



PALADIN
BLOCKCHAIN SECURITY

Smart Contract Security Assessment

Final Report

For LayerZero
(Solana OFT)

09 December 2025



paladinsec.co



info@paladinsec.co

Table of Contents

Table of Contents	2
Disclaimer	6
1 Overview	7
1.1 Summary	7
1.2 Contracts Assessed	8
1.3 Findings Summary	10
1.3.1 lz_receive.rs	11
1.3.2 errors.rs	11
1.3.3 lib.rs	11
1.3.4 events.rs	11
1.3.5 types.rs	11
1.3.6 msg_codec.rs	12
1.3.7 compose_msg_codec.rs	12
1.3.8 state/mod.rs	12
1.3.9 oft.rs	12
1.3.10 token.rs	12
1.3.11 peer_config.rs	12
1.3.12 rate_limiter.rs	13
1.3.13 instructions/mod.rs	13
1.3.14 send.rs	13
1.3.15 lz_receive_type_info.rs	13
1.3.16 lz_receive_types_v2.rs	13
1.3.17 quote_send.rs	13
1.3.18 quote_of.rs	14
1.3.19 instructions/privileged/mod.rs	14
1.3.20 init.rs	14
1.3.21 set_admin.rs	14

1.3.22 set_alt.rs	14
1.3.23 set_peer_config.rs	15
1.3.24 register_token.rs	15
1.3.25 sync_mint_authority.rs	15
1.3.26 configure_rate_limiter.rs	15
1.3.27 close_rate_limiter.rs	15
1.3.28 set_global_paused.rs	15
1.3.29 set_id_paused.rs	16
2 Findings	17
2.1 lz_receive.rs	17
2.1.1 Issues & Recommendations	18
2.2 errors.rs	23
2.2.1 Issues & Recommendations	23
2.3 lib.rs	24
2.3.1 Privileged Functions	24
2.3.2 Issues & Recommendations	24
2.4 events.rs	25
2.4.1 Issues & Recommendations	25
2.5 types.rs	26
2.5.1 Issues & Recommendations	26
2.6 msg_codec.rs	27
2.6.1 Issues & Recommendations	27
2.7 compose_msg_codec.rs	28
2.7.1 Issues & Recommendations	28
2.8 state/mod.rs	29
2.8.1 Issues & Recommendations	29
2.9 oft.rs	30
2.9.1 Issues & Recommendations	30
2.10 token.rs	31
2.10.1 Issues & Recommendations	31

2.11 peer_config.rs	32
2.11.1 Issues & Recommendations	32
2.12 rate_limiter.rs	33
2.12.1 Issues & Recommendations	33
2.13 instructions/mod.rs	34
2.13.1 Issues & Recommendations	34
2.14 send.rs	35
2.14.1 Issues & Recommendations	35
2.15 lz_receive_type_info.rs	36
2.15.1 Issues & Recommendations	36
2.16 lz_receive_types_v2.rs	37
2.16.1 Issues & Recommendations	37
2.17 quote_send.rs	38
2.17.1 Issues & Recommendations	38
2.18 quote_of.rs	39
2.18.1 Issues & Recommendations	40
2.19 instructions/privileged/mod.rs	41
2.19.1 Issues & Recommendations	41
2.20 init.rs	42
2.20.1 Privileged Functions	42
2.20.2 Issues & Recommendations	43
2.21 set_admin.rs	44
2.21.1 Privileged Functions	44
2.21.2 Issues & Recommendations	44
2.22 set_alt.rs	45
2.22.1 Privileged Functions	45
2.22.2 Issues & Recommendations	45
2.23 set_peer_config.rs	46
2.23.1 Privileged Functions	46
2.23.2 Issues & Recommendations	46

