AegisX

Smart Contract Audits | KYC

PALLADIUM

Security Assessment

AntNetworX dApp

November 1, 2022

Table of Contents

Assessment Summary

This report has been prepared for AntNetworX dApp 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	1	1	0
Medium	0	0	0
Minor	4	4	0
Informational	2	2	0
Total	7	7	0

Project Overview

Contract Summary

Parameter	Result
Address	0x9186359F82c3c0Cc005A0b3563Dc4Ccd2627D82A
Name	AntNetworX
Token Tracker	AntNetworX (WorkV3)
Decimals	N/A
Supply	N/A
Platform	Binance Smart Chain
compiler	v0.8.7+commit.e28d00a7
Contract Name	WorkV3
Optimization	200
LicenseType	MIT
Language	Solidity
Codebase	https://bscscan.com/address/0x59a8fcadfcfc6479ce3df1e2bae628ea85e4121a#code
Payment Tx	

Main Contract Assessed Contract Name

Name	Contract	Live
AntNetworX	0x9186359F82c3c0Cc005A0b3563Dc4Ccd2627D82A	Yes

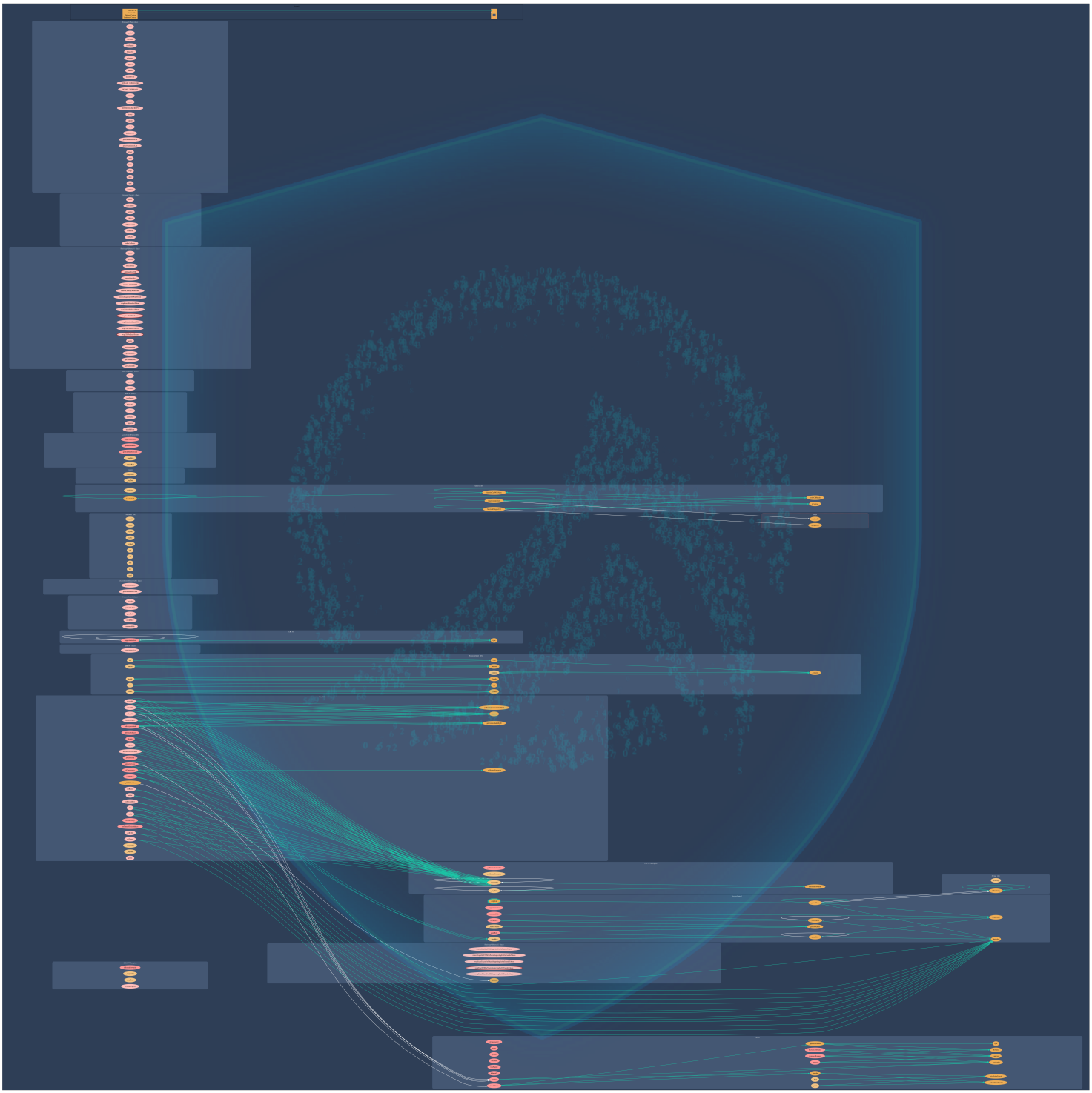
TestNet Contract was Not Assessed

Solidity Code Provided

SolID	File Sha-1	FileName
WorkV3	4de086b5da9dc053dc4638e06aca791ff0f62529	WorkV3.sol

Call Graph

The contract for AntNetworX has the following call graph structure.



