RD
AUDITORS

Shade Protocol, Code Review and Security Analysis Report

Customer: Shade Protocol
Prepared on: 7th March 2022
Platform: Secret Network
Language: Rust

rdauditors.com

Table of Contents

Disclaimer

This document may contain confidential information about its systems and intellectual property of the customer as well as information about potential vulnerabilities and methods of their exploitation.

The report containing confidential information can be used internally by the customer or it can be disclosed publicly after all vulnerabilities are fixed - upon the decision of the customer.

Document

Name	Smart Contract Code Review and Security Analysis Report of Shade Protocol
Platform	Secret Network
File 1	contract.rs
MD5 hash	ABA1A4B641DFD15D88859E9B79E13988
SHA256 hash	B6D82ABCEF4DAED4BF44D865141DBB19E30635C1F49394EDFED5BC309BAF030D
File 2	msg.rs
MD5 hash	01D947AC88C09A0B36A39D13AAB4EC3A
SHA256 hash	B6D82ABCEF4DAED4BF44D865141DBB19E30635C1F49394EDFED5BC309BAF030D
File 3	permit.rs
MD5 hash	8319635E1DC715802F8FD18A0A2F6C7C
SHA256 hash	01FE64E1FDDF8364E497307B2D8C940E40DCE3482EFD1F73E8E557C9AC5BB440
File 5	rand.rs

MD5 hash	72371B78AE659E86AD981F6103AEEB3F
SHA256 hash	D065A9D884EE7D02A1E75EC8BEFE14F02D923EFB8F8F5E70E555E28F4209BF55
File 6	receiver.rs
MD5 hash	4BBD5DBADEB024799A401CB5A5FEE08F
SHA256 hash	3AB97C4C92F180FBB9E4B53AB4282466B6C991D8613A8C1A16A8F9E69DF893AE
File 7	staking.rs
MD5 hash	DBBF7E53AC4E44E781F9449D7ED15E9E
SHA256 hash	8CACF14690FF72094A5CC924FE26DEB9E437021DA7551F3145EF2F8DE26202F4
File 8	state.rs
MD5 hash	842FE2A3982BE0F91D1841BF12C9F2F1
SHA256 hash	2E9FC09F654F310BFE93AC784E2A8C2381847517B4556ECA1409460E87CA8C8A
File 9	transaction_history.rs
MD5 hash	05857D3AA840C4DFEBAED149DF5C2EBB

SHA256 hash	844B22BF68E9902AE39A9B804632543594FDE12521692828B1C30FB032E9708F
File 10	utils.rs
MD5 hash	15CF7E63871FDDEBD335B1DE212B19E9
SHA256 hash	C2E40F8DE92C9DF81B290F6CE34612B67B0F843BBEB592D57B2F3E37B10CC8C8
File 11	viewingKey.rs
MD5 hash	258136E439AA93A9BDF73DD5CE22800F
SHA256 hash	41F7308A6D4A75AF6A0E03B76C429B157ACC2BC4289FDAE090C425BEEBFDEA83
File12	voting.rs
MD5 hash	D87F97292DC6C0F1EE629D5BAD580476
SHA256 hash	7A8AE41B8365BCB5795B326511DA85144F9B2A2F0DCE61009E73947B29F54DC9
Date	7/03/2022

Introduction

RD Auditors (Consultant) were contracted by Shade Protocol (Customer) to conduct a Smart Contracts Code Review and Security Analysis. This report represents the findings of the security assessment of the customer`s smart contracts and its code review conducted between 17th February - 7th March 2022.

This contract consists of twelve files.

Project Scope

The scope of the project is a smart contract. We have scanned this smart contract for commonly known and more specific vulnerabilities, below are those considered (the full list includes but is not limited to):

- Missing signer
- Integer overflow & underflow
- Arbitrary signed program invocation
- Account confusions
- Other Known / Possible vulnerabilities

This is a 'Privacy First' based Secret Network using RUST smart contract as the coding language. It is a relatively modern innovative approach towards applications which are both permissionless and privacy-preserving. The lists of known vulnerabilities are relatively low, however we have checked/tested all possible areas including logical conflict and code flow projections.

