Defense By Thesis

Security Audit Report

NEAR Satoshi Bridge

NEAR Satoshi Bridge Smart Contracts

Initial Report // June 18, 2025
Final Report // August 12, 2025

Team Members

Ahmad Jawid Jamiulahmadi // Senior Security Auditor
Mukesh Jaiswal // Senior Security Auditor

Table of Contents

About Thesis Defense

Defense is the security auditing arm of Thesis, Inc., the venture studio behind tBTC, Fold, Mezo, Acre, Taho, Etcher, and Embody. At Defense, we fight for the integrity and empowerment of the individual by strengthening the security of emerging technologies to promote a decentralized future and user freedom. Defense is the leading Bitcoin applied cryptography and security auditing firm. Our team of security auditors have carried out hundreds of security audits for decentralized systems across a number of ecosystems including Bitcoin, Ethereum + EVMs, Stacks, Cosmos SDK, NEAR and more. We offer our services within a variety of technologies including smart contracts, bridges, cryptography, node implementations, wallets and browser extensions, and dApps.

Defense will employ the Defense Audit Approach and Audit Process to the in scope service. In the event that certain processes and methodologies are not applicable to the in scope services, we will indicate as such in individual audit or design review SOWs. In addition, Thesis Defense provides clear guidance on successful Security Audit Preparation.

Section 1.0
Scope

Technical Scope

- **Repository:** https://github.com/Near-Bridge-Lab/btc-bridge
- **Audit Commit:** `4b7a48eae8c303f0f1e29fc571906a93846464aa`
- **Verification Commit:** `fb15ddeb7b9eb627430da012ce3de2813f03d545`
- **Directory in Scope:** contracts/satoshi-bridge/*

Section 2.0
Executive Summary

Schedule

This security audit was conducted from May 7, 2025 to June 17, 2025 by 2 senior security auditors for a total of 11 person weeks.

Overview

Near Satoshi Bridge (NSP) is built using NEAR Chain Abstraction. It enables asset transfers between chains and allows users on the source chain to directly interact with dApps on the NEAR network. NSP aims to enhance the Web3 experience for Bitcoin (BTC) holders by providing seamless access to the entire NEAR ecosystem—without requiring users to create a new wallet or generate a new key pair.

During this audit our team reviewed both the Satoshi Bridge and Account implementations. This report details the issues found in the Satoshi Bridge component.

Threat Model

Our team created a threat model that guided the areas investigated during this security Audit. We focused on the critical financial and operational risks in the protocol's handling of Bitcoin bridging, transaction orchestration, and token conversion logic. We investigated several areas including:

Fund Accessibility Risks

- Potential for irreversible asset lockup due to protocol design constraints
- Scenarios where user funds could become unspendable or trapped

Economic Risk to Participants

- Situations where honest relayers or users may suffer financial losses
- Imbalances between upfront costs and guaranteed compensation
- Exploit paths that allow draining of protocol or user-held balances

Execution Flow Integrity

- Inconsistencies in how multi-step transactions are processed
- Risks arising from unordered or replayable transactions
- Partial or duplicate execution of user intentions

Abuse of Public Interfaces

- Unrestricted access to sensitive functions leading to unintended behavior
- Weak controls around who can initiate or finalize critical operations

Denial-of-Service (DoS) Conditions

- Resource exhaustion vectors that degrade performance or block participation
- Limits or thresholds that unintentionally prevent necessary operations

Replay and Front-Running Scenarios

- Potential for adversaries to exploit timing gaps or lack of nonce enforcement
- Opportunities for malicious actors to intercept and re-use valid operations

Permission and Access Control Gaps

- Missing validations on restricted actions or critical state changes
- Scenarios where unauthorized users can trigger privileged behaviors

Calculation Reliability and Precision

- Loss of accuracy due to integer math or improper scaling
- Overflow risks in core economic calculations
- Incorrect fee or token amounts resulting from rounding errors

Configurable Logic Vulnerabilities

- Removal or misconfiguration of parameters affecting system stability
- Failure conditions when critical configuration states are invalid

Protocol Misuse or Ambiguity

- Design assumptions that can be violated in edge cases
- Lack of enforcement of intent ordering or transactional dependencies

Security by Design

The system reflects a forward-looking architectural model with structured transaction intent handling, flexible deposit mechanisms, and modular validation layers. However, several design choices reveal gaps that could undermine the protocol's long-term dependability and security guarantees. For instance, conflicting rules around UTXO management pose a latent risk of gradual fragmentation and eventual bridge lockup, a concern rooted in policy-level misalignment rather than implementation.

