Amun

Rebalance Manager

SMART CONTRACT AUDIT

21.07.2021

Made in Germany by Chainsulting.de

Table of contents

1. Disclaimer

The audit makes no statements or warrantees about utility of the code, safety of the code, suitability of the business model, investment advice, endorsement of the platform or its products, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bug free status. The audit documentation is for discussion purposes only.

The information presented in this report is confidential and privileged. If you are reading this report, you agree to keep it confidential, not to copy, disclose or disseminate without the agreement of Amun Limited. If you are not the intended receptor of this document, remember that any disclosure, copying or dissemination of it is forbidden.

Major Versions / Date	Description
0.1 (18.07.2021)	Layout
0.2 (19.07.2021)	Test Deployment
0.5 (20.07.2021)	Automated Security Testing Manual Security Testing
0.6 (20.07.2021)	Testing SWC Checks
0.7 (21.07.2021)	Verify Claims
0.9 (21.07.2021)	Summary and Recommendation
1.0 (21.07.2021)	Final document
1.1 (TBA)	Added deployed contract addresses

2. About the Project and Company

Company address:

Amun Limited
Suite 202 2nd Floor
Eden Plaza, Eden Island
PO Box 1352, Mahe
Seychelles

Website: https://amun.com

Twitter: twitter.com/amuntokens

Medium: https://medium.com/amun-tokens

Telegram: https://t.me/AmunTokens

Discord: https://discord.gg/MVJmyCzahH

Github: https://github.com/amun/contracts

2.1 Project Overview

Amun DeFi is a set of Ethereum-based tools that allow users to maximize their returns across major DeFi platforms through lending and staking by helping them choose the best product at any given time. Amun DeFi supports integrations with popular protocols representing tens of billions of total locked value (TVL). From purchasing major cryptocurrencies quickly and with ease, to swapping between popular ERC-20 tokens, all the way to lending tokens at the best available rates, Amun DeFi makes it simple to monitor and interact with a user's holdings on one secure, easy-to-use platform.

Through gateways with various lending platforms and AMMs, Amun DeFi enables users to execute their desired investment strategy with relative ease. Because there are so many different platforms that users can use to achieve maximum returns, the average DeFi user might not know precisely where to begin. Further, many retail investors will be unable to implement their intended strategy due to a lack of knowledge or technical sophistication.

Amun's goal is to reduce the fragmentation of this space by offering access to all the top protocols in one platform. They intend to offer the following features to the DeFi community :

1. Dashboard that monitors your holdings and depicts your exposure across the various protocols ;
2. Ability to invest in the top liquidity pools and lending protocols ;
3. Offers our own products (autopilot strategies) that automatically moves your funds to the protocol offering the best yields ;
4. Seamlessly diversify your investments through our basket tokens.

3. Vulnerability & Risk Level

Risk represents the probability that a certain source-threat will exploit vulnerability, and the impact of that event on the organization or system. Risk Level is computed based on CVSS version 3.0.

Level	Value	Vulnerability	Risk (Required Action)
Critical	9 – 10	A vulnerability that can disrupt the contract functioning in a number of scenarios, or creates a risk that the contract may be broken.	Immediate action to reduce risk level.
High	7 – 8.9	A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.	Implementation of corrective actions as soon as possible.
Medium	4 – 6.9	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.	Implementation of corrective actions in a certain period.
Low	2 – 3.9	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
Informational	0 – 1.9	A vulnerability that have informational character but is not effecting any of the code.	An observation that does not determine a level of risk

4. Auditing Strategy and Techniques Applied

Throughout the review process, care was taken to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices. To do so, reviewed line-by-line by our team of expert pentesters and smart contract developers, documenting any issues as there were discovered.

