MixBytes()

TIMELOANS.FINANCE

SMART CONTRACT

AUDIT REPORT

NOVEMBER 05

2020

TABLE OF CONTENTS

01 INTRODUCTION TO THE AUDIT

SCOPE OF THE AUDIT

The scope of the audit includes following smart contract at: from **TimeLoans.sol#L547** and **TimeLoans.sol#L848**.

The audited commit identifier is: cab5f5b742aba2a4c237c460e1b436c5e8b71ba4

02 SECURITY ASSESSMENT PRINCIPLES

CLASSIFICATION OF ISSUES

CRITICAL

Bugs leading to Ether or token theft, fund access locking or any other loss of Ether/tokens to be transferred to any party (for example, dividends).

MAJOR

Bugs that can trigger a contract failure. Further recovery is possible only by manual modification of the contract state or replacement.

WARNINGS

Bugs that can break the intended contract logic or expose it to DoS attacks.

COMMENTS

Other issues and recommendations reported to/acknowledged by the team.

SECURITY ASSESSMENT METHODOLOGY

Two auditors independently verified the code.

Stages of the audit were as follows:

1. “Blind” manual check of the code and its model
2. “Guided” manual code review
3. Checking the code compliance to customer requirements
4. Discussion of independent audit results
5. Report preparation

03 DETECTED ISSUES

CRITICAL

Not found.

MAJOR

Not found.

WARNINGS

1. No check that the address is not zero

TimeLoans.sol#L619

No need to burn a token for a zero address.

In a well-known library for secure development of smart contracts, a check is done: **ERC20.sol#L250**

Despite the fact that `_burn` is an internal function and is called now only from the `withdraw` function, it is necessary to check the address for zero. It is possible that this code will then be used again and the developer can call the `_burn` function in another place where ‘msg.sender’ is no longer used as an address.

TimeLoans.sol#L610

Need to check the address for zero.

Despite the fact that `_mint` is an internal function and is called now only from the `deposit` function, it is necessary to check the address for zero. It is possible that this code will then be used again and the developer can call the `_mint` function in another place where `msg.sender` is no longer used as an address.

Status:

ACKNOWLEDGED

COMMENTS

1. Code refactoring is recommended

TimeLoans.sol#L926

The (`spenderAllowance! = uint (-1)`) check can be removed because all operations to change the number of tokens are implemented using the safe mathematics library.

It is recommended to rewrite the `transferFrom` function in accordance with this: **ERC20.sol#L152**

Status:

ACKNOWLEDGED

04 CONCLUSION AND RESULTS

The smart contract was audited and no critical or major issues were found.

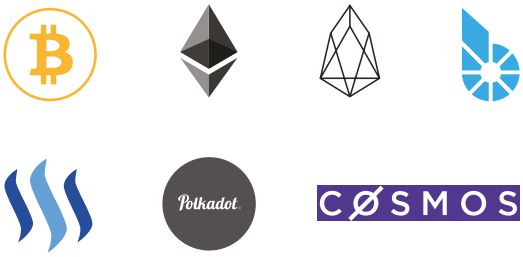
One suspicious location has been identified (marked as warning), but it's assumed as not critical.

ABOUT MIXBYTES

MixBytes is a team of blockchain developers, auditors and analysts keen on decentralized systems. We build open-source solutions, smart contracts and blockchain protocols, perform security audits, work on benchmarking and software testing solutions, consult universities and enterprises, do research, publish articles and documentation.

Stack

Blockchains



JOIN US