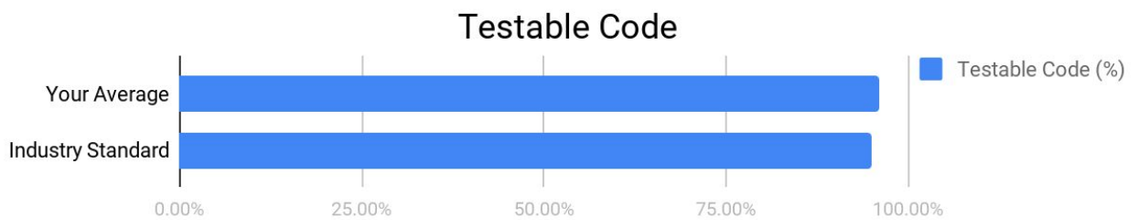
Bob’s Repair Contract Audit

by Hosho, April 2018

Executive Summary

This document outlines the overall security of Bob's Repair smart contract as evaluated by Hosho's Smart Contract auditing team. The scope of this audit was to analyze and document Bob's Repair token contract codebase for quality, security, and correctness.

All issues have been remediated. (See Complete Analysis)



Testable code is on par with industry standard. (See Coverage Report)

It should be noted that this audit is not an endorsement of the reliability or effectiveness of the contract, merely an assessment of its logic and implementation. In order to ensure a secure contract that's able to withstand the Ethereum network's fast-paced and rapidly changing environment, we at Hosho recommend that the Bob's Repair Team put in place a bug bounty program to encourage further and active analysis of the smart contract.

Table Of Contents

1. Auditing Strategy and Techniques Applied

The Hosho Team has performed a thorough review of the smart contract code, the latest version as written and updated on March 30, 2018. All main contract files were reviewed using the following tools and processes. (See All Files Covered)

Throughout the review process, care was taken to ensure that the token contract:

- Implements and adheres to existing ERC-20 Token standard appropriately and effectively;
- Documentation and code comments match logic and behavior;
- Distributes tokens in a manner that matches calculations;
- Follows best practices in efficient use of gas, without unnecessary waste; and
- Uses methods safe from reentrance attacks.
- Is not affected by the latest vulnerabilities

The Hosho Team has followed best practices and industry-standard techniques to verify the implementation of Bob's Repair's token contract. To do so, reviewed line-by-line by our team of expert pentesters and smart contract developers, documenting any issues as they were discovered. Part of this work included writing a unit test suite using the Truffle testing framework. In summary, our strategies consist largely of manual collaboration between multiple team members at each stage of the review:

1. Due diligence in assessing the overall code quality of the codebase.
2. Cross-comparison with other, similar smart contracts by industry leaders.
3. Testing contract logic against common and uncommon attack vectors.
4. Thorough, manual review of the codebase, line-by-line.
5. Deploying the smart contract to testnet and production networks using multiple client implementations to run live tests.

