GrowFarm
SMART CONTRACT AUDIT

ZOKYO.

May 20th, 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

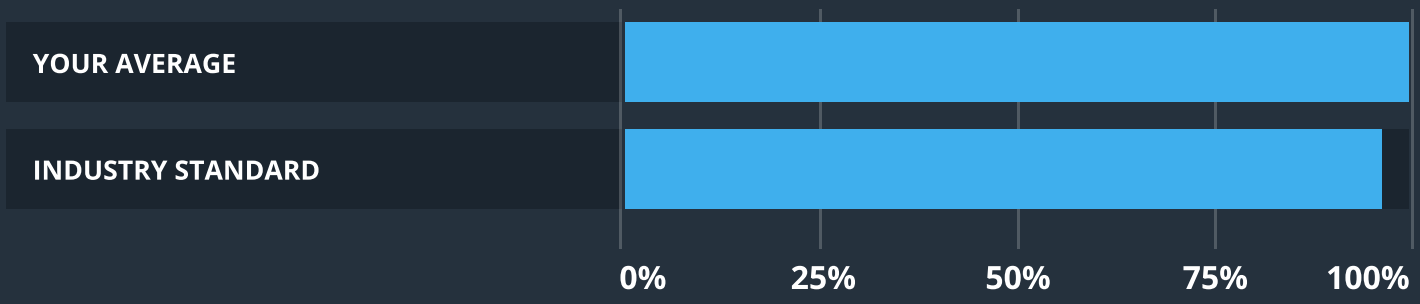
This document outlines the overall security of the GrowFarm smart contracts, evaluated by Zokyo's Blockchain Security team.

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

Contract Status

There were no critical issues found during the audit.

Testable Code



The 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 GrowFarm 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 GrowFarm repository.

Repository: https://github.com/nugbase/growfarm-erc721

SHA256 of the audited code archive -
4f946399637413fce27510882cfa651145473cf0fd660534994bf36ce7fda774

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 GrowFarm smart contracts. To do so, the code is reviewed line-by-line by our smart contract developers, documenting any issues as they are discovered. Part of this work includes writing a unit test suite using the Truffle testing framework. 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 no critical issues found during the audit. All the mentioned findings may have an effect only in case of specific conditions performed by the contract owner.

The findings during the audit have a slight impact on contract performance and code style.

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 contract in such a way that funds may be lost, allocated incorrectly, or otherwise result in a significant loss.

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.

COMPLETE ANALYSIS

LOW | UNRESOLVED

No revert message in the require

Growfarm.sol : tokenData(uint256)
Require statement must have error message in order to be informative.

Recommendation:
Add revert message to the require.

LOW | UNRESOLVED

Functions should be declared as external

In order to get gas savings and increase the overall security it is recommended to declare external functions:
setBaseURI(string) should be declared external
tokenData(uint256) should be declared external
mint(address,bytes) should be declared external
pause() should be declared external
unpause() should be declared external
setCompleted(uint256) should be declared external

Recommendation:
Declare functions as external.

LOW | UNRESOLVED

Unused ApprovedSpender.sol contract

Contract ApprovedSpender is empty and not used.

Recommendation:
Delete ApprovedSpender.sol contract

LOW | UNRESOLVED

Unnecessary Migrations.sol contract

Contract Migrations.sol is standard and Unnecessary.

Recommendation:
Delete Migrations.sol contract

	GrowFarm
Re-entrancy	Pass
Access Management Hierarchy	Pass
Arithmetic Over/Under Flows	Pass
Unexpected Ether	Pass
Delegatecall	Pass
Default Public Visibility	Pass
Hidden Malicious Code	Pass
Entropy Illusion (Lack of Randomness)	Pass
External Contract Referencing	Pass
Short Address/ Parameter Attack	Pass
Unchecked CALL Return Values	Pass
Race Conditions / Front Running	Pass
General Denial Of Service (DOS)	Pass
Uninitialized Storage Pointers	Pass
Floating Points and Precision	Pass
Tx.Origin Authentication	Pass
Signatures Replay	Pass
Pool Asset Security (backdoors in the underlying ERC-20)	Pass

CODE COVERAGE AND TEST RESULTS FOR ALL FILES

Tests written by Zokyo team

As part of our work assisting GrowFarm in verifying the correctness of their contract code, our team was responsible for writing integration tests using Truffle testing framework.
Tests were based on the functionality of the code, as well as review of the GrowFarm contract requirements for details about issuance amounts and how the system handles these.

```
Contract: Growfarm
  Mint permissions
    √ ROOT_ACCOUNT has only PRIV_ROOT and cannot mint (449ms)
    √ CREATOR_ACCOUNT has PRIV_MINT and can mint (233ms)
  Public token info
    tokenData
      √ token 2 data
      √ nonexistent token data (479ms)
    ownerOf
      √ token 1
      √ token 2
  tokenURI
    properly configured
      √ valid token should have tokenURI (38ms)
      √ nonexistent token should revert
    PRIV_ROOT required to set the metadata contract address
      √ with root (86ms)
      √ without root (59ms)
  executeMetaTransaction Simple
    √ setApprovalForAll MetaTransaction Test (4072ms)
  Pause and unpause functionality
    √ Pause can be performed by the account with the PAUSER_ROLE (527ms)
    √ Pause can't be performed by the account without the PAUSER_ROLE (124ms)
    √ Unpause can be performed by the account with the PAUSER_ROLE (135ms)
    √ Unpause can't be performed by the account without the PAUSER_ROLE (58ms)
  Support interface
    ERC165
      ERC165's supportsInterface(bytes4)
        √ uses less than 30k gas [skip-on-coverage]
        √ claims support
    ERC721
      ERC165's supportsInterface(bytes4)
        √ uses less than 30k gas [skip-on-coverage]
        √ claims support


19 passing (14s)
```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
growfarm\	100	90	100	100	
Growfarm.sol	100	90	100	100	
All files	100	90	100	100	

We are grateful to have been given the opportunity to work with the GrowFarm 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 GrowFarm team put in place a bug bounty program to encourage further analysis of the smart contract by third parties.

ZOKYO.