Aave

Aave Delivery Infrastructure Contract Review Round 2

Version: 1.1

October, 2023

Contents

Introduction

Sigma Prime was commercially engaged to perform a time-boxed security review of the Aave smart contracts. The review focused solely on the security aspects of the Solidity implementation of the contract, though general recommendations and informational comments are also provided.

Disclaimer

Sigma Prime makes all effort but holds no responsibility for the findings of this security review. Sigma Prime does not provide any guarantees relating to the function of the smart contract. Sigma Prime makes no judgements on, or provides any security review, regarding the underlying business model or the individuals involved in the project.

Document Structure

The first section provides an overview of the functionality of the Aave smart contracts contained within the scope of the security review. A summary followed by a detailed review of the discovered vulnerabilities is then given which assigns each vulnerability a severity rating (see Vulnerability Severity Classification), an *open/closed/resolved* status and a recommendation. Additionally, findings which do not have direct security implications (but are potentially of interest) are marked as *informational*.

Outputs of automated testing that were developed during this assessment are also included for reference (in the Appendix: Test Suite).

The appendix provides additional documentation, including the severity matrix used to classify vulnerabilities within the Aave smart contracts.

Overview

The Aave CrossChain Infrastructure is the new cross-chain communication layer for Aave. It is the system responsible for the communication across different networks for Aave Governance V3. It enables the communication using different bridges that allow the system to receive and forward messages to and from different chains.

The CrossChain Infrastructure has also an emergency mode. If the cross chain communication and execution infrastructure breaks, this mode is triggered and gives the permissions to the *Guardian* to replace bridges and change other configurations.

Security Assessment Summary

This review was conducted on the files hosted on the Aave Delivery Infrastructure repository and were assessed at commit *759e28a*.

A previous security review of the repository had been conducted at commit *a85bc4c*, the findings of the previous review have been shared in a separate report.

Specifically, the files in scope are as follows:

- `BaseCrossChainController.sol`
- `CrossChainController.sol`
- `CrossChainControllerWithEmergencyMode.sol`
- `CrossChainForwarder.sol`
- `CrossChainReceiver.sol`
- `EmergencyConsumer.sol`
- `EmergencyRegistry.sol`
- `ChainIds.sol`
- `EncodingUtils.sol`
- `Erros.sol`
- `SameChainAdapter.sol`
- `BaseAdapter.sol`

Interfaces of the previous contract are included in the scope too.

Note: the OpenZeppelin and Solidity Utils libraries and dependencies were excluded from the scope of this assessment.

The manual code review section of the report is focused on identifying any and all issues/vulnerabilities associated with the business logic implementation of the contracts. This includes their internal interactions, intended functionality and correct implementation with respect to the underlying functionality of the Ethereum Virtual Machine (for example, verifying correct storage/memory layout). Additionally, the manual review process focused on all known Solidity anti-patterns and attack vectors. These include, but are not limited to, the following vectors: re-entrancy, front-running, integer overflow/underflow and correct visibility specifiers. For a more thorough, but non-exhaustive list of examined vectors, see [1, 2].

To support this review, the testing team used the following automated testing tools:

- Mythril: `https://github.com/ConsenSys/mythril`
- Slither: `https://github.com/trailofbits/slither`
- Surya: `https://github.com/ConsenSys/surya`

Output for these automated tools is available upon request.

Findings Summary

The testing team identified no issues during this assessment.