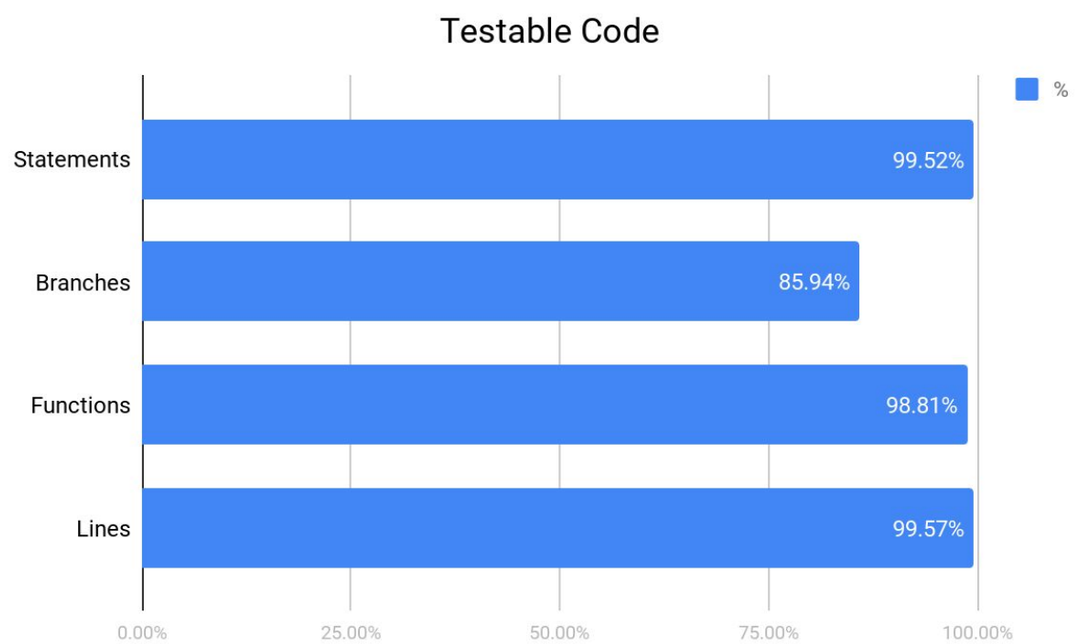
2. Structure Analysis and Test Results

2.1. Summary

The Bob's Repair contracts implement an ERC-827 token and multiple additions that extend the functionality of the token such as burnable and minting capabilities. There are also full systems implemented to handle vesting, airdrops and a manual crowdsale. Finally, there is a contract that allows for the exchange of an already deployed token, BOBP, for the token in these contracts, BOB.

2.2 Coverage Report

As part of our work assisting Bob's Repair in verifying the correctness of their contract code, our team was responsible for writing a unit test suite using the Truffle testing framework.



For individual files see Additional Coverage Report

2.3 Failing Tests

No failing tests.

See Test Suite Results for all tests.

3. Complete Analysis

For ease of navigation, sections are arranged from most critical to least critical. Issues are tagged "Resolved" or "Unresolved" depending on whether they have been fixed or addressed. Furthermore, the severity of each issue is written as assessed by the risk of exploitation or other unexpected or otherwise unsafe behavior:

- **Informational** - The issue has no impact on the contract's ability to operate.
- **Low** - The issue has minimal impact on the contract's ability to operate.
- **Medium** - The issue affects the ability of the contract to operate in a way that doesn't significantly hinder its behavior.
- **High** - The issue affects the ability of the contract to compile or operate in a significant way.
- **Critical** - The issue affects the contract in such a way that funds may be lost, allocated incorrectly, or otherwise result in a significant loss.

3.1. Resolved, Low: Potential Overflow

AirdropToken.sol

Explanation

Depending on how many decimal places the token is, the multiplier percent used during an airdrop round, and account balances, there is the possibility of an overflow contained in the AirdropToken contract on line 72. While unlikely, this would cause the contract to lockup and prevent all holders from receiving their balances.

Resolution

Verification has been added to the multiplier to prevent this case.

3.2. Resolved, Low: Unused Parameter

BOBCrowdsale.sol

Explanation

On line 45 of this contract, there is an untitled and unused string parameter.

Resolution

The Bob's Repair Team properly implemented the string parameter.

3.3. Resolved, High: Unused Contract

Explanation

PausableToken.sol is not used anywhere within the code and can be removed.

Resolution

PausableToken.sol has been replaced with Pausable.sol by the Bob’s Repair Team.

4. Closing Statement

We are grateful to have been given the opportunity to work with the Bob's Repair Team.

As a small team of experts, having backgrounds in all aspects of blockchain, cryptography, and cybersecurity, we can say with confidence that the Bob's Repair contract is free of any critical issues.

The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them.

We at Hosho recommend that the Bob's Repair Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.

5. Test Suite Results

Contract: BOBCrowdsale

✓ Should require token cap to be greater than zero (128ms)

✓ Should require minting a token with no vesting to not have a vesting date

✓ Should allow owner to mint tokens with no vesting (99ms)

✓ Should allow owner to mint tokens with vesting (346ms)

✓ Should require minting vesting end time to be greater than vesting start time (41ms)

✓ Should require minted tokens to remain under token cap (339ms)

✓ Should require crowdsale to not be finalized for minting tokens (358ms)

✓ Should require beneficiaries count to match amounts count when distributing tokens

✓ Should allow owner to distribute tokens (144ms)

