



LIDO WSTETH ON BNB DEPLOYMENT VERIFICATION

CONTENTS

1. REPORT OVERVIEW	3
1.1. SCOPE	4
1.2. CONCLUSION	6
2. VERIFICATION DETAILS	7
2.1. NETWORK SPECIFIC BEHAVIOR	8
2.2. SCOPE CHECKING	9
2.3. AUDIT REPORT INVESTIGATION	10
2.4. DEPLOY SCRIPT CHECK.....	11
2.5. DEPLOYMENT VERIFICATION	12
2.6. INITIALIZATION PARAMETERS CHECK.....	14
2.7. ROLE MODEL VERIFICATION	15
2.8. STORAGE CHECK	16
2.9. DOCUMENTATION VERIFICATION.....	17
3. APPENDIX	18
3.1. DISCLAIMER	19
3.2. ABOUT OXORIO	20

1

REPORT OVERVIEW

1.1 SCOPE

Contract	Network	Address
Token	Ethereum	0x7f39C581F595B53c5cb19bD0b3f8dA6c935E2Ca0
Token	BNB	0x26c5e01524d2E6280A48F2c50fF6De7e52E9611C
NTT Manager	Ethereum	0xb948a93827d68a82F6513Ad178964Da487fe2BD9
NTT Manager	BNB	0x6981F5621691CBfE3DdD524dE71076b79F0A0278
Wormhole Transceiver	Ethereum	0xA1ACC1e6edaB281Febd91E3515093F1DE81F25c0
Wormhole Transceiver	BNB	0xbe3F7e06872E0dF6CD7FF35B7aa4Bb1446DC9986
Axelar Transceiver	Ethereum	0x723AEAD29acee7E9281C32D11eA4ed0070c41B13
Axelar Transceiver	BNB	0x723AEAD29acee7E9281C32D11eA4ed0070c41B13
Transceiver Structs (used by NTT Manager and Wormhole Transceiver)	Ethereum	0xf0396a8077eda579f657B5E6F3c3F5e8EE81972b
Transceiver Structs (used by NTT Manager and Wormhole Transceiver)	BNB	0xf0396a8077eda579f657B5E6F3c3F5e8EE81972b
Transceiver Structs (used by Axelar Transceiver)	Ethereum	0xa12bc993d8144404a8c8c812816048275a066ced
Transceiver Structs (used by Axelar Transceiver)	BNB	0x27a3daf3b243104e9b0afae6b56026a416b852c9

Audit reports

- ◆ **wstEthL2Token:** [Cyfrin wstEth L2 Token Audit Report](#)
- ◆ **Axelar Transceiver:** [Cyfrin Axelar Transceiver Audit Report](#)
- ◆ **Wormhole:** [Cyfrin Wormhole Foundation EVM-NTT Audit Report](#)
- ◆ **Wormhole:** [Cyfrin Wormhole Foundation EVM-NTT\(v1.1.0+evm\) Audit Report](#)
- ◆ **Wormhole:** [Cantina Wormhole Native Token Transfers Audit Report](#)

Deployment scripts

- ◆ NttManager: [DeployWormholeNtt.s.sol](#)
- ◆ AxelarTransceiver: [DeployAxelarTransceiver.sol](#)

Initialized roles

Token

- ◆ Current owner on BNB: [0x3e277051019fDBF6A759ff847D197FE657Ca74fe](#)
- ◆ Future owner on BNB: [0x8E5175D17f74d1D512de59b2f5d5A5d8177A123d](#)

NTT Manager

- ◆ Current owner / pauser on Ethereum: [0x83271E76df1eF8f77487A88fc6aE1478280396bD](#)
- ◆ Future owner on Ethereum: [0x3e40D73EB977Dc6a537aF587D48316feE66E9C8c](#)
- ◆ Future pauser on Ethereum: [0x73b047fe6337183A454c5217241D780a932777bD](#)
- ◆ Current owner / pauser on BNB: [0x3e277051019fDBF6A759ff847D197FE657Ca74fe](#)
- ◆ Future owner on BNB: [0x8E5175D17f74d1D512de59b2f5d5A5d8177A123d](#)
- ◆ Future pauser on BNB: [0xC2b778fCc3FF311Cf1abBF4E53880277bfD14C8f](#)

