safepress

Smart Contract Audit

Aptos Endpoint of the Multi-Chain NFT Bridge

CUSTOMER: XP.NETWORK

DATE: Oct 13th, 2022

Code review and security report

IMPORTANT: This document likely contains critical information about the Client's software and hardware systems, security susceptibilities, descriptions of possible exploits and attack vectors. The document shall remain undisclosed until any significant vulnerabilities are remedied.

CLIENT:	XP.NETWORK	START DATE:	Oct 1st, 2022
TYPE, SUBTYPE:	NFT Bridge	END DATE:	Oct 13th, 2022

Scope

REPOSITORY:	https://github.com/XP-NETWORK/aptos-integration-move
DOCUMENTATION:	No documentation
TESTS:	Passing
AUDITORS:	Zain Franci, Brandon Botosh
REVIEW $ APPROVAL:	Ryan Rhiel Madsen
SMART CONTRACT AUDITED:	sources / bridge.move

Commit hashes:

BASE:	E2DAA0CA775A508F40B6A7CAC6B8019A26688CFC (INCL.)
UPDATE 1:	a6440970003ef7bc6afda48a26142d92558a4865

Definitions of vulnerability classification

CRITICAL

Bug / Logic failures in the code that cause loss of assets / data manipulation.

HIGH

Difficult to exploit problems which could result in elevated privileges, data loss etc.

MEDIUM

Bug / Logic failures in the code which need to be fixed but cannot lead to loss of assets / data manipulation.

LOW

Mostly related to unused code, style guide violations, code snippets with low effect etc.

Findings

1	0	1	0	0
Critical	High	Medium	Low	Informational

Summary

XPAPT-01	Improper implementation for enforcing uniqueness.	● Critical	✓ Fixed
XPAPT-02	Code repetition	● Medium	✓ Fixed

Finding: XPAPT-01

Improper implementation for enforcing uniqueness.

Base	Function `initialize`	Lines 123 - 148		● Critical	✓ Fixed

Description

The `initialize` function has the parameter `admin` In the function, we are checking if an instance of struct `Bridge` exists for that particular address or not on line 128.

Anyone can call the `initialize` function and anyone can initialize an instance of the struct `Bridge` since `admin` is a generic signer.

This leads to the issue of `action_id` uniqueness in all the other functions, since two different instances of the struct `Bridge` for two different admin addresses can have the same `action_id`.

Recommendation

It is recommended to have a hardcoded admin address and use that address to ensure that the `initialize` function can be called only once and can be called only by the admin.

Another recommendation is to use `init_module` which is a function that only runs once during deployment and never again [1]. The documentation for this function is sparse.

Finding: XPAPT-02

Unimplemented TODOs in code

Base	Function	`pause`	Line 155-167	● Medium	✓ Fixed
Base	Function	`unpause`	Line 182-200	● Medium	✓ Fixed
Base	Function	`update_group_key`	Line 211-230	● Medium	✓ Fixed
Base	Function	`validate_whitelist`	Line 246-266	● Medium	✓ Fixed
Base	Function	`validate_blacklist`	Line 284-304	● Medium	✓ Fixed
Base	Function	`validate_withdraw_fees`	Line 322-341	● Medium	✓ Fixed
Base	Function	`validate_transfer_nft`	Line 366-384	● Medium	✓ Fixed
Base	Function	`withdraw_nft`	Line 432-437	● Medium	✓ Fixed
Base	Function	`validate_burn_nft`	Line 479-501	● Medium	✓ Fixed
Base	Function	`freeze_nft`	Line 527-531	● Medium	✓ Fixed
Base	Function	`validate_unfreeze_nft`	Line 575-598	● Medium	✓ Fixed

Description

The code to verify that

- the bridge has been initialized,
- the bridge is not paused and
- the `action_id` is not being reused

is repeated in the functions mentioned above. The repetition is cluttering the program and will make it difficult to maintain the program in future.

Recommendation

It is recommended to extract the repeated code for the checks into their own utility function which can be reused in the above mentioned functions.

Executive Summary

Based on the audit findings the Client’s contracts are: Well Secured

○ Not Secure	○ Insufficiently Secured	○ Secured	● **Well Secured**

Disclaimers

SafePress Disclaimer

The smart contracts given for audit have been analyzed by the best industry practices at the date of this report, with cybersecurity vulnerabilities and issues in smart contract source code, the details of which are disclosed in this report (Source Code); the Source Code compilation, deployment, and functionality (performing the intended functions). The audit makes no statements or warranties on the security of the code. It also cannot be considered a sufficient assessment regarding the utility and safety of the code, bug-free status, or any other contract statements. While we have done our best in conducting the analysis and producing this report, it is important to note that you should not rely on this report only — we recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts.

Technical Disclaimer

Smart contracts are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the smart contract can have vulnerabilities that can lead to hacks. Thus, the audit cannot guarantee the explicit security of the audited smart contracts.

References

[1] Aptos Dev, Step 4.2: Understanding the MoonCoin module,
https://aptos.dev/tutorials/your-first-coin/#step-42-understanding-the-mooncoin-module