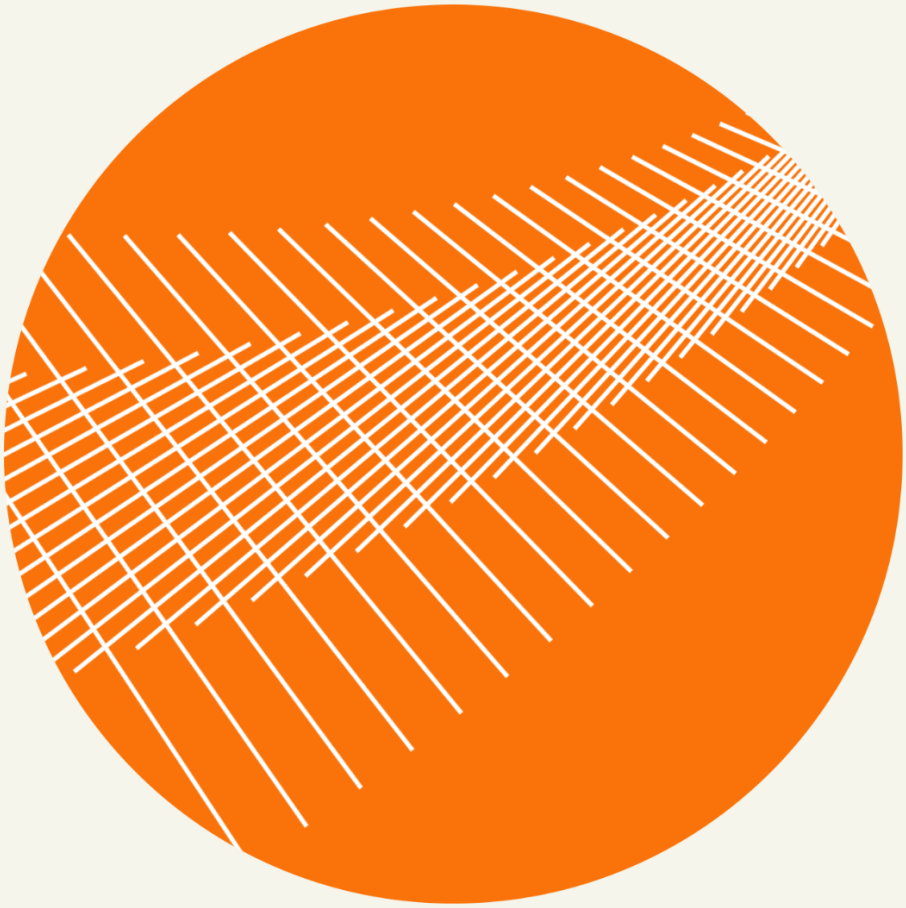
CTDSEC

YOU ARE PROTECTED



TERMS

Smart contract security audit

DEXAggregator

Table of Contents

1.0 Introduction **3**
- 1.1 Project engagement 3
- 1.2 Disclaimer 3

2.0 Coverage **4**
- 2.1 Target Code and Revision 4
- 2.2 Attacks made to the contract 5

3.0 Security Issues **7**
- 3.1 High severity issues 7
- 3.2 Medium severity issues 8
- 3.3 Low severity issues 9
- 3.4 Informational Findings 10

4.0 Testing coverage - python **11**

5.0 Annexes **17**

6.0 Summary of the audit **24**

1.0 Introduction

1.1 Project engagement

During February of 2024, VaporFiteam engaged CTDSec to audit smart contracts that they created. The engagement was technical in nature and focused on identifying security flaws in the design and implementation of the contracts. VaporFi provided CTDSec with access to their code repository and whitepaper.

1.2 Disclaimer

It should be noted that this audit is not an endorsement of the reliability or effectiveness of the contract, rather limited to an assessment of the logic and implementation. In order to ensure a secure contract that's able to withstand the network's fast-paced and rapidly changing environment, we at CTDSec recommend that Vaporfi team put in place a bug bounty program to encourage further and active analysis of the smart contract.

2.0 Coverage

2.1 Target Code and Revision

For this audit, we performed research, investigation, and review of the DEXAggregator contracts followed by issue reporting, along with mitigation and remediation instructions outlined in this report.

The following code files are considered in-scope for the review:

Source file:

https://github.com/VaporFi/dex-aggregator-v2

Fix:

Source file - [Commit - d41b2bbb4f6665b7f4908fcd712e584d111730f7]

https://github.com/VaporFi/dex-aggregator-v2/pull/25

2.2 Attacks made to the contract

In order to check for the security of the contract, we tested several attacks in order to make sure that the contract is secure and follows best practices.

№	Issue description.	Checking status
1	Compiler warnings.	PASSED
2	Race conditions and Reentrancy. Cross-function race conditions.	HIGH ISSUES
3	Possible delays in data delivery.	PASSED
4	Oracle calls.	PASSED
5	Front running.	PASSED
6	Timestamp dependence.	PASSED
7	Integer Overflow and Underflow.	PASSED
8	DoS with Revert.	PASSED
9	DoS with block gas limit.	PASSED
10	Methods execution permissions.	PASSED
11	Economy model. If application logic is based on an incorrect economic model, the application would not function correctly and participants would incur financial losses. This type of issue is most often found in bonus rewards systems, Staking and Farming contracts, Vault and Vesting contracts, etc.	PASSED
12	The impact of the exchange rate on the logic.	PASSED
13	Private user data leaks.	PASSED
14	Malicious Event log.	PASSED
15	Scoping and Declarations.	MEDIUM ISSUES
16	Uninitialized storage pointers.	PASSED

17	Arithmetic accuracy.	PASSED
18	Design Logic.	HIGH ISSUES
19	Cross-function race conditions.	PASSED
20	Safe Zeppelin module.	PASSED
21	Fallback function security.	PASSED
22	Overpowered functions / Owner privileges	MEDIUM ISSUES

3.0 Security Issues

3.1 High severity issues

1. Reentrancy in claiming function [Fixed ✅]

Contract: ReferralProgram

Functions: claimFees()

Issue: The function claimFees() is exploitable by re-entrancy attacks when a user executes a claim.

