QuillAudits

Audit Report
October, 2023

For

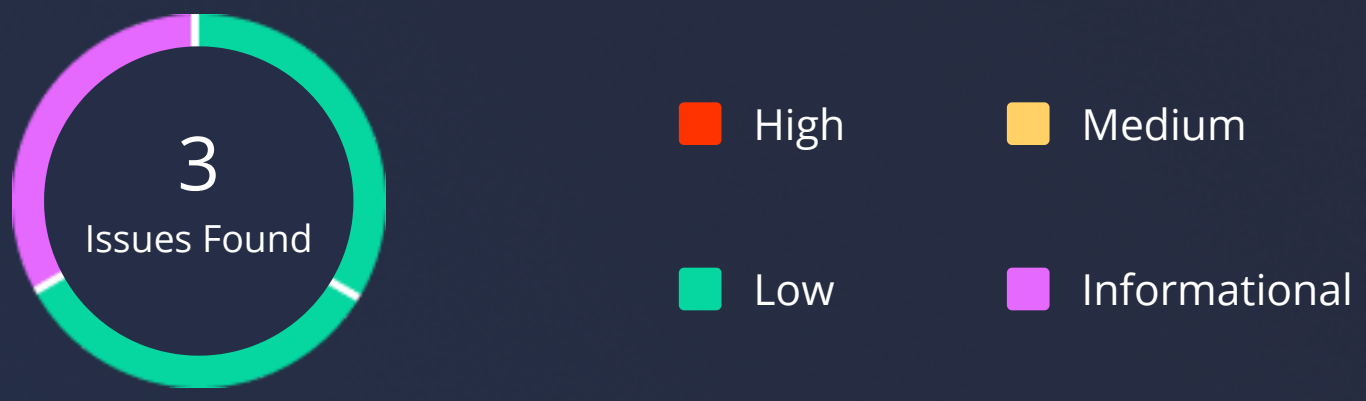
Table of Content

Table of Content

Executive Summary

Project Name	FAYA World
Project URL	*https://faya.world/*
Overview	FAYA is an ERC20 token with functionality to recover other ERC20 and native tokens.
Audit Scope	*https://bscscan.com/address/0x04e3e226bedfd572521984435 61b57c0a6456e9b#code*
Contracts in Scope	FAYA
Commit Hash	NA
Language	Solidity
Blockchain	BSC
Method	Manual Analysis, Functional Testing, Automated Testing
Review 1	27th October 2023 - 31st October 2023
Updated Code Received	NA
Review 2	NA
Fixed In	NA

Number of Security Issues per Severity



	High	Medium	Low	Informational
Open Issues	0	0	0	0
Acknowledged Issues	0	0	2	1
Partially Resolved Issues	0	0	0	0
Resolved Issues	0	0	0	0

Checked Vulnerabilities

- Access Management
- Arbitrary write to storage
- Centralization of control
- Ether theft
- Improper or missing events
- Logical issues and flaws
- Arithmetic Correctness
- Race conditions/front running
- SWC Registry
- Re-entrancy
- Timestamp Dependence
- Gas Limit and Loops
- Exception Disorder
- Gasless Send
- Use of tx.origin
- Malicious libraries
- Compiler version not fixed
- Address hardcoded
- Divide before multiply
- Integer overflow/underflow
- ERC's conformance
- Dangerous strict equalities
- Tautology or contradiction
- Return values of low-level calls
- Missing Zero Address Validation
- Private modifier
- Revert/require functions
- Multiple Sends
- Using suicide
- Using delegatecall
- Upgradeable safety
- Using throw

Checked Vulnerabilities

- Using inline assembly
- Style guide violation
- Unsafe type inference
- Implicit visibility level

Techniques and Methods

Throughout the audit of smart contracts, care was taken to ensure:

- The overall quality of code.
- Use of best practices.
- Code documentation and comments, match logic and expected behaviour.
- Token distribution and calculations are as per the intended behaviour mentioned in the whitepaper.
- Implementation of ERC standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods, and tools were used to review all the smart contracts.

Structural Analysis

In this step, we have analyzed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.

Static Analysis

A static Analysis of Smart Contracts was done to identify contract vulnerabilities. In this step, a series of automated tools are used to test the security of smart contracts.

Code Review / Manual Analysis

Manual Analysis or review of code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analyzed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of the automated analysis were manually verified.

Gas Consumption

In this step, we have checked the behavior of smart contracts in production. Checks were done to know how much gas gets consumed and the possibilities of optimization of code to reduce gas consumption.

Tools and Platforms used for Audit

Remix IDE, Truffle, Solhint, Mythril, Slither, Solidity Statistic Analysis.

Types of Severity

Every issue in this report has been assigned to a severity level. There are four levels of severity, and each of them has been explained below.

High Severity Issues

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality, and we recommend these issues be fixed before moving to a live environment.

Medium Severity Issues

The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems, and they should still be fixed.

Low Severity Issues

Low-level severity issues can cause minor impact and are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

Informational

These are four severity issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Types of Issues

Open

Security vulnerabilities identified that must be resolved and are currently unresolved.

Resolved

These are the issues identified in the initial audit and have been successfully fixed.

Acknowledged

Vulnerabilities which have been acknowledged but are yet to be resolved.

