SHERLOCK

SHERLOCK SECURITY REVIEW FOR

Prepared for:	Avail
Prepared by:	Sherlock
Lead Security Expert:	0x52
Dates Audited:	January 19 - January 22, 2024
Prepared on:	February 20, 2024

Introduction

The essential base layer for modern blockchains.

Scope

Repository: availproject/contracts

Branch: main

Commit: ce7d2408158d8d74fc3bbf6c1dbe48e675e36579

For the detailed scope, see the contest details.

Findings

Each issue has an assigned severity:

- Medium issues are security vulnerabilities that may not be directly exploitable or may require certain conditions in order to be exploited. All major issues should be addressed.
- High issues are directly exploitable security vulnerabilities that need to be fixed.

Issues found

Medium	High
1	0

Issues not fixed or acknowledged

Medium	High
0	0

Issue M-1: Very large sends from AvailBridge will break receiving bridge and cause loss of funds

Source: https://github.com/sherlock-audit/2023-12-avail-judging/issues/85

Found by

Summary

When sending tokens, up to uint256.max can be sent by AvailBridge, however lib.rs can only receive amounts up to uint128.max. Any values over uint128.max will cause the AVAIL side of the bridge to panic and deposit will be permanently lost.

Vulnerability Detail

lib.rs#L394-L399

```
T::Currency::transfer(
        &Self::account_id(),
        &destination_account_id,
        amount.as_u128().saturated_into(),
        ExistenceRequirement::AllowDeath,
)?;
```

When receiving a token transfer, lib.rs attempts to convert the amount from uint256 to uint128 via `as_u128()`. When amount is more than uint128.max this will panic and prevent processing of the transaction. Since the bridge doesn't have any way handle failed transactions, the deposit will be permanently lost.

Impact

Large deposits will break the receiving end of the bridge and cause loss of funds

Code Snippet

AvailBridge.sol#L383-L411

Tool used

Manual Review

Recommendation

Cause all `send` functions to revert for amounts that are greater than uint128.max.

Discussion

QEDK

We primarily use `uint256` because it's cheaper on EVM, any `uint128` transfers are highly unlikely because the total supply of all tokens is much lower than that. For e.g. taking the token with the highest supply today: `((1.774455 * (10 ** 17)) * (10 ** 18))  < type(uint128).max` would return `true`.

sherlock-admin

2 comment(s) were left on this issue during the judging contest.

tsvetanovv commented:

> Low. The chance of someone sending tokens for such a large value is small

takarez commented:

> invalid because {invalid: very unlikely to happen}

IAm0x52

Escalate

Protocol is designed to work with arbitrary ERC20 token. I've clearly shown an edge case to be addressed that causes loss of funds.

sherlock-admin

The protocol team fixed this issue in PR/commit https://github.com/availproject/contracts/pull/3.

sherlock-admin

The Lead Senior Watson signed-off on the fix.

Disclaimers

Sherlock does not provide guarantees nor warranties relating to the security of the project.

Usage of all smart contract software is at the respective users' sole risk and is the users' responsibility.