Appendix A Test Suite

A non-exhaustive list of tests were constructed to aid this security review and are provided alongside this document. The `brownie` framework was used to perform these tests and the output is given below.

```
test_constructor                                          PASSED  [2%]
test_get_trusted_remote_by_chain_id                       PASSED  [4%]
test_register_received_message                            PASSED  [6%]
test_register_received_message_delegatecall               PASSED  [8%]
test_basic                                                PASSED  [10%]
test_constructor                                          PASSED  [12%]
test_initialize                                           PASSED  [14%]
test_forward_message                                      PASSED  [16%]
test_forward_message_no_bridge_adapter                    PASSED  [18%]
test_forward_message_wrong_caller                         PASSED  [20%]
test_retry_envelope                                       PASSED  [22%]
test_retry_envelope_non_registered_envelope               PASSED  [24%]
test_retry_envelope_no_bridge                             PASSED  [26%]
test_retry_transaction                                    PASSED  [28%]
test_retry_transaction_no_bridge                          PASSED  [30%]
test_retry_transaction_not_prev_forwarded_transaction     PASSED  [32%]
test_retry_transaction_use_the_same_bridge_twice          PASSED  [34%]
test_approve_senders                                      PASSED  [36%]
test_remove_senders                                       PASSED  [38%]
test_enable_bridge_adapters                               PASSED  [40%]
test_enable_bridge_adapters_zero_address                  PASSED  [42%]
test_disable_bridge_adapters                              PASSED  [44%]
test_receive_cross_chain_message_case_1                   PASSED  [46%]
test_receive_cross_chain_message_case_2                   PASSED  [48%]
test_receive_cross_chain_message_case_3                   PASSED  [51%]
test_deliver_envelope                                     PASSED  [53%]
test_deliver_envelope_invalid_state                       PASSED  [55%]
test_allow_receiver_bridge_adapters                       PASSED  [57%]
test_allow_receiver_bridge_adapters_same_adapter_same_chain  PASSED  [59%]
test_allow_receiver_bridge_adapters_zero_address_bridge   PASSED  [61%]
test_disable_receiver_bridge_adapters                     PASSED  [63%]
test_update_confirmations                                 PASSED  [65%]
test_update_confirmations_incorrect_nb_confirmation       PASSED  [67%]
test_update_messages_validity_timestamp                   PASSED  [69%]
test_update_messages_validity_timestamp_invalid_timestamp  PASSED  [71%]
test_update_cl_emergency_oracle                           PASSED  [73%]
test_update_cl_emergency_oracle_not_owner                 PASSED  [75%]
test_update_cl_emergency_oracle_zero_address              PASSED  [77%]
test_solve_emergency                                      PASSED  [79%]
test_solve_emergency_not_in_emergency                     PASSED  [81%]
test_emergency_token_transfer                             PASSED  [83%]
test_emergency_ether_transfer                             PASSED  [85%]
test_set_emergency                                        PASSED  [87%]
test_set_emergency_same_chain                             PASSED  [89%]
test_set_emergency_wrong_caller                           PASSED  [91%]
test_forward_message                                      PASSED  [93%]
test_native_to_infra_chain_id                             PASSED  [95%]
test_infra_to_native_chain_id                             PASSED  [97%]
test_get_trusted_remote_by_chain_id                       PASSED  [100%]
```

Appendix B Vulnerability Severity Classification

This security review classifies vulnerabilities based on their potential impact and likelihood of occurance. The total severity of a vulnerability is derived from these two metrics based on the following matrix.

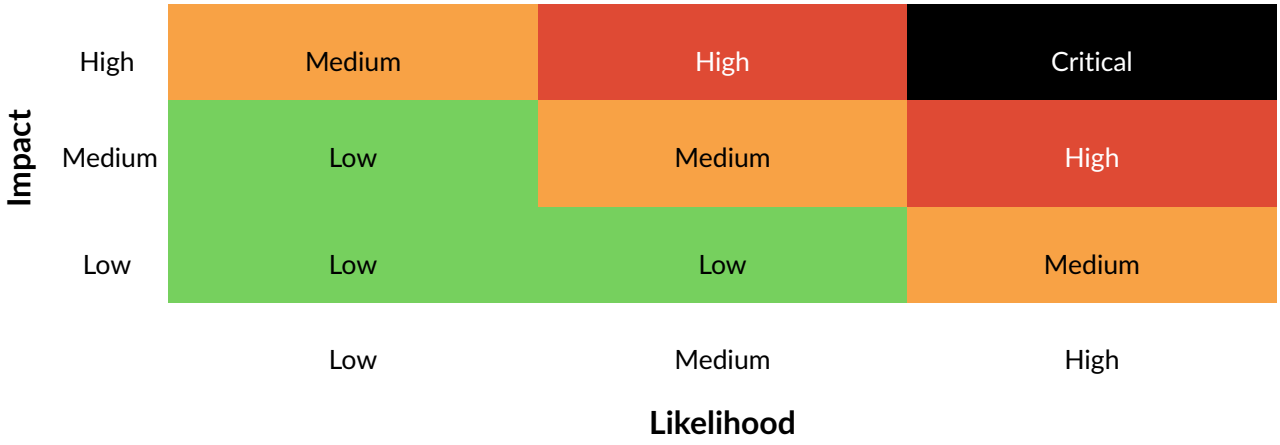

Table 1: Severity Matrix - How the severity of a vulnerability is given based on the *impact* and the *likelihood* of a vulnerability.

References

[1] Sigma Prime. Solidity Security. Blog, 2018, Available: https://blog.sigmaprime.io/solidity-security.html. [Accessed 2018].

[2] NCC Group. DASP - Top 10. Website, 2018, Available: http://www.dasp.co/. [Accessed 2018].