✓ Should require greater than zero distribution amount before attempting token minting (38ms)

Destructible

✓ Allow owner to destroy

✓ Allow owner to destroy and send

Contract: BOBToken

✓ Should allow owner to set founder

✓ Should allow transfer (132ms)

✓ Should allow transfer from (156ms)

✓ Should require transfer to be enabled for non-owner to transfer (50ms)

Pausble

✓ Should require unpaused to pause (61ms)

✓ Should require paused to be unpaused

HasNoContracts

✓ Should allow reclaiming contract (187ms)

HasNoEther

✓ Should reject ETH payment

✓ Should allow reclaiming ether

✓ Should allow reclaiming ether (secondary

CanReclaimToken

✓ Should allow owner to reclaim tokens (207ms)

HasNoTokens

✓ Should allow ERC23 `tokenFallback`

Contract: BOBPExchange

✓ Should allow owner to set fee compensation

✓ Should allow notifying BOBPExchange during BOBP transfer (269ms)

✓ Should allow notifying BOBPExchange during BOBP transferFrom (290ms)

✓ Should require BOB Promo token transfers to be enabled before notifying BOBPExchange (197ms)

✓ Should require token sale (BOBPExchange) to be set on BOB promo token to notify BOBPExchange (179ms)

✓ Should require minimum transfer amount to get extra compensation in exchange token transfer (253ms)

✓ Should require token transfer notify to only be callable from the BOB promo token

✓ Should require token transfer notify to only be callable from the BOB promo token (secondary check) (328ms)

✓ Should require BOB promo token balance to be enough for token transfer notify (208ms)

✓ Should require BOB exchange to have enough balance for compensation reward (271ms)

✓ Should require BOB token transfer to succeed for promo token transfer notify (182ms)

Contract: BurnableToken

✓ Should allow balance holders to burn tokens (51ms)

✓ Should require holder to have greater balance than amount to burn

Contract: PausableERC827Token

✓ Should allow approval with extra data

✓ Should allow approval increase with extra data (47ms)

✓ Should allow decrease approval with extra data (40ms)

✓ Should require valid spending address for approval with extra data

✓ Should require valid spending address for increase with extra data

✓ Should require valid spending address for decrease approval with extra data

Contract: AirdropToken

✓ Should allow owner to start drop (69ms)

✓ Should require drop to not have already been started (103ms)

✓ Should require drop percent to be greater than 100

✓ Should allow airdrop holders (100ms)

✓ Should require holder to not already be dropped for current airdrop (122ms)

✓ Should allow finish drop after all addresses are airdropped (183ms)

✓ Should require no more un-dropped balance to finish drop (155ms)

Contract: OpenZeppelin [MintableToken, StandardToken] tests for BOBToken

✓ Should deploy a token with the proper configuration (48ms)

✓ Should require minting to not be finished (47ms)

✓ Should allocate tokens per the minting function, and validate balances (168ms)

✓ Should transfer tokens from 0x0e6635349f4dac8c15f2f645e97af951f782d800 to 0xc64d33cd93bfbbc8cf717693eb4d1dd013cbba85 (68ms)

✓ Should not transfer negative token amounts

✓ Should not transfer more tokens than you have

✓ Should allow 0x28697ab59a4546c2888888a752e8437687d773f7 to authorize 0x6c6061667fde616d6ed450d7f296d9fe1e470854 to transfer 1000 tokens (52ms)

✓ Should allow 0x28697ab59a4546c2888888a752e8437687d773f7 to zero out the 0x6c6061667fde616d6ed450d7f296d9fe1e470854 authorization (39ms)

✓ Should allow 0x0130b57eb1127659376cfbabe4b81434b01c6b44 to authorize 0x25a6e4307e6f3e035c95b077a9b1c869b688febd for 1000 token spend, and

0x25a6e4307e6f3e035c95b077a9b1c869b688febd should be able to send these tokens to 0x6c6061667fde616d6ed450d7f296d9fe1e470854 (143ms)

✓ Should not allow 0x25a6e4307e6f3e035c95b077a9b1c869b688febd to transfer negative tokens from 0x0130b57eb1127659376cfbabe4b81434b01c6b44

