



SMART CONTRACT SECURITY AUDIT OF



Summary

Audit Firm: Guardian Audits

Client Firm: Bridges Exchange

Final Report Date - May 29, 2022

Audit Summary

After a line by line manual analysis and automated review, Guardian Audits has concluded that:

- Bridges' smart contracts have a **LOW RISK SEVERITY**
- Bridges' smart contracts have an **ACTIVE OWNERSHIP**
- Bridges' smart contract owner has multiple "write" privileges. Centralization risk correlated to the active ownership is **MEDIUM**

Notice that the examined smart contracts are not resistant to internal exploit. For a detailed understanding of risk severity, source code vulnerability, and potential attack vectors, refer to the complete audit report below.

 Blockchain network: **BSC**

 Verify the authenticity of this report on Guardian's GitHub: <https://github.com/guardianaudits>


 Comprehensive penetration + fuzzing test suite:
<https://github.com/GuardianAudits/BridgesTestSuite>

Table of Contents

Project Information

Project Overview 4

Audit Scope & Methodology 5

Smart Contract Risk Assessment

Inheritance Graph 7

Findings & Resolutions 8

Report Summary

Auditor’s Verdict 18

Addendum

Disclaimer 19

About Guardian Audits 20

Project Overview

Project Summary

Project Name	Bridges
Language	Solidity
Codebases	https://github.com/bridges-team/bridges-exchange-farm https://github.com/bridges-team/TokenVesting https://github.com/bridges-team/bridges-exchange-periphery https://github.com/bridges-team/bridges-exchange-swap-core
Commits	1e90d6c31870e778f316e3eaec76d09da59ec940 699436ff2cb35716c887ae653b55a68659d2e4d1 Eba4fcfc3d0ca2c11e93069897a0c4e9428bab8a 3d3bddefb9dd26632ff7e6d3085dfc641bde45fe

Audit Summary

Delivery Date	May 29, 2022
Audit Methodology	Static Analysis, Manual Review, Fuzzing

Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
● Critical	0	0	0	0	0	0
● High	0	0	0	0	0	0
● Medium	5	0	0	0	1	4
● Low	4	0	0	0	0	4

Audit Scope & Methodology

Scope

ID	File	SHA-1 Checksum
GG	GoldenGate.sol	6DB849074A31AFCA02CFE7DDB36062A21B3F9BA4
TV	TokenVault.sol	5CB5EF9D4ABCAB491CDDA8737020231C0029F667
BRT	BridgesRouter.sol	D2E911EDD0EBFCCFC4A060B39DEF675FCE595BF8
BRF	BridgesRef.sol	326435A36C632A2DCE26B2DCCF0C9F42DEB7676A
FACT	BridgesFactory.sol	7F784FE5EB937411DEB10C773F1A6D7481E508E9
PAIR	BridgesPair.sol	378376FA2640038FA241383D9F509737D01D8399

