expelee

Building the Futuristic Blockchain Ecosystem

SECURITY AUDIT REPORT

Cremation Token

SCOPE AND SUMMARY

The following smart contracts were in scope of the audit:

- cremation_token

The following number of issues were found, categorized by their severity:

Severity	Found	Pending	Resolved
● High	0	1	0
● Medium	0	0	0
● Low	0	4	0
● Informational	0	0	0
Total	0	5	0

TABLE OF CONTENTS

OVERVIEW

This audit report evaluates the security of the "cremation-token" Rust smart contract. The contract is designed to implement a token with specific functionalities, including buy, sell, and transfer taxes, as well as interactions with the Terraswap router for token swaps. The contract also inherits and extends functionalities from the cw20_base and classic_terraswap libraries.

Audit Date	**11 Aug 2023**

PRIVILEGES

Creator Privileges

- set_config

Owner Privileges

- update_owner
- update_collecting_tax_address
- update_tax_info
- set_tax_free_address

RISK CLASSIFICATION

High Risk

Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, of the contract and its functions. Must be fixed as soon as possible.

Medium Risk

Bugs or issues with that may be subject to exploit, though their impact is somewhat limited. Must be fxed as soon as possible.

Low Risk

Effects are minimal in isolation and do not pose a signifcant danger to the project or its users. Issues under this classifcation are recommended to be fixed nonetheless.

Informational

A vulnerability that have informational character but is not effecting any of the code

HIGH RISK FINDINGS

[H-01] No Validation for Tax Fractions and Limit in update_tax_info

Severity : High

Description

• The update_tax_info function allows the contract owner to update tax information, including buy tax, sell tax, and transfer tax fractions. How ever, there is no validation for the provided tax fractions, which can lead to unintended behavior or vulnerabilities if incorrect fractions are used.

Mitigation

• Before updating the tax information, validate the provided tax fractions to ensure they are valid fractions and within acceptable ranges.

LOW RISK FINDINGS

[L-01] Missing Address Validation in set_config

Severity : Low

Description

- The set_config function does not validate the provided terraswap_router and terraswap_pair addresses.

Mitigation

- Validate the *terraswap_router* and *terraswap_pair* addresses before storing them in the contract's storage. Use the *addr_validate* function to ensure that the addresses are correctly formatted and correspond to legitimate contracts within the Terra ecosystem. If the validation fails, reject the transaction and emit an InvalidAddress error. This will prevent unauthorized changes to critical contract configuration, ensuring that only authorized addresses can be used for routing and pairing.

LOW RISK FINDINGS

[L-02] Missing Address Validation in update_owner

Severity : Low

Description

• The update_owner function allows the current owner to update the contract's owner to a new address without validating if the new owner's address is valid. This can lead to unintended behavior or potential vulnerabilities if an invalid or malicious address is used.

Mitigation

• Before updating the contract's owner to the new address, validate the new owner's address using the deps.api.addr_validate function. If the address is not valid, reject the transaction and emit an InvalidAddress error.

LOW RISK FINDINGS

[L-02] Missing Address Validation in update_owner

Severity : Low

Description

• The update_owner function allows the current owner to update the contract's owner to a new address without validating if the new owner's address is valid. This can lead to unintended behavior or potential vulnerabilities if an invalid or malicious address is used.

Mitigation

• Before updating the contract's owner to the new address, validate the new owner's address using the deps.api.addr_validate function. If the address is not valid, reject the transaction and emit an InvalidAddress error.

LOW RISK FINDINGS

[L-04] Missing Address Validation in set_tax_free_address

Severity : Low

Description

• Function allows the contract owner to mark addresses as tax-free, which exempts them from taxation. The audit focused on input validation and owner privileges related to this function, as well as ensuring that the owner's authority is not compromised.

• The set_tax_free_address function does not validate the address provided as input. This lack of validation could allow the owner to mark an invalid address as tax-free, leading to unexpected behavior or vulnerabilities.

• The set_tax_free_address function allows the contract owner to mark an address as tax-free. However, there is no mechanism to ensure that the owner is assigning this privilege to a legitimate address. This could lead to the owner unintentionally or maliciously assigning the tax-free privilege to an unauthorized address.

