Universal Deposit Audit

Gnosis Ltd - Report by Côme du Crest

2025-09-17

Table of contents

Universal Deposit Audit

This document presents the findings of a smart contract audit conducted by Côme du Crest for Gnosis Ltd.

Scope

The scope includes all contracts within `gnosischain/universal-deposit` as of commit `0x92d1bdf`.

Context

Universal deposit accounts are created at a deterministic address via a proxy factory. These accounts are configured to send a specific token to a recipient on a different chain via Stargate. The account queries the universal deposit manager for information about the Stargate route. The universal deposit manager hold Stargate route information set by its owner.

Status

The report has been sent to the core developer.

Issues have been fixed and last reviewed commit is `0x90a6c97`.

Issues

[Low] No slippage protection on settle()

Summary

The `settle()` function has no slippage parameter on amount of token received by the user on destination chain. The `sendParam.minAmountLD` is populated by whatever `Stargate` yields for `receipt.amountReceivedLD` which may incur a fee charged in transferred token.

Vulnerability Detail

The `settle()` function uses `quoteStargateFee()` to compute the `sendParam` sent to the Stargate router:

```
1   function settle(
2     address srcToken
3   ) public payable returns (MessagingReceipt memory msgReceipt, OFTReceipt
      memory oftReceipt) {
4     IUniversalDepositManager.StargateTokenRoute memory stargateTokenRoute =
        udManager.getStargateRoute(srcToken);
5
6     uint256 bridgeAmount = IERC20(srcToken).balanceOf(address(this));
7
8     ...
9
10    (uint256 valueToSend, SendParam memory sendParam, MessagingFee memory
        messagingFee) =
11      quoteStargateFee(bridgeAmount, stargateTokenRoute.srcStargateToken);
12    if (address(this).balance < valueToSend) revert InsufficientNativeToken(
        address(this).balance, valueToSend);
13
14    (msgReceipt, oftReceipt,) =
15      IStargate(stargateTokenRoute.srcStargateToken).sendToken{value:
          valueToSend}(sendParam, messagingFee, owner()); // Use owner as
          refundAddress
16    ...
17  }
```

The function `quoteStargateFee()` quotes Stargate for the minimum amount of tokens to receive `minAmountLD`:

```
1   function quoteStargateFee(
2     uint256 amount,
3     address srcStargateToken
4   ) public view returns (uint256 valueToSend, SendParam memory sendParam,
      MessagingFee memory messagingFee) {
5     if (amount == 0) revert AmountNotEnough(amount);
6     sendParam = SendParam({
```

```
        dstEid: udManager.chainIdToEidMap(dstChainId),
        to: Utils._addressToBytes32(recipient),
        amountLD: amount,
        minAmountLD: amount, // Will be updated with quote
        extraOptions: new bytes(0), // Default, can be customized
        composeMsg: new bytes(0), // Default, can be customized
        oftCmd: '' // Empty for taxi mode
      });

      // Get accurate minimum amount from quote
      (,, OFTReceipt memory receipt) = IStargate(srcStargateToken).quoteOFT(
          sendParam);
      sendParam.minAmountLD = receipt.amountReceivedLD;

      ...
    }
```

This amount may incur a fee charged on transferred token that the user calling `settle()` did not explicitly approve.

Impact

The amount of tokens received by the user may be unexpected due to Stargate fee.

Code Snippets

https://github.com/gnosischain/universal-deposit/blob/92d1bdf382829154ae9e721ccc2462523a187a5a/packages/contracts/src/UniversalDepositAccount.sol#L189-L190

Recommendation

Add a user-specified value for the minimum amount of tokens to receive.

Response

As of commit `0x5f23b44`, a slippage parameter has been added to `settle()` and `quoteStargateFee()` that make the functions revert if expected received amount is too low. Unfortunately, the slippage is checked on `receipt.amountSentLD` the amount of token to send instead of `receipt.amountReceivedLD` the amount of token to receive.

This has later been fixed in commit `0x90a6c97` by using `receipt.amountReceivedLD` to check slippage.

[Info] Cannot route one local token to two different chains

Summary

The mappings in `UniversalDepositManager` use `srcToken` address as key, which means you cannot route one local token to two different chains.

Vulnerability Detail

The mapping `tokenRouteMap` uses `srcToken` as key:

```
contract UniversalDepositManager ... {
    ...
    mapping(address srcToken => TokenRoute tokenRoute) public tokenRouteMap;
    ...
}
```

Impact

Cannot route one token on local chain to different chains.

Code Snippets

https://github.com/gnosischain/universal-deposit/blob/92d1bdf382829154ae9e721ccc2462523a187a5a/packages/con

Recommendation

Add `dstChainId` as a key in the mapping `tokenRouteMap` to be able to configure a different route for each `(token, destination)` pair.

Response

The mapping has been updated to add `dstChainId` as a key:

```
    mapping(address srcToken => mapping(uint256 dstChainId =>
        StargateTokenRoute stargateTokenRoute)) public stargateTokenRouteMap;
```

[Info] UniversalDepositManager.setRoute() is useless without setStargateRoute()

Summary

The function `UniversalDepositManager.setRoute()` sets the route as enabled but the route does not actually work until `setStargateRoute()` is called. I don't see the point of this function right now.

Vulnerability Detail

The function `setRoute()` sets the route as supported, but the route does not work unless the `stargateTokenRouteMap` mapping is set by `setStargateRoute()`:

```
function setRoute(
  TokenRoute calldata tokenRoute
) public onlyOwner {
  tokenRouteMap[tokenRoute.srcToken] = tokenRoute;
  isTokenSupported[tokenRoute.srcToken] = true;
  bytes32 routeKey = getRouteKey(tokenRoute);
  isRouteSupported[routeKey] = true;
}

  function setStargateRoute(
  StargateTokenRoute calldata stargateTokenRoute
) public onlyOwner {
  stargateTokenRouteMap[stargateTokenRoute.tokenRoute.srcToken] =
      stargateTokenRoute;
  setRoute(stargateTokenRoute.tokenRoute);
}
```

The account `settle()` function calls `udManager.getStargateRoute()` to get information about Stargate route:

```
function settle(
  address srcToken
) public payable returns (MessagingReceipt memory msgReceipt, OFTReceipt
    memory oftReceipt) {
  IUniversalDepositManager.StargateTokenRoute memory stargateTokenRoute =
      udManager.getStargateRoute(srcToken);
  ...
}
```

`getStargateRoute()` uses the `stargateTokenRouteMap` mappign set by `setStargateRoute()`

```
function getStargateRoute(
  address srcToken
) public view returns (StargateTokenRoute memory stargateTokenRoute) {
  return stargateTokenRouteMap[srcToken];
```

```
5     }
```

Impact

The function `UniversalDepositManager.setRoute()` is pointless as an external function.

Code Snippets

https://github.com/gnosischain/universal-deposit/blob/92d1bdf382829154ae9e721ccc2462523a187a5a/packages/contracts/src/UniversalDepositManager.sol#L64-L71

Recommendation

Make the function `setRoute()` internal or remove it.

Response

The function `setRoute()` has been removed and `setStargateRoute()` now sets the required mappings.

Optimisations and miscellaneous

This part lists minor gas/code optimizations that shouldn't make the code less readable or improve overall readability. It also lists questions about unclear code segments.

Typo

uses `givernAddress` instead of `givenAddress`

https://github.com/gnosischain/universal-deposit/blob/92d1bdf382829154ae9e721ccc2462523a187a5a/packages/contracts/src/UniversalDepositAccount.sol#L56-L57

This has been fixed.