BlockChain Media Security Audit Report

For WINNAZ Token Contract

Introduction

Blockchain-media.io is an officially registered company, based in the USA. We specialize in **Blockchain Security & Marketing**, including **Smart Contract Audits** and **KYC verification** for project teams.

This report assesses potential security issues in the **WINNAZ Token** smart contract, reviews inconsistencies between the code and intended functionality, and provides recommendations for improvement.

Disclaimer

Blockchain-media.io reports are **not an endorsement or disapproval** of any project. They do **not** guarantee absolute bug-free functionality, nor do they provide investment advice. Users should conduct their own due diligence before engaging with any project.

Project Overview

Project Name	WINNAZ Token
About The Project	Pepe, Turbo, Shiba and more - what's the point of having them locked in your wallet? Unleash their true potential with high-stakes lotteries, crafted to tap into each coin's unique power, all driven by Crypto $WINNAZ.
Token Symbol	$WINNAZ
Chain	Ethereum
Language	Solidity
Contract	WINNAZToken.sol
Unit Tests	Not Provided

Social Media & Links

Platform	**Link**
Website	https://cryptowinnaz.com
X	https://x.com/cryptowinnaz
Telegram	https://t.me/+kvP8SFnMre4yNmE9
Instagram	https://instagram.com/cryptowinnaz
Facebook	https://facebook.com/cryptowinnaz

Audit Summary

Version	**Delivery Date**	**Change Log**
v1.0	28th May 2025	Initial Audit

Note: This audit focuses on **security vulnerabilities** and **best practices**, not functional correctness (e.g., internal calculations).

File Overview

File Name	SHA-1 Hash
WINNAZToken.sol	2cad1bfef719df90cad15c51c907f53d5e138ca6

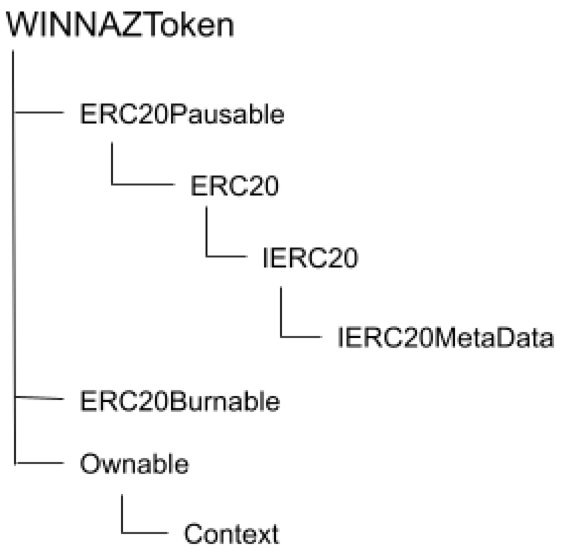
Imported Packages

- OpenZeppelin **ERC20Pausable**
- OpenZeppelin **ERC20Burnable**
- OpenZeppelin **Ownable**

Capabilities

Solidity Version	Can Receive Funds?	Uses Assembly?	Upgradeable?
^0.8.20	❌ No	❌ No	❌ No

Inheritance Graph



Audit Findings

1. Centralization Risks (Medium Severity)

Issue: The contract owner retains significant privileges, including:

- **Minting new tokens** (via `mint()`).
- **Pausing/unpausing all transfers** (via `pause()`/`unpause()`).

Risk: If the owner's private key is compromised, an attacker could:

- Mint unlimited tokens (up to `MAX_SUPPLY`).
- Freeze all transactions indefinitely.

Recommendation:

- Use a **multi-signature wallet** (e.g., Gnosis Safe) for ownership.
- Implement a **timelock** (48–72 hours) for privileged functions.
- Consider **renouncing ownership** after initial setup.

Resolution: The project team has informed us of their ownership renouncing schedule that resolves this issue entirely. The contract ownership will be renounced in 2 weeks time, after which this audit will be updated.

2. Lack of External Package Verification (Informational)

Issue: The contract imports OpenZeppelin libraries but does not verify their integrity via npm or direct source links.
Risk: Flattened contracts could contain modified or outdated versions.
Recommendation: Import OpenZeppelin directly via `@openzeppelin/contracts`.

3. Missing Zero-Address Checks (Low Severity)

Issue: The `mint()` function does not validate if `to` is a zero address.
Risk: Accidental token burns if tokens are minted to `address(0)`.
Fix: Add (in solidity):

```
  require(to != address(0), "Mint to zero address");
```

Resolution: This issue will be removed after the contract ownership is renounced.

4. No Supply Hard Cap Enforcement (Low Severity)

Issue: While `MAX_SUPPLY` is defined, the contract does not enforce it post-construction (e.g., preventing mints beyond deployment).
Recommendation: Renounce minting ability after initial supply distribution.
Resolution: This issue will be removed after the contract ownership is renounced.

Overall Security Assessment

Category	**Status**
Upgradeability	✅ Non-upgradeable
Ownership *	❌ Centralized (Owner-controlled)
Minting	✅ Restricted to `MAX_SUPPLY`
Burning	✅ Allowed (via `ERC20Burnable`)
Pausing	✅ Owner-controlled (Emergency use)

* Contract ownership will be renounced as per the schedule within 2 weeks time.

Final Audit Results

Severity	**Issues Found**
Critical	0
High	0
Medium	1 (Centralization)
Low	2 (Zero-address, Supply cap)
Informational	1 (Package imports)

Final Recommendations

1. **Mitigate centralization risks** via multi-sig + timelock.
2. **Verify OpenZeppelin imports** from npm.
3. **Add zero-address checks** in `mint()`.
4. **Consider renouncing ownership** after setup.

Note: It is noted that the project team has a schedule to renounce contract ownership in 2 weeks time. After the contract ownership is renounced, points 1, 3 and 4 are void.

Signed,
BlockChain-Media.io
Blockchain Security and Marketing | Smart Contract Audits | KYC | PR & Marketing