SMART CONTRACT AUDIT REPORT

Customer: *Coin98*

Date: *May 19th, 2022*

TABLE OF CONTENT

SUMMARY 3
DISCLAIMER 3
SCOPE OF WORK 4
RISK ASSESSMENT LEVEL DEFINITIONS 5
AUDIT RESULTS 5
FINDINGS 6
List of Findings 6
Details 7
APPENDIX : VULNERALBILITIES CHECKLIST 8
CONCLUSION 8

SUMMARY

99

No vulnerabilities were found or all

detected ones have been resolved

DISCLAIMER

This document may contain confidential information about Aegis's Customer, the intellectual property of the Customer, and information about potential vulnerabilities and methods of their exploitation.

This audit focused on identifying security flaws in the code of the Customer's Smart Contract. The scope of the audit is limited to the source code files provided to Aegis.

SCOPE of WORK

Name	COIN98
Platform	ETHEREUM
Language	SOLIDITY
Methods	AUTOMATION SCAN, ARCHITECTURE REVIEW, FUNCTIONAL TESTING, MANUAL CODE REVIEW
Technical Documentation	N0
Repository	HTTPS://GITHUB.COM/COIN98/COIN98-DOLLAR-MINT-BURN (#4A9281F) HTTPS://GITHUB.COM/COIN98/COIN98-SUPERLINK (#CDBBCC8)
Timeline	MAY 12TH 2022 – MAY 17TH 2022

File Checksums

COIN98-DOLLAR-MINT-BURN-MAIN

No.	Hash	Name
1	2342E243BF3A7A2D67420543D92D396FA8D00E82	Coin98DollarMintBurn.sol

COIN98-SUPERLINK-MAIN

No.	Hash	Name
1	85D89A73440EE214AA758C968A3871BE917A75CF	SuperLinkPartner.sol
2	02BCB47F021017774112F945988C750E1564A8BE	SuperLink.sol

RISK ASSESSMENT LEVEL DEFINITIONS

Severity Level	Description
High	A vulnerability that can disrupt the contract's functioning creating data loss and financial risks to the contract, needs to be fixed immediately.
Medium	A vulnerability that can put parts of users' sensitive information at risk, would be detrimental to the Customer's reputation if exploited and needs to be fixed with priority.
Low	A vulnerability that is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low impact for the sake of the Customer.
Info	An issue that can be considered less important as it does not have significant impacts.

AUDIT RESULTS

According to the assessment, the Customer's Smart Contract get **99/100** score in terms of security level

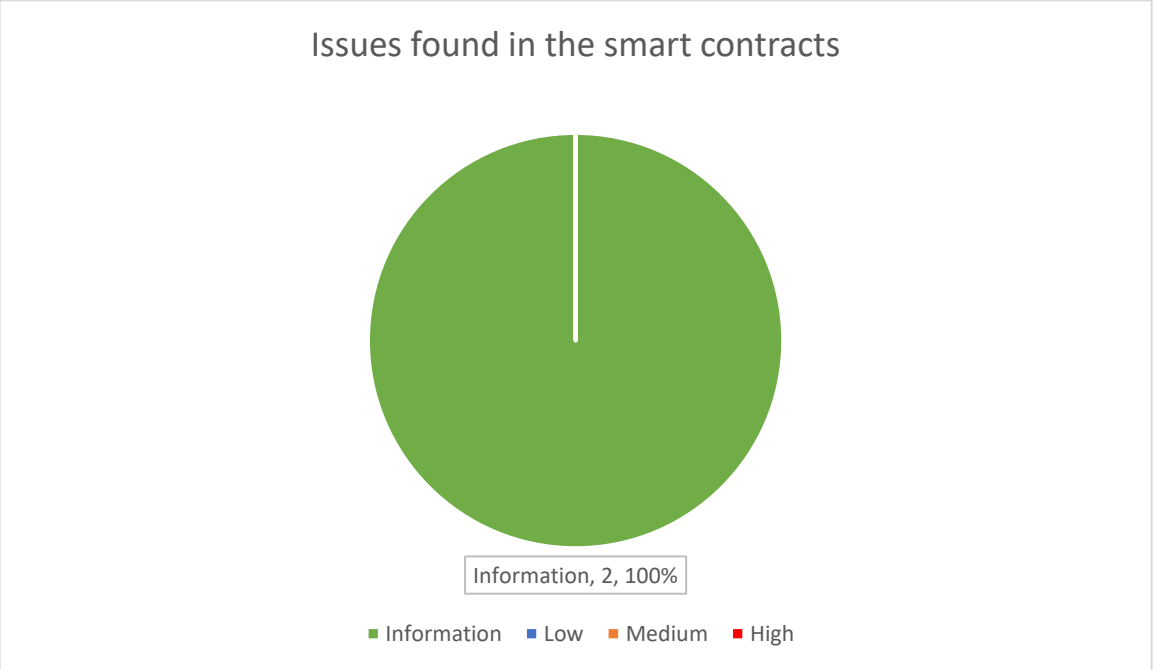
Rate	Description
95-100	No vulnerabilities were found or all detected ones have been resolved
70-94	Unresolved Low-level vulnerabilities have been found or existed
35-69	Unresolved Medium-level vulnerabilities have been found or existed
0-34	Unresolved High-level vulnerabilities have been found or existed

FINDINGS

List of Findings

Our team performed an analysis of code functionality, manual audit, and automated checks All issues found during automated analysis were manually reviewed, and important vulnerabilities are listed below:

ID	Risk Level	Name	Amount	Status
AE1	Info	Unlocked Pragma	2	Resolved



Details

Unlocked Pragma

- **Description**

Contracts should be deployed with the same compiler version and flags that they have been thoroughly tested. Locking the pragma helps to ensure that contracts do not accidentally get deployed using.

- **Impact**

```
1   // SPDX-License-Identifier: MIT
2   pragma solidity >=0.8.0;
```

An outdated compiler version that might introduce bugs that affect the contract system negatively.

- **Recommendation**

Lock the pragma version and also consider known bugs (https://github.com/ethereum/solidity/releases) for the chosen compiler version.

Pragma statements can be allowed to float when a contract is intended for consumption by other developers, as in the case with contracts in a library or EthPM package. Otherwise, the developer would need to manually update the pragma in order to compile locally.

- **Location**

coin98-superlink-main:: All Contract
coin98-dollar-mint-burn-main:: All Contract

Appendix : VULNERABILITIES CHECKLIST

The following is the list of commonly known vulnerabilities that were considered during the audit:

- ☹ Integer Overflow and Underflow
- ☹ Timestamp Dependence
- ☹ Race Conditions
- ☹ Transaction-Ordering Dependence
- ☹ DoS with (Unexpected) revert
- ☹ DoS with Block Gas Limit
- ☹ Gas Usage, Gas Limit and Loops
- ☹ Redundant fallback function
- ☹ Unsafe internal call
- ☹ Reentrancy
- ☹ Explicit visibility of functions state variables (external, internal, private and public)
- ☹ Business Logic Flaws
- ☹ Bad Randomness
- ☹ Arithmetic operations

CONCLUSION

Aegis completed the assessment using manual, static, and dynamic analysis techniques.
The reports should not be considered as investment advice under any circumstances.

Please feel free to direct any questions on this assessment to:
audit@agis.so