AstraSec

Buzzed Beras
Security Audit Report

April 16, 2025

Contents

1 Introduction

1.1 About Buzzed Beras

Buzzed Beras is a deflationary, on-chain Player-versus-Player (PvP) survival game deployed on the Berachain blockchain. The protocol centers around Grizzly NFTs, which players mint by burning $COMB tokens and must sustain by feeding $COMB daily to prevent permanent on-chain deactivation ("death"). Players can acquire $COMB through competitive wallet raiding or by staking in the "Detox Den." The game's deflationary mechanism progressively reduces the $COMB supply through burns, culminating in an endgame where the last three wallets holding surviving Grizzly NFTs share the remaining liquidity pool, termed the "Final Fix." This strategic survival mechanic incentivizes high-stakes competition for a jackpot-style reward.

1.2 Source Code

The following source code was reviewed during the audit:

▶ https://github.com/chimpytuts/buzzed-beras-contracts

▶ CommitID: d569e44315def5ed0824514fa31328119e5e7172

And this is the final version representing all fixes implemented for the issues identified in the audit:

▶ https://github.com/chimpytuts/buzzed-beras-contracts

▶ CommitID: 94aa081713da87c62e8434fcee7076f917e18511

1.3 Revision History

Version	Date	Scope
v1.0	April 16, 2025	$COMB contract

2 Overall Assessment

This report has been compiled to identify issues and vulnerabilities within the $COMB contract of the Buzzed Beras protocol. Throughout this audit, we identified a total of 4 issues spanning various severity levels. By employing auxiliary tool techniques to supplement our thorough manual code review, we have discovered the following findings.

Severity	Count	Acknowledged	Won't Do	Addressed
Critical	—	—	—	—
High	—	—	—	—
Medium	2	1	—	1
Low	2	1	—	1
Informational	—	—	—	—
Total	4	2	—	2

3 Vulnerability Summary

3.1 Overview

Click on an issue to jump to it, or scroll down to see them all.

M-1	Improved Initial Liquidity Launch for $COMB
M-2	Inconsistency Between Document And Implementation
L-1	Potential Risks Associated with Centralization
L-2	Suggested Event Emission for Key Operations

3.2 Security Level Reference

In web3 smart contract audits, vulnerabilities are typically classified into different severity levels based on the potential impact they can have on the security and functionality of the contract. Here are the definitions for critical-severity, high-severity, medium-severity, and low-severity vulnerabilities:

Severity	Acknowledged
C-X (Critical)	A severe security flaw with immediate and significant negative consequences. It poses high risks, such as unauthorized access, financial losses, or complete disruption of functionality. Requires immediate attention and remediation.
H-X (High)	Significant security issues that can lead to substantial risks. Although not as severe as critical vulnerabilities, they can still result in unauthorized access, manipulation of contract state, or financial losses. Prompt remediation is necessary.
M-X (Medium)	Moderately impactful security weaknesses that require attention and re-mediation. They may lead to limited unauthorized access, minor financial losses, or potential disruptions to functionality.
L-X (Low)	Minor security issues with limited impact. While they may not pose significant risks, it is still recommended to address them to maintain a robust and secure smart contract.
I-X (Informational)	Warnings and things to keep in mind when operating the protocol. No immediate action required.
U-X (Undetermined)	Identified security flaw requiring further investigation. Severity and impact need to be determined. Additional assessment and analysis are necessary.

3.3 Vulnerability Details

3.3.1 [M-1] Improved Initial Liquidity Launch for $COMB

TARGET	CATEGORY	IMPACT	LIKELIHOOD	STATUS
COMB.sol	Business Logic	Medium	Low	Confirmed

