ClarityAlliance

PONTIS BRIDGE SECURITY REVIEW

Conducted by:
KRISTIAN APOSTOLOV, 0X3B AND ABA

JUNE 24TH, 2024

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

1. About Clarity Alliance

Clarity Alliance is a team of expert whitehat hackers specialising in securing protocols on Stacks.

They have disclosed vulnerabilities that have saved millions in live TVL and conducted thorough reviews for some of the largest projects across the Stacks ecosystem.

Learn more about Clarity Alliance at clarityalliance.org.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

2. Disclaimer

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts Clarity Alliance to perform a security assessment.

This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. Clarity Alliance’s position is that each company and individual are responsible for their own due diligence and continuous security. Clarity Alliance’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by Clarity Alliance are subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis.

Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third parties. Notice that smart contracts deployed on the blockchain are not resistant from internal/external exploit. Notice that active smart contract owner privileges constitute an elevated impact to any smart contract’s safety and security. Therefore, Clarity Alliance does not guarantee the explicit security of the audited smart contract, regardless of the verdict.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

3. Introduction

A time-boxed security review of Pontis Bridge, where Clarity Alliance reviewed the scope, whilst simultaneously building out a testing suite for the protocol.

4. About Pontis Bridge

The Pontis Bridge system is designed to enhance the Bitcoin DeFi experience by allowing users to leverage native BTC and Bitcoin-issued assets like Runes or Ordinals to engage with smart contracts on other Layer 2 networks.

To ensure the security and decentralization of the bridge, multiple nodes from various reputable projects on Stacks and Bitcoin will be established. This approach prevents any single project or entity from having undue control over user funds.

On the Bitcoin network, users interact with Pontis multisignature wallets to deposit their assets for bridging. On Layer 2 networks like Stacks, they receive the corresponding assets.

5. Risk Classification

Severity	Impact: High	Impact: Medium	Impact: Low
Likelihood: High	Critical	High	Medium
Likelihood: Medium	High	Medium	Low
Likelihood: Low	Medium	Low	Low

5.1 Impact

- High - leads to a significant material loss of assets in the protocol or significantly harms a group of users.
- Medium - only a small amount of funds can be lost (such as leakage of value) or a core functionality of the protocol is affected.
- Low - can lead to any kind of unexpected behavior with some of the protocol's functionalities that's not so critical.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

5.2 Likelihood

- High - attack path is possible with reasonable assumptions that mimic on-chain conditions, and the cost of the attack is relatively low compared to the amount of funds that can be stolen or lost.

- Medium - only a conditionally incentivized attack vector, but still relatively likely.

- Low - has too many or too unlikely assumptions or requires a significant stake by the attacker with little or no incentive.

5.3 Action required for severity levels

- Critical - Must fix as soon as possible (if already deployed)
- High - Must fix (before deployment if not already deployed)
- Medium - Should fix
- Low - Could fix

6. Security Assessment Summary

The **Pontis Bridge Contracts** that were audited consist of the Stacks Bitcoin L2 component that will facilitate the Bitcoin L1 to Stacks L2 bridging. BTC 1:1, ordinal and runes transfers were implemented and audited.

Note, the entire bridging system contains several other off-chain components that were not part of the current audit.

Review Commit Hash:
7cb0e82b92d7838da4ade13caf25cc30af82b5dd

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

7. Executive Summary

Over the course of the security review, Kristian Apostolov engaged with Pontis to review Pontis Bridge. In this period of time a total of **19** issues were uncovered.

Protocol Summary

Protocol Name	Pontis Bridge
Repository	https://github.com/Pontis-Labs/bridge-contracts
Date	June 24th, 2024
Protocol Type	Bridge

