Alethea AI | Smart Contract Audit | BNB Chain

tags: `alethea` , `audit`

Latest revision: December 10th

Miguel Palhas **mpalhas@gmail.com** (mailto:mpalhas@gmail.com)

Table of Contents

- Alethea AI | Smart Contract Audit | BNB Chain
  - Table of Contents
  - Overview
  - Dates
  - Process
  - Coverage
  - Areas of Concern
  - Findings
  - Progress
    - AliERC20v2Base
    - BinanceAliERC20v2
    - NFTFactoryV2
  - Findings Summary
  - Detailed Findings
    - 1. BinanceAliERC20v2 comments refer to Polygon

Overview

A permissioned NFT minting factory in Solidity

Dates

- **March 10th**: Start date
- **March 11th**: Preliminary report

Process

This document, and all suggestions detailed here, is meant to be scrutinized and
discussed between both parties, in an ongoing collaborative process after the
first report delivery. Each suggestion may not be needed due to contextual
reasons, or limited understanding of external scope by the auditor.

Coverage

The following repositories were considered in-scope for the review:

- **https://github.com/AletheaAI/alethea-contracts** (https://github.com/AletheaAI/alethea-contracts) (revision `3dfcc8b`)

In particular, this audit focuses solely on the `BinanceAliERC20v2` and `NFTFactoryV2` contracts.

Areas of Concern

The investigation focused on the following:

- Looking for attack vectors that could impact the intended behaviour of the
  smart contract
- Checking test coverage, particularly for potentially dangerous scenarios and
  edge-cases
- Interaction with 3rd party contracts
- Use of solidity best practices
- Permissions and roles after deploy script is executed

Findings

Each issue has an assigned severity:

- **Informational**
- **Minor**
  issues are subjective in nature. They are typically suggestions around best
  practices or readability. Code maintainers should use

their judgement as to
whether to address such issues.

- **Gas**
  issues are related to gas optimizations.
- **Medium**
  issues are objective in nature but are not security vulnerabilities. These
  should be addressed unless there is a clear reason not to.
- **High**
  issues are security vulnerabilities that may
  not be directly exploitable or may require certain conditions in order to be
  exploited. All major issues should be addressed.
- **Critical** issues are directly exploitable security vulnerabilities that need to be fixed.

Progress

This table helps tracks the auditor’s progress through each section of the
codebase, and is not indicative of final results.

`AliERC20v2Base`

identifier	progr
TOKEN_UID	✅
name	✅
symbol	✅
decimals	✅
totalSupply	✅
tokenBalances	✅
votingDelegates	✅
votingPowerHistory	✅
totalSupplyHistory	✅
nonces	✅
usedNonces	✅
transferAllowances	✅
FEATURE_TRANSFERS	✅
FEATURE_UNSAFE_TRANSFERS	✅
FEATURE_OWN_BURNS	✅
FEATURE_BURNS_ON_BEHALF	✅
FEATURE_DELEGATIONS	✅
FEATURE_DELEGATIONS_ON_BEHALF	✅
FEATURE_ERC1363_TRANSFERS	✅
FEATURE_ERC1363_APPROVALS	✅
FEATURE_EIP2612_PERMITS	✅
FEATURE_EIP3009_TRANSFERS	✅
FEATURE_EIP3009_RECEPTIONS	✅
ROLE_TOKEN_CREATOR	✅
ROLE_TOKEN_DESTROYER	✅
ROLE_ERC20_RECEIVER	✅
ROLE_ERC20_SENDER	✅

identifier	progr
DOMAIN_TYPEHASH	✅
DOMAIN_SEPARATOR	✅
DELEGATION_TYPEHASH	✅
PERMIT_TYPEHASH	✅
TRANSFER_WITH_AUTHORIZATION_TYPEHASH	✅
RECEIVE_WITH_AUTHORIZATION_TYPEHASH	✅
CANCEL_AUTHORIZATION_TYPEHASH	✅
constructor	✅
supportsInterface	✅
transferAndCall/2	✅
transferAndCall/3	✅
transferFromAndCall/3	✅
transferFromAndCall/4	✅
approveAndCall/2	✅
approveAndCall/3	✅
_notifyTransferred	✅
_notifyApproved	✅
balanceOf	✅
transfer	✅
transferFrom	✅
safeTransferFrom	✅
unsafeTransferFrom	✅
__transferFrom	✅
approve	✅
__approve	✅
allowance	✅
increaseAllowance	✅
decreaseAllowance	✅

identifier	progr
mint	✅
burn	✅
permit	✅
authorizationState	✅
transferWithAuthorization	✅
receiveWithAuthorization	✅
cancelAuthorization	✅
__deriveSigner	✅
__useNonce	✅
votingPowerOf	✅
votingPowerAt	✅
votingPowerHistoryOf	✅
votingPowerHistoryLength	✅
totalSupplyAt	✅
entireSupplyHistory	✅
totalSupplyHistoryLength	✅
delegate	✅
__delegate	✅
delegateWithAuthorization	✅
__moveVotingPower	✅
__updateHistory	✅
__binaryLookup	✅
add	✅
sub	✅

`BinanceAliERC20v2`

identifier	progress
underlying	✅
constructor	✅

`NFTFactoryV2`

identifier	progress
totalSupplyHardcat	✅
usedNonces	✅
DOMAIN_SEPARATOR	✅
dwDOMAIN_TYPEHASH	✅
MINT_WITH_AUTHORIZATION_TYPEHASH	✅
CANCEL_AUTHORIZATION_TYPEHASH	✅
FEATURE_MINTING_WITH_AUTH	✅
ROLE_FACTORY_MINTER	✅
constructor	✅
mint	✅
__mint	✅
mintWithAuthorization	✅
authorizationState	✅
cancelAuthorization/5	✅
cancelAuthorization/1	✅
__deriveSigner	✅
__useNonce	✅

Findings Summary

Findings are listed in chronological order of discovery.

title	severity
1. BinanceAliERC20v2 comments refer to Polygon	**Informational**

Detailed Findings

1. BinanceAliERC20v2 comments refer to Polygon

Severity: Informational

The comments for `BinanceAliERC20v2` contract appear to be copied from the original `PolygonAliERC20v2` contract, and are not updated to reflect the new target

blockchain.