



BEOSIN
Blockchain Security

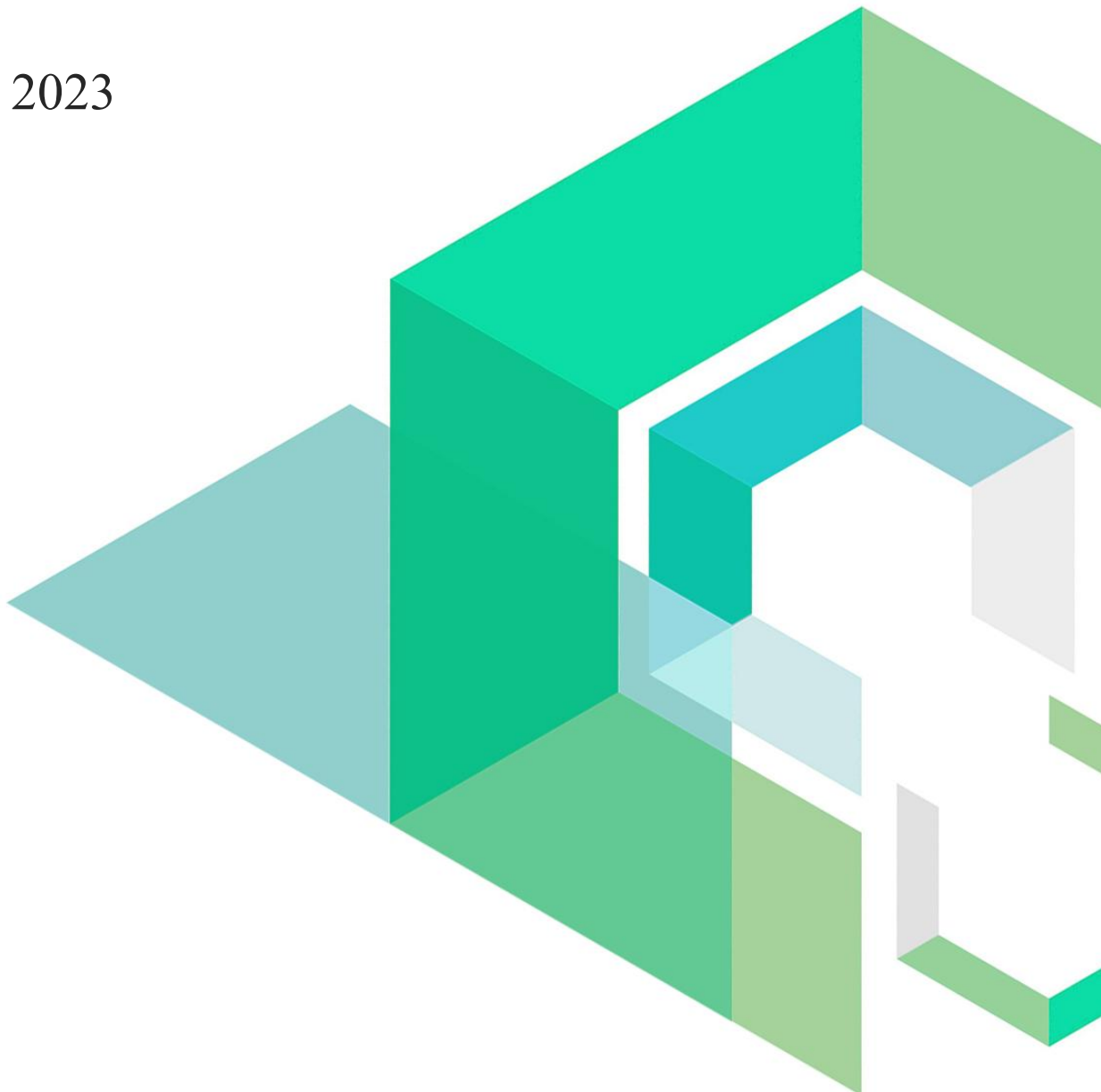
ARBI

Smart Contract Security Audit

V1.0

No. 202304171422

Apr 17th, 2023

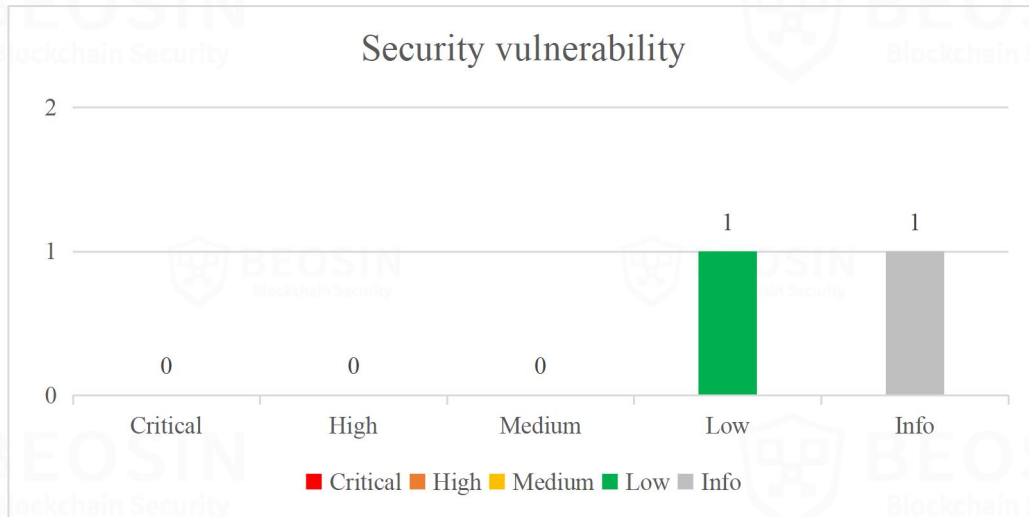


Contents

Summary of Audit Results	1
1 Overview	3
1.1 Project Overview	3
1.2 Audit Overview	3
2 Findings	4
[ARBI - 1] Centralization risk	5
[ARBI - 2] Missing event trigger	6
3 Appendix	7
3.1 Vulnerability Assessment Metrics and Status in Smart Contracts	7
3.2 Audit Categories	9
3.3 Disclaimer	11
3.4 About Beosin	12

Summary of Audit Results

After auditing, 1 Low risk and 1 Info items were identified in the ARBI project. Specific audit details will be presented in the Findings section. Users should pay attention to the following aspects when interacting with this project:



***Notes:**

- **Risk Description:**

Users should concern conditions as below:

1. Users will not get any rewards for staking assets in the contract, and the official reply will be rewarded through other businesses.
2. Users cannot use emergencyWithdraw to stake assets in the contract. If you want to get back the assets, you need to call the *Unstake* function to unlock them. If called repeatedly, the recorded events will be overwritten and the lock time will be recalculated.
3. Users should pay attention to the update of the `_lockupDuration` variable after staking assets, which will affect the lockup time, and there is no limit to the setting range of the `lockupDuration` variable, and the official reply will set it to 14 days.

Project Description:

1. Basic Token Information

Token name	Arbipad
Token symbol	ARBI
Decimals	18
Pre-mint	10 billion
Total supply	10 billion
Token type	ERC20

Table 1 Basic information of ARBI

2. Business overview

This audit of the ARBI project includes two parts, namely the stake project and the token project on the Arbitrum chain. In the stake contract, users can use designated tokens for staking, and after getting the stake record in the contract, they can get rewards set in other businesses, and the stake funds can only be retrieved after the lock-up time ends. In the token project, all tokens will be transferred to the address set during deployment, and 10 billion ARBI tokens will be minted during deployment, and the owner will also be transferred to the address set during deployment. After deployment, ARBI token cannot be in mint, but only burn ARBI tokens.