Executive Summary

According to the assessment, the customer’s RUST smart contract is **well-secured.**

You are Here

🟥 **Insecure** 🟧 **Poorly Secured** 🟩 **Secure** 🟦 **Well-Secured**

Manual and localized checks are done. All issues were performed by our team, which included the analysis of code functionality, manual audit found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the audit overview section. The general overview is presented in the AS-IS section and all issues found are located in the audit overview section.

We found the following;

Total Issues	0
🟥 Critical	0
🟧 High	0
🟨 Medium	0
🟩 Low	0
🟦 Very Low	0

Code Quality

Please find a link that, within this report, uses RUST libraries specially designed for smart contracts taken from the popular open source.

https://github.com/securesecrets/staking-derivative/tree/master/src

The libraries within this smart contract are part of a logical algorithm. A library is a different type of smart contract that contains reusable code. Once deployed on the blockchain (only once), it is assigned to a specific address and its properties/methods can be reused many times by other contracts.

The Shade Protocol team has provided scenario and unit test codes, which helped to determine the integrity of the code in an automated way.

Overall, the code is almost commented. Commenting provides rich documentation for functions, return variables and more.

Documentation

The hash of that file is mentioned in the table. As mentioned above, It's well commented smart contract code, so anyone can quickly understand the programming flow as well as complex code logic.

Comments are very helpful in understanding the overall architecture of the protocol. It also provides a clear overview of the system components, including helpful details, like the lifetime of the background script.

Use of Dependencies

As per our observation, the libraries and other open source codes are used in this smart contract infrastructure. Those were based on well-known industry standard open source projects and even core code blocks that are written well and systematically.

AS-IS Overview

File And Function Level Report

File:	Contract.rs
Observation:	Passed
Test Report:	Passed
Score:	Passed
Conclusion	Passed

Sl.	Function	Observation	Test Report	Conclusion	Score
1	init	Passed	All Passed	No Issue	Passed
2	Pad_response	Passed	All Passed	No Issue	Passed
3	handle	Passed	All Passed	No Issue	Passed
4	query	Passed	All Passed	No Issue	Passed
5	permit	Passed	All Passed	No Issue	Passed
6	Viewing_keys_queries	Passed	All Passed	No Issue	Passed
7	query_admins	Passed	All Passed	No Issue	Passed
8	query_cattoffs	Passed	All Passed	No Issue	Passed
9	query_msg_limits	Passed	All Passed	No Issue	Passed
10	query_vote_count	Passed	All Passed	No Issue	Passed
11	query_my_prop_vote	Passed	All Passed	No Issue	Passed
12	query_all_runnings	Passed	All Passed	No Issue	Passed
13	query_contract_hist	Passed	All Passed	No Issue	Passed
14	query_my_vote_hist	Passed	All Passed	No Issue	Passed
15	query_tally_info	Passed	All Passed	No Issue	Passed

16	query_contract_hist	Passed	All Passed	No Issue	Passed
17	query_my_vote_hist	Passed	All Passed	No Issue	Passed
18	query_tally_info	Passed	All Passed	No Issue	Passed
19	query_running_count	Passed	All Passed	No Issue	Passed
20	query_transfer_fees	Passed	All Passed	No Issue	Passed
21	query_staking_info	Passed	All Passed	No Issue	Passed
22	query_holdings	Passed	All Passed	No Issue	Passed
23	query_unbondings	Passed	All Passed	No Issue	Passed
24	query_token_info	Passed	All Passed	No Issue	Passed
25	query_contract_status	Passed	All Passed	No Issue	Passed
26	query_transfers	Passed	All Passed	No Issue	Passed
27	query_transactions	Passed	All Passed	No Issue	Passed
28	query_balance	Passed	All Passed	No Issue	Passed
29	try_reopen_voting	Passed	All Passed	No Issue	Passed
30	try_change_cutoffs	Passed	All Passed	No Issue	Passed
31	try_change_msg_lims	Passed	All Passed	No Issue	Passed
32	try_change_interval	Passed	All Passed	No Issue	Passed
33	try_reset_batch	Passed	All Passed	No Issue	Passed
34	try_change_unbond_time	Passed	All Passed	No Issue	Passed
35	try_rebalance	Passed	All Passed	No Issue	Passed
36	try_Vote	Passed	All Passed	No Issue	Passed
37	try_running_count	Passed	All Passed	No Issue	Passed