Findings Count

Severity	Amount
Critical	1
High	3
Medium	2
Low	4
QA	9
Total Findings	**19**

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

Summary of Findings

ID	Title	Severity	Status
[C-01]	User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism	Critical	Resolved
[H-01]	Ordinal Stacks to Bitcoin Bridging Can Be Corrupted	High	Resolved
[H-02]	mint Attempts to Mint Duplicate IDs	High	Resolved
[H-03]	Incorrect Fee Calculations	High	Resolved
[M-01]	Bitcoin Ordinal NFT Collections Are Jumbled on Stacks	Medium	Resolved
[M-02]	Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs	Medium	Resolved
[L-01]	Bridge Token Name and Symbol Should Not Be Changeable	Low	Resolved
[L-02]	MAX-BTC-BRIDGE-MINIMUM is set to the wrong value	Low	Resolved
[L-03]	Pausing Stacks Pegging-In May Cause System Problems	Low	Resolved
[L-04]	Implement 2-Step Ownable	Low	Resolved
[QA-01]	Duplicated Ownable Trait Operator File	QA	Resolved
[QA-02]	Use Errors Instead of Panicking	QA	Resolved
[QA-03]	Redundant or Dead Code	QA	Resolved
[QA-04]	Reverse Parameters for Consistency	QA	Resolved
[QA-05]	Group UTXO Operations to Reduce Code Size	QA	Resolved
[QA-06]	Principal Instance Cast as a Trait	QA	Resolved
[QA-07]	Isolate Expression into a Function	QA	Resolved
[QA-08]	Confusing Error Messages	QA	Resolved
[QA-09]	Isolate Iterator Expression into a Function	QA	Resolved

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

8. Findings

8.1. High Findings

[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism

Description

When a user wants to bridge assets from Bitcoin to Stacks, they send their assets to the bridge wallet on the Bitcoin blockchain. After node validation, the Pontis bridge listener initiates the Stacks mint transaction via the `mint-*` functions of the `pontis-bridge-v1` contract.

All three existing functions `mint-btc` , `mint-runes-batch` and `mint-ordinals-batch` are called with `tx-hash(es)-and-vout` (Bitcoin transaction `TXID:VOUT` pairing). This input is then checked to ensure it is new on Stacks; otherwise, the transaction reverts.

```
(try! (check-if-exists-and-mark tx-hash-and-vout))
```

This mechanism exists to prevent the listener from resubmitting a bridge in case of a malfunction that restarted it without the last processed bridging being marked as such. When a user wants to bridge out from Stacks to the Bitcoin blockchain, they call the `pontis-bridge-v1` contract `peg-out*` functions.

Of particular interest are the `peg-out-runes` and `peg-out-btc` functions. Both of these functions accept a `tx-hash-and-vout` argument which is then checked and marked if it has never been seen before:

```
(try! (check-if-exists-and-mark tx-hash-and-vout))
```

For pegging out, this operation is user-initiated, meaning the `tx-hash-and-vout` value is user-controlled. A successful call to the `peg-out*` functions alters the user's balance so there is no possibility of a redo making the `check-if-exists-and-mark` mechanism redundant here.

However, because it is present and because it saves the user-provided `tx-hash-and-vout` value in the same mapping that is then checked during bridging in, an attacker may observe newly-added bridging on the Bitcoin side and front-run the bridging transactions on Stacks (the `mint-*` calls) with a call to the `peg-out-runes` or `peg-out-btc` functions using the pending `tx-hash-and-vout` from the minting calls.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

By doing this, the Stacks mints will fail with the `ERR-HASH-EXISTS` error code, which will be interpreted by the listener as if the bridging has already been executed and will not be retried. Thus, the user's bridged assets will be stuck, stranded in the Bitcoin bridging-in wallet, requiring manual intervention to unstick.

The `peg-out*` functions have a minimum amount to be used, which makes the availability of this attack as a DOS less attractive. However, it still leaves blocking user funds as a viable benefit with minimal cost (bridging out fees) to the attacker.

Recommendation

Users should not be allowed to input arbitrary UTXOs regardless of the implemented solution. After discussions with the developer, the final recommended solution was:

- store a list of available UTXOs on-chain
- an off-chain system would continuously populate it
- all the peg-out functions would incrementally get the next available UTXOs

This solution results in no reliance on user input, no possibility of front-running other peg-out operations, and the trade-off is a slight gas cost per each UTXO batch upload by the project team and the need to continuously update the on-chain list.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

8.2. High Findings

[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted

Description

When a user initiates an ordinal NFT bridging out (peg-out) from Stacks to the Bitcoin blockchain, they are required to call the `peg-out-ordinals-batch` function from the `pontis-bridge-v1` contract. This function requires a list of inscription IDs, recipients, and UTXOs to be used as anchors on the Bitcoin side.

Pontis bridging-out implements a logic that utilizes the provided UTXOs as utility 546 satoshi UTXOs, which will be sent from the Bitcoin multisig wallet back to itself.

If issues arise in the off-chain software that initiates the bridging, the information that a UTXO was targeted on Stacks bridging-out is available (provided as input to the `peg-out-ordinals-batch` function by the user) and the confirmation if it was executed if it was sent on the Bitcoin blockchain from the multisig to itself The mechanism effectively leverages the on-chain storage system of two different blockchains for error handling.

