G AUDIT

Shibarium

SMART CONTRACT AUDIT

Audit Date: 06/14/2022

By: GAudit.org

DISCLAIMER

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. To get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and GAudit and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (GAudit) owe no duty of care towards you or any other person, nor does GAudit make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and GAudit hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, GAudit hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against GAudit, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Auditing Strategy and Technique Applied

Throughout the review process, care was taken to evaluate the smart contract for security-related issues, code quality, and adherence to specification and best practices. Reviewed line-by-line by our team of expert contract auditors and developers, documenting issues as they were discovered.

Methodology

The auditing process follows a routine series of steps:

1) Code review that includes the following:
    a. Review of the specifications, sources, and instructions provided to GAudit to make sure we understand the size, scope, and functionality of the smart contract.
    b. Manual review of code, which is the process of reading source code line-by-line to identify potential vulnerabilities.
    c. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to GAudit describe
2) Testing and automated analysis that includes the following:
    a. Test coverage analysis, which is the process of determining whether the test cases cover the code fully and how much code is exercised when we run those test cases.
    b. Symbolic execution, which is analyzing a program to determine what inputs causes each part of a program to execute.
3) Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4) Specific, itemized, actionable recommendations to help you take steps to secure your smart contracts.

Contract Details

Project Name	Shibarium
Contract Name	Shibarium
Blockchain	Binance Smart Chain Main Net (SHIB)
Contract Address	0xFb60A89a9AE1194Bbe38fD3E1d4e50d398A8c00b
Language	Solidity
Compiler Version	v0.8.7+commit.e28d00a7

Contract Wallets

Creator	0xCEA39B6BD2Ad5Dab19CB68abE90d4a45493EDCc3
ceoAddr	0xcea39b6bd2ad5dab19cb68abe90d4a45493edcc3
devAddr	0x57b1ecfb40db8a661a372e23c947b3c04c61fdf7

Project Links

Contract	https://bscscan.com/address/0xFb60A89a9AE1194Bbe38fD3E1d4e50d398A8c00b#code
Website	https://shibarium.lol/
Whitepaper	https://shibarium.lol/Whitepaper.pdf
Telegram	https://t.me/shibarium_dapp
Twitter	https://twitter.com/ShibariumL

Logo

Risk & Vulnerability

Risk represents the probability that a source-threat will exploit vulnerability, and the impact of that event on the organization or system. Risk Level is computed based on Common Vulnerability Scoring System (CVSS) version 3.0.

Level	Value	Vulnerability	Risk (Required Action)
Critical	9-10	A vulnerability that can disrupt the contract functioning in several scenarios or creates a risk that the contract may be broken.	Immediate action to reduce risk level.
High	7-9	A vulnerability that affects the desired outcome when using a contract or provides the opportunity to use a contract in an unintended way.	Implementation of corrective actions as soon as possible.
Medium	4-7	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.	Implementation of corrective actions in a certain period.
Low	2-4	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
Very Low	0-2	A vulnerability that have informational character but is not affecting any of the code.	An observation that does not determine a level of risk.

