PALADIN
BLOCKCHAIN SECURITY

Smart Contract Security Assessment

Final Report

For SmartCoin

18 October 2021

paladinsec.co

info@paladinsec.co

Table of Contents

Disclaimer **3**

1 Overview **4**

1.1 Summary 4

1.2 Contracts Assessed 4

1.3 Findings Summary 5

1.3.1 SmartCoin 6

1.3.2 MasterChef 6

2 Findings **7**

2.1 SmartCoin 7

2.1.1 Token Overview 7

2.1.2 Privileged Operations 7

2.1.3 Issues & Recommendations 8

2.2 MasterChef 10

2.2.1 Privileged Operations 10

2.2.2 Issues & Recommendations 11

Disclaimer

Paladin Blockchain Security ("Paladin") has conducted an independent audit to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the codes that were provided for the scope of this audit. This audit report does not constitute agreement, acceptance or advocation for the Project that was audited, and users relying on this audit report should not consider this as having any merit for financial advice in any shape, form or nature. The contracts audited do not account for any economic developments that may be pursued by the Project in question, and that the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are completely free of exploits, bugs, vulnerabilities or deprecation of technologies. Further, this audit report shall not be disclosed nor transmitted to any persons or parties on any objective, goal or justification without due written assent, acquiescence or approval by Paladin.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence regardless of the findings presented in this report. Information is provided 'as is', and Paladin is under no covenant to the completeness, accuracy or solidity of the contracts audited. In no event will Paladin or its partners, employees, agents or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions and/or actions with regards to the information provided in this audit report.

Cryptocurrencies and any technologies by extension directly or indirectly related to cryptocurrencies are highly volatile and speculative by nature. All reasonable due diligence and safeguards may yet be insufficient, and users should exercise considerable caution when participating in any shape or form in this nascent industry.

The audit report has made all reasonable attempts to provide clear and articulate recommendations to the Project team with respect to the rectification, amendment and/or revision of any highlighted issues, vulnerabilities or exploits within the contracts provided. It is the sole responsibility of the Project team to sufficiently test and perform checks, ensuring that the contracts are functioning as intended, specifically that the functions therein contained within said contracts have the desired intended effects, functionalities and outcomes of the Project team.

1 Overview

This report has been prepared for SmartCoin on the Avalanche network. Paladin provides a user-centred examination of the smart contracts to look for vulnerabilities, logic errors or other issues from both an internal and external perspective.

1.1 Summary

Project Name	SmartCoin
URL	https://smartcoin.farm/
Platform	Avalanche
Language	Solidity

1.2 Contracts Assessed

Both SmartCoin's token and Masterchef contracts are based on Trader Joe's respective contracts, thus we have listed them in the list of contracts assessed.

Name	Contract	Live Code Match
SmartCoin	https://gist.github.com/jeanclaudevanbammm/a158ca37d29b44ed0166b44f76d42a67	
MasterChefSmartyV2	https://gist.github.com/jeanclaudevanbammm/a158ca37d29b44ed0166b44f76d42a67	
MasterChefJoeV3	https://github.com/traderjoe-xyz/joe-core/blob/main/contracts/MasterChefJoeV3.sol	
JoeToken	https://github.com/traderjoe-xyz/joe-core/blob/main/contracts/JoeToken.sol	

1.3 Findings Summary

Severity	Found	Resolved	Partially Resolved	Acknowledged (no change made)
🔴 **High**	0	-	-	-
🟠 **Medium**	0	-	-	-
🟡 **Low**	3	-	-	3
🟣 **Informational**	3	-	-	3
Total	**6**	**0**	**-**	**6**

Classification of Issues

Severity	Description
🔴 **High**	Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, or impairment of the contract and its functions. Issues under this classification are recommended to be fixed with utmost urgency.
🟠 **Medium**	Bugs or issues with that may be subject to exploit, though their impact is somewhat limited. Issues under this classification are recommended to be fixed as soon as possible.
🟡 **Low**	Effects are minimal in isolation and do not pose a significant danger to the project or its users. Issues under this classification are recommended to be fixed nonetheless.
🟣 **Informational**	Consistency, syntax or style best practices. Generally pose a negligible level of risk, if any.

1.3.1 SmartCoin

ID	Severity	Summary	Status
01	LOW	`mint` function can be used to pre-mint large amounts of tokens before ownership is transferred to the Masterchef	ACKNOWLEDGED
02	INFO	Governance functionality is broken	ACKNOWLEDGED
03	INFO	`delegateBySig` can be frontrun and cause denial of service	ACKNOWLEDGED

1.3.2 MasterChef

ID	Severity	Summary	Status
04	LOW	Deposits and withdrawals may fail if an invalid Rewarder contract is used	ACKNOWLEDGED
05	LOW	Excess minting of dummy tokens may result in skewed reward emissions	ACKNOWLEDGED
06	INFO	Pools use the contract balance to figure out the total deposits	ACKNOWLEDGED

2 Findings

2.1 SmartCoin

The contract allows for tokens to be minted when the `mint` function is called by Owner, whom at the time of deployment would be the deployer. Ownership is generally transferred to the Masterchef via the `transferOwnership` function for emission rewards to be minted and distributed to users staking in the Masterchef.

The `mint` function can be used to pre-mint tokens for various uses including injection of initial liquidity, token presale, airdrops, and others.