Due to this system, and the requirement for a user to input the UTXO, an attacker can front-run any other user's ordinal peg-out with their own while using the same UTXOs. Since there is no on-chain validation in `peg-out-ordinals-batch` to check if a UTXO is already used, both peg-outs will succeed.

The listener software would then face a situation where it executes the first bridging out, but for the second, it will internally determine that the peg-out has already happened. In the best-case scenario, a user's peg-out will be blocked and require manual intervention by the team to complete it on the Bitcoin side.

This front-running behavior can also be used to corrupt the pegging out of runes and BTC by providing a UTXO that is stored in the `available-peg-out-key-utxo` mapping.

By doing so, the listener software would use the UTXOs from the map with the attacker's transactions while not noting them on-chain. Subsequent rune and BTC peg-outs would be successful on-chain, but the listener would see the UTXOs as already used, marking them as complete.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

Recommendation

Use the existing on-chain pending UTXO logic, which is already implemented and used by the `peg-out-runes` and `peg-out-btc` functions, to avoid any other issues. This includes users simply providing an invalid UTXO to the peg- out function (those that call the function directly and do not go through the dApp).

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

[H-02] `mint` Attempts to Mint Duplicate IDs

Description

The `mint` function uses `last-id` to mint NFTs. However, if our `inscription-hash-to-id` is smaller than the `last-id`, the contract will attempt to mint `last-id` instead of the correct `id`. This results in a transaction revert since `last-id` is already minted.

```
(nft-mint? wrapper (var-get last-id) recipient)
```

Example

NFTs minted: 1, 2, 3, 5 (minted with `inscription-hash-to-id`)

1. Bob tries to mint an NFT with `inscription-hash-to-id` of 4.
2. `id` is set to 4.
3. `is-new-id` is false as `4 > 5 → false`.
4. The contract calls `mint` on the wrapper with `last-id` 5 instead of `id` 4.
5. The transaction reverts because it tries to mint NFT 5 again.

Recommendation

Mint `id` instead of `last-id`.

```
- (nft-mint? wrapper (var-get last-id) recipient)
+ (nft-mint? wrapper id recipient)
```

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

[H-03] Incorrect fee calculations

Description

'Percent' fees are withheld on each BTC or Rune peg out. As suggested by their name, they are intended to subtract a certain percentage from the amount being pegged-out. The issue lies within how they get calculated in `pontis-bridge-fee-manager` .

```
(define-read-only (calculate-btc-percent-fee (amount uint))
(/ (* amount (var-get btc-percent-fee)) MAX-PERCENT-FEE)
)
```

```
(define-read-only (calculate-btc-percent-fee (amount uint))
(/ (* amount (var-get runes-percent-fee)) MAX-PERCENT-FEE)
)
```

The logic does the following: `(amount * fee) / max_fee` i.e., given the following numbers:

```
amount = 100
fee = 0.01 # 1%
max_fee = 0.01 # also 1%
```

We would yield `100` , which is 100x more than intended and thus incorrect.

The above is due to `MAX-PERCENT-FEE` being used as a divisor.

Recommendation

Consider implementing the following formula instead:

```
(amount * fee) / 1e8 # 100%
```

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

8.3. Medium Findings

[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks

Description

When bridging Bitcoin ordinal NFTs to Stacks, all the newly-minted NFTs on Stacks are attributed to the same wrapper contract, `pontis-bridge-ordinals-nft` . The IDs for the Stacks wrappers also have an incremental value, regardless of the ordinal collection.

Example: Stacks wrapper ID `#1` can be `Bitcoin AwesomeCollection #946` and Stacks wrapper ID `#2` can be `Ordinal NiceCollection #3735` .

This model has several issues:

1. Stacks NFT marketplaces will not be able to have separate collections for each ordinal collection without a major custom implementation specifically for the bridge contract. This is unlikely to happen since the sparse IDs would require a custom mapping for each collection. Even if the marketplaces do implement such a mechanism, the mapping itself must be provided and maintained by the Pontis team for each new ordinal that bridges.
2. The metadata URL is set for all Ordinals on the same contract. To be defined as a bridged NFT, the metadata for each NFT would need to be the same (at least visually) as the original versions. Since the Pontis team is the only one capable of setting the metadata, it will require collecting and storing all the metadata for each ordinal collection it bridges. This adds severe overhead and makes it impossible to use permanent storage like IPFS since the team would need to continuously update the metadata.
3. Adoption is slowed as Stacks integrators are less likely to interact with the NFTs due to the lack of a simple on-chain ordinal collection-holder identification solution. For example, an integrator decides to give a discount on a protocol if users hold a specific ordinal collection. They would need to store on-chain the IDs specific to each migrated Ordinal NFT after they collect it from the bridge, instead of a simple contract holder check. Off-chain, they can use the `pontis-bridge-ordinals-nft.get-hash-from-id` function to determine this, followed by ordinal collection lookup.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