Fix: Add nonReentrant guard modifier.

2. Incorrect Contract State Initialization [Acknowledge]

Contract: DiamondInit

Issue: The contract's state is improperly initialized due to setting address[] adapters and setTrustedTokens using an Args struct during the initialization phase, leading to incorrect configuration.

Fix: Remove adapters and setTrustedtokens from the initialization and use setters to assign the variables states.

3. Reentrancy in swaps [Fixed ✅]

Contract: LegacyRouterFacet

Functions: swapNoSplit(), swapNoSplitFromAVAX(), swapNoSplitToAVAX()

Issue: The function claimFees() is exploitable by re-entrancy attacks when a user executes a swap.

Fix: Add nonReentrant guard modifier.

3.2 Medium severity issues

1. Unlimited fees [Fixed ✅]

Contract: LegacyRouterFacet.sol, RamsesV2Adapter, UniswapV3Adapter, ReservoirAdapter, UniswapV2Adapter

Functions: LegacyRouterFacet [setMintFee(), swapNoSplit(), RamsesV2Adapter, UniswapV3Adapter, ReservoirAdapter [enableFeeAmounts()], UniswapV2Adapter [Constructor: _fee].

Issue: The fees do not have any set limit, allowing if they are set wrong the contract can become a honey pot or have abusive conditions if the limit is set to 100%.

Fix: We recommend adding a limit to 10-15%.

2. Unchecked Zero-Length Arrays Leading to Function Hangs [Fixed ✅]

Contract: LegacyRouterFacet

Functions: _swapNoSplit()

Issue: The contract does not validate for zero-length in Trade Path and adapters arrays, potentially leading to adverse contract behavior and stalling function calls.

Fix: Set a require checking that the trade path and adapters can't be zero length.

3. Wrong condition assignment [Fixed ✅]

Contract: LegacyRouterFacet

Functions: findBestPathWithGas(), findBestPath()

Issue: If _maxSteps is wrongly set to 'AND', it should be 'OR'.

Fix: Change the if condition from AND to OR.

4. Redundant Function Definitions in Contracts Pose Diamond Cut Operation Risks [Fixed ✅]

Contract: DiamondManagerFacet & LegacyRouterFacet

Functions: setAdapters(), setTrustedTokens()

Issue: In both contracts, there are redundant definitions for the functions setAdapters and setTrustedTokens. This redundancy can introduce errors when attempting to incorporate their signatures into a proxy's diamond cut operation.

Fix: Consolidate the setAdapters and setTrustedTokens functions into a single contract to eliminate redundancy and prevent errors during the diamond cut process for proxy contracts. This ensures that these functionalities are distinctly managed and updated through a singular, authoritative source within the system's architecture.

3.3 Low severity issues

1. Zero address Validation Missing [Fixed ✅]

Contract: LegacyRouterFacet, BalancerV2Adapter, GmxAdapter, Curve1Adapter, Curve2Adapter, CurveMetaAdapter, CurveMetaV2Adapter, CurvePlain128Adapter, CurvePlainV2Adapter, SaddleAdapter, SaddleMetaAdapter, CurveMetaWithSwapperAdapter, DodoV1Adapter, DxSwapAdapter, LB2Adapter, VelodromeAdapter, GeodeWPAdapter, KyberElasticAdapter, UniswapV3likeAdapter, LiquidityBookAdapter, RamsesV2Adapter, UniswapV3Adapter, ReservoirAdapter, UniswapV2Adapter, WoofiAdapter, WoofiV2Adapter

Functions:

LegacyRouterFacet: Function: _swapNoSplit(), from and to.

BalancerV2Adapter, GmxAdapter: Function: constructor(), vault variable.

Contract: Curve1Adapter, Curve2Adapter, CurveMetaAdapter, CurveMetaV2Adapter,

CurvePlain128Adapter, CurvePlainV2Adapter, SaddleAdapter, SaddleMetaAdapter: Constructor() pool variable.

CurveMetaWithSwapperAdapter: constructor() metaPool(), basePool(), and swapper variables.

DodoV1Adapter: Constructor() helper variable.

DxSwapAdapter, LB2Adapter, VelodromeAdapter: Constructor() _factory variable.

GeodeWPAdapter: Constructor() _portal variable.

KyberElasticAdapter, UniswapV3likeAdapter: constructor() _quoter variable.

LiquidityBookAdapter: Constructor(), _router variable.

RamsesV2Adapter, UniswapV3Adapter, ReservoirAdapter: constructor(), _factory and _quoter variables.

UniswapV2Adapter: constructor() _factory variable.

WoofiAdapter, WoofiV2Adapter: constructor(), _pool variable

WoofiAdapter, WoofiV2Adapter: setRebateCollector(), _rebateCollector variable.

