SB SECURITY

FeliX

Security Review

Nov 17, 2024

Conducted by:
Blckhv, Lead Security Researcher
Slavcheww, Lead Security Researcher

Contents

1. About SBSecurity

SBSecurity is a duo of skilled smart contract security researchers. Based on the audits conducted and numerous vulnerabilities reported, we strive to provide the absolute best security service and client satisfaction. While it's understood that 100% security and bug-free code cannot be guaranteed by anyone, we are committed to giving our utmost to provide the best possible outcome for you and your product.

Book a Security Review with us at sbsecurity.net or reach out on Twitter @Slavcheww.

2. Disclaimer

A smart contract security review can only show the presence of vulnerabilities **but not their absence**. Audits are a time, resource, and expertise-bound effort where skilled technicians evaluate the codebase and their dependencies using various techniques to find as many flaws as possible and suggest security-related improvements. We as a company stand behind our brand and the level of service that is provided but also recommend subsequent security reviews, on-chain monitoring, and high whitehat incentivization.

3. Risk classification

	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

3.1. Impact

- **High** - leads to a significant loss of assets in the protocol or significantly harms a group of users.
- **Medium** - leads to a moderate loss of assets in the protocol or some disruption of the protocol's functionality.
- **Low** - funds are not at risk.

3.2. Likelihood

- **High** - almost **certain** to happen, easy to perform, or highly incentivized.
- **Medium** - only **conditionally possible**, but still relatively likely.
- **Low** - requires specific state or **little-to-no incentive**.

3.3. Action required for severity levels

- High - **Must** fix (before deployment if not already deployed).
- Medium - **Should** fix.
- Low - **Could** fix.

4. Executive Summary

Overview

Project	FeliX
Repository	Private
Commit Hash	-
Resolution	-
Timeline	November 10 - November 17, 2024

Scope

/contracts

Issues Found

Critical Risk	1
High Risk	1
Medium Risk	4
Low/Info Risk	2

5. Findings

5.1. Critical severity

5.1.1. Blackbox distribution ranges are miscalculated

Severity: Critical Risk

Description: Ranges that are used for the `BlackBox` allocation assume `TitanX` is with less than 9 decimals. For example, the biggest range should happen when there is 1M - 10M TitanX in $$$ value, but in the code, we have 1.2e12 - 12e12, which is less than a dollar.

As a result in the first draw, if that goes unnoticed participants will expect 5-6 figure rewards but will end up with < $1 due to the miscalculated ranges.

```
{
  // Adding ranges with associated distribution patterns
  addRange(1.5e9, 3e9 - 1, 0); // Uses distributionId 0
  addRange(3e9,   4.5e9 - 1, 1); // Uses distributionId 1
  addRange(4.5e9, 6e9 - 1, 2); // Uses distributionId 2
  addRange(6e9,   15e9 - 1, 3); // Uses distributionId 3
  addRange(15e9,  30e9 - 1, 4); // Uses distributionId 4
  addRange(30e9,  40e9 - 1, 5); // Uses distributionId 5
  addRange(40e9,  50e9 - 1, 6); // Uses distributionId 6
  addRange(50e9,  120e9 - 1, 7); // Uses distributionId 7
  addRange(120e9, 900e9 - 1, 8); // Uses distributionId 8
  addRange(900e9, 1.2e12 - 1, 9); // Uses distributionId 9
  addRange(1.2e12, 12e12, 10); // Uses distributionId 10
}
```

Recommendation: Scale the decimals of all the ranges, so we are aligned with the docs.

Resolution: Fixed

5.2. High severity

5.2.1. LP creation can be blocked with pool donations

Severity: High Risk

Description: Initial LP for the `FeliX/TitanX` can be permanently blocked by donating at least 1 wei `TitanX`. addLiquidity will first check if this pool exists and if not, will create it. Although anyone can create a pool before the FeliX code, just creating it is not a problem, but if he creates it and donates `TitanX`, then `createAndFundLPs` will always **revert**.