2.1.1 Token Overview

Address	TBC
Token Supply	Unlimited
Decimal Places	18
Transfer Max Size	None
Transfer Min Size	None
Transfer Fees	None
Pre-mints	TBC

2.1.2 Privileged Operations

The following functions can be called by the owner of the contract:

- `mint`

2.1.3 Issues & Recommendations

Issue #01	**`mint` function can be used to pre-mint large amounts of tokens before ownership is transferred to the Masterchef**
Severity	● LOW SEVERITY
Description	The `mint` function could be used to pre-mint tokens for legitimate uses including, but not limited to, the injection of initial liquidity, token presale, or airdrops; however, this function may also be used to pre-mint and dump tokens when the token contract has been deployed but before ownership is set to the Masterchef contract. This risk is prevalent amongst less-reputable projects, and any pre-mints can be prominently seen on the Blockchain.
Recommendation	Consider being forthright if this `mint` function has been used by letting your community know how much was minted, where they are currently stored, if a vesting contract was used for token unlocking, and finally the purpose of the mints.
Resolution	● ACKNOWLEDGED

Issue #02	Governance functionality is broken
Severity	● INFORMATIONAL
Description	Although there is YAM-related delegation code in the token contract which is usually used for governance and voting, the delegation code can be abused as the delegates are not moved during transfers and burns. This allows for double spending attacks on the voting mechanism. It should be noted that this issue is present in pretty much every single farm out there including PancakeSwap and even SushiSwap.
Recommendation	The broken delegation-related code can be removed to reduce the size of the contract. If voting is ever desired, it can still be done through snapshot.org, used by many of the larger projects.
Resolution	● ACKNOWLEDGED

Issue #03	`delegateBySig` can be frontrun and cause denial of service
Severity	● INFORMATIONAL
Description	Currently if `delegateBySig` is executed twice, the second execution will be reverted. It is thus in theory possible for a bot to pick up `delegateBySig` transactions in the mempool and execute them before a contract can. The issue with this is that the rest of said contract functionality would be lost as well. This could be a problem in case it would have been executed by a contract that would have rewarded you for your delegation for example.
Recommendation	Similar to the broken governance functionality issue, this can just be removed.
Resolution	● ACKNOWLEDGED

2.2 MasterChef

The SmartCoin protocol uses the TraderJoe V3 Masterchef, which is a fork of Goose Finance's Masterchef which has the `migrator` code from Pancakeswap removed. We commend SmartCoin on forking a relatively safe version of the Masterchef as the migrator has been used maliciously by several projects in the past to steal large sums of user funds. Additionally, there are no deposit fees in any pools. There are external calls made to the Rewarder contract, which is not covered in the audit scope, presumably to provide double rewards.

2.2.1 Privileged Operations

The following functions can be called by the owner of the contract:

- `add`
- `set`
- `dev`
- `setDevPercent`
- `setTreasuryAddr`
- `setTreasuryPercent`
- `setInvestorAddr`
- `setInvestorPercent`
- `updateEmissionRate`

2.2.2 Issues & Recommendations

Issue #04	Deposits and withdrawals may fail if an invalid Rewarder contract is used
Severity	● LOW SEVERITY

Location	Lines 275-278, 300-303
	`IRewarder _rewarder = pool.rewarder;`
	`if (address(_rewarder) != address(0)) {`
	`    _rewarder.onJoeReward(msg.sender, user.amount);`

Description	In the `deposit` and `withdraw` functions, external calls are made to the Rewarder contract. Should that be an invalid contract, then the external call and thus these functions (as well as `pendingTokens`) would fail.
Recommendation	The dev team should ensure that only valid, non-malicious Rewarder contracts will be used, as there is the ability to swap out the Rewarder contract in the `set` function. Additionally, if for whatever reason an invalid Rewarder contract was used, the `set` function mentioned above can be used to correct this. This issue will be marked as Resolved once the client acknowledges this.
Resolution	● ACKNOWLEDGED The client will not be using the Rewarder functionality.

Issue #05	Excess minting of dummy tokens may result in skewed reward emissions
Severity	● LOW SEVERITY
Description	The dummy token is to be deposited into a specific, pre-determined pool in the V2 Masterchef, which will funnel all emission rewards from that pool to the stakers in V3 Masterchef. When the init function is called, dummy tokens are transferred from `msg.sender` to the V3 Masterchef, and then on to the V2 Masterchef. If there are absolutely no other dummy tokens in circulation, then the V3 Masterchef would receive 100% of the dummy token pool's emissions in the V2 contract. However, if for whatever reason, an additional party also has dummy tokens, then they will be able to deposit that into the V2 Masterchef and thus receive staking rewards, diluting the yields available to V3.
Recommendation	Consider ensuring that there are no additional dummy tokens minted, apart from what will be used to call the `init` function. The simplest solution to this would be to renounce ownership of the dummy token contract after the appropriate minting amount has been done.
Resolution	● ACKNOWLEDGED

Issue #06	Pools use the contract balance to figure out the total deposits
Severity	● INFORMATIONAL
Description	As with pretty much all Masterchefs, the total number of tokens in the Masterchef contract is used to determine the total number of deposits. This can cause dilution of rewards when people accidentally send tokens to the Masterchef.
Recommendation	Consider adding an `lpSupply` variable to the `PoolInfo` that keeps track of the total deposits.
Resolution	● ACKNOWLEDGED