ELLUFA

Your Portal To Unlimited Possibilities

SMART CONTRACT AUDIT

ZOKYO.

Mar 19, 2021 | v. 1.0

PASS

Zokyo's Security Team has concluded that this smart contract passes security qualifications to be listed on digital asset exchanges.

TECHNICAL SUMMARY

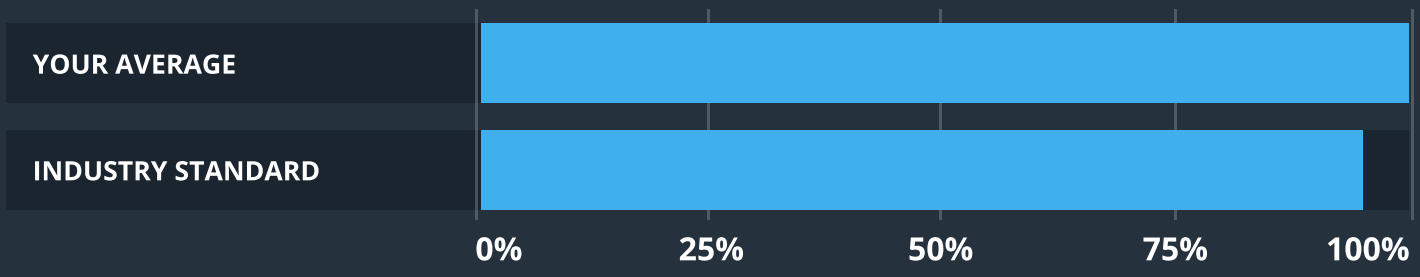
This document outlines the overall security of the Ellufa smart contracts, evaluated by Zokyo's Blockchain Security team.

The scope of this audit was to analyze and document the Ellufa smart contract codebase for quality, security, and correctness.

Contract Status

There were two critical issues found during the audit, which were successfully resolved.

Testable Code



Testable code is 100% which is above the industry standard of 95%.

It should be noted that this audit is not an endorsement of the reliability or effectiveness of the contract, rather limited to an assessment of the logic and implementation. In order to ensure a secure contract that's able to withstand the Ethereum network's fast-paced and rapidly changing environment, we at Zokyo recommend that the Ellufa team put in place a bug bounty program to encourage further and active analysis of the smart contract.

TABLE OF CONTENTS

AUDITING STRATEGY AND TECHNIQUES APPLIED

The Smart contract's source code was taken from the following github repository – https://github.com/zokyo-secured/Ellufa.
Commit id – 0146ae81c6441aa0949e12aa6ffe3e34092bb019.

Throughout the review process, care was taken to ensure that the token contract:

- Implements and adheres to existing Token standards appropriately and effectively;
- Documentation and code comments match logic and behavior;
- Distributes tokens in a manner that matches calculations;
- Follows best practices in efficient use of gas, without unnecessary waste;
- Uses methods safe from reentrance attacks;
- Is not affected by the latest vulnerabilities;
- Whether the code meets best practices in code readability, etc.

Zokyo's Security Team has followed best practices and industry-standard techniques to verify the implementation of Ellufa smart contracts. To do so, the code is reviewed line-by-line by our smart contract developers, documenting any issues as they are discovered. In summary, our strategies consist largely of manual collaboration between multiple team members at each stage of the review:

1	Due diligence in assessing the overall code quality of the codebase.	3	Testing contract logic against common and uncommon attack vectors.
2	Cross-comparison with other, similar smart contracts by industry leaders.	4	Thorough, manual review of the codebase, line-by-line.

EXECUTIVE SUMMARY

There were two critical issues found during the audit, which were successfully resolved. All the mentioned findings may have an effect only in case of specific conditions performed by the contract owner.

Contracts are well written and structured. The findings during the audit have no impact on contract performance or security, so it is fully production-ready.

STRUCTURE AND ORGANIZATION OF DOCUMENT

For ease of navigation, sections are arranged from most critical to least critical. Issues are tagged “Resolved” or “Unresolved” depending on whether they have been fixed or addressed. Furthermore, the severity of each issue is written as assessed by the risk of exploitation or other unexpected or otherwise unsafe behavior:

Critical

The issue affects the ability of the contract to compile or operate in a significant way.

High

The issue affects the ability of the contract to compile or operate in a significant way.

Medium

The issue affects the ability of the contract to operate in a way that doesn’t significantly hinder its behavior.

Low

The issue has minimal impact on the contract’s ability to operate.

Informational

The issue has no impact on the contract’s ability to operate.

MANUAL REVIEW

CRITICAL | RESOLVED

Contract owner can change Ellufa token address any time by calling method addELFTAddress.