Additionally, the absence of enforced transaction ordering and atomic execution during intention replacement introduces race conditions and potential fund duplication scenarios. Mechanisms intended to enhance control—such as Replace-By-Fee (RBF)—currently lack the nuance to distinguish cancellation intents, which may inadvertently render transactions permanently unresolvable. Similarly, allowing confirmation strategies to be removed without ensuring that at least one remains risks disabling key verification logic essential to deposit recognition.

These findings suggest that while the system is conceptually well-structured, a number of design-level assumptions warrant re-examination to strengthen resilience under edge conditions and adversarial use.

Secure Implementation

The implementation exhibits significant care in structuring validation logic and smart contract operations, yet several vulnerabilities emerge from inconsistent enforcement paths and insufficient access protections. Key functions such as `verify_deposit` and `sign_near_txs` are accessible without appropriate restrictions, creating opportunities for front-running, duplicate execution, or bypassed preconditions—particularly when handling `extra_msg` fields or critical relayer operations.

Gas usage and nonce handling are susceptible to subtle edge cases, such as nonce resets enabling replay attacks or unbounded gas calculations draining contract balances. Financial computations, including `yoctonear` conversions, are prone to both precision loss and overflow risks, which could affect fairness or trigger runtime failures if not handled with fixed-point arithmetic. The protocol also permits restricted user actions under certain payment modes, and lacks safeguards to prevent confirmation strategy misconfigurations, which may delay or block deposit finalization.

Although many issues can be addressed with targeted fixes and validations, the overall posture suggests that tighter coupling between the intended logic and real-world execution paths is needed to fully uphold the protocol's security goals.

Use of Dependencies

Oracles

The contract currently relies on two oracles `Pyth and Price Oracle` to retrieve asset prices. However, when updating the price of a specific gas token, the `yoctonear` calculation may suffer from precision loss when using the `Pyth Oracle`. Additionally, in the current implementation of the price_oracle module, the `yoctonear` calculation may be prone to potential overflow.

Relayer

The system relies heavily on relayers to process transactions, validate deposits, and execute actions on behalf of users. This dependency introduces risks: if relayers behave incorrectly, are front-run by others, or fail to follow expected validation steps, it can lead to lost funds, duplicate executions, or denial-of-service conditions. Additionally, malicious actors can exploit gaps in relayer logic to drain balances or bypass safeguards, making the protocol's security and reliability closely tied to relayer behavior and protections.

Tests

The implementation has good test coverage but significant gaps remain that could impact the contract's reliability; strengthening the test suite is strongly recommended to enable more robust protection against potential vulnerabilities.

Project Documentation

The documentation available was limited in scope and did not comprehensively reflect all changes introduced by the development team. We recommend improving the project documentation.

Section 3.0
Key Findings Table

Issues	Severity	Status
ISSUE #1 Conflicting UTXO Management Rules May Eventually Lead to Bridge Lockup	High	Fixed
ISSUE #2 Certain Limits Can Avoid Creating Cancellation RBFs	High	Fixed
ISSUE #3 Deposit Messages Containing `extra_msg` Can Be Directly Submitted to `verify_deposit`	High	Fixed
ISSUE #4 Missing Validation in `remove_confirmations_strategy` Function	Low	Fixed

Severity definitions can be found in Appendix A

Section 4.0
Findings

We describe the security issues identified during the security audit, along with their potential impact. We also note areas for improvement and optimizations in accordance with best practices. This includes recommendations to mitigate or remediate the issues we identify, in addition to their status before and after the fix verification.

ISSUE#1

Conflicting UTXO Management Rules May Eventually Lead to Bridge Lockup

High Fixed

Location

contracts/satoshi-bridge/src/psbt.rs#L147-L154

contracts/satoshi-bridge/src/psbt.rs#L38-L40

Description

The referenced check (`output_value < min_input_amount`) inside the `check_withdraw_psbt` function enforces that the `change` output value be smaller than the smallest input selected for withdrawal. This gradually shrinks UTXO sizes as smaller and smaller change outputs are produced.

In parallel, once the number of UTXOs exceeds the `passive_management_upper_limit` , passive UTXO management policies are applied to reduce UTXO count by consolidating small UTXOs into fewer, larger outputs. However, the aforementioned check conflicts with this policy, since consolidation naturally produces a larger output that may exceed the smallest input value, violating the `output_value < min_input_amount` rule. New deposits may temporarily delay reaching the point where no valid combination of inputs exists that satisfies both the `output_value < min_input_amount` rule and the passive UTXO management rule. However, they do not eliminate the risk entirely.