The Buzzed Beras protocol utilizes $COMB, a GameFi token, which can be purchased during an ICO using $HONEY at a 1:1 price ratio via the COMB::mintForFee() function. After the ICO, 80% of the raised $HONEY and minted $COMB are allocated to establish a <$COMB, $HONEY> liquidity pool (LP) on Kodiak, with LP tokens locked by a multi-signature Protocol-Owned Liquidity (POL) wallet. A potential risk may exist in the liquidity pool initialization process. Specifically, a malicious actor could front-run the POL's liquidity addition by creating the LP on Kodiak in advance and adding minimal liquidity with an artificially high $COMB price. When the POL subsequently adds liquidity using the ICO-raised funds, the skewed price ratio would result in the POL contributing nearly all of its $HONEY but receiving disproportionately little $COMB in the pool. The attacker could then sell their $COMB at the inflated price, draining a significant portion of the $HONEY from the pool and undermining the protocol's liquidity.

```
                      buzzed-beras-contracts - COMB.sol
618   function _mintForFee(uint256 _amount) private {
619       require(finished,"You are to late");
620       require(saleStarted,"Comb nor ready yet");
621       require(preMinted + _amount <= maxPreMint,"No more comb in the market");
622       uint mintFee = _amount * mintPrice;
623       uint teamFee = mintFee * teamCut / 1000;
624
625       IERC20(honeyToken).transfer(POL, mintFee- teamFee);
626       IERC20(honeyToken).transfer(owner(), teamFee);
627
628       uint polToMint = _amount - (teamCut * _amount / 1000);
629       preMinted += _amount;
630       _mint(msg.sender, _amount);
631       _mint(POL,polToMint);
632   }
```

Remediation Implement a safeguarded liquidity launch process, check and rebalance the price before adding the initial liquidity.

Response By Team The team confirmed they will follow the safeguarded launch process.

3.3.2 [M-2] Inconsistency Between Document And Implementation

TARGET	CATEGORY	IMPACT	LIKELIHOOD	STATUS
COMB.sol	Coding Mistakes	Medium	Medium	Addressed

The $COMB token contract, integral to the Buzzed Beras protocol, defines a constant maxPreMint variable initialized to 2,000,000 $COMB, representing the maximum mint capacity for the Initial Coin Offering (ICO). The contract also mints an additional 80% of the minted $COMB to the Protocol-Owned Liquidity (POL) address for the initial liquidity pool on Kodiak, resulting in a total mint capacity of 3,600,000 $COMB (2 million + 1.6 million).

```
                          buzzed-beras-contracts - COMB.sol
604   uint256 public constant maxPreMint = 2_000_000e18;
605   address public immutable honeyToken; // HONEY token address
606
607   address public immutable POL; // address to send the tokens that are going to be used as POL
608
609   function mintForFee(uint amount) public{
610       require(amount > 0, "0 tokens");
611       IERC20(honeyToken).transferFrom(msg.sender,address(this), amount);
612       _mintForFee(amount);
613   }
```

However, the project's documentation states that the maximum mint capacity for the ICO is 5,000,000 $COMB, with a total supply of 9,000,000 $COMB after accounting for the additional 80% minted for liquidity (5 million + 4 million). This discrepancy between the contract implementation (maxPreMint = 2,000,000) and the documentation (5,000,000 $COMB) indicates an inconsistency that could lead to operational or user trust issues. Specifically, the maxPreMint variable does not reflect the documented ICO capacity, potentially causing confusion or unexpected behavior during the ICO process.
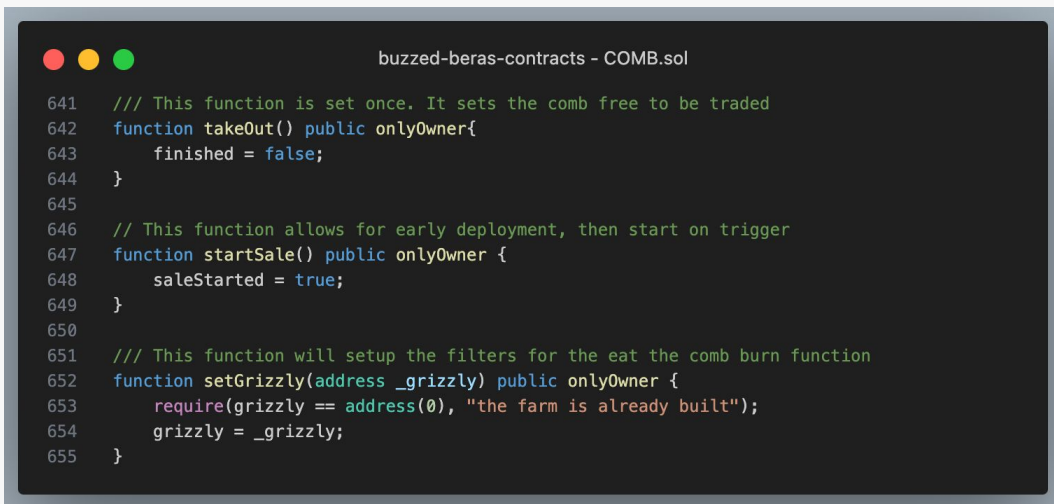
> The BeraNomics of the Buzzed Bera project are entirely contingent on the outcome and participation of "The Great Pollination Event". However with a maximum mint cap of 5,000,000 $COMB, the maximum total supply can only be 9,000,000 $COMB. From there deflationary mechanics take effect.

