soken

SMART CONTRACT
SECURITY AUDIT

FLUUS

Scan and check this report
was posted at Soken Github

May, 2023

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 of the project's smart contract.

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. Scan and verify report's presence in the GitHub repository by a qr-code at the title page. 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.

Basic Security Recommendation

Unlike hardware and paper wallets, hot wallets are connected to the internet and store private keys online, which exposes them to greater risk. If a company or an individual holds significant amounts of cryptocurrency in a hot wallet, they should consider using MultiSig addresses. Wallet security is enhanced when private keys are stored in different locations and are not controlled by a single entity.
More info: **https://medium.com/soken-labs/how-to-gnosis-multisig-46b1386ba8e5**

Token Contract Details for 25.04.2023

Contract Name: **ERC1967Proxy**

Deployed address: **0xDBabd6c54D43361E457A06D677341049AC23EFac**

Total Supply: **1,000,000,000**

Token Tracker: **FLUUS**

Decimals: **18**

Token holders: **9**

Transactions count: **11**

Top 100 holders dominance: **100.00%**

Audit Details

Project Name: **FLUUS**

Language: **Solidity**

Compiler Version: **v0.8.9**

Blockchain: **Avalanche**

Social Profiles

Project Website: **https://www.fluus.com/**

Project Twitter: **https://twitter.com/itsFLUUS**

Project Telegram: **https://t.me/itsFLUUS**

Project Discord: **https://discord.com/invite/hKZpzwu9AE**

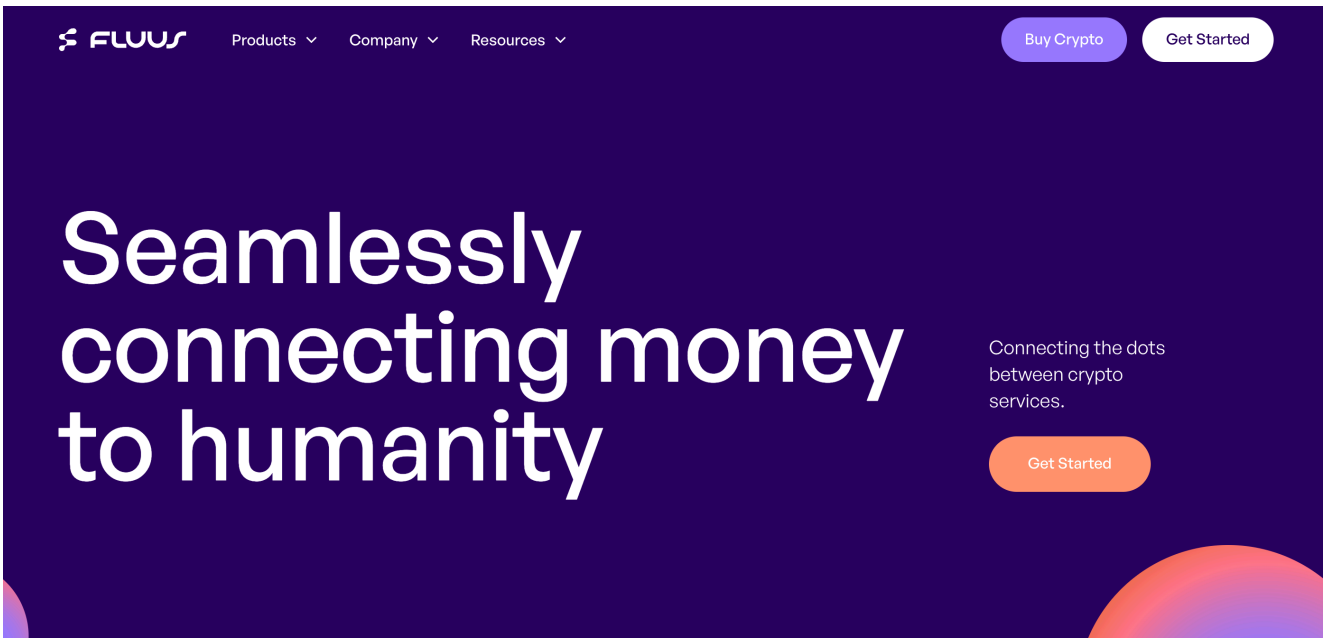
Project LinkedIn: **https://www.linkedin.com/company/itsfluus/**