Metrics

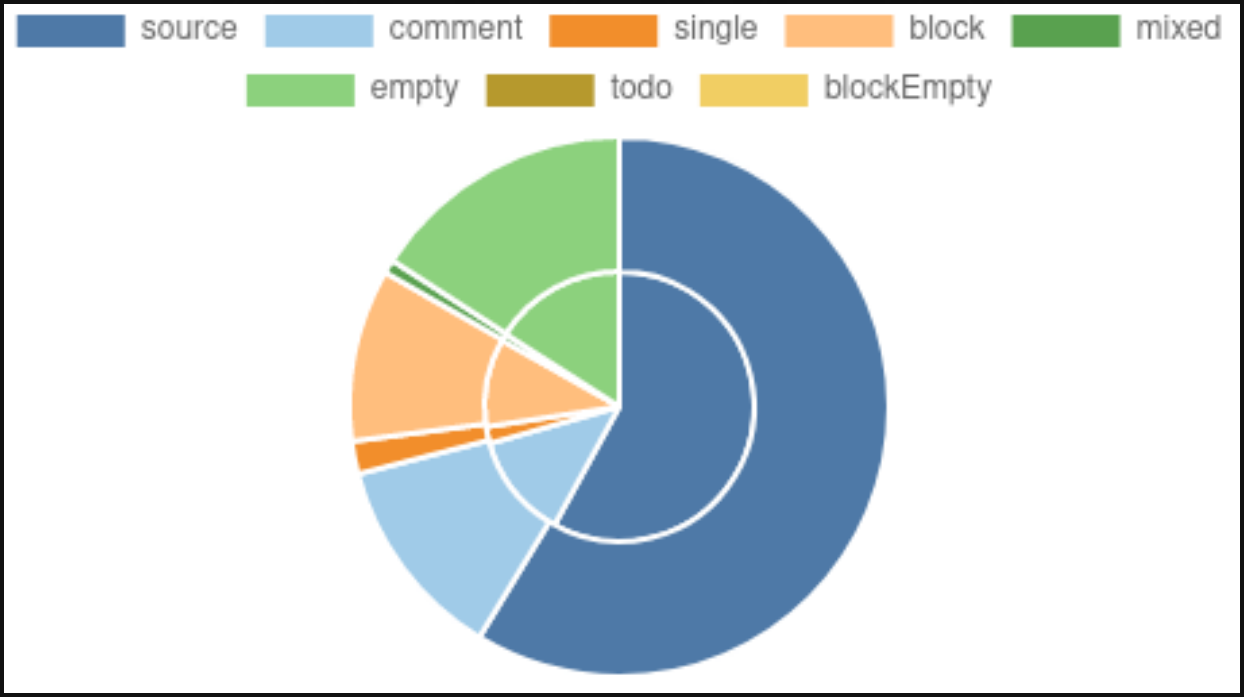
Source Units in Scope

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
	REPORTS/25/shibarium.sol	4	2	525	510	345	75	307	
	Totals	**4**	**2**	**525**	**510**	**345**	**75**	**307**	

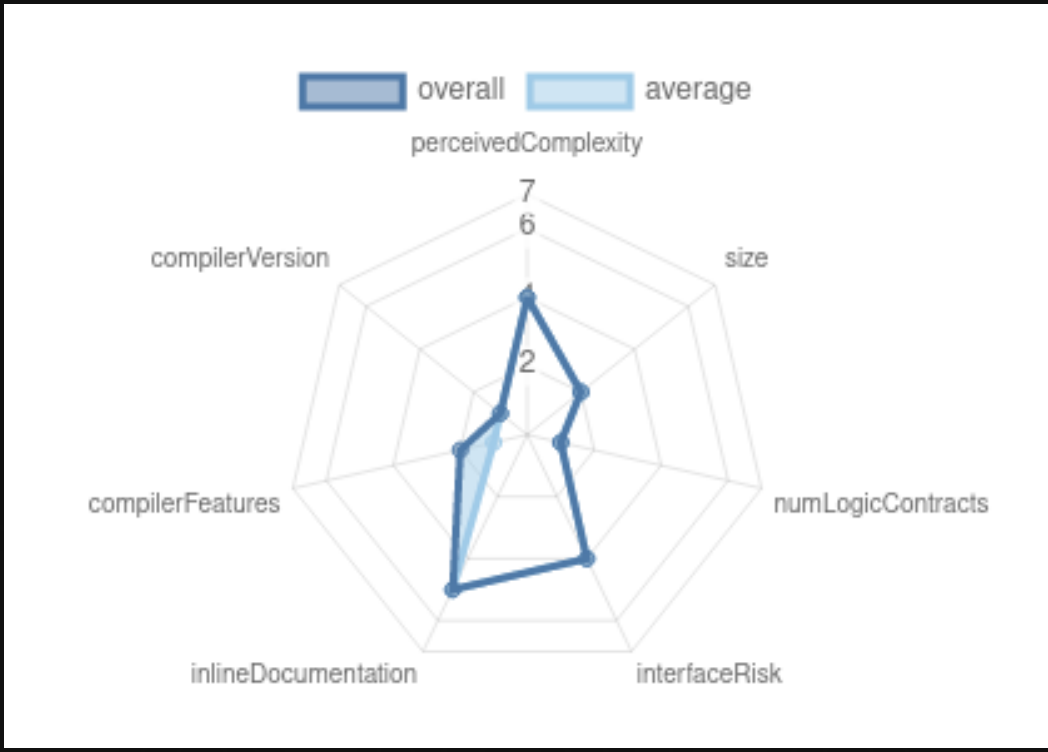
Legend: [-]

- **Lines**: total lines of the source unit
- **nLines**: normalized lines of the source unit (e.g. normalizes functions spanning multiple lines)
- **nSLOC**: normalized source lines of code (only source-code lines; no comments, no blank lines)
- **Comment Lines**: lines containing single or block comments
- **Complexity Score**: a custom complexity score derived from code statements that are known to introduce code complexity (branches, loops, calls, external interfaces, ...)

