G0 Group

Security Review of
Hashi

June 27, 2024

Hashi / June 2024

Files in scope Hashi

Listed files in https://github.com/gnosis/hashi/tree/6f5bf9e15e37901965c169e40a7ef907f9019eca/packages/evm/contracts

```
Hashi.sol
Yaho.sol
Yaru.sol
utils/
    HeaderStorage.sol
    MessageHashCalculator.sol
    MessageIdCalculator.sol
ownable/
    ShoyuBashi.sol
    ShuSo.sol
```

Files in scope AMB integration

Listed files in https://github.com/crosschain-alliance/tokenbridge-contracts/tree/0cef7054be1be91203b09333aac8f7621b07afd5/contracts

```
upgradeable_contracts/
    MessageRelay.sol
    VersionableBridge.sol
    BasicBridge.sol
    Upgradeable.sol
    Validatable.sol
    Ownable.sol
    Claimable.sol
    DecimalShiftBridge.sol
    ValidatorStorage.sol
    arbitrary_message/
        ForeignAMB.sol
        HomeAMB.sol
        BasicForeignAMB.sol
        BasicHomeAMB.sol
        BasicAMB.sol
        AsyncInformationProcessor.sol
        ForeignAMBWithGasToken.sol
        MessageDelivery.sol
        MessageProcessor.sol
        VersionableAMB.sol
upgradeability/
    EternalSotrage.sol
libraries/
    Message.sol
    ArbitraryMessage.sol
```

Files in scope xDAI integration

Listed files in https://github.com/crosschain-alliance/tokenbridge-contracts/tree/7e60b0c46e168d1b73e766877c0d24cedbad6db6/contracts

```
upgradeable_contracts/
    ValidatorsFeeManager.sol
    BlockRewardBridge.sol
    BaseFeeManager.sol
    FeeTypes.sol
    ValidatorStorage.sol
    BlockRewardFeeManager.sol
    ERC20Bridge.sol
    BasicForeignBridge.sol
    Validatable.sol
    BasicBridge.sol
    BasicTokenBridge.sol
    Ownable.sol
    DecimalShiftBridge.sol
    InitializableBridge.sol
    Initializable.sol
    Claimable.sol
    VersionableBridge.sol
    Upgradeable.sol
    OtherSideBridgeStorage.sol
    BasicHomeBridge.sol
    HomeOverdrawManagement.sol
    BaseOverdrawManagement.sol
    RewardableBridge.sol
    GSNForeignERC20Bridge.sol
    erc20_to_native/
        FeeManagerErcToNative.sol
        FeeManagerErcToNativePOSDAO.sol
        ForeignBridgeErcToNative.sol
        HomeBridgeErcToNative.sol
        RewardableHomeBridgeErcToNative.sol
        XDaiForeignBridge.sol
        SavingsDaiConnector.sol
        InterestConnector.sol
upgradeability/
    EternalSotrage.sol
gsn/
    BaseRelayRecipient.sol
```

Current status

All reported issues have been either fixed or acknowledged by the developer.

Issues

1. Any single validator can DoS BasicHomeBridge using HomeOverdrawManagement

type: security / severity: medium

There’s an issue with integration of `HomeOverdrawManagement` with `BasicHomeBridge.executeAffirmat` through `HomeBridgeErcToNative.onFailedAffirmat`. Firstly `onFailedAffirmatio` is called whether there’s enough affirmations or not when the limit is potentially passed, this means: any validator can brick the `HomeOverdrawManagement` by submitting a very large nonsense transfer which will cause `outOfLimitAmount` to be `maxuint`. Second, any validator can “revive” old transfers in `HomeOverdrawManagement` that will not be able to be cleared by `HomeOverdrawManagement.fixAssetsAboveLi` because of the `isAlreadyProcessed` check, this might lead to the contract being permanently (until contract is fixed by an upgrade) bricked if there’s enough transactions in the history to reach the `uint` limit (probably not a practical attack in most cases).

status - acknowledged

The issue has been acknowledged, in an eventuality of an exploit, the validator will be removed and contract rescued through contract upgrade mechanism.