4.1 Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:
   i. Review of the specifications, sources, and instructions provided to Chainsulting to make sure we understand the size, scope, and functionality of the smart contract.
   ii. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
   iii. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Chainsulting describe.
2. Testing and automated analysis that includes the following:
   i. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
   ii. Symbolic execution, which is analysing a program to determine what inputs causes each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, actionable recommendations to help you take steps to secure your smart contracts.

4.2 Used Code from other Frameworks/Smart Contracts (direct imports)

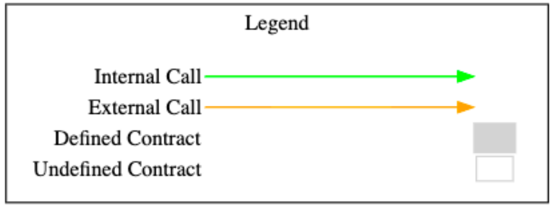
Dependency / Import Path	Source
@openzeppelin/contracts/token/ERC20/IERC20.sol	https://github.com/OpenZeppelin/openzeppelin-contracts/blob/v3.2.1-solc-0.7/contracts/token/ERC20/IERC20.sol
@pangolindex/exchange-contracts/contracts/pangolin-periphery/interfaces/IPangolinRouter.sol	https://github.com/pangolindex/exchange-contracts/blob/main/contracts/pangolin-periphery/interfaces/IPangolinRouter.sol
@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router02.sol	https://github.com/Uniswap/uniswap-v2-periphery/blob/master/contracts/interfaces/IUniswapV2Router02.sol
@uniswap/v3-periphery/contracts/interfaces/IQuoterV2.sol	https://github.com/Uniswap/uniswap-v3-periphery/blob/main/contracts/interfaces/IQuoterV2.sol
@uniswap/v3-periphery/contracts/interfaces/ISwapRouter.sol	https://github.com/Uniswap/uniswap-v3-periphery/blob/main/contracts/interfaces/ISwapRouter.sol

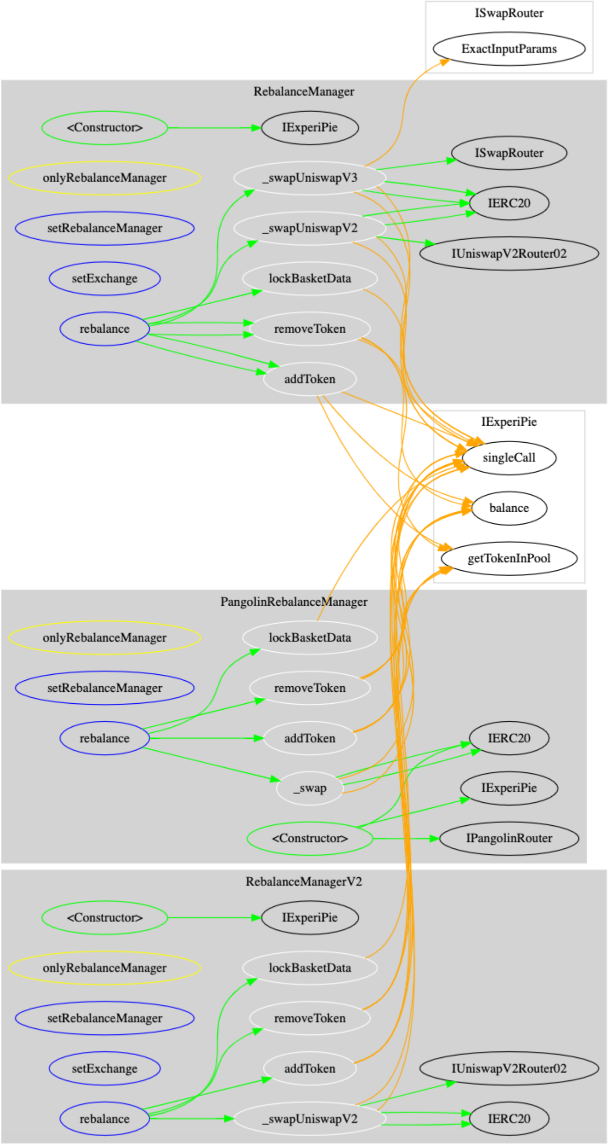
4.3 Tested Contract Files

The following are the MD5 hashes of the reviewed files. A file with a different MD5 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different MD5 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review

