MiniKishu

Smart Contract Security Audit

Date 23/October/2021

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and ApeAudits and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (ApeAudits) owe no duty of care towards you or any other person, nor does ApeAudits make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and ApeAudits hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, ApeAudits hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against ApeAudits, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Purpose

MiniKishu reached out to us here at Ape Audits to conduct a full security audit with static and review analysis. The Auditing began October 14th and was concluded on October 21st. The assessment and testing was focused on staking functionality.

The purpose of the audit and unit testing was to:

-Solidify that smart contract functionality worked accordingly.

-Identify any code-breaking errors and potential security issues

This document details ApeAudits' findings and recommended solutions. Our audit was performed in the weeks leading up to MiniKishu's Staking launch. The contracts we audited were Alpha Crispe and KISHUCAKEPUNKSMALE

Testing Strategy and Audit Approach:

- Manual test of us and safety for the critical Solidity variables and functions.
- Unit Testing by custom scripts.
- Scanning of files for vulnerabilities or bugs. (Mythx)
- Static Analysis (slither)

Findings

NO.	Audit Items	Audit Subclass	Audit Subclass Result
1	Overflow Audit	N/A	Passed
2	Race Conditions Audit	N/A	Passed
3	Authority Control Audit	Permission Vulnerability Audit Excessive Auditing Authority	Passed Passed
4	Safe Design Audit	Zeppelin Module Safe Compiler Version Hard-coded Version Fallback Function Safeuse Show Coding Security Function Return Value Security Call Function Security	Passed Passed Passed Passed Passed Passed Passed
5	Denial of Service Audit	N/A	Passed
6	Gas Optimization Audit	N/A	Passed
7	Design Logic Audit	N/A	Passed
8	Malicious Event Log Audit	N/A	Passed
9	“False Deposit” Vulnerability Audit	N/A	Passed
10	Uninitialized Storage Pointers Audit	N/A	Passed
11	Arithmetic Accuracy Deviation Audit	N/A	Passed

Contract Summary:

- From IERC165

  - supportsInterface(bytes4) (external)

+ Contract ERC20 (Most derived contract)

- From ERC20

  - approve(address,uint256) (external)

- balanceOf(address) (external)
- decimals() (external)
- transfer(address,uint256) (external)
- transferFrom(address,address,uint256) (external)

+ Contract SafeMath (Most derived contract)
  - From SafeMath
    - add(uint256,uint256) (internal)
    - div(uint256,uint256) (internal)
    - div(uint256,uint256,string) (internal)
    - mod(uint256,uint256) (internal)
    - mod(uint256,uint256,string) (internal)
    - mul(uint256,uint256) (internal)
    - sub(uint256,uint256) (internal)
    - sub(uint256,uint256,string) (internal)

+ Contract IERC721
  - From IERC165
    - supportsInterface(bytes4) (external)
  - From IERC721
    - approve(address,uint256) (external)
    - balanceOf(address) (external)
    - getApproved(uint256) (external)
    - isApprovedForAll(address,address) (external)
    - ownerOf(uint256) (external)

- safeTransferFrom(address,address,uint256) (external)
- safeTransferFrom(address,address,uint256,bytes) (external)
- setApprovalForAll(address,bool) (external)
- transferFrom(address,address,uint256) (external)

+ Contract IERC721Receiver (Most derived contract)
- From IERC721Receiver
- onERC721Received(address,address,uint256,bytes) (external)

+ Contract IERC721Metadata
- From IERC721
- approve(address,uint256) (external)
- balanceOf(address) (external)
- getApproved(uint256) (external)
- isApprovedForAll(address,address) (external)
- ownerOf(uint256) (external)
- safeTransferFrom(address,address,uint256) (external)
- safeTransferFrom(address,address,uint256,bytes) (external)
- setApprovalForAll(address,bool) (external)
- transferFrom(address,address,uint256) (external)
- From IERC165
- supportsInterface(bytes4) (external)
- From IERC721Metadata
- name() (external)
- symbol() (external)

- tokenURI(uint256) (external)

+ Contract Address (Most derived contract)
  - From Address
    - _verifyCallResult(bool,bytes,string) (private)
    - functionCall(address,bytes) (internal)
    - functionCall(address,bytes,string) (internal)
    - functionCallWithValue(address,bytes,uint256) (internal)
    - functionCallWithValue(address,bytes,uint256,string) (internal)
    - functionDelegateCall(address,bytes) (internal)
    - functionDelegateCall(address,bytes,string) (internal)
    - functionStaticCall(address,bytes) (internal)
    - functionStaticCall(address,bytes,string) (internal)
    - isContract(address) (internal)
    - sendValue(address,uint256) (internal)