1.2 CONCLUSION

Based on the analysis of the [proposal](#) and subsequent updates ([Update and Mainnet Configuration](#), [Interim Multisig Update](#), [Additional details on the ownership/pauser transfer](#), [Report Response](#)), and considering the described parameters, deployment process, and contract configuration, the following results were obtained:

- ◆ Network Specific Behavior: **PASS**
- ◆ Scope Checking: **PASS**
- ◆ Audit Report Investigation: **PASS**
- ◆ Deploy Script Check: **PASS**
- ◆ Deployment Verification: **PASS**
- ◆ Initialization Parameters Check: **PASS**
- ◆ Role Model Verification: **PASS**
- ◆ Storage Check: **PASS**
- ◆ Documentation Verification: **PASS**

2 VERIFICATION DETAILS

2.1 NETWORK SPECIFIC BEHAVIOR

Description

The influence of all network features on the protocol's operation was investigated. The virtual machine, the process of message transmission inside and outside the main network, as well as the network operation is compared with networks where the wstETH token is already deployed, are analyzed.

Status: **PASS**

- ◆ `NttManager`, `WormholeTransceiver` contracts and `TransceiverStructs` (used by `NTTManager`) library on the Ethereum and BNB networks have the same compiler version `0.8.19` and EVM version `London`.
- ◆ The `AxelarTransceiver` contract and `TransceiverStructs` (used by `AxelarTransceiver`) libraries for the contract on the Ethereum and BNB networks have the same compiler version `0.8.23` and EVM version `London`.
- ◆ The `wstETH` contract on the Ethereum network uses the default EVM version, while on the BNB network it uses a newer compiler version `0.8.23` and a different EVM version `Paris`.
- ◆ Alignment of the latest BNB and Ethereum network versions ensures the correct operation of all `NttManager` contract functions on both networks. Despite using a newer compiler version and EVM version in the BNB network for the `wstETH` contract, the functionalities remain compatible due to the alignment of base parameters with the Ethereum network.

2.2 SCOPE CHECKING

Description

Verification of the provided scope, study of dependencies, and construction of the protocol architecture. Project documentation is analyzed, tests are run, their coverage and contract models are checked for logical errors, and the architecture is checked for conceptual errors.

Status: **PASS**

The declared scope fully covers all dependent contracts and libraries and corresponds to the described architecture.

2.3 AUDIT REPORT INVESTIGATION

Description

Verification of the presence of an audit report, scope compliance, critical vulnerability remediation (or documented infeasibility), and final commit alignment with recommendations.

Status: **PASS**

- ◆ During the last audit by Cyfrin for `NttManager` and the audit for `WstETH`, no issues requiring corrections were found.
- ◆ The latest audit by Cyfrin for `NttManager` was conducted on the code diff between the previous audited commit and one of the recent commits:

The current diff audit reviewed all changes made to the EVM Solidity contracts from the previously audited commit [f4e2277](#) to the current commit [0d37b0f](#).

This causes confusion, as the findings from the previous report should be included in the new report. Ideally, the diff should have been made from the commit with all corrections following the last audit.

- ◆ The report from Cantina for the `NttManager` contract does not contain information about whether the issues found were fixed, but both Medium Risk issues were resolved as a result of other audits, including those by Cyfrin. However, it should be noted that an additional 18 Low Risk and 9 Informational issues were found, which were not published in the mentioned report:

The present report only outlines the critical, high and medium risk issues.

Descriptions for Low Risk and Informational issues are missing. It is unclear how they were fixed, or if they were fixed at all.