Recommendation

Similar to how Runes are migrated (each on its own FT contract), have ordinals minted each to a separate contract with an Ordinal collection ID to ID minting. Having it like this does not invalidate the SIP-09 standard as there is no requirement for no gaps or the ID logic to be incremental.

This way, each contract has its own metadata and can be easily managed by other teams. Teams that want to have their Ordinals bridged will use a contract provided by the Pontis team, similar to how `pontis-bridge-nft` is provided for rune contracts.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs

Description

The bridge owner mints ordinal NFTs on the Stacks blockchain by providing the Bitcoin ordinal *hash* along side the deposit transaction hash. The `mint-ordinals-batch` function from the `pontis-bridge-v1` contract, which is responsible for minting the NFTs on Stacks, allows arbitrary transaction hashes with a maximum length of 40 bytes.

Bitcoin ordinals are identified by their inscription IDs. These IDs contain the genesis transaction hash (or reveal transaction hash) and an index of the inscription, preceded by the "i" character.

Example:

`521f8eccffa4c41a3a7728dd012ea5a4a02feed81f41159231251ecf1e5c79dai0`

If this identifier is provided when minting the NFT, the operation succeeds, but an issue arises with the token URI. The `pontis-bridge-ordinals-nft::get-token-uri` function returns the URI as a combination of a base URI and the same ordinal hash provided during minting. However, when converting the hash to a string, the inner function `reducer-1-string-string` can handle only 32 bytes of data, representing just a transaction hash.

If the intent is to set only the reveal or genesis transaction hash, then the existing code will work. However, users and external integrators will not be able to identify the Ordinal by its genesis hash, creating a different issue altogether. The hash-ID mapping can be retrieved via `get-hash-from-id` or `get-hash-from-id-fold`.

Recommendation

Modify the `buff-to-hex` and `reducer-1-string-string` functions to support converting the inscription ID into a string and pass that as the hash for minting. If we encode the transaction as bytes (buff), then the first 32 bytes can be split and converted to a string using the existing `reducer-1-string-string`, and the 16 last 2 bytes can correspond to "iN" where N is the inscription index number, an integer, up to the value 6.

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

8.4. Low Findings

[L-01] Bridge Token Name and Symbol Should Not Be Changeable

Description

The fungible wrappers for `psBTC`, `Runes` and `Bridge Token`, defined in `pontis-bridge-psBTC`, `pontis-bridge-rune` and `pontis-bridge-ft` provide the option to change the symbol and name of the fungible token.

The name and symbol, in conjunction with the contract address, should not be changeable. Any external integrator or price aggregator that uses these two elements in their UI will cause user confusion if they are ever changed.

Although SIP-10 does not mention this, it is generally understood that once a fungible token has launched, its name and symbol should never change.

Recommendation

Remove the `set-symbol` and `set-name` functions from all the `bridge-token` token contracts.

CONTENTS

1. About Clarity Alliance 2

2. Disclaimer 3

3. Introduction 4

4. About Pontis Bridge 4

5. Risk Classification 4

5.1. Impact 4

5.2. Likelihood 5

5.3. Action required for severity levels 5

6. Security Assessment Summary 5

7. Executive Summary 6

8. Findings 7

8.1. Critical Findings 8

[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8

8.2. High Findings 10

[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10

[H-02] mint Attempts to Mint Duplicate IDs 12

[H-03] Incorrect Fee Calculations 13

8.3. Medium Findings 14

[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14

[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16

8.4. Low Findings 17

[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17

[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18

[L-03] Pausing Stacks Pegging-In May Cause System Problems 19

[L-04] Implement 2-step Ownable 20

8.5. QA Findings 21

[QA-01] Duplicated Ownable Trait Operator File 21

[QA-02] Use Errors Instead of Panicking 22

[QA-03] Redundant or Dead Code 23

[QA-04] Reverse Parameters for Consistency 24

[QA-05] Group UTXO Operations to Reduce Code Size 25

[QA-06] Principal Instance Cast as a Trait 26

[QA-07] Isolate Expression into a Function 27

[QA-08] Confusing Error Messages 28

[QA-09] Isolate Iterator Expression into a Function 29

[L-02] `MAX-BTC-BRIDGE-MINIMUM` is set to wrong value

Description

The `MAX-BTC-BRIDGE-MINIMUM` is set to 0.001 BTC instead of 0.005 BTC as mentioned in the comment above.

```
;; maximum possible for min bridge is 0.005 BTC
(define-constant MAX-BTC-BRIDGE-MINIMUM u100000)
```