Fix: Check to implement validations in order to verify that the address is not the zero address (0x0) in the affected functions/contracts considering gas expenses.

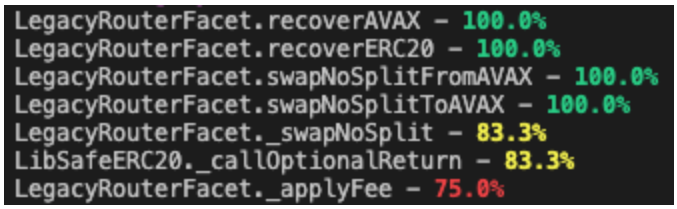
3.4 Informational Findings

No informational issues were found.

4.0 Testing coverage - python

During the testing phase, custom use cases were written to cover all the logic of contracts in python language. *Check “5 Annexes” to see the testing code.*

Legacy router:



```
tests/test_legacy_router_facet.py::test_setter RUNNING
Transaction sent: 0x06904abb038a6067de6856632078ed2d13a633a24ce2e6978ac8d9fddd8dc642
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  DiamondCutFacet.constructor confirmed   Block: 1   Gas used: 941145 (7.84%)
  DiamondCutFacet deployed at: 0x3194cBDC3dbcd3E11a07892e7bA5c3394048Cc87

Transaction sent: 0x142463d512a9616fa2bf90162e10267b3d8d3ac5feef4dfc6cb462b23b52bc8a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  DexAggregatorDiamond.constructor confirmed   Block: 2   Gas used: 293742 (2.45%)
  DexAggregatorDiamond deployed at: 0x602C71e4DAC47a042Ee7f46E0aee17F94A3bA0B6

Transaction sent: 0x8ac8dcf2e8db2620d77232d3de0979a11cfda08a05457d6f4b9a76a397a6c43d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  DiamondInit.constructor confirmed   Block: 3   Gas used: 170473 (1.42%)
  DiamondInit deployed at: 0xE7eD6747FaC5360f88a2EFC03E00d25789F69291

Transaction sent: 0x0d88d6799217a3e418f1b77dab45d7c252f7c32c8b199d63869232378f90b9f3
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  LibDiamond.constructor confirmed   Block: 4   Gas used: 72217 (0.60%)
  LibDiamond deployed at: 0x6951b5Bd815043E3F842c1b026b0Fa888Cc2DD85

Transaction sent: 0x2a2d691aa6bd3bbdff0f5756d01339c50e34d763e358b2a7cfc281342305af4c
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  AuthorizationFacet.constructor confirmed   Block: 5   Gas used: 147655 (1.23%)
  AuthorizationFacet deployed at: 0xe0aA552A10d7EC8760Fc6c246D391E698a82dDf9

Transaction sent: 0x1164d3da81ad3b853d49a0f8dffa6d154e2f4a4f40ca7306ca74acf377687664
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  DiamondLoupeFacet.constructor confirmed   Block: 6   Gas used: 405138 (3.38%)
  DiamondLoupeFacet deployed at: 0x6b4BDe1086912A6Cb24ce3dB43b3466e6c72AFd3

Transaction sent: 0x8913b04c01da927e289970741e631911075df8149c0c46dfdbb3141298598b04
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 6
  LegacyRouterFacet.constructor confirmed   Block: 7   Gas used: 2292687 (19.11%)
  LegacyRouterFacet deployed at: 0x9E4c14403d7d9A8A782044E86a93CAE09D7B2ac9

Transaction sent: 0xc75282c5aa2b19e46e397e4e7610c88fb2b7a1154f4b3488239c961c2200b1af
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 7
  OwnershipFacet.constructor confirmed   Block: 8   Gas used: 165421 (1.38%)
  OwnershipFacet deployed at: 0xcCB53c9429d32594F404d01fbe9E65ED1DCda8D9

Transaction sent: 0x478245da22f32798d52ee1b4ea252259eba32873d9edcac6ae7e25705d96b001
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 8
  Transaction confirmed   Block: 9   Gas used: 1349304 (11.24%)

Transaction sent: 0x4e99c06967904044bc9352b0473d7530324f8e84a949375fcc704daf493a3f0d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 0
  Transaction confirmed (LibDiamond__OnlyOwner: )   Block: 10   Gas used: 24163 (0.20%)

Transaction sent: 0x3c810cb3de5622442b270bec8537e24590e54b44043e43bf8c86a300f9d25e87
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 1
  Transaction confirmed (LibDiamond__OnlyOwner: )   Block: 11   Gas used: 24396 (0.20%)

Transaction sent: 0xff4ec442952c627270333567d99ef502695f493c39e5552400021bf3a32baacd
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 2
  Transaction confirmed (LibDiamond__OnlyOwner: )   Block: 12   Gas used: 25111 (0.21%)

Transaction sent: 0x24414a39190b07d217c57d55cbd7a7e998c2e2ce5e009c51fd93b8ac3f9f1fb0
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 3
  Transaction confirmed (LibDiamond__OnlyOwner: )   Block: 13   Gas used: 25132 (0.21%)

Transaction sent: 0xe309e5e5fbd128e89e18bd1052a0073f47ca42a3ee8d32ff82c50a95fbefd9a4
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 9
  Transaction confirmed   Block: 14   Gas used: 46267 (0.39%)

Transaction sent: 0x4057d916e327bfd72b1a8451a4ed280b5f8417e580a4c12eadf71270d7e16835
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 10
  Transaction confirmed   Block: 15   Gas used: 68875 (0.57%)
```

```
tests/test_legacy_router_facet.py::test_setter PASSED