This can be achieved by anyone donating the other token to the pair and then calling UniswapV2Pair.sync(), this will update the other token's reserve. Then when `addLiquidity()` is called, since the tokens reserves are not 0, it will enter the else here and then revert in UniswapV2Library.quote.

Recommendation: Before calling `addLiquidity()`, get the token reserves and if there is `TitanX`, mint `FeliX`, donate it to the pair and call `sync()`, making sure the reserves are the same before adding liquidity.

Resolution: Fixed

5.3. Medium severity

5.3.1. PresaleMinter NFTs can be transferred between VRF request fulfilment and winner selection

Severity: Medium Risk

Description: Minter NFTs give an additional 20% to their holder if he is selected as a `BlackBox` winner. The attack that can be pulled is the following:

1. The request is fulfilled, and **random words** can be observed from the transaction.

2. NFT holder saw them and reverse-engineered the formula used to pick the winners.

3. He saw that one of his other addresses would win this draw and transfer the NFT there, frontrunning the automated `finalizeRandomness` function.

Repeat the same on each draw and with a single NFT takes holder bonus from multiple addresses.

Recommendation: Override the `_beforeTokenTransfer` function from the `ERC721A` not to allow transferring the NFT until the current request is finalized.

Resolution: Fixed

5.3.2. Transfers are temporarily blocked while the VRF request is pending

Severity: Medium Risk

Description: While **VRF** request is pending, which will happen each month for ~1 minute, all the `FeliX` transfers will be blocked temporarily. This will happen because BlackBox entries/removals happen in the `_update` function of the token and these functions have `noPendingRandomness` modifier applied, which will be reverting.

The risk is even bigger if the VRF request is stuck in a pending state, this will have a critical impact blocking the entire protocol.

Recommendation: In add/remove participants functions, instead of reverting when the request is pending, consider returning early. The tradeoff is that users who are eligible for distribution won't be added as participants and should perform another trade.

Resolution: Fixed

5.3.3. Buy and Burn has no slippage control

Severity: Medium Risk

Description: `BuyAndBurn::swapTitanXToFeliXAndBurn` has no slippage control applied, as a result in periods of less activity, anyone can sandwich himself extracting profit on top of the incentive fees given. Although there are transfer fees, they won't prevent the attack from being performed when `amountAllocated` is large enough.

Recommendation: Consider adding `whitelist` mapping and allowing the owner to add community members which will call the function.

Resolution: Fixed

5.3.4. Distribution patterns for 12 winners and above will take more than 100% of the defined allocation

Severity: Medium Risk

Description: No matter the `TitanX` amount and number of participants `BlackBox` should always distribute ~100% of the defined range. This is not the case for all the patterns, starting from 12 winners and above, as it will always distribute 100.02%, taking more funds than expected, especially given that there will be `PresaleMinter` NFT holders amongst the winners that will increase the % even more.

Recommendation: Instead of equally giving 0.0278e18 to the lower-end winners, decrease the number to 0.0277e17.

Resolution: Fixed

5.4. Low/Info severity

5.4.1. No dead blocks after LP is created

Severity: Low Risk

Description: After LP is created `FeliX` transfers should be disabled for at least 3 blocks, in order to prevent bots from extracting value from the inflated price. Currently, `FeliX::setLp` doesn't set the `lpCreationBlock` and transfers will be immediately available.

Recommendation: Add the `lpCreationBlock` to the `setLp`:

Resolution: Fixed

5.4.2. If Lp is not created 48 hours after launch anyone can brick the pool and lock the collected X28

Severity: Low Risk

Description: `30B` X28 is needed to form the LP through the minting contract. If 48 hours after minting starts, the amount hasn't been collected, anyone will be able to create it himself by providing the Felix token that has been claimed. That will result in a skewed `X28/Felix` ratio.

Recommendation: Adjust the numbers, depending on the expected participation interest.

Resolution: Acknowledged