Recommendation

Either update the value or correct the comment.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

[L-03] Pausing Stacks Pegging-In May Cause System Problems

Description

The Pontis cross-chain bridge will initially provide bridging from Bitcoin to the Stacks blockchain for runes, ordinals and BTC. For all of these three assets, there are corresponding pausing mechanisms on the Stacks side via the flags `ordinals-paused` , `btc-paused` and `rune-tokens` that disables the bridging process.

On the Stacks side, both minting (the final step in bridging in) and pegging-out (bridging out) the assets are controlled by the indicated flags. However, the Bitcoin blockchain does not provide a way to prevent people from initiating bridging via direct transfers to the bridge address.

If the nodes themselves are not stopped, the bridging transactions will fail, resulting in gas loss and causing users to lose a small amount of funds during the bridging process. It is also cost-efficient to check off-chain if the Stacks bridge is paused before attempting bridging and verifying if the revert is due to a pause error.

There is another potential issue with pausing the pegging-in mechanism. Bridging from the Bitcoin blockchain to the Stacks blockchain involves a slight delay until all nodes agree and sign off on the bridged transaction. This operation can last from a few seconds to a few minutes.

During this time, if a user has already initiated bridging from Bitcoin to Stacks and the Bitcoin block coincides with the Stacks block that includes the pause activation, their bridging will be reverted on the Stacks side even though they initiated it chronologically before the pause was activated.

Recommendation

Create getters for the `btc-paused` and `ordinals-paused` flags, and have the node software check the flags before attempting to validated the transaction. For the `rune-tokens` flag, there already is a getter function available.

Additionally, consider thoroughly documenting how the system is expected to behave when a pause is initiated.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

[L-04] Implement 2-step Ownable

Description

Currently, all contracts implementing the `ownable-trait` have a simple setter for `contract-owner` - `set-contract-owner`. This function performs access control and directly changes the active admin account.

Recommendation

Consider implementing a 2-step pull system where the current admin proposes a new admin, and the new principal must manually claim the role through a pull mechanism.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

8.5. QA Findings

[QA-01] Duplicated Ownable Trait Operator File

Description

The `trait-operator.clar` source file defines an `ownable-trait` that is never used. This trait is already correctly defined and utilized in the `trait-operator.clar` source file.

Recommendation

Remove the `trait-operator.clar` file entirely, or rename it and modify the `set-contract-owner` function within it to an appropriate operator approval function.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

[QA-02] Use errors Instead of Panicking

Description

In the `pontis-bridge-ordinals-nft` contract, there are instances where `unwrap-panic` is used instead of `unwrap!` with a custom error. Specifically, this occurs in the `reducer-1-string-string` and `buff-to-hex-single` functions.

Ending execution in a panic results in a runtime error. Runtime errors cannot be handled by the caller and do not provide meaningful information about the execution, making them undesirable.

Recommendation

Replace `unwrap-panic` with `unwrap!` and a custome error.

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

[QA-03] Redundant or Dead Code

Description

Throughout the codebase, there are constants and variables that are declared but not used.

In `pontis-bridge-ordinals-nft`:

```
(define-constant ERR-TRANSFER (err u103))
(define-constant ERR-EMPTY-OWNER (err u108))
(define-constant ERR-UNWRAP-HASH (err u112))
(define-constant ERR-UNWRAP-TX-HASH (err u113))
(define-constant ERR-HASH-EXISTS (err u114))
```

In `pontis-bridge-ft.clar`:

```
(define-constant ERR-TRANSFER-FAILED (err u3000))
...
(define-map processed-tx-hashes (buff 36) bool)
```

Recommendation

Remove all unnecessary and unused code.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

[QA-04] Reverse Parameters for Consistency

Description

This function `set-custom-min-runes-bridge` currently has the `min-bridge` parameter, the mutation value, before the `rune` parameter, which is the mutated entry.