Partially Resolved

Considerable efforts have been invested to reduce the risk/impact of the security issue, but are not completely resolved.

A. Contract - FAYA

NOTE: In contract FAYA the ERC20MIntable is used which is not standard and is modified to meet the protocol requirements.

High Severity Issues

No issues were found.

Medium Severity Issues

No issues were found.

Low Severity Issues

A.1 Donations can't be sent to the contract

Description

In contract FAYA the constructor is payable so by means when contract is deployed with native token it can be recovered. But after that if someone tries to send native token as the contract doesn't have a **receive()** function it won't be able to accept donations afterwards.

Remediation

If the team wants to accept donations via contract then adding **receive()** function will work to transfer native token.

Status

Acknowledged

FAYA Team's Comment

We will promote that we cannot accept and donations on the contract address.

Informational Issues

A.2: Unlocked pragma (pragma solidity ^0.8.18)

Description

Contracts should be deployed with the same compiler version and flags that they have been tested with thoroughly. Locking the pragma helps to ensure that contracts do not accidentally get deployed using, for example, an outdated compiler version that might introduce bugs that affect the contract system negatively.

Remediation

Here all the in-scope contracts have an unlocked pragma, it is recommended to lock the same. Moreover, we strongly suggest not to use third-party unaudited libraries. If necessary, refactor the current code base to only use stable features.

Status

Acknowledged

B. Contract - TokenRecover

High Severity Issues

No issues were found.

Medium Severity Issues

No issues were found.

Low Severity Issues

B.1 Use safeTransfer instead of transfer for ERC20 tokens

Description

In contract TokenRecover transfer is used to transfer erc20 tokens to the owner which are sent as donations but using transfer sometimes does not work properly and can give true even if the transaction failed for some tokens such as USDT.

Remediation

Please make sure to use safeTransfer for ERC20 transactions.

Status

Acknowledged

FAYA Team's Comment

We will promote that we cannot accept and donations on the contract address.

Informational Issues

No issues were found.

Functional Tests

Some of the tests performed are mentioned below:

- [PASS] testFail_If_MoreThanApprovedAmountTriedToBurned() (gas: 15273)
- [PASS] testFail_If_NativeTokenSent() (gas: 17293)
- [PASS] testFail_If_OwnerIsAbleToBurnUsersAmount() (gas: 50792)
- [PASS] testFail_If_PreviousOwnerTriedToDoOperations() (gas: 23923)
- [PASS] testFail_If_TriedToMintAfterCallingMintingFinished() (gas: 12864)
- [PASS] testFail_If_TryToMintTokensAfterRenouncingOwnership() (gas: 22488)
- [PASS] testFail_ShouldNotBeAbleToBurnMoreThanCappedAmount() (gas: 29681)
- [PASS] testFail_ShouldNotBeAbleToMintMoreThanCappedAmount() (gas: 18693)
- [PASS] testFuzz_mintFunction(uint256) (runs: 256, μ: 24903, ~: 25975)
- [PASS] test_CheckIfEtherOrNativeTokenWillBeLocked() (gas: 1397736)
- [PASS] test_RenounceOwnership() (gas: 53215)
- [PASS] test_ShouldBeAbleToRecoverERC20Token() (gas: 61576)
- [PASS] test_UsersShouldBeAbleTransferTokens() (gas: 76548)
- [PASS] test_checkBalance() (gas: 12897)

Automated Tests

No major issues were found. Some false positive errors were reported by the tools. All the other issues have been categorized above according to their level of severity.

Closing Summary

In this report, we have considered the security of the FAYA World. We performed our audit according to the procedure described above.

Some issues of Low and informational severity were found, Some suggestions and best practices are also provided in order to improve the code quality and security posture.

Disclaimer

QuillAudits Smart contract security audit provides services to help identify and mitigate potential security risks in FAYA World smart contracts. However, it is important to understand that no security audit can guarantee complete protection against all possible security threats. QuillAudits audit reports are based on the information provided to us at the time of the audit, and we cannot guarantee the accuracy or completeness of this information. Additionally, the security landscape is constantly evolving, and new security threats may emerge after the audit has been completed.

Therefore, it is recommended that multiple audits and bug bounty programs be conducted to ensure the ongoing security of FAYA World smart contracts. One audit is not enough to guarantee complete protection against all possible security threats. It is important to implement proper risk management strategies and stay vigilant in monitoring your smart contracts for potential security risks.

QuillAudits cannot be held liable for any security breaches or losses that may occur subsequent to and despite using our audit services.. It is the responsibility of the FAYA World to implement the recommendations provided in our audit reports and to take appropriate steps to mitigate potential security risks.

About QuillAudits

QuillAudits is a secure smart contracts audit platform designed by QuillHash Technologies. We are a team of dedicated blockchain security experts and smart contract auditors determined to ensure that Smart Contract-based Web3 projects can avail the latest and best security solutions to operate in a trustworthy and risk-free ecosystem.

850+
Audits Completed

$30B
Secured

$30B
Lines of Code Audited

Follow Our Journey

Audit Report
October, 2023

For

QuillAudits

Canada, India, Singapore, UAE, UK

www.quillaudits.com

audits@quillhash.com