Project Instagram: **https://www.instagram.com/itsFLUUS/**

Project Facebook: **https://www.facebook.com/itsFLUUS**

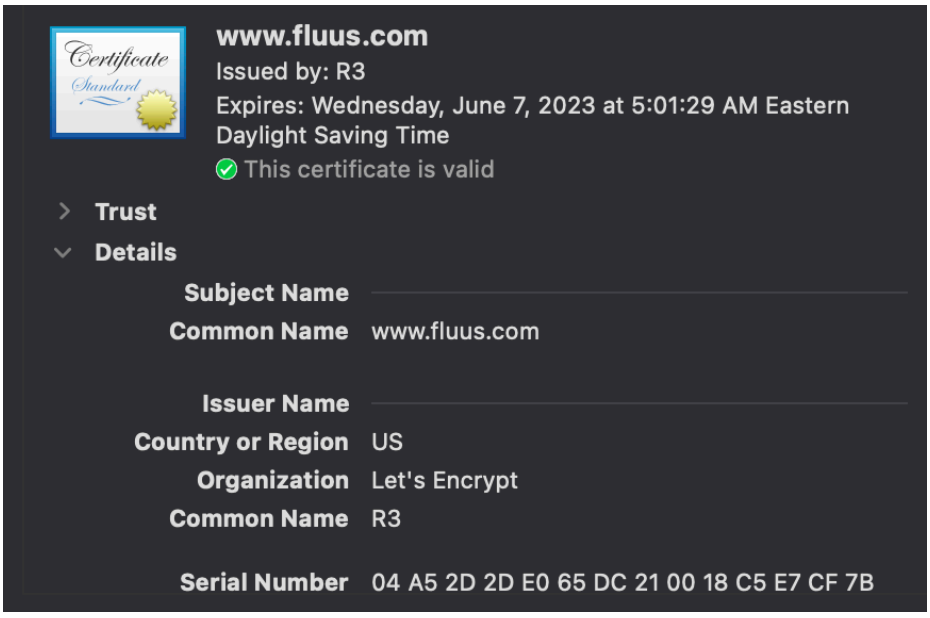
Project Youtube: **https://www.youtube.com/@itsFLUUS**

Project Website Overview



✓JavaScript errors hasn't been found.
✓Malware pop-up windows hasn't been detected.
✓No issues with loading elements, code, or stylesheets.

Project Website SSL Certification

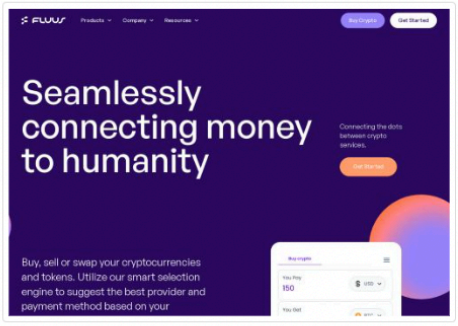


Project Website Optimization for Desktop

Performance

Values are approximate and subject to change. The performance level is calculated directly from these metrics. Show calculator

▲ 0–49 ■ 50–89 ● 90–100



INDICATORS Expand

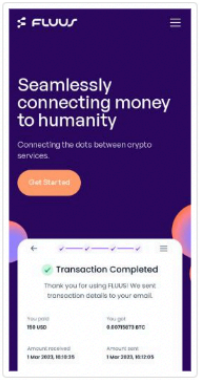
- ● First Contentful Paint 0.6 sec.
- ▲ Total Blocking Time 390 ms
- ● Speed Index 0.7 sec.
- ● Largest Contentful Paint 0.6 sec.
- ● Cumulative Layout Shift 0.008

Project Website Optimization for Mobile

Performance

Values are approximate and subject to change. The performance level is calculated directly from these metrics. Show calculator

▲ 0–49 ■ 50–89 ● 90–100



INDICATORS

- ■ First Contentful Paint 2.0 sec.
- ▲ Total Blocking Time 3040 ms
- ■ Speed Index 4.2 sec.
- ● Largest Contentful Paint 2.0 sec.
- ■ Cumulative Layout Shift 0.239

Contract Function Details

+ BeaconProxy
- [Int] _beacon
- [Int] _implementation
- [Int] _setBeacon

+ UpgradeableBeacon
- [Pub] implementation
- [Pub] upgradeTo
- [Prv] _setImplementation

+ ERC1967Proxy
- [Int] _implementation

+ TransparentUpgradeableProxy
- [Ext] admin
- [Ext] implementation
- [Ext] changeAdmin
- [Ext] upgradeTo
- [Ext] upgradeToAndCall
- [Int] _admin
- [Int] _beforeFallback

+ ProxyAdmin
- [Pub] getProxyImplementation
- [Pub] getProxyAdmin
- [Pub] changeProxyAdmin
- [Pub] upgrade
- [Pub] upgradeAndCall

+ IBeacon
- [Ext] implementation

+ Proxy
- [Int] _delegate
- [Int] _implementation
- [Int] _fallback
- [Int] _beforeFallback