Source Lines



Risk Level



Components

Contracts	Libraries	Interfaces	Abstract
2	2	2	0

State Variables

Total	🌐Public
28	15

Exposed Functions

🌐Public	💰Payable
33	1

External	Internal	Private	Pure	View
9	23	1	5	27

Capabilities

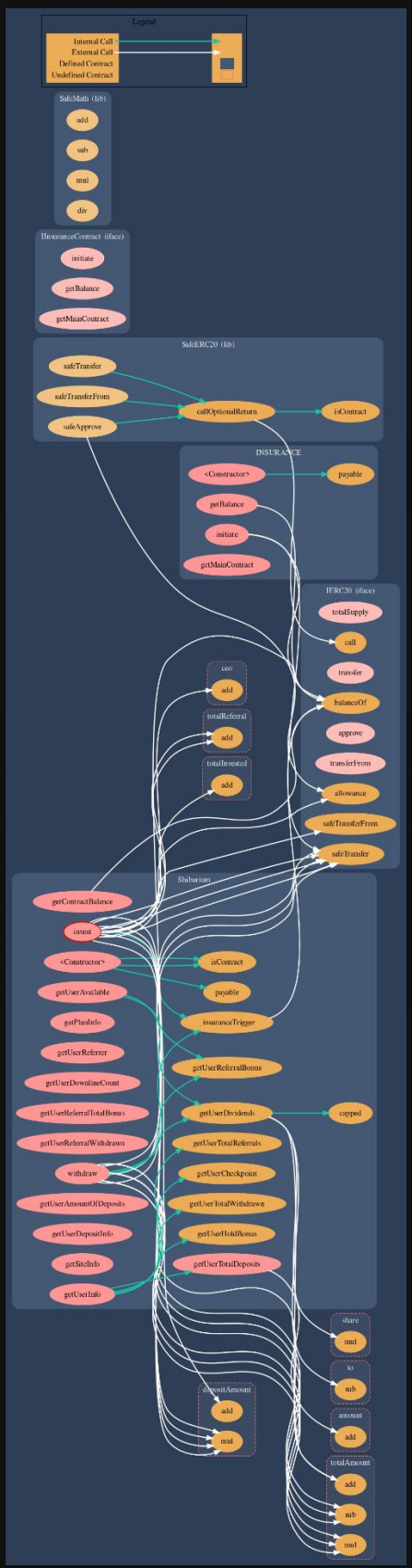
Solidity Versions observed	🧪 Experimental Features	💰 Can Receive Funds	🖥️ Uses Assembly	💣 Has Destroyable Contracts
`0.8.7`		`yes`	`yes` (2 asm blocks)	

📤 Transfers ETH	⚡ Low-Level Calls	👥 DelegateCall	🧮 Uses Hash Functions	📕 ECRecover	🌀 New/Create/Create2
					`yes` → `NewContract:INSURANCE`

♻️ TryCatch	Σ Unchecked

Inheritance Graph

Call Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
SafeMath (lib)
add
sub
mul
div
IInsuranceContract (iface)
initiate
getBalance
getMainContract
SafeERC20 (lib)
safeTransfer
safeTransferFrom
safeApprove
callOptionalReturn
isContract
INSURANCE
<Constructor>
payable
getBalance
initiate
getMainContract
IERC20 (iface)
totalSupply
call
transfer
balanceOf
approve
transferFrom
allowance
safeTransferFrom
safeTransfer
ceo
add
totalReferral
add
totalInvested
add
Shibarium
getContractBalance
invest
<Constructor>
getUserAvailable
getPlanInfo
getUserReferrer
getUserDownlineCount
getUserReferralTotalBonus
getUserReferralWithdrawn
withdraw
getUserAmountOfDeposits
getUserDepositInfo
getSiteInfo
getUserInfo
isContract
payable
insuranceTrigger
getUserReferralBonus
getUserDividends
capped
getUserTotalReferrals
getUserCheckpoint
getUserTotalWithdrawn
getUserHoldBonus
getUserTotalDeposits
depositAmount
add
mul
share
mul
to
sub
amount
add
totalAmount
add
sub
mul