Recommendation

Consider reversing their order for consistency:

```
(define-public (set-custom-min-runes-bridge (rune principal)
   (min-bridge unit)) ...)
```

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

[QA-05] Group UTXO operations to reduce code size

Description

When pegging out runes and BTC, the functions `peg-out-runes` and `peg-out-btc` are respectively called. In both of these functions, the following operations are duplicated as code:

```
(let
    (
      ;; ... code ...
      (key-utxo-index (var-get current-key-utxo))
      (key-utxo (unwrap!
        (map-get? available-peg-out-key-utxo key-utxo-index) ERR-NO-KEY-UTXO))
    )
    (var-set current-key-utxo (+ key-utxo-index u1))
```

Recommendation

Since code size impacts on-chain gas costs and limitations, consider creating a private helper function that retrieves the last available UTXO and increments the index. Here is a basic example of how this can be done:

```
(define-private (get-and-increment-utxo)
  (let
    (
      (key-utxo-index (var-get current-key-utxo))
    )
      (var-set current-key-utxo (+ key-utxo-index u1))
    (ok (unwrap!
      (map-get? available-peg-out-key-utxo key-utxo-index) ERR-NO-KEY-UTXO))
  )
)
```

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

[QA-06] Principal Instance Cast as a Trait

Description

Currently, `migrate-bridge-instance` receives `bridge`, the bridge instance, as a `principal`. This could lead to a non-bridge contract being mistakenly set as the current instance.

Recommendation

Consider implementing a bridge trait for `pontis-bridge-v1` and subsequently using it as the parameter type for `bridge` in `migrate-bridge-instance`.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

[QA-07] Isolate Expression into a Function

Description

The following expression appears multiple times across various contracts:

```
(asserts! (is-eq contract-caller (unwrap!
   (contract-call? .pontis-bridge-controller get-latest-bridge-instance)
    ERR-BRIDGE-INSTANCE)) ERR-NOT-AUTHORIZED)
```

Recommendation

Consider isolating this expression into a function to reduce redundancy.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

[QA-08] Confusing Error Messages

Description

Currently, the `nft-transfer?` call in `transfer-fold` is wrapped in an `unwrap!` with a custom error message. This is not optimal as the aforementioned `SIP009` function can return errors ranging from `u1` to `u3`. This will make debugging any batch ordinal transfers into unserializable operations.

Recommendation

Consider throwing the original error instead and switching the line to use a `try!` expression.

CONTENTS

1. About Clarity Alliance 2
2. Disclaimer 3
3. Introduction 4
4. About Pontis Bridge 4
5. Risk Classification 4
5.1. Impact 4
5.2. Likelihood 5
5.3. Action required for severity levels 5
6. Security Assessment Summary 5
7. Executive Summary 6
8. Findings 7
8.1. Critical Findings 8
[C-01] User Bridged-In Assets Can Be Stranded by Abusing the Peg-Out Mechanism 8
8.2. High Findings 10
[H-01] Ordinal Stacks to Bitcoin Bridging Can Be Corrupted 10
[H-02] mint Attempts to Mint Duplicate IDs 12
[H-03] Incorrect Fee Calculations 13
8.3. Medium Findings 14
[M-01] Bitcoin Ordinal NFT Collections Are Jumbled on Stacks 14
[M-02] Ordinal Token URI Retrieval Mechanism is Incompatible with Inscription IDs 16
8.4. Low Findings 17
[L-01] Bridge Token Name and Symbol Should Not Be Changeable 17
[L-02] MAX-BTC-BRIDGE-MINIMUM is set to the wrong value 18
[L-03] Pausing Stacks Pegging-In May Cause System Problems 19
[L-04] Implement 2-step Ownable 20
8.5. QA Findings 21
[QA-01] Duplicated Ownable Trait Operator File 21
[QA-02] Use Errors Instead of Panicking 22
[QA-03] Redundant or Dead Code 23
[QA-04] Reverse Parameters for Consistency 24
[QA-05] Group UTXO Operations to Reduce Code Size 25
[QA-06] Principal Instance Cast as a Trait 26
[QA-07] Isolate Expression into a Function 27
[QA-08] Confusing Error Messages 28
[QA-09] Isolate Iterator Expression into a Function 29

[QA-09] Isolate Iterator Expression into a Function

Description

Currently, several functions that iterate over a list of values use the following expression to isolate a list of iterator IDs for looping:

```
(it (unwrap! (slice? ITERATOR u0 (len <PARAM_NAME>)) ERR-UNWRAP-ITERATOR))
```

Recommendation

Consider isolating the above expression into a separate function and using that function instead.