Additionally, UTXOs that are equal to or barely larger than the protocol's `min_change_amount` will become unusable, as they cannot satisfy both the aforementioned check and the minimum change size constraint simultaneously. As UTXOs get progressively smaller over time, the system may eventually reach a state where withdrawals become impossible, resulting in locking of funds within the bridge.

Impact

- As UTXOs shrink over time, the bridge may eventually reach a state where no valid withdrawal combinations exist, resulting in bridge fund lockup.
- Certain withdrawal requests may become impossible to fulfill due to conflicting constraints between passive UTXO management and change output validation.
- UTXOs near or equal to `min_change_amount` become unspendable, effectively locking portions of the bridge's funds.

Recommendation

To help ensure long-term bridge operability and to prevent gradual fund lockup due to excessive UTXO fragmentation we recommend implementing an active UTXO management process that periodically consolidates fragmented small UTXOs into larger ones.

ISSUE#2

Certain Limits Can Avoid Creating Cancellation RBFs

High | Fixed

Location

contracts/satoshi-bridge/src/rbf/mod.rs#L32-L35

contracts/satoshi-bridge/src/psbt.rs#L115-L121

contracts/satoshi-bridge/src/rbf/cancel_withdraw.rs#L49-L51

Description

The protocol includes Cancel RBF functionality to allow cancellation of BTC transactions that have been pending for an extended period. However, current design constraints imposed on the Replace-By-Fee (RBF) mechanism can inadvertently block these cancellation attempts. Specifically:

- The `rbf_num_limit` —which caps the number of RBF attempts for a given pending BTC transaction—is enforced within the `set_rbf_pending_info` function. Once this limit is reached, even cancellation RBFs are disallowed.
- If the original BTC transaction or its last RBF attempt already set the `gas_fee` to the protocol's configured maximum allowed gas amount (`max_btc_gas_fee`), the cancellation RBF cannot exceed this cap—effectively making the transaction non-cancellable via RBF.

Impact

These limitations can result in scenarios where a BTC transaction becomes permanently stuck in an unconfirmed state, with no mechanism available to cancel it or accelerate its inclusion in the blockchain.

Recommendation

We recommend decoupling Cancel RBFs from normal RBF logic by:

- Exempting cancellation RBF attempts from the standard `rbf_num_limit` .
- Allowing cancellation RBFs to override the `max_btc_gas_fee` cap.

ISSUE#3

Deposit Messages Containing `extra_msg` Can Be Directly Submitted to `verify_deposit`

High Fixed

Location

contracts/satoshi-account/src/api/bridge.rs#L38

contracts/satoshi-bridge/src/api/bridge.rs#L21

contracts/satoshi-account/src/api/token_receiver.rs#L56-L58

contracts/satoshi-bridge/src/btc_light_client/deposit.rs#L106-L111

Description

The `verify_deposit` function can be called directly by anyone, allowing any relayer or user to process deposit messages that include an `extra_msg` intended for the `csna_verify_deposit` function. When `verify_deposit` is called directly, the `extra_msg` is ignored while post actions are still executed, causing tokens to be transferred without properly processing prerequisite actions such as CSNA account creation or storage deposits.

In this scenario, if CSNA account creation was part of the `extra_msg` , tokens are sent from an uncreated CSNA account directly to the `receiver_id` specified in each post action. The relayer fee post action is refunded via the `ft_on_transfer` function due to the absence of debt information, but the handling of other post action transfers depends on how each receiving contract processes unexpected transfers.

Furthermore, since anyone can call `verify_deposit` and claim the relayer fee, this creates an incentive for third parties to front-run `whitelisted` relayers. An `whitelisted` relayer calling `csna_verify_deposit` after a non- `whitelisted` relayer has already executed `verify_deposit` will experience a failed call due to the deposit already being processed, while still incurring costs from executing `extra_msg` operations.

Impact

- Third parties can front-run `whitelisted` relayers to claim relayer fees without properly handling `extra_msg` processing.
- Post actions may execute prematurely, resulting in unexpected behavior depending on how receiving contracts handle token transfers without prior storage deposits or account creation.
- Users may be unable to access their tokens until the CSNA account is manually created, potentially leaving funds locked indefinitely.
- `whitelisted` relayers may incur losses if they attempt to process already completed deposits, absorbing unrecoverable costs from processing `extra_msg` operations.

