Security Audit Report

Voltage Finance Routers Contracts

PREPARED FOR:

Volt Router Exchange

ARCADIA CONTACT INFO
Email: audits@arcadiamgroup.com
Telegram: https://t.me/thearcadiagroup

Repository

https://github.com/voltfinance/voltage-router

Revision history

Date	Commit
7/29/2023	#ee758af099be7630ca110a3bb9ff70b9930cee1c
9/01/2023	#72c7cda8a06fe191aea47abda9fd747248bf32a2

Table of Contents

Executive Summary

Introduction

Voltage Finance engaged Arcadia to perform a security audit of their smart contracts within the voltage-routers repository within the Voltage organization. Our review of their codebase occurred on the commit hash #**ee758af099be7630ca110a3bb9ff70b9930cee1c** and the fix on the commit hash **#72c7cda8a06fe191aea47abda9fd747248bf32a2**

Review Team

1. Tuan "Anhnt" Nguyen - Security Researcher and Engineer
2. Joel Farris - Project Manager

Project Background

volt-router is the trusted settlement layer for the permissionless global exchange of value forked from **0xprotocol** with modifications in relation to removal of trading fees, meta/otc/native orders.

Coverage

For this audit, we performed research, test coverage, investigation, and review of Volt followed by issue reporting, along with mitigation and remediation instructions as outlined in this report. The following code repositories are considered in scope for the review.

Files
contracts/router/contracts/src/IRouter.sol
contracts/router/contracts/src/Router.sol
contracts/router/contracts/src/RouterOptimized.sol
contracts/router/contracts/src/errors/LibCommonRichErrors.sol
contracts/router/contracts/src/errors/LibLiquidityProviderRichErrors.sol
contracts/router/contracts/src/errors/LibMetaTransactionsRichErrors.sol

contracts/router/contracts/src/errors/LibNFTOrdersRichErrors.sol
contracts/router/contracts/src/errors/LibNativeOrdersRichErrors.sol
contracts/router/contracts/src/errors/LibOwnableRichErrors.sol
contracts/router/contracts/src/errors/LibProxyRichErrors.sol
contracts/router/contracts/src/errors/LibSignatureRichErrors.sol
contracts/router/contracts/src/errors/LibSimpleFunctionRegistryRichErrors.sol
contracts/router/contracts/src/errors/LibTransformERC20RichErrors.sol
contracts/router/contracts/src/errors/LibWalletRichErrors.sol
contracts/router/contracts/src/external/FeeCollector.sol
contracts/router/contracts/src/external/FeeCollectorController.sol
contracts/router/contracts/src/external/FlashWallet.sol
contracts/router/contracts/src/external/IFlashWallet.sol
contracts/router/contracts/src/external/ILiquidityProviderSandbox.sol
contracts/router/contracts/src/external/LibFeeCollector.sol
contracts/router/contracts/src/external/LiquidityProviderSandbox.sol
contracts/router/contracts/src/external/PermissionlessTransformerDeployer.sol
contracts/router/contracts/src/external/TransformerDeployer.sol
contracts/router/contracts/src/features/BootstrapFeature.sol
contracts/router/contracts/src/features/ERC165Feature.sol
contracts/router/contracts/src/features/FundRecoveryFeature.sol
contracts/router/contracts/src/features/LiquidityProviderFeature.sol
contracts/router/contracts/src/features/OwnableFeature.sol
contracts/router/contracts/src/features/PancakeSwapFeature.sol
contracts/router/contracts/src/features/SimpleFunctionRegistryFeature.sol
contracts/router/contracts/src/features/TransformERC20Feature.sol
contracts/router/contracts/src/features/UniswapFeature.sol
contracts/router/contracts/src/features/UniswapV3Feature.sol
contracts/router/contracts/src/features/interfaces/IBootstrapFeature.sol
contracts/router/contracts/src/features/interfaces/IERC1155OrdersFeature.sol
contracts/router/contracts/src/features/interfaces/IERC165Feature.sol
contracts/router/contracts/src/features/interfaces/IERC721OrdersFeature.sol
contracts/router/contracts/src/features/interfaces/IFeature.sol