Write Functions

Symbol	Meaning
🔴	Function can modify state
💵	Function is payable

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
IERC20	Interface			
└	totalSupply	External !		NO !
└	balanceOf	External !		NO !
└	transfer	External !	🔴	NO !
└	allowance	External !		NO !
└	approve	External !	🔴	NO !
└	transferFrom	External !	🔴	NO !
SafeERC20	Library			
└	safeTransfer	Internal 🔒	🔴	
└	safeTransferFrom	Internal 🔒	🔴	
└	safeApprove	Internal 🔒	🔴	
└	callOptionalReturn	Private 🔐	🔴	
└	isContract	Internal 🔒		
IInsuranceContract	Interface			
└	initiate	External !	🔴	NO !
└	getBalance	External !		NO !
└	getMainContract	External !		NO !
INSURANCE	Implementation			
└		Public !	🔴	NO !
└	initiate	Public !	🔴	NO !
└	getBalance	Public !		NO !
└	getMainContract	Public !		NO !
Shibarium	Implementation			
└		Public !	🔴	NO !
└	invest	Public !	💵	NO !
└	withdraw	Public !	🔴	NO !
└	_insuranceTrigger	Internal 🔒	🔴	
└	getContractBalance	Public !		NO !
└	getPlanInfo	Public !		NO !
└	getUserDividends	Public !		NO !
└	capped	Public !		NO !
└	getUserTotalWithdrawn	Public !		NO !
└	getUserCheckpoint	Public !		NO !
└	getUserReferrer	Public !		NO !
└	getUserDownlineCount	Public !		NO !
└	getUserTotalReferrals	Public !		NO !
└	getUserReferralBonus	Public !		NO !
└	getUserReferralTotalBonus	Public !		NO !
└	getUserReferralWithdrawn	Public !		NO !
└	getUserAvailable	Public !		NO !
└	getUserAmountOfDeposits	Public !		NO !
└	getUserTotalDeposits	Public !		NO !
└	getUserDepositInfo	Public !		NO !
└	getUserHoldBonus	Public !		NO !
└	getSiteInfo	Public !		NO !
└	getUserInfo	Public !		NO !
└	isContract	Internal 🔒		
SafeMath	Library			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		

Conclusion

High Issue

ROI – The system used is called ROI and must be considered as high-risk

User principal deposits cannot be withdrawn, user can get dividends and referral commission. Dividends are paid from deposits of other users. Do always invest with proper investigation and knowledge.

Our description of Functionality

smart contract provides the opportunity to invest from any amount in SHIB in the contract and get daily profit on investment if the contract balance has enough funds for payment.

All the dividends are calculated at the moment of request and are available for withdrawal every 24 hours.

Each subsequent Deposit is kept separately in the contract, to maintain the payment amount for each Deposit.

Launch Date

Based on the audit date (June 14, 2022) the project is launched.

The launch date was June 08, 2022 at 21:11:19 UTC.

Contract Owners Fee

Deposit Fee	8%
Withdraw Fee	5%

Investment Plan(s)

Total Plans: 1

Plan	Total Return	Daily Profit	Days
0	270%	9.0%	30

The minimum deposit amount is 10 SHIB

The maximum deposit amount is 5,000,000 SHIB

Users can withdraw only every 24 hours

Withdraw Bonus and Rules

For every withdrawal:

55% will be sent to the user's wallet

20% will be set as a bonus

20% will be sent to the insurance contract

5% will be sent to the owner's wallet

Users can withdraw only once a day

The bonus will be added to the deposit amount of the next stake

The deposit amount should be equal to or greater than the previous one

Insurance System

A second contract manages the insurance system

20% of all withdrawals sent to the insurance contract

If the main contract balance decreases less than 25% of the highest contract balance amount in the last 7 days, the insurance contract balance will be returned to the main contract

Referral System

Level	Commission
1	12%

Referral should be an active user; it means the referral address has at least one deposit

The referrer is specified once at the time of the first deposit and is assigned to the user without the possibility of changing. From each subsequent Deposit, the referrer will get their percentage.

Owner is set as the upline if there is no valid upline referrer

Additional and Updated Audit Information May be Found at GAudit.org

GAudit Information

Website	GAudit.org
Telegram Group	@GAudit_org
Admin Telegram	@GAudit
Twitter	@GAudit_org
Email	contact@GAudit.org

G

AUDIT