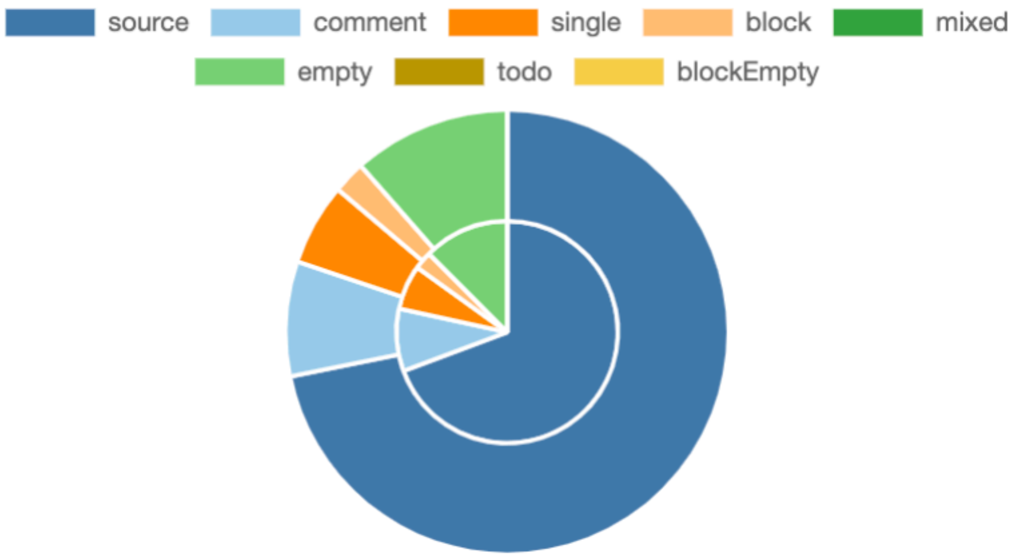
File	Fingerprint (MD5)
./basket/callManagers/RebalanceManager.sol	ce4c5ef5ed27aa8704eeec013a4fd7d1
./basket/callManagers/RebalanceManagerV2.sol	591fa6ff4a96433970878fc32473c2c1
./basket/callManagers/PangolinRebalanceManager.sol	ca6dc8c0a3918af1cab36ba4c2929ca6

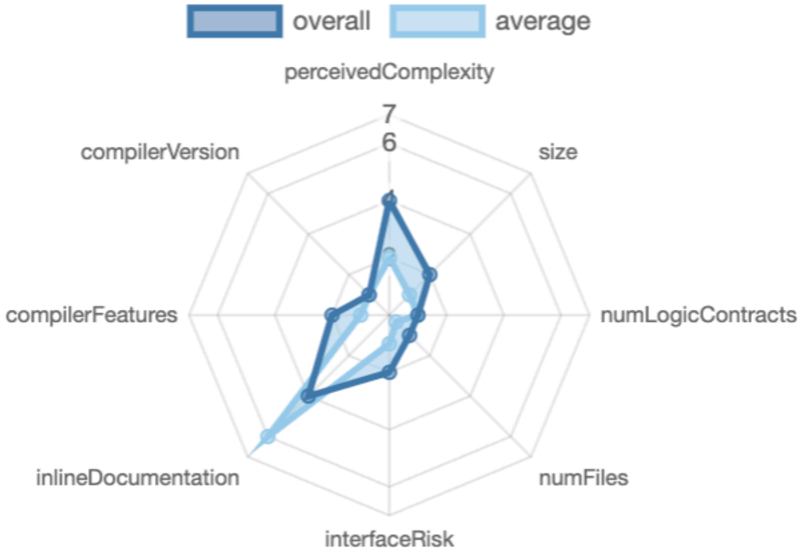
4.4 Metrics / CallGraph





4.5 Metrics / Source Lines & Risk





4.6 Metrics / Capabilities

Solidity Versions observed	🧪 Experimental Features	💰 Can Receive Funds	🖥 Uses Assembly	💣 Has Destroyable Contracts
`^0.7.5`	`ABIEncoderV2`		**** (0 asm blocks)	