38	try_mark_closed	Passed	All Passed	No Issue	Passed
39	try_tally	Passed	All Passed	No Issue	Passed
40	try_unbond	Passed	All Passed	No Issue	Passed
41	try_unbond_batch	Passed	All Passed	No Issue	Passed
42	try_Claim_reward	Passed	All Passed	No Issue	Passed
43	try_delegate	Passed	All Passed	No Issue	Passed
44	try_compound	Passed	All Passed	No Issue	Passed
45	try_claim	Passed	All Passed	No Issue	Passed
46	try_panic_unbond	Passed	All Passed	No Issue	Passed
47	try_panic_withdraw	Passed	All Passed	No Issue	Passed
48	try_Update_fees	Passed	All Passed	No Issue	Passed
49	try_stake	Passed	All Passed	No Issue	Passed
50	try_mint_impl	Passed	All Passed	No Issue	Passed
51	try_set_key	Passed	All Passed	No Issue	Passed
52	try_set_key	Passed	All Passed	No Issue	Passed
53	try_create_key	Passed	All Passed	No Issue	Passed
54	set_contract_status	Passed	All Passed	No Issue	Passed
55	query_allowance	Passed	All Passed	No Issue	Passed
56	try_change_fees	Passed	All Passed	No Issue	Passed
57	try_transfer_impl	Passed	All Passed	No Issue	Passed
58	try_transfer	Passed	All Passed	No Issue	Passed
59	try_batch_transfer	Passed	All Passed	No Issue	Passed
60	try_add_receiver_api_Callback	Passed	All Passed	No Issue	Passed
61	try_send_impl	Passed	All Passed	No Issue	Passed

62	try_expose_balance	Passed	All Passed	No Issue	Passed
63	try_send	Passed	All Passed	No Issue	Passed
64	try_batch_send	Passed	All Passed	No Issue	Passed
65	try_register_receive	Passed	All Passed	No Issue	Passed
66	insufficient_allowance	Passed	All Passed	No Issue	Passed
67	use_allowance	Passed	All Passed	No Issue	Passed
68	try_transfer_from_impl	Passed	All Passed	No Issue	Passed
69	try_transfer_from	Passed	All Passed	No Issue	Passed
70	try_batch_transfer_from	Passed	All Passed	No Issue	Passed
71	try_send_from_impl	Passed	All Passed	No Issue	Passed
72	try_send_from	Passed	All Passed	No Issue	Passed
73	try_batch_send_from	Passed	All Passed	No Issue	Passed
74	try_burn_from	Passed	All Passed	No Issue	Passed
75	try_batch_burn_from	Passed	All Passed	No Issue	Passed
76	try_increase_allowance	Passed	All Passed	No Issue	Passed
77	try_decrease_allowance	Passed	All Passed	No Issue	Passed
78	try_set_Vals	Passed	All Passed	No Issue	Passed
79	add_admins	Passed	All Passed	No Issue	Passed
80	remove_admins	Passed	All Passed	No Issue	Passed
81	try_burn	Passed	All Passed	No Issue	Passed

82	Reform_transfer	Passed	All Passed	No Issue	Passed
83	revoke_permit	Passed	All Passed	No Issue	Passed
84	is_admin_read_only	Passed	All Passed	No Issue	Passed
85	check_if_admin_read_only	Passed	All Passed	No Issue	Passed
86	is_admin	Passed	All Passed	No Issue	Passed
87	check_if_admin	Passed	All Passed	No Issue	Passed
88	is_Valid_name	Passed	All Passed	No Issue	Passed
89	check_contract_status	Passed	All Passed	No Issue	Passed
90	claim_rewards	Passed	All Passed	No Issue	Passed
91	gen_delegate_msg	Passed	All Passed	No Issue	Passed
92	get_rewards	Passed	All Passed	No Issue	Passed
93	get_delegatable	Passed	All Passed	No Issue	Passed
94	get_bonded_scrt	Passed	All Passed	No Issue	Passed
95	update_reserves	Passed	All Passed	No Issue	Passed
96	handle_available_scrt	Passed	All Passed	No Issue	Passed
97	filte_vals	Passed	All Passed	No Issue	Passed
98	sort_vals	Passed	All Passed	No Issue	Passed
99	query_available_scrt	Passed	All Passed	No Issue	Passed
100	delegate	Passed	All Passed	No Issue	Passed
101	sort_for_dels	Passed	All Passed	No Issue	Passed
102	Process_pending_Unbonds	Passed	All Passed	No Issue	Passed
103	get_Prng	Passed	All Passed	No Issue	Passed

