CERTIK

Security Assessment

Lucky Lion

Oct 12th, 2021

Table of Contents

Summary

This report has been prepared for Lucky Lion to discover issues and vulnerabilities in the source code of the Lucky Lion project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

Project Summary

Project Name	Lucky Lion
Description	Lucky Lion distributes Lion $Lucky token as credit to play games and win rewards. They support swapping via PancakeSwap and provides the farming by staking token in the revenue sharing pool.
Platform	BSC
Language	Solidity
Codebase	
Commit	259c10d1656d2434650edd5ad92add88b105d317

Audit Summary

Delivery Date	Oct 12, 2021
Audit Methodology	Static Analysis, Manual Review
Key Components	LuckyToken, MasterChef, SyrupBar

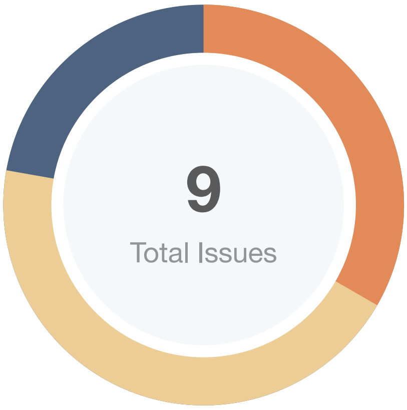
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
● Critical	0	0	0	0	0	0
● Major	3	0	0	1	2	0
● Medium	0	0	0	0	0	0
● Minor	4	0	0	4	0	0
● Informational	2	0	0	2	0	0
● Discussion	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
LTL	contracts/LuckyToken.sol	7938df5d32a5e0c42741d7a93e9687cb19d2c4c72bf199f188ac04b4a75a9e08
MLL	contracts/Masterchef.sol	7df558e517fb07184d079ea340ea8f5e4bd9d21231223b0bedfe08514e310bf4
MSL	contracts/MasterchefShield.sol	53f81e6eae25bce43f84928b5fa9dd4b853dc1ebedc55a7ec7461e6fb0b701d0
SBL	contracts/SyrupBar.sol	10d97bb3b9260e1abbd9535b03bd5f0d26f421408cec14a399f8c2223d1f198f
TLL	contracts/Timelock.sol	b66b8435b0f601a809ae1b13cecfa99e099ff0765a3d9268415295d3e6620596

Findings



🟥 **Critical**	**0** (0.00%)
🟧 **Major**	**3** (33.33%)
🟨 **Medium**	**0** (0.00%)
🟨 **Minor**	**4** (44.44%)
🟦 **Informational**	**2** (22.22%)
🟩 **Discussion**	**0** (0.00%)

ID	Title	Category	Severity	Status
LTL-01	Missing Input Validation	Volatile Code	● Minor	ⓘ Acknowledged
LTL-02	Centralization Risk	**Centralization / Privilege**	● **Major**	ⓘ Acknowledged
MLL-01	Missing Input Validation	Volatile Code	● Minor	ⓘ Acknowledged
MLL-02	Unbounded Loop	Gas Optimization	● Informational	ⓘ Acknowledged
MLL-03	Centralization Risk	**Centralization / Privilege**	● **Major**	◷ Partially Resolved
MLL-04	Storage Manipulation in `view` function	Gas Optimization	● Informational	ⓘ Acknowledged
MLL-05	Danger use of `Migrate`	**Centralization / Privilege**	● **Major**	◷ Partially Resolved
MSL-01	Ignored Input Value `_harvestIntervalInMinutes` and `_farmStartIntervalInMinutes`	Volatile Code	● Minor	ⓘ Acknowledged
TLL-01	Missing Input Validation	Volatile Code	● Minor	ⓘ Acknowledged

LTL-01 | Missing Input Validation

Category	Severity	Location	Status
Volatile Code	● Minor	contracts/LuckyToken.sol: 24~26	ⓘ Acknowledged

Description

The given input is missing the check for the non-zero address.

Recommendation

We advise adding the check for the passed-in values to prevent unexpected error as below:

Alleviation

Dev team had set the owner, eco and warchest wallet address on deploy contract to mainnet with the solid addresses which are shown in Git doc: https://docs.luckylion.io/security/contract-wallet-addresses."

LTL-02 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	● Major	contracts/LuckyToken.sol: 53, 45	ⓘ Acknowledged

Description

In the contract `SyrupBar`, the role `owner` has the authority over the following function:

- `mint`
- `_mint`

Any compromise to the `owner` account may allow the hacker to take advantage of this.

Recommendation

We advise the client to carefully manage the `owner` account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here is some feasible suggestions that would also mitigate the potential risk at the different level in term of short-term and long-term:

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key;
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

Alleviation

LuckyLion team has resolved this issue by implementing a timelock mechanism. Platform users could monitor the execution of functions in the timelock and act accordingly. The MasterChef contract is owned

by the Timelock contract with 7 days delay and 2 days minimum delay. Timelock contract with 2 days minimum delay: https://bscscan.com/address/0x4b6c8959a41475347226d51f37ec9a1e09f39a92#code MasterChef contract: https://bscscan.com/address/0xb6fe67c8a28d50c50f65fdb5847ee4477c550568#code Ownership transfer of MasterChef to Timelock contract: https://bscscan.com/tx/0xb54a48f780f6912f283b0113dfbb9fbef4d0f9e421bc532bb9c41a4
3cc15140f#eventlog

MLL-01 | Missing Input Validation

Category	Severity	Location	Status
Volatile Code	● Minor	contracts/Masterchef.sol: 104~105	ⓘ Acknowledged

Description

The given input is missing the check for the non-zero address.