Audit Scope & Methodology

Methodology

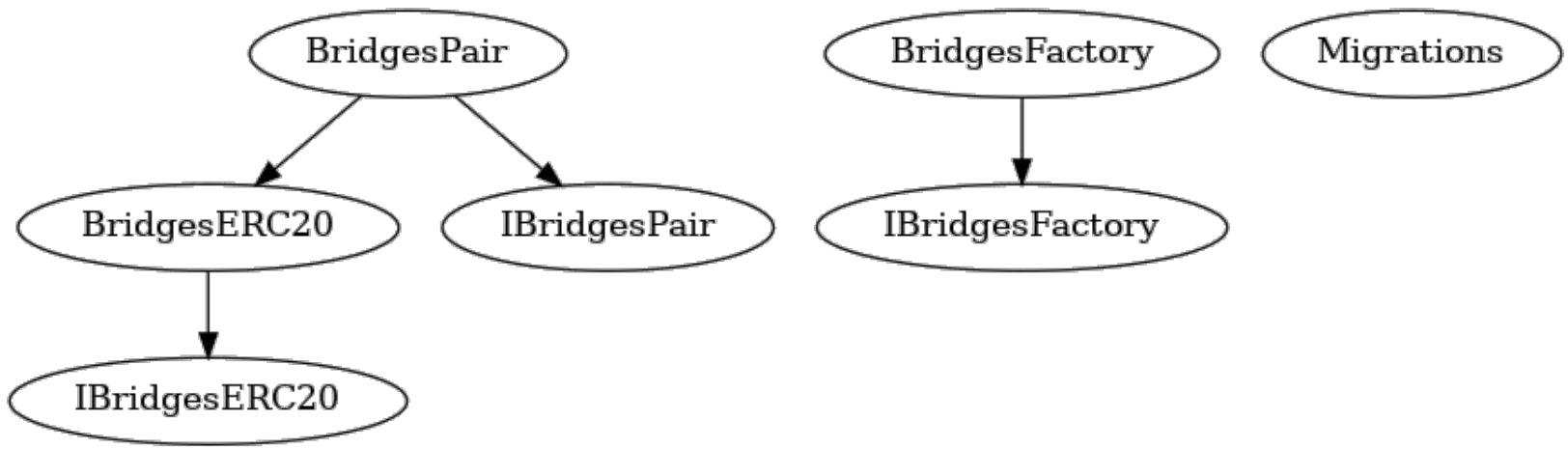
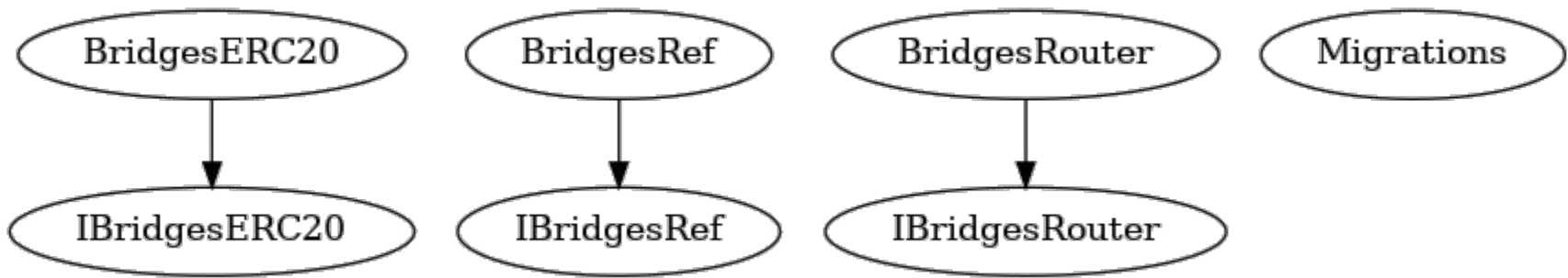
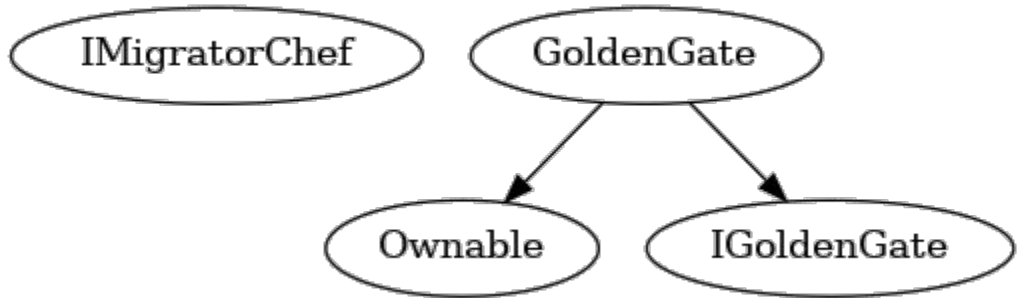
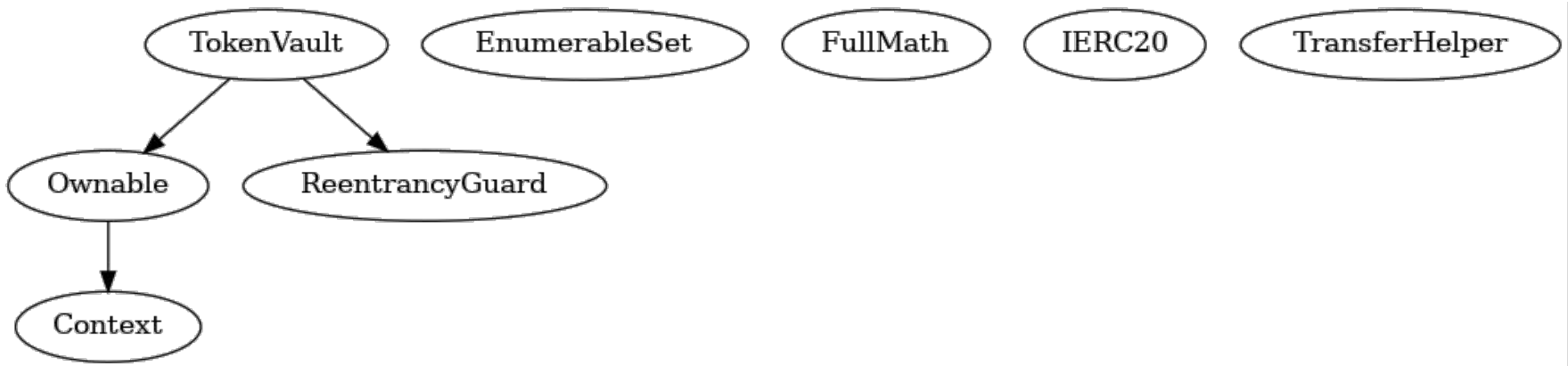
The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross-referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