+ Contract Context
  - From Context
    - _msgData() (internal)
    - _msgSender() (internal)

+ Contract Strings (Most derived contract)
  - From Strings
    - toHexString(uint256) (internal)
    - toHexString(uint256,uint256) (internal)

    - toString(uint256) (internal)

+ Contract ERC165
  - From ERC165
    - supportsInterface(bytes4) (public)

+ Contract ERC721
  - From Context
    - _msgData() (internal)
    - _msgSender() (internal)
  - From ERC721
    - _approve(address,uint256) (internal)
    - _baseURI() (internal)
    - _beforeTokenTransfer(address,address,uint256) (internal)
    - _burn(uint256) (internal)
    - _checkOnERC721Received(address,address,uint256,bytes) (private)
    - _exists(uint256) (internal)
    - _isApprovedOrOwner(address,uint256) (internal)
    - _mint(address,uint256) (internal)
    - _safeMint(address,uint256) (internal)
    - _safeMint(address,uint256,bytes) (internal)
    - _safeTransfer(address,address,uint256,bytes) (internal)
    - _transfer(address,address,uint256) (internal)
    - approve(address,uint256) (public)
    - balanceOf(address) (public)

- constructor(string,string) (public)
- getApproved(uint256) (public)
- isApprovedForAll(address,address) (public)
- name() (public)
- ownerOf(uint256) (public)
- safeTransferFrom(address,address,uint256) (public)
- safeTransferFrom(address,address,uint256,bytes) (public)
- setApprovalForAll(address,bool) (public)
- supportsInterface(bytes4) (public)
- symbol() (public)
- tokenURI(uint256) (public)
- transferFrom(address,address,uint256) (public)

+ Contract IERC721Enumerable
  - From IERC721
    - approve(address,uint256) (external)
    - balanceOf(address) (external)
    - getApproved(uint256) (external)
    - isApprovedForAll(address,address) (external)
    - ownerOf(uint256) (external)
    - safeTransferFrom(address,address,uint256) (external)
    - safeTransferFrom(address,address,uint256,bytes) (external)
    - setApprovalForAll(address,bool) (external)
    - transferFrom(address,address,uint256) (external)
  - From IERC165

- supportsInterface(bytes4) (external)

- From IERC721Enumerable

- tokenByIndex(uint256) (external)

- tokenOfOwnerByIndex(address,uint256) (external)

- totalSupply() (external)

+ Contract ERC721Enumerable

- From ERC721

- _approve(address,uint256) (internal)

- _baseURI() (internal)

- _burn(uint256) (internal)

- _checkOnERC721Received(address,address,uint256,bytes) (private)

- _exists(uint256) (internal)

- _isApprovedOrOwner(address,uint256) (internal)

- _mint(address,uint256) (internal)

- _safeMint(address,uint256) (internal)

- _safeMint(address,uint256,bytes) (internal)

- _safeTransfer(address,address,uint256,bytes) (internal)

- _transfer(address,address,uint256) (internal)

- approve(address,uint256) (public)

- balanceOf(address) (public)

- constructor(string,string) (public)

- getApproved(uint256) (public)

- isApprovedForAll(address,address) (public)

- name() (public)

- ownerOf(uint256) (public)
- safeTransferFrom(address,address,uint256) (public)
- safeTransferFrom(address,address,uint256,bytes) (public)
- setApprovalForAll(address,bool) (public)
- symbol() (public)
- tokenURI(uint256) (public)
- transferFrom(address,address,uint256) (public)

- From Context
- _msgData() (internal)
- _msgSender() (internal)

- From ERC721Enumerable
- _addTokenToAllTokensEnumeration(uint256) (private)
- _addTokenToOwnerEnumeration(address,uint256) (private)
- _beforeTokenTransfer(address,address,uint256) (internal)
- _removeTokenFromAllTokensEnumeration(uint256) (private)
- _removeTokenFromOwnerEnumeration(address,uint256) (private)
- supportsInterface(bytes4) (public)
- tokenByIndex(uint256) (public)
- tokenOfOwnerByIndex(address,uint256) (public)
- totalSupply() (public)

+ Contract ERC721URIStorage

- From ERC721
- _approve(address,uint256) (internal)
- _baseURI() (internal)

