Contract Audit Results

Prepared on: Mar 24, 2021

Contract: AUD434

Prepared by:

Charles Holtzkampf

Sentnl

Prepared for:

0xLeia

Unicly

Table of Contents

Executive Summary

This document outlines any issues found during the audit of the contracts:

- Converter.sol,GovernorAlpha.sol,TimeLock.sol,Unic.sol UnicFactory.sol,UnicFarm.sol,UnicGallery.sol,UnicPumper.sol

The contract has 0 flaws or security vulnerabilities. The risk associated with this contract is low

REMARK	MINOR	MAJOR	CRITICAL
1	2	0	0

Severity Description

REMARK

Remarks are instances in the code that are worthy of attention, but in no way represent a security flaw in the code. These issues might cause problems with the user experience, confusion with new developers working on the project, or other inconveniences.

Things that would fall under remarks would include:

- Instances where best practices are not followed
- Spelling and grammar mistakes
- Inconsistencies in the code styling and structure

MINOR

Issues of Minor severity can cause problems in the code, but would not cause the code to crash unexpectedly or for funds to be lost. It might cause results that would be unexpected by users, or minor disruptions in operations. Minor problems are prone to become major problems if not addressed appropriately.

Things that would fall under minor would include:

- Logic flaws (excluding those that cause crashes or loss of funds)
- Code duplication
- Ambiguous code

MAJOR

Issues of major security can cause the code to crash unexpectedly, or lead to deadlock situations.

Things that would fall under major would include:

- Logic flaws that cause crashes
- Timeout exceptions
- Incorrect ABI file generation
- Unrestricted resource usage (for example, users can lock all RAM on contract)

CRITICAL

Critical issues cause a loss of funds or severely impact contract usage.

Things that would fall under critical would include:

- Missing checks for authorization
- Logic flaws that cause loss of funds
- Logic flaws that impact economics of system
- All known exploits (for example, on_notification fake transfer exploit)

Methodology

Throughout the review process, we check that the token contract:

- Documentation and code comments match logic and behaviour
- Is not affected by any known vulnerabilities

Our team follows best practices and industry-standard techniques to verify the proper implementation of the smart contract. Our smart contract developers reviewed the contract line by line, documenting any issues as they were discovered.

Our strategies consist largely of manual collaboration between multiple team members at each stage of the review, including:

I. Due diligence in assessing the overall code quality of the codebase.
II. Testing contract logic against common and uncommon attack vectors.
III. Thorough, manual review of the codebase, line-by-line.

Our testing includes

- Overflow Audit
- Authority Control Audit Authority Vulnerability Audit
- Authority Excessive Audit
- Safety Design Audit Hard-coded Audit
- Show coding Audit
- Abnormal check Audit
- Type safety Audit
- Denial of Service Audit
- Performance Optimization Audit
- Design Logic Audit
- False Notice Audit
- False Error Notification Audit

- Counterfeit Token Audit
- Random Number Security Audit
- Rollback Attack Audit

Audit Results - Unicly

REMARK

Consider using reentrance guards (https://github.com/itinance/openzeppelin-solidity/blob/master/contracts/utils/ReentrancyGuard.sol) on external functions. That way, you can be completely certain there's no open re-entrant exploits, or race conditions

Converter.sol

MINOR

Possible timeout / expensive function, as poolInfo.length appears to be unbounded.

UnicFarm.sol#L167

Audit Results - Unicly

MINOR

It's worth reviewing up-gradability of contracts such as Converter.sol, similar to how the dev key can be set in UnicFarm.sol. It seems that if the issuer (msg.sender in createUTokens) loses their key, they may lose complete access to some of the contract features.

Contract Files

Filename	SHA256
Converter.sol	f5542cdcf43367e325bd5284d8bcaa4b84370fd63d304f2a24a9638fe6b12335
GovernorAlpha.sol	676715e2670d08f9c1ac971d0abd78c0569aada4e61a5476ebf6a903d8f620b0
TimeLock.sol	254310f8b67f63ee6a3040fb48d70fecfbf6133294db593126c8bff4bf88c6e1
Unic.sol	7b6711982ad8b9070a8e4dca030c63f904be9ed321b5a85f9cd79a49e4be18a7
UnicFactory.sol	d55fc0c3fb970d3efd83da6df9852346c18bc185eb0e1da3d425174515e8b524
UnicFarm.sol	fdc7994eaec11e54b536d90881d84ab51a17cab987e83229eb9ec905e1e297ec
UnicGallery.sol	d9da0d35e760bcecbf41572714da6588b165156e58473c5bc11d81fd6e2734ef
UnicPumper.sol	4c43f3fa7afd1bb0e6161ea8035e54c82d1c3c6088450b27a172ee2ba2833205