AegisX

Smart Contract Audits | KYC

Security Assessment

AntNetworX Token

October 24, 2022

Table of Contents

Assessment Summary

This report has been prepared for AntNetworX Token on the Binance Smart Chain network. AegisX provides both client-centered and user-centered examination of the smart contracts and their current status when applicable. This report represents the security assessment made to find issues and vulnerabilities on the source code along with the current liquidity and token holder statistics of the protocol.

A comprehensive examination has been performed, utilizing Cross Referencing, Static Analysis, In-House Security Tools, and line-by-line Manual Review.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Inspecting liquidity and holders statistics to inform the current status to both users and client when applicable.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Verifying contract functions that allow trusted and/or untrusted actors to mint, lock, pause, and transfer assets.
- 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.

Technical Findings Summary

Classification of Risk

Severity	Description
Critical	Risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
Major	Risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
Medium	Risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform
Minor	Risks can be any of the above but on a smaller scale. They generally do not compromise the overall integrity of the Project, but they may be less efficient than other solutions.
Informational	Errors are often recommended to improve the code's style or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

Findings

Severity	Found	Pending	Resolved
Critical	0	0	0
Major	0	0	0
Medium	0	0	0
Minor	2	2	0
Informational	0	0	0
Total	2	2	0

Project Overview

Token Summary

Parameter	Result
Address	0x9186359F82c3c0Cc005A0b3563Dc4Ccd2627D82A
Name	AntNetworX
Token Tracker	AntNetworX (ANTX)
Decimals	18
Supply	115,823,755
Platform	Binance Smart Chain
compiler	v0.8.17+commit.8df45f5f
Contract Name	AntNetworx
Optimization	5000
LicenseType	MIT
Language	Solidity
Codebase	https://bscscan.com/address/0x9186359F82c3c0Cc005A0b3563Dc4Ccd2627D82A#code
Payment Tx	

Project Overview

Risk Analysis Summary

Parameter	Result
Buy Tax	1.6%
Sale Tax	1.6%
Is honeypot?	Clean
Can edit tax?	Yes
Is anti whale?	Yes
Is blacklisted?	Yes
Is whitelisted?	Yes
Holders	Clean
Security Score	95/100
Auditor Score	95/100
Confidence Level	High

The following quick summary it's added to the project overview; however, there are more details about the audit and its results. Please read every detail.

Main Contract Assessed Contract Name

Name	Contract	Live
AntNetworX	0x9186359F82c3c0Cc005A0b3563Dc4Ccd2627D82A	Yes

TestNet Contract Assessed Contract Name

Name	Contract	Live
AntNetworX	0x72a9583E2180f5e6eF1fDF22F9a64F89a467a403	Yes

Solidity Code Provided

SolID	File Sha-1	FileName
AntNetworx	3b433f46c5cef90408dbedcc97638dec1ea94a08	AntNetworx.sol

Mint Check

The project owners of AntNetworX do not have a mint function in the contract, owner cannot mint tokens after initial deploy.

The Project has a Total Supply of 115,823,755 and cannot mint any more than the Max Supply.

Mint Notes:

Auditor Notes: A mint function was not found.

Project Owner Notes:

Fees Check

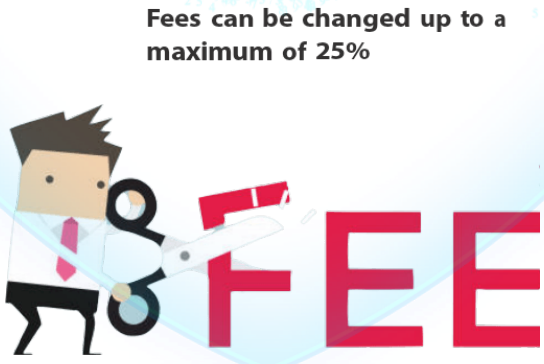
The project owners of AntNetworX do not have the ability to set fees higher than 25% .

The team May have fees defined; however, they can't set those fees higher than 25% or may not be able to configure the same.

Tax Fee Notes:

Auditor Notes: The contract currently has 1.6% buy and 1.6% sale taxes, the owner cannot set roundtrip tax higher than 10%.

Project Owner Notes: .



Blacklist Check

The project owners of AntNetworX have the ability to Blacklist holders from transferring their tokens.

We recommend the Team be careful with a blacklist function as this can prevent a holder from buying/selling/transferring their assets. Malicious or compromised owners can trap contracts relying on tokens with a blacklist

Blacklist Notes:

Auditor Notes: The contract has antiSnipe functions, the owner can blacklist an address as a sniper account.