Recommendation

We recommend implementing stricter access control and validation to ensure that deposit messages containing `extra_msg` are only processed through the `csna_verify_deposit` function. Specifically:

- Restrict execution of `verify_deposit` for deposit messages that include an `extra_msg` . If `extra_msg` is present, enforce that only `whitelisted` relayers are permitted to process the deposit through `csna_verify_deposit` .
- Introduce a validation mechanism that rejects direct `verify_deposit` calls when `extra_msg` data is attached.

Verification Status

The introduction of `extra_msg_delta` can extensively reduce the chances of a whitelisted relayer being front-run. However, it does not remediate the issue completely. Additionally, it is still possible for a whitelisted relayer to directly call the `verify_deposit` function with a deposit message containing `extra_msg` .

ISSUE#4

Missing Validation in `remove_confirmations_strategy` Function

Low Fixed

Location

contracts/satoshi-bridge/src/api/management.rs

Description

The `remove_confirmations_strategy` function allows for the removal of confirmation strategies without first validating if at least one strategy exists within `confirmations_strategy` . This could lead to a state where no confirmation strategies are defined, potentially impacting transaction processing, as the system relies on these strategies to determine the number of confirmations required based on the Bitcoin transaction amount.

Impact

The bridge relies on the `confirmations_strategy` to validate user actions, such as deposits. If this list becomes empty, the bridge loses its ability to verify these actions. As a result, deposits may remain in an unconfirmed state, with their processing contingent on the relayer retrying the transaction. In the absence of relayer intervention, users may be forced to contact the protocol team directly to resolve the issue. This scenario introduces the risk of user funds becoming temporarily locked and delays in accessing core bridge functionalities.

Recommendation

We recommend adding a check for `confirmation_strategy` to ensure at least one strategy is always present, even after removing another.

Section 5.0
Appendix A

Severity Rating Definitions

At Thesis Defense, we utilize the Immunefi Vulnerability Severity Classification System - v2.3.

Severity	Definition
Critical	• Manipulation of governance voting result deviating from voted outcome and resulting in a direct change from intended effect of original results • Direct theft of any user funds, whether at-rest or in-motion, other than unclaimed yield • Direct theft of any user NFTs, whether at-rest or in-motion, other than unclaimed royalties • Permanent freezing of funds • Permanent freezing of NFTs • Unauthorized minting of NFTs • Predictable or manipulable RNG that results in abuse of the principal or NFT • Unintended alteration of what the NFT represents (e.g. token URI, payload, artistic content) • Protocol insolvency
High	• Theft of unclaimed yield • Theft of unclaimed royalties • Permanent freezing of unclaimed yield • Permanent freezing of unclaimed royalties • Temporary freezing of funds • Temporary freezing NFTs
Medium	• Smart contract unable to operate due to lack of token funds • Enabling/disabling notifications • Griefing (e.g. no profit motive for an attacker, but damage to the users or the protocol) • Theft of gas • Unbounded gas consumption
Low	• Contract fails to deliver promised returns, but doesn't lose value
None	• We make note of issues of no severity that reflect best practice recommendations or opportunities for optimization, including, but not limited to, gas optimization, the divergence from standard coding practices, code readability issues, the incorrect use of dependencies, insufficient test coverage, or the absence of documentation or code comments.

Section 6.0
Appendix B

Thesis Defense Disclaimer

Thesis Defense conducts its security audits and other services provided based on agreed-upon and specific scopes of work (SOWs) with our Customers. The analysis provided in our reports is based solely on the information available and the state of the systems at the time of review. While Thesis Defense strives to provide thorough and accurate analysis, our reports do not constitute a guarantee of the project's security and should not be interpreted as assurances of error-free or risk-free project operations. It is imperative to acknowledge that all technological evaluations are inherently subject to risks and uncertainties due to the emergent nature of cryptographic technologies.

Our reports are not intended to be utilized as financial, investment, legal, tax, or regulatory advice, nor should they be perceived as an endorsement of any particular technology or project. No third party should rely on these reports for the purpose of making investment decisions or consider them as a guarantee of project security.

Links to external websites and references to third-party information within our reports are provided solely for the user's convenience. Thesis Defense does not control, endorse, or assume responsibility for the content or privacy practices of any linked external sites. Users should exercise caution and independently verify any information obtained from third-party sources.

The contents of our reports, including methodologies, data analysis, and conclusions, are the proprietary intellectual property of Thesis Defense and are provided exclusively for the specified use of our Customers. Unauthorized disclosure, reproduction, or distribution of this material is strictly prohibited unless explicitly authorized by Thesis Defense. Thesis Defense does not assume any obligation to update the information contained within our reports post-publication, nor do we owe a duty to any third party by virtue of making these analyses available.