104	init_helper	Passed	All Passed	No Issue	Passed
105	my_mock_dependencies	Passed	All Passed	No Issue	Passed
106	_claim_rewards	Passed	All Passed	No Issue	Passed
107	unbond	Passed	All Passed	No Issue	Passed
108	delegate	Passed	All Passed	No Issue	Passed
109	_get_balance	Passed	All Passed	No Issue	Passed
110	_get_delegations	Passed	All Passed	No Issue	Passed
111	Update_balance	Passed	All Passed	No Issue	Passed
112	remove_Validators	Passed	All Passed	No Issue	Passed
113	_add_Validator	Passed	All Passed	No Issue	Passed
114	raw_query	Passed	All Passed	No Issue	Passed
115	query	Passed	All Passed	No Issue	Passed
116	custom_query	Passed	All Passed	No Issue	Passed
117	query_validators	Passed	All Passed	No Issue	Passed
118	query_all_delegations	Passed	All Passed	No Issue	Passed
119	query_balance	Passed	All Passed	No Issue	Passed
120	init_helper_with_Validators	Passed	All Passed	No Issue	Passed
121	init_helper_with_config	Passed	All Passed	No Issue	Passed
122	extract_error_msg	Passed	All Passed	No Issue	Passed
123	ensure_sucess	Passed	All Passed	No Issue	Passed

File:	msg.rs
Observation:	Passed
Test Report:	Passed
Score:	Passed
Conclusion	Passed

Sl.	Function	Observation	Test Report	Conclusion	Score
1	get_validation_params	Passed	All Passed	No Issue	Passed
2	new	Passed	All Passed	No Issue	Passed
3	state_level_to_u8	Passed	All Passed	No Issue	Passed
4	u8_to_status_level	Passed	All Passed	No Issue	Passed
5	Vote_option_to_u8	Passed	All Passed	No Issue	Passed
6	u8_vote_option_to_string	Passed	All Passed	No Issue	Passed
7	space_pad	Passed	All Passed	No Issue	Passed

File:	Permit.rs
Observation:	Passed
Test Report:	Passed
Score:	Passed
Conclusion	Passed

Sl.	Function	Observation	Test Report	Conclusion	Score
1	check_token	Passed	All Passed	No Issue	Passed
2	check_permission	Passed	All Passed	No Issue	Passed
3	from_Params	Passed	All Passed	No Issue	Passed
4	new	Passed	All Passed	No Issue	Passed
5	new	Passed	All Passed	No Issue	Passed
6	from_context	Passed	All Passed	No Issue	Passed
7	from_Params	Passed	All Passed	No Issue	Passed
8	is_permit_revoked	Passed	All Passed	No Issue	Passed
9	revoke_Permit	Passed	All Passed	No Issue	Passed
10	validate	Passed	All Passed	No Issue	Passed
11	pubkey_to_account	Passed	All Passed	No Issue	Passed

File:	rand.rs
Observation:	Passed
Test Report:	Passed
Score:	Passed
Conclusion	Passed

Sl.	Function	Observation	Test Report	Conclusion	Score
1	sha_256	Passed	All Passed	No Issue	Passed
2	new	Passed	All Passed	No Issue	Passed
3	_bytes	Passed	All Passed	No Issue	Passed
4	next_u32	Passed	All Passed	No Issue	Passed
5	get_rng	Passed	All Passed	No Issue	Passed
6	extend_entropy	Passed	All Passed	No Issue	Passed

File:	receivert.rs
Observation:	Passed
Test Report:	Passed
Score:	Passed
Conclusion	Passed

Sl.	Function	Observation	Test Report	Conclusion	Score
1	new	Passed	All Passed	No Issue	Passed
2	into_binary	Passed	All Passed	No Issue	Passed
3	into_cosmos_msg	Passed	All Passed	No Issue	Passed
4	new	Passed	All Passed	No Issue	Passed
5	into_binary	Passed	All Passed	No Issue	Passed
6	into_cosmos_msg	Passed	All Passed	No Issue	Passed