2.24 register_token.rs	47
2.24.1 Privileged Functions	47
2.24.2 Issues & Recommendations	47
2.25 sync_mint_authority.rs	48
2.25.1 Privileged Functions	48
2.25.2 Issues & Recommendations	48
2.26 configure_rate_limiter.rs	49
2.26.1 Privileged Functions	49
2.26.2 Issues & Recommendations	49
2.27 close_rate_limiter.rs	50
2.27.1 Privileged Functions	50
2.27.2 Issues & Recommendations	50
2.28 set_global_paused.rs	51
2.28.1 Privileged Functions	51
2.28.2 Issues & Recommendations	51
2.29 set_id_paused.rs	52
2.29.1 Privileged Functions	52
2.29.2 Issues & Recommendations	52



Disclaimer

Paladin Blockchain Security ("Paladin") has conducted an independent audit to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the codes that were provided for the scope of this audit. This audit report does not constitute agreement, acceptance or advocacy for the Project that was audited, and users relying on this audit report should not consider this as having any merit for financial advice in any shape, form or nature. The contracts audited do not account for any economic developments that may be pursued by the Project in question, and that the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are completely free of exploits, bugs, vulnerabilities or deprecation of technologies. Further, this audit report shall not be disclosed nor transmitted to any persons or parties on any objective, goal or justification without due written assent, acquiescence or approval by Paladin.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence regardless of the findings presented in this report. Information is provided 'as is', and Paladin is under no covenant to the completeness, accuracy or solidity of the contracts audited. In no event will Paladin or its partners, employees, agents or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions and/or actions with regards to the information provided in this audit report.

Cryptocurrencies and any technologies by extension directly or indirectly related to cryptocurrencies are highly volatile and speculative by nature. All reasonable due diligence and safeguards may yet be insufficient, and users should exercise considerable caution when participating in any shape or form in this nascent industry.

The audit report has made all reasonable attempts to provide clear and articulate recommendations to the Project team with respect to the rectification, amendment and/or revision of any highlighted issues, vulnerabilities or exploits within the contracts provided. It is the sole responsibility of the Project team to sufficiently test and perform checks, ensuring that the contracts are functioning as intended, specifically that the functions therein contained within said contracts have the desired intended effects, functionalities and outcomes of the Project team.

Paladin retains the right to re-use any and all knowledge and expertise gained during the audit process, including, but not limited to, vulnerabilities, bugs, or new attack vectors. Paladin is therefore allowed and expected to use this knowledge in subsequent audits and to inform any third party, who may or may not be our past or current clients, whose projects have similar vulnerabilities. Paladin is furthermore allowed to claim bug bounties from third-parties while doing so.

1 Overview

This report has been prepared for LayerZero's contracts on the Solana network. Paladin provides a user-centred examination of the smart contracts to look for vulnerabilities, logic errors or other issues from both an internal and external perspective.

1.1 Summary

Project Name	LayerZero
URL	https://layerzero.network/
Platform	Solana
Language	Rust
Preliminary Contracts	https://github.com/ondoprotocol/gm-lz-bridge/tree/5bb78825c3ab2b27685c79bacc36c3963e6abecf/solana/programs/oft/src https://github.com/LayerZero-Labs/LayerZero-v2/pull/178 (PR changes only)
Resolution	https://github.com/ondoprotocol/gm-lz-bridge/tree/ab928dd831f2b5d8e7c47233598b5cbb77fcb4d8

1.2 Contracts Assessed

- compose_msg_codec.rs
- errors.rs
- events.rs
- lib.rs
- msg_codec.rs
- types.rs
- /state/
 - mod.rs
 - oft.rs
 - peer_config.rs
 - rate_limiter.rs
 - token.rs
- /instructions/
 - lz_receive.rs
 - lz_receive_types_info.rs
 - lz_receive_types_v2.rs
 - mod.rs
 - quote_of.rs
 - quote_send.rs
 - send.rs
- /instructions/privileged
 - close_rate_limiter.rs
 - configure_rate_limiter.rs
 - deregister_token.rs
 - init.rs
 - mod.rs
 - register_token.rs
 - set_admin.rs
 - set_alt.rs

