da2b2354b4f1e51a1b6b1f8a2ce2e0034bb3c03e8bd2f3d5275  ./NFTArtwork/SimpleEggArtwork.sol
46664b1aab7cc7f99bea867215f0e21d8be266875bfce974301727fa3a403eed  ./NFTArtwork/GenerativeEggArtwork.sol
18392f088baa69bfb0b57b66017c13b05e9333d527691ad1d772d6fe65d99d95  ./NFTArtwork/EggTraitWeights.sol
```

This audit focused on the changes introduced by the following pull requests as requested by the client:

- https://github.com/liquity/ChickenBond/pull/161
- https://github.com/liquity/ChickenBond/pull/162
- https://github.com/liquity/ChickenBond/pull/164
- https://github.com/liquity/ChickenBond/pull/170
- https://github.com/liquity/ChickenBond/pull/184
- https://github.com/liquity/ChickenBond/pull/190
- https://github.com/liquity/ChickenBond/pull/195
- https://github.com/liquity/ChickenBond/pull/205
- https://github.com/liquity/ChickenBond/pull/196
- https://github.com/liquity/ChickenBond/pull/199
- https://github.com/liquity/ChickenBond/pull/167
- https://github.com/liquity/ChickenBond/pull/182

- https://github.com/liquity/ChickenBond/pull/206
- https://github.com/liquity/ChickenBond/pull/201
- https://github.com/liquity/ChickenBond/pull/202
- https://github.com/liquity/ChickenBond/pull/207
- https://github.com/liquity/ChickenBond/pull/219

These pull requests implement several changes related to: the protocol data exposed in the protocol generated NFTs, these NFT's SVG artwork generation, new tests related to the funds shift functionality, new events emission, source code documentation improvements, addition of bond counters intended to improve the user interface, and new permit-based bond creation interface.

Coinspect identified a weakness related to the new permit-based bond creation feature, that enables attackers to block users from creating bonds as detailed in issue **LCB-10**.

3. Summary of Findings

Id	Title	Total Risk	Fixed
LCB-10	Bond creation with permit DoS	Low	✔

4. Detailed Findings

LCB-10 Bond creation with permit DoS

Total Risk	Impact	Location
Low	Low	`ChickenBondManager.sol`
Fixed ✔	Likelihood Low	

Description

Attackers can block users from creating new bonds.

The `createBondWithPermit` function enables users to create new bonds through the LUSD permit feature. This allows creating bonds with only 1 transaction, instead of requiring first an `approve` call before the `createBond` transaction.

```
function createBondWithPermit(
    address owner,
    uint256 amount,
    uint256 deadline,
    uint8 v,
    bytes32 r,
    bytes32 s
) external returns (uint256) {
    lusdToken.permit(owner, address(this), amount, deadline, v, r, s);
    return createBond(amount);
}
```

Attackers wanting to disrupt ChickenBond users can extract the permit parameters from a call to `createBondWithPermit` and then call LUSD's permit themselves, before (front-running) the original `createBondWithPermit` call, which will revert as the nonce has been already used.
Users would be still able to create bonds by using the previous interface and 2 transactions: `approve` and `CreateBond`. **However, if users rely on a front-end that**

does not provide them with this option, they will be effectively prevented from creating bonds.

Attackers willing to harm users and the protocol must pay the transaction cost for submitting the permit transaction for each bond creation attempt they block.

Recommendation

Consider ignoring the LUSD permit revert in `createBondWithPermit` function. Alternatively, ignore the permit if the user already has enough allowance.

Status

Fixed.

Coinspect verified the issue was properly addressed by pull request 211:
https://github.com/liquity/ChickenBond/pull/211/

5. Disclaimer

The information presented in this document is provided "as is" and without warranty. The present security audit does not cover any off-chain systems or frontends that communicate with the contracts, nor the general operational security of the organization that developed the code.