soken

SMART CONTRACT
SECURITY AUDIT

Kandyland DAO

December, 2021

Website: soken.io

Table of Contents

Disclaimer

This is a comprehensive report based on our automated and manual examination of cybersecurity vulnerabilities and framework flaws. We took into consideration smart contract based algorithms, as well. Reading the full analysis report is essential to build your understanding of project's security level. It is crucial to take note, though we have done our best to perform this analysis and report, that you should not rely on the our research and cannot claim what it states or how we created it. Before making any judgments, you have to conduct your own independent research. We will discuss this in more depth in the following disclaimer - please read it fully.

DISCLAIMER: You agree to the terms of this disclaimer by reading this report or any portion thereof. Please stop reading this report and remove and delete any copies of this report that you download and/or print if you do not agree to these conditions. This report is for non-reliability information only and does not represent investment advice. No one shall be entitled to depend on the report or its contents, and Soken and its affiliates shall not be held responsible to you or anyone else, nor shall Soken provide any guarantee or representation to any person with regard to the accuracy or integrity of the report. Without any terms, warranties or other conditions other than as set forth in that exclusion and Soken excludes hereby all representations, warrants, conditions and other terms (including, without limitation, guarantees implied by the law of satisfactory quality, fitness for purposes and the use of reasonable care and skills). The report is provided as "as is" and does not contain any terms and conditions. Except as legally banned, Soken disclaims all responsibility and responsibilities and no claim against Soken is made to any amount or type of loss or damages (without limitation, direct, indirect, special, punitive, consequential or pure economic loses or losses) that may be caused by you or any other person, or any damages or damages, including without limitations (whether innocent or negligent).

Security analysis is based only on the smart contracts. No applications or operations were reviewed for security. No product code has been reviewed.

Procedure

Our analysis contains following steps:

1. Project Analysis;

2. Manual analysis of smart contracts:
- Deploying smart contracts on any of the network(Ropsten/Rinkeby) using Remix IDE
- Hashes of all transaction will be recorded
- Behaviour of functions and gas consumption is noted, as well.

3. Unit Testing:
- Smart contract functions will be unit tested on multiple parameters and under multiple conditions to ensure that all paths of functions are functioning as intended.
- In this phase intended behaviour of smart contract is verified.
- In this phase, we would also ensure that smart contract functions are not consuming unnecessary gas.
- Gas limits of functions will be verified in this stage.

4. Automated Testing:
- Mythril
- Oyente
- Manticore
- Solgraph

Terminology

We categorize the finding into 4 categories based on their vulnerability:

- Low-severity issue — less important, must be analyzed
- Medium-severity issue — important, needs to be analyzed and fixed
- High-severity issue —important, might cause vulnerabilities, must be analyzed and fixed
- Critical-severity issue —serious bug causes, must be analyzed and fixed.

Limitations

The security audit of Smart Contract cannot cover all vulnerabilities. Even if no vulnerabilities are detected in the audit, there is no guarantee that future smart contracts are safe. Smart contracts are in most cases safeguarded against specific sorts of attacks. In order to find as many flaws as possible, we carried out a comprehensive smart contract audit. Audit is a document that is not legally binding and guarantees nothing.

