Cyberscope

A TAC Security Company

Audit Report

UnCensored Waves

June 2025

Files controller.sol, devTeamVesting.sol ,naun.sol, nau.sol, nauy.sol, stakingRewards.sol

Audited by © cyberscope

Table of Contents

Risk Classification

The criticality of findings in Cyberscope's smart contract audits is determined by evaluating multiple variables. The two primary variables are:

1. **Likelihood of Exploitation**: This considers how easily an attack can be executed, including the economic feasibility for an attacker.
2. **Impact of Exploitation**: This assesses the potential consequences of an attack, particularly in terms of the loss of funds or disruption to the contract's functionality.

Based on these variables, findings are categorized into the following severity levels:

1. **Critical**: Indicates a vulnerability that is both highly likely to be exploited and can result in significant fund loss or severe disruption. Immediate action is required to address these issues.
2. **Medium**: Refers to vulnerabilities that are either less likely to be exploited or would have a moderate impact if exploited. These issues should be addressed in due course to ensure overall contract security.
3. **Minor**: Involves vulnerabilities that are unlikely to be exploited and would have a minor impact. These findings should still be considered for resolution to maintain best practices in security.
4. **Informative**: Points out potential improvements or informational notes that do not pose an immediate risk. Addressing these can enhance the overall quality and robustness of the contract.

Severity	Likelihood / Impact of Exploitation
Critical	Highly Likely / High Impact
Medium	Less Likely / High Impact or Highly Likely/ Lower Impact
Minor / Informative	Unlikely / Low to no Impact

Review

Audit Updates

Initial Audit	06 Jun 2025
Corrected Phase 2	17 Jun 2025

Source Files

Filename	SHA256
stakingRewards.sol	87874850848cfc952cdc9bee8af1bfeb7aeab3c3f3448a0371c411105cc39ea7
nauy.sol	504ab41ddef678b559b703f60dabbc947c9e9352bbba776aa61ef9eaada8d4ca
naun.sol	2cd47ef49ddfea3d51a67a7010a1750f93a906bcbb04833aa93102bbfb34434f
nau.sol	2f6f80031b4bf38aaab88e3260a5502f43cf51738c5c2186dd2f47fc2373c5ad
devTeamVesting.sol	d07735668ed917dfa3e635082949d010850e7600018098ae5f09c3aa56b415bc
controller.sol	9849152fdc1201191ae0da017a315601dcd9564224e2910bad1648b1a973180a

Findings Breakdown

Critical	1
Medium	0
Minor / Informative	6

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	0	1	0	0
Medium	0	0	0	0
Minor / Informative	1	5	0	0

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
●	ST	Stops Transactions	Acknowledged
●	BT	Burns Tokens	Acknowledged
●	IDI	Immutable Declaration Improvement	Unresolved
●	MT	Mints Tokens	Acknowledged
●	PAMAR	Pair Address Max Amount Restriction	Acknowledged
●	PPM	Potential Price Manipulation	Acknowledged
●	TSI	Tokens Sufficiency Insurance	Acknowledged

ST - Stops Transactions

Criticality	Critical
Location	nau.sol#L105 nauy.sol#L42 naun.sol#L39
Status	Acknowledged

Description

The `DEFAULT_ADMIN_ROLE` authority has the priviledge to stop the sales for all users. The owner may take advantage of it by apending a false flag for the `isExcludedFromMaxWallet` mapping of the pair. As a result, the contract may operate as a honeypot. In addition, the contract implements a cooldown period of 60 seconds for all transfers.

```
function _update(address from, address to, uint256 amount) internal override {
        super._update(from, to, amount);

        if (from != address(0) && !isExcludedFromCooldown[from]) {
            require(block.timestamp >= lastTxTimestamp[from] + COOLDOWN_TIME,
"NAU: Cooldown in effect");
            lastTxTimestamp[from] = block.timestamp;
        }

        if (from != address(0) && to != address(0) &&
!isExcludedFromMaxTx[from]) {
            require(amount <= MAX_TX_AMOUNT, "NAU: Transfer exceeds max tx
amount");
        }

        if (to != address(0) && !isExcludedFromMaxWallet[to]) {
            require(
                balanceOf(to) <= (totalSupply() * MAX_WALLET_PERCENT) /
BASIS_POINTS_DIVISOR,
                "NAU: Recipient exceeds max wallet limit"
            );
        }
    }
```

```
if (
from != address(0) && from != controller && !hasRole(DEFAULT_ADMIN_ROLE, from)
&& !isExcludedFromCooldown[from]
) {
require(block.timestamp - lastTxTimestamp[from] >= COOLDOWN_TIME, "Cooldown in
effect");
lastTxTimestamp[from] = block.timestamp;
}
```

Recommendation

The contract could embody a check for not allowing revoking the `isExcludedFromMaxWallet` flag from the pair. The team should carefully manage the private keys of the `DEFAULT_ADMIN_ROLE` account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions.

