




Smart Contract Audit

EVM Endpoint of the Multi-Chain FT Bridge

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:	EMMET.FINANCE	START DATE:	Jul 1st, 2024
TYPE, SUBTYPE:	FT Bridge	END DATE:	JUL 18th, 2024

Scope

REPOSITORY:	https://github.com/Emmet-Finance/Emmet-CCM/tree/main/contracts/bridge
DOCUMENTATION:	Inline Documentation
TESTS:	Passing
AUDITORS:	Zain Franci, Brandon Botosh
REVIEWS APPROVAL:	Ryan Rhiel Madsen
AUDITED CONTRACTS:	<div><div>1.</div>EmmetBridge.sol,<div>2.</div>EmmetBridgeHelper.sol,<div>3.</div>EmmetBridgeManager.sol</div>

Commit hashes

BASE:	dfe86aabf0fd641e17a1186868f71a48ac0f25b4
UPDATE 1:	3d3d747f103d57201634bdcf29830779c76da29f
UPDATE 2:	fbb24773519b440cd267ab6599784f518bb7a979

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



Summary

EFEB-01	Lack of input validation.	● Low	✓ Fixed
EFEB-02	Missing access control.	● Medium	✓ Fixed
EFEBH-01	Lack of module address validation.	● Low	✓ Fixed
EFEBM-01	Lack of IBridgeModule interface support.	● Low	✓ Fixed

Finding: EFEB-01

Lack of input validation.

Base	Function <code>sendInstallment</code> lines 116-183	● Low	✓ Fixed
------	---	-------	---------

Description

The `params` are not validated, potentially reverting without an apparent revert reason for the user.

Recommendation

Add params validation.

Finding: EFEB-02

Missing access control.

Base	Function <code>withdrawStuck</code> lines 211-249	● Medium	✓ Fixed
------	---	----------	---------

Description

The `withdrawStuck` function lacks a modifier to restrict access, allowing any caller to attempt withdrawals for transactions they did not initiate.

Recommendation

Add an `onlySender(txHash)` modifier to restrict access to the original sender of the transaction.

Finding: EFEBH-01

Lack of module address validation.

Base	Function <code>switchOutgoing</code> lines 59-117	● Medium	✓ Fixed
------	---	----------	---------

Description

If the `module` is equal to address zero, and the module address is not set to the requested token strategy, the `IBridge(module)sendInstallment` will fail without an apparent revert reason.

Recommendation

Add the module address check and the revert reason.

Finding: EFEBM-01

Lack of IBridgeModule interface support.

Base	Function <code>addBridgeModule</code> lines 40-77	<div>● Low</div>	<div>✓ Fixed</div>
------	---	------------------	--------------------

Description

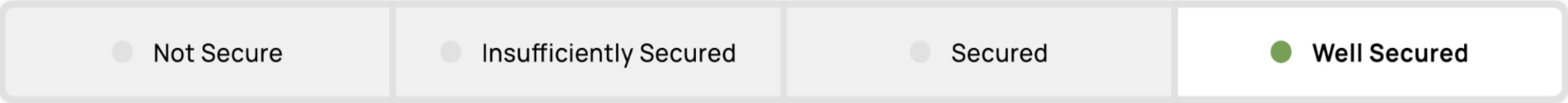
All further transactions will fail if the added module does not support the IBridgeModule.

Recommendation

Add params IBridgeModule support during the module addition.

Executive Summary

Based on the audit findings, the Client's contracts are: **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] Anchor-Lang Book, Enforcing Uniqueness

https://book.anchor-lang.com/anchor_in_depth/PDAs.html#enforcing-uniqueness

[2] Anchor-Lang Book, Creation of a PDA

https://book.anchor-lang.com/anchor_in_depth/PDAs.html#creation-of-a-pda