- set_global_paused.rs
- set_id_paused.rs
- set_peer_config.rs
- sync_mint_authority.rs



1.3 Findings Summary

Severity	Found	Resolved	Partially Resolved	Acknowledged (no change made)
● High	0	-	-	0
● Medium	0	-	-	-
● Low	2	-	-	2
● Informational	10	1	-	9
Total	12	1	0	11

Classification of Issues

Severity	Description
● High	Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, or impairment of the contract and its functions. Issues under this classification are recommended to be fixed with utmost urgency.
● Medium	Bugs or issues with that may be subject to exploit, though their impact is somewhat limited. Issues under this classification are recommended to be fixed as soon as possible.
● Low	Effects are minimal in isolation and do not pose a significant danger to the project or its users. Issues under this classification are recommended to be fixed nonetheless.
● Informational	Consistency, syntax or style best practices. Generally pose a negligible level of risk, if any.

1.3.1 lz_receive.rs

ID	Severity	Summary	Status
01	LOW	Missing pause validation in lz_receive()	ACKNOWLEDGED
02	INFO	Incorrect account order when calling Token-2022 mint_to in credit	ACKNOWLEDGED
03	INFO	Unchecked slicing of remaining_accounts	ACKNOWLEDGED
04	INFO	Receivers could potentially attempt to extract rent	ACKNOWLEDGED
05	INFO	Typographical issues	ACKNOWLEDGED

1.3.2 errors.rs

ID	Severity	Summary	Status
06	INFO	Typographical issues	✓ RESOLVED

1.3.3 lib.rs

No issues found.

1.3.4 events.rs

No issues found.

1.3.5 types.rs

No issues found.



1.3.6 msg_codec.rs

No issues found.

1.3.7 compose_msg_codec.rs

No issues found.

1.3.8 state/mod.rs

No issues found.

1.3.9 oft.rs

No issues found.

1.3.10 token.rs

No issues found.

1.3.11 peer_config.rs

No issues found.



1.3.12 rate_limiter.rs

No issues found.

1.3.13 instructions/mod.rs

No issues found.

1.3.14 send.rs

No issues found.

1.3.15 lz_receive_type_info.rs

No issues found.

1.3.16 lz_receive_types_v2.rs

No issues found.

1.3.17 quote_send.rs

No issues found.



1.3.18 quote_ofc.rs

ID	Severity	Summary	Status
07	LOW	quote_ofc does not enforce requested amount within rate limits	ACKNOWLEDGED

1.3.19 instructions/privileged/mod.rs

No issues found.

1.3.20 init.rs

ID	Severity	Summary	Status
08	INFO	Init allows default_admin to be zero address	ACKNOWLEDGED
09	INFO	Spelling error of PBF_LOADER_UPGRADEABLE_PROGRAM_ID	ACKNOWLEDGED

1.3.21 set_admin.rs

ID	Severity	Summary	Status
10	INFO	Missing two-step ownership transfer	ACKNOWLEDGED

1.3.22 set_alt.rs

No issues found.

1.3.23 set_peer_config.rs

No issues found.

1.3.24 register_token.rs

ID	Severity	Summary	Status
11	INFO	Incorrect comment	ACKNOWLEDGED

1.3.25 sync_mint_authority.rs

No issues found.

1.3.26 configure_rate_limiter.rs

ID	Severity	Summary	Status
12	INFO	The rate limiter account bump only needs to be set when the account is not initialized	ACKNOWLEDGED

1.3.27 close_rate_limiter.rs

No issues found.

1.3.28 set_global_paused.rs

No issues found.

1.3.29 set_id_paused.rs



No issues found.

