Guard

Smart contracts security assessment

Final report

Tariff: Standard

Paca Finance

March 2025

0xguard.com

hello@0xguard.com

Contents

Introduction

The report has been prepared for **Paca Finance**.

The contract is available at 0x3fF44D639a4982A4436f6d737430141aBE68b4E1 in the BNB Smart Chain.

The contract is an upgradeable proxy (TransparentUpgradeableProxy by OpenZeppelin) with the ProxyAdmin contract at 0x2bc294bE6699442931093E950B95f76EE856236e as an upgrade executor. The ProxyAdmin contract is an ownable contract with single owner, which is set to 0xbf12D3b827a230F7390EbCc9b83b289FdC98ba81 EOA address at the time of writing this report.

The audited implementation of 0x3fF44D639a4982A4436f6d737430141aBE68b4E1 proxy is PacaFinanceWithBoostAndScheduleUSDT at 0x95A2C2324692bf840b7f6786177d7A1751A0A5c3 . The PacaFinanceWithBoostAndScheduleUSDT has 2 lists of privileged accounts of unknown lengths that can be checked with `owners` and `authorizedBots` getters.

Verified owners are `0xEa49eF0EfBC1B8E5ae471f3A30D68B3D0029b999` and `0x41970Ce76b656030A79E7C1FA76FC4EB93980255`, but the list may be incomplete.

The PacaFinanceWithBoostAndScheduleUSDT contract allows users to stake BSC-USD tokens for a fixed lockup period with a fixed reward rate. The rewards are paid in the same BSC-USD tokens. Users' funds are swept out by the contract owners, rewards are usually paid out immediately as a part of freshly withdrawn funds.

The PacaFinanceWithBoostAndScheduleUSDT contract allows to create vestings of fixed schedule for a limited list of supported ERC20 tokens. Users' vested funds can be withdrawn by the owners of the contract.

The further path of the withdrawn funds is unknown.

We've asked the owner to provide proof of reserves and list of authorized addresses. The owner refused to disclose that information.

The contract has properties of Ponzi scheme with about 120% APR at the time of writing.

The contract is operated in state of permanent rug pull, with rewards and withdrawals dependent on the good will of the owners.

Name	Paca Finance
Audit date	2025-03-04 - 2025-03-10
Language	Solidity
Platform	Binance Smart Chain

Contracts checked

Name	Address
PacaFinanceWithBoostAndScheduleUSDT	0x3fF44D639a4982A4436f6d737430141aBE68b4E1

Procedure

We perform our audit according to the following procedure:

Automated analysis

- Scanning the project's smart contracts with several publicly available automated Solidity analysis tools
- Manual verification (reject or confirm) all the issues found by the tools

Manual audit

- Manually analyze smart contracts for security vulnerabilities
- Smart contracts' logic check

Known vulnerabilities checked

Title	Check result
Unencrypted Private Data On-Chain	passed
Code With No Effects	passed
Message call with hardcoded gas amount	passed
Typographical Error	passed
DoS With Block Gas Limit	passed
Presence of unused variables	passed
Incorrect Inheritance Order	passed
Requirement Violation	passed
Weak Sources of Randomness from Chain Attributes	passed
Shadowing State Variables	passed
Incorrect Constructor Name	passed
Block values as a proxy for time	passed
Authorization through tx.origin	passed
DoS with Failed Call	passed
Delegatecall to Untrusted Callee	passed
Use of Deprecated Solidity Functions	passed
Assert Violation	passed
State Variable Default Visibility	passed
Reentrancy	passed
Unprotected SELFDESTRUCT Instruction	passed
Unprotected Ether Withdrawal	passed
Unchecked Call Return Value	passed

Floating Pragma	passed
Outdated Compiler Version	passed
Integer Overflow and Underflow	passed
Function Default Visibility	passed

Classification of issue severity

High severity	High severity issues can cause a significant or full loss of funds, change of contract ownership, major interference with contract logic. Such issues require immediate attention.
Medium severity	Medium severity issues do not pose an immediate risk, but can be detrimental to the client's reputation if exploited. Medium severity issues may lead to a contract failure and can be fixed by modifying the contract state or redeployment. Such issues require attention.
Low severity	Low severity issues do not cause significant destruction to the contract's functionality. Such issues are recommended to be taken into consideration.

Issues

High severity issues

1. Owner privileges (PacaFinanceWithBoostAndScheduleUSDT)
Status: Open

Members of the owners list can add or remove other owners.

Members of the owners list can add or remove authorized bots.