- _beforeTokenTransfer(address,address,uint256) (internal)
- _checkOnERC721Received(address,address,uint256,bytes) (private)
- _exists(uint256) (internal)
- _isApprovedOrOwner(address,uint256) (internal)
- _mint(address,uint256) (internal)
- _safeMint(address,uint256) (internal)
- _safeMint(address,uint256,bytes) (internal)
- _safeTransfer(address,address,uint256,bytes) (internal)
- _transfer(address,address,uint256) (internal)
- approve(address,uint256) (public)
- balanceOf(address) (public)
- constructor(string,string) (public)
- getApproved(uint256) (public)
- isApprovedForAll(address,address) (public)
- name() (public)
- ownerOf(uint256) (public)
- safeTransferFrom(address,address,uint256) (public)
- safeTransferFrom(address,address,uint256,bytes) (public)
- setApprovalForAll(address,bool) (public)
- supportsInterface(bytes4) (public)
- symbol() (public)
- transferFrom(address,address,uint256) (public)

- From Context
  - _msgData() (internal)
  - _msgSender() (internal)

  - From ERC721URIStorage

    - _burn(uint256) (internal)

    - _setTokenURI(uint256,string) (internal)

    - tokenURI(uint256) (public)

+ Contract Ownable

  - From Context

    - _msgData() (internal)

    - _msgSender() (internal)

  - From Ownable

    - constructor() (internal)

    - owner() (public)

    - renounceOwnership() (public)

    - transferOwnership(address) (public)

+ Contract PupContract (Most derived contract)

  - From PupContract

    - burn(uint256) (external)

    - graduated(uint256) (external)

    - male(uint256) (external)

    - ownerOf(uint256) (external)

    - retired(uint256) (external)

    - tokenURI(uint256) (external)

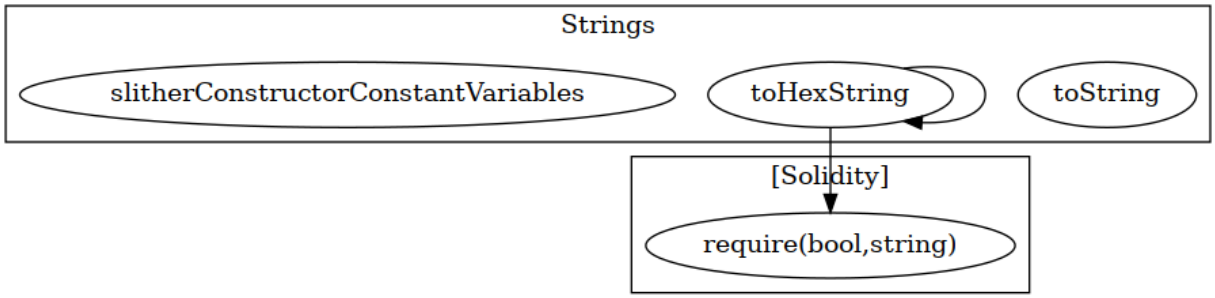
+ Contract KISHUCAKEPUNKSMALE (Most derived contract)

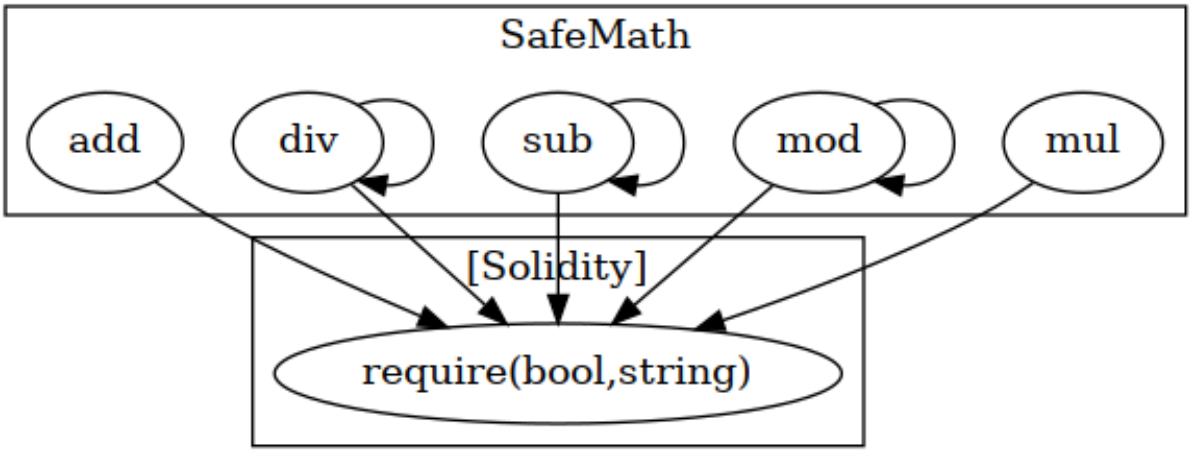
- From Ownable
  - owner() (public)
  - renounceOwnership() (public)
  - transferOwnership(address) (public)
- From Context
  - _msgData() (internal)
  - _msgSender() (internal)
- From ERC721URIStorage
  - _setTokenURI(uint256,string) (internal)
- From ERC721
  - _approve(address,uint256) (internal)
  - _checkOnERC721Received(address,address,uint256,bytes) (private)
  - _exists(uint256) (internal)
  - _isApprovedOrOwner(address,uint256) (internal)
  - _mint(address,uint256) (internal)
  - _safeMint(address,uint256) (internal)
  - _safeMint(address,uint256,bytes) (internal)
  - _safeTransfer(address,address,uint256,bytes) (internal)
  - _transfer(address,address,uint256) (internal)
  - approve(address,uint256) (public)
  - balanceOf(address) (public)
  - constructor(string,string) (public)
  - getApproved(uint256) (public)
  - isApprovedForAll(address,address) (public)
  - name() (public)

