Cyberscope

Audit Report

My Bro

December 2024

Repository : https://github.com/breadNbutter42/BRO/tree/main

Commit : c86a5863a11247c5bf2bc8e173a4fedbf50992b8

Audited by © cyberscope

Analysis

🔴 Critical 🟠 Medium ⚪ Minor / Informative 🔵 Pass

Severity	Code	Description	Status
🔵	ST	Stops Transactions	Passed
🔵	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
●	DDP	Decimal Division Precision	SemiResolved
●	HV	Hardcoded Values	Unresolved
●	L04	Conformance to Solidity Naming Conventions	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

Repository	https://github.com/breadNbutter42/BRO/tree/main
Commit	c86a5863a11247c5bf2bc8e173a4fedbf50992b8

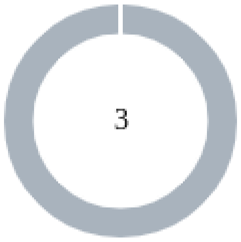
Audit Updates

Initial Audit	10 Dec 2024 https://github.com/cyberscope-io/audits/blob/main/6-bro/v1/audit.pdf
Corrected Phase 2	12 Dec 2024
Test Deploy	https://sepolia.etherscan.io/address/0x760e884B15669eA40cF1562dB6876f7bdCdd1B9a

Source Files

Filename	SHA256
BroTokenWithPresale.sol	d4c6d2c28a8f13e00ad053e5c597012316bedaef6e8edf97c39069d9fbface5a

Findings Breakdown



●	Critical	0
●	Medium	0
●	Minor / Informative	3

Severity		Unresolved	Acknowledged	Resolved	Other
●	Critical	0	0	0	0
●	Medium	0	0	0	0
●	Minor / Informative	2	0	0	1

DDP - Decimal Division Precision

Criticality	Minor / Informative
Location	BroTokenWithPresale.sol#L479
Status	SemiResolved

Description

Division of decimal (fixed point) numbers can result in rounding errors due to the way that division is implemented in Solidity. Thus, it may produce issues with precise calculations with decimal numbers.

Solidity represents decimal numbers as integers, with the decimal point implied by the number of decimal places specified in the type (e.g. decimal with 18 decimal places). When a division is performed with decimal numbers, the result is also represented as an integer, with the decimal point implied by the number of decimal places in the type. This can lead to rounding errors, as the result may not be able to be accurately represented as an integer with the specified number of decimal places.

Hence, the splitted shares will not have the exact precision and some funds may not be calculated as expected.

```
amount_ = (totalAvaxUserSent[buyer_] * PRESALERS_BRO_SUPPLY) /
totalAvaxPresale;
```

Recommendation

The team is advised to take into consideration the rounding results that are produced from the solidity calculations. The contract could calculate the subtraction of the divided funds in the last calculation in order to avoid the division rounding issue.

HV - Hardcoded Values

Criticality	Minor / Informative
Location	BroTokenWithPresale.sol#L95
Status	Unresolved

Description

The contract contains multiple instances where numeric values are directly hardcoded into the code logic rather than being assigned to constant variables with descriptive names. Hardcoding such values can lead to several issues, including reduced code readability, increased risk of errors during updates or maintenance, and difficulty in consistently managing values throughout the contract. Hardcoded values can obscure the intent behind the numbers, making it challenging for developers to modify or for users to understand the contract effectively.

```
uint256 private constant _TRILLIONS_SUPPLY = 420690 * 10**9 *
10**18;
uint256 private constant _FULL_MOON_TIME = 1734254100;
uint256 private constant _HOURS_TO_PREP_IDO = 2 hours;
uint256 private constant _TIMESTAMP_BUFFER = 1 minutes;
uint256 private constant _LENGTH_OF_WL_PHASE = 5 minutes;
uint256 private constant _TOTAL_PHASES = 4;
```

Recommendation

It is recommended to replace hardcoded numeric values with variables that have meaningful names. This practice improves code readability and maintainability by clearly indicating the purpose of each value, reducing the likelihood of errors during future modifications. Additionally, consider using constant variables which provide a reliable way to centralize and manage values, improving gas optimization throughout the contract.

