



# Cyberscope

A **TAC Security** Company

## Audit Report **XNAP Token**

January 2026

Network      BSC\_TESTNET

Address      0xE46A3C711Ca7CF58B233a1F8FEE2e0De56056cB2

Audited by    © cyberscope

# Analysis

● Critical   ● Medium   ● Minor / Informative   ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Unresolved
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

## Diagnostics

● Critical   ● Medium   ● Minor / Informative

Severity	Code	Description	Status
●	PAMAR	Pair Address Max Amount Restriction	Unresolved
●	MVN	Misleading Variables Naming	Unresolved
●	NWES	Nonconformity with ERC-20 Standard	Unresolved
●	UAR	Unexcluded Address Restrictions	Unresolved
●	CCR	Contract Centralization Risk	Unresolved
●	L14	Uninitialized Variables in Local Scope	Unresolved

# Table of Contents

<b>Analysis</b>	<b>2</b>
<b>Diagnostics</b>	<b>3</b>
<b>Table of Contents</b>	<b>4</b>
<b>Risk Classification</b>	<b>5</b>
<b>Review</b>	<b>6</b>
Audit Updates	6
Source Files	6
<b>Findings Breakdown</b>	<b>6</b>
ST - Stops Transactions	8
Description	8
Recommendation	8
PAMAR - Pair Address Max Amount Restriction	9
Description	9
Recommendation	10
MVN - Misleading Variables Naming	11
Description	11
Recommendation	12
NWES - Nonconformity with ERC-20 Standard	13
Description	13
Recommendation	13
UAR - Unexcluded Address Restrictions	14
Description	14
Recommendation	14
CCR - Contract Centralization Risk	15
Description	15
Recommendation	15
L14 - Uninitialized Variables in Local Scope	16
Description	16
Recommendation	16
<b>Functions Analysis</b>	<b>17</b>
<b>Inheritance Graph</b>	<b>19</b>
<b>Flow Graph</b>	<b>20</b>
<b>Summary</b>	<b>21</b>
<b>Disclaimer</b>	<b>22</b>
<b>About Cyberscope</b>	<b>23</b>

# 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

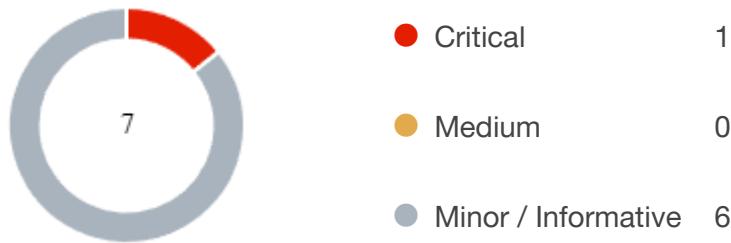
<b>Initial Audit</b>	06 Oct 2025
	<a href="https://github.com/cyberscope-io/audits/blob/main/xnap/v1/audit.pdf">https://github.com/cyberscope-io/audits/blob/main/xnap/v1/audit.pdf</a>
<b>Corrected Phase 2</b>	20 Oct 2025
	<a href="https://github.com/cyberscope-io/audits/blob/main/xnap/v2/audit.pdf">https://github.com/cyberscope-io/audits/blob/main/xnap/v2/audit.pdf</a>
<b>Corrected Phase 3</b>	04 Nov 2025
	<a href="https://github.com/cyberscope-io/audits/blob/main/xnap/v3/audit.pdf">https://github.com/cyberscope-io/audits/blob/main/xnap/v3/audit.pdf</a>
<b>Corrected Phase 4</b>	31 Dec 2025
	<a href="https://github.com/cyberscope-io/audits/blob/main/xnap/v4/audit.pdf">https://github.com/cyberscope-io/audits/blob/main/xnap/v4/audit.pdf</a>
<b>Corrected Phase 5</b>	12 Jan 2026

<b>Explorer</b>	<a href="https://testnet.bscscan.com/address/0xe46a3c711ca7cf58b233a1f8fee2e0de56056cb2">https://testnet.bscscan.com/address/0xe46a3c711ca7cf58b233a1f8fee2e0de56056cb2</a>
-----------------	---

## Source Files

Filename	SHA256
<b>XNAPToken.sol</b>	bf89d5cbbe620bb0cb485eb230afd4fdf7532bca4861071668e27bee26c5705

## Findings Breakdown



Severity	Unresolved	Acknowledged	Resolved	Other
Critical	1	0	0	0
Medium	0	0	0	0
Minor / Informative	6	0	0	0