📤 Transfers ETH	⚡ Low-Level Calls	👥 DelegateCall	🧮 Uses Hash Functions	🔖 ECRecover	🌀 New/Create/Create2

Exposed Functions

This section lists functions that are explicitly declared public or payable. Please note that getter methods for public stateVars are not included.

🌐Public	💰Payable
8	0

External	Internal	Private	Pure	View
8	37	0	0	0

StateVariables

Total	🌐Public
10	9

4.7 Metrics / Source Unites in Scope

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
📝	basket/callManagers/RebalanceManager.sol	1		254	233	180	22	92	🧪
📝	basket/callManagers/PangolinRebalanceManager.sol	1		181	164	125	18	78	🧪
📝	basket/callManagers/RebalanceManageV2.sol	1		179	163	122	16	64	🧪
📝	**Totals**	**3**		**614**	**560**	**427**	**56**	**234**	🧪

Legend: [➖]

- **Lines**: total lines of the source unit
- **nLines**: normalized lines of the source unit (e.g. normalizes functions spanning multiple lines)
- **nSLOC**: normalized source lines of code (only source-code lines; no comments, no blank lines)
- **Comment Lines**: lines containing single or block comments
- **Complexity Score**: a custom complexity score derived from code statements that are known to introduce code complexity (branches, loops, calls, external interfaces, ...)

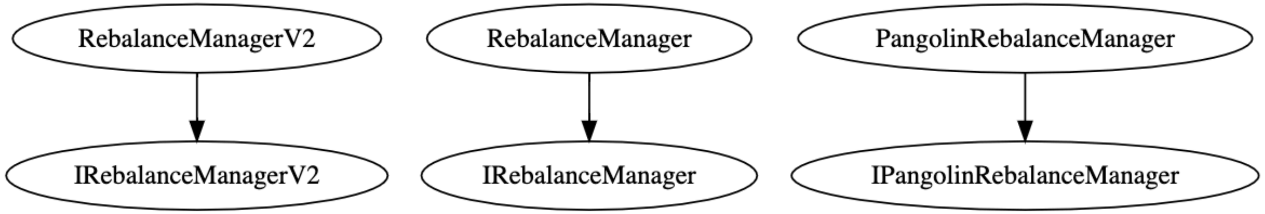
5. Scope of Work

The Amun Team provided us with the files that needs to be tested. The scope of the audit are the Rebalance Manager contracts.

The team put forward the following assumptions regarding the security, usage of the contracts:

- The smart contract is coded according to the newest standards and in a secure way
- Compare codebase with https://github.com/pie-dao/PieVaults/tree/master/contracts and point out the diffs and check them twice.
- List all audits from Pie-DAO and check if the findings effecting Amun and if they are fixed.

The main goal of this audit was to verify these claims. The auditors can provide additional feedback on the code upon the client's request.



5.1 Manual and Automated Vulnerability Test

CRITICAL ISSUES

During the audit, Chainsulting‘s experts found **no Critical issues** in the code of the smart contract.

HIGH ISSUES

During the audit, Chainsulting’s experts found **no High issues** in the code of the smart contract.

