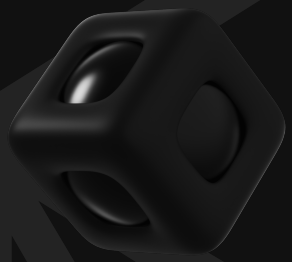
KEKKAI



WrappedElon

Smart Contract Security Audit

Audited by

KEKKAI

Supported by

BEOSIN
Blockchain Security

No.2023121102

Dec 11th, 2023

Contents

Summary Of Audit Results

After auditing, 1 Info item was identified in the WrappedElon project. Specific audit details will be presented in the Findings section. Users should pay attention to the following aspects when interacting with this project:

Info

Fixed:0 Acknowledged:1

Risk Description:

1. Users should note that wrapping is only possible in increments of 0.0001 ELON tokens. Tokens with precision smaller than this value won't be able to be wrapped and will remain in the original address.

Project Description:

1. Basic Token Information

Token name	Dogelon Mars
Token symbol	ELON
Decimals	4
Token type	ERC20

2. Business overview

The WrappedElon contract is a token contract that allows users to wrap and unwrap Elon tokens. The contract allows users, when the wrap module is enabled, to deposit a certain amount of ELON tokens and receive a corresponding amount of Dogelon Mars tokens. Similarly, when the unwrap module is enabled, users can burn a certain amount of Dogelon Mars tokens to withdraw a corresponding amount of ELON tokens. The decimals of Dogelon Mars tokens are set to 4, suitable for bridging assets. The owner can use the `setEnabledState` function to enable and disable the wrapping and unwrapping modules for the token.

1. Overview

1.1 Project Overview

Project Name	WrappedElon
Project language	Solidity
Platform	Ethereum
File hash	1E01DFE4AA7235D770A418DA06EA9EE17212335DE2BBD7DC3CCE287AE450915E

1.2 Audit Overview

Audit work duration: Dec 11, 2023 – Dec 11, 2023

Audit team: KEKKAI Security Team, Beosin Security Team

1.3 Audit Method

-Formal Verification

Formal verification is a technique that uses property-based approaches for testing and verification. Property specifications define a set of rules using library of security expert rules. These rules call into the contracts under analysis and make various assertions about their behavior. The rules of the specification play a crucial role in the analysis. If the rule is violated, a concrete test case is provided to demonstrate the violation.

-Manual Review

Using manual auditing methods, the code is read line by line to identify potential security issues. This ensures that the contract's execution logic aligns with the client's specifications and intentions, thereby safeguarding the accuracy of the contract's business logic.

The manual audit is divided into three groups to cover the entire auditing process:
The Basic Testing Group is primarily responsible for interpreting the project's code and conducting comprehensive functional testing.

The Simulated Attack Group is responsible for analyzing the audited project based on the collected historical audit vulnerability database and security incident attack models. They identify potential attack vectors and collaborate with the Basic Testing Group to conduct simulated attack tests.

The Expert Analysis Group is responsible for analyzing the overall project design, interactions with third parties, and security risks in the on-chain operational environment. They also conduct a review of the entire audit findings.

1. Findings

Index	Risk description	Security level	Status
WrappedElon-01	Missing event trigger	Info	Acknowledged

Finding Details

[WrappedElon-01]Missing event trigger

Severity Level	Info
Type	General Vulnerability
Lines	Wrapped.sol #L75-78
Description	In the WrappedElon contract, the Owner utilizes the setEnabledState function to configure critical parameters of the contract. However, the setEnabledState function does not trigger an event within its implementation. This is not considered a good practice and can hinder the ability to obtain contract information.

```
function setEnabledState(bool _wrapEnabled, bool _unwrapEnabled)
public onlyOwner {
        wrapEnabled = _wrapEnabled;
        unwrapEnabled = _unwrapEnabled;
         }
```

Recommendation	It is recommended to emit events when modifying critical variables is a recommended practice as it provides a standardized way to capture and communicate important changes within the contract. Events enable transparency and allow external systems and users to easily track and react to these modifications.
Status	**Acknowledged**

3. Appendix

3.1 Vulnerability Assessment Metrics and Status in Smart Contracts

3.1.1 Metrics

In order to objectively assess the severity level of vulnerabilities in blockchain systems, this report provides detailed assessment metrics for security vulnerabilities in smart contracts with reference to CVSS 3.1 (Common Vulnerability Scoring System Ver 3.1).

According to the severity level of vulnerability, the vulnerabilities are classified into four levels: "critical", "high", "medium" and "low". It mainly relies on the degree of impact and likelihood of exploitation of the vulnerability, supplemented by other comprehensive factors to determine of the severity level.

Likelihood \ Impact	Severe	High	Medium	Low
Probable	Critical	High	Medium	Low
Possible	High	Medium	Medium	Low
Unlikely	Medium	Medium	Low	Info
Rare	Low	Low	Info	Info

3.1.2 Degree of impact

Severe

Severe impact generally refers to the vulnerability can have a serious impact on the confidentiality, integrity, availability of smart contracts or their economic model, which can cause substantial economic losses to the contract business system, large-scale data disruption, loss of authority management, failure of key functions, loss of credibility, or indirectly affect the operation of other smart contracts associated with it and cause substantial losses, as well as other severe and mostly irreversible harm.