Remediation To resolve the inconsistency, the maxPreMint variable in the $COMB contract should be updated to reflect the documented ICO mint capacity.

3.3.3 [L-1] Potential Risks Associated with Centralization

TARGET	CATEGORY	IMPACT	LIKELIHOOD	STATUS
COMB.sol	Security	Low	Low	Acknowledged

The $COMB token contract, a core component of the Buzzed Beras protocol, grants significant administrative privileges to a single owner address, enforced by the onlyOwner modifier. The owner can execute critical functions, including starting and stopping the Initial Coin Offering (ICO) and setting the address of the Grizzly contract authorized to burn $COMB tokens. This high degree of centralization introduces risks to the protocol's decentralization and security, as the owner's actions could unilaterally affect the protocol's operations.

```
                          buzzed-beras-contracts - COMB.sol
641    /// This function is set once. It sets the comb free to be traded
642    function takeOut() public onlyOwner{
643        finished = false;
644    }
645
646    // This function allows for early deployment, then start on trigger
647    function startSale() public onlyOwner {
648        saleStarted = true;
649    }
650
651    /// This function will setup the filters for the eat the comb burn function
652    function setGrizzly(address _grizzly) public onlyOwner {
653        require(grizzly == address(0), "the farm is already built");
654        grizzly = _grizzly;
655    }
```

Remediation To mitigate the identified issue, it is recommended to introduce multi-sig mechanism to undertake the role of the privileged accounts. Moreover, it is advisable to implement timelocks to govern all modifications to the privileged operations.

Response By Team This issue has been acknowledged by the team.

3.3.4 [L-2] Suggested Event Emission for Key Operations

TARGET	CATEGORY	IMPACT	LIKELIHOOD	STATUS
COMB.sol	Coding Practice	Low	Low	Addressed

The $COMB token contract, a critical component of the Buzzed Beras protocol, does not emit sufficient events to log operations or state changes, particularly for privileged actions restricted by the onlyOwner modifier. We recommend emitting events for significant operations to ensure transparency, auditability, and off-chain monitoring. The absence of these events reduces the contract's observability and could hinder user trust and ecosystem integration.

```
                        buzzed-beras-contracts - COMB.sol
641    /// This function is set once. It sets the comb free to be traded
642    function takeOut() public onlyOwner{
643        finished = false;
644    }
645
646    // This function allows for early deployment, then start on trigger
647    function startSale() public onlyOwner {
648        saleStarted = true;
649    }
650
651    /// This function will setup the filters for the eat the comb burn function
652    function setGrizzly(address _grizzly) public onlyOwner {
653        require(grizzly == address(0), "the farm is already built");
654        grizzly = _grizzly;
655    }
```

Remediation To enhance transparency and align with smart contract best practices, we recommend implementing event emissions for all critical and privileged operations in the $COMB token contract.

4 Appendix

4.1 About AstraSec

AstraSec is a blockchain security company that serves to provide high-quality auditing services for blockchain-based protocols. With a team of blockchain specialists, AstraSec maintains a strong commitment to excellence and client satisfaction. The audit team members have extensive audit experience for various famous DeFi projects. AstraSec's comprehensive approach and deep blockchain understanding make it a trusted partner for the clients.

4.2 Disclaimer

The information provided in this audit report is for reference only and does not constitute any legal, financial, or investment advice. Any views, suggestions, or conclusions in the audit report are based on the limited information and conditions obtained during the audit process and may be subject to unknown risks and uncertainties. While we make every effort to ensure the accuracy and completeness of the audit report, we are not responsible for any errors or omissions in the report.

   We recommend users to carefully consider the information in the audit report based on their own independent judgment and professional advice before making any decisions. We are not responsible for the consequences of the use of the audit report, including but not limited to any losses or damages resulting from reliance on the audit report.

   This audit report is for reference only and should not be considered a substitute for legal documents or contracts.

4.3 Contact

Phone	+86 156 0639 2692
Email	contact@astrasec.ai
Twitter	https://twitter.com/AstraSecAI