Project Owner Notes:

MaxTx Check

The Project Owners of AntNetworX can set max tx amount.

The ability to set MaxTx can be used as a bad actor, this can limit the ability of investors to sell their tokens at any given time if is set too low.

We recommend the project to set MaxTx to Total Supply or simiar to avoid swap or transfer from failures.

MaxTX Notes:

Auditor Notes: There is a Max Wallet function in the contract.

Project Owner Notes:

Pause Trade Check

The Project Owners of AntNetworX don't have the ability to stop or pause trading.

The Team has done a great job to avoid stop trading, and investors has the ability to trade at any given time without any problems

Pause Trade Notes:

Auditor Notes: The project Owner can enable trading, however trading cannot be disabled once started.

Project Owner Notes:

Contract Ownership

The contract ownership of AntNetworX is not currently renounced. The ownership of the contract grants special powers to the protocol creators, making them the sole addresses that can call sensible ownable functions that may alter the state of the protocol.

The current owner is the address 0x6cb58284478dbc6bbd32065ec5ee7d6aee5fbf70 which can be viewed:
HERE

The owner wallet has the power to call the functions displayed on the privileged functions chart below, if the owner's wallet is compromised, they could exploit these privileges.

We recommend the team renounce ownership at the right time, if possible, or gradually migrate to a timelock with governing functionalities regarding transparency and safety considerations.

We recommend the team use a Multisignature Wallet if the contract is not going to be renounced; this will give the team more control over the contract.

