Cyberscope

Audit Report

Tea-Fi

June 2025

Files TeaFiMysteryBoxManager

Sha256 9b47ab038a1f0d9468f6e47e13656ae8f80ab6018deeb15583dfffec209e564c

Audited by © cyberscope

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

Audit Scope

The current contract heavily relies on the `trustedForwarder_` external contract, to perform crucial functionalities. While this dependency enables important functionality, any interactions with this external contract should be carefully reviewed and handled, as it is beyond the scope of this audit. The behavior and security of this external contract have not been assessed as part of this audit, and any interactions with it should be treated with caution to mitigate potential risks.

Audit Updates

Initial Audit	24 Jun 2025 https://github.com/cyberscope-io/audits/blob/main/tea-fi/v1/mysteryBox.pdf
Corrected Phase 2	29 Jun 2025

Source Files

Filename	SHA256
TeaFiMysteryBoxManager.sol	9b47ab038a1f0d9468f6e47e13656ae8f80ab6018deeb15583dfffec209e564c
permit/PermitManagement.sol	a3d5ae129978749fdec94246e1b6735bf0a869ba4ad69a08ecda8359520258d9
permit/interfaces/IPermitManager.sol	4b9c998c45ccd0a676da9d0bbf684d55df55c893c031d74006ce36cf93ef2277
interfaces/ZeroAddressError.sol	83642b852ae173732f849ec7dfe02b6ba5bf0fbc54f4571253c95628ae2cd1aa
interfaces/ITeaFiMysteryBoxManager.sol	25bf6e1da541843f82d4d47c678ac3e35805832d6d1db7df1a480e928878265d

Overview

TeaFiMysteryBoxManager Contract

The `TeaFiMysteryBoxManager` smart contract is a solution for managing and validating the opening of mystery boxes within the TeaFi ecosystem. It incorporates security measures, decentralized execution capabilities, and integrates meta-transaction support, ensuring seamless interaction with blockchain users through trusted forwarders.

openMysteryBox Functionalities

The `openMysteryBox` function is responsible for managing the unlocking of mystery boxes. It validates each operation against predefined criteria and ensures that users do not claim boxes more than once per day. Upon successful validation, the contract records the last claimed day for each user, enhancing both operational efficiency and user experience.

Additionally, the function includes **fee management**, allowing the collection of fees in either native tokens (e.g., ETH) or ERC20 tokens. If a fee is defined, it is securely processed and sent to the configured treasury address. For native fees, excess ETH is refunded to the sender. For ERC20 fees, the contract leverages permit-based mechanisms through the `PermitManagement` module, enabling seamless and gas-efficient transfers.

Signature Verification and Security Measures

To maintain security, the contract employs EIP-712-based signature verification mechanisms, ensuring that only authorized operators can initiate box openings. Each transaction is validated using cryptographic signatures, protecting against tampering. Additionally, nonce management prevents replay attacks and ensures transaction integrity for each operator–user pair.

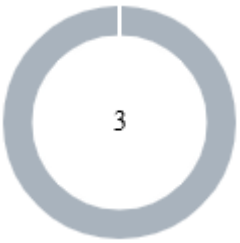
Operator Role and Access Control

The contract uses OpenZeppelin's access control features to manage operator roles. By assigning the `OPERATOR_ROLE`, the contract restricts access to critical operations, ensuring that only authorized operators can validate and approve box openings. This helps prevent misuse and preserves the integrity of the system.

Integration with External Interfaces

The contract supports EIP-712 for standardized message signing and verification. This enables compatibility with wallets and decentralized applications (dApps) that support off-chain signatures and meta-transactions, offering a seamless experience to users and external services interacting with the TeaFi ecosystem.

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	0	3	0	0

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
●	AME	Address Manipulation Exploit	Acknowledged
●	CCR	Contract Centralization Risk	Acknowledged
●	SRE	Spender Role Ensurance	Acknowledged

AME - Address Manipulation Exploit

Criticality	Minor / Informative
Location	TeaFiMysteryBoxManager.sol#L67,174 interfaces/ITeaFiMysteryBoxManager.sol#L24
Status	Acknowledged