Token Contract Details for 26.12.2021

Contract Name: **KandyERC20Token**

Deployed address: **0x37deD665a387a6f170FB60376B3057f09df6c0Ea**

Total Supply: **250,000**

Token Tracker: **$KANDY**

Decimals: **9**

Token holders: **2**

Transactions count: **3**

Top 100 holders dominance: **100.00%**

Audit Details

Project Name: **Kandyland DAO**

Language: **Solidity**

Compiler Version: **v0.7.5**

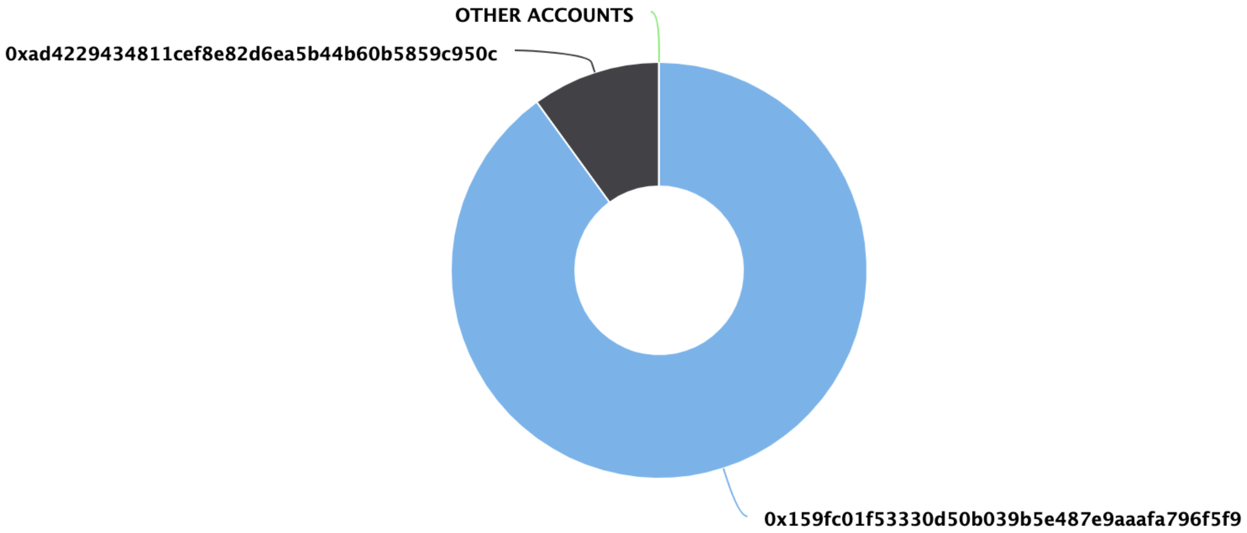
Blockchain: **Avalanche**

Social Profiles

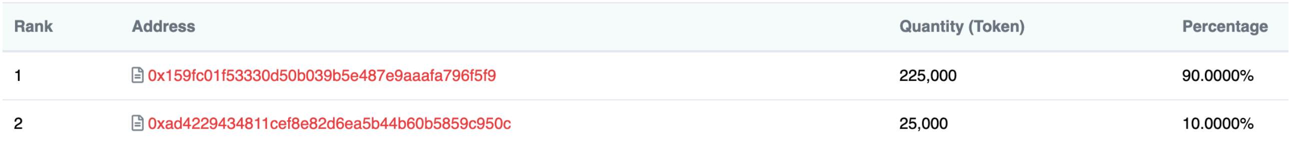
Project Website: **https://kandylanddao.com/**

Project Linktree: **https://linktr.ee/KandylandDAO**

$KANDY Token Distribution



$KANDY Top Holders



Rank	Address	Quantity (Token)	Percentage
1	0x159fc01f53330d50b039b5e487e9aaafa796f5f9	225,000	90.0000%
2	0xad4229434811cef8e82d6ea5b44b60b5859c950c	25,000	10.0000%

Contract Function Details

+ Contract Source Code
- [Prv] _add
- [Prv] _remove
- [Prv] _contains
- [Prv] _length
- [Prv] _at
- [Prv] _getValues
- [Prv] _insert
- [Int] add
- [Int] remove
- [Int] contains
- [Int] length
- [Int] at
- [Int] getValues
- [Int] insert
- [Int] add
- [Int] remove
- [Int] contains
- [Int] length
- [Int] at
- [Int] getValues
- [Int] insert
- [Int] add
- [Int] remove
- [Int] contains
- [Int] length
- [Int] at
- [Int] getValues
- [Int] insert
- [Int] add
- [Int] remove
- [Int] contains
- [Int] length
- [Int] at
- [Int] add
- [Int] remove
- [Int] contains
- [Int] length
- [Int] at
- [Ext] totalSupply
- [Ext] balanceOf
- [Ext] transfer

