Asfalia

Security Audit

Omega Finance

11/07/22

Table of Contents

Summary

This report has been prepared for Omega Finance to discover issues and vulnerabilities in the source code of the Omega Finance project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilising Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase 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 codebase by industry experts.

The security assessment resulted in findings that ranged from high 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:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in
- public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

Project Summary

Project Name	Omega Finance
Platform	Ethereum
Language	Solidity
Codebase	Files provided
Commit	Not provided

Audit Summary

Delivery Date	11/07/2022
Audit Methodology	Static Analysis, Manual Review
Key Components	

Vulnerability Summary

Security Scoring: 85 / 100

Great

Risk Level	Total	Pending	Acknowledge	Unresolved	Partially Resolved	Resolved
Critical	0	0	0	0	0	0
High	6	0	3	0	0	3
Medium	4	0	1	2	0	1
Low	7	0	3	1	0	3
Informational	14	0	2	9	1	2
Optimization	7	0	0	1	1	5

Scope

Repository:	N/A
Commit:	N/A
Technical Documentation:	N/A
JS tests:	N/A
Contracts:	ethSwap2.sol

Project Overview

N/A

Project Architecture & Fee Models

Fees: sellTokens sends fees to the creator 0.2%

dToken is the stock placeholder token

ethSwap2 contract is where the placeholder tokens are stored and sold at the current api returned price from <polygon feed>.

Users can call sellTokens() to swap stock placeholder tokens for eth.

Users can call buyTokens() to swap eth for placeholder tokens.

Contract Dependencies

N/A

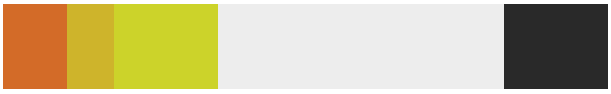
Privileged Roles

N/A

Findings

Contracts: ethSwap2.sol

● Critical	0
● High	6
● Medium	4
● Low	7
○ Informational	14
● Optimization	7



Total Issues: **38**

ID	Title	Type	Categories	Severity	Status
#1	sellTokens	Custom	Volatile Code	High	Acknowledged
#2	Compiling Issue	Custom	Volatile Code	High	Resolved
#3	Compiling Issue	Custom	Volatile Code	High	Resolved
#4	Reentrancy	SWC-107	Logical Issue	High	Acknowledged
#5	Reentracy	SWC-107	Volatile Code	High	Resolved
#6	Transaction order Dependence	SWC-114	Volatile Code	High	Acknowledged
#7	Requirement Violation	SWC-123	Coding Style	Low	Acknowledged
#8	Requirement Violation	SWC-123	Coding Style	Low	Unresolved
#9	Requirement Violation	SWC-123	Coding Style	Low	Resolved
#10	Assert Violation	SWC-110	Coding Style	Medium	Resolved
#11	Deprecated Solidity Functions	SWC-111	Volatile Code	Medium	Unresolved
#12	Typographical Error	SWC-129	Coding Style	Informational	Unresolved
#13	Typographical Error	SWC-129	Coding Style	Medium	Acknowledged
#14	Typographical Error	SWC-129	Coding Style	Informational	Acknowledged
#15	Deprecated Solidity Functions	SWC-111	Volatile style	Medium	Unresolved
#16	Presence of unused Variables	SWC-131	Coding Style	Optimization	Resolved

#17	Variables	Custom	Coding Style	Optimization	Partially Resolved
#18	Code with No Effects	SWC-135	Coding Style	Optimization	Resolved
#19	Code with No Effects	SWC-135	Coding Style	Informational	Resolved
#20	Code with No Effects	SWC-135	Coding Style	Informational	Partially Resolved
#21	Improper initialization	CWE-655	Coding Style	Low	Unresolved
#22	Improper initialization	CWE-655	Coding Style	Informational	Unresolved
#23	Improper initialization	CWE-655	Coding Style	Informational	Unresolved
#24	Improper initialization	CWE-655	Coding Style	Informational	Unresolved
#25	State Variable Default Visibility	SWC-108	Gas Optimization	Optimization	Resolved
#26	State Variable Default Visibility	SWC-108	Gas Optimization	Optimization	Resolved
#27	Incorrect Inheritance Order	SWC-125	Gas Optimization	Optimization	Unresolved
#28	State Variable Default Visibility	SWC-108	Gas Optimization	Optimization	Resolved
#29	Code with No Effects	SWC-135	Coding Style	Informational	Unresolved
#30	Incorrect Inheritance Order	SWC-125	Coding Style	Low	Acknowledged
#31	State Variable Missing	Custom	Coding Style	Low	Acknowledged
#32	Unchecked Call Return Value	SWC-104	Coding Style	Low	Unresolved
#33	Misformated	Custom	Coding Style	Informational	Acknowledged
#34	Error Message	Custom	Coding Style	Informational	Unresolved
#35	Error Message	Custom	Coding Style	Informational	Unresolved
#36	timeAmount	Custom	Coding Style	Informational	Resolved
#37	Error Message	Custom	Coding Style	Informational	Unresolved
#38	Coding Style	Custom	Coding Style	Informational	Unresolved