✓ Should not allow 0x25a6e4307e6f3e035c95b077a9b1c869b688febd to transfer tokens from 0x0130b57eb1127659376cfbabe4b81434b01c6b44 to 0x0

✓ Should not transfer tokens to 0x0

✓ Should not allow 0x25a6e4307e6f3e035c95b077a9b1c869b688febd to transfer more tokens than authorized from 0x0130b57eb1127659376cfbabe4b81434b01c6b44

✓ Should allow an approval to be set, then increased, and decreased (180ms)

✓ Should allow safe functions (292ms)

Contract: TokenVesting

✓ Should require beneficiary be a valid address (238ms)

✓ Should require vesting cliff to be less time than vesting duration (201ms)

✓ Should show correct releasable amount

✓ Should require greater than zero vested balance to release

✓ Should release correct amounts during vesting duration (475ms)

✓ Should allow revoking unvested tokens (91ms)

✓ Should require no previous revoke (177ms)

✓ Should require revocable to be enabled (288ms)

6. All Contract Files Tested

File	Fingerprint (SHA256)
contracts/AirdropToken.sol	c8c62daed98d1852b6edbddef10e6b6244b5d8894593693f813e551c6f2a6b8f
contracts/BOBCrowdsale.sol	d9f99a4a1906c2800ea7a58d6ee7b6befc682462e13fca572cbae9df86f00dda
contracts/BOBPExchange.sol	e0bfbfda0174274c8ec4b1f86881cd8dfac0a73775c074009026b15d211fec31
contracts/BOBPToken.sol	c297762cefea3d9c4fdf567a6e553ef1056df0c253e84393b47321930b78ba9e
contracts/BOBToken.sol	3ce885b126238186823aae1b29ccd9c936b3203b7803de30a21aae49beed1a3f
contracts/BOBTokenVesting.sol	6e9ae2ed752eba0177622583b6c42b443bc5bd0ee97f20c4e69692090ba21ed4
contracts/BurnableToken.sol	bce2635eb2878b8c0e04cfcea6cb98419161058d26a664857908697a3f093673
contracts/PausableERC827Token.sol	f2132a58e08d88c601aaac92332e232e65079efa5ca95c8077bab4bd72639915
contracts/TokenReceiver.sol	710d5a6ecbf2113c8c3680ac72bae40cc6b476a7dd6537b0ea62c42ba9b48dc8
contracts/zeppelin/lifecycle/Destructible.sol	550d5ad716a13c2697316e82b55eaf7b8b401db9e150e33255675db723b35f73
contracts/zeppelin/lifecycle/Pausable.sol	78bf21e029fc3f1c38151915db9ccce2f0553bfeae9b6685fde1c297091cdb6f
contracts/zeppelin/math/SafeMath.sol	596bb5e82ff5009c31049f0cc9e5b8b95172fc7b38da5335bbdd09585c692f9c
contracts/zeppelin/ownership/CanReclaimToken.sol	4d60cbf85fa7699d4c75e74b1138a4bee36a17e29f5a2d7c5a20b7f8acf40e5c
contracts/zeppelin/ownership/HasNoContracts.sol	331974e7e007ccd9cec976f028474d2b5a913cd11939e61ce18213b85385e71d
contracts/zeppelin/ownership/HasNoEther.sol	d791b6a0cda1ad850da42899dc4bcf654f3cf768645a536e4fda565f23c69a01
contracts/zeppelin/ownership/HasNoTokens.sol	eff96eee7948ff4cca5d45b08529e17c12a44507c2774068e9bd4c51565588f5
contracts/zeppelin/ownership/NoOwner.sol	3f99b22cb4d4386bace5dd5bc0bd3c05d076deb39b4b42623ed36f1ddf06c0d4
contracts/zeppelin/ownership/Ownable.sol	fd49860e2f11bc70dc265e0d001048eeed921900ad2e4c9599fb4e9a8e3cae5c
contracts/zeppelin/token/ERC20/BasicToken.sol	2662ea846c0c6ca2bd32dfdb97ab20ae4108e7abade53962f146f634e9f4eba7