High

High impact generally refers to the vulnerability can have a relatively serious impact on the confidentiality, integrity, availability of the smart contract or its economic model, which can cause a greater economic loss, local functional unavailability, loss of credibility and other impact to the contract business system.

Medium

Medium impact generally refers to the vulnerability can have a relatively minor impact on the confidentiality, integrity, availability of the smart contract or its economic model, which can cause a small amount of economic loss to the contract business system, individual business unavailability and other impact.

Low

Low impact generally refers to the vulnerability can have a minor impact on the smart contract, which can pose certain security threat to the contract business system and needs to be improved.

3.1.3 Likelihood of Exploitation

Probable

Probable likelihood generally means that the cost required to exploit the vulnerability is low, with no special exploitation threshold, and the vulnerability can be triggered consistently.

Possible

Possible likelihood generally means that exploiting such vulnerability requires a certain cost, or there are certain conditions for exploitation, and the vulnerability is not easily and consistently triggered.

Unlikely

Unlikely likelihood generally means that the vulnerability requires a high cost, or the exploitation conditions are very demanding and the vulnerability is highly difficult to trigger.

Rare

Rare likelihood generally means that the vulnerability requires an extremely high cost or the conditions for exploitation are extremely difficult to achieve.

3.1.4 Fix Results Status

Status	Description
Fixed	The project party fully fixes a vulnerability.
Partially Fixed	The project party did not fully fix the issue, but only mitigated the issue.
Acknowledged	The project party confirms and chooses to ignore the issue.

3.2 Audit Categories

- **Coding Conventions**

Audit whether smart contracts follow recommended language security coding practices. For example, smart contracts developed in Solidity language should fix the compiler version and do not use deprecated keywords.

- **General Vulnerability**

General Vulnerability include some common vulnerabilities that may appear in smart contract projects. These vulnerabilities are mainly related to the characteristics of the smart contract itself, such as integer overflow/underflow and denial of service attacks.

- **Business Security**

Business security is mainly related to some issues related to the business realized by each project, and has a relatively strong pertinence. For example, whether the lock-up plan in the code match the white paper, or the flash loan attack caused by the incorrect setting of the price acquisition oracle.

*Note that the project may suffer stake losses due to the integrated third-party protocol. This is not something KEKKAI can control. Business security requires the participation of the project party. The project party and users need to stay vigilant at all times.

3.3 Disclaimer

The Audit Report issued by KEKKAI is related to the services agreed in the relevant service agreement. The Project Party or the Served Party (hereinafter referred to as the "Served Party") can only be used within the conditions and scope agreed in the service agreement. Other third parties shall not transmit, disclose, quote, rely on or tamper with the Audit Report issued for any purpose.

The Audit Report issued by KEKKAI is made solely for the code, and any description, expression or wording contained therein shall not be interpreted as affirmation or confirmation of the project, nor shall any warranty or guarantee be given as to the absolute flawlessness of the code analyzed, the code team, the business model or legal compliance.

The Audit Report issued by KEKKAI is only based on the code provided by the Served Party and the technology currently available to KEKKAI. However, due to the technical limitations of any organization, and in the event that the code provided by the Served Party is missing information, tampered with, deleted, hidden or subsequently altered, the audit report may still fail to fully enumerate all the risks.

The Audit Report issued by KEKKAI in no way provides investment advice on any project, nor should it be utilized as investment suggestions of any type. This report represents an extensive evaluation process designed to help our customers improve code quality while mitigating the high risks in blockchain.a

3.4 About KEKKAI

KEKKAI provides a web3.0 anti-fraud security solution for the consumer side based in Japan. It now offers a product range that includes an anti-fraud browser extension and a mobile application, and solution for web3.0 security such as smart contract security audit and penetration test. The aim of KEKKAI is to build the security layer of Web3 for consumers. It provides not only a firewall for daily crypto trading but also an environment where users can browse the Web3 world with peace of mind. Since its launch in February 2023, KEKKAI has gained over 40,000 users all over the world using KEKKAI's product. It is now protecting more than $200M of user asset from not being attacked, and many of projects for security auditing.

3.5 About Beosin

Beosin is the first institution in the world specializing in the construction of blockchain security ecosystem. The core team members are all professors, postdocs, PhDs, and Internet elites from world-renowned academic institutions. Beosin has more than 20 years of research in formal verification technology, trusted computing, mobile security and kernel security, with overseas experience in studying and collaborating in project research at well-known universities. Through the security audit and defense deployment of more than 2,000 smart contracts, over 50 public blockchains and wallets, and nearly 100 exchanges worldwide, Beosin has accumulated rich experience in security attack and defense of the blockchain field, and has developed several security products specifically for blockchain.

This time, Beosin is participating as supporter regarding as doing double check of the auditing result.

KEKKAI

0xKEKKAI

https://kekkai.io