  - ownerOf(uint256) (public)
  - safeTransferFrom(address,address,uint256) (public)
  - safeTransferFrom(address,address,uint256,bytes) (public)
  - setApprovalForAll(address,bool) (public)
  - symbol() (public)
  - transferFrom(address,address,uint256) (public)
- From ERC721Enumerable
  - _addTokenToAllTokensEnumeration(uint256) (private)
  - _addTokenToOwnerEnumeration(address,uint256) (private)
  - _removeTokenFromAllTokensEnumeration(uint256) (private)
  - _removeTokenFromOwnerEnumeration(address,uint256) (private)
  - tokenByIndex(uint256) (public)
  - tokenOfOwnerByIndex(address,uint256) (public)
  - totalSupply() (public)
- From KISHUCAKEPUNKSMALE
  - _baseURI() (internal)
  - _beforeTokenTransfer(address,address,uint256) (internal)
  - _burn(uint256) (internal)
  - constructor() (public)
  - contractURI() (public)
  - getPrice() (public)
  - graduatePupToMaleBreeder(uint256) (public)
  - isBreederTooOld(uint256) (public)
  - isExpired(uint256) (public)
  - killPresale() (public)

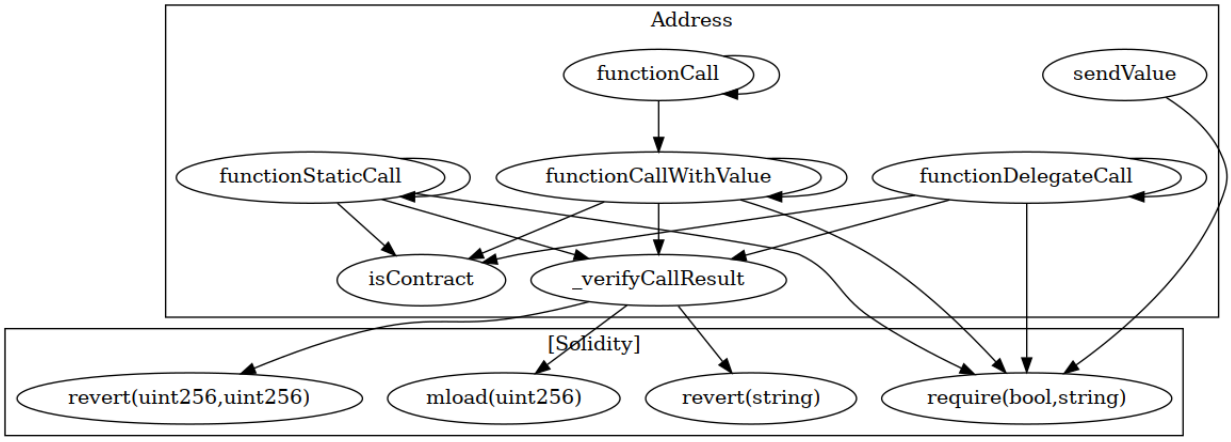
- mintMeSome(uint256) (public)
- onERC721Received(address,address,uint256,bytes) (public)
- retireMaleBreeder(uint256) (public)
- returnStake() (public)
- setOracle(address) (public)
- setPupTokenAddress(address) (public)
- setTeamWallet(address) (public)
- setTokenURI(string) (public)
- setTokenURI(uint256,string) (public)
- setTokenURIByOracle(uint256,string) (public)
- stakeMinikishu(uint256) (public)
- supportsInterface(bytes4) (public)
- toggleBurningEnabled() (public)
- toggleMintingEnabled() (public)
- tokenURI(uint256) (public)
- whitelistMinter() (public)
- withdrawETH() (public)