## ST - Stops Transactions

Criticality	Critical
Location	contracts/XNAPToken.sol#L217
Status	Unresolved

### Description

The contract may stop the transfers for all users, due to restrictions enforced during user transactions. While limits are active, the maximum wallet amount is applied to the liquidity pair address, which may halt sell operations and prevent users from selling their tokens, as detailed in the PAMAR finding

```
Rust
require(
    _balances[to] + sendAmount <= maxWalletAmount,
    "max wallet"
);
```

### Recommendation

The team is advised to follow the recommendations outlined in the PAMAR findings and implement the necessary safeguards to mitigate the identified risks, ensuring that the contract does not exhibit honeypot behavior

## PAMAR - Pair Address Max Amount Restriction

Criticality	Minor / Informative
Location	contracts/XNAPToken.sol#L217
Status	Unresolved

### 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.

```
Shell
    require(_balances[to] + sendAmount <=
maxWalletAmount, "max wallet" );
```

## 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.

## MVN - Misleading Variables Naming

<b>Criticality</b>	Minor / Informative
<b>Location</b>	XNAPToken.sol#L202
<b>Status</b>	Unresolved

### Description

Variables can have misleading names if their names do not accurately reflect the value they contain or the purpose they serve. The contract uses some variable names that are too generic or do not clearly convey the information stored in the variable. Misleading variable names can lead to confusion, making the code more difficult to read and understand. Specifically, the BUY\_BURN\_BP and SELL\_BURN\_BP are used to calculate the burnAmount if the users perform a buy or a sell but they are also applied during the addition and removal of liquidity from the pair.

```
Shell
if (pancakePair != address(0)) {
    if (from == pancakePair) {
        burnAmount = (value * BUY_BURN_BP) /
BP_DENOM;
    } else if (to == pancakePair) {
        burnAmount = (value * SELL_BURN_BP) /
BP_DENOM;
    } else {
        burnAmount = (value * TRANSFER_BURN_BP) /
BP_DENOM;
    }
}
```

## Recommendation

It's always a good practice for the contract to contain variable names that are specific and descriptive. The team is advised to keep in mind the readability of the code.

## NWES - Nonconformity with ERC-20 Standard

Criticality	Minor / Informative
Location	XNAPToken.sol#L198
Status	Unresolved

### Description

The contract does not fully conform to the ERC20 Standard. Specifically, according to the standard, transfers of 0 values must be treated as normal transfers and fire the Transfer event. However, the contract implements a conditional check that prohibits transfers of 0 values. This discrepancy between the contract's implementation and the ERC20 standard may lead to inconsistencies and incompatibilities with other contracts.

```
Shell
function _transfer(address from, address to,
uint256 value)
{
    ...
    require(value > 0, "zero");
```

### Recommendation

The incorrect implementation of the ERC20 standard could potentially lead to problems when interacting with the contract, as other contracts or applications that expect the ERC20 interface may not behave as expected. The team is advised to review and revise the implementation of the transfer mechanism to ensure full compliance with the ERC20 standard. <https://eips.ethereum.org/EIPS/eip-20>.

## UAR - Unexcluded Address Restrictions

Criticality	Minor / Informative
Location	XNAPToken.sol#L217
Status	Unresolved

### Description

The contract incorporates operational restrictions on transactions, which can hinder seamless interaction with decentralized applications (dApps) such as launchpads, presales, lockers, or staking platforms. In scenarios where an external contract, such as a launchpad factory, needs to integrate with the contract, it should be exempt from the limitations to ensure uninterrupted service and functionality. Failure to provide such exemptions can block the successful process and operation of services reliant on this contract.

```
Shell
require( _balances[to] + sendAmount <=
maxWalletAmount, "max wallet");
```

### Recommendation

It is advisable to modify the contract by incorporating functionality that enables the exclusion of designated addresses from transactional restrictions. This enhancement will allow specific addresses, such as those associated with decentralized applications (dApps) and service platforms, to operate without being hindered by the standard constraints imposed on other users. Implementing this feature will ensure smoother integration and functionality with external systems, thereby expanding the contract's versatility and effectiveness in diverse operational environments.

## CCR - Contract Centralization Risk

<b>Criticality</b>	Minor / Informative
<b>Location</b>	XNAPToken.sol#L137,163
<b>Status</b>	Unresolved

### Description

The contract's functionality and behavior are heavily dependent on external parameters or configurations. While external configuration can offer flexibility, it also poses several centralization risks that warrant attention. Centralization risks arising from the dependence on external configuration include Single Point of Control, Vulnerability to Attacks, Operational Delays, Trust Dependencies, and Decentralization Erosion.

Shell

```
function removeLimits() external onlyOwner

function setExcluded(address account, bool excluded)
external onlyOwner
```

### Recommendation

To address this finding and mitigate centralization risks, it is recommended to evaluate the feasibility of migrating critical configurations and functionality into the contract's codebase itself. This approach would reduce external dependencies and enhance the contract's self-sufficiency. It is essential to carefully weigh the trade-offs between external configuration flexibility and the risks associated with centralization.