tests/test_legacy_router_facet.py::test_recover_erc20 RUNNING
Transaction sent: 0x9c852cf1cdac282f8efcca04d87754d7711fee30e471b4d388ae1d9ad520098d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 12
  ERC20Mock.constructor confirmed   Block: 17   Gas used: 619873 (5.17%)
  ERC20Mock deployed at: 0x2c15A315610Bfa5248E4CbCbd693320e9D8E03Cc

Transaction sent: 0xd81e4b40f986c3fc401782e8041a99ccdbe4e647a23ac34c4225d0eed790743b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 13
  ERC20Mock.mint confirmed   Block: 18   Gas used: 65637 (0.55%)

Transaction sent: 0x2c3de5dc9f925c819685d6db52c1522b09aaa27a4aa55536686cc3e6f51602b7
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 14
  DiamondCutFacet.constructor confirmed   Block: 19   Gas used: 941145 (7.84%)
  DiamondCutFacet deployed at: 0xe65A7a341978d59d40d30FC23F5014FACB4f575A

Transaction sent: 0xe53d5ad1c88d2a1a75f9340f7b8a81443bfc1d3e4cebd78f78ad714abbe062d3
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 15
  DexAggregatorDiamond.constructor confirmed   Block: 20   Gas used: 293742 (2.45%)
  DexAggregatorDiamond deployed at: 0x30375B532345B01cB8c2AD12541b09E9Aa53A93d

Transaction sent: 0xceb8adc3f357d38f76f442ae701f633ab53688e376f82e61afa113880a91a0ab
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 16
  DiamondInit.constructor confirmed   Block: 21   Gas used: 170473 (1.42%)
  DiamondInit deployed at: 0x26f15335BB1C6a4C0B660eDd694a0555A9F1cce3

Transaction sent: 0xa11a6251f0b199c1d4173a661bb92e5f95ec9e3f80d65b4d8c3e31fbc2bb216d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 17
  LibDiamond.constructor confirmed   Block: 22   Gas used: 72217 (0.60%)
  LibDiamond deployed at: 0xFbD588c72B438faD4Cf7cD879c8F730Faa213Da0

Transaction sent: 0x97fb6152e87fa6a9535efd77b94b0044882b3cbd1e343b8a8c77945213b81a28
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 18
  AuthorizationFacet.constructor confirmed   Block: 23   Gas used: 147655 (1.23%)
  AuthorizationFacet deployed at: 0xed00238F9A0F7b4d93842033cdF56cCB32C781c2

Transaction sent: 0x01b55f747c9eea877b7fca9cca4e21554ab81c3e63d952a97e77c31b21c3a49d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 19
  DiamondLoupeFacet.constructor confirmed   Block: 24   Gas used: 405138 (3.38%)
  DiamondLoupeFacet deployed at: 0xDae02e4fE488952cFB8c95177154D188647a0146

Transaction sent: 0x8fc6c6ff5c09798d75378a476e9c611895ad599c7785d8d8b5bd3af8e8c3a7a6
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 20
  LegacyRouterFacet.constructor confirmed   Block: 25   Gas used: 2292687 (19.11%)
  LegacyRouterFacet deployed at: 0xdCF93F11ef216cEC9C07fd31dD801c9b2b39Afb4

Transaction sent: 0x5321ff15228903a1a083e40b85df1a496c0fe816a5865b30ec5d20a1e3aa0f3f
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 21
  OwnershipFacet.constructor confirmed   Block: 26   Gas used: 165421 (1.38%)
  OwnershipFacet deployed at: 0xBcb61491F1859f53438918F1A5aFCA542Af9D397

Transaction sent: 0xb7e41596a6e1bba7a7be24404a31e0787e1df6c54c78d8474572dfc05dd1120d
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 22
  Transaction confirmed   Block: 27   Gas used: 1349304 (11.24%)

Transaction sent: 0x868cff73ea135197de941b7fd9f378b3bdcbef7eaa18d0d00248fc9fde82d971
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 4
  Transaction confirmed (LibDiamond__OnlyOwner: )   Block: 28   Gas used: 24623 (0.21%)

Transaction sent: 0x20d15602738e09c36390ffdb5974455a2e5f0d2700c2cf3cfca8e23fc0d6a4f1
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 23
  Transaction confirmed (LegacyRouterFacet__NothingToRecover: )   Block: 29   Gas used: 24655 (0.21%)

Transaction sent: 0xa96d45732b176a2d042bca95ce678c7678a4c63625528448cf13210c6ad62493
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 24
  Transaction confirmed (SafeERC20: low-level call failed)   Block: 30   Gas used: 27915 (0.23%)
```

```
Transaction sent: 0x584b7e397e9dc79cb803affda61065f56374e0ee890c73007b0247b619f72657
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 25
  ERC20Mock.transfer confirmed   Block: 31   Gas used: 50970 (0.42%)

Transaction sent: 0xfb4d5d8cf7717f64f273d50b2f77b7b65578bbe16fd57ebfd89b0704602d1230
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 26
  Transaction confirmed   Block: 32   Gas used: 42236 (0.35%)

tests/test_legacy_router_facet.py::test_recover_erc20 PASSED
tests/test_legacy_router_facet.py::test_recover_avax RUNNING
Transaction sent: 0x67461c42fbd0fc472a2fa1116d2cf9eef3ced9c35ea824b0f62e3a704ad37941
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 27