Recommendation:
Verify that Ellufa token address is not set before setting value.

CRITICAL | RESOLVED

Method initDeposit doubles state variable current_staked instead of increasing it by tokens fee amount.

HIGH | UNRESOLVED

Method initDeposit is not deducting fees from deposit amount.

Recommendation:
Deduct fees from deposit amount before increasing max & left earnings for user.

MEDIUM | UNRESOLVED

USDT address is hardcoded and points to test ERC20 contract.

Recommendation:
Retrieve chain id and based on that set proper address for variable usdt_address.

LOW | UNRESOLVED

Use SafeMath for all arithmetic operations.

INFORMATIONAL | UNRESOLVED

Smart contract is not covered by NatSpec annotations.

Recommendation:
Cover by NatSpec all Contract methods.

INFORMATIONAL | UNRESOLVED

Contract Ellufa state variables have different naming styles.

Recommendation:
Use recommended camelcase naming style for variables on Solidity.

INFORMATIONAL | UNRESOLVED

DRY, code to check if current caller is owner is copied multiple times.

Recommendation:
Use modifiers to check if current caller is owner of contract.

CODE COVERAGE AND TEST RESULTS FOR ALL FILES

Tests written by Zokyo Secured team

Contract: KickPad contract

Ellufa creating phase
- ✓ should owner be correct (88ms)
- ✓ should companyaddress be correct (72ms)
- ✓ should usdt address be correct (61ms)
- ✓ should phaseversion be correc (57ms)
- ✓ should tokendebit be correct (60ms)
- ✓ should multiplier be correct (69ms)
- ✓ should min_withdrawal be correct (72ms)
- ✓ should staking_status be correct (115ms)
- ✓ should merchant_status be correct (76ms)
- ✓ should token_share be correct (79ms)
- ✓ should package initial status and maxPayout be correct (93ms)

Ellufa init deposit
- ✓ should init deposit correct if phase version equals 1 (3697ms)
- ✓ shouldn't deposit if MAX CAP NOT REACHED (2113ms)
- ✓ shouldn't deposit if USDT APPROVAL FAILED (562ms)
- ✓ should init deposit correct if phase version equals 2 (3598ms)

Ellufa addNodeAddress
- ✓ should addNodeAddress correctly (362ms)
- ✓ should change addNodeAddress correctly (488ms)
- ✓ shouldn't addNodeAddress by not an owner (135ms)
- ✓ should addPayout correctly (4632ms)
- ✓ shouldn't add payout by not an owner (346ms)

Ellufa withdraw phase
- ✓ should withdraw correct (5175ms)

Ellufa investStaking phase
- ✓ should investStaking correct (5414ms)
- ✓ should addELFTAddress correct (419ms)
- ✓ shouldn't addELFTAddress by not an owner (126ms)
- ✓ shouldn't set ELFTAddress equal zero address (304ms)
- ✓ shouldn't addELFTAddress twice (472ms)

```
    ✓ should  addTokenPrice correct (348ms)
    ✓ shouldn't  addTokenPrice by not an owner (307ms)
    ✓ should  investStaking correct when ELFTAddress is set (6452ms)
    ✓ shouldn't  investStaking when phase version is 1 (128ms)
    ✓ shouldn't  enablePhase2 by not an owner (132ms)
  Ellufa addMerchant phase
    ✓ should  addMerchant correct (670ms)
    ✓ shouldn't  addMerchant by not an owner (131ms)
    ✓ should  addPackage correct (292ms)
    ✓ shouldn't  addPackage by not an owner (119ms)
    ✓ shouldn't  addPackage with maxPayout less 2 (134ms)
    ✓ should  addLeaderAddress correct (1951ms)
    ✓ shouldn't  addLeaderAddress by not an owner (123ms)
    ✓ should  updateTokenShares correct (151ms)
    ✓ shouldn't  updateTokenShares by not an owner (126ms)
    ✓ should  payMerchant correct when ELFTAddress is set (6756ms)
    ✓ shouldn't  payMerchant when phase version is 1 (131ms)

  42 passing (3m)
```

FILE	% STMTS	% BRANCH	% FUNCS	% LINES	UNCOVERED LINES
contracts/	100.00	91.67	100.00	100.00	
Ellufa.sol	100.00	91.67	100.00	100.00	
All files	**100.00**	**91.67**	**100.00**	**100.00**	

We are grateful to have been given the opportunity to work with the Ellufa team.

The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them.

Zokyo's Security Team recommends that the Ellufa team put in place a bug bounty program to encourage further analysis of the smart contract by third parties.

ZOKYO.