#1 Custom - sellTokens

	Severity	Location	Status
Volatile Code	High	Line 149	Acknowledged

Description

Contract vulnerable to having sellTokens function be unusable. If user purchased tokens and price of those tokens rises from underlying stock price increase, then users sell enough tokens the contract will not have enough ETH to payout users.

Recommendation

Team must ensure that the contract has enough ETH on it to ensure contract functions correctly.

Contract is vulnerable to not having enough ETH on hand to fulfill sellTokens()

Alleviation

N/A

#2 Custom - Compiling Issue

	Severity	Location	Status
Volatile Code	High	Line 57	Resolved

Description

string _linkJob missing parameter

Recommendation

Data location must be “memory” or “calldata” for parameter in function, but none was given.

Alleviation

“memory” parameter added.

#3 Custom - Compiling Issue

Category	Severity	Location	Status
Volatile Code	High	Line 64	Resolved

Description

string _newAPIkey && _newAPIkey2 missing parameter

Recommendation

Data location must be “memory” or “calldata” for parameter in function, but none was given.

Alleviation

“memory” parameters added.

#4 SWC-107 - Reentrancy

Category	Severity	Location	Status
Logical Issue	High	Line 74-114	Acknowledged

Description

Team process for upating pricing is unclear.

Recommendation

Create clear process for updating oracle data with a clear timer interval.

Alleviation

N/A

#5 SWC-107 - Reentrancy

Category	Severity	Location	Status
Volatile Code	High	Line 125, 139	Resolved

Description

Re-entrancy potential attack vector.

Recommendation

Add modifier/re-entrancy gaurd https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/security/ReentrancyGuard.sol

Alleviation

Re-Entrancy Guard added.

#6 SWC-114 - Transaction Order Dependence

Category	Severity	Location	Status
Volatile Code	High	Line 125, 139	Acknowledged

Description

getPrice() not internally called before sellTokens and perfectvalue is historical price

Recommendation

Re-work the sellTokens function to ensure that the perfectValue is correct at all times.

Alleviation

N/A

#7 SWC-123 - Requirement Violation

Category	Severity	Location	Status
Coding Style	Low	Line 57	Acknowledged

Description

Additional requirements missing

Recommendation

address _linkOracle, string _linkJob, uint _linkFee all have no requirements, suggest adding boundaries for each variable to avoid future issues.

Alleviation

N/A

#8 SWC-123 - Requirement Violation

Category	Severity	Location	Status
Coding Style	Low	Line 64	Unresolved

Description

Requirements missing

Recommendation

string _newAPIkey, string _newAPIkey2 all have no requirements, suggest adding boundaries for each variable to avoid future issues.

Alleviation

N/A

#9 SWC-123 - Requirement Violation

Category	Severity	Location	Status
Coding Style	Low	Line 69	Resolved

Description

Requirement incorrect

Recommendation

Potential logic issue ‘require(501 > _newFees);’ suggest updating this to require(_newFees <= 500);

Alleviation

Recommendation implemented.

#10 SWC-110 -Assert Violation

Category	Severity	Location	Status
Coding Style	Medium	Line 127	Resolved

Description

msg.value unstable

Recommendation

uint256 value = msg.value

Alleviation

Recommendation implemented.