contracts/router/contracts/src/features/interfaces/IFundRecoveryFeature.sol
contracts/router/contracts/src/features/interfaces/ILiquidityProviderFeature.sol
contracts/router/contracts/src/features/interfaces/IOwnableFeature.sol
contracts/router/contracts/src/features/interfaces/IPancakeSwapFeature.sol
contracts/router/contracts/src/features/interfaces/ISimpleFunctionRegistryFeature.sol
contracts/router/contracts/src/features/interfaces/ITokenSpenderFeature.sol
contracts/router/contracts/src/features/interfaces/ITransformERC20Feature.sol
contracts/router/contracts/src/features/interfaces/IUniswapFeature.sol
contracts/router/contracts/src/features/interfaces/IUniswapV3Feature.sol
contracts/router/contracts/src/features/libs/LibNFTOrder.sol
contracts/router/contracts/src/features/libs/LibNativeOrder.sol
contracts/router/contracts/src/features/libs/LibSignature.sol
contracts/router/contracts/src/features/nft_orders/ERC1155OrdersFeature.sol
contracts/router/contracts/src/features/nft_orders/ERC721OrdersFeature.sol
contracts/router/contracts/src/features/nft_orders/NFTOrders.sol
contracts/router/contracts/src/fixins/FixinCommon.sol
contracts/router/contracts/src/fixins/FixinEIP712.sol
contracts/router/contracts/src/fixins/FixinERC1155Spender.sol
contracts/router/contracts/src/fixins/FixinERC721Spender.sol
contracts/router/contracts/src/fixins/FixinProtocolFees.sol
contracts/router/contracts/src/fixins/FixinReentrancyGuard.sol
contracts/router/contracts/src/fixins/FixinTokenSpender.sol
contracts/router/contracts/src/liquidity-providers/CurveLiquidityProvider.sol
contracts/router/contracts/src/liquidity-providers/MooniswapLiquidityProvider.sol
contracts/router/contracts/src/migrations/FullMigration.sol
contracts/router/contracts/src/migrations/InitialMigration.sol
contracts/router/contracts/src/migrations/LibBootstrap.sol
contracts/router/contracts/src/migrations/LibMigrate.sol
contracts/router/contracts/src/storage/LibERC1155OrdersStorage.sol
contracts/router/contracts/src/storage/LibERC721OrdersStorage.sol
contracts/router/contracts/src/storage/LibMetaTransactionsStorage.sol
contracts/router/contracts/src/storage/LibMetaTransactionsV2Storage.sol

contracts/router/contracts/src/storage/LibNativeOrdersStorage.sol
contracts/router/contracts/src/storage/LibOwnableStorage.sol
contracts/router/contracts/src/storage/LibProxyStorage.sol
contracts/router/contracts/src/storage/LibReentrancyGuardStorage.sol
contracts/router/contracts/src/storage/LibSimpleFunctionRegistryStorage.sol
contracts/router/contracts/src/storage/LibStorage.sol
contracts/router/contracts/src/storage/LibTransformERC20Storage.sol
contracts/router/contracts/src/transformers/FillQuoteTransformer.sol
contracts/router/contracts/src/transformers/IERC20Transformer.sol
contracts/router/contracts/src/transformers/LibERC20Transformer.sol
contracts/router/contracts/src/transformers/LogMetadataTransformer.sol
contracts/router/contracts/src/transformers/PayTakerTransformer.sol
contracts/router/contracts/src/transformers/Transformer.sol
contracts/router/contracts/src/transformers/WethTransformer.sol
contracts/router/contracts/src/transformers/bridges/AbstractBridgeAdapter.sol
contracts/router/contracts/src/transformers/bridges/ArbitrumBridgeAdapter.sol
contracts/router/contracts/src/transformers/bridges/AvalancheBridgeAdapter.sol
contracts/router/contracts/src/transformers/bridges/BSCBridgeAdapter.sol
contracts/router/contracts/src/transformers/bridges/BridgeAdapter.sol
contracts/router/contracts/src/transformers/bridges/BridgeProtocols.sol
contracts/router/contracts/src/transformers/bridges/CeloBridgeAdapter.sol
contracts/router/contracts/src/transformers/bridges/EthereumBridgeAdapter.sol
contracts/router/contracts/src/transformers/bridges/FantomBridgeAdapter.sol
contracts/router/contracts/src/transformers/bridges/IBridgeAdapter.sol
contracts/router/contracts/src/transformers/bridges/OptimismBridgeAdapter.sol
contracts/router/contracts/src/transformers/bridges/PolygonBridgeAdapter.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinAaveV2.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinAaveV3.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinBalancer.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinBalancerV2Batch.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinBancorV3.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinBarter.sol

