



Security Audit
Futura

13/08/22

Table of Contents

Summary

Overview

Project Summary
Audit Summary
Vulnerability Summary
Scope
Project Overview

Findings

[#1 - SWC-115 : Authorization](#)
[#2 - SWC-129 : Typographical Error](#)
[#3 - SWC-104 : Unchecked Call Return Value](#)
[#4 - Custom : Call Functions](#)
[#5 - Custom : Excluded From Fees](#)
[#6 - SWC-123 : Requirement Violation](#)
[#7 - Custom : Burn Frequency](#)
[#8 - SWC-135 : Code With No Effects](#)
[#9 - SWC-135 : Code With No Effects](#)

Appendix

Disclaimer

About

Summary

This report has been prepared for Futura to discover issues and vulnerabilities in the source code of the Futura 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 Medium 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	Futura
Platform	Ethereum
Language	Solidity
Codebase	Files provided
Commit	Not provided







Audit Summary

Delivery Date	13/08/2022
Audit Methodology	Static Analysis, Manual Review
Key Components	

Vulnerability Summary

Security Scoring: 97 / 100

Excellent

Risk Level	Total	Pending	Acknowledge	Unresolved	Partially Resolved	Resolved
 Critical	0	0	0	0	0	0
 High	0	0	0	0	0	0
 Medium	0	0	0	0	0	0
 Low	3	0	3	0	0	0
 Informational	3	0	3	0	0	0
 Optimization	3	0	3	0	0	0

Scope

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

Project Overview

N/A

Project Architecture & Fee Models

N/A

Contract Dependencies




N/A

Privileged Roles

N/A

Findings

Contracts: futura.sol

 Critical	0
 High	0
 Medium	0
 Low	3
 Informational	3
 Optimization	3



Total Issues: 9

ID	Title	Type	Categories	Severity	Status
#1	Authorization	SWC-115	Volatile Code	Low	Acknowledged
#2	Typographical Error	SWC-129	Mathematical Operations	Low	Acknowledged
#3	Unchecked Call Return	SWC-104	Coding Style	Low	Acknowledged
#4	Call Functions	Custom	Centralization / Privilege	Informational	Acknowledged
#5	Excluded From Fees	Custom	Coding Style	Informational	Acknowledged
#6	Requirement Violation	SWC-123	Coding Style	Informational	Acknowledged
#7	Burn Frequency	Custom	Gas Optimization	Optimization	Acknowledged
#8	Code With No Effects	SWC-135	Gas Optimization	Optimization	Acknowledged
#9	Code With No Effects	SWC-135	Gas Optimization	Optimization	Acknowledged

#1 [SWC-115](#) - Authorization through tx.origin

Category	Severity	Location	Status
Volatile Code	Low	Line 626, 627	Acknowledged

Description

tx.origin is a global variable in Solidity which returns the address of the account that sent the transaction. A call could be made to the vulnerable contract that passes the authorization check since tx.origin returns the original sender of the transaction, which could be a malicious contract. In the Futura contract, this is only applied at launch and when transferDelayEnabled = true, plus is only used to set the _holderLastTransferTimestamp of the buyer/seller. Thus, potential impact was quite limited and is now no longer applicable.

Recommendation

Avoid use of tx.origin

Alleviation

N/A

#2 [SWC-129](#) - Typographical Error

Category	Severity	Location	Status
Mathematical Operations	Low	Line 690, 691, 692, 693, 698, 699, 700, 701, 706, 707, 708, 709, 963, 979, 980	Acknowledged

Description

Multiplication performed on result of a division. In Solidity, this can result in rounding errors.

Recommendation

Ensure that in all mathematical operations multiplications are enacted before divisions.

Alleviation

N/A

#3 SWC-104 - Unchecked Call Return Value

Category	Severity	Location	Status
Coding Style	Low	Line 750, 849, 887, 890, 923, 926	Acknowledged

Description

Return values for external contract calls to the dexRouter and lpPair contracts are ignored. This can potentially allow for these external function calls to fail within Futura functions. For example liquidity may not be created when desired, but due to the contract design subsequent functions should attempt to perform the same external function calls.

Recommendation

Ensure that the bool return values for external function calls are checked to ensure Futura is operating correctly.

Alleviation

N/A

#4 Custom - Call Functions

Category	Severity	Location	Status
Centralization / Privilege	Informational	Line 559, 578, 586, 594, 601, 868, 966	Acknowledged

Description

When calling functions with onlyOwner checks that make significant changes to the contract, events should be emitted so that external parties can more easily monitor centralization risks that these functions confer.

Recommendation

Add events to the functions listed.

Alleviation

N/A

#5 Custom - Excluded From Fees

Category	Severity	Location	Status
Coding Style	Informational	Line 613	Acknowledged

Description

`_isExcludedFromFees` is described in comment code on L349 as declaring addresses that are excluded from fees and max transaction amount. On L613, `_isExcludedFromFees` also grants ability to call `_transfer` functions when `tradingActive = false`.

Recommendation

Include this extra functionality in comment code for clarity.

Alleviation

N/A

#6 [SWC-123](#) - Requirement Violation

Category	Severity	Location	Status
Volatile Code	Informational	Line 477	Acknowledged

Description

Input variable `_lpPair` lacks a zero check. Marked informational rather than Low as the function has already been called and cannot be called again.

Recommendation

```
require(_lpPair != address(0));
```

Alleviation

N/A

#7 Custom - Burn Frequency

Category	Severity	Location	Status
Gas Optimization	Optimization	Line 334	Acknowledged

Description

`manualBurnFrequency` is declared as a state variable and never changed. It should therefore be declared as constant to save gas costs on calls.

Recommendation

```
uint256 public constant manualBurnFrequency = 30 minutes;
```

Alleviation

N/A

#8 [SWC-135](#) - Code With No Effects

Category	Severity	Location	Status
Gas Optimization	Optimization	Line 778	Acknowledged

Description

`ethBalance` does not need to be declared, logic can be moved to following line.

Recommendation

```
uint256 ethForLiquidity = address(this).balance;
```

Alleviation

N/A

#9 [SWC-135](#) - Code With No Effects

Category	Severity	Location	Status
Gas Optimization	Optimization	Line 870	Acknowledged

Description

The requirement statement contains an unnecessary check. `_percent` is a uint, meaning that by definition it is always ≥ 0 .

Recommendation

```
require(_percent <= 1000, "Must set auto LP burn percent between 0% and 10%");
```

Alleviation

N/A

Appendix

Finding Categories

Centralization / Privilege

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

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM 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, incorrect operations etc.

Logical Issue

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

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result 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.