2. Checks-effects-interactions pattern not respected in _emitUserRequestForSignatureMaybeRelayDataWithHashiAndIncreaseNonce

type: code fragility / severity: minor

Updating the nonce after the hashi call in `BasicHomeBridge._emitUserRequestForSignatureMaybeRelayDataWithHashiAndIncrea` and `BasicForeignBridge._emitUserRequestForAffirmationMaybeRelayDataWithHashiAndIncrea` is potentially problematic. While re-entrancy is not possible with the audited version of the `Yaho` contract, it could become possible in the future, and then it would enable an attacker to submit multiple transfers with the same nonce.

status - fixed

The issue has been fixed and is no longer present inhttps://github.com/crosschain-alliance/tokenbridge-contracts/tree/fb6bae7589a102613b48c12addb425b72836574e/contracts

3. Fee recalculation in HomeBridgeErcToNative.onSignaturesCollected might break accounting on configuration change

type: code fragility / severity: medium

The fact fee is recaculated in `HomeBridgeErcToNative.onSignaturesColle` means that if fee configuration changes while a transaction is pending it might lead to a different value being paid out than has been collected in `nativeTransfer`.

status - acknowledged

The issue has been acknowledged.

4. In case of an overdraw refund, fee …

type: implementation / severity: medium

If `FeeManagerErcToNative` is used, when the refund transfer is generated in `HomeOverdrawManagement.fixAssetsAboveLimits` and then fee from the refund transfer is distributed in `HomeBridgeErcToNative.onSignaturesCollected` the contract will lack funds to pay out the fee since the tokens haven’t been minted yet.

status - acknowledged

The issue has been acknowledged. It’s being avoided in production through configuration.

5. Unnecessary storage write in Yaho

type: optimization / severity: minor

In `Yaho.dispatchMessageToAdapters` and `Yaho.dispatchMessagesToAdapters`, `_pendingMessageHashes[messageId]` is set, only to be immediately deleted again leading to a waste of gas.

status - fixed

The issue has been fixed and is no longer present in https://github.com/gnosis/hashi/tree/e3fe9cfaa89ef0607d7bce0ef16b787870162d6a/packages/evm/contracts

6. Only message hashes instead of whole messages could be submitted to Yaho

type: optimization / severity: minor

Gas could be saved by sending over just message hashes instead of whole messages, since in `onMessage` methods of bridges the message is immediately hashed anyway.

status - acknowledged

The issue has been acknowledged

7. GSNForeignERC20Bridge.executeSignaturesGSN is missing hashi integration

type: implementation / severity: note

Hashi is not integrated in `GSNForeignERC20Bridge.executeSignaturesGSN`

status - acknowledged

Developer’s response: The `GSNForeignERC20Bridge.executeSignaturesGSN` method is no longer in use so doesn’t require Hashi integration.

8. Checks-effects-interactions pattern not respected in XDaiForeignBridge.onExecuteMessageGSN

type: code fragility / severity: minor

In `XDaiForeignBridge.onExecuteMessage` `ensureEnoughTokens` is called before the `super.onExecuteMessageGSN` is called. The issue is that `ensureEnoughTokens` contains an external call, one that likely doesn't allow re-entrancy right now, but could at some point in the future and if that becomes the case, this will allow the max executed per day check to be bypassed, because the state update happens after the external call.

status - acknowledged

Developer's response: GSN is not used anymore.

9. Hashi confirmation requirement blocks collection of affirmation signatures

type: implementation / severity: note

In `executeAffirmation` functions, the fact hashi check is made brefore the required amount of affirmations has been collected means that no signatures can be collected before the Hashi confirmation is processed.

status - acknowledged

Developer's response: It's a breaking change, and we are aware of it. Even if we put the hashi check only after having checked the required amount of affirmation, it would mean that the validator responsible for triggering the message execution would need to wait for the Hashi confirmation. This would imply that the validator triggering `handleMessage` would need to know it is triggering it and that Hashi has confirmed the message, introducing more complexity.

10. Validators can call executeAffirmation in place of confirmInformation

type: security / severity: note

A validator can call `BasicHomeAMB.executeAffirmation` instead of `AsyncInformationProcessor.confirmInformation` to add affirmation to the `AsyncInformationProcessor.confirmInformation` call. This probably doesn't break anything, but a wrong event will be emitted.

status - acknowledged

The issue has been acknowledged

11. In AMB Hashi integration, validators can execute arbitrary messages

type: security / severity: critical

The hashi implementation in AMB only transmits `msgId`, but that doesn't encode the actual content of the message, so there's no mechanism to ensure the correct message is executed by validators.

status - fixed

The issue has been fixed and is no longer present inhttps://github.com/crosschain-alliance/tokenbridge-contracts/tree/14e330ed2705539147b3bb7ac106589a8ae1473c/contracts

12. In GasTokenConnector, transactions can be unnecessarily rejected

type: implementation / severity: minor

In `GasTokenConnector._collectGasTok` the amount of gas tokens that can be transferred (instead of minted) from the sender is derived from allowance, but in the ideal case it would be `min(allowance, balance`, to allow the call to succeed even in cases where balance is lower than allowance and the gas token target mint value.

status - acknowledged

The issue has been acknowledged. The contract is not currently used in production.