  ERC20Mock.constructor confirmed   Block: 33   Gas used: 619873 (5.17%)
  ERC20Mock deployed at: 0x832698Daec363C9A7aB036C224Af5B21280b3AC6

Transaction sent: 0x1e6968e7784c0c70e44bf9925f295597efd529e1609b7ada71bddc0a5e4de03c
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 28
  ERC20Mock.mint confirmed   Block: 34   Gas used: 65637 (0.55%)

Transaction sent: 0x3ded013b9b5403fbf0c88643c90e4919945329582741 27e84b880d1ebf67e89b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 29
  DiamondCutFacet.constructor confirmed   Block: 35   Gas used: 941145 (7.84%)
  DiamondCutFacet deployed at: 0x42E8D004c84E6B5Bad559D3b5CE7947AADb9E0bc

Transaction sent: 0xaba8962068bee24391f3d4360f540782ffe315fc8465b0409a749dbf0f878dc1
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 30
  DexAggregatorDiamond.constructor confirmed   Block: 36   Gas used: 293742 (2.45%)
  DexAggregatorDiamond deployed at: 0xF06D5f5BfFFCB6a52c84cfebc03AD35637728E73

Transaction sent: 0x72ee10637bcd5d403e865c7649d69af25b0b773d82b53bc889f968be1c279ece
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 31
  DiamondInit.constructor confirmed   Block: 37   Gas used: 170473 (1.42%)
  DiamondInit deployed at: 0x82c83b7f88aef2eD99d4869D547b6ED28e69C8df

Transaction sent: 0x448ffa540e7b40c5ad0980d230fe25464c7ba0931f31c72a99ec73153df35a2a
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 32
  LibDiamond.constructor confirmed   Block: 38   Gas used: 72217 (0.60%)
  LibDiamond deployed at: 0x724Ca58E1e6e64BFB1E15d7Eec0fe1E5f581c7bD

Transaction sent: 0x84e6d9f3a35e201d5784c1022d1aa562a4f24e75127b0e55d9aa9c959bd91b2c
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 33
  AuthorizationFacet.constructor confirmed   Block: 39   Gas used: 147655 (1.23%)
  AuthorizationFacet deployed at: 0x34b97ffa01dc0DC959c5f1176273D0de3be914C1

Transaction sent: 0x377e822e1b582ec84455e1a03b38b94aa8c5b0a715bba58400ca2f8854440bcd
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 34
  DiamondLoupeFacet.constructor confirmed   Block: 40   Gas used: 405138 (3.38%)
  DiamondLoupeFacet deployed at: 0xc830Ad2FDfCC2f368fE5DeC93b1Dc72ecABb3691

Transaction sent: 0x2aacbdd0da3ef8020e618fe10dc0512139a412d03944be6c8e3e6c6557d0a728
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 35
  LegacyRouterFacet.constructor confirmed   Block: 41   Gas used: 2292687 (19.11%)
  LegacyRouterFacet deployed at: 0xbc8eCccb89650c3E796e803CB009BF9b898CB359

Transaction sent: 0x21b3b555444aebeea700010185c2d63b32c5617a1e5e8f83fb64ca31ef8c8305
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 36
  OwnershipFacet.constructor confirmed   Block: 42   Gas used: 165421 (1.38%)
  OwnershipFacet deployed at: 0x741e3E1f81041c62C2A97d0b6E567AcaB09A6232

Transaction sent: 0xa847feb40f75aef155a6c41cee860ee9b0f46c0bb210981a467d3d37cee60a84
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 37
  Transaction confirmed   Block: 43   Gas used: 1349316 (11.24%)

Transaction sent: 0xfbed4dbf0cce9f4b41d71759b64d278bf69c347b1852eaef92b92e52a7e8cfc5
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 5
  Transaction confirmed (LibDiamond__OnlyOwner: )   Block: 44   Gas used: 24128 (0.20%)
```

```
Transaction sent: 0x48126e80e848c566f4f090d2e983d9e1c051d51d877b0159c7b669b872e7501e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 38
  Transaction confirmed (LegacyRouterFacet__NothingToRecover: )   Block: 45   Gas used: 24160 (0.20%)

Transaction sent: 0x70a3eee0e3e31530f871f530a109d6fa0205c51eaf0ceabeeda53a57e9a4d2d0
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 39
  Transaction confirmed (reverted)   Block: 46   Gas used: 31683 (0.26%)

Transaction sent: 0xf0359903db03aa2ab0f27e8f65826e661ae8c374babd02819350138a419317bf
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 40
  Transaction confirmed   Block: 48   Gas used: 33118 (0.28%)

tests/test_legacy_router_facet.py::test_recover_avax PASSED
tests/test_legacy_router_facet.py::test_swap RUNNING
Transaction sent: 0xfb37c35c2c5eda2639566542d8c0a92808817a263799e28ad9392a76e93e039b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 41
  WETH9.constructor confirmed   Block: 49   Gas used: 476546 (3.97%)
  WETH9 deployed at: 0x5847798CE8c89e3Fff59AE5fA30BEC0d406b5687

Transaction sent: 0x085bcfb9d1cc5de1a908068c8960ff0a4f512e142cf0f106aeebcef23cde4937
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 42
  ERC20Mock.constructor confirmed   Block: 50   Gas used: 619921 (5.17%)
