3rd MARCH 2021

SMART CONTRACT AUDIT REPORT

-

version v2.0

Smart Contract Security Audit and General Analysis

HAECHI AUDIT

Table of Contents

1 Issues (1 Critical, 0 Major, 0 Minor) Found

About HAECHI AUDIT

HAECHI AUDIT is a global leading smart contract security audit and development firm operated by HAECHI LABS. HAECHI AUDIT consists of professionals with years of experience in blockchain R&D and provides the most reliable smart contract security audit and development services.

So far, based on the HAECHI AUDIT's security audit report, our clients have been successfully listed on the global cryptocurrency exchanges such as Huobi, Upbit, OKEX, and others.

Our notable portfolios include SK Telecom, Ground X by Kakao, and Carry Protocol while HAECHI AUDIT has conducted security audits for the world's top projects and enterprises.

Trusted by the industry leaders, we have been incubated by Samsung Electronics and awarded the Ethereum Foundation Grants and Ethereum Community Fund.

Contact : audit@haechi.io
Website : audit.haechi.io

01. Introduction

This report was written to provide a security audit for the BSCBridge smart contract. HAECHI AUDIT conducted the audit focusing on whether BSCBridge smart contract is designed and implemented in accordance with publicly released information and whether it has any security vulnerabilities.

The issues found are classified as CRITICAL, MAJOR, MINOR or TIPS according to their severity.

CRITICAL	Critical issues are security vulnerabilities that MUST be addressed in order to prevent widespread and massive damage.
MAJOR	Major issues contain security vulnerabilities or have faulty implementation issues and need to be fixed.
MINOR	Minor issues are some potential risks that require some degree of modification.
TIPS	Tips could help improve the code's usability and efficiency

HAECHI AUDIT advises addressing all the issues found in this report.

02. Summary

The code used for the audit can be found at GitHub (https://github.com/itamgames/bsc-bridge-core). The last commit for the code audited is at "d7c58b8530fd7acbe74bd13a9c04bff4de97fa6a".

Issues

HAECHI AUDIT has 1 Critical Issue, 0 Major Issues, and 0 Minor Issues.

Severity	Issue	Status
CRITICAL	**BSCBridge#transit(), ETHBridge#withdraw() is vulnerable to replay attack.**	(Found v1.0) (Fixed v2.0)
TIPS	**Use ECDSA library instead of implementing**	(Found v1.0) (Fixed v2.0)
Notice	**Signer can withdraw from ETHBridge**	(Acknowledged)
Notice	**Owner or Signer can disable withdrawal request even signature is issued**	(Acknowledged)
Notice	**Owner can mint unlimited amount of ItamERC20 and burn user's balance**	(Acknowledged)

Update

[v2.0] - 1 critical issue and 1 tips has been resolved with new commit hash f7e951b35501ad8e7974795a55b80de454a35053

03. Overview

Contracts Subject to Audit

- BEP20.sol
- BSCBridge.sol
- ETHBridge.sol
- TransferHelper.sol

Roles

The BSCBridge Smart contract has the following authorizations:

- **Owner**
- **Signer**

The features accessible by each level of authorization is as follows:

Role	Functions

Owner	• BSCBridge
	  ○ changeTransitFee
	  ○ changeSigner
	  ○ withdrawFee
	  ○ transferTokenOwnership
	• ETHBridge
	  ○ changeSigner
	• ItamERC20
	  ○ mint
	  ○ burn

Signer	-

Notice

- **Signer can withdraw from ETHBridge**

  ETHBridge holds ERC20 assets and these assets can only be transferred by withdraw() function. And this withdraw() function can be called by Signer without any guards since the only requirement is if signature is signed by signer.

  This is a centralization issue and Itamgames has confirmed this is acknowledged.

- **Owner or Signer can disable withdrawal request even signature is issued**

  Even though the user has received a valid signature, this can be voided when another signature has been issued and submitted or Owner changes the Signer.

  This is a centralization issue and Itamgames has confirmed this is acknowledged.

- **Owner can mint unlimited amount of ItamERC20 and burn user's balance**

  ItamERC20 contract has mint() and burn() function and which allows the owner to mint unlimited amounts and burn the user's balance.

04. Issues Found

CRITICAL : BSCBridge#transit(), ETHBridge#withdraw() is vulnerable to replay attack. (Found - v1.0) (Resolved - v2.0)

CRITICAL

Problem Statement

BSCBridge#transit() and ETHBridge#withdraw() uses ecrecover method to verify that message has been signed by Signer. But this message has the same structure which enables replaying signatures between transit() and withdraw().

Recommendation

Include chain id and contract address(address(this)) to signing message

Update

[v2.0] - Itamgames has applied recommendation

TIPS : Use ECDSA library instead of implementing (Found - v1.0) (Resolved - v2.0)

TIPS

Problem Statement

BSCBridge#_recoverAddress(), ETHBridge#_recoverAddress() has been implemented as an address recover function which checks signature validity and returns signatory. Although this is a fairly good approach, using an existing library is a much safer way to use the ecdsa algorithm.

Recommendation

Use known libraries to prevent non-valid length signature and signature malleability.

Update

[v2.0] - Itamgames has applied recommendations by using openzeppelin cryptography library.

05. Disclaimer

This report is not an advice on investment, nor does it guarantee adequacy of a business model and/or a bug-free code. This report should be used only to discuss known technical problems. The code may include problems on Ethereum that are not included in this report. It will be necessary to resolve addressed issues and conduct thorough tests to ensure the safety of the smart contract.