L04 - Conformance to Solidity Naming Conventions

Criticality	Minor / Informative
Location	BroTokenWithPresale.sol
Status	Unresolved

Description

The Solidity style guide is a set of guidelines for writing clean and consistent Solidity code. Adhering to a style guide can help improve the readability and maintainability of the Solidity code, making it easier for others to understand and work with.

The followings are a few key points from the Solidity style guide:

1. Use camelCase for function and variable names, with the first letter in lowercase (e.g., myVariable, updateCounter).
2. Use PascalCase for contract, struct, and enum names, with the first letter in uppercase (e.g., MyContract, UserStruct, ErrorEnum).
3. Use uppercase for constant variables and enums (e.g., MAX_VALUE, ERROR_CODE).
4. Use indentation to improve readability and structure.
5. Use spaces between operators and after commas.
6. Use comments to explain the purpose and behavior of the code.
7. Keep lines short (around 120 characters) to improve readability.

```
uint256 public constant PRESALERS_BRO_SUPPLY =
(_TOTAL_SUPPLY_TO_MINT * _FIFTY_PERCENT) / 100;
```

Recommendation

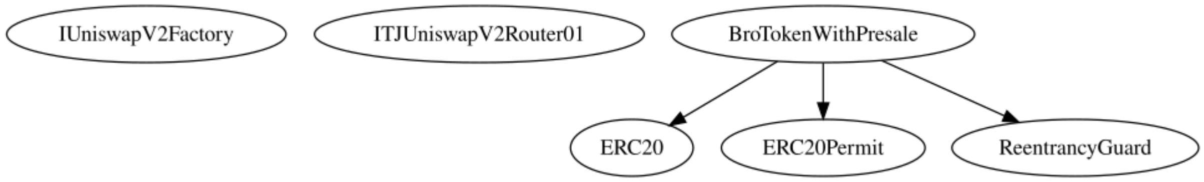
By following the Solidity naming convention guidelines, the codebase increased the readability, maintainability, and makes it easier to work with.
Find more information on the Solidity documentation
https://docs.soliditylang.org/en/stable/style-guide.html#naming-conventions.