  ERC20Mock deployed at: 0x0C60536783db9ED5A2B216970B10FF2243d317dD

Transaction sent: 0x65462c0dbc91fd57fa7f75c6155ef47f71e23f9ea90ee2d7c8fdfcbe7ceedaef
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 43
  ERC20Mock.mint confirmed   Block: 51   Gas used: 65637 (0.55%)

Transaction sent: 0x3b967f8fd660e348092cf0bf46979ad3719f8802885734835eb6a274d9b930d6
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 44
  WAvaxAdapter.constructor confirmed   Block: 52   Gas used: 1192351 (9.94%)
  WAvaxAdapter deployed at: 0xaA7e46855e0506401214c9b1C35f3d889669609e

Transaction sent: 0x3f06afed6f18c5ae6e9127c767c6a6ce9192653ba31cb1c2f390184e645c6a5b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 45
  DiamondCutFacet.constructor confirmed   Block: 53   Gas used: 941145 (7.84%)
  DiamondCutFacet deployed at: 0xa1910C6e0Fbd0E38cfBabAeC5D1C1D539F81CC63

Transaction sent: 0xe0464f65fec2fedb18cbc86155e43b23a63756f65a181d7fda5b852381f8f873
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 46
  DexAggregatorDiamond.constructor confirmed   Block: 54   Gas used: 293742 (2.45%)
  DexAggregatorDiamond deployed at: 0x8BE16B874F47371C42498b499837Ed87D7661E86

Transaction sent: 0x0e079b93c53c6845b848621c145871c64a337b90fec4f1aa1a38b509760b1047
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 47
  DiamondInit.constructor confirmed   Block: 55   Gas used: 170473 (1.42%)
  DiamondInit deployed at: 0x1aa96efd2B002541E830CB7f60e473AA24e31F9A

Transaction sent: 0x3a39cdd93a31ffdd40a24a6cf5bb7927fd4300c15ec8e1cd451b26ef197747e8
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 48
  LibDiamond.constructor confirmed   Block: 56   Gas used: 72217 (0.60%)
  LibDiamond deployed at: 0x8b1B440724DCe2EE9779B58af841Ec59F545838B

Transaction sent: 0x5f7c038d1049cb0cd3c0d1a2e62cd4a6f3283b2eecf6ff402aa126f13bacafa9
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 49
  AuthorizationFacet.constructor confirmed   Block: 57   Gas used: 147655 (1.23%)
  AuthorizationFacet deployed at: 0xC6D563d5c2243b27e7294511063f563ED701EA2C

Transaction sent: 0xe1ef84ee8b3a931f6eb8c550e6c7e85379f0c9b80912faea8b193589146390e7
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 50
  DiamondLoupeFacet.constructor confirmed   Block: 58   Gas used: 405138 (3.38%)
  DiamondLoupeFacet deployed at: 0xD537bF4b795b7D07Bd5F4bAf7017e3ce8360B1DE
```

```
Transaction sent: 0xa3c1a6b6f0c4717c2a9c3d45a48ca325763211f524b5de9765c2b2dafa747b56
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 51
  LegacyRouterFacet.constructor confirmed   Block: 59   Gas used: 2292687 (19.11%)
  LegacyRouterFacet deployed at: 0x70bC6D873D110Da59a9c49E7485a27B0F605E5db

Transaction sent: 0x7387da1d6a9bcf9f9954ec2a8dfa37980a37568468728f6953dfe9cb876ff06e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 52
  OwnershipFacet.constructor confirmed   Block: 60   Gas used: 165421 (1.38%)
  OwnershipFacet deployed at: 0xFb7C5F938835aE34aF48c278C6763E134907Acdb

Transaction sent: 0x9a26bf5b82860031254552205a20ae11136eacb4bb7bf0467812f630a234c988
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 53
  Transaction confirmed   Block: 61   Gas used: 1349304 (11.24%)

Transaction sent: 0xf0805d53582133eac04fec5e3f6c0fd12fcf34c540d02465ffcbd60a4f344411
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 54
  ERC20Mock.approve confirmed   Block: 62   Gas used: 44160 (0.37%)

Transaction sent: 0xc21f075b8aeb5ce1a89b2f18dc60d01ba4086d9398065035f819400e66a1a08e
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 55
  Transaction confirmed   Block: 63   Gas used: 124036 (1.03%)

Transaction sent: 0x3180e712b124263c20f03a3440c2a281693d0833a3df4ea03f70b9ed48ab4e62
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 56
  Transaction confirmed (LegacyRouterFacet__InvalidPath: )   Block: 64   Gas used: 27440 (0.23%)

Transaction sent: 0xc0eb914791f4e93a1a49fdb22f122d166e05f76ca68e73b53ba4d9a96d2725a2
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 57
  Transaction confirmed   Block: 65   Gas used: 121934 (1.02%)

Transaction sent: 0x3662cac96f971a7d32ad1c911d8032e6ceede66377c76012ccd944e0f6b8ec83
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 58
  Transaction confirmed (LegacyRouterFacet__InvalidPath: )   Block: 66   Gas used: 27711 (0.23%)

Transaction sent: 0x88ef3fed7ba9a1a4dd2444d677919db54707a510cb2998a8810ad2223995147b
  Gas price: 0.0 gwei   Gas limit: 12000000   Nonce: 59
  Transaction confirmed   Block: 67   Gas used: 85394 (0.71%)

tests/test_legacy_router_facet.py::test_swap PASSED
```

