Aave V3 sAVAX Verification and Listing Stewards Audit

Scope

The scope of the assessment is the contracts `StakedAvax.sol` (`sAVAX`) which was formally verified, and `sAVAXOracleAdapter.sol` and `AaveV3SAVAXListingSteward.sol` which were manually audited due to the fact that the implementation involves only calls to function with concrete values. The latter two contracts define the `sAvax` configuration on the Aave V3 platform on Avalanche. These contracts had 2 security engineers and 1 security researcher reviewing the code in detail.

The verification of the token was finished on the 28th of June, reviewing the deployed contract on Avalanche.

The audit was finished on the 15th of June, reviewing commit 1d00da87 of the `sAVAX` listing steward.

Contracts Overview

As part of our continuous formal verification for Aave, we've inspected the `sAVAX` token for security issues and non-trivial features. You can see the results in our Aave dashboard.

The audited contracts' purpose is configuring `sAvax` as a collateral token on the Aave V3 platform on the Avalanche network.

Two contracts were audited:

1. `sAVAXOracleAdapter.sol`, which implements an `sAvax` to USD oracle adapter using two components: 1.1. Avax to USD aggregator - see in code.

   1.2. sAvax-Avax pool ratio (using the original `sAvax` pool contract) - see in code.

2. `AaveV3SAVAXListingSteward.sol`, which configures `sAvax` as a traded token on the AAVE V3 platform. The contract contains 3 steps:

   2.1. Setting a price feed on the AAVE oracle for the `sAvax` token - see in code.

   2.2. Creating a new E-mode category on Aave V3 protocol for `sAvax` and `wAvax` - see in code.

   2.3. Listing the token onto Aave V3 protocol and configuring it as a collateral asset - see in code.

Audit Goals

During the review of the code, the following checks have been performed:

sAVAXOracleAdapter

1. All addresses of external contracts that are used match the existing contracts on the Avalanche network.
2. The `lastAnswer()` method returns a correct representation of 1 `sAvax` in USD, i.e. the formula returns USD/sAvax.
3. The current order of arithmetic operations in `lastAnswer()` preserves the precision of the oracle's decimals.
4. Examining the possibility of price manipulation on the oracle, bearing in mind the xSushi attack. A short description of the attack can be also found in the appendix.

AaveV3SAVAXListingSteward

1. All addresses of external contracts used match the existing contracts on the Avalanche network.

E-mode Category Checks

2. The ID of the new e-mode category is unique, i.e. it is not currently in use.
3. Decimals of the e-mode category parameters match the Aave standard for those parameters.
4. Which additional tokens are added to the same e-mode category as `sAvax`.
5. The e-mode parameters' values are properly configured, i.e. better loan conditions than regular loans, and proper relations between the LTV, Threshold, and Liquidation bonus.

Correct Setting of Parameters and Values

6. The asset is being added to the system correctly with all the correct `InitReserveInput` struct values. Additional parameters are passed to the correct methods and with the right decimals, e.g. `SUPPLY_CAP`, `RESERVE_FACTOR`, and `LIQ_PROTOCOL_FEE`.
7. The asset is configured as collateral with proper relations between the LTV, threshold, and liquidation bonus.

Privileges

8. `AaveV3SAVAXListingSteward` has the necessary roles to execute `listAssetAddingOracle()` without reverting.
9. The roles are renounced from the contract at the end of execution.

Related Checks

In the recent past, Certora has verified the original `stakedAvax` contract deployed by Benqi onto the Avalanche network. In this verification process, we verified several properties, including - `totalSupply` does indeed correctly track the sum of shares in the pool, the preservation of correlation between Avax and `sAvax`, and more. You can read more about the Benqi report here.

Findings And Recommendations

Severity: Low

Issue:	Missing E-mode configuration of `wAvax`.
Description:	In `AaveV3SAVAXListingSteward`, `wAvax` is absent from the configuration of the same e-mode category as `sAvax`. It is expected that pegged tokens will be in the same e-mode category so they could be traded with improved conditions.
Aave Response:	This issue was fixed in commit ff957ede.

Recommendation

Issue:	Verify that `AaveV3SAVAXListingSteward` has the necessary privileges.

Issue:	Verify that `AaveV3SAVAXListingSteward` has the necessary privileges.
Description:	Many function calls in the listing process require the listing contract to have both `Asset Listing Admin` and `Risk Admin` roles in order to be successfully executed. It is important to remember granting those roles prior to calling `listAssetAddingOracle()`.
Recommendation:	Add a dedicated require condition at the beginning of `listAssetAddingOracle()` in order to give a clearer error message in case of such a revert.

Informational - Non-Standard Behavior:

1. The token is pausable.
2. The token is mintable

You can see the full result in our AAVE dashboard.

Conclusions

sAVAXOracleAdapter

1. All addresses specified in the contract match existing relevant contracts on the Avalanche blockchain.
2. The units of measure in which `lastAnswer()` returns the data is indeed `USD/sAvax`. The price feed returns an answer in `USD/AVAX` units, and `getPooledAvaxByShares` returns the ratio `AVAX/sAVAX`.
3. The order of operations preserves the precision of the oracle. By multiplying first and dividing the result, the precision is guaranteed to be preserved.
4. The `sAVAXOracleAdapter` was inspected bearing in mind the XSushi attack from November 2021. In the case of the `sAvax` oracle, the ratio of sAvax-Avax is being taken directly from the `sAvax` contract, where the calculation is done correctly, i.e. by tracking the amount of deposited Avax tokens, as opposed to the balance of Avax held by the contract. Moreover, the exchange rate that the oracle retrieves is given in USD/sAvax, which is the intended unit, and the precision of the Avax-USD oracle is not compromised as a result of the additional calculation in the adapter.

AaveV3SAVAXListingSteward

1. All addresses specified in the contract match existing relevant contracts on the Avalanche blockchain.

E-mode Category Checks

2. E-mode category id 2 is indeed unique and doesn't exist in the platform. Currently, the only existing e-mode category is id 1 which represents stablecoins.
3. The assigned decimals in the contract match the Aave standard.
4. `sAVAX` was configured as the only asset in this category. More on that can be read in the above issue "Missing E-mode configuration of wAvax".
5. The relations between the LTV, Threshold, and Liquidation bonus were configured reasonably, and the e-mode configuration is indeed improved compared to the regular listings.

Correct Setting of Parameters and Values

6. The asset was added to the system correctly with all the correct `InitReserveInput` struct values, and all parameters are passed to the correct methods and with the right decimals.
7. The asset was configured as collateral with proper relations between the LTV, threshold, and liquidation bonus.

Privileges

8. The roles of the contract are given externally. Therefore, the executor should delegate the necessary privileges before trying to execute.
9. At the end of the process the contract renounces its privileges in a correct manner.

Disclaimer

Appendix

xSushi Attack Description

In the xSushi attack, the ratio between the two tokens - Sushi and xSushi, was calculated as the balance of Sushi that the xSushi contract holds, divided by the total supply of xSushi.

$$Sushi - xSushi\ ratio = \frac{Sushi.balanceOf(xSushi)}{xSushi.totalSupply()}$$

This calculation left a gap for price manipulation where Sushi could have been sent directly to the xSushi contract without minting any xSushi, thereby increasing the value of xSushi according to the Aave oracle adapter.