contracts/router/contracts/src/transformers/bridges/mixins/MixinCompound.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinCryptoCom.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinCurve.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinCurveV2.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinDodo.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinDodoV2.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinGMX.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinKyberDmm.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinKyberElastic.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinLido.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinMStable.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinMakerPSM.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinMooniswap.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinNerve.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinPlatypus.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinSolidly.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinSynthetix.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinTraderJoeV2.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinUniswap.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinUniswapV2.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinUniswapV3.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinWOOFi.sol
contracts/router/contracts/src/transformers/bridges/mixins/MixinZeroExBridge.sol
contracts/router/contracts/src/vendor/IERC1155Token.sol
contracts/router/contracts/src/vendor/IERC721Token.sol
contracts/router/contracts/src/vendor/IFeeRecipient.sol
contracts/router/contracts/src/vendor/ILiquidityProvider.sol
contracts/router/contracts/src/vendor/IMooniswapPool.sol
contracts/router/contracts/src/vendor/IPropertyValidator.sol
contracts/router/contracts/src/vendor/ITakerCallback.sol
contracts/router/contracts/src/vendor/IUniswapV2Pair.sol
contracts/router/contracts/src/vendor/IUniswapV3Pool.sol

contracts/router/contracts/src/vendor/v3/IERC20Bridge.sol
contracts/router/contracts/src/vendor/v3/IStaking.sol

Methodology

Arcadia completed this security review using various methods, primarily consisting of dynamic and static analysis. This process included a line-by-line analysis of the in-scope contracts, optimization analysis, analysis of key functionalities and limiters, and reference against intended functionality.

The followings are the steps we have performed while auditing the smart contracts:

- Investigating the project and its technical architecture overview through its documentation
- Understanding the overview of the smart contracts, the functions of the contracts, the inheritance, and how the contracts interface with each others thanks to the graph created by Solidity Visual Developer
- Manual smart contract audit:
  - Review the code to find any issue that could be exploited by known attacks listed by Consensys
  - Identifying which existing projects the smart contracts are built upon and what are the known vulnerabilities and remediations to the existing projects
  - Line-by-line manual review of the code to find any algorithmic and arithmetic related vulnerabilities compared to what should be done based on the project's documentation
  - Find any potential code that could be refactored to save gas
  - Run through the unit-tests and test-coverage if exists
- Static Analysis:
  - Scanning for vulnerabilities in the smart contracts using Static Code Analysis Software
  - Making a static analysis of the smart contracts using Slither
- Additional review: a follow-up review is done when the smart contracts have any new update. The follow-up is done by reviewing all changes compared to the audited commit revision and its impact to the existing source code and found issues.

Summary

There were **6** issues found, **0** of which were deemed to be ‘critical’, and **1** of which were rated as ‘high’.

Severity Rating	Number of Original Occurrences	Number of Remaining Occurrences
CRITICAL	0	0
HIGH	1	0
MEDIUM	0	0
LOW	0	0
INFORMATIONAL	5	1

Findings in Manual Audit

(VoR-1) Draining all ETH balance when transforming.

Status

Resolved

Risk Level

Severity: High

Code Segment

```
function transform(TransformContext calldata context) external override returns (bytes4
magicBytes) {
      // .. before settle code block
       state.ethRemaining = address(this).balance;
       // .. after settle code block
       if (state.ethRemaining > 0 && data.refundReceiver != address(0)) {
          bool transferSuccess;
           if (data.refundReceiver == REFUND_RECEIVER_RECIPIENT) {
             (transferSuccess,) = context.recipient.call{value: state.ethRemaining}("");
           } else if (data.refundReceiver == REFUND_RECEIVER_SENDER) {
             (transferSuccess,) = context.sender.call{value: state.ethRemaining}("");
          } else {
             (transferSuccess,) = data.refundReceiver.call{value: state.ethRemaining}("");
          }
          require(transferSuccess, "FillQuoteTransformer/ETHER_TRANSFER_FALIED");
      }
}
```

Description

Due to the removal of the `protocolFee` (with `protocolFeePaid` is always zero) and the subsequent decision to return all ETH to the sender, caution is advised when utilizing the following code snippet
`state.ethRemaining = address(this).balance;`

It's important to note that the contract operates using `erc2535`, which means invoking the transformation from `FillQuoteTransformer` will result in a complete depletion of any remaining ETH. This introduces a significant element of risk into the process.

Code Location

```
contracts/router/contracts/src/transformers/FillQuoteTransformer.sol
```

Recommendation

Return the `msg.value` or no return at all.

(VoR-2) `protocolFee` variable is unused.

Status

Resolved

Risk Level

Severity: Informational

Code Segment

```
/// @dev Intermediate state variables to get around stack limits.
struct FillState {
   uint256 ethRemaining;
   uint256 boughtAmount;
   uint256 soldAmount;
   uint256 protocolFee;
   uint256 takerTokenBalanceRemaining;
}
```

Description

The `protocolFee` field is unused.

Code Location

```
contracts/router/contracts/src/transformers/FillQuoteTransformer.sol
```

Recommendation

Remove the unused field.

(VoR-3) Lots of unutilized code, inadequate documentation, and absent deployment scripts.