## L14 - Uninitialized Variables in Local Scope

Criticality	Minor / Informative
Location	contracts/XNAPToken.sol#L200
Status	Unresolved

### Description

Using an uninitialized local variable can lead to unpredictable behavior and potentially cause errors in the contract. It's important to always initialize local variables with appropriate values before using them.

```
Shell
nt256 burnAmount;
```

### Recommendation

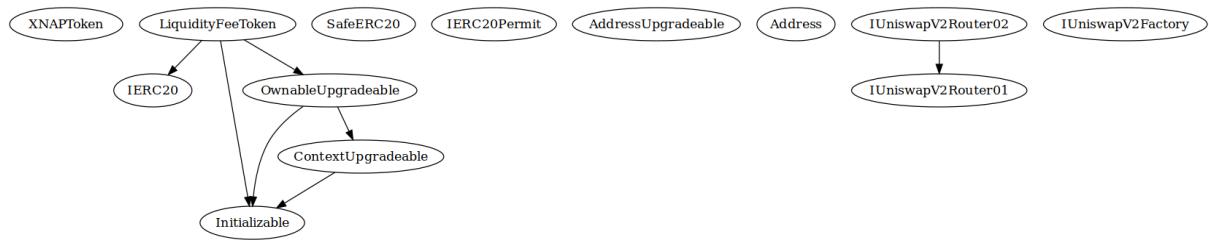
By initializing local variables before using them, the contract ensures that the functions behave as expected and avoid potential issues.

# Functions Analysis

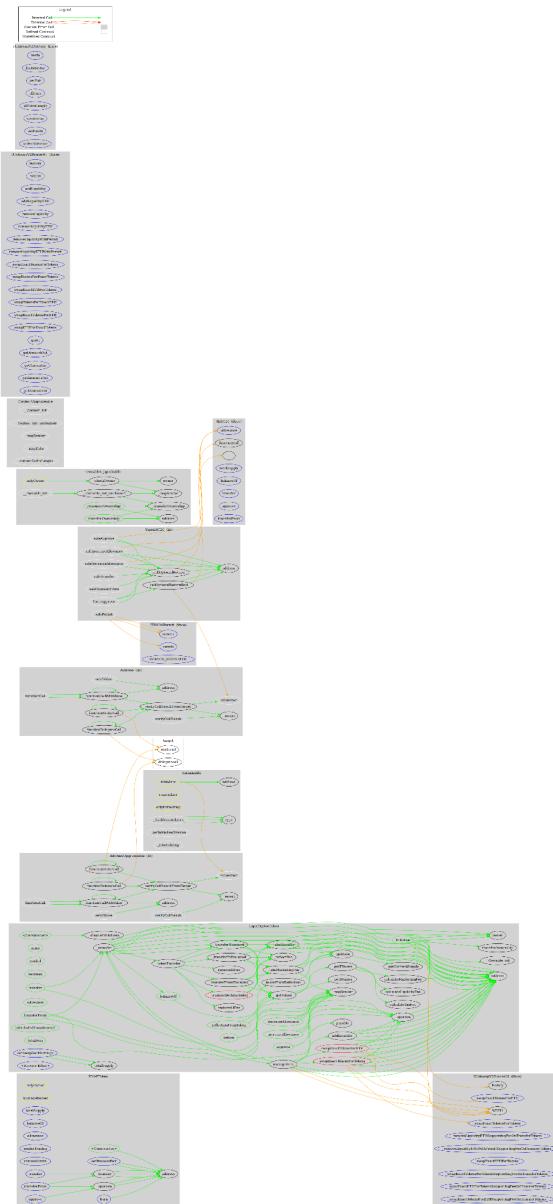
Contract		Type	Bases		
	Function Name		Visibility	Mutability	Modifiers
XNAPToken	Implementation				
		Public	✓	-	
	totalSupply	External		-	
	balanceOf	External		-	
	allowance	External		-	
	setPancakePair	External	✓		onlyOwner
	enableTrading	External	✓		onlyOwner
	removeLimits	External	✓		onlyOwner
	transfer	External	✓	-	
	approve	External	✓	-	
	transferFrom	External	✓	-	
	_transfer	Internal	✓		tradingAllowed
	_approve	Internal	✓		
	burn	External	✓	-	
IERC20	Interface				
	totalSupply	External		-	
	balanceOf	External		-	
	transfer	External	✓	-	

	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-

# Inheritance Graph



# Flow Graph



# Summary

XNAP Token contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. XNAP Token is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues.

## 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 TAC 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](http://cyberscope.io)