MEDIUM ISSUES

During the audit, Chainsulting’s experts found **no Medium issues** in the code of the smart contract

LOW ISSUES

5.1.1 No package.json

Severity: LOW
Status: ACKNOWLEDGED
File(s) affected: Package.json

Attack / Description	Code Snippet	Result/Recommendation

In the current implementation, and repository is no package.json included to correctly install the needed dependencies.	Line: 5-8 (PangolinRebalanceManager.sol)
	`import {`
	`    IPangolinRouter`
	`} from "@pangolindex/exchange-contracts/contracts/pangolin-periphery/interfaces/IPangolinRouter.sol";`
	`import "@openzeppelin/contracts/token/ERC20/IERC20.sol";`
		We highly recommend using npm and package.json in order to guarantee that the right libraries and versions are installed.
		"@openzeppelin/contracts": "^3.2.1-solc-0.7",
		"@pangolindex/exchange-contracts": "1.0.1",
		"@uniswap/v3-periphery": "1.1.1",
		"@uniswap/v2-periphery": "1.1.0",

	Line: 5-10 (RebalanceManager.sol)
	`import "@openzeppelin/contracts/token/ERC20/IERC20.sol";`
	
	`import "@uniswap/v3-periphery/contracts/interfaces/ISwapRouter.sol";`
	`import "@uniswap/v3-periphery/contracts/interfaces/IQuoterV2.sol";`
	
	`import "@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router02.sol";`
	
	Line 5-7 (RebalanceManagerV2.sol)
	`import "@openzeppelin/contracts/token/ERC20/IERC20.sol";`
	
	`import "@uniswap/v2-periphery/contracts/interfaces/IUniswapV2Router02.sol";`	

5.2. SWC Attacks

ID	Title	Relationships	Test Result
SWC-131	Presence of unused variables	CWE-1164: Irrelevant Code	✅
SWC-130	Right-To-Left-Override control character (U+202E)	CWE-451: User Interface (UI) Misrepresentation of Critical Information	✅
SWC-129	Typographical Error	CWE-480: Use of Incorrect Operator	✅
SWC-128	DoS With Block Gas Limit	CWE-400: Uncontrolled Resource Consumption	✅
SWC-127	Arbitrary Jump with Function Type Variable	CWE-695: Use of Low-Level Functionality	✅
SWC-125	Incorrect Inheritance Order	CWE-696: Incorrect Behavior Order	✅
SWC-124	Write to Arbitrary Storage Location	CWE-123: Write-what-where Condition	✅
SWC-123	Requirement Violation	CWE-573: Improper Following of Specification by Caller	✅

ID	Title	Relationships	Test Result
SWC-122	Lack of Proper Signature Verification	CWE-345: Insufficient Verification of Data Authenticity	✅
SWC-121	Missing Protection against Signature Replay Attacks	CWE-347: Improper Verification of Cryptographic Signature	✅
SWC-120	Weak Sources of Randomness from Chain Attributes	CWE-330: Use of Insufficiently Random Values	✅
SWC-119	Shadowing State Variables	CWE-710: Improper Adherence to Coding Standards	✅
SWC-118	Incorrect Constructor Name	CWE-665: Improper Initialization	✅
SWC-117	Signature Malleability	CWE-347: Improper Verification of Cryptographic Signature	✅
SWC-116	Timestamp Dependence	CWE-829: Inclusion of Functionality from Untrusted Control Sphere	✅
SWC-115	Authorization through tx.origin	CWE-477: Use of Obsolete Function	✅
SWC-114	Transaction Order Dependence	CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')	✅

ID	Title	Relationships	Test Result
SWC-113	DoS with Failed Call	CWE-703: Improper Check or Handling of Exceptional Conditions	✅
SWC-112	Delegatecall to Untrusted Callee	CWE-829: Inclusion of Functionality from Untrusted Control Sphere	✅
SWC-111	Use of Deprecated Solidity Functions	CWE-477: Use of Obsolete Function	✅
SWC-110	Assert Violation	CWE-670: Always-Incorrect Control Flow Implementation	✅
SWC-109	Uninitialized Storage Pointer	CWE-824: Access of Uninitialized Pointer	✅
SWC-108	State Variable Default Visibility	CWE-710: Improper Adherence to Coding Standards	✅
SWC-107	Reentrancy	CWE-841: Improper Enforcement of Behavioral Workflow	✅
SWC-106	Unprotected SELFDESTRUCT Instruction	CWE-284: Improper Access Control	✅
SWC-105	Unprotected Ether Withdrawal	CWE-284: Improper Access Control	✅
SWC-104	Unchecked Call Return Value	CWE-252: Unchecked Return Value	✅

