VIBRANIUM AUDITS

# Security Assessment

# Yasp Finance

Verified by Vibranium Audits on 30 March 2024

Revised on 04 April 2024

Vibranium Audits Verified on March 30th, 2024
Revised April 4th, 2024

# YaspFi

The security assessment was prepared by Vibranium Audits.

## Executive Summary

| TYPES | ECOSYSTEM | METHODS |
|---|---|---|
| DeFi/Router/AMM | Ethereum/EVM | Manual Review, penetration testing and Static Analysis |
| **LANGUAGE** | **TIMELINE** | **KEY COMPONENTS** |
| Solidity | Delivered on 03/04/2024 | EIP712/IERC20 |

| CODEBASE | COMMITS |
|---|---|
| https://github.com/yasp-fi/defi-router/tree/main | 7e19299da0c3224ae8969d48cc25692dd98f0248 |

## Vulnerability Summary

| 5 | 3 | 0 | 0 | 5 | 2 | 0 |
|---|---|---|---|---|---|---|
| Total Findings | Resolved | Mitigated | Partially Resolved | Acknowledged | Declined | Unresolved |

| | | | |
|---|---|---|---|
| 1 | Critical | 1 Resolved | Critical vulnerabilities are usually straightforward to exploit and can lead to the loss of user funds or contract state manipulation by external or internal actors. |
| 1 | High | 1 Resolved | High vulnerabilities are usually harder to exploit, requiring specific conditions, or have a more limited scope, but can still lead to the loss of user funds or contract state manipulation by external or internal actors. |
| 0 | Medium | 0 Resolved | Medium vulnerabilities are usually limited to state manipulations, but cannot lead to assets loss. Major deviations from best practices are also in this category. |
| 2 | Low | 1 Resolved<br>1 Declined | Low vulnerabilities are related to outdated and unused code or minor gas optimization. These issues won't have a significant impact on code execution, but affect the code quality. |
| 1 | Informational | 1 Declined | Informational errors are often recommendations to improve the style of the code or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code. |

# TABLE OF CONTENTS | YASP FINANCE

# CODEBASE | YASP FINANCE

## Repository

https://github.com/yasp-fi/defi-router/tree/main

## Commits

7e19299da0c3224ae8969d48cc25692dd98f0248

# AUDIT SCOPE | YASP FINANCE

4 files audited ● 4 file with Acknowledged findings ● 4 files with Resolved findings

| ID | Files | Commit Hash |
|---|---|---|
| ● DVA | DeFiRouter.sol | 96eddb03b9287c9c1f599160b00186253cdab263 |
| ● EVA | Executor.sol | caf6f61714e4beb3330cb064dd4b80ab820f5b65 |
| ● PVA | PermitVerifier.sol | c609acad6c89a83ef7e334c1589e2d2f5a3c3259 |

# APPROACH & METHODS | YASP FINANCE

This report has been prepared for YASP FINANCE(2024) to discover issues and vulnerabilities in the source code of the YASP FINANCE project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Manual Review, rigorous Penetration Testing and Static Analysis techniques.

The auditing process pays special attention to the following considerations:

- Pen-Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the code base to ensure compliance with current best practices and industry standards
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire code base by industry experts.

The security assessment resulted in findings that ranged from critical to informational We recommend addressing these findings to ensure a high level of security standards and industry practices.
We suggest recommendations that could better serve the project from the security perspective:

- Testing the smart contracts against both common and uncommon attack vectors.
- Enhance general coding practices for better structures of source codes.
- Review unit tests to cover the possible use cases.
- Review functions for readability, especially for future development work.

# FINDINGS | YASP FINANCE

| 5 | 1 | 1 | 0 | 2 | 1 |
|---|---|---|---|---|---|
| Total Findings | Critical | High | Medium | Low | Informational |

This report has been prepared to discover issues and vulnerabilities for YASP FINANCE. Through this audit, we have uncovered 11 issues ranging from different severity levels. Utilizing the techniques of Manual Review, Penetration Testing & Static Analysis to complement rigorous manual code reviews, we discovered the following findings:

| ID | Title | Category | Severity | Status |
|---|---|---|---|---|
| GVA-01 | Centralized Ownership | Design Issue | Minor | Acknowledged |
| DVA-01<br>EVA-01 | Reentrancy-ETH | Logical Issue | Critical | Acknowledged |
| GVA-02 | Floating Pragma | Logical Issue | Minor | Acknowledged |
| EVA-02 | Unchecked Return Values | Logical Issue | High | Acknowledged |
| PVA-01 | Lack of Event Emitting | Coding Style | Informational | Acknowledged |

# GVA-01 | Centralized Ownership

| Categor y | Severity | Location | Status |
|---|---|---|---|
| Design Issue | Minor | General | Declined |

## Description

Generally, the reviewed Smart Contracts rely on Openzeppelin's 'Ownable.sol' or a custom version of it to manage ownership of the contracts. Although no particular vulnerability is related to said used smart contract, having a single address gain ownership over the entire architecture could lead to severe issues outside of the project's or developers' control such as:

- Loss of the Owner address.
- Owner address private key gets compromised by external party.