Vulnerability Classifications

Vulnerability Level	Classification
● Critical	Easily exploitable by anyone, causing loss/manipulation of assets or data.
● High	Arduously exploitable by a subset of addresses, causing loss/manipulation of assets or data.
● Medium	Inherent risk of future exploits that may or may not impact the smart contract execution.
● Low	Minor deviation from best practices.

Inheritance Graph



Findings & Resolutions

ID	Title	Category	Severity	Status
<u>GLOBAL-1</u>	Centralization Risk	Centralization / Privilege	● Medium	Partially Resolved
<u>GG-1</u>	Unable to Emergency Withdraw	Logical Error	● Medium	Resolved
<u>FACT-1</u>	Immutability Modifiers	Mutability	● Low	Resolved
<u>PAIR-1</u>	Diluted Dividends	Logical Error	● Medium	Resolved
<u>PAIR-2</u>	Mutability Modifiers	Mutability	● Low	Resolved
<u>PAIR-3</u>	Superfluous Code	Optimization	● Low	Resolved
<u>PAIR-4</u>	Superfluous Code	Optimization	● Low	Resolved
<u>BRT-1</u>	Unexpected AmountOut	Logical Error	● Medium	Resolved

GLOBAL-1 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	● Medium	Global	Partially Resolved

Description

Privileged addresses have authority over many functions that may be used to negatively disrupt the project. Some important privileges include:

GoldenGate

- owner can withdraw all funds.
- owner can set admins which are able to dilute allocation of other pools.
- owner can set the migrator contract which can lead to loss of LP if malicious.

TokenVesting

- owner can arbitrarily set the fee and fee address which can lead to loss of user funds.

BridgesRef

- feeToSetter can arbitrarily set the distribution rate.
- feeToSetter can withdraw any ERC-20 token in the contract.

BridgesRouter

- feeSetter can set a arbitrary referral and dividend tracker contract.

Recommendation

Ensure that the privileged addresses are multi-sig and/or introduce timelock for improved community oversight. Optionally introduce require statements to limit the scope of the exploits that can be carried out by the privileged addresses.

GLOBAL-1 | Centralization Risk

Category	Severity	Location	Status
Centralization / Privilege	● Medium	Global	Partially Resolved

Resolution

Bridges Team:

- All centralized BNB and token withdrawal functions have been removed from the GoldenGate contract,
- The possibility to change fees on the TokenVault has been removed as well.
- The GoldenGate migrator has to be there for an eventual V2 in the future.
- For the same reason all update functions on the tracker are necessary.
- Every privileged address will be a multi-sig with trusted members in production.

GG-1 | Unable to Emergency Withdraw

Category	Severity	Location	Status
Denial-of-Service	● Medium	GoldenGate.sol:284	Resolved

Description

Due to `require(block.timestamp >= user.stakeUntil, "Locked")` in `emergencyWithdraw`, if a user has LP tokens that are not locked alongside LP tokens that are indeed locked, the user would have to wait until their locked LP tokens become unlocked before they can `emergencyWithdraw`.

