OpenZeppelin | security

Universal Router 2.1 And V4 Periphery Audit

Uniswap

November 11, 2025

Table of Contents

Summary

Type	DeFi
Timeline	From 2025-10-30 To 2025-11-03
Languages	Solidity

Total Issues	2 (2 resolved)
Critical Severity Issues	0 (0 resolved)
High Severity Issues	0 (0 resolved)
Medium Severity Issues	0 (0 resolved)
Low Severity Issues	1 (1 resolved)
Notes & Additional Information	1 (1 resolved)

Scope

OpenZeppelin performed a differential audit of the following two pull requests:

- Pull request #457 in the Uniswap/v4-periphery repository at commit 4be7e48.
- Pull request #497 in the Uniswap/v4-periphery repository at commit 76f8813.

For pull request #457, the following files were in scope:

```
contracts
├── base
│   ├── Dispatcher.sol
│   └── RouteSigner.sol
├── deploy
│   └── UnsupportedProtocol.sol
├── interfaces
│   ├── external
│   │   └── IV3SpokePool.sol
│   └── IUniversalRouter.sol
├── libraries
│   ├── Commands.sol
│   ├── Locker.sol
│   └── MaxInputAmount.sol
├── modules
│   ├── ChainedActions.sol
│   └── uniswap
│       ├── v2
│       │   └── V2SwapRouter.sol
│       └── v3
│           └── V3SwapRouter.sol
└── UniversalRouter.sol
```

For pull request #497, only the `V4Router.sol` and `IV4Router.sol` files were in scope.

System Overview

This audit covers a series of changes and additions to Uniswap's Universal Router and v4 Periphery. The changes include the introduction of two new contracts: `ChainedActions.sol` and `RouteSigner.sol`, as well as the addition of per-hop slippage checks for swaps.

The per-hop slippage protection mechanism enables users to specify an array of expected price ratios for each intermediate hop in multi-step trades. This supplements existing safeguards like the minimum output and maximum input thresholds for exact input and exact output trades respectively, addressing prior limitations where individual pool pricing could not be enforced for multi-step trades.

The `RouteSigner` contract helps secure off-chain signed commands for a router system by verifying EIP-712 signatures, recovering the signer's address using ECDSA, and checks nonces to prevent replay attacks. It uses transient storage to hold the signer's details, intent, and data for the duration of a transaction, making it gas-efficient and secure without leaving permanent data behind. Apart from this, `ChainedActions.sol` was introduced which enables cross-chain bridging using the Across protocol.

Security Model and Trust Assumptions

Per-hop slippage checks are optional and can be unused by passing in an empty `maxHopSlippage` array. Nonces are not marked as checked if the nonce value is `type(uint256).max`. There is a slight rounding down of the `price` which is used for validation of per-slippage checks, but the effects are negligible due to the scaling up of `amountIn` by `1e18`.

Privileged Roles

The pull requests in review do not introduce or change any privileged roles.

Low Severity

L-01 `UniversalRouter` May Grant Allowance to Arbitrary Address

The `_acrossV4DepositV3` function of the `ChainedActions` contract implements the logic required to support bridging using the Across protocol. Prior to the call to the local `SpokePool`, `UniversalRouter` grants the target `spokePool` an allowance for the `inputAmount`. `spokePool` and `inputAmount` parameters are user-supplied values which makes it possible for an arbitrary account to be granted allowance by the router for the supplied `inputAmount` of `inputToken`. The `UniversalRouter` is not intended to hold any user funds and thus the impact is limited to external actor error. However, it is good practice to limit unintended approvals. For supported chains Across deploys one `SpokePool` per network.

Consider adding the local network's `SpokePool` address as an immutable value to ensure that approvals are only ever granted to the intended contract address.

Update: *Resolved in pull request #461 by making the Spoke Pool address immutable. The team stated:*

> *Fixed*

Notes & Additional Information

N-01 Missing Named Parameters in Mapping

Since Solidity `0.8.18`, mappings can include named parameters to provide more clarity about their purpose. Named parameters allow mappings to be declared in the form `mapping(KeyType KeyName? => ValueType ValueName?)`. This feature enhances code readability and maintainability.

In the `noncesUsed` state variable of the `RouteSigner` contract, the mapping does not have any named parameters.

Consider adding named parameters to mappings in order to improve the readability and maintainability of the codebase.

Update: *Resolved in pull request #461. The team stated:*

> *Fixed*

Conclusion

The audited changes introduce two distinct features: firstly, the V4 Router has been upgraded to enable per-hop slippage checks. Secondly, the Universal Router has been upgraded to allow cross-chain bridging as part of its accepted commands.

No significant issues were identified. The feature implementations were found to be robust and well-structured.

The Uniswap Labs team is appreciated for their support during the audit.