#11 SWC-111 - Deprecated Solidity Functions

Category	Severity	Location	Status
Volatile Code	Medium	Line 133	Unresolved

Description

Use safeTransfer where possible

Recommendation

Update transfer to safeTransfer to avoid callback issues

Alleviation

N/A

#12 SWC-129 - Typographical Error

Category	Severity	Location	Status
Coding Style	Informational	Line 141	Unresolved

Description

Logic mismatch

Recommendation

require(_amount <= token.balanceOf(msg.sender));

Alleviation

N/A

#13 SWC-129 - Typographical Error

Category	Severity	Location	Status
Coding Style	Medium	Line 144	Acknowledged

Description

10**9 is used as perfectValue multiplier

Recommendation

Not sure if this is specific to the dToken or chainlink response please acknowledge this is performing as expected. If not, suggest 10**18.

Alleviation

N/A

#14 SWC-129 - Typographical Error

Category	Severity	Location	Status
Coding Style	Informational	Line 149	Acknowledged

Description

Requirement ambigious

Recommendation

etherAmount is always the same, require(etherAmount > 1 * 10**16, ‘Must sell more than 0.01 ETH at a time’);

Alleviation

N/A

#15 SWC-111 - Deprecated Solidity Functions

Category	Severity	Location	Status
Volatile Code	Medium	Line 155	Unresolved

Description

Use safeTransferFrom where possible

Recommendation

Update transfer to safeTransferFrom to avoid callback issues

Alleviation

N/A

#16 SWC-131 - Presence of unused Variables

Category	Severity	Location	Status
Coding Style	Optimization	Line 11	Resolved

Description

Unused state variable declared

Recommendation

Remove ‘name’ variable

Alleviation

‘name’ removed.

#17 Custom - Variables

Category	Severity	Location	Status
Coding Style	Optimization	Line 14 - 24	Partially Resolved

Description

Gas inefficient order of declared state variables

Recommendation

Re-order variables so all uint256's are listed together

Alleviation

Most uint256's re-arranged to be listed together, except for uint256 volume on L14. Should be moved to L15, after bytes32 jobId.

#18 SWC-135 - Code With No Effects

Category	Severity	Location	Status
Coding Style	Optimization	Line 80 & 103	Resolved

Description

Unused code

Recommendation

Remove unecessary comment code

Alleviation

Unecessary comment code removed.

#19 SWC-135 - Code With No Effects

Category	Severity	Location	Status
Coding Style	Informational	Line 18	Resolved

Description

; misformated

Recommendation

uint256 public stockprice;

Alleviation

Recommendation implemented.

#20 SWC-135 - Code With No Effects

Category	Severity	Location	Status
Coding Style	Informational	Line 25, 47, 72, 120-123	Partially Resolved

Description

unused line spacer

Recommendation

Remove unused line

Alleviation

Unused lines still in place at L70, L114-117.

#21 CWE-655 - Improper Initialization

Category	Severity	Location	Status
Coding Style	Low	Line 43	Unresolved

Description

Modifier without controller

Recommendation

Controller constructor to define creator on deployment.

Add controller options such as addCreator, transferCreator.

Alleviation

N/A

#22 CWE-655 - Improper Initialization

Category	Severity	Location	Status
Coding Style	Low	Line 49	Unresolved

Description

ConfrimedOwner custom owner contract

Recommendation

“Use newer ownable contract provided by OpenZepplin (https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/access/Ownable.sol)

Alleviation

N/A

#23 CWE-655 - Improper Initialization

Category	Severity	Location	Status
Coding Style	Informational	Line 52	Unresolved

Description

jobld set in constructor

Recommendation

joblid could be defined in contract instead of constructor

Alleviation

N/A

#24 CWE-655 - Improper Initialization

Category	Severity	Location	Status
Coding Style	Informational	Line 53	Unresolved

Description

fee set in constructor

Recommendation

fee could be defined in contract instead of constructor

Alleviation

N/A

#25 SWC-108 - State Variable Default Visibility

Category	Severity	Location	Status
Gas Optimization	Optimization	Line 57	Resolved

Description

Function is public

Recommendation

Update function to external for gas savings

Alleviation

Function visibility changed to external.

#26 SWC-108 - State Variable Default Visibility

Category	Severity	Location	Status
Gas Optimization	Optimization	Line 64	Resolved

Description

Function is public

Recommendation

Update function to external for gas savings

Alleviation

Function visibility changed to external.