+ ERC1967Upgrade
- [Int] _getImplementation
- [Prv] _setImplementation
- [Int] _upgradeTo
- [Int] _upgradeToAndCall

- [Int] _upgradeToAndCallSecure
- [Int] _upgradeBeaconToAndCall
- [Int] _getAdmin
- [Prv] _setAdmin
- [Int] _changeAdmin
- [Int] _getBeacon
- [Prv] _setBeacon

+ Address
- [Int] isContract
- [Int] sendValue
- [Int] functionCall
- [Int] functionCall
- [Int] functionCallWithValue
- [Int] functionCallWithValue
- [Int] functionStaticCall
- [Int] functionStaticCall
- [Int] functionDelegateCall
- [Int] functionDelegateCall
- [Prv] _verifyCallResult

+ StorageSlot
- [Int] getAddressSlot
- [Int] getBooleanSlot
- [Int] getBytes32Slot
- [Int] getUint256Slot

+ Ownable
- [Pub] owner
- [Pub] renounceOwnership
- [Pub] transferOwnership

+ Context
- [Int] _msgSender
- [Int] _msgData

Vulnerabilities checking

Issue Description	Checking Status
Compiler Errors	Completed
Delays in Data Delivery	Completed
Re-entrancy	Low
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) Missing Payable in Call Function: Critical
Line #782

The contract is using a `.call()` method to make external calls along with passing some Ether as `msg.value`. Since the function `functionCallWithValue` is not marked as `payable`, the transaction will fail.

Recommendation:

If the function needs to pass some Ether as `msg.value` inside a function, make sure to set that function as `payable`. No changes are required if the use case is to send Ether from the contract's balance.

2) Presence of Overpowered Role: Informational
Lines #124-127, #346-348, #357-359, #369-371, #991-994, #1000-1004

The overpowered owner (i.e., the person who has too much power) is a project design where the contract is tightly coupled to their owner (or owners); only they can manually invoke critical functions. Due to the fact that this function is only accessible from a single address, the system is heavily dependent on the address of the owner. In this case, there are scenarios that may lead to undesirable consequences for investors, e.g., if the private key of this address is compromised, then an attacker can take control of the contract.

Recommendation:

We recommend designing contracts in a trust-less manner. For instance, this functionality can be implemented in the contract's constructor. Another option is to use a MultiSig wallet for this address. For systems that are provisioned for a single user, you can use [Ownable.sol]. For systems that require provisioning users in a group, you can use [@openzeppelin/Roles.sol] or [@hq20/Whitelist.sol].

Conclusion for project owner

Critical, high and informational-severity issues exist within smart contracts.