Functions Analysis

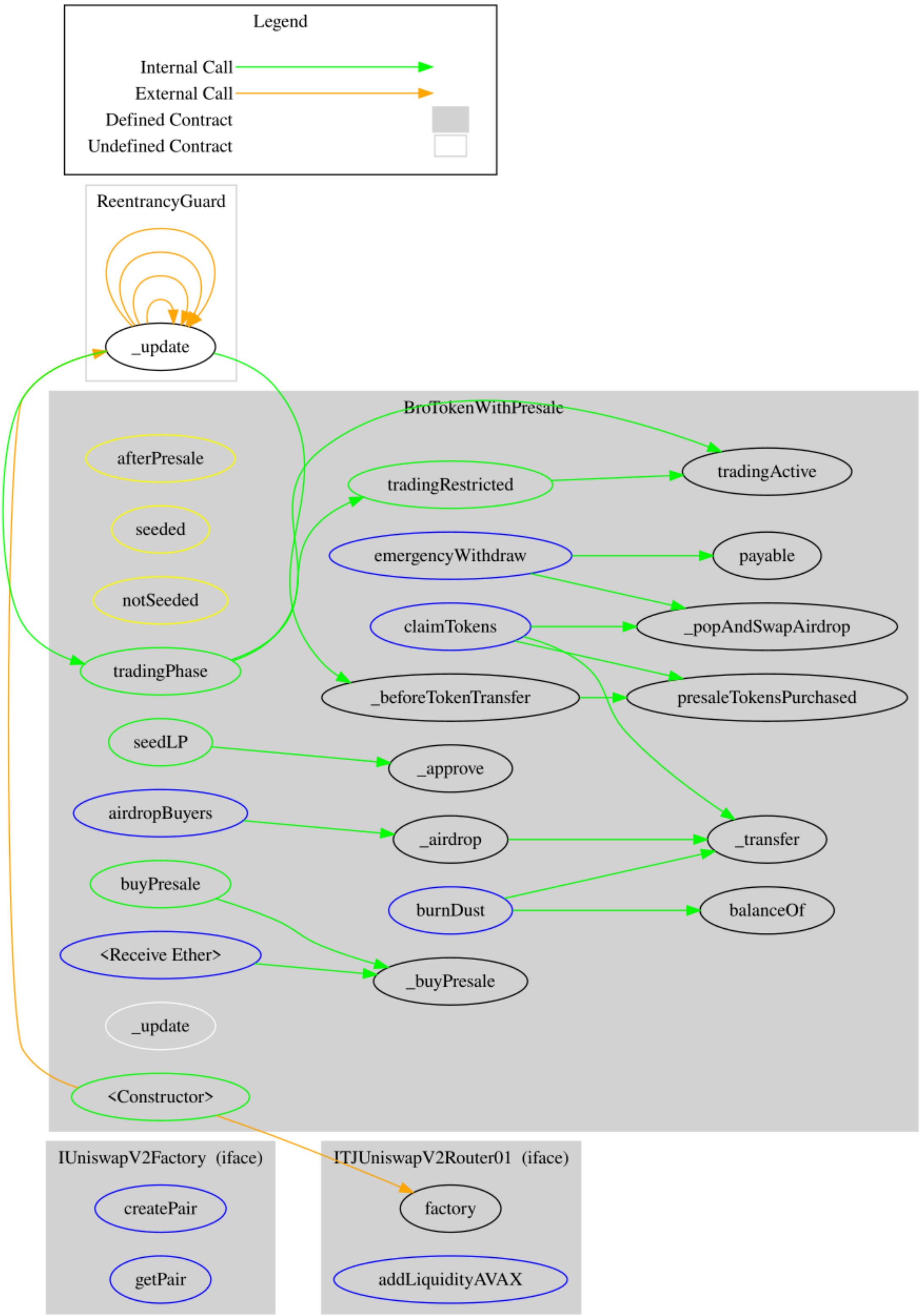
Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
BroTokenWithPresale	Implementation	ERC20, ERC20Permit, ReentrancyGuard		
		Public	✓	ERC20 ERC20Permit
	tradingActive	Public		-
	tradingRestricted	Public		-
	tradingPhase	Public		-
	presaleTokensPurchased	Public		afterPresale
	seedLP	Public	✓	nonReentrant afterPresale notSeeded
	buyPresale	Public	Payable	-
	emergencyWithdraw	External	✓	nonReentrant notSeeded
	burnDust	External	✓	nonReentrant
	airdropBuyers	External	✓	nonReentrant afterPresale seeded
	claimTokens	External	✓	nonReentrant afterPresale seeded
	_popAndSwapAirdrop	Private	✓	
	_update	Internal	✓	

	_beforeTokenTransfer	Private	✓	
	_buyPresale	Private	✓	nonReentrant notSeeded
	_airdrop	Private	✓	
		External	Payable	-

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
ReentrancyGuard
_update
BroTokenWithPresale
afterPresale
seeded
notSeeded
tradingPhase
seedLP
airdropBuyers
buyPresale
<Receive Ether>
_update
<Constructor>
tradingRestricted
emergencyWithdraw
claimTokens
_beforeTokenTransfer
_approve
_airdrop
burnDust
_buyPresale
tradingActive
payable
_popAndSwapAirdrop
presaleTokensPurchased
_transfer
balanceOf
IUniswapV2Factory (iface)
createPair
getPair
ITJUniswapV2Router01 (iface)
factory
addLiquidityAVAX

Summary

My Bro contract implements a token and a presale mechanism. This audit investigates security issues, business logic concerns and potential improvements. My Bro 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 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.

The Cyberscope team

cyberscope.io