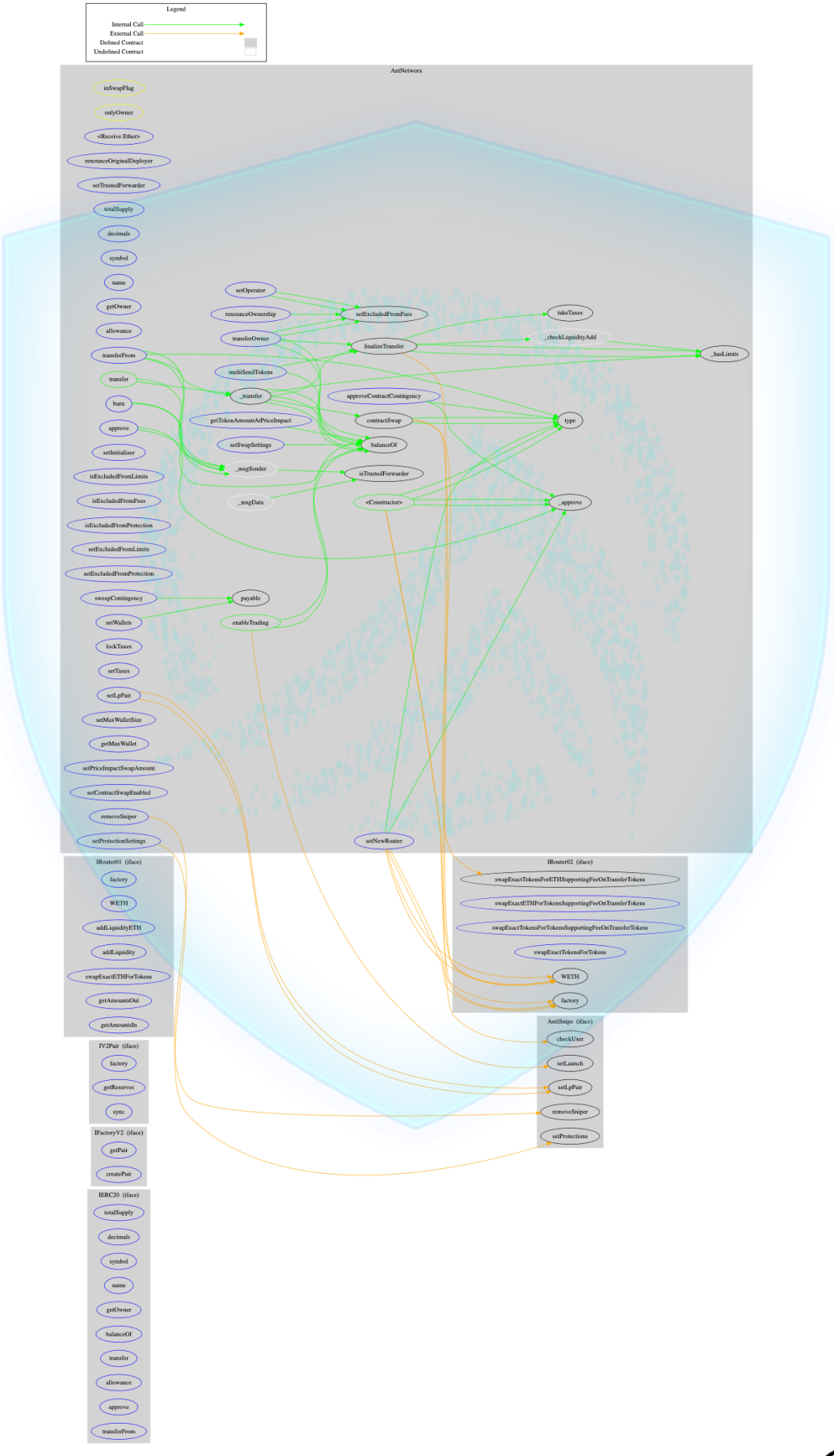
Liquidity Ownership

Most of the liquidity is currently locked; the lock can be seen here:

Liquidity Locker Link can be viewed from:
HERE

Call Graph

The contract for AntNetworX has the following call graph structure.



AegisX

KYC Information

The Project Owners of AntNetworX have provided KYC Documentation.

KYC Certificate can be found on the Following: KYC Data

KYC Information Notes:

Auditor Notes: KYC done by PinkSale Finance

Project Owner Notes:

Smart Contract Vulnerability Checks

ID	Severity	Name	File	location
SWC-100	Pass	Function Default Visibility	AntNetworx.sol	L: 0 C: 0
SWC-101	Pass	Integer Overflow and Underflow.	AntNetworx.sol	L: 0 C: 0
SWC-102	Pass	Outdated Compiler Version file.	AntNetworx.sol	L: 0 C: 0
SWC-103	Low	A floating pragma is set.	AntNetworx.sol	L: 6 C: 0
SWC-104	Pass	Unchecked Call Return Value.	AntNetworx.sol	L: 0 C: 0
SWC-105	Pass	Unprotected Ether Withdrawal.	AntNetworx.sol	L: 0 C: 0
SWC-106	Pass	Unprotected SELFDESTRUCT Instruction	AntNetworx.sol	L: 0 C: 0
SWC-107	Pass	Read of persistent state following external call.	AntNetworx.sol	L: 0 C: 0
SWC-108	Low	State variable visibility is not set..	AntNetworx.sol	L: 105 C: 30, L:142 C: 9, L: 153 C: 14
SWC-109	Pass	Uninitialized Storage Pointer.	AntNetworx.sol	L: 0 C: 0
SWC-110	Pass	Assert Violation.	AntNetworx.sol	L: 0 C: 0
SWC-111	Pass	Use of Deprecated Solidity Functions.	AntNetworx.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-112	Pass	Delegate Call to Untrusted Callee.	AntNetworx.sol	L: 0 C: 0
SWC-113	Pass	Multiple calls are executed in the same transaction.	AntNetworx.sol	L: 0 C: 0
SWC-114	Pass	Transaction Order Dependence.	AntNetworx.sol	L: 0 C: 0
SWC-115	Low	Authorization through tx.origin.	AntNetworx.sol	L: 467 C: 15
SWC-116	Pass	A control flow decision is made based on The block.timestamp environment variable.	AntNetworx.sol	L: 0 C: 0
SWC-117	Pass	Signature Malleability.	AntNetworx.sol	L: 0 C: 0
SWC-118	Pass	Incorrect Constructor Name.	AntNetworx.sol	L: 0 C: 0
SWC-119	Pass	Shadowing State Variables.	AntNetworx.sol	L: 0 C: 0
SWC-120	Low	Potential use of block.number as source of randonmness.	AntNetworx.sol	L: 561 C: 47
SWC-121	Pass	Missing Protection against Signature Replay Attacks.	AntNetworx.sol	L: 0 C: 0
SWC-122	Pass	Lack of Proper Signature Verification.	AntNetworx.sol	L: 0 C: 0
SWC-123	Pass	Requirement Violation.	AntNetworx.sol	L: 0 C: 0
SWC-124	Pass	Write to Arbitrary Storage Location.	AntNetworx.sol	L: 0 C: 0
SWC-125	Pass	Incorrect Inheritance Order.	AntNetworx.sol	L: 0 C: 0
SWC-126	Pass	Insufficient Gas Griefing.	AntNetworx.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-127	Pass	Arbitrary Jump with Function Type Variable.	AntNetworx.sol	L: 0 C: 0
SWC-128	Pass	DoS With Block Gas Limit.	AntNetworx.sol	L: 0 C: 0
SWC-129	Pass	Typographical Error.	AntNetworx.sol	L: 0 C: 0
SWC-130	Pass	Right-To-Left-Override control character (U+202E).	AntNetworx.sol	L: 0 C: 0
SWC-131	Pass	Presence of unused variables.	AntNetworx.sol	L: 0 C: 0
SWC-132	Pass	Unexpected Ether balance.	AntNetworx.sol	L: 0 C: 0
SWC-133	Pass	Hash Collisions with Multiple Variable Length Arguments.	AntNetworx.sol	L: 0 C: 0
SWC-134	Pass	Message call with hardcoded gas amount.	AntNetworx.sol	L: 0 C: 0
SWC-135	Pass	Code With No Effects (Irrelevant/Dead Code).	AntNetworx.sol	L: 0 C: 0
SWC-136	Pass	Unencrypted Private Data On-Chain.	AntNetworx.sol	L: 0 C: 0