ID	Title	Relationships	Test Result
SWC-103	Floating Pragma	CWE-664: Improper Control of a Resource Through its Lifetime	X
SWC-102	Outdated Compiler Version	CWE-937: Using Components with Known Vulnerabilities	✅
SWC-101	Integer Overflow and Underflow	CWE-682: Incorrect Calculation	✅
SWC-100	Function Default Visibility	CWE-710: Improper Adherence to Coding Standards	✅

5.3. Associated audits with the forked codebase

Mixbytes	https://github.com/pie-dao/audits/blob/main/Mixbytes%20-%20ExperiPie_Smart_Contrac%202020-12-11.pdf	Dec. 2020

5.4. Verify Claims

5.4.1 The smart contract is coded according to the newest standards and in a secure way

Status: tested and verified ✅

5.4.2 Compare codebase with https://github.com/pie-dao/PieVaults/tree/master/contracts and point out the diffs and check them twice.

Status: tested and verified ✅

```
diff -qr ./contracts-master/contracts/basket ./PieVaults-master/contracts | sort

Files ./contracts-master/contracts/basket/callManagers/LendingManager/LendingLogicAave.sol and ./PieVaults-
master/contracts/callManagers/LendingManager/LendingLogicAave.sol differ
Files ./contracts-master/contracts/basket/callManagers/LendingManager/LendingLogicAaveV2.sol and
./PieVaults-master/contracts/callManagers/LendingManager/LendingLogicAaveV2.sol differ
Files ./contracts-master/contracts/basket/callManagers/LendingManager/LendingLogicCompound.sol and
./PieVaults-master/contracts/callManagers/LendingManager/LendingLogicCompound.sol differ
Files ./contracts-master/contracts/basket/callManagers/LendingManager/LendingManager.sol and ./PieVaults-
master/contracts/callManagers/LendingManager/LendingManager.sol differ
Files ./contracts-master/contracts/basket/callManagers/LendingManager/LendingRegistry.sol and ./PieVaults-
master/contracts/callManagers/LendingManager/LendingRegistry.sol differ
Files ./contracts-master/contracts/basket/callManagers/LendingManager/StakeSushi.sol and ./PieVaults-
master/contracts/callManagers/LendingManager/StakeSushi.sol differ
```

```
Files ./contracts-master/contracts/basket/callManagers/LendingManager/StakingLogicYGov.sol and ./PieVaults-
master/contracts/callManagers/LendingManager/StakingLogicYGov.sol differ
Files ./contracts-master/contracts/basket/callManagers/RSIManager.sol and ./PieVaults-
master/contracts/callManagers/RSIManager.sol differ
Files ./contracts-master/contracts/basket/facets/Basket/BasketFacet.sol and ./PieVaults-
master/contracts/facets/Basket/BasketFacet.sol differ
Files ./contracts-master/contracts/basket/facets/Basket/LibBasketStorage.sol and ./PieVaults-
master/contracts/facets/Basket/LibBasketStorage.sol differ
Files ./contracts-master/contracts/basket/facets/Call/CallFacet.sol and ./PieVaults-
master/contracts/facets/Call/CallFacet.sol differ
Files ./contracts-master/contracts/basket/facets/Call/LibCallStorage.sol and ./PieVaults-
master/contracts/facets/Call/LibCallStorage.sol differ
Files ./contracts-master/contracts/basket/facets/ERC20/ERC20Facet.sol and ./PieVaults-
master/contracts/facets/ERC20/ERC20Facet.sol differ
Files ./contracts-master/contracts/basket/facets/ERC20/LibERC20.sol and ./PieVaults-
master/contracts/facets/ERC20/LibERC20.sol differ
Files ./contracts-master/contracts/basket/facets/ERC20/LibERC20Storage.sol and ./PieVaults-