#27 SWC-125 - Incorrect Inheritance Order

Category	Severity	Location	Status
Gas Optimization	Optimization	Line 57-71	Unresolved

Description

Function are not used regularly

Recommendation

Update less used function locations in the contract to after commonly used functions for gas efficeincy.

Alleviation

N/A

#28 SWC-108 - State Variable Default Visibility

Category	Severity	Location	Status
Gas Optimization	Optimization	Line 68	Resolved

Description

Function is public

Recommendation

Update function to external for gas savings

Alleviation

Function visibility changed to external.

#29 SWC-135 - Code With No Effects

Category	Severity	Location	Status
Coding Style	Informational	Line 91, 97, 114	Unresolved

Description

Misformatted

Recommendation

Remove unused spacing and code format

Alleviation

N/A

#30 SWC-125 - Incorrect Inheritance Order

Category	Severity	Location	Status
Coding Style	Low	Line 130	Acknowledged

Description

Requirement to be checked first.

Recommendation

Move requirement to the beginning of the function.

Alleviation

N/A

#31 Custom - State Variable Missing

Category	Severity	Location	Status
Coding Style	Low	Line 149	Acknowledged

Description

Mininium swap amount set as state variable.

Recommendation

For future uses, update the min swap amount to variable with getter / setter.

Alleviation

N/A

#32 SWC-104 - Unchecked Call Return Value

Category	Severity	Location	Status
Coding Style	Low	Line 156, 157	Unresolved

Description

No callback is used when sending ETH

Recommendation

Add successful result and requirement eg. require(success, “ETH_TRANSFER_FAILED”);

Alleviation

N/A

#33 Custom - Informational

Category	Severity	Location	Status
Coding Style	Low	Line 43, 64, 68	Acknowledged

Description

Misformatted.

Recommendation

Add line spacing between functions and modifiers.

Alleviation

N/A

#34 Custom - Informational

Category	Severity	Location	Status
Coding Style	Low	Line 44	Unresolved

Description

No error message on require statement.

Recommendation

Add error message.

Alleviation

N/A

#35 Custom - Informational

Category	Severity	Location	Status
Coding Style	Low	Line 69	Unresolved

Description

No error message on require statement

Recommendation

Add error message

Alleviation

N/A

#36 Custom - Informational

Category	Severity	Location	Status
Coding Style	Low	Line 83	Resolved

Description

Comment states that result is multiplied by 1000000000000000000, but timesAmount declared as 100 on Line 84.

Recommendation

Change comment to reflect function logic. (OR, they need to change timesAmount?)

Alleviation

Comment Removed.

#37 Custom - Informational

Category	Severity	Location	Status
Coding Style	Low	Line 130, 141, 150	Unresolved

Description

No error message on require statement.

Recommendation

Add error message.

Alleviation

N/A

#38 Custom - Informational

Category	Severity	Location	Status
Coding Style	Low	Line 30, 34, 40, 41, 57, 68, 127, 139, 144, 145, 146	Unresolved

Description

uint256’s referred to with uint

Recommendation

Use uint256 rather than uint for consistency.

Alleviation

N/A

Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that actagainst the nature of decentralization, such as explicit ownership or specialized access roles incombination with a mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimalEVM opcodes resulting in a reduction on the total gas cost of a transaction.

Mathematical Operations

Mathematical Operation findings relate to mishandling of math formulas, such as overflows, incorrectoperations etc.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on howblock.timestamp works.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that mayresult in a vulnerability.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

Disclaimer

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 Asfalia’s 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 Asfalia to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-freenature 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. Asfalia’s position is that each company and individual are responsible for their own due diligence and continuous security. Asfalia’s 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 Asfalia 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.

Project is potentially vulnerable to 3rd party failures of service - namely in the form of APIs providing the price for the currencies used by the project. Project could become at risk if these APIs provided incorrect pricing.

Audit does not claim to address any off-chain functions utilized by the project.

About

The firm was started by a team with over ten years of network security experience to become a global force. Our goal is to make the blockchain ecosystem as secure as possible for everyone.

With over 30 years of combined experience in the DeFi space, our team is highly dedicated to delivering a product that is as streamlined and secure as possible. Our mission is to set a new standard for security in the auditing sector, while increasing accessibility to top tier audits for all projects in the crypto space. Our dedication and passion to continuously improve the DeFi space is second to none.