BlockchainLabsNZ / BitHeroToken Public

Code Issues 9 Pull requests Actions Projects Wiki Security Insights

audit

BitHeroToken / audit / readme.md

112 lines (81 sloc) | 7.32 KB

BitHeroToken Smart Contracts Audit Report

Prepared by:

- Matt Lough, matt@blockchainlabs.nz
- Bruce Li, bruce@blockchainlabs.nz

Preamble

This audit report was undertaken for BitHero, by their request, and has subsequently been shared publicly without any express or implied warranty.

Solidity contracts were sourced directly from the BitHero team and the most recent commit we have audited is this commit www.github.com/BlockchainLabsNZ/BitHeroToken

We would encourage all community members and token holders to make their own assessment of the contracts.

Scope

The following contracts were subject for analysis:

- `contracts/`
  - `AllocatedCrowdsale.sol`

- `AllocatedCrowdsaleMixin.sol`
- `Crowdsale.sol`
- `CrowdsaleBase.sol`
- `CrowdsaleToken.sol`
- `DefaultFinalizeAgent.sol`
- `FinalizeAgent.sol`
- `FlatPricing.sol`
- `FractionalERC20.sol`
- `GnosisWallet.sol`
- `Haltable.sol`
- `Migrations.sol`
- `MintableToken.sol`
- `PricingStrategy.sol`
- `Recoverable.sol`
- `ReleasableToken.sol`
- `SafeMathLib.sol`
- `StandardTokenExt.sol`
- `UpgradeAgent.sol`
- `UpgradeableToken.sol`

Framework Analysis

Full report can be read here

Functional Testing

Full report can be read here

Issues

Severity Description

Minor	A defect that does not have a material impact on the contract execution and is likely to be subjective.
Moderate	A defect that could impact the desired outcome of the contract execution in a specific scenario.
Major	A defect that impacts the desired outcome of the contract execution or introduces a weakness that may be exploited.

Critical	A defect that presents a significant security vulnerability or failure of the contract across a range of scenarios.

Minor

- Use .transfer instead of .send - `Best practice` This is a very minor issue because `.send` is still valid, but `.transfer` has a richer interface and allows you to override the gas limit, which `.send` does not. There is some discussion on `.send` vs `.transfer` here: ethereum/solidity#610 #L208 #L361 ... View on GitHub
- Avoid magic number. - `Best practice` We recommend avoiding the use of magic numbers, using a variable here would improve readability and make the code more maintainable for the future. #L41 This could be better written as `0.1 ether` View on GitHub
- Make visibility explicitly declared on functions - `Best practice` Examples below, no visibility found: #L37 #L148 #L159 View on GitHub
- Use explicit types for variables - `Best practice` It is best practice to explicitly define your types. For example you should specify `uint256` instead of `uint` #L44 View on GitHub
- Using require is recommended over throw - `Best practice` #L70 It is best practice to use `require(somethingHappened)` instead of `if (!somethingHappened) throw` View on GitHub
- Typo in function name setEarlyParicipantWhitelist - `Best practice #L410 There is a TODO which has not been completed, there is still a typo in this function name View on Github

Moderate

- Confusing logic in isBreakingCap function - `Correctness` The name of this function makes it sound like you are checking to see if you are trying to buy too many tokens, but there is also a check to see if the amount is above a minimum #L41 Is this intentional or a mistake? View on GitHub
- CrowdsaleBase has no Solidity version specified - `Correctness` You should always specify the version of Solidity that should be used for compiling View on GitHub
- Old versions of Solidity used - `Best practice` It is recommended to use a consistent version of Solidity for each contract, and to use the latest stable version. Version `0.4.6`, `0.4.8`, `0.4.12` are being used View on GitHub

Major

- None found

Critical

- None found

Observations

- Before the Audit the team has deployed their contracts to mainnet at addresses 0x91e8782aed0213659caea7d80975ac20ce9ebb38 & 0x5ba422338f85d19d92eaab161ce0ee2d93165116. To date neither of these contracts have been source verified. We would recommend that the team source verified these contracts to confirm that they are the same as the code that has been audited.
- The MultiSigWallet contract is using a different name, GnosisWallet.sol, as the file name. It doesn't break the contract's behaviour or connections. However, we don't recommend this. Naming the file by contract name is better for readbility.

Conclusion

tbd

Disclaimer

Our team uses our current understanding of the best practises for Solidity and Smart Contracts. Development in Solidity and for Blockchain is an emerging area of software engineering which still has a lot of room to grow, hence our current understanding of best practise may not find all of the issues in this code and design.

We have not analysed any of the assembly code generated by the Solidity compiler. We have not verified the deployment process and configurations of the contracts. We have only analysed the code outlined in the scope. We have not verified any of the claims made by any of the organisations behind this code.

Security audits do not warrant bug-free code. We encourge all users interacting with smart contract code to continue to analyse and inform themselves of any risks before interacting with any smart contracts.