Cyberscope

A TAC Security Company

Audit Report

XNAP Token

October 2025

Contract XNAPToken

SHA256 d10dd9fddf0f36fa0095caf64ded09dffb8924ff5f92675af15dd360bd7acd18

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	Unresolved
●	BT	Burns Tokens	Unresolved
●	BC	Blacklists Addresses	Passed

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
	PAF	Potential Approval Frontrun	Unresolved
	CCR	Contract Centralization Risk	Unresolved
	L19	Stable Compiler Version	Unresolved

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

Badge Eligibility	Must Fix Criticals

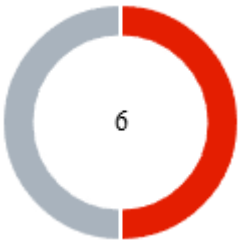
Audit Updates

Initial Audit	07 Oct 2025

Source Files

Filename	SHA256
Xnap_Smart_Contract.pdf	d10dd9fddf0f36fa0095caf64ded09dffb8924ff5f92675af15dd360bd7acd18

Findings Breakdown



Critical	3
Medium	0
Minor / Informative	3

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	3	0	0	0
Medium	0	0	0	0
Minor / Informative	3	0	0	0

ST - Stops Transactions

Criticality	Critical
Location	xnaptoken.sol#L59,72,101,109
Status	Unresolved

Description

The contract owner has the authority to stop the sales for all users excluding the owner. The owner may take advantage of it by using the `pause` functionality. As a result, the contract may operate as a honeypot.

Shell

```
modifier notPaused() {
require(!paused, "Transfers are paused");
_;
}
function pause() external onlyOwner {
paused = true;
emit Paused(msg.sender);
}
function transfer(address to, uint256 value) external
notPaused returns (bool)
function transferFrom(address from, address to, uint256
value) external notPaused returns (bool)
```

Recommendation

The team should carefully manage the private keys of the owner's 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 ownership, which will eliminate the threats but it is non-reversible.

MT - Mints Tokens

Criticality	Critical
Location	xnaptoken.sol#L80
Status	Unresolved

Description

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

```
Shell
function mint(address to, uint256 amount) external
onlyOwner {
require(to != address(0), "Zero address");
_totalSupply += amount;
_balances[to] += amount;
emit Mint(to, amount);
emit Transfer(address(0), to, amount);

}
```

Recommendation

The team should carefully manage the private keys of the owner's 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 ownership, which will eliminate the threats but it is non-reversible.

BT - Burns Tokens

Criticality	Critical
Location	xnaptoken.sol#L87
Status	Unresolved

Description

The contract owner has the authority to burn tokens from a specific address. The owner may take advantage of it by calling the `burn` function. As a result, the targeted address will lose the corresponding tokens.

```
Shell
function burn(address from, uint256 amount) external
onlyOwner {
require(from != address(0), "Zero address");
uint256 bal = _balances[from];
require(bal >= amount, "Insufficient balance");
_balances[from] = bal - amount;
_totalSupply -= amount;
emit Burn(from, amount);
emit Transfer(from, address(0), amount);
}
```

Recommendation

The team should carefully manage the private keys of the owner's 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 ownership, which will eliminate the threats but it is non-reversible.

PAF - Potential Approval Frontrun

Criticality	Minor / Informative
Location	xnaptoken.sol#L105
Status	Unresolved

Description

The contract implements only the standard `approve()` function to set allowances and omits the optional `increaseAllowance()` and `decreaseAllowance()` helpers. This exposes users to the ERC20 allowance race condition , where a spender may front-run an allowance change and spend both the old and new allowance amounts.

```
Shell
function approve(address spender, uint256 value) external
notPaused returns (bool) {
_approve(msg.sender, spender, value);
return true;
}
```

Recommendation

The team is recommended to implement the `increaseAllowance()` and `decreaseAllowance()` functions to safely adjust allowance values and follow ERC-20 best practices.

CCR - Contract Centralization Risk

Criticality	Minor / Informative
Location	xnaptoken.sol#L63,68,72,76,80,87
Status	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 transferOwnership(address newOwner)
external onlyOwner
function renounceOwnership() external onlyOwner

function pause() external onlyOwner
function unpause() external onlyOwner {
function mint(address to, uint256 amount) external
onlyOwner
function burn(address from, uint256 amount)
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.

L19 - Stable Compiler Version

Criticality	Minor / Informative
Location	xnaptoken.sol#L3
Status	Unresolved

Description

The `^` symbol indicates that any version of Solidity that is compatible with the specified version (i.e., any version that is a higher minor or patch version) can be used to compile the contract. The version lock is a mechanism that allows the author to specify a minimum version of the Solidity compiler that must be used to compile the contract code. This is useful because it ensures that the contract will be compiled using a version of the compiler that is known to be compatible with the code.

```shell
pragma solidity ^0.8.24;
```

