SB SECURITY

# JakeX Universe Security Review


Nov 5, 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 | JakeX Universe |
| Repository | Private |
| Commit Hash | 3da620ac51f82dab08d1a8194d3cf4494dec799f |
| Resolution | c11326455b4316b43960079ac9d3784603f8fdcd |
| Timeline | November 4 - November 5, 2024 |

**Scope**

| |
|---|
| JakeXUniverse.sol |
| PiggyBank.sol |

**Issues Found**

| | |
|---|---|
| Critical Risk | 0 |
| High Risk | 0 |
| Medium Risk | 2 |
| Low/Info Risk | 2 |

# 5. Findings

## 5.1. Medium severity

### 5.1.1. Depositing airdropped NFTs can temporarily block the `PiggyBank`

**Severity:** Medium Risk

**Description:** Airdropped `JakeXUniverse` NFTs can also be deposited into `PiggyBank`, but the problem is that when minted they don't send any `JakeX` tokens to the Bank. Depositing such NFTs will make the other normal minters to not be able to deposit theirs because of the insufficient JakeX balance.

**Recommendation:** After discussing with the protocol team we advise to remove the airdrop function entirely.

**Resolution:** Fixed

### 5.1.2. Approved users can take over the NFT ownership

**Severity:** Medium Risk

**Description:** Any approved user can directly transfer the `JakeXUniverse` NFT to `PiggyBank` and then take over the ownership of the token by calling the withdrawal functions.

As a result, the owner won't receive any `JakeX` tokens back, as opposed to if he would have deposited it himself, the attacker only has to pay the initial 100k `JakeX`, from which he can retrieve `97k` back by depositing it again in the `PiggyBank`.

**Recommendation:** One approach is to override the `onERC721Received` of `PiggyBank` to check if the owner of the given `tokenId` is the `msg.sender`, but you also must disable the normal `transferFrom` in `JakeXUniverse`.

**Resolution:** Fixed, the team acknowledged the case that Smart Wallets and AA won't able to interact with the PiggyBank.sol.

## 5.2. Low/Info severity

### 5.2.1. Wrong interface used

**Severity:** Informational Risk

**Description:** `JakeXUniverse` is an ERC721A implementation, but in `PiggyBank` the original IERC721 interface is used.

**Recommendation:** Replace `IERC721` with `IERC721A`.

**Resolution:** Acknowledged

### 5.2.2. CEI pattern is not followed

**Severity:** Informational Risk

**Description:** In `PiggyBank` there is no unified way of following the CEI which is one of the key protections from reentrancy vulnerabilities.

For example, `withdrawWithJakeX` first transfers the `JakeXUniverse` NFTs to the caller and then takes the total price of the `JakeX` token. Currently, the only available protection is the `nonReentrant` modifier.

**Recommendation:** Refactor `withdrawWithJakeX` to take the payment token first and then send the NFTs.

**Resolution:** Fixed