We scan the contract for additional security issues using MYTHX and industry-standard security scanning tools.

Smart Contract Vulnerability Details

SWC-103 - Floating Pragma.

CWE-664: Improper Control of a Resource Through its Lifetime.

References:

Description:

Contracts should be deployed with the same compiler version and flags that they have been tested with thoroughly. Locking the pragma helps to ensure that contracts do not accidentally get deployed using, for example, an outdated compiler version that might introduce bugs that affect the contract system negatively.

Remediation:

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

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.

References:

Ethereum Smart Contract Best Practices - Lock pragmas to specific compiler version.

Smart Contract Vulnerability Details

SWC-108 - State Variable Default Visibility

CWE-710: Improper Adherence to Coding Standards

Description:

Labeling the visibility explicitly makes it easier to catch incorrect assumptions about who can access the variable.

Remediation:

Variables can be specified as being public, internal or private. Explicitly define visibility for all state variables.

References:

Ethereum Smart Contract Best Practices - Explicitly mark visibility in functions and state variables

Smart Contract Vulnerability Details

SWC-115 - Authorization through tx.origin

CWE-477: Use of Obsolete Function

Description:

tx.origin is a global variable in Solidity which returns the address of the account that sent the transaction. Using the variable for authorization could make a contract vulnerable if an authorized account calls into a malicious contract. 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 in this case is the authorized account.

Remediation:

tx.origin should not be used for authorization. Use msg.sender instead.

References:

Solidity Documentation - tx.origin

Ethereum Smart Contract Best Practices - Avoid using tx.origin

SigmaPrime - Visibility.

Smart Contract Vulnerability Details

SWC-120 - Weak Sources of Randomness from Chain Attributes

CWE-330: Use of Insufficiently Random Values

Description:

Solidity allows for ambiguous naming of state variables when inheritance is used. Contract A with a variable x could inherit contract B that also has a state variable x defined. This would result in two separate versions of x, one of them being accessed from contract A and the other one from contract B. In more complex contract systems this condition could go unnoticed and subsequently lead to security issues.

Shadowing state variables can also occur within a single contract when there are multiple definitions on the contract and function level.

Remediation:

Using commitment scheme, e.g. RANDAO. Using external sources of randomness via oracles, e.g. Oraclize. Note that this approach requires trusting in oracle, thus it may be reasonable to use multiple oracles. Using Bitcoin block hashes, as they are more expensive to mine.

References:

How can I securely generate a random number in my smart contract?)

When can BLOCKHASH be safely used for a random number? When would it be unsafe?

The Run smart contract.

Inheritance

The contract for AntNetworX has the following inheritance structure.

Privileged Functions (onlyOwner)