## Recommendation

- Implement a multi-sig address as owner of the smart contracts, thus requiring multiple confirmations before executing one of the key and critical functionalities.
- OR implement a multi-owner structure (similar to Access Control) where multiple address on the smart contracts, thus preventing the complete loss of ownership if one owner address is lost.

## Current Status

The Yasp Finance team found it more suitable to keep the ownership structure as it is and more fitting to their goals.

# DVA-01 EVA-01 | Reentrancy-ETH

| Category | Severity | Location | Status |
|---|---|---|---|
| Logical Issue | Critical | DeFiRouter.sol<br>Executor.sol | Resolved |

## Description

A reentrancy is a programmatic approach in which an attacker performs recursive withdrawals to steal all Ether/Tokens locked in a contract.

Both reviewed Smart Contracts implement critical functionalities with token manipulation in the purpose of executing FlashLoans and other DeFi purposes:

- **function executeOperation (address[], uint256[], uint256[], address, bytes)** -> Executor.sol
- **function receiveFlashLoan (IERC20[], uint256[], uint256[], bytes)** -> Executor.sol
- **function payGas(address, uint256, address, uint256)** -> Executor.sol
- **function execute(bytes) ->** DeFiRouter.sol

These functions are highly likely to be vulnerable to the famous 'reentrancy-eth' attack **especially and not exclusively** that they do not implement SafeERC20 and have unchecked return values by IERC20 functions.

## Recommended

There are multiple ways to fix this issue, the most efficient and fitting for the purpose of these functionalities would be to simply implement OpenZeppelin's ReentrancyGuard.sol smart contract and simply use the 'nonReentrant' modifier on the scoped functions. One other approach would be to restructure the functions to implement the Check-Effects-Interactions pattern.

## Resolution

The "Yasp Finance" team successfully implemented OpenZepplin's ReentrancyGuard smart contract 'nonReentrant' modifier on all relevant functionalitites, alogside the implementation of safeERC20 for all relevant IERC20 manipulations.

# GVA-02 | Floating Pragma

| Categor y | Severity | Location | Status |
|---|---|---|---|
| Logical Issue | Minor | General | Resolved |

## Description

Both reviewed Smart Contracts rely on a floating solidity version. It is recommended to specify a fixed compiler version to ensure that the bytecode produced does not vary between builds. This is especially important if you rely on bytecode-level verification of the code.

## Recommendation

Use a new specific stable Solidity version e.g 0.8.20, also avoid using the latest version avoiding any potential yet undiscovered bugs.

## Resolution

The Yasp Finance Team successfully implemented a fixed solidity version across their smart contracts, thus mititgating the risk.

# EVA-02 | Unchecked return values

| Categor y | Severity | Location | Status |
|---|---|---|---|
| Logical Issue | High | Executor.sol | Resolved |

## Description

The reviewed Smart Contract “Executor.sol” implement ERC20 for token manipulation either by users or admins within external functions. ERC20 functions like transfer(), transferFrom() etc.. do not implement return value checks leading to;

- if insufficient tokens are present, no revert occurs but a result of "false" is returned.
- if the transferFrom() returns false, it would continue the call to withdraw token from the contract and send it to the caller. Thus a user could withdraw free tokens, and eventually some users will be unable to withdraw their tokens.

Reference: https://github.com/code-423n4/2022-06-nested-findings/issues/8

## Recommended

Implement SafeERC20 for all ERC20 actions within the function calls in both smart contracts. (i.e safeTransfer(), safeTransferFrom() etc..)

## Resolution

The Yasp Finance team successfully implemented SafeERC20 within all relevant functionalities, thus mitigating the risk.

# PVA-01 | Lack of event emitting

| Categor y | Severity | Location | Status |
|---|---|---|---|
| Logical Issue | Informational | PermitVerifier.sol | Declined |

## Description

The reviewed Smart Contract “PermitVerifier” contain key Admin/Owner functionalities that lack the emitting of events. There is no specific vulnerability associated with this matter.
It just would be better to add events to these functionalities for greater transparency with the community in Transactions and for future reference.