Description

The contract's design includes functions that accept external contract addresses as parameters without performing adequate validation or authenticity checks. This lack of verification introduces a significant security risk, as input addresses could be controlled by attackers and point to malicious contracts. Such vulnerabilities could enable attackers to exploit these functions, potentially leading to unauthorized actions or the execution of malicious code under the guise of legitimate operations.

```
function openMysteryBox(OpenBoxParam calldata param) external
payable nonReentrant {
    _validateInput(param);
    if (param.fee.amount > 0) {
        _sendFee(param.fee);
    }
    //...
}

function _sendFee(Fee calldata fee) private {
    //...
    else {
        if (msg.value > 0) revert NotNativeTransfer();
        _receivePayment(fee.token, treasury, fee.amount,
fee.tokenData, fee.permit2Data);
    }
}
```

```
struct Fee {
    address token;
    uint256 amount;
    bytes tokenData;
    bytes permit2Data;
}
```

Recommendation

To mitigate this risk and enhance the contract's security posture, it is imperative to incorporate comprehensive validation mechanisms for any external contract addresses passed as parameters to functions. This could include checks against a whitelist of approved addresses, verification that the address implements a specific contract interface or other methods that confirm the legitimacy and integrity of the external contract. Implementing such validations helps prevent malicious exploits and ensures that only trusted contracts can interact with sensitive functions.

Team Update

The team has acknowledged that this is not a security issue and states: *The fee.token value is part of the signed payload validated using EIP-712. This payload is generated and signed by the backend operator that holds the OPERATOR_ROLE. As part of the signing process, the backend enforces strict validation, including checking that the fee.token address is on an allowlisted set of approved tokens. Because the contract only accepts box-opening requests accompanied by valid operator signatures, no arbitrary or malicious inputs can be introduced on-chain.*

CCR - Contract Centralization Risk

Criticality	Minor / Informative
Location	TeaFiMysteryBoxManager.sol#L46,47,106
Status	Acknowledged

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.

```
_grantRole(DEFAULT_ADMIN_ROLE, owner);
_grantRole(OPERATOR_ROLE, operator);
...
if (!hasRole(OPERATOR_ROLE, operator) || operator ==
address(0))
```

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.

Team Update

The team has acknowledged that this is not a security issue and states: *The default admin role is held by a Gnosis Safe multisig wallet configured with 3 signers and requiring a minimum of 2 approvals to authorize any action, so that administrative control is not centralized in a single entity. Operator roles are assigned to wallets managed through Utila's MPC infrastructure, which securely distributes private key control across multiple independent parties, reducing the risk of key compromise or single-point failure.*

SRE - Spender Role Ensurance

Criticality	Minor / Informative
Location	TeaFiMysteryBoxManager.sol#L174
Status	Acknowledged

Description

The contract is using the `PermitManagement` 's `_receivePayment` function to send an amount of tokens to the `treasury` address. However, this function calls a `permitManager` 's `executePermitTransfer` that reverts if the caller does not have the `SPENDER_ROLE` .

```
_receivePayment(fee.token, treasury, fee.amount, fee.tokenData,
fee.permit2Data);
```

Recommendation

The team should ensure that `TeaFiMysteryBoxManager` has the `SPENDER_ROLE` in the `permitManager` contract to ensure that the function works as intended.

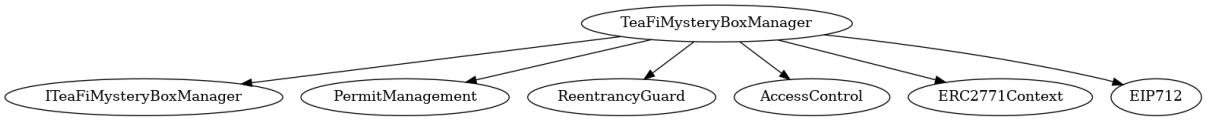
Team Update

The team has acknowledged that this is not a security issue and states: *Assigning the SPENDER_ROLE to TeaFiMysteryBoxManager is a required step during deployment and is part of the established deployment procedure. This role grant is explicitly handled in deployment scripts and thoroughly documented so that developers do not overlook it.*

Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
TeaFiMysteryBoxManager	Implementation	ITeaFiMysteryBoxManager, PermitManagement, ReentrancyGuard, AccessControl, ERC2771Context, EIP712		
		Public	✓	ERC2771Context EIP712 PermitManagement
	openMysteryBox	External	Payable	nonReentrant
	hashTypedDataV4	External		-
	_validateInput	Private	✓	
	_verifySignature	Private	✓	
	_sendFee	Private	✓	
	_msgSender	Internal		
	_msgData	Internal		
	_contextSuffixLength	Internal		
ITeaFiMysteryBoxManager	Interface			
	openMysteryBox	External	Payable	-

Inheritance Graph



Summary

Tea-Fi contract implements a utility and rewards mechanism. This audit investigates security issues, business logic concerns and potential improvements.

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