Pashov Audit Group

BOB Token Security Review

October 18th 2025 - October 21st 2025

Contents

1. About Pashov Audit Group

Pashov Audit Group consists of 40+ freelance security researchers, who are well proven in the space - most have earned over $100k in public contest rewards, are multi-time champions or have truly excelled in audits with us. We only work with proven and motivated talent.

With over 300 security audits completed — uncovering and helping patch thousands of vulnerabilities — the group strives to create the absolute very best audit journey possible. While 100% security is never possible to guarantee, we do guarantee you our team's best efforts for your project.

Check out our previous work here or reach out on Twitter @pashovkrum.

2. Disclaimer

A smart contract security review can never verify the complete absence of vulnerabilities. This is a time, resource and expertise bound effort where we try to find as many vulnerabilities as possible. We can not guarantee 100% security after the review or even if the review will find any problems with your smart contracts. Subsequent security reviews, bug bounty programs and on-chain monitoring are strongly recommended.

3. Risk Classification

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

Impact

- **High** - leads to a significant material loss of assets in the protocol or significantly harms a group of users
- **Medium** - leads to a moderate material loss of assets in the protocol or moderately harms a group of users
- **Low** - leads to a minor material loss of assets in the protocol or harms a small group of users

Likelihood

- **High** - attack path is possible with reasonable assumptions that mimic on-chain conditions, and the cost of the attack is relatively low compared to the amount of funds that can be stolen or lost
- **Medium** - only a conditionally incentivized attack vector, but still relatively likely
- **Low** - has too many or too unlikely assumptions or requires a significant stake by the attacker with little or no incentive

4. About BOB Token

BOB Token is an upgradeable ERC20 governance token with voting capabilities.

5. Executive Summary

A time-boxed security review of the **bob-collective/bob-token** repository was done by Pashov Audit Group, during which **zark**, **Tejas Warambhe**, **IvanFitro**, **afriauditor** engaged to review **BOB Token**. A total of **2** issues were uncovered.

Protocol Summary	
Project Name	BOB Token
Protocol Type	ERC20 Token
Timeline	October 18th 2025 - October 21st 2025

Review commit hash:

- ba519a023c9d0239b48f887b185fbedc9b7b6139 (bob-collective/bob-token)

Scope

`BobTokenV2.sol` `BobTokenV2Upgrade.sol`

6. Findings

Findings count

Severity	Amount
Low	2
Total findings	**2**

Summary of findings

ID	Title	Severity	Status
[L-01]	`ERC20BurnableUpgradeable` and others not initialized in `BobToken`	Low	Acknowledged
[L-02]	`setCCIPAdmin` uses `DEFAULT_ADMIN_ROLE` not `onlyOwner` as stated	Low	Acknowledged

Low findings

[L-01] `ERC20BurnableUpgradeable` and others not initialized in `BobToken`

`BobToken` inherits from `ERC20BurnableUpgradeable`, `AccessControlUpgradeable`, and NoncesUpgradeable but they are not initialized. It is considered best practice to initialize these parent contracts.

Recommendation: Initialize `ERC20BurnableUpgradeable`, `AccessControlUpgradeable`, and `NoncesUpgradeable`.

[L-02] `setCCIPAdmin` uses `DEFAULT_ADMIN_ROLE` not `onlyOwner` as stated

Docs say **"only the owner can call this function"**, but the code enforces:

```
function setCCIPAdmin(address newAdmin) public onlyRole(DEFAULT_ADMIN_ROLE) {
    ...
}
```

After ownership is transferred (via `OwnableUpgradeable`), the **new owner may not have** `DEFAULT_ADMIN_ROLE`. Since roles aren't auto-synced on ownership changes, the **old default admin** (initial owner) can still call `setCCIPAdmin`, while the **new owner cannot**.

Recommendations

- Change modifier to `onlyOwner`.

```
function setCCIPAdmin(address newAdmin) public onlyOwner {
    address currentAdmin = s_ccipAdmin;
    s_ccipAdmin = newAdmin;
    emit CCIPAdminTransferred(currentAdmin, newAdmin);
}
```