contracts/zeppelin/token/ERC20/ERC20.sol	6b75acd05c29968b057ec1facf659c064dbe0a79ac01444530629f01ef3a3abf
contracts/zeppelin/token/ERC20/ERC20Basic.sol	86c0a5fc6cb564ae77140da57a8ff9a22f46404240e69a6782ff741e286d373a
contracts/zeppelin/token/ERC20/MintableToken.sol	cba240b59fed2016bb9ed001a8c50a74150cf3e68eab41474914c7af21ff6427
contracts/zeppelin/token/ERC20/SafeERC20.sol	bde4d7e6d38ac1c64de5327ca306a61c3aea596b1e3ffe46fa21e8ec8b52cdc6
contracts/zeppelin/token/ERC20/StandardToken.sol	77e45da1164753f886d7395987b46deb036eca32c2e7322ef7a2764a08f7c5da
contracts/zeppelin/token/ERC20/TokenVesting.sol	ff85019deeae0c5ef19c5c3a29556c9ec5d0b3586764a8094094c977229d2d5f
contracts/zeppelin/token/ERC827/ERC827.sol	54711ac05c4b0006e8ca1388e0e36789bbbfef642bc1dcaa1535be67167156b8
contracts/zeppelin/token/ERC827/ERC827Token.sol	4a914ae91b7182ab730283b71496d436b08d1956ec4286880c710f23961e2fb2

7. Individual File Coverage Report

File	% Statements	% Branches	% Functions	% Lines
contracts/AirdropToken.sol	100.00%	83.33%	100.00%	100.00%
contracts/BOBCrowdsale.sol	100.00%	94.44%	100.00%	100.00%
contracts/BOBPExchange.sol	100.00%	92.86%	100.00%	100.00%
contracts/BOBPToken.sol	100.00%	100.00%	100.00%	100.00%
contracts/BOBToken.sol	100.00%	100.00%	100.00%	100.00%
contracts/BOBTokenVesting.sol	100.00%	100.00%	100.00%	100.00%
contracts/BurnableToken.sol	100.00%	100.00%	100.00%	100.00%
contracts/PausableERC827Token.sol	100.00%	100.00%	100.00%	100.00%
contracts/TokenReceiver.sol	100.00%	100.00%	100.00%	100.00%
contracts/zeppelin/lifecycle/Destructible.sol	100.00%	100.00%	100.00%	100.00%
contracts/zeppelin/lifecycle/Pausable.sol	100.00%	100.00%	100.00%	100.00%
contracts/zeppelin/math/SafeMath.sol	91.67%	50.00%	100.00%	91.67%
contracts/zeppelin/ownership/CanReclaimToken.sol	100.00%	100.00%	100.00%	100.00%
contracts/zeppelin/ownership/HasNoContracts.sol	100.00%	100.00%	100.00%	100.00%
contracts/zeppelin/ownership/HasNoEther.sol	100.00%	50.00%	100.00%	91.67%
contracts/zeppelin/ownership/HasNoTokens.sol	100.00%	100.00%	100.00%	100.00%
contracts/zeppelin/ownership/NoOwner.sol	100.00%	100.00%	100.00%	100.00%

contracts/zeppelin/ownership/Ownable.sol	100.00%	75.00%	100.00%	100.00%
contracts/zeppelin/token/ERC20/BasicToken.sol	100.00%	100.00%	100.00%	100.00%
contracts/zeppelin/token/ERC20/ERC20.sol	100.00%	100.00%	100.00%	100.00%
contracts/zeppelin/token/ERC20/ERC20Basic.sol	100.00%	100.00%	100.00%	100.00%
contracts/zeppelin/token/ERC20/MintableToken.sol	100.00%	100.00%	100.00%	100.00%
contracts/zeppelin/token/ERC20/SafeERC20.sol	100.00%	100.00%	100.00%	100.00%
contracts/zeppelin/token/ERC20/StandardToken.sol	100.00%	100.00%	100.00%	100.00%
contracts/zeppelin/token/ERC20/TokenVesting.sol	100.00%	100.00%	100.00%	100.00%
contracts/zeppelin/token/ERC827/ERC827.sol	100.00%	100.00%	100.00%	100.00%
contracts/zeppelin/token/ERC827/ERC827Token.sol	100.00%	65.00%	100.00%	100.00%
All files	**99.52%**	**85.94%**	**98.81%**	**99.57%**