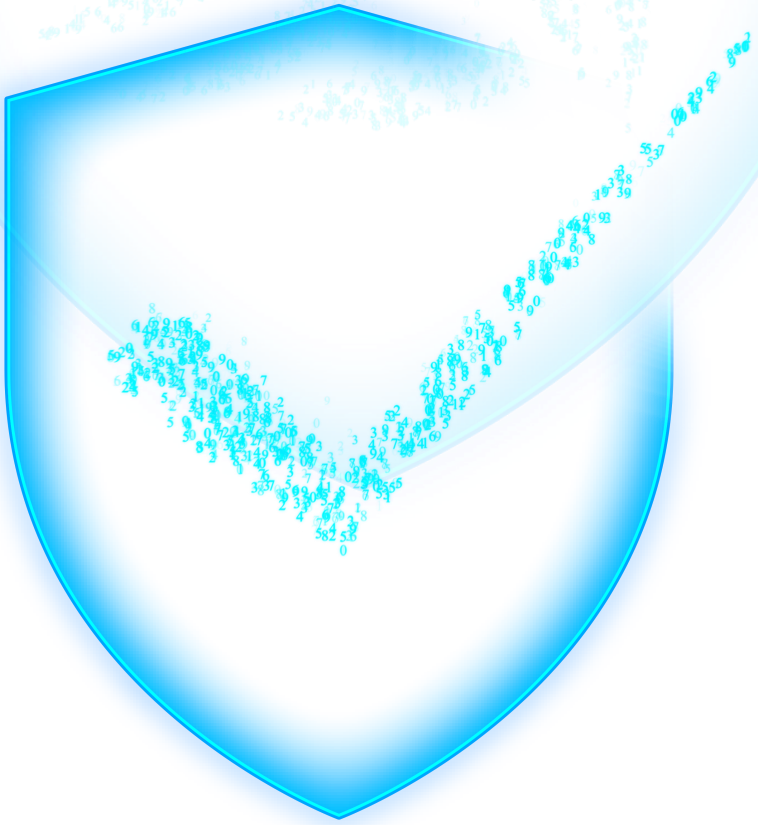
Function Name	Parameters	Visibility
transferOwner	none	external
renounceOwnership	none	external
setTrustedForwarder	none	external
approveContractContingency	none	external
setNewRouter	none	external
setLpPair	none	external
setInitializer	none	external
setExcludedFromLimits	none	external
setExcludedFromFees	none	external
setExcludedFromProtection	none	external
removeSniper	none	external

Function Name	Parameters	Visibility
setProtectionSettings	none	external
lockTaxes	none	external
setTaxes	none	external
setWallets	none	external
setMaxWalletSize	none	external
setSwapSettings	none	external
setPriceImpactSwapAmount	none	external
setContractSwapEnabled	none	external
enableTrading	none	public
sweepContingency	none	external
multiSendTokens	none	external

Assessment Results

- The owner can charge round trip fee up to 10%.
- The owner can set max wallet size as low as 1% of the total supply.
- The owner can blacklist a wallet address with removeSniper function.
- No high-risk Exploits/Vulnerabilities Were Found in the Source Code.

Audit Passed



ANTX-03 | Lack of Input Validation.

Category	Severity	Location	Status
Volatile Code	Minor	AntNetworx.sol: 289,9, 418,9, 450,9,	Pending

Description

The given input is missing the check for the non-zero address.

Remediation

We advise the client to add the check for the passed-in values to prevent unexpected errors as below:

```
...
 require(receiver != address(0), "Receiver is the zero address");
...
```

ANTX-05 | Missing Event Emission.

Category	Severity	Location	Status
Volatile Code	Minor	AntNetworx.sol: 288,5, 337,5, 351,5, 408,5, 417,5, 421,5	Pending

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.The linked code does not create an event for the transfer.

Remediation

Emit an event for critical parameter changes. It is recommended emitting events for the sensitive functions that are controlled by centralization roles.

Social Media Checks

Social Media	URL	Result
Website	https://www.antx.work/	Pass
Telegram	https://t.me/antnetworx	Pass
Twitter	https://twitter.com/antnetworx	Pass
OtherSocial	https://discord.gg/JY7vjRGfNE	Pass

We recommend to have 3 or more social media sources including a completed working websites.

Social Media Information Notes:

Auditor Notes: undefined

Project Owner Notes:

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.

Logical Issue

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

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functionsbeing invoke-able by anyone under certain circumstances.

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 makethe codebase more legible and, as a result, easily maintainable.

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code,such as a constructor assignment imposing different require statements on the input variables than a setterfunction.

Coding Best Practices

ERC 20 Conding Standards are a set of rules that each developer should follow to ensure the code meet a set of creterias and is readable by all the developers.

Disclaimer

AegisX has conducted an independent security assessment to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the reviewed code for the scope of this assessment. This report does not constitute agreement, acceptance, or advocation for the Project, and users relying on this report should not consider this as having any merit for financial advice in any shape, form, or nature. The contracts audited do not account for any economic developments that the Project in question may pursue, and the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude, and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are entirely free of exploits, bugs, vulnerabilities or deprecation of technologies.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence, regardless of the findings presented. Information is provided 'as is, and AegisX is under no covenant to audited completeness, accuracy, or solidity of the contracts. In no event will AegisX or its partners, employees, agents, or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions or actions with regards to the information provided in this audit report.

The assessment services provided by AegisX are subject to dependencies and are under continuing development. You agree that your access 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 with high levels of technical risk and uncertainty. The assessment reports could include false positives, negatives, and unpredictable results. The services may access, and depend upon, multiple layers of third parties.