master/contracts/facets/ERC20/LibERC20Storage.sol differ
Files ./contracts-master/contracts/basket/facets/shared/Access/CallProtection.sol and ./PieVaults-
master/contracts/facets/shared/Access/CallProtection.sol differ
Files ./contracts-master/contracts/basket/facets/shared/Reentry/LibReentryProtectionStorage.sol and
./PieVaults-master/contracts/facets/shared/Reentry/LibReentryProtectionStorage.sol differ
Files ./contracts-master/contracts/basket/facets/shared/Reentry/ReentryProtection.sol and ./PieVaults-
master/contracts/facets/shared/Reentry/ReentryProtection.sol differ
Files ./contracts-master/contracts/basket/factories/PieFactoryContract.sol and ./PieVaults-
master/contracts/factories/PieFactoryContract.sol differ
Files ./contracts-master/contracts/basket/interfaces/IAToken.sol and ./PieVaults-
master/contracts/interfaces/IAToken.sol differ
Files ./contracts-master/contracts/basket/interfaces/IATokenV2.sol and ./PieVaults-
master/contracts/interfaces/IATokenV2.sol differ
Files ./contracts-master/contracts/basket/interfaces/IAaveLendingPool.sol and ./PieVaults-
master/contracts/interfaces/IAaveLendingPool.sol differ
Files ./contracts-master/contracts/basket/interfaces/IAaveLendingPoolV2.sol and ./PieVaults-
master/contracts/interfaces/IAaveLendingPoolV2.sol differ
```

```
Files ./contracts-master/contracts/basket/interfaces/IBasketFacet.sol and ./PieVaults-
master/contracts/interfaces/IBasketFacet.sol differ
Files ./contracts-master/contracts/basket/interfaces/ICToken.sol and ./PieVaults-
master/contracts/interfaces/ICToken.sol differ
Files ./contracts-master/contracts/basket/interfaces/ICallFacet.sol and ./PieVaults-
master/contracts/interfaces/ICallFacet.sol differ
Files ./contracts-master/contracts/basket/interfaces/IERC20Facet.sol and ./PieVaults-
master/contracts/interfaces/IERC20Facet.sol differ
Files ./contracts-master/contracts/basket/interfaces/IExperiPie.sol and ./PieVaults-
master/contracts/interfaces/IExperiPie.sol differ
Files ./contracts-master/contracts/basket/interfaces/ILendingLogic.sol and ./PieVaults-
master/contracts/interfaces/ILendingLogic.sol differ
Files ./contracts-master/contracts/basket/interfaces/IPriceReferenceFeed.sol and ./PieVaults-
master/contracts/interfaces/IPriceReferenceFeed.sol differ
Files ./contracts-master/contracts/basket/interfaces/ISynthetix.sol and ./PieVaults-
master/contracts/interfaces/ISynthetix.sol differ
Files ./contracts-master/contracts/basket/interfaces/IXSushi.sol and ./PieVaults-
master/contracts/interfaces/IXSushi.sol differ
Files ./contracts-master/contracts/basket/interfaces/IYVault.sol and ./PieVaults-
master/contracts/interfaces/IYVault.sol differ
Files ./contracts-master/contracts/basket/test/DiamondFactoryContract.sol and ./PieVaults-
master/contracts/test/DiamondFactoryContract.sol differ
Files ./contracts-master/contracts/basket/test/ERC20FactoryContract.sol and ./PieVaults-
master/contracts/test/ERC20FactoryContract.sol differ
Files ./contracts-master/contracts/basket/test/ManualPriceReferenceFeed.sol and ./PieVaults-
master/contracts/test/ManualPriceReferenceFeed.sol differ
Files ./contracts-master/contracts/basket/test/MockAToken.sol and ./PieVaults-
master/contracts/test/MockAToken.sol differ