File:	Staking.rs
Observation:	Passed
Test Report:	Passed
Score:	Passed
Conclusion	Passed

Sl.	Function	Observation	Test Report	Conclusion	Score
1	to_humanized	Passed	All Passed	No Issue	Passed
2	save	Passed	All Passed	No Issue	Passed
3	remove	Passed	All Passed	No Issue	Passed

4	load	Passed	All Passed	No Issue	Passed
5	may_load	Passed	All Passed	No Issue	Passed

File:	State.rs
Observation:	Passed
Test Report:	Passed
Score:	Passed
Conclusion	Passed

Sl.	Function	Observation	Test Report	Conclusion	Score
1	from_storage	Passed	All Passed	No Issue	Passed
2	as_readonly	Passed	All Passed	No Issue	Passed
3	constants	Passed	All Passed	No Issue	Passed
4	transfer_fees	Passed	All Passed	No Issue	Passed
5	total_supply	Passed	All Passed	No Issue	Passed
6	tx_count	Passed	All Passed	No Issue	Passed
7	ser_bin_data	Passed	All Passed	No Issue	Passed
8	deser_bin_data	Passed	All Passed	No Issue	Passed
9	set_bin_data	Passed	All Passed	No Issue	Passed
10	get_bin_data	Passed	All Passed	No Issue	Passed
11	from_storage	Passed	All Passed	No Issue	Passed
12	as_readonly	Passed	All Passed	No Issue	Passed
13	constants	Passed	All Passed	No Issue	Passed
14	set_constants	Passed	All Passed	No Issue	Passed
15	transfer_fees	Passed	All Passed	No Issue	Passed

16	set_transfer_fees	Passed	All Passed	No Issue	Passed
17	total_supply	Passed	All Passed	No Issue	Passed
18	set_total_supply	Passed	All Passed	No Issue	Passed
19	contract_status	Passed	All Passed	No Issue	Passed

File:	transaction_history.rs
Observation:	Passed
Test Report:	Passed
Score:	Passed
Conclusion	Passed

Sl.	Function	Observation	Test Report	Conclusion	Score
1	into_humanized	Passed	All Passed	No Issue	Passed
2	to_u8	Passed	All Passed	No Issue	Passed
3	from_u8	Passed	All Passed	No Issue	Passed
4	transfer	Passed	All Passed	No Issue	Passed
5	mint	Passed	All Passed	No Issue	Passed
6	burn	Passed	All Passed	No Issue	Passed
7	into_humanized	Passed	All Passed	No Issue	Passed
8	new	Passed	All Passed	No Issue	Passed
9	into_humanized	Passed	All Passed	No Issue	Passed
10	from_stored_legacy_transfer	Passed	All Passed	No Issue	Passed

11	increment_tx_count	Passed	All Passed	No Issue	Passed
12	store_transfer	Passed	All Passed	No Issue	Passed
13	store_mint	Passed	All Passed	No Issue	Passed
14	store_burn	Passed	All Passed	No Issue	Passed
15	append_tx	Passed	All Passed	No Issue	Passed
16	append_transfer	Passed	All Passed	No Issue	Passed
17	get_txs	Passed	All Passed	No Issue	Passed
18	get_transfers	Passed	All Passed	No Issue	Passed

File:	utils.rs
Observation:	Passed
Test Report:	Passed
Score:	Passed
Conclusion	Passed

Sl.	Function	Observation	Test Report	Conclusion	Score
1	ct_slice_compare	Passed	All Passed	No Issue	Passed
2	create_hashed_password	Passed	All Passed	No Issue	Passed

File:	viewingkey.rs
Observation:	Passed
Test Report:	Passed
Score:	Passed
Conclusion	Passed

Sl.	Function	Observation	Test Report	Conclusion	Score
1	check_viewing_key	Passed	All Passed	No Issue	Passed
2	new	Passed	All Passed	No Issue	Passed
3	to_hashed	Passed	All Passed	No Issue	Passed
4	as_bytes	Passed	All Passed	No Issue	Passed
5	fmt	Passed	All Passed	No Issue	Passed

File:	voting.rs
Observation:	Passed
Test Report:	Passed
Score:	Passed
Conclusion	Passed