Referral:

```
OwnableUpgradeable._checkOwner - 100.0%
ReferralProgram.setDiamondAggregator - 100.0%
ReferralProgram.setUSDCLimit - 100.0%
ReferralProgram.claimFees - 87.5%
ReferralProgram.swap - 80.4%
OwnableUpgradeable.transferOwnership - 75.0%
ReferralProgram.initialize - 75.0%
ReferralProgram._deleteToken - 65.0%
```

5.0 Annexes

Testing code:

Legacy Router:

```
from brownie import (
    reverts,
    LegacyRouterFacet
)

from brownie.network.contract import Contract

from scripts.helpful_scripts import (
    ZERO_ADDRESS,
    get_account,
)

from scripts.deploy import (
    deploy_dex_aggregator_diamond,
    deploy_mock_erc20,
    deploy_wavax_adapter,
    deploy_weth,
)

def test_setter(only_local):
    # arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)
    diamond = deploy_dex_aggregator_diamond(owner=owner,
fee_claimer="0x12733BfE3B8559d2D425BFBCabcc9A976896657E",
w_native="0x9bE5Ff131fAC20c3De24B58f25BE067B5Ef536A0")
    diamond_manager = Contract.from_abi("LegacyRouterFacet",
diamond.address, LegacyRouterFacet.abi)

    # assert
    with reverts("LibDiamond__OnlyOwner: "):
        diamond_manager.setMinFee(1, {"from": other})
    with reverts("LibDiamond__OnlyOwner: "):
```

```
        diamond_manager.setFeeClaimer(extra, {"from": other})
    with reverts("LibDiamond__OnlyOwner: "):
        diamond_manager.setAdapters([extra], {"from": other})
    with reverts("LibDiamond__OnlyOwner: "):
        diamond_manager.setTrustedTokens([extra], {"from": other})

    tx = diamond_manager.setMinFee(10, {"from": owner})
    assert tx.events['UpdatedMinFee'][0]['oldMinFee'] == 0
    assert tx.events['UpdatedMinFee'][0]['newMinFee'] == 10
    assert diamond_manager.trustedTokensCount() == 0
    assert diamond_manager.adaptersCount() == 0
    diamond_manager.setAdapters([extra], {"from": owner})
    diamond_manager.setTrustedTokens([extra], {"from": owner})
    assert diamond_manager.trustedTokensCount() == 1
    assert diamond_manager.adaptersCount() == 1

def test_recover_erc20(only_local):
    # arrange
    owner = get_account(0)
    other = get_account(1)
    erc20 = deploy_mock_erc20(owner, "test", "test")
    erc20.mint(owner, 10e18)
    diamond = deploy_dex_aggregator_diamond(owner=owner,
fee_claimer="0x12733BfE3B8559d2D425BFBCabcc9A976896657E",
w_native="0x9bE5Ff131fAC20c3De24B58f25BE067B5Ef536A0")
    diamond_manager = Contract.from_abi("LegacyRouterFacet",
diamond.address, LegacyRouterFacet.abi)

    with reverts("LibDiamond__OnlyOwner: "):
        diamond_manager.recoverERC20(erc20.address, 0, {"from": other})
    with reverts("LegacyRouterFacet__NothingToRecover: "):
        diamond_manager.recoverERC20(erc20.address, 0, {"from": owner})

    with reverts("SafeERC20: low-level call failed"): # Nothing to recover
        diamond_manager.recoverERC20(erc20.address, 1e18, {"from": owner})
    erc20.transfer(diamond_manager.address, 5e18, {"from": owner})
    tx = diamond_manager.recoverERC20(erc20.address, 1e18, {"from": owner})
    assert tx.events['Recovered'][0]['asset'] == erc20.address
    assert tx.events['Recovered'][0]['amount'] == 1e18

def test_recover_avax(only_local):
    # arrange
```

```
    owner = get_account(0)
    other = get_account(1)
    erc20 = deploy_mock_erc20(owner, "test", "test")
    erc20.mint(owner, 10e18)
    diamond = deploy_dex_aggregator_diamond(owner=owner,
fee_claimer="0x12733BfE3B8559d2D425BFBCabcc9A976896657E",
w_native="0x9bE5Ff131fAC20c3De24B58f25BE067B5Ef536A0")
    diamond_manager = Contract.from_abi("LegacyRouterFacet",
diamond.address, LegacyRouterFacet.abi)

    # asserts
    with reverts("LibDiamond__OnlyOwner: "):
        diamond_manager.recoverAVAX(0, {"from": other})
    with reverts("LegacyRouterFacet__NothingToRecover: "):
        diamond_manager.recoverAVAX(0, {"from": owner})

    with reverts(): # Nothing to recover
        diamond_manager.recoverAVAX(1e18, {"from": owner})
    other.transfer(diamond_manager.address, "1 ether")
    tx = diamond_manager.recoverAVAX(1e18, {"from": owner})
    assert tx.events['Recovered'][0]['asset'] == ZERO_ADDRESS
    assert tx.events['Recovered'][0]['amount'] == 1e18

def test_swap(only_local):
    # arrange
    owner = get_account(0)
    other = get_account(1)
    fee_claimer = get_account(5)

    weth = deploy_weth(owner)
    erc20 = deploy_mock_erc20(owner, "token1", "token1")
    erc20.mint(owner, 10e18)
    adapter = deploy_wavax_adapter(owner)
    diamond = deploy_dex_aggregator_diamond(owner=owner,
fee_claimer=fee_claimer, w_native=weth.address)
    diamond_manager = Contract.from_abi("LegacyRouterFacet",
diamond.address, LegacyRouterFacet.abi)

    erc20.approve(diamond_manager.address, 5e18, {"from": owner})
    tx = diamond_manager.swapNoSplit([1e18, 1e18, [erc20.address,
weth.address], [adapter.address]], other, 100, {"from": owner})
```

```
    with reverts("LegacyRouterFacet__InvalidPath: "):
        diamond_manager.swapNoSplitFromAVAX([1e18, 1e18, [erc20.address,
weth.address], [adapter.address]], other, 100, {"from": owner})

    tx = diamond_manager.swapNoSplitFromAVAX([1e18, 1e18, [weth.address,
erc20.address], [adapter.address]], other, 100, {"from": owner, "value":
1e18})

    with reverts("LegacyRouterFacet__InvalidPath: "):
        diamond_manager.swapNoSplitToAVAX([1e18, 1e18, [weth.address,
erc20.address], [adapter.address]], other, 100, {"from": owner})
    diamond_manager.swapNoSplitToAVAX([1e18, 1e18, [erc20.address,
weth.address], [adapter.address]], other, 0, {"from": owner})
```