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

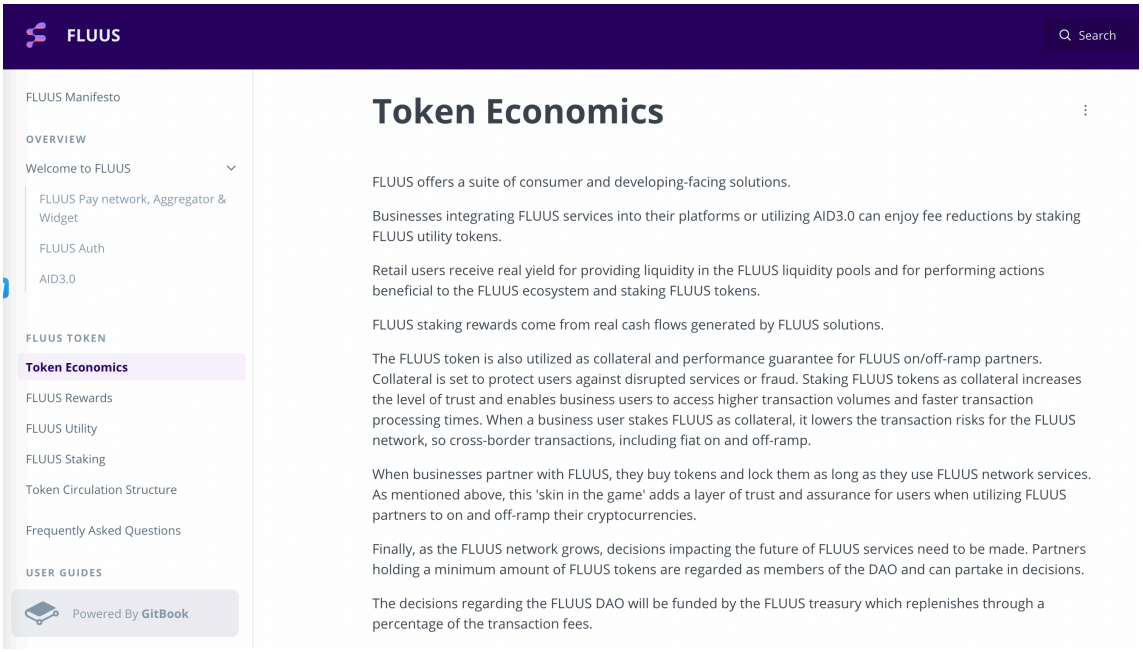
SECURITY REPORT
FOR COMMUNITY

FLUUS

soken

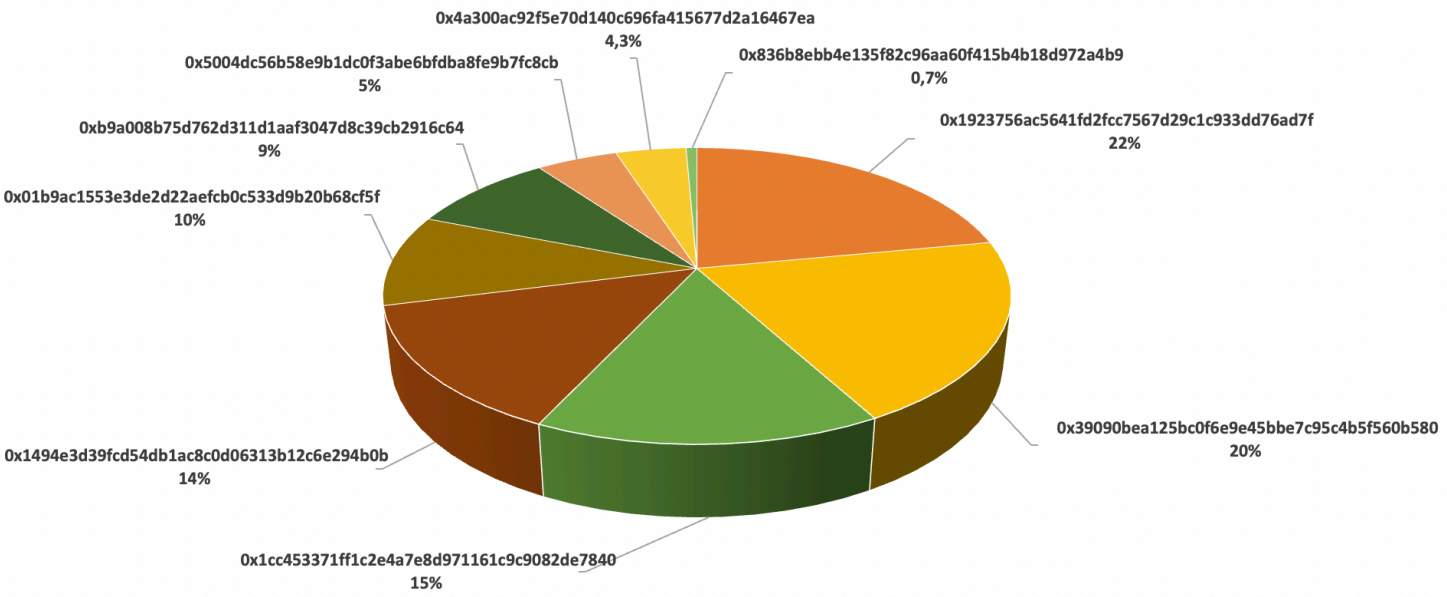
Whitepaper of the project

The whitepaper of FLUUS project has been verified on behalf of Soken team.



Whitepaper link: **https://fluus.gitbook.io/fluus/fluus-token/token-economics**

FLUUS Token Distribution



FLUUS Top 10 Holders

Rank	Address	Quantity (Token)	Percentage
1	0x1923756ac5641fd2fcc7567d29c1c933dd76ad7f	220,000,000	22.0000%
2	0x39090bea125bc0f6e9e45bbe7c95c4b5f560b580	200,000,000	20.0000%
3	0x1cc453371ff1c2e4a7e8d971161c9c9082de7840	150,000,000	15.0000%
4	0x1494e3d39fcd54db1ac8c0d06313b12c6e294b0b	140,000,000	14.0000%
5	0x01b9ac1553e3de2d22aefcb0c533d9b20b68cf5f	100,000,000	10.0000%
6	0xb9a008b75d762d311d1aaf3047d8c39cb2916c64	90,000,000	9.0000%
7	0x5004dc56b58e9b1dc0f3abe6bfdba8fe9b7fc8cb	50,000,000	5.0000%
8	0x4a300ac92f5e70d140c696fa415677d2a16467ea	43,333,334	4.3333%
9	0x836b8ebb4e135f82c96aa60f415b4b18d972a4b9	6,666,666	0.6667%

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