Sl.	Function	Observation	Test Report	Conclusion	Score
1	create_vote_count	Passed	All Passed	No Issue	Passed
2	create_vote_count	Passed	All Passed	No Issue	Passed
3	create_vote_history	Passed	All Passed	No Issue	Passed

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to lost tokens etc.
High	High level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g. public access to crucial functions.
Medium	Medium level vulnerabilities are important to fix; however, they cannot lead to lost tokens.
Low	Low level vulnerabilities are most related to outdated, unused etc. These code snippets cannot have a significant impact on execution.
Lowest Code Style/ Best Practice	Lowest level vulnerabilities, code style violations and information statements cannot affect smart contract execution and can be ignored.

Audit Findings

Critical:

No critical severity vulnerabilities were found.

High:

No high severity vulnerabilities were found.

Medium:

No medium severity vulnerabilities were found.

Low:

No low severity vulnerabilities were found.

Very Low

No very low severity vulnerabilities were found.

Conclusion

We have used all possible tests based on the given object. The contract is written systematically, so it is now ready to go for production.

Since possible test cases can be unlimited and developer level documentation (SRS, Architecture, code flow diagram with function level description) not provided, for such an extensive smart contract protocol, we provide no such guarantee of future outcomes nor publicly unknown vulnerabilities (which may be detected in the future).

We have used all the latest static tools and manual observations to cover maximum possible test cases to scan everything.

The security state of the reviewed contract is now “well-secured”.

Note For contract User

The Software Requirement Specification and developer level documentation were not provided, so our test cases/observations were limited in scope.

We did not perform an extensive audit, it was not under the scope of the request. We do not guarantee any discrepancy raised from any dependent library/macros (external objects) included and/or tempered.

Technical auditing does not guarantee the project's ethical side. Please do your due diligence before investing. Our audit report is never an investment advice.

Our Methodology

We like to work with a transparent process and make our reviews a collaborative effort. The goals of our security audits are to improve the quality of systems we review and aim for sufficient remediation to help protect users. The following is the methodology we use in our security audit process.

Manual Code Review

In manually reviewing all of the code, we look for any potential issues with code logic, error handling, protocol and header parsing, cryptographic errors, and random number generators. We also watch for areas where more defensive programming could reduce the risk of future mistakes and speed up future audits. Although our primary focus is on the in-scope code, we examine dependency code and behavior when it is relevant to a particular line of investigation.

Vulnerability Analysis

Our audit techniques included manual code analysis, user interface interaction, and whitebox penetration testing. We look at the project's web site to get a high level understanding of what functionality the software under review provides. We then meet with the developers to gain an appreciation of their vision of the software. We install and use the relevant software, exploring the user interactions and roles. While we do this, we brainstorm threat models and attack surfaces. We read design documentation, review other audit results, search for similar projects, examine source code dependencies, skim open issue tickets, and generally investigate details other than the implementation.

Documenting Results

We follow a conservative, transparent process for analyzing potential security vulnerabilities and seeing them through successful remediation. Whenever a potential issue is discovered, we immediately create an Issue entry for it in this document, even though we have not yet verified the feasibility and impact of the issue. This process is conservative because we document our suspicions early even if they are later shown to not represent exploitable vulnerabilities.

We generally follow a process of first documenting the suspicion with unresolved questions, then confirming the issue through code analysis, live experimentation, or automated tests. Code analysis is the most tentative, and we strive to provide test code, log captures, or screenshots demonstrating our confirmation. After this we analyze the feasibility of an attack in a live system.

Suggested Solutions

We search for immediate mitigations that live deployments can take, and finally we suggest the requirements for remediation engineering for future releases. The mitigation and remediation recommendations should be scrutinized by the developers and deployment engineers, and successful mitigation and remediation is an ongoing collaborative process after we deliver our report, and before the details are made public.

Disclaimers

The smart contracts given for audit have been analyzed in accordance with the best industry practices at the date of this report, in relation to: 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).

Because the total number of test cases are unlimited, the audit makes no statements or warranties on the security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the code, bugfree status or any other statements of the contract. 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 security of smart contracts.

Technical Disclaimer

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

RD
AUDITORS

Email: info@rdauditors.com

Website: www.rdauditors.com