Stonewall
Smart Contract Security

Lemonad DEX Security Review

Introduction

A time-boxed security review of the **Lemonad DEX** 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 Lemonad DEX

Lemonad DEX is a Uniswap V2 fork providing decentralized token swaps on Monad:

- **LemonFactory**: Creates and manages liquidity pair contracts
- **LemonRouter**: Handles swap routing, liquidity addition/removal
- **LemonPair**: AMM liquidity pool implementation (constant product formula)
- **FeeCollector**: Aggregates and distributes trading fees
- **WMON**: Wrapped MON token for native token compatibility

Privileged Roles & Actors

Role	Description
Factory Owner	Can set fee recipient, update fee parameters
FeeCollector Owner	Controls fee distribution logic

Observations

- DEX closely follows the well-audited Uniswap V2 architecture
- Standard constant product (x*y=k) AMM model
- Custom fee collection mechanism via FeeCollector
- WMON follows standard WETH pattern

Risk Classification

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

Impact

- **High**: Leads to significant loss of user funds, protocol insolvency, or complete protocol failure
- **Medium**: Leads to partial loss of funds, temporary denial of service, or governance manipulation
- **Low**: Leads to minor issues, inconvenience, or suboptimal behavior

Likelihood

- **High**: Attack is easy to perform and likely to happen
- **Medium**: Attack requires specific conditions but is feasible
- **Low**: Attack requires significant effort, resources, or unlikely conditions

---

Security Assessment Summary

Review Details	
Protocol Name	Lemonad DEX
Repository	Private
Commit	`4bcdafa9703e197329cbc0cff193a50457decc6c`
Review Date	January 2026
Methods	Manual review, static analysis
Network	Monad Mainnet (Chain ID: 143)

Deployed Contract Addresses

Contract	Address
LemonFactory	`0x0FEFaB571c8E69f465103BeC22DEA6cf46a30f12`
LemonRouter	`0x491789a9Ad4d465B94843EE2e60120cD62d057dc`
FeeCollector	`0xA83cEA2E789dCFb3394164Cc468e671d133Ae433`
WMON	`0x48b43c8F46509a27454a4992dB656cD60c455e38`
LEMON/MON Pair	`0x310416b514e3C5AEc74A2A30ac21711753d4Dd6E`
MON/USDC Pair	`0x6c0BE7EAAB6b63ffD9f26Ec173985e49cfE2D28B`
USDC/LEMON Pair	`0xCf6Cd8929Cf28c99dDdC0A7d404d07Cce806B783`

Project Links

Platform	Link
Website	lemonad.one
Twitter	@LeMONAD_Factory
Telegram	LeMONAD_Factory
Discord	Join Discord

Docs	GitBook

Scope

Contract	SLOC
`dex/LemonRouter.sol`	~200
`dex/LemonPair.sol`	~300
`dex/LemonFactory.sol`	~100
`dex/FeeCollector.sol`	~80
`dex/WMON.sol`	~50

Findings Summary

ID	Title	Severity	Status
[L-01]	First liquidity provider can manipulate initial price	Low	Open
[L-02]	No slippage protection defaults in router	Low	Open

Findings

[L-01] First liquidity provider can manipulate initial price

Severity: Low

Impact: Medium - First LP can set an unfavorable initial price.

Likelihood: Low - Requires being the first LP and is economically risky for attacker.

Location: `dex/LemonPair.sol`

Description:

The first liquidity provider sets the initial price ratio of the pair. While this is standard Uniswap V2 behavior, it can be exploited if:

1. Attacker provides liquidity at skewed ratio
2. Unsuspecting users swap at unfavorable rates
3. Attacker extracts value through arbitrage

This is a known limitation of AMMs, not a bug.

Recommendation:

- Document the risk for users
- Consider adding minimum liquidity requirements for new pairs
- Frontend should warn users about new/low-liquidity pairs

[L-02] No slippage protection defaults in router

Severity: Low

Impact: Low - Users can be sandwiched if they don't set proper slippage.

Likelihood: Low - Requires user error in not setting slippage.

Location: `dex/LemonRouter.sol`

Description:

Router functions accept user-provided `amountOutMin` parameters. If users pass 0 or very low values, they can be sandwich attacked.

Recommendation:

Frontend should enforce reasonable slippage defaults (0.5-1%) and warn users setting very low slippage protection.

Informational Findings

[I-01] DEX Follows Standard Patterns (Positive)

The DEX contracts closely follow the well-audited Uniswap V2 implementation. Key security properties maintained:

- Reentrancy protection via balance checks and update ordering
- Proper use of SafeMath patterns (though not needed in 0.8.x)

- Flash loan protection through k-value checks
- Correct permit implementation for gasless approvals

[I-02] FeeCollector Integration

The custom FeeCollector adds a layer between swaps and fee distribution. This is properly integrated and doesn't introduce additional attack vectors.

Security Patterns Observed

Positive

- Solidity ^0.8.20+ with overflow protection
- Follows battle-tested Uniswap V2 architecture
- Proper reentrancy protection via CEI pattern
- K-value invariant checks prevent flash loan exploits
- Minimum liquidity burned to prevent price manipulation

Concerns

- Standard AMM limitations apply (impermanent loss, front-running)
- Centralized fee parameter control

Conclusion

The Lemonad DEX is a standard Uniswap V2 fork with no significant deviations from the well-audited original implementation. The codebase benefits from years of battle-testing of the Uniswap V2 architecture.

No critical or high severity issues were found. The low severity findings are inherent AMM limitations rather than implementation bugs.

Overall Risk Assessment: Low

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