Recommendation

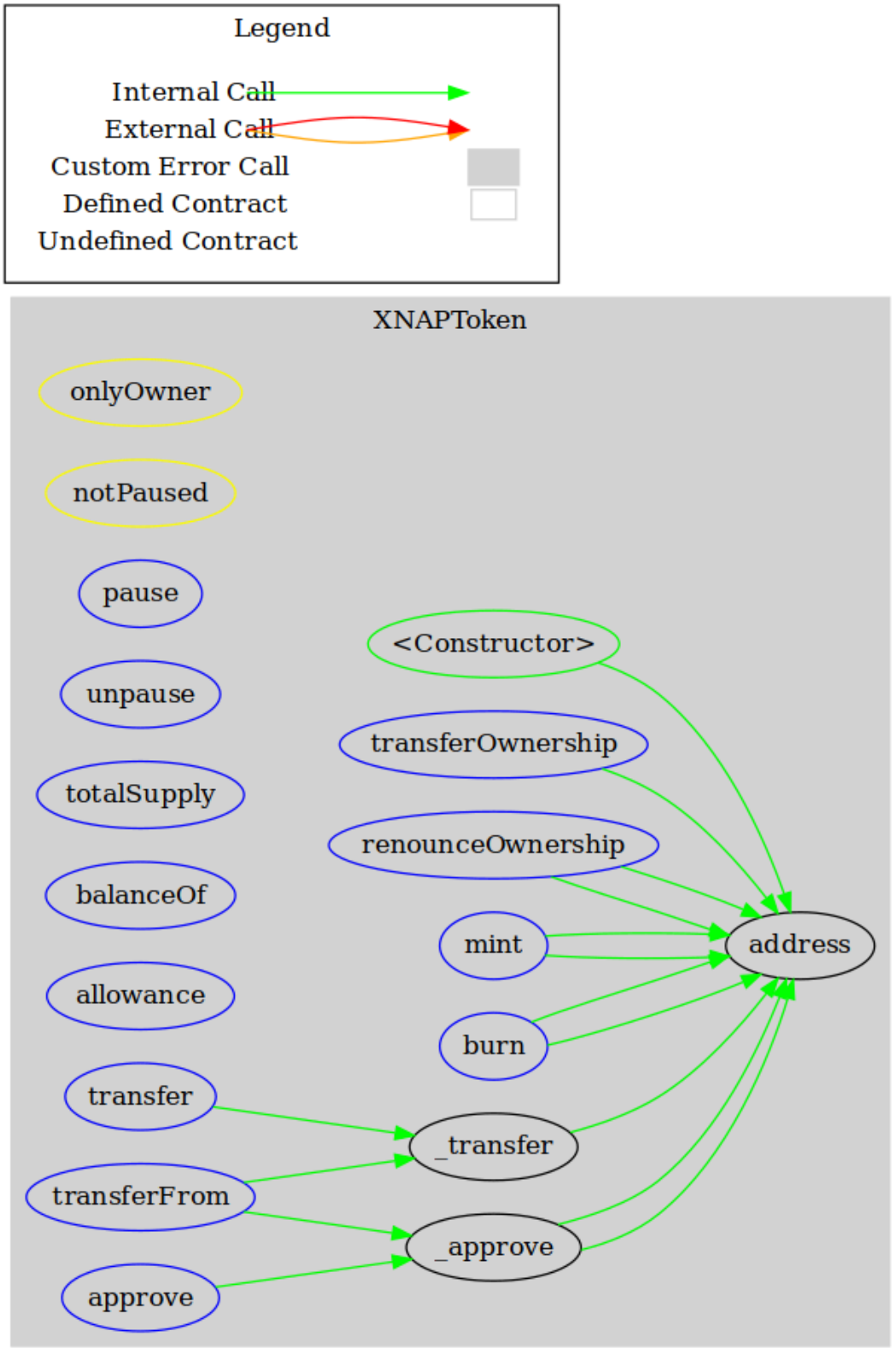
The team is advised to lock the pragma to ensure the stability of the codebase. The locked pragma version ensures that the contract will not be deployed with an unexpected version. An unexpected version may produce vulnerabilities and undiscovered bugs. The compiler should be configured to the lowest version that provides all the required functionality for the codebase. As a result, the project will be compiled in a well-tested LTS (Long Term Support) environment.

Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
XNAPToken	Implementation			
		Public	✓	-
	transferOwnership	External	✓	onlyOwner
	renounceOwnership	External	✓	onlyOwner
	pause	External	✓	onlyOwner
	unpause	External	✓	onlyOwner
	mint	External	✓	onlyOwner
	burn	External	✓	onlyOwner
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	transfer	External	✓	notPaused
	approve	External	✓	notPaused
	transferFrom	External	✓	notPaused
	_transfer	Internal	✓	
	_approve	Internal	✓	

Inheritance Graph

Flow Graph


Legend
Internal Call
External Call
Custom Error Call
Defined Contract
Undefined Contract
XNAPToken
onlyOwner
notPaused
pause
unpause
totalSupply
balanceOf
allowance
transfer
transferFrom
approve
<Constructor>
transferOwnership
renounceOwnership
mint
burn
_transfer
_approve
address

Summary

XNAP contract implements a token mechanism. This audit investigates security issues, business logic concerns and potential improvements. There are some functions that can be abused by the owner like stop transactions and mint tokens. If the contract owner abuses the mint functionality, then the contract will be highly inflated. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

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