KYC Information

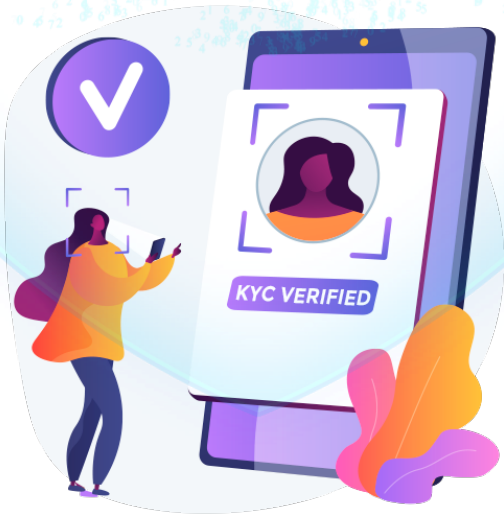
The Project Owners of AntNetworX have provided KYC Documentation.

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	WorkV3.sol	L: 0 C: 0
SWC-101	Pass	Integer Overflow and Underflow.	WorkV3.sol	L: 0 C: 0
SWC-102	Pass	Outdated Compiler Version file.	WorkV3.sol	L: 0 C: 0
SWC-103	Low	A floating pragma is set.	WorkV3.sol	L: 8 C: 0, L: 52 C: 0, L: 117 C: 0, L: 487 C: 0, L: 515 C: 0, L: 546 C: 0, L: 624 C: 0, L: 715 C: 0, L: 748 C: 0, L: 978 C: 0, L: 1203 C: 0, L: 1230 C: 0, L: 1479 C: 0, L: 1545 C: 0, L: 1630 C: 0, L: 1660 C: 0, L: 2043 C: 0
SWC-104	Pass	Unchecked Call Return Value.	WorkV3.sol	L: 0 C: 0
SWC-105	Pass	Unprotected Ether Withdrawal.	WorkV3.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-106	Pass	Unprotected SELFDESTRUCT Instruction	WorkV3.sol	L: 0 C: 0
SWC-107	Pass	Read of persistent state following external call.	WorkV3.sol	L: 0 C: 0
SWC-108	Low	State variable visibility is not set..	WorkV3.sol	L: 2427 C: 12, L: 2429 C: 12, L: 2430 C: 12, L: 2452 C: 9
SWC-109	Pass	Uninitialized Storage Pointer.	WorkV3.sol	L: 0 C: 0
SWC-110	Pass	Assert Violation.	WorkV3.sol	L: 0 C: 0
SWC-111	Pass	Use of Deprecated Solidity Functions.	WorkV3.sol	L: 0 C: 0
SWC-112	Pass	Delegate Call to Untrusted Callee.	WorkV3.sol	L: 0 C: 0
SWC-113	Pass	Multiple calls are executed in the same transaction.	WorkV3.sol	L: 0 C: 0
SWC-114	Pass	Transaction Order Dependence.	WorkV3.sol	L: 0 C: 0
SWC-115	Pass	Authorization through tx.origin.	WorkV3.sol	L: 467 C: 15
SWC-116	Pass	A control flow decision is made based on The block.timestamp environment variable.	WorkV3.sol	L: 0 C: 0
SWC-117	Pass	Signature Malleability.	WorkV3.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-118	Pass	Incorrect Constructor Name.	WorkV3.sol	L: 0 C: 0
SWC-119	Pass	Shadowing State Variables.	WorkV3.sol	L: 0 C: 0
SWC-120	Pass	Potential use of block.number as source of randonmness.	WorkV3.sol	L: 561 C: 47
SWC-121	Pass	Missing Protection against Signature Replay Attacks.	WorkV3.sol	L: 0 C: 0
SWC-122	Pass	Lack of Proper Signature Verification.	WorkV3.sol	L: 0 C: 0
SWC-123	Pass	Requirement Violation.	WorkV3.sol	L: 0 C: 0
SWC-124	Pass	Write to Arbitrary Storage Location.	WorkV3.sol	L: 0 C: 0
SWC-125	Pass	Incorrect Inheritance Order.	WorkV3.sol	L: 0 C: 0
SWC-126	Pass	Insufficient Gas Griefing.	WorkV3.sol	L: 0 C: 0
SWC-127	Pass	Arbitrary Jump with Function Type Variable.	WorkV3.sol	L: 0 C: 0
SWC-128	Pass	DoS With Block Gas Limit.	WorkV3.sol	L: 0 C: 0
SWC-129	Pass	Typographical Error.	WorkV3.sol	L: 0 C: 0
SWC-130	Pass	Right-To-Left-Override control character (U+202E).	WorkV3.sol	L: 0 C: 0
SWC-131	Pass	Presence of unused variables.	WorkV3.sol	L: 0 C: 0
SWC-132	Pass	Unexpected Ether balance.	WorkV3.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-133	Pass	Hash Collisions with Multiple Variable Length Arguments.	WorkV3.sol	L: 0 C: 0
SWC-134	Pass	Message call with hardcoded gas amount.	WorkV3.sol	L: 0 C: 0
SWC-135	Pass	Code With No Effects (Irrelevant/Dead Code).	WorkV3.sol	L: 0 C: 0
SWC-136	Pass	Unencrypted Private Data On-Chain.	WorkV3.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:

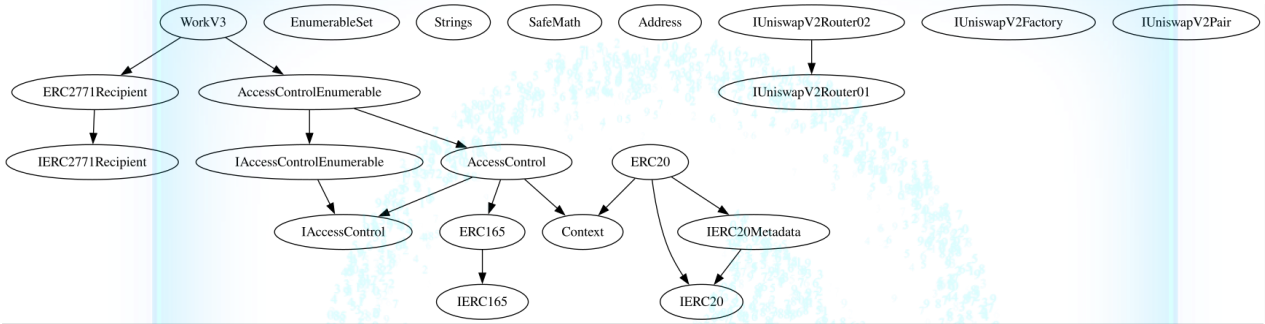
Variables can be specified as being public, internal or private. Explicitly define visibility for all state variables.

References:

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

Inheritance

The contract for AntNetworX has the following inheritance structure.



AegisX

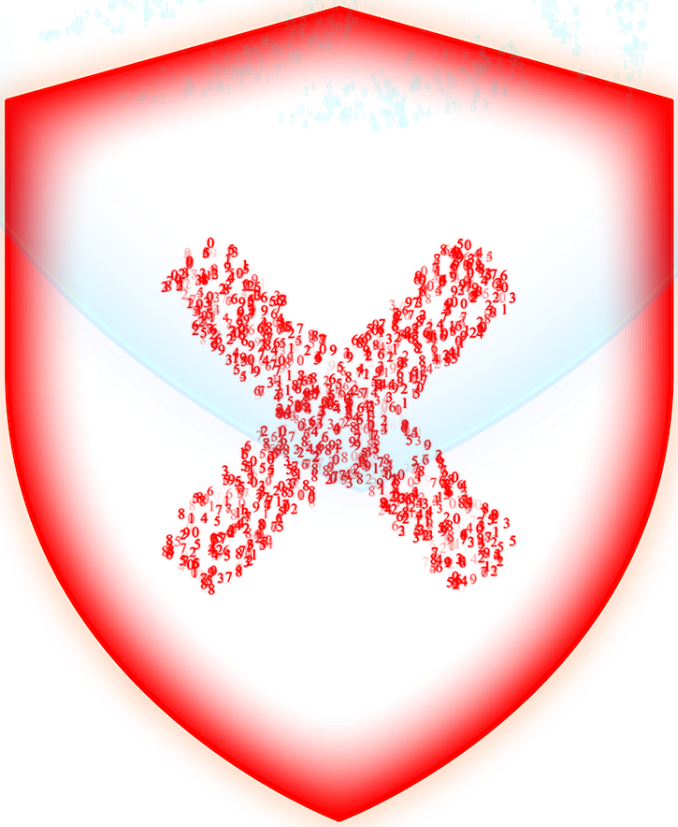
Privileged Functions (onlyOwner)

Function Name	Parameters	Visibility
N/A	N/A	N/A

Assessment Results

- There isn't a limit on how much the owner can set the fee to.
- An admin can ban a user with the function ban.
- CA: 0x86C80a8aa58e0A4fa09A69624c31Ab2a6CAD56b8 (https://bscscan.com/address/0x86c80a8aa58e0a4fa09a69624c31ab2a6cad56b8) is the current trusted forwarder, the contract isn't verified, must make sure that the forwarder is in a trustworthy state, as it is a key component of EIP 2771.
- A modification has been made to abstract contract AccessControl's function _checkRole at 1332,5 compared to OpenZeppelin's standard AccessControl contract. As this is only a view function, has not been added as an item to be flagged.
- Excessive use of libraries, the contract has room for further simplification.