1 Overview

1.1 Project Overview

Project Name	ARBI
Platform	Arbitrum
Contract address	https://arbiscan.io/token/0x07DD5BEAffb65B8fF2e575d500BDf324a05295DC
File Hash(SHA256)	Staking contract: 357f936733fa0b359438fe3961908027a8f1456a3ae29a233a241cdd61332b9a a021f0987a4bffc375d72822f65b4a46239ac62245d87f1448668cde9b35585 b3bdb4753f6017e1c73d892732321a16319bacd851111ba0daa370e6ce1c2020 a021f0987a4bffc375d72822f65b4a46239ac62245d87f1448668cde9b35585

1.2 Audit Overview

Audit work duration: Apr 14, 2023 –Apr 17, 2023

Audit methods: Formal Verification, Static Analysis, Typical Case Testing and Manual Review.

Audit team: Beosin Security Team.

2 Findings

Index	Risk description	Severity level	Status
ARBI - 1	Centralization risk	Low	Acknowledged
ARBI - 2	Missing event trigger	Info	Fixed

Status Notes:

1. ARBI -1 is not fixed and if the owner's private key is lost, it may cause a large amount of token loss.

Finding Details:

[ARBI - 1] Centralization risk

Severity Level	Low
Type	Business Security
Lines	ArbipadToken.sol #L9-14
Description	All tokens will be minted to the wallet address in the <i>constructor</i> function, and transfer to a non-multi-signature wallet address in the deployed contract, which may have certain centralization risks.

```

9  contract ArbipadToken is Ownable, ERC20Burnable {
10     constructor(address wallet, uint256 totalSupply) Ownable() ERC20("Arbipad","ARBI") {
11         _mint(wallet, totalSupply);
12         transferOwnership(wallet);
13     }
14 }

```

Figure 1 Source code of *constructor* function

Recommendations	It is recommended to use multi-signature wallet to manage ARBI tokens.
Status	Acknowledged. The official reply will be managed using a multi-signature wallet.

[ARBI - 2] Missing event trigger

Severity Level	Info
Type	Business Security
Lines	Staking.sol#L64-66
Description	Important variable changes, missing event.

```

64     function updateStakingInfo(uint256 _minAmount, uint256 _lockupDuration) external onlyOwner {
65         stakeInfo = Stake(_minAmount, _lockupDuration);
66     }

```

Figure 2 Source code of *updateStakingInfo* function(unfixed)

Recommendations	It is recommended to add an event trigger or limit the setting range of the <i>_lockupDuration</i> variable.
------------------------	--

Status	Fixed.
---------------	--------

```

69     function updateStakingInfo(uint256 _minAmount, uint256 _lockupDuration) external onlyOwner {
70         stakeInfo = Stake(_minAmount, _lockupDuration);
71         emit UpdateStakeInfo(block.timestamp, msg.sender, _minAmount, _lockupDuration);
72     }

```

Figure 3 Source code of *updateStakingInfo* function(fixed)

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.

Impact Likelihood	Severe	High	Medium	Low
Probable	Critical	High	Medium	Low
Possible	High	High	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.

3.1.4 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.

3.1.5 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

No.	Categories	Subitems
1	Coding Conventions	Compiler Version Security
		Deprecated Items
		Redundant Code
		require/assert Usage
		Gas Consumption
2	General Vulnerability	Integer Overflow/Underflow
		Reentrancy
		Pseudo-random Number Generator (PRNG)
		Transaction-Ordering Dependence
		DoS (Denial of Service)
		Function Call Permissions
		call/delegatecall Security
		Returned Value Security
		tx.origin Usage
		Replay Attack
		Overriding Variables
Third-party Protocol Interface Consistency		
3	Business Security	Business Logics
		Business Implementations
		Manipulable Token Price
		Centralized Asset Control
		Asset Tradability
		Arbitrage Attack

Beosin classified the security issues of smart contracts into three categories: Coding Conventions, General Vulnerability, Business Security. Their specific definitions are as follows:

- **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 Beosin 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 Beosin 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 Beosin 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 Beosin is only based on the code provided by the Served Party and the technology currently available to Beosin. 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 Beosin 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.

3.4 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.



Official Website

<https://www.beosin.com>

Telegram

<https://t.me/+dD8Bnqd133RmNWNl>

Twitter

https://twitter.com/Beosin_com

Email

Contact@beosin.com