## Current Status

**The Yasp Finance team decided to keep the ‘PermitVerifier” smart contract as it is, fitting more to their goals and structure.**

# DISCLAIMER | VIBRANIUM AUDITS

This report is subject to the terms and conditions (including without limitation description of services confidentiality disclaimer and limitation of liability) set forth in the Services Agreement or the scope of services and terms and conditions provided to you (“Customer” or the “Company”) in connection with the Agreement This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement This report may not be transmitted disclosed referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Vibranium Audits prior written consent in each instance

This report is not nor should be considered an “endorsement” or “disapproval” of any particular project or team This report is not nor should be considered an indication of the economics or value of any “product” or “asset”created by any team or project that contracts Vibranium Audits to perform a security assessment This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed nor do they provide any indication of the technologies proprietors business business model or legal compliance

This report should not be used in any way to make decisions around investment or involvement with any particular project This report in no way provides investment advice nor should be leveraged as investment advice of any sort This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology

Blockchain technology and cryptographic assets present a high level of ongoing risk Vibranium Audits position is that each company and individual are responsible for their own due diligence and continuous security Vibranium Audits goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze

The assessment services provided by Vibranium Audits is subject to dependencies and under continuing development You Agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty The assessment reports could include false positives false negatives and other unpredictable results The services may access and depend upon multiple layers of third-parties

ALL SERVICES THE LABELS THE ASSESSMENT REPORT WORK PRODUCT OR OTHER MATERIALS OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED “AS IS” AND “AS AVAILABLE” AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW VIBRANIUM AUDITS HEREBY DISCLAIMS ALL WARRANTIES WHETHER EXPRESS IMPLIED STATUTORY OR OTHERWISE WITH RESPECT TO THE SERVICES ASSESSMENT REPORT OR OTHER MATERIALS WITHOUT LIMITING THE FOREGOING VIBRANIUM AUDITS SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY FITNESS FOR A PARTICULAR PURPOSE TITLE AND NON-INFRINGEMENT AND ALL WARRANTIES ARISING FROM COURSE OF DEALING USAGE OR TRADE PRACTICE WITHOUT LIMITING THE FOREGOING VIBRANIUM AUDITS MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES THE LABELS THE ASSESSMENT REPORT WORK PRODUCT OR OTHER MATERIALS OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF WILL MEET CUSTOMER’S OR ANOTHER PERSON’S REQUIREMENTS ACHIEVE ANY INTENDED RESULT BE COMPATIBLE OR WORK WITH ANY SOFTWARE SYSTEM OR OTHER SERVICES OR BE SECURE ACCURATE COMPLETE FREE OF HARMFUL CODE

OR ERROR-FREE WITHOUT LIMITATION TO THE FORGOING, VIBRANIUM AUDITS PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KINDTHAT WILL MEET CUSTOMER' S REQUIREMENTS ACHIEVE ANY INTENDED RESULTS BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE APPLICATIONS SYSTEMS OR SERVICES OPERATE WITHOUT INTERRUPTION MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED

WITHOUT LIMITING THE FOREGOING NEITHER VIBRANIUM AUDITS NOR ANY OF VIBRANIUM AUDITS AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND EXPRESS OR IMPLIED AS TO THE ACCURACY RELIABILITY ORCURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE VIBRANIUM AUDITS WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS MISTAKES OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE OF ANY NATURE WHATSOEVER RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES ASSESSMENT REPORT OR OTHER MATERIALS

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNEROR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS

THE SERVICES ASSESSMENT REPORT AND ANY OTHER MATERIALS HERE UNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT NOR MAY COPIES BE DELIVERED TO ANY OTHER PERSON WITHOUT VIBRANIUM AUDITS PRIOR WRITTEN CONSENT IN EACH INSTANCE

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES ASSESSMENT REPORT AND ANY ACCOMPANYING MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST VIBRANIUM AUDITS WITH RESPECT TO SUCH SERVICES ASSESSMENT REPORT AND ANY ACCOMPANYING MATERIALS

THE REPRESENTATIONS AND WARRANTIES OF VIBRANIUM AUDITS CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER ACCORDINGLY NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST VIBRANIUM AUDITS WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE

FOR AVOIDANCE OF DOUBT THE SERVICES INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS ORMATERIALS SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

# Vibranium | Securing the Web3 World

Vibranium Audits is a blockchain security company that was founded in 2021 by professors from the University of Greenwich and cyber-security engineers from ITI Capital. As pioneers in the field, Vibranium Audits utilizes best-in-class Formal Verification and AI technology to secure and monitor blockchains, smart contracts, and Web3 apps.