- [Ext] allowance
- [Ext] approve
- [Ext] transferFrom
- [Int] add
- [Int] sub
- [Int] sub
- [Int] mul
- [Int] div
- [Int] div
- [Int] mod
- [Int] mod
- [Int] sqrrt
- [Int] percentageAmount
- [Int] substractPercentage
- [Int] percentageOfTotal
- [Int] average
- [Int] quadraticPricing
- [Int] bondingCurve
- [Pub] name
- [Pub] symbol
- [Pub] decimals
- [Pub] totalSupply
- [Pub] balanceOf
- [Pub] transfer
- [Pub] allowance
- [Pub] approve
- [Pub] transferFrom
- [Pub] increaseAllowance
- [Pub] decreaseAllowance
- [Int] _transfer
- [Int] _mint
- [Int] _burn
- [Int] _approve
- [Int] _beforeTokenTransfer
- [Int] current
- [Int] increment
- [Int] decrement
- [Ext] permit
- [Ext] nonces
- [Pub] permit
- [Pub] nonces
- [Ext] owner
- [Ext] renounceOwnership
- [Ext] transferOwnership
- [Pub] owner

- [Pub] renounceOwnership
- [Pub] transferOwnership
- [Ext] setVault
- [Pub] vault
- [Ext] mint
- [Pub] burn
- [Pub] burnFrom
- [Pub] _burnFrom

Dapp audit scope

- Distributor.sol
- KandyBondDepository
- KandyBondingCalculator
- KandyCirculatingSupplyContract.sol
- KandyDAO.sol
- KandyERC20Token.sol
- KandySale.sol
- KandyStaking.sol
- KandyTreasury.sol
- RedeemHelper.sol
- StakingHelper.sol
- StakingWarmup.sol
- sKandy.sol

Vulnerabilities checking

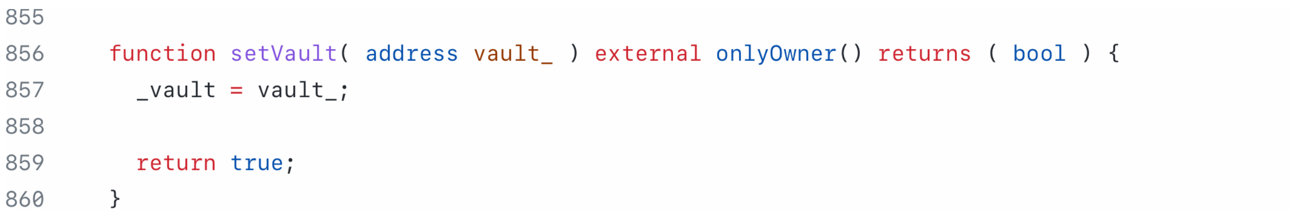
Issue Description	Checking Status
Compiler Errors	Completed
Delays in Data Delivery	Completed
Re-entrancy	Completed
Transaction-Ordering Dependence	Completed
Timestamp Dependence	Completed
Shadowing State Variables	Completed
DoS with Failed Call	Completed
DoS with Block Gas Limit	Completed
Outdated Complier Version	Completed
Assert Violation	Completed
Use of Deprecated Solidity Functions	Completed
Integer Overflow and Underflow	Completed
Function Default Visibility	Completed
Malicious Event Log	Completed
Math Accuracy	Completed
Design Logic	Completed
Fallback Function Security	Completed
Cross-function Race Conditions	Completed
Safe Zeppelin Module	Completed

Security Issues

1) Owner Privileges

The contract contains ownership functionality and ownership is not renounced which allows the creator or current owner to modify contract behaviour (for example, disable selling or mint new tokens).

2) Inexistent Zero Address Check | KandyERC20Token.sol



```
  function setVault( address vault_ ) external onlyOwner() returns ( bool ) {
    _vault = vault_;

    return true;
  }
```