2.4 DEPLOY SCRIPT CHECK

Description

A review of the deployment script for contracts, focusing on verifying initialization parameters. The process ensures that interrupting protocol deployment won't cause incorrect initialization.

Status: **PASS**

During the analysis of the contract deployment process, it was discovered that the deployment script is only available for `NttManager` and `AxelarTransceiver` separately from each other. Analysis of transactions and contract code revealed that some parameters were set manually after deployment.

However, a complete list of all transactions, one after the other, describing the entire deployment and contract configuration process was provided on the forum:

The configuration transactions detailed in *Wormhole x Axelar | Lido Bridge: Implementation for wstETH on BNB Chain - #31* by *Wormhole-Axelar* [post](#) are ordered top to bottom and divided into semantic sections (`NttManager`, `WormholeTransceiver`, `AxelarTransceiver`, `WstEthL2Token`).

The *Wormhole x Axelar | Lido Bridge: Implementation for wstETH on BNB Chain - #41* by *Wormhole-Axelar* [post](#) immediately preceding this one details the ownership/pauser transfer transactions to the publicly verified multisig and the `WstEthL2Token` configuration transactions

2.5 DEPLOYMENT VERIFICATION

Description

Verification of deployed contract bytecode against code in the final commit. All contracts undergo additional verification in a block explorer. A thorough check is performed to ensure that the bytecode of deployed contracts cannot be modified without proper authorization.

Status: **PASS**

- ◆ The bytecode of `NttManager`, `WormholeTransceiver` contracts and `TransceiverStructs` library for `NttManager` were compiled with the `--via-ir` flag, which prevents full-fledged debugging and monitoring of internal transactions through Tenderly and Etherscan.
- ◆ A comparison was made of the bytecode stored in the blockchain and the bytecode of the contracts and libraries in the scope, compiled at the time of the commits, which were taken for audits (during the latest audits, no problems were identified that required corrections). This verification confirmed that the creation bytecode was identical. However, there are some differences in the deployed bytecode due to the use of immutable variables such as `msg.sender` and `address(this)`, which fix their values at the time the contract is deployed.
- ◆ The `AxelarTransceiver` contract uses the `TransceiverStructs` library, but a different version from the one specified for `NttManager`. The library variant for `AxelarTransceiver` was deployed a few weeks ago.

The source code of the libraries differs. The difference lies in the imported `BytesParsing` library. In the case of `NttManager`, the [current version](#) of `BytesParsing` is used, while `AxelarTransceiver` uses an [older version](#).

For example, in the BNB network, these differences can be seen at the following links:

- Library for [NttManager](#) on line 83.
 - Library for [AxelarTransceiver](#) on line 1431.
- Despite this, if the current source code of the library is compiled, the resulting bytecode matches the bytecode of the deployed libraries for `NttManager` and `AxelarTransceiver`, given the following compilation conditions:
- Library for `NttManager` compiled with `solc_version = "0.8.19"` and the flag `--via_ir = true`
 - Library for `AxelarTransceiver` compiled with `solc_version = "0.8.23"` and the flag `--via_ir = false`
- ◆ According to recommendation R-1 in the [guide](#) from the Lido team, it is advised to provide sources for verification in explorers using the standard JSON input format. However, the `AxelarTransceiver` contract uses the flattened format.

- ◆ According to recommendation R-13 in the [guide](#) from the Lido team, contract addresses with the same logic in different networks should differ. However, contracts with identical addresses were found:
 - The `Axe1arTransceiver` contract has the same address in both the Ethereum and BNB chains: `0x723AEAD29acee7E9281C32D11eA4ed0070c41B13`
 - The `TransceiverStruct` contract (used by `NTT Manager` and `Wormhole Transceiver`) has the same address in both the Ethereum and BNB chains: `0xf0396a8077eda579f657B5E6F3c3F5e8EE81972b`

2.6 INITIALIZATION PARAMETERS CHECK

Description

At this stage, values are extracted from storage in verified contracts. Then they are checked for compliance with the parameters specified in the deployment script. Auditors ensure that all contracts are initialized, preventing re-initialization by attackers.