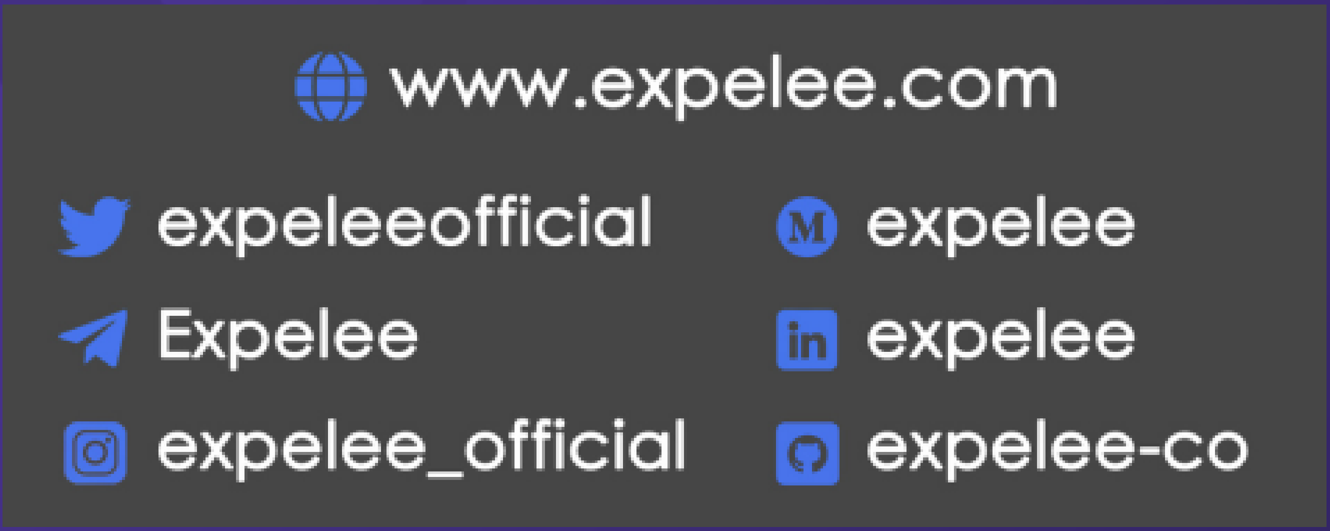
Mitigation

• Prior to marking an address as tax-free, validate the address using the deps.api.addr_validate function to ensure it adheres to the Terra blockchain's address format.

• Implement an additional level of verification before allowing the owner to mark an address as tax-free. For instance, require the owner to confirm the new address through a multi-step process or through a secondary authorization method.

ABOUT EXPELEE

Expelee is a product-based aspirational Web3 start-up. Coping up with numerous solutions for blockchain security and constructing a Web3 ecosystem from deal making platform to developer hosting open platform, while also developing our own commercial and sustainable blockchain.



www.expelee.com

expeleeofficial

Expelee

expelee_official

expelee

expelee

expelee-co

expelee

Building the Futuristic Blockchain Ecosystem

DISCLAIMER

This audit report is prepared for informational purposes only and does not constitute financial, legal, or professional advice. The audit has been conducted based on the information provided by the project team and is limited to the code and documentation available up to the audit date. The assessment is focused on identifying potential vulnerabilities and security concerns within the provided codebase. It does not guarantee the absence of vulnerabilities or the security of the system. The audit does not include a comprehensive review of the project's business model, economic viability, or other non-technical aspects.

The audit report is not an endorsement of the project, and readers should exercise their own judgment and due diligence before using, investing, or participating in the project. The project team and the auditors are not liable for any losses, damages, or expenses that may arise from actions taken based on this audit report.

The audit is based on the current state of the code and may not account for future changes, updates, or modifications. Security is an ongoing process, and the project team is responsible for addressing any identified issues and maintaining the security of the system beyond the scope of this audit.

Readers are encouraged to consult with their own professional advisors for advice tailored to their individual circumstances before making any decisions related to the project.

This audit report is provided on an “as is” basis and without any warranties, representations, or guarantees.