SOLIDIFIED

Audit Report for Elect Global Enterprises Limited - August 14, 2022

Summary

Audit Report prepared by Solidified covering the Payout claim distributor smart contract.

Process and Delivery

Two (2) independent Solidified experts performed an unbiased and isolated audit of the code. The debrief was on 12 August 2022.

Audited Files

The source code has been supplied as files:

sha256 hash of the file audited:
`00f94db170a2bae2dfa21f672c7884877e840bc8386c43bbf5d0aa9cf685c2c6`

```
└── PayoutClaimDistributor.sol
```

Intended Behavior

The smart contracts implement an ERC20 token distribution function controlled by merkle proofs.

Code Complexity and Test Coverage

Smart contract audits are an important step to improve the security of smart contracts and can find many issues. However, auditing complex codebases have their limits and a remaining risk is present (see disclaimer).

Users of a smart contract system should exercise caution. In order to help with the evaluation of the remaining risk, we provide a measure of the following key indicators: **code complexity**, **code readability**, **level of documentation**, and **test coverage**.

Note that high complexity or lower test coverage does equate to a higher risk. Certain bugs are more easily detected in unit testing than a security audit and vice versa. It is, therefore, more likely that undetected issues remain if the test coverage is low or non-existent.

Criteria	Status	Comment
Code complexity	Low	-
Code readability and clarity	High	-
Level of Documentation	High	-
Test Coverage	-	-

Issues Found

Solidified found that the PayoutClaimDistributor contract contained no critical issues, no major issues, no minor issues and 2 informational notes.

We recommend all issues are amended, while the notes are up to the team's discretion, as they refer to best practices.

Issue #	Description	Severity	Status
1	PayoutClaimDistributor.sol: Consider updating solidity version	Note	
2	PayoutClaimDistributor.sol: Ignores transferFrom return value	Note	

Critical Issues

No issues found

Major Issues

No issues found

Minor Issues

No issues found

Notes

1. PayoutClaimDistributor.sol: Consider updating solidity version

The contract uses the solidity compiler version less than `0.8.0`. The recent Solidity releases include several bug fixes which are missing from the version used for the contract.

```
pragma solidity >=0.7.6 <0.8.0;
```

Recommendation
Consider updating the compiler version to `0.8.4` or greater.

2. PayoutClaimDistributor.sol: Ignores transferFrom return value

The method `claimPayout` ignores the return value of the `transferFrom` call. Though this is not an issue with the Animoca implementation of ERC20 token, it is recommended to validate the return value always especially when the interface allows it and it is easy to overlook when using a different token address for payouts.

```
IERC20(token).transferFrom(distAddress, account, amount);
```

Recommendation

Consider validating the return value if present during external function calls. The implementation can be inspired from OpenZeppelin's `SafeTransferFrom` method.

Disclaimer