Status: **PASS**

The initialization parameters and their values match those stated in the scope. All initialization parameters are set correctly during the initialization process, and their values correspond to those stated in the deployment documentation.

2.7 ROLE MODEL VERIFICATION

Description

During a protocol review, the access control structure is analyzed to identify redundant roles or roles with broader permissions than necessary. It is also checked that all access rights are in line with the previously defined structure and that the owners of multi-signatures are valid.

Status: **PASS**

In the presented architecture, the owner role was assigned to an interim multisig was created with the participation of Lido contributors: address [0x3e277051019fDBF6A759ff847D197FE657Ca74fe](#) for the BNB network and [0x83271E76df1eF8f77487A88fc6aE1478280396bD](#) for the Ethereum network.

Addresses of the new interim multisig owners (same for Ethereum and BNB networks):

- ◆ [0x4687759DAb0B8319E8dcc59007116b4723838FB1](#) , member from the Axelar Foundation
- ◆ [0xbFf94d4afA68b04532b36ca54A14F3258Ba32a2B](#) , member from xLabs
- ◆ [0x1377C31BB16018e1F0B0C76dF63A6a1a75967AAf](#) , member from the Wormhole Foundation
- ◆ [0x48c2538bBD4E37bFeAe1bA06B97910b9ab473d8D](#) , Lido contributor
- ◆ [0xf16b3C8B2AdF34336DFA2d6853C11Db6790f63F1](#) , Lido contributor

All participants of the new multisig provided verification of signatures on the [forum](#). The provided signatures were independently verified on [etherscan](#) and [shawntabrizi](#).

2.8 STORAGE CHECK

Description

This section verifies the validity of values stored in the project contracts' storage, as well as the transactions in which these values were set. This is done to identify potential vulnerabilities related to access to data or its modification.

Status: **PASS**

During the contract storage check, no vulnerabilities related to access to data or its modification were found. All protocol storage parameters are set correctly during the initialization process, and their values correspond to those stated in the deployment documentation.

2.9 DOCUMENTATION VERIFICATION

Description

Document verification encompasses the analysis of functions and their passed values that directly modify the contract storage.

Status: **PASS**

During the verification of functions and their parameters, it was discovered that the primary sources of documentation include the [official Wormhole docs](#) and the [Github README](#) for the NTT framework. The EVM cross-chain message lifecycle is detailed in the [Message Lifecycle docs](#). Additionally, a [forum](#) provides descriptions for all parameters of project deployment and their values.

3

APPENDIX

3.1 DISCLAIMER

At the request of client, Oxorio consents to the public release of this report. The information contained in this report is provided "as is" without any representations or warranties whatsoever. Oxorio disclaims any responsibility for damages that may arise from or in relation to this verification report. Oxorio retains copyright of this report.

The verification report makes no statements or warranties about the utility of the code, safety of the code, suitability of the business model, investment advice, endorsement of the platform or its products, regulatory regime for the business model, or any other statements about the fitness of the contracts to purpose, or their bug free status.

3.2 ABOUT OXORIO

OXORIO is a blockchain security firm that specializes in smart contracts, zk-SNARK solutions, and security consulting. With a decade of blockchain development and five years in smart contract auditing, our expert team delivers premier security services for projects at any stage of maturity and development.

Since 2021, we've conducted key security audits for notable DeFi projects like Lido, 1Inch, Rarible, and deBridge, prioritizing excellence and long-term client relationships. Our co-founders, recognized by the Ethereum and Web3 Foundations, lead our continuous research to address new threats in the blockchain industry. Committed to the industry's trust and advancement, we contribute significantly to security standards and practices through our research and education work.

Our contacts:

- ◆ oxor.io
- ◆ ping@oxor.io
- ◆ [Github](#)
- ◆ [Linkedin](#)
- ◆ [Twitter](#)

THANK YOU FOR CHOOSING

O X  R I O