Referral Program:

```
from brownie import (
    reverts,
)

from scripts.helpful_scripts import (
    ZERO_ADDRESS,
    get_account,
    lib_percentages
)

from scripts.deploy import (
    deploy_mock_aggregator,
    deploy_mock_erc20,
    deploy_referral_program
)


def test_set_diamond_aggregator(only_local):
    # arrange
    owner = get_account(0)
    other = get_account(1)
    aggregator = deploy_mock_aggregator(owner)
    erc20 = deploy_mock_erc20(owner, "test", "test")
    referral = deploy_referral_program(owner)
```

```
    referral.initialize(aggregator.address, owner, 5e18, erc20.address,
{"from": owner})

    new_aggregator = deploy_mock_aggregator(owner)
    # asserts
    with reverts():
        referral.setDiamondAggregator(new_aggregator.address, {"from":
other})
    with reverts("ReferralProgram__AddressIsZero: "):
        referral.setDiamondAggregator(ZERO_ADDRESS, {"from": owner})

    assert referral.getDexAggregator() == aggregator.address
    referral.setDiamondAggregator(new_aggregator.address, {"from": owner})
    assert referral.getDexAggregator() == new_aggregator.address

def test_set_usdc_limit(only_local):
    # arrange
    owner = get_account(0)
    other = get_account(1)
    aggregator = deploy_mock_aggregator(owner)
    erc20 = deploy_mock_erc20(owner, "test", "test")
    referral = deploy_referral_program(owner)

    referral.initialize(aggregator.address, owner, 5e18, erc20.address,
{"from": owner})

    # asserts
    with reverts():
        referral.setUSDCLimit(10e18, {"from": other})
    with reverts("ReferralProgram__USDCLimitParamZero: "):
        referral.setUSDCLimit(0, {"from": owner})

    assert referral.getUSDCLimit() == 5e18
    referral.setUSDCLimit(10e18, {"from": owner})
    assert referral.getUSDCLimit() == 10e18

def test_swap(only_local):
    # arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)
    aggregator = deploy_mock_aggregator(owner)
```

```
    usdc = deploy_mock_erc20(owner, "usdc", "usdc")
    token = deploy_mock_erc20(owner, "token", "token")
    referral = deploy_referral_program(owner)

    referral.initialize(aggregator.address, owner, 5e18, usdc.address,
{"from": owner})

    with reverts("ReferralProgram_UnauthorizedAccess: "):
        referral.swap(lib_percentages(10e18, 100), other, token.address,
extra, {"from":owner})

    with reverts("ReferralProgram__InconsistentParams: "):
        referral.swap(0, other, token.address, extra,
{"from":aggregator.address})

    with reverts("ReferralProgram__InconsistentParams: "):
        referral.swap(lib_percentages(10e18, 100), other, token.address,
ZERO_ADDRESS, {"from":aggregator.address})

    tx = referral.swap(lib_percentages(5e18, 100), other, token.address,
extra, {"from":aggregator.address})
    tx = referral.swap(lib_percentages(5e18, 100), extra, token.address,
other, {"from":aggregator.address})

def test_claim_fees(only_local):
    # arrange
    owner = get_account(0)
    other = get_account(1)
    extra = get_account(2)
    aggregator = deploy_mock_aggregator(owner)
    usdc = deploy_mock_erc20(owner, "usdc", "usdc")
    token = deploy_mock_erc20(owner, "token", "token")
    referral = deploy_referral_program(owner)

    referral.initialize(aggregator.address, owner, 5e18, usdc.address,
{"from": owner})

    with reverts("ReferralProgram__EarningsZero: "):
        referral.claimFees(token.address, {"from": owner})

    tx = referral.swap(lib_percentages(5e18, 100), other, token.address,
extra, {"from":aggregator.address})
```

```
    with reverts():
        referral.claimFees(token.address, {"from": other})

    token.mint(referral.address, 1e18)
    tx = referral.claimFees(token.address, {"from": other})
    assert tx.events['Transfer'][0]['from'] == referral.address
    assert tx.events['Transfer'][0]['to'] == other
```

6.0 Summary of the audit

High and medium vulnerabilities are identified that can affect the funds managed by the contract and their behavior, which is why they must be resolved before being deployed. After a thorough review by the development team, the vulnerabilities are found and it is **safe to deploy**.

Vulnerability Level	Total	Pending	Not Apply	Acknowledged	Resolved
High	3			1	2
Medium	3				3
Low	1				1
Informational	0				