Status

Resolved

Risk Level

Severity: Informational

Code Segment

Description

The project has eliminated the MetaTransaction, OTC, and Native order functionalities from the fork. However, numerous traces of code related to these components/modular functionalities remain in the codebase. Given the extensive codebase, we strongly recommend including deployment scripts and thorough documentation for review. This proactive approach will help mitigate potential production issues.

Recommendation

Revise the documentation, enhance deployment scripts, and eliminate redundant code.

(VoR-4) `protocolFeePaid` is always zero

Status

Resolved

Risk Level

Severity: Low

Code Segment

```
function transform(TransformContext calldata context) external override
returns (bytes4 magicBytes)
{
  …
  FillOrderResults memory results = _fillBridgeOrder(data.bridgeOrders[i], data,
state);
  …
  state.ethRemaining = state.ethRemaining.safeSub(results.protocolFeePaid);
}
```

Description

The function `_fillBridgeOrder` doesn't set any value for protocolFeePaid, that means the calculation for ethRemaining is not needed.

Code Location

```
contracts/router/contracts/src/transformers/FillQuoteTransformer.sol
```

Recommendation

Remove the `ethRemaining`,`protocolFeePaid` fields and the related code.

(VoR-5) Event parameters should be indexed.

Status

Unresolved - the team acknowledged

Risk Level

Severity: Informational

Code Segment

```
// file contracts/router/contracts/src/transformers/FillQuoteTransformer.sol
event ProtocolFeeUnfunded(bytes32 orderHash);
```

```
// file contracts/router/contracts/src/transformers/LogMetadataTransformer.sol
event TransformerMetadata(address sender, address taker, bytes data);

// file contracts/router/contracts/src/transformers/bridges/IBridgeAdapter.sol
event BridgeFill(
       bytes32 source,
       IERC20Token inputToken,
       IERC20Token outputToken,
       uint256 inputTokenAmount,
       uint256 outputTokenAmount
);

// file contracts/router/contracts/src/features/interfaces/IERC721OrdersFeature.sol
event ERC721OrderFilled(
       LibNFTOrder.TradeDirection direction,
       address maker,
       address taker,
       uint256 nonce,
       IERC20Token erc20Token,
       uint256 erc20TokenAmount,
       IERC721Token erc721Token,
       uint256 erc721TokenId,
       address matcher
);
event ERC721OrderCancelled(address maker, uint256 nonce);
event ERC721OrderPreSigned(
       LibNFTOrder.TradeDirection direction,
       address maker,
       address taker,
       uint256 expiry,
       uint256 nonce,
       IERC20Token erc20Token,
       uint256 erc20TokenAmount,
       LibNFTOrder.Fee[] fees,
       IERC721Token erc721Token,
       uint256 erc721TokenId,
      LibNFTOrder.Property[] erc721TokenProperties
);
//
```

Description

Code Location

```
contracts/router/contracts/src/transformers/FillQuoteTransformer.sol
contracts/router/contracts/src/transformers/LogMetadataTransformer.s
ol
contracts/router/contracts/src/transformers/bridges/IBridgeAdapter.s
ol
contracts/router/contracts/src/features/interfaces/IERC721OrdersFeat
ure.sol
```

Recommendation

Conduct a review and contemplate indexing the significant events.

(VoR-6) Old and mixed solidity versions

Status

Unresolved - the team acknowledged

Risk Level

Severity: Informational

Description

The codebase necessitates Solidity version ^0.8.19 for certain libraries and ^0.6.5 for the main codebase.

Recommendation

Contemplate migrating to version ^0.8.x .

Automated Tests and Tooling

Static Analysis with Slither

As a part of our engagement with Volt, we ran a static analysis against the source code using Slither, which is a Solidity static analysis framework written in Python. Slither runs a suite of vulnerability detectors and prints visual information about contract details. Slither enables developers to find vulnerabilities, enhance their code comprehension, and quickly prototype custom analyses.

While Slither is not the primary element of Arcadia's offering, in some cases, it can be useful. The following shows the results found by the static analysis by Slither. We reviewed the results, and all of the issues found by Slither were at that point in time false positives.

Conclusion

Arcadia identified issues that occurred at hash **#ee758af099be7630ca110a3bb9ff70b9930cee1c** and we have collaborated with the Voltage team to address and resolve these issues in commit **72c7cda8a06fe191aea47abda9fd747248bf32a2**.

Disclaimer

While best efforts and precautions have been taken in the preparation of this document, The Arcadia Group and the Authors assume no responsibility for errors, omissions, or damages resulting from the use of the provided information. Additionally, Arcadia would like to emphasize that the use of Arcadia's services does not guarantee the security of a smart contract or set of smart contracts and does not guarantee against attacks. One audit on its own is not enough for a project to be considered secure; that categorization can only be earned through extensive peer review and battle testing over an extended period.