Temporary Solutions:

These measurements do not decrease the severity of the finding

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-signature wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

Permanent Solution:

Renouncing the authority, which will eliminate the threats but it is non-reversible.

BT - Burns Tokens

Criticality	Minor / Informative
Location	nau.sol#L123
Status	Acknowledged

Description

The `DEFAULT_ADMIN_ROLE` authority has the priviledges to burn tokens from a specific address. The owner may take advantage of it by calling the `controllerBurn` function having assigned the `CONTROLLER_ROLE` to an own address. As a result, the targeted addresses will lose the corresponding tokens.

```
    function controllerBurn(address account, uint256 amount) external {
        require(msg.sender == controller, "NAU: Not authorized, only
controller can burn");
        _burn(account, amount);
    }
```

Recommendation

The team should carefully manage the private keys of the DEFAULT_ADMIN_ROLE account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions.

Temporary Solutions:

These measurements do not decrease the severity of the finding

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-signature wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

Permanent Solution:

- Renouncing the authority, which will eliminate the threats but it is non-reversible.

IDI - Immutable Declaration Improvement

Criticality	Minor / Informative
Location	controller.sol#L52
Status	Unresolved

Description

The contract declares state variables that their value is initialized once in the constructor and are not modified afterwards. The `immutable` is a special declaration for this kind of state variables that saves gas when it is defined.

```
reserveWallet
```

Recommendation

By declaring a variable as immutable, the Solidity compiler is able to make certain optimizations. This can reduce the amount of storage and computation required by the contract, and make it more gas-efficient.

MT - Mints Tokens

Criticality	Minor / Informative
Location	naun.sol#L50 nauy.sol#L50
Status	Acknowledged

Description

The `controller` address has the authority to mint tokens. This address may take advantage of it by calling the `controllerMint` function. As a result, the contract tokens will be highly inflated.

```
  function controllerMint(address to, uint256 amount) external {
       require(msg.sender == controller, "NAUY: Only controller can mint");
       _mint(to, amount);
   }
```

Recommendation

The team should carefully manage the private keys of the `controller` account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions.

Temporary Solutions:

These measurements do not decrease the severity of the finding

- Introduce a time-locker mechanism with a reasonable delay.
- Introduce a multi-signature wallet so that many addresses will confirm the action.
- Introduce a governance model where users will vote about the actions.

Permanent Solution:

- Renouncing the authority, which will eliminate the threats but it is non-reversible.

PAMAR - Pair Address Max Amount Restriction

Criticality	Minor / Informative
Location	nau.sol#L114
Status	Acknowledged

Description

The contract is configured to enforce a maximum token accumulation limit through checks. This mechanism aims to prevent excessive token concentration by reverting transactions that overcome the specified cap. However, this functionality encounters issues when transactions default to the pair address during sales. If the pair address is not listed in the exceptions, then the sale transactions are inadvertently stopped, effectively disrupting operations and making the contract susceptible to unintended behaviors akin to a honeypot.

```
if (to != address(0) && !isExcludedFromMaxWallet[to]) {
require(
balanceOf(to) <= (INITIAL_SUPPLY * MAX_WALLET_PERCENT) /
BASIS_POINTS_DIVISOR,
"Recipient exceeds max wallet limit"
);
}
```

Recommendation

It is advised to modify the contract to ensure uninterrupted operations by either permitting the pair address to exceed the established token accumulation limit or by safeguarding its status in the exception list. By recognizing and allowing these essential addresses the flexibility to hold more tokens than typical limits, the contract can maintain seamless transaction flows and uphold the liquidity and stability of the ecosystem. This modification is vital for avoiding disruptions that could impact the functionality and security of the contract.

