DRAFT of Tune Contract Audit

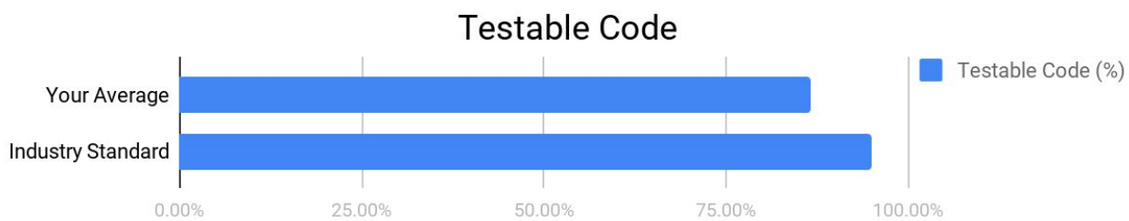
by Hosho, April 2018

NOTICE: *This document is a draft and is not representative of a completed audit.*

Overview

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

There is one high risk issue and multiple lower level issues that need to be addressed. (See Issues Found)



Testable code is lower than industry standard and can be improved.

It should be noted that this audit is not an endorsement of the reliability or effectiveness of the contract; it is 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, the Hosho Team recommends that the Tune staff put in place a bug bounty program to encourage further and active analysis of the smart contract.

Issues Found

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:

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

1.1. Unresolved, High: Deprecation System Unusable

TuneTokens

Explanation

Because the ERC-20 standard relies on msg.sender, it is not possible to blindly forward transaction state information to the remote contract. The msg.sender is overwritten when the new function is called, making msg.sender the ERC-20 source contract, not the txn.origin. As txn.origin is not safe, this entire system needs to be reviewed.

1.2. Unresolved, Medium: Missing Event

BasicToken

Explanation

The fee transfer in this contract does not emit a transfer event, breaking 3rd party integrations.

1.3. Unresolved, Medium: Missing Event

StandardToken

Explanation

The fee transfer in this contract does not emit a transfer event, breaking 3rd party integrations.

1.4. Unresolved, Medium: Missing Event

TuneTokens

Explanation

The issuance function does not trigger a transfer event, as it should for all creation of tokens.

1.5. Unresolved, Medium: Missing Event

TuneTokens

Explanation

The redeem function does not trigger a transfer event, as it should for the destruction of tokens.

1.6. Unresolved, Medium: Missing Event

TuneTokens

Explanation

If tokens are issued during contract generation, there is no transfer event issued.

1.7. Unresolved, Low: Transfer to 0x0

BasicToken

Explanation

Within the `transfer` function there is no validation preventing tokens from being sent to 0x0.

1.8. Unresolved, Low: Transfer to 0x0

StandardToken

Explanation

Within the `transferFrom` function there is no validation preventing tokens from being sent to 0x0.

1.9. Unresolved, Low: Not Needed Check

TuneTokens

Explanation

There are two initial checks within the `issue` function of this contract, one regarding `totalSupply` and the other regarding the balance. If the check for `totalSupply` does not fail, it is not possible that the balance check will then fail, rendering the second check unnecessary.

1.10. Unresolved, Informational: Possible Incorrect Token Issuance

TuneTokens

Explanation

The `_initialSupply` argument is not multiplied by 10^`decimals`. Thus, issuing 400m tokens, as per the 2_deploy_contracts.js file, will only issue 400m token units, and not 400m full tokens as it seems is intended.

1.11. Unresolved, Informational: Unused Code

Contactable

Explanation

The Contactable contract is never included and is not used.

Test Results

Failing Tests

1. Contract: Ownership Tests for TuneToken. Transfer Should not allow the owner to transfer to 0x0. (See Issue 1.7)
2. Contract: ERC-20 Tests for TuneToken. Should not allow 0x53353ef6da4bbb18d242b53a17f7a976265878d5 to transfer tokens from 0x667632a620d245b062c0c83c9749c9bfadf84e3b to 0x0. (See Issue 1.7)
3. Contract: ERC-20 Tests for TuneToken. Should not transfer tokens to 0x0. (See Issue 1.7)
4. Contract: TuneToken Custom Code. Should allow deprecation and calling into the upgraded contract. (See Issue 1.1)