Global Listing Exchange

Code Review - GLX Dollar(X)

Document Revision History

Version	Date	Author	Change Description
0.1 Draft	29 May 2017	Ajay Singh	Interim - code review

Table of Contents

1 FUNCTIONALITY REVIEW

- **The code only creates token, no provision for issuance of the token.**

  GLX Dollar(X) requires to issue the token to the users, but the current code base do not have any provision to issue the token. It only creates the token but do not issue or disburse it to the users.

- **Created tokens are not tradeable.**

  The code has not implemented the standard ERC20 transfer function which makes it non-tradeable on Ethereum exchanges.

  The transfer function written has additional flag supplied in the parameter which makes it non-standard ERC20 transfer function.

2 TECHINCAL REVIEW

- **Absence of separation of technical and business concern.**

  No architecture has been followed to separate the business logic from data structure. Presently business logic and data structure are tightly coupled which makes it difficult to change business logic without impacting data structure and vice-versa. Please refer Appendix A.

- **Short Address Attack Not Handled.**

  The written Solidity contract has a bug which make it prone to Short Address Attack. It is assumed that the User will input 20-byte long address, but the length of the addresses is not actually checked.

  Allowing a user to input a shorter transfer address shifts the "amount of tokens to transfer" value to the left, making the value larger. It is also very easy to find a private key to an Ethereum address with zeros in the end of the address, e.g. 0xaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa0000.
  Therefore, the owner of this address can enter 0xaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa (skipping zeros) in the service interface. The attacker could then order a transfer of some value X from the service, to the inserted malformed address.
  This would actually cause a transfer of a value shifted by 16 bits, i.e. 65536 times larger than X, to attacker's Ethereum account!

- **Zero Address Check.**

  Need to check the target address is not 0x0.
  Absence of this check will lead to burn of the GLX Dollar(x). Any coin which by mistake gets transferred to 0x0 address will result to the burn of the transferred coin. This will lead to the loss of coin in circulation.

- **Absence of event generation in case class B converts to Class A.**

  Whenever GLX Dollar is transferred then full audit trails need to be maintained for external system. Events need to be logged in case of GLX Dollar(X) Class B is converted to GLX Dollar(X) Class A token.

- **Code Maintenance.**

  Code is written in a single sol (solidity) file. When the code base will increase, it will become difficult to maintain and optimize.

- **Code Re-Usability.**

  The code should be written in the form of a library which can be re-used in future implementation of different GLX platform, e.g. the safemath implementation could be maintained in the form of library which can be imported in all future mathematical implementations.