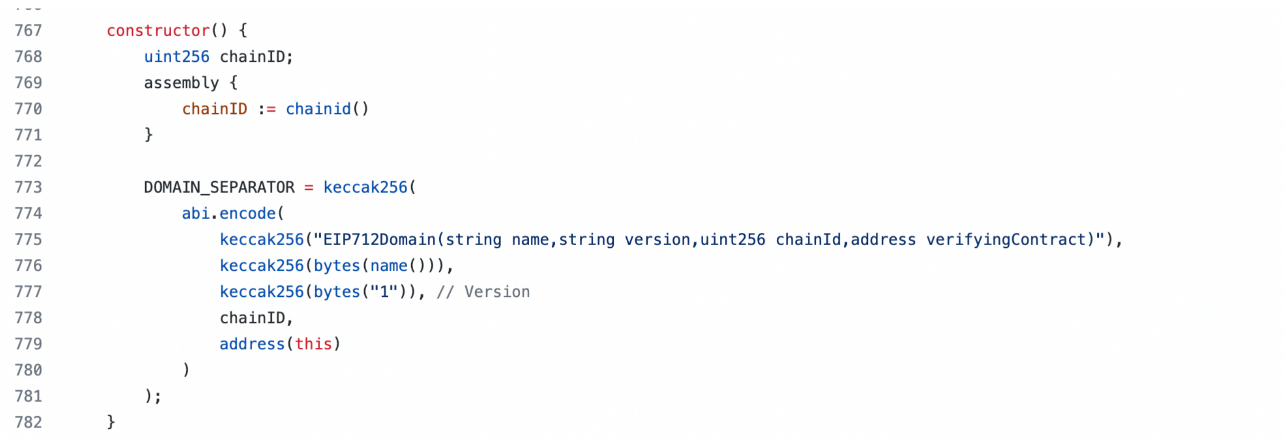
The setVault function of VaultOwned does not properly sanitize the input vault_ argument.

Recommendation:

We advise a zero-address check to be imposed to ensure that the system cannot be misconfigured. Additionally, we advise an event to be emitted signalling this change as it is considered a sensitive system change.

3) Cross-Chain Replay Attack | KandyERC20Token.sol

The DOMAIN_SEPARATOR of the KandyERC20Token contract is calculated once during the contract's constructor which is not compliant with the **EIP-712** recommended implementation of domainSeparator.



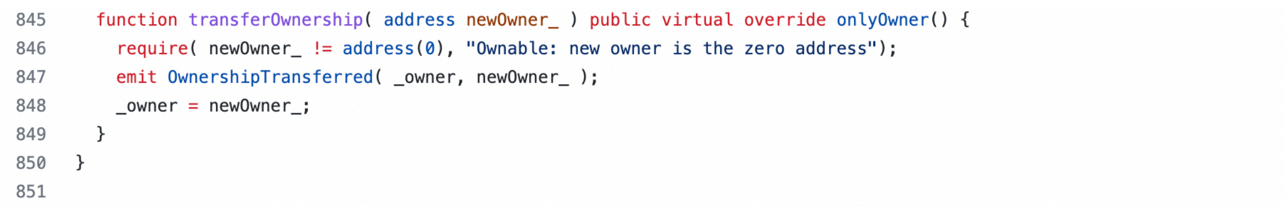
```
    constructor() {
        uint256 chainID;
        assembly {
            chainID := chainid()
        }

        DOMAIN_SEPARATOR = keccak256(
            abi.encode(
                keccak256("EIP712Domain(string name,string version,uint256 chainId,address verifyingContract)"),
                keccak256(bytes(name())),
                keccak256(bytes("1")), // Version
                chainID,
                address(this)
            )
        );
    }
```

Recommendation:

We advise an implementation like the **EIP-712 Draft** of OpenZeppelin to be utilized, whereby the separator is cached and newly calculated if the current chain's ID mismatches the one used in the constructor.

4) Pull-Over-Push Pattern I KandyERC20Token

The transferOwnership function overwrites the previously set _owner with the newOwner_ without validating that the new owner is able to actuate transactions on the blockchain.



```
  function transferOwnership( address newOwner_ ) public virtual override onlyOwner() {
    require( newOwner_ != address(0), "Ownable: new owner is the zero address");
    emit OwnershipTransferred( _owner, newOwner_ );
    _owner = newOwner_;
  }
}
```

Recommendation:

We advise the pull-over-push pattern to be applied here whereby a new owner is first proposed and consequently needs to accept ownership in a separate transaction indicating that they are able to transaction the blockchain and are aware of the particular contract's ownership.

Conclusion

Low-severity issues exist within smart contracts. Smart contracts are free from any critical or high-severity issues.

NOTE: Please check the disclaimer above and note, that audit makes no statements or warranties on business model, investment attractiveness or code sustainability.

Soken Contact Info

Website: www.soken.io

Mob: (+1)416-875-4174

32 Britain Street, Toronto, Ontario, Canada

Telegram: @team_soken

GitHub: sokenteam

Twitter: @soken_team