Ethbet Contract Audit

January, 2018

Alexander Wade

Contents

1. Overview

This document serves as the official audit report of a token betting and finalization contract, created and published by Ethbet. Ethbet has created a peer-to-peer dicing game, where the players are able to act as their own casinos, setting a “house edge” for placed bets, if they want. Players place bets, choosing whether to prioritize speed over expected return, as entering players are more likely to enter into bets where the “house” has a negative edge.

The contract audited is Ethbet’s bet settling contract, which is used to transfer tokens between players based on Ethbet’s reports on wins or losses. The contract features a central relay address, used by the Ethbet team to simulate the effects of a won or lost bet.

2. Introduction

2.1 Authenticity

The audited contract is a specific file in the Ethbet github repository: **https://github.com/Ethbet/ethbet/blob/master/alpha/contracts/Ethbet.sol**.

The version used for this audit is commit **f941b414b9f5d4eb2de6771ee6d051b988331926**.

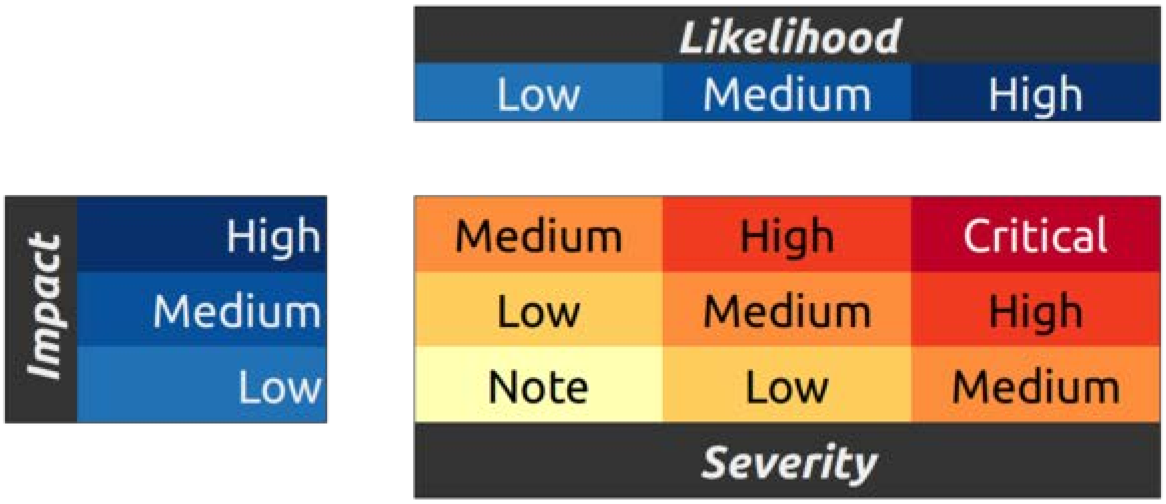
2.2 Scope

This audit covered only the solidity file linked to in the above section. This audit does not cover files located in any other folder.

2.3 Methodology

This audit focuses heavily on not only inspecting the smart contracts for vulnerabilities and potential for losses in funds, but also on working closely with the Ethbet team to scrutinize the contracts for execution of intent. The end goal of this audit is to help the team not only secure their contract, but also to ensure their vision for the project is best represented by the project they put forward. As a result, additional concerns such as efficiency and design are included in this report as well.

2.4 Terminology



		Likelihood		
		Low	Medium	High
Impact	High	Medium	High	Critical
	Medium	Low	Medium	High
	Low	Note	Low	Medium
		Severity		

This audit categorizes vulnerabilities using the OWASP risk rating method based on impact and likelihood. Each vulnerability is subjectively given impact and vulnerability scores, which are used to give a more accurate estimation of a vulnerability's overall severity. An additional factor in severity is the relative ease with which a vulnerability is fixed: an issue which requires extreme refactoring will be weighted higher than one with the same severity which is a quick fix.

2.5 Disclaimer

This document reflects the understanding of security flaws and vulnerabilities as they are known to the Authio team, and as they relate to the reviewed project. This document makes no statements on the viability of the project, or the safety of its contracts. This audit is not intended to represent investment advice.

3. Findings

3.1 General

The Ethbet team utilizes a centralized server to relay offers and bet confirmations, for efficiency and speed. When a bet is entered into by two users, the relay executes it and then calls the smart contract to settle the bet. The relay uses provably-fair random number generation to determine the outcome of the bet, and then calls the smart contract to reflect the outcome of the bet.

3.2 Contract Explanation

The contract audited is `Main.sol`:
This contract allows users to deposit and withdraw EBET tokens for use in betting. Additionally, a privileged relay address handles locking and unlocking balances when bets are placed and removed, as well as finalizing bets and transfer of tokens upon completion of the bet.

3.3 Critical

No critical severity issues were found.

3.4 High Severity

No high severity issues were found.

3.5 Medium Severity

No medium severity issues were found.

3.6 Low Severity

Define and implement a changeRelay function:
Currently, a new contract will need to be deployed if the relay address needs to be changed for any reason.

Users may intuitively send EBET tokens to the Ethbet contract:
Through the UI, the deposit function is automatically called, but users may intuitively transfer their EBET tokens directly to the contract, at which point the tokens would be locked permanently. A solution could allow the relay address to move tokens that are over the amount transferred via the deposit function.

3.7 Notes & Recommendations

Use the latest version of Solidity:
The latest version is 0.4.19.

Create, define, and import an interface for the Ethbet token:
Using an interface will allow for much lower cost when deploying, as space for public variables does not need to be allocated.

Use an updated version of SafeMath:

SafeMath is imported along with the token contract, which is using an older version of the library. Grab the latest library here: **https://github.com/OpenZeppelin/zeppelin-solidity/blob/master/contracts/math/SafeMath.sol**

Mark "deposit" as a public function:

This will silence the compiler warning, but is not explicitly needed for the function of the contract.

Users can withdraw "0" tokens:

There is no check for a '0' amount, which could allow a user to burn gas and not change state.

Relay address can lock and unlock "0" tokens:

There is no check made for a '0' amount, which would allow the relay to burn gas and not change state. If a user is able to force the relay address to lock or unlock '0' tokens repeatedly, the user could waste Ether in the relay address.

4. Documents & Resources

4.1 Line-By-Line Comments

Line-by-line commenting can be found in the EthereumAuthio github repository:

https://github.com/EthereumAuthio/Audits/tree/master/Ethbet

4.2 Project Code

The current Ethbet code can be found in their public github repository:

https://github.com/Ethbet/ethbet

5. Conclusion

The Authio team would like to recommend that the Ethbet team now continues with this auditing and security process by posting public bug bounties and asking for community input. More eyes are always better, and after a rigorous audit, a bug bounty can serve as a valuable sanity check for all parties involved.

The Authio team would like to congratulate Ethbet on a well-written project. The codebase was clean and well-organized. Racing conditions were the main potential issue with this contract, and Ethbet has handled the problem gracefully, creating a robust and secure EBET deposit, withdraw, and bet settling contract.