Recommendation

If this is intended behavior, keep as is. Otherwise, refactor `emergencyWithdraw` such that users may withdraw their unlocked positions.

Resolution

Bridges Team:

- This is indeed expected behavior, if you have a locked position you cannot `emergencyWithdraw` any part of your position.

FACT-1 | Immutability Modifiers

Category	Severity	Location	Status
Mutability	● Low	BridgesFactory.sol:10	Resolved

Description

The GoldenGate address is not set after the constructor and can therefore be declared immutable.

Recommendation

Either make a setter for GoldenGate or declare it immutable.

Resolution

Bridges Team:

- Implemented a setter for GoldenGate.

PAIR-1 | Diluted Dividends

Category	Severity	Location	Status
Logical Error	● Medium	BridgesPair.sol:229, 232	Resolved

Description

The rewards for the BridgesPair contract are ignored on line 232 by adjusting the rewardDebt, but they are not excluded in the magnifiedDividendPerShare calculation, therefore decreasing the dividends received by every other holder.

Recommendation

Subtract the BridgesPair’s balance from the totalSupply on line 229.

Resolution

Bridges Team:

- Removed the BridgesPair contract balance from the dividends calculation.

PAIR-2 | Mutability Modifiers

Category	Severity	Location	Status
Mutability	● Low	BridgesPair.sol: 39	Resolved

Description

nullAddress is not changed anywhere and can therefore be declared constant.

Recommendation

Declare nullAddress constant.

Resolution

Bridges Team:

- Declared nullAddress constant.

PAIR-3 | Superfluous Code

Category	Severity	Location	Status
Optimization	● Low	BridgesPair.sol: 74	Resolved

Description

The UserInfo struct now only contains a rewardDebt, therefore the userInfo mapping can simply be a mapping of address => uint where the uint is the rewardDebt.

Recommendation

Delete the UserInfo struct and convert the userInfo mapping to a simple address => uint mapping storing the rewardDebt directly.

Resolution

Bridges Team:

- The mapping is now simply rewardDebt.

PAIR-4 | Superfluous Code

Category	Severity	Location	Status
Optimization	● Low	BridgesPair.sol	Resolved

Description

The `sendToGate`, `sendToGateFrom0`, `sendToDevFromGate`, and `sendToDevFrom0` functions all do the same thing just with different addresses.

Recommendation

Make one function that does this computation that accepts configurable addresses as arguments and add a `require` statement to limit the scope of which addresses can be used.

Resolution

Bridges Team:

- Combined these functions into one `withdrawSpecial` function.

BRT-1 | Unexpected AmountOut

Category	Severity	Location	Status
Logical Error	● Medium	BridgesRouter.sol: 154	Resolved

Description

Because the tradingFee is taken after the calculation of getAmountsIn, the user will receive $1000 - \text{tradingFee} / 10\%$ of amountOut, rather than getting the whole amountOut. If the tradingFee is 30, the user will receive only 97% of the specified amountOut.

Recommendation

If it is desired to receive the amountOut at minimum, take the fee in the same manner as in getAmountsIn, where the amountIn is simply increased in order to maintain the amountOut.

Resolution

Bridges Team:

- Removed the fee calculation logic as 3% slippage is handled on the frontend.

Auditor's Verdict

After a line by line manual analysis and automated review, Guardian Audits has concluded that:

- Bridges' smart contracts have a **LOW RISK SEVERITY**
- Bridges' smart contracts have an **ACTIVE OWNERSHIP**
- Bridges' smart contract owner has multiple "write" privileges. Centralization risk correlated to the active ownership is **MEDIUM**

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 Guardian 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. Guardian’s position is that each company and individual are responsible for their own due diligence and continuous security. Guardian’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 Guardian is 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, Guardian does not guarantee the explicit security of the audited smart contract, regardless of the verdict.

About Guardian Audits

Founded in 2022 by DeFi experts, Guardian Audits is a leading audit firm in the DeFi smart contract space. With every audit report, Guardian Audits upholds best-in-class security while achieving our mission to relentlessly secure DeFi.

To learn more, visit <https://guardianaudits.com>

To view our audit portfolio, visit <https://github.com/guardianaudits>

To book an audit, message <https://t.me/guardianaudits>