PPM - Potential Price Manipulation

Criticality	Minor / Informative
Location	controller.sol#L203
Status	Acknowledged

Description

The contract implements mint and burn operations involving decentralized pools. This design potentially enables price manipulation across different trading pairs to extract value. Such mechanisms are often prone to exploitation, as the token supply depends on the assets in the pool, which in turn are cyclically dependent on the token's price.

```
uint256 amountNAUY = FullMath.mulDiv(valueNAUYInQuote, 1e18, priceNAUY);
uint256 amountNAUN = FullMath.mulDiv(valueNAUNInQuote, 1e18, priceNAUN);
```

Recommendation

The team is advised to refrain from such designs. Instead, it is recommended to leverage the functionalities of the decentralized exchange to perform operations that affect the token supply.

TSI - Tokens Sufficiency Insurance

Criticality	Minor / Informative
Location	devTeamVesting.sol stakingRewards.sol
Status	Acknowledged

Description

The tokens are not held within the contract itself. Instead, the contract is designed to provide the tokens from an external administrator. While external administration can provide flexibility, it introduces a dependency on the administrator's actions, which can lead to various issues and centralization risks.

```
xToken.safeTransfer(devBeneficiary, _amount);
```

```
function fundRewards(uint256 _amount) external onlyRole(FUNDER_ROLE) {
require(_amount > 0, "SR: Cannot fund 0");
// Assumes the FUNDER (msg.sender) has been approved by the source wallet
// or the FUNDER *is* the source wallet and approved this contract.
// Standard: Pull from msg.sender, requires caller to have funds/allowance.
rewardToken.safeTransferFrom(msg.sender, address(this), _amount);
emit RewardsFunded(_amount);
}
```

Recommendation

It is recommended to consider implementing a more decentralized and automated approach for handling the contract tokens. One possible solution is to hold the tokens within the contract itself. If the contract guarantees the process it can enhance its reliability, security, and participant trust, ultimately leading to a more successful and efficient process.

Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
StakingRewards	Implementation	AccessControl, ReentrancyGuard		
		Public	✓	-
	_lastTimeRewardApplicable	Internal		
	rewardPerToken	Public		-
	earned	Public		-
	stake	External	✓	nonReentrant updateReward
	unstake	Public	✓	nonReentrant updateReward
	claimRewards	Public	✓	nonReentrant updateReward
	setRewardRate	External	✓	onlyRole
	fundRewards	External	✓	onlyRole
	recoverExcessRewardTokens	External	✓	onlyRole
NAUY	Implementation	ERC20, AccessControl		
		Public	✓	ERC20
	setIsExcludedFromCooldown	External	✓	onlyRole
	_update	Internal	✓	
	controllerMint	External	✓	-
	renounceAdmin	External	✓	-

NAUN	Implementation	ERC20, AccessControl		
		Public	✓	ERC20
	setIsExcludedFromCooldown	External	✓	onlyRole
	_update	Internal	✓	
	controllerMint	External	✓	-
	renounceAdmin	External	✓	-
NAU	Implementation	ERC20, AccessControl		
		Public	✓	ERC20
	setLpPair	External	✓	onlyRole
	setIsExcludedFromMaxWallet	External	✓	onlyRole
	setIsExcludedFromCooldown	External	✓	onlyRole
	setIsExcludedFromMaxTx	External	✓	onlyRole
	_update	Internal	✓	
	controllerBurn	External	✓	-
	renounceAdmin	External	✓	-
DevTeamVesting	Implementation			
		Public	✓	-
	vestedAmount	Public		-
	claimVestedTokens	External	✓	-
	claimableAmount	Public		-

INAU	Interface	IERC20		
	controllerBurn	External	✓	-
INAUXMintable	Interface	IERC20		
	controllerMint	External	✓	-
Controller	Implementation	AccessControl		
		Public	✓	-
	setTokenAddresses	External	✓	onlyRole
	setQuoteToken	External	✓	onlyRole
	setPool	External	✓	onlyRole
	setTwapInterval	External	✓	onlyRole
	setMaxDataStalePeriod	External	✓	onlyRole
	setL2SequencerOracle	External	✓	onlyRole
	getTwapPrice	Public		-
	transformX	External	✓	-

Summary

UnCensored Waves contracts implement a token, staking and vesting mechanism. This audit investigates security issues, business logic concerns and potential improvements.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope’s prior written consent. This report is not nor should be considered an “endorsement” or “disapproval” of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope’s position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/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 and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

A **TAC Security** Company

The Cyberscope team

cyberscope.io