Members of the owners list can modify pool's parameters without safety limits.

Members of the owners list can modify pool's sell tax, restake bonus, unlock delay, sale minimum without safety limits.

Members of the bots list can clear stake data for any user address.

Members of the bots list can create stakes without depositing.

Recommendation: Publish the full list of owners, remove EOA owners, secure remaining members with Multisig and/or Timelock contract, provide monitoring methods for future changes in the list of owners.

2. Unknown source of rewards (PacaFinanceWithBoostAndScheduleUSDT)

Status: Open

The staking rewards are promised to be made in the same token with fixed reward rate. The rewards can be deposit only by the owners of the contract. If the deposited rewards are not available, user's funds are withdrawn from the contract's balance that is shared across all stakes and vestings. In that case there will be lack of funds to maintain all withdrawals, according to the Ponzi scheme.

Recommendation: Separate staked funds and rewards in the calculations. Secure withdrawals and keep track of unclaimed rewards after stake is withdrawn.

Medium severity issues

1. Total staked amount can't be tracked (PacaFinanceWithBoostAndScheduleUSDT)

Status: Open

Pool's `totalStaked` contains not only users' stakes but also information of withdrawn rewards. Sum `pool.totalStaked+withdrawLiabilities` represents the total amount of tokens that needs to be proved via proof of reserves.

```
function claimStake(uint256 _stakeIndex) external nonReentrant {
    Stake storage stake = stakes[msg.sender][_stakeIndex];
    uint256 _amount = stake.amount;
    uint rewards = getPoolRewards(msg.sender, _stakeIndex);
    _amount = _amount + rewards;
    ...
    if (pool.totalStaked >= _amount) {
        pool.totalStaked -= _amount;
```

```
    } else {
        pool.totalStaked = 0;
    }
}
```

Low severity issues

1. Possible DoS by gas (PacaFinanceWithBoostAndScheduleUSDT)

Status: Open

`claimRewards` function iterates over non-decreasing array of user's stakes.

```
function claimRewards() external nonReentrant {
    ...
    for (uint256 i = 0; i < stakes[msg.sender].length; ++i) {
        Stake storage stake = stakes[msg.sender][i];
        if (stake.amount > 0) {
            uint rewards = getPoolRewards(msg.sender, i);
            totalReward = totalReward + rewards;
            stake.lastClaimed = block.timestamp;
        }
    }
}
```

2. Vesting schedule can exceed 100% (PacaFinanceWithBoostAndScheduleUSDT)

Status: Open

Vesting schedule is allowed to be greater than 100% in the `claimVesting` function.

```
function claimVesting(uint256 _vestingIndex) external nonReentrant {
    ...
    vesting.claimedAmount = vesting.claimedAmount + amountToClaim;
    if (vesting.claimedAmount >= vesting.amount) {
        vesting.complete = true;
    }
}
```

3. Possible math underflow (PacaFinanceWithBoostAndScheduleUSDT)

Status: Open

Possible math underflow error in `getUserTotalUnclaimedUsdValue` function, because `Vesting.claimedAmount` is allowed to be greater than `Vesting.amount`.

4. Potential storage collision (PacaFinanceWithBoostAndScheduleUSDT)

Status: Open

Vesting bonus can be claimed as ordinary element of `withdrawStake` array with additional 1e6 offset. The one million offset is big value but not unreachable.

```
function claimBonus(uint256 _vestingIndex) external nonReentrant {
    ...
    withdrawStake[msg.sender].push(
        WithdrawStake({
            stakeId: _vestingIndex + 1e6,
            amount: bonusToClaim,
            unlockTime: block.timestamp + unlockDelay
        })
    );
}
```

5. Vesting unlocking schedules updating (PacaFinanceWithBoostAndScheduleUSDT)

Status: Open

`unlockSchedules` mapping can't be set up or updated without full upgrade of proxy implementation.

Conclusion

Paca Finance PacaFinanceWithBoostAndScheduleUSDT contract was audited. 2 high, 1 medium, 5 low severity issues were found.

The contract is upgradeable with single EOA admin.

The contract has lists of owners and bots of unknown lengths.

The owners transfer out users' funds to unknown destinations.

The contract doesn't hold enough tokens to paid out rewards, not to mention stakes and vested funds.

The contract is extremely dependent on the EOA owners and should be considered as dangerous to interact with.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability)set forth in the Services Agreement, or the scope of services, and terms and conditions provided to the Company in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes without 0xGuard prior written consent.

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts 0xGuard to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

0x Guard