Audit Report of Smart contracts for Equadex ICO

Prepared for

Equadex

Prepared By

Praveen Dagdi

F-103, Amrapali Plaza, Amrapali Circle, Vaishali Nagar
Jaipur – 302021, RAJASTHAN,
INDIA

Delivered on

April 20th, 2018

Disclaimer

The audit report is not a legally binding document. The audit makes no statements or warrantees about utility of the code, safety of the code, suitability of the business model, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bug free status. The audit documentation is for discussion purposes only.

Overview

The smart contracts for EDEX ICO are very well-written concerning security, vulnerabilities, targeted and proper incentive distribution, and time-line based execution.

We have thoroughly inspected the code for any vulnerabilities it might contain; reverse engineering it on the Testnet produced positive results in regards to the health and security of the contract. Both contracts are congruent with 15% of tokens being transferred to the grantVestedEDEX address following the investor's purchase of the EDEX token.

Though we found some anomalies with respect to the entire Ethereum ecosystem that are not only limited to the technical aspects of the contracts. Our findings do not require any extensive changes to the code; however, there are some suggestions which should not be overlooked and must be taken into consideration before going live.

Positive Features

1. Use of "SafeMath" contract is commendable. For example, 0 – (any positive number) would result in 2**256 instead of positive number with minus sign.
2. Short addresses are vulnerable to transferring very large amounts versus a nominal amount requested by simply adding zeros to the transaction. This is commendable because Equadex is flagging them in the contract.
3. Requiring (parameter != address(0)) check flag in various functions is a great way to avoid creating a new contract at a different address beyond an existing contract.
4. Use of "constant" keyword is also commendable as it uses *.call()* instead of *sendtranscation()* of WEB3 hence no gas is spent.

Critical Severity

There are no critical issues in solidity files.

High Severity

Use of block.number instead of block.timestamp presents a serious issue. The contract assumes the IcoEndBlock number is based on the average time of the validation of a block and also on average of block addition traffic to the Ethereum network on a daily basis. Equadex intends to run the ICO for 60 days and thus assumes 345,600 blocks until the deadline. We categorize this issue as a high severity because the Ethereum network intends to shift from a proof of work mining protocol to a proof of stake mining protocol. Therefore, if this occurs while the ICO is open to the public, then average validating time periods will be drastically reduced and, as a consequence, the 345,600 block is at risk of being mined far in advance of the Equadex ICO's 60 day time period.

NOTE: Please keep track of Ethereum network activities during the coming days.

Medium Severity

We mentioned some technical glitches in the initial audit report and those glitches are based on solidity version 0.4.21 but since Equadex is using version 0.4.16, most of those issues can be overlooked, if not all of them. In theEDEX.sol file within some of the functions, a constant keyword is used and those functions are prone to changes in the state of the variables. These are the vulnerable functions –

1. **icoBottomIntegerPrice()** (on line 309) – On line 310 safeSub function, on line 318 , and 322 safeMul functions are potentially changing the value of the state variable currentPrice.bottomInteger.
2. **checkLiquidationValue()** (on line 385) – On line 388 safeSub function is changing the value of the state variable.

Low Severity

In file grantVestedEDEX.sol state variables – firstTeamWithdrawal, secondTeamWithdrawal, thirdTeamWithdrawal, and later teamwithdrawal functions can be declared as private state variables since they are unlikely to be relevant to external use.

Conclusion

In our thorough analysis and testing of all the functions mentioned in the contracts, we found them to be working as intended with the proper security and checks.

Contact Us

If you have any questions or concerns regarding the report or our services, please contact us through the following details:

Skype ID – live:techkopra
Email – techkopra@gmail.com
Phone: +91-977-2989-235 / +91-141-4108378