Stone wall
Smart Contract Security

PulseFun Betting Security Review

Introduction

A time-boxed security review of the **PulseFun Betting** contracts was conducted by **Stonewall**, with a focus on the security aspects of the smart contract implementation.

Disclaimer

A smart contract security review can never verify the complete absence of vulnerabilities. This is a time, resource and expertise bound effort where we try to find as many vulnerabilities as possible. We can not guarantee 100% security after the review or even if the review will find any problems with your smart contracts. Subsequent security reviews, bug bounty programs and on-chain monitoring are strongly recommended.

About Stonewall

Stonewall is an independent smart contract security firm delivering immovable protection for Web3 protocols. Our team brings deep expertise in DeFi security, having reviewed DEXs, yield farming protocols, gaming contracts, and complex financial systems.

About PulseFun Betting

PulseFun Betting is a decentralized prediction market system with two bet types:

- **Price Bets**: Users bet on whether a token's price will reach a target by a deadline
- **Custom Bets**: Admin-resolvable bets on arbitrary outcomes (sports, events, etc.)
- **Yield Integration**: Idle wager funds are deposited into PoolV3 lending to earn yield

Privileged Roles & Actors

Role	Description
Owner	Can manage admins, set dev wallet, configure tokens
Admin	Can whitelist tokens, set core tokens, configure pools
Bet Creator	Creates bets, receives 50% of fees, can cancel bets, resolve manually
DevWallet	Receives 50% of protocol fees

Observations

- Integrates with Gearbox-style PoolV3 lending pools
- Supports whitelisted tokens and FLUX factory tokens
- Price resolution via external oracle
- Manual resolution fallback for creator

Risk Classification

	High Impact	Medium Impact	Low Impact
High Likelihood	Critical	High	Medium
Medium Likelihood	High	Medium	Low
Low Likelihood	Medium	Low	Low

Security Assessment Summary

Review Details	
Protocol Name	PulseFun Betting
Repository	Private
Commit	`0055958b44f63163dabf0acc419472d2832cbee0`
Review Date	January 2026

Methods	Manual review, static analysis
Network	PulseChain

Project Links

Platform	Link
Twitter	@nexionpulse
Telegram	NexionPulse
DexScreener	NEON/PLS

Scope

Contract	SLOC
`Betting/BettingFactory.sol`	~370
`Betting/PriceBet.sol`	~550
`Betting/CustomBet.sol`	~400

Findings Summary

ID	Title	Severity	Status
[M-01]	Anyone can resolve bet with manual price	Medium	Open
[M-02]	Oracle price manipulation possible with low liquidity tokens	Medium	Open
[M-03]	recoverERC20 allows creator to steal wager tokens	Medium	Open
[L-01]	No minimum bet amount allows dust griefing	Low	Open
[L-02]	Creator can cancel bet after resolution time	Low	Open
[L-03]	Pool shares rounding can cause minor value loss	Low	Open

Findings

[M-01] Anyone can resolve bet with manual price

Severity: Medium

Impact: High - Attacker can resolve bet with manipulated price, stealing from legitimate winners.

Likelihood: Medium - Requires attacker to front-run legitimate resolution.

Location: `Betting/PriceBet.sol:238`

Description:

The `resolveBet(uint256 _finalPrice)` function has no access control:

```
function resolveBet(uint256 _finalPrice) external  {
    if (status != BetStatus.Active) revert BetNotActive();
    if (block.timestamp < resolutionTime) revert ResolutionTimeNotReached();

    finalPrice = _finalPrice;
    // ... resolution logic
}
```

Anyone can call this with any price after `resolutionTime`, potentially front-running the legitimate oracle resolution.

Attack Scenario:

1. Large bet pool exists with 70% betting YES (price >= $100)
2. Actual price is $110 (YES wins)
3. Attacker front-runs with `resolveBet(90)` making NO win
4. Attacker who bet NO takes the pool

Recommendation:

Add access control to manual resolution:

```
function resolveBet(uint256 _finalPrice) external {
    if (msg.sender != creator && !IBettingFactory(factory).admins(msg.sender))
        revert OnlyCreatorOrAdmin();
    // ...
}
```

[M-02] Oracle price manipulation possible with low liquidity tokens

Severity: Medium

Impact: High - Incorrect bet resolution due to manipulated prices.

Likelihood: Low - Requires low liquidity and significant capital.

Location: `Betting/PriceBet.sol:191`

Description:

The contract fetches price from an oracle at resolution time:

```
address oracle = IBettingFactory(factory).priceOracle();
finalPrice = IPriceOracleBase(oracle).getPrice(predictionToken);
```

If the oracle uses spot prices from DEXs, an attacker can manipulate the price in the same block via flash loan sandwich.

Recommendation:

- Use TWAP oracles
- Add price deviation checks
- Require multi-block price confirmation

[M-03] recoverERC20 allows creator to steal wager tokens

Severity: Medium

Impact: High - Creator can drain all bet funds.

Likelihood: Low - Requires malicious creator.

Location: `Betting/PriceBet.sol:341-345`

Description:

```
function recoverERC20(address tokenAddress, uint256 amount) external {
    if (msg.sender != creator) revert OnlyCreator();
    require(IERC20(tokenAddress).transfer(creator, amount), "Transfer failed");
}
```

No check prevents recovering the wager token itself, allowing creator to drain bet funds.

Recommendation:

```
function recoverERC20(address tokenAddress, uint256 amount) external {
    if (msg.sender != creator) revert OnlyCreator();
    require(tokenAddress != wagerToken, "Cannot recover wager token");
    require(IERC20(tokenAddress).transfer(creator, amount), "Transfer failed");
}
```

[L-01] No minimum bet amount allows dust griefing

Severity: Low

Location: `PriceBet.sol:156`

Description:

Users can place bets with 1 wei, creating many small positions that increase gas costs for resolution.

Recommendation:

Add minimum bet amount in factory configuration.

[L-02] Creator can cancel bet after resolution time

Severity: Low

Location: `PriceBet.sol:315-321`

Description:

```
function cancelBet() external {
    if (msg.sender != creator) revert OnlyCreator();
    if (status == BetStatus.Resolved) revert BetAlreadyResolved();
    // No check for block.timestamp >= resolutionTime
    status = BetStatus.Cancelled;
}
```

Creator can cancel after resolution time but before anyone calls `resolveBet()`, avoiding unfavorable outcomes.

Recommendation:

```
require(block.timestamp < resolutionTime, "Cannot cancel after resolution time");
```

[L-03] Pool shares rounding can cause minor value loss

Severity: Low

Location: `PriceBet.sol:459-468`

Description:

When depositing to lending pool, small rounding differences between shares received and expected can accumulate.

Security Patterns Observed

Positive

- Solidity ^0.8.17 with overflow protection
- Proper use of immutable variables
- Fee calculation prevents overflow (max 10%)
- Lending integration is optional (graceful fallback)

Concerns

- Missing access control on manual resolution
- No oracle manipulation protection
- Recovery function can steal wager tokens

- Centralized creator control

Conclusion

The PulseFun Betting contracts provide a flexible prediction market system with lending yield integration. The main concerns are around bet resolution security - both manual resolution access control and oracle manipulation risks.

We recommend addressing the Medium severity findings before mainnet deployment.

Overall Risk Assessment: Medium

This security review was conducted by Stonewall. For questions or clarifications, contact our team.