Files ./contracts-master/contracts/basket/test/MockATokenV2.sol and ./PieVaults-
master/contracts/test/MockATokenV2.sol differ
Files ./contracts-master/contracts/basket/test/MockAaveLendingPool.sol and ./PieVaults-
master/contracts/test/MockAaveLendingPool.sol differ
Files ./contracts-master/contracts/basket/test/MockAaveLendingPoolV2.sol and ./PieVaults-
master/contracts/test/MockAaveLendingPoolV2.sol differ
```

```
Files ./contracts-master/contracts/basket/test/MockCToken.sol and ./PieVaults-
master/contracts/test/MockCToken.sol differ
Files ./contracts-master/contracts/basket/test/MockLendingLogic.sol and ./PieVaults-
master/contracts/test/MockLendingLogic.sol differ
Files ./contracts-master/contracts/basket/test/MockSynthetix.sol and ./PieVaults-
master/contracts/test/MockSynthetix.sol differ
Files ./contracts-master/contracts/basket/test/MockToken.sol and ./PieVaults-
master/contracts/test/MockToken.sol differ
Files ./contracts-master/contracts/basket/test/MockXSushi.sol and ./PieVaults-
master/contracts/test/MockXSushi.sol differ
Files ./contracts-master/contracts/basket/test/MockYVault.sol and ./PieVaults-
master/contracts/test/MockYVault.sol differ
Only in ./PieVaults-master/contracts/callManagers/LendingManager: DepositLogicDecimalWrapper.sol
Only in ./PieVaults-master/contracts/interfaces: IDecimalWrapper.sol
Only in ./PieVaults-master/contracts/test: MockDecimalWrapper.sol
Only in ./contracts-master/contracts/basket/callManagers: PangolinRebalanceManager.sol
Only in ./contracts-master/contracts/basket/callManagers: RebalanceManageV2.sol
Only in ./contracts-master/contracts/basket/callManagers: RebalanceManager.sol
Only in ./contracts-master/contracts/basket/interfaces: IPangolinRebalanceManager.sol
Only in ./contracts-master/contracts/basket/interfaces: IRebalanceManager.sol
Only in ./contracts-master/contracts/basket/interfaces: IRebalanceManagerV2.sol
Only in ./contracts-master/contracts/basket/interfaces: IUniswapV2Factory.sol
Only in ./contracts-master/contracts/basket/interfaces: IUniswapV2Pair.sol
Only in ./contracts-master/contracts/basket/interfaces: IUniswapV2Router01.sol
Only in ./contracts-master/contracts/basket/interfaces: IUniswapV2Router02.sol
Only in ./contracts-master/contracts/basket/test: MockPangolinRouter.sol
Only in ./contracts-master/contracts/basket/test: WETH9.sol
Only in ./contracts-master/contracts/basket: SingleTokenJoin.sol
```

5.4.3 List all audits from Pie-DAO and check if the findings effecting Amun and if they are fixed.

Status: tested and verified ✅

6. Executive Summary

Two (2) independent Chainsulting experts performed an unbiased and isolated audit of the smart contract codebase. The final debriefs took place on the July 21, 2021. The overall code quality of the project is very good, and the modifications of the forked Pie-DAO Vault contracts decrease the attack surface, as they are already used since month and have been audited.

The main goal of the audit was to verify the claims regarding the security of the smart contract and the functions. During the audit, no critical issues were found after the manual and automated security testing and the claims been successfully verified.

7. Deployed Smart Contract

PENDING

Contract is deployed here:

…