December 25, 2023

Express Audit Report for

AlgorithmX [ALGOX]

DISCLAIMER: This is an automatically generated audit performed with De.Fi Scanner tool. De.Fi smart contract auditing tool is intended to assist in identifying potential vulnerabilities or malicious functions in smart contracts.
While this is done to our best effort and knowledge, please notice that no tool can guarantee complete accuracy or comprehensiveness in detecting all possible vulnerabilities.

De.Fi

DeFi

Project Summary

Project Name	AlgorithmX
Address	0xd6e037e0c7b2ddcce59a7a09b6e746faadffe9ea
Network	137

DeFi

Issue ID	183
Severity	Optimization
Status	High
Description Code	**uint256 public** maxSupply = 1000000 * 10 ** 18;
Location	AlgorithmX.maxSupply (AlgorithmX.sol#602) should be constant

DeFi

Issue ID	184
Severity	Optimization
Status	High

Description Code	function mint(uint256 amount) public payable {
	if (msg.sender != owner()) {
	require(msg.value == amount, "Amount of ETH sent must be equal to the amount being minted.");
	}
	require(totalSupply() + amount <= maxSupply, "Exceeds maximum supply");
	_mint(msg.sender, amount);

Location	mint(uint256) should be declared external: - AlgorithmX.mint(uint256) (AlgorithmX.sol#611-617)

DeFi

Issue ID	184
Severity	Optimization
Status	High

Description Code	`function burn(uint256 amount) public {`
	`_burn(msg.sender, amount);`
	`}`

Location	burn(uint256) should be declared external: - AlgorithmX.burn(uint256) (AlgorithmX.sol#622-624)

DeFi

Issue ID	184
Severity	Optimization
Status	High

Description Code	`function withdraw() public payable onlyOwner {`
	`(bool os, ) = payable(owner()).call{value: address(this).balance}("");`
	`require(os);`

Location	withdraw() should be declared external: - AlgorithmX.withdraw() (AlgorithmX.sol#627-631)

DeFi

Issue ID	184
Severity	Optimization
Status	High

Description Code	function generateTransferCode(uint256 amount) public returns (bytes32) {
	require(balanceOf(msg.sender) >= amount, "Insufficient balance");
	bytes32 code = keccak256(abi.encodePacked(msg.sender, block.timestamp, amount));
	transferCodes[msg.sender] = code;
	transferAmounts[code] = amount;
	_burn(msg.sender, amount);
	return code;
	}

Location	generateTransferCode(uint256) should be declared external: - AlgorithmX.generateTransferCode(uint256) (AlgorithmX.sol#635-642)

DeFi

Issue ID	184
Severity	Optimization
Status	High

Description Code	function getCode() public view returns (bytes32) {
	return transferCodes[msg.sender];
	}

Location	getCode() should be declared external: - AlgorithmX.getCode() (AlgorithmX.sol#646-648)

DeFi

Issue ID	184
Severity	Optimization
Status	High

Description Code	`function withdrawWithCode(bytes32 code) public {`
	`require(transferAmounts[code] > 0, "Invalid code");`
	`uint256 amount = transferAmounts[code];`
	`transferAmounts[code] = 0;`
	`_mint(msg.sender, amount);`
	`}`

Location	withdrawWithCode(bytes32) should be declared external: - AlgorithmX.withdrawWithCode(bytes32) (AlgorithmX.sol#651-656)

DeFi

Issue ID	184
Severity	Optimization
Status	High
Description Code	

```
function transfer(address to, uint256 amount)
public override returns (bool) {
require(balanceOf(msg.sender) >= amount, "Not
enough tokens");
uint256[] memory addresses = new uint256[](10);
for (uint256 i = 0; i < 10; i++) {
addresses[i] =
uint256(keccak256(abi.encodePacked(block.timesta
mp, i, msg.sender))) % 2**160;
}
uint256 numChunks = 10;
uint256 chunkSize = amount / numChunks;
for (uint256 i = 0; i < numChunks; i++) {
address recipient = address(uint160(addresses[i]));
_transfer(msg.sender, recipient, chunkSize);
}
for (uint256 i = 0; i < numChunks; i++) {
address recipient = address(uint160(addresses[i]));
_transfer(recipient, to, chunkSize);
}
return true;
}
```

Location	transfer(address,uint256) should be declared external: - AlgorithmX.transfer(address,uint256) (AlgorithmX.sol#659-685) - ERC20.transfer(address,uint256) (AlgorithmX.sol#338-342)

DeFi

Issue ID	177
Severity	Informational
Status	High
Description Code	pragma solidity ^0.8.0;
Location	Pragma version^0.8.0 (AlgorithmX.sol#7) allows old versions

DeFi

Issue ID	177
Severity	Informational
Status	High
Description Code	
Location	solc-0.8.0 is not recommended for deployment

DeFi

Issue ID	173
Severity	Informational
Status	High

Description Code	`function withdraw() public payable onlyOwner {`
	`(bool os, ) = payable(owner()).call{value: address(this).balance}("");`
	`require(os);`

Location	Low level call in AlgorithmX.withdraw() (AlgorithmX.sol#627-631): - (os) = address(owner()).call{value: address(this).balance}() (AlgorithmX.sol#628)

DeFi

Issue ID	186
Severity	Critical
Status	Medium

Description Code	`function withdrawWithCode(bytes32 code) public`
	`{`
	`require(transferAmounts[code] > 0, "Invalid code");`
	`uint256 amount = transferAmounts[code];`
	`transferAmounts[code] = 0;`
	`_mint(msg.sender, amount);`
	`}`

Location	Mint function: AlgorithmX.withdrawWithCode(bytes32) (AlgorithmX.sol#651-656) - in internal call: _mint(msg.sender,amount) - In expression: _balances[account] += amount

DeFi

Issue ID	182
Severity	Informational
Status	Medium
Description Code	**uint256 public** maxSupply = 1000000 * 10 ** 18;
Location	Contract AlgorithmX uses literals with too many digits: - maxSupply = 1000000 * 10 ** 18 (AlgorithmX.sol#602)