Protocol Audit Report

Version 1.0

Tura11

February 14, 2026

Protocol Audit Report

Tura11

February 14 2026

Prepared by: Tura11 Lead Auditor: Tura11

Table of Contents

Protocol Summary

1. Call the `enterRaffle` function with the following parameters:

    1. `address[] participants`: A list of addresses that enter. You can use this to enter yourself multiple times, or yourself and a group of your friends.

2. Duplicate addresses are not allowed
3. Users are allowed to get a refund of their ticket & `value` if they call the `refund` function
4. Every X seconds, the raffle will be able to draw a winner and be minted a random puppy
5. The owner of the protocol will set a feeAddress to take a cut of the `value`, and the rest of the funds will be sent to the winner of the puppy.

    - Compatibilities
- Roles
- Known Issues

Disclaimer

The Tura11 team makes all effort to find as many vulnerabilities in the code in the given time period, but holds no responsibilities for the findings provided in this document. A security audit by the team is not an endorsement of the underlying business or product. The audit was time-boxed and the review of the code was solely on the security aspects of the Solidity implementation of the contracts.

Risk Classification

		Impact		
		High	Medium	Low
	High	H	H/M	M
Likelihood	Medium	H/M	M	M/L
	Low	M	M/L	L

We use the CodeHawks severity matrix to determine severity. See the documentation for more details.

Audit Details

- Commit Hash: 2a47715b30cf11ca82db148704e67652ad679cd8
- In Scope:

Scope

```
./src/ PuppyRaffle.sol
```

Roles

Owner - Deployer of the protocol, has the power to change the wallet address to which fees are sent through the `changeFeeAddress` function. Player - Participant of the raffle, has the power to enter the raffle with the `enterRaffle` function and refund value through `refund` function.

Executive Summary

Ive learnt a lot from this audit, process was pretty easliy and fast. ## Issues found'

Severity	Number of issues found
Critical	1
High	2
Medium	0
Low	2
Info	3
Gas	2
Total	10

Findings

Critical

[C-1] Reentrancy allow entrant to drain raffle balance

High

[H-1] Denial of Service

[H-2] TITLE Weak RNG allow user to influecne or predict the winner

Medium

Low

[L-1] Old version of solidity allows overflow and underflow numbers.

[L-2] `PuppyRaffle::getActivePlayerIndex` returns 0 for non-existent players and for players at index 0, causing a player at index 0 to incorrectly thin they have not enteret the raffle

Informational

[I-1] Missing zero address check in constuctor

[I-2] Missing emit the events

[I-3] Dead code

Gas

[G-1] Unchanged state variables should be constant or immutable.

[G-2] Storage variables in loops should be cached