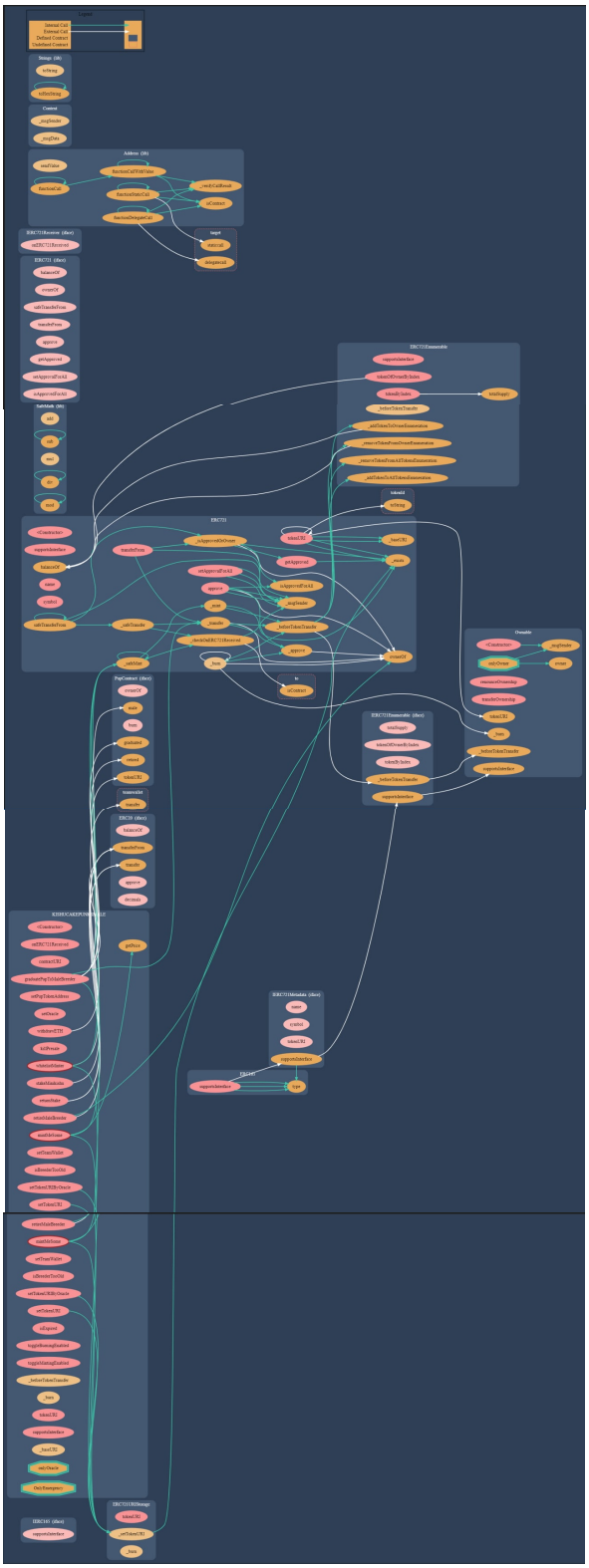
Visuals:


Strings
slitherConstructorConstantVariables
toHexString
toString
[Solidity]
require(bool,string)


SafeMath
add
div
sub
mod
mul
[Solidity]
require(bool,string)


PupContract
graduated
retired
tokenURI
ownerOf
male
burn


Address
functionCall
sendValue
functionStaticCall
functionCallWithValue
functionDelegateCall
isContract
_verifyCallResult
[Solidity]
revert(uint256,uint256)
mload(uint256)
revert(string)
require(bool,string)



High Severity Issues

- None.

Moderate Severity Issues

- None.

Low Severity Issues

- Warning: Contract code size exceeds 24576 bytes (a limit introduced in Spurious Dragon).

  This contract may not be deployable on mainnet. Consider enabling the optimizer (with a low "runs" value!), turning off revert strings, or using libraries

  - teamwallet.transfer (address(this).balance);

  - function stakeMinikishu(uint256 _qty) public {

    ERC20 _minikishu = ERC20 (minikishuAddress);

    _minikishu.transferFrom (msg.sender, address(this), _qty * stakingAmountRequired);

    totalStakedByUser[msg.sender] += _qty;

    userStakeExpiration[msg.sender] = block.timestamp + minimumstaketime;

    userMintable[msg.sender] += _qty

(Ignores return value by minikishu.transferFrom(msg.sender,address)

Should not be in an issue

ApeAudits note:
Please check the disclaimer above and note - the report is provided ‘as-is’ and makes no statements or warranties whatsoever. The report is provided only for the contract(s) mentioned in the report.