Audit Fail



WorkV3-02 | Function Visibility Optimization.

Category	Severity	Location	Status
Gas Optimization	Informational	WorkV3.sol:	Pending

Description

The following functions are declared as public and are not invoked in any of the contracts contained within the projects scope:

Function Name	Parameters	Visibility
isTrustedForwarder		public
getTrustedForwarder		public
supportsInterface		public
hasRole		public
getRoleAdmin		public
grantRole		public
revokeRole		public
renounceRole		public
getRoleMember		public
getRoleMemberCount		public
name		public
symbol		public

Function Name	Parameters	Visibility
decimals		public
totalSupply		public
balanceOf		public
transfer		public
allowance		public
approve		public
transferFrom		public
increaseAllowance		public
decreaseAllowance		public
getPendingContractIndexByID		public
getContractIndexByID		public
addToMods		public
askForServices		public
jobDelivered		public
deliveryAccepted		public
deliveryRefused		public
dispute		public
setAutoVerify		public
setTimerrChoiceAddress		public

The functions that are never called internally within the contract should have external visibility

Remediation

We advise that the function's visibility specifiers are set to external, and the array-based arguments change their data location from memory to calldata, optimizing the gas cost of the function.

References:

external vs public best practices.

WorkV3-03 | Lack of Input Validation.

Category	Severity	Location	Status
Volatile Code	Minor	WorkV3.sol: 2688,5, 2744,5	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");
...
```

WorkV3-05 | Missing Event Emission.

Category	Severity	Location	Status
Volatile Code	Minor	WorkV3.sol: 2688,5, 2744,5	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.

WorkV3-08 | Dead Code Elimination.

Category	Severity	Location	Status
Coding Style	Major	WorkV3.sol: 2505,5	Pending

Description

Functions that are not used in the contract, and make the size of the codes unncecessarily bigger.

The function swapBUSDforTimerr is a private function that is never called.
swapBUSDforTimerr gone, uniswap libraries can be removed as well.
The fuction setRouter does use uniswap library.
No token swap occurring by the contract, it lacks a purpose of existence.
2 libaries and 2 functions along with a few lines of codes can potentially be removed.

Remediation

Remove unused functions. dead-code elimination (also known as DCE, dead-code removal, dead-code stripping, or dead-code strip) is a compiler optimization to remove code which does not affect the program results. Removing such code has several benefits: it shrinks program size, an important consideration in some contexts, and it allows the running program to avoid executing irrelevant operations, which reduces its running time. It can also enable further optimizations by simplifying program structure.

https://docs.soliditylang.org/en/latest/cheatsheet.html

WorkV3-11 | Modification on library EnumerableSet.

Category	Severity	Location	Status
Informational	Informational	WorkV3.sol: 395, 5	Pending

Description

During our review we noticed that a modification has been made on library EnumerableSet's function values (line 395). Perhaps it was done to save gas?

Remediation

Waiting for dev's response on reason why.

Project Action

Pending Customer Response

WorkV3-14 | Unnecessary Use Of SafeMath

Category	Severity	Location	Status
Logical Issue	Informational	WorkV3.sol: 760, 0	Pending

Description

The SafeMath library is used unnecessarily. With Solidity compiler versions 0.8.0 or newer, arithmetic operations
will automatically revert in case of integer overflow or underflow.
9 library SafeMath {
An implementation of SafeMath library is found.
9 using SafeMath for uint256;
SafeMath library is used for uint256 type in contract.

SafeMath.div is called in distribute function of contract.
Note: Only a sample of 2 SafeMath library usage in this contract (out of 18) are shown above.

Remediation

We advise removing the usage of SafeMath library and using the built-in arithmetic operations provided by the
Solidity programming language

Project Action

Removal advised. Yes, the safemath version is for compiler 0.8+, however during our review we have noticed that there is a very limited use of safemath functions.

WorkV3-15 | Modification on library Address.

Category	Severity	Location	Status
Informational	Minor	WorkV3.sol: 1109,9, 1136,9, 1163,9	Pending

Description

During our review we have noticed that a few modifications have been made on library Address. Additional validation requiring isContract(target).

Remediation

Require dev's acknowledgement on the modification, may be fine for this dApp's purpose but this type of modification can cause an issue on typical token contracts.

Project Action

Pending Customer Response

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.