Recommendation

We advise adding the check for the passed-in values to prevent unexpected error as below:

Alleviation

Dev team had set the owner, eco and warchest wallet address on deploy contract to mainnet with the solid addresses which are shown in Git doc: https://docs.luckylion.io/security/contract-wallet-addresses."

MLL-02 | Unbounded Loop

Category	Severity	Location	Status
Gas Optimization	● Informational	contracts/Masterchef.sol: 193~195	ⓘ Acknowledged

Description

The `for` loop within functions `massUpdatePools()` takes the following variable `poolInfo.length`, as the maximal iteration times. If the size of the array is very large, it could exceed the gas limit to execute the functions. In this case, the contract might suffer from DoS (Denial of Service) situation.

Recommendation

We recommend the team ensure this pools would not cause loss to the project.

Alleviation

The team has prepared a testing process on the local network (environment with the same settings as the main network) which has the same number of pools as the mainnet.

MLL-03 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	contracts/Masterchef.sol: 152, 132, 198, 377, 383, 377, 370	Partially Resolved

Description

In the contract `MasterChef`, the role `owner` has the authority over the following function:

- setMigrator
- add
- set
- setDevAddress
- transferLuckyOwnership
- updateLuckyPerBlock

Any compromise to the `owner` account may allow the hacker to take advantage of this

Recommendation

We advise the client to carefully manage the `[fixme]` account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here is some feasible suggestions that would also mitigate the potential risk at the different level in term of short-term and long-term:

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key;
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

Alleviation

By process,The tean have now made a shield to cover Masterchef so the owner can't use functions that affect investors such as migrator, transfer lucky ownership and havest lock-up. This shield is public on gitbook but not deployed yet as it's waiting for audit completion, ownership masterchef will be transferred to shield and timelock will take over the shield again (owner->timelock->shield->masterchef)

MLL-04 | Storage Manipulation in `view` function

Category	Severity	Location	Status
Gas Optimization	● Informational	contracts/Masterchef.sol: 186	ⓘ Acknowledged

Description

There should not be any storage variable manipulation in the `view` function

Recommendation

We advise the client to consider changing `storage` into `memory` for `data` in L352 and use a local variable to store the value of `result`

MLL-05 | Danger use of `Migrate`

Category	Severity	Location	Status
Centralization / Privilege	● Major	contracts/Masterchef.sol: 21, 208~210	Partially Resolved

Description

The `migrate` function could be used to transferring user deposited tokens to another address. Any compromise to the migrator address could cause user funds loss.

Recommendation

We advise the client to carefully manage the `owner` account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here is some feasible suggestions that would also mitigate the potential risk at the different level in term of short-term and long-term:

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key;
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.

Alleviation

By process, The team have now made a shield to cover Masterchef so the owner can't use functions that affect investors such as migrator, transfer lucky ownership and havest lock-up.

MSL-01 | Ignored Input Value `_harvestIntervalInMinutes` and `_farmStartIntervalInMinutes`

Category	Severity	Location	Status
Volatile Code	● Minor	contracts/MasterchefShield.sol: 14~16, 18~20	ⓘ Acknowledged

Description

The input variables `_harvestIntervalInMinutes` and `_farmStartIntervalInMinutes` is not used in function `add` and `set`, but 0 is used to invoke masterchef.

```
14  function add(uint256 _allocPoint, IERC20 _lpToken, uint256
_harvestIntervalInMinutes,uint256 _farmStartIntervalInMinutes) external onlyOwner {
15    masterchef.add(_allocPoint, _lpToken, 0, 0);
16  }
17
18  function set(uint256 _pid, uint256 _allocPoint, uint256
_harvestIntervalInMinutes,uint256 _farmStartIntervalInMinutes) external onlyOwner {
19    masterchef.set(_pid, _allocPoint, 0, 0);
20  }
```

The pool added or modified by MasterchedSheild would have 0 harvest interval and 0 farm start interval but ignore the values from `_harvestIntervalInMinutes` and `_farmStartIntervalInMinutes`.

Recommendation

We suggest dev team use the input intervals to avoid 0 harvest interval and 0 farm start interval.

Alleviation

[Lucky Lion Team]: We add this parameter for check a signature to execution function on timelock.

TLL-01 | Missing Input Validation

Category	Severity	Location	Status
Volatile Code	● Minor	contracts/Timelock.sol: 43	ⓘ Acknowledged

Description

The given input is missing the check for the non-zero address.

Recommendation

We advise adding the check for the passed-in values to prevent unexpected error as below:

Alleviation

Dev team had set the owner, eco and warchest wallet address on deploy contract to mainnet with the solid addresses which are shown in Git doc: https://docs.luckylion.io/security/contract-wallet-addresses."

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux "sha256sum" command against the target file.

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 you ("Customer" or 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, nor may copies be delivered to any other person other than the Company, without CertiK's prior written consent in each instance.

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 CertiK 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.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK's position is that each company and individual are responsible for their own due diligence and continuous security. CertiK's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED "AS IS" AND "AS

AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING

MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

About

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.