2 Findings

2.1 lz_receive.rs

lz_receive handles incoming cross-chain token transfers by validating LayerZero messages, checking rate limits, and minting tokens to recipients on Solana. It also processes optional compose messages for complex operations

2.1.1 Issues & Recommendations

Issue #01	Missing pause validation in <code>lz_receive()</code>
Severity	 LOW SEVERITY
Description	<p><code>send()</code> instruction checks if either the OFT or the used token is paused.</p> <p>This validation, however, is not present in <code>lz_receive()</code>, which allows the OFT to mint tokens on this chain even when it is paused. That means that a compromised remote chain can keep inflating supply on this chain via <code>lz_receive</code>.</p> <p>In a cross-chain exploit situation, this is exactly what pause flags are supposed to mitigate.</p>
Recommendation	Consider adding <code>inspect_paused(&ctx.accounts.oft_store, &ctx.accounts.token_store)?;</code> to <code>lz_receive()</code> .
Resolution	 ACKNOWLEDGED
	The lack of pause validation is intentional and aligns with our EVM implementation.

Issue #02	Incorrect account order when calling Token-2022 mint_to in credit
Severity	<div><div></div> INFORMATIONAL</div>
Description	<p>The credit passes the CPI accounts to <code>spl_token_2022::instruction::mint_to</code> in the incorrect order, causing the CPI to fail for every call. As a result, <code>lz_receive</code> cannot mint tokens and all inbound transfers fail.</p> <p>The instruction is built as <code>mint_to(program, token_mint, token_dest, authority, oft_store, amount)</code>, but the <code>invoke_signed</code> call passes accounts in the order:</p> <pre>0: token_dest 1: token_mint 2: mint_authority 3: oft_store</pre> <p>So the accounts expected by the CPI (<code>[mint, dest, authority, signer]</code>) are paired with:</p> <ul style="list-style-type: none">- mint - token_dest- dest - token_mint- authority - mint_authority- signer - oft_store <p>That means that the CPI will always fail with an account mismatch error when trying to mint.</p>
Recommendation	Fix the account order in credit: mint to mint, and dest to dest.
Resolution	<div><div></div> ACKNOWLEDGED</div>

Issue #03**Unchecked slicing of remaining_accounts****Severity** INFORMATIONAL**Description**

Malformed calls to `lz_receive` can cause the program to panic with an index-out-of-bounds error before the LayerZero clear endpoint is even invoked.

This does not break state integrity, but it is noisy and makes failures less diagnosable.


If `ctx.remaining_accounts.len() < Clear::MIN_ACCOUNTS_LEN`, the slice `0..Clear::MIN_ACCOUNTS_LEN` will panic at runtime with "index out of bounds" before returning an `OFTErrors`.

The same pattern exists implicitly for the compose accounts (they slice from `Clear::MIN_ACCOUNTS_LEN..`), but the first slice is enough to crash.

Recommendation

Consider adding an explicit length check before slicing.

```
require!(  
    ctx.remaining_accounts.len() >= Clear::MIN_ACCOUNTS_LEN,  
    OFTErrors::InvalidExtraAccountMetaList  
);
```

Resolution ACKNOWLEDGED

Issue #04**Receivers could potentially attempt to extract rent****Severity** INFORMATIONAL**Description**

If needed on `lz_receive()`, the `token_dest` account is initialized and funded with rent. After the transaction is complete the owner of the `token_dest` account can close the account to get the rent.

It is expected behaviour when on Solana but if the initialisation expense is not included in the fee on the source chain, the receiver can try to extract value by constantly closing the account and receiving the rent amount after each transfer.

Roughly this amounts to 0.00203 SOL or around 0.23 cents as of writing.

Recommendation

Consider adding this potential expense when the destination chain is Solana (if not done already). Other options (which are not ideal) could be making receivers deposit some amount for rent beforehand or requiring the receiver to have his token account already initialised.

Resolution ACKNOWLEDGED

The rent cost for ATA creation is already included in the fee quote during `send()`, so executors are fully compensated upfront.



Issue #05	Typographical issues
Severity	<div data-bbox="456 203 485 230"></div> INFORMATIONAL
Description	<p>A more explicit and readable constraint could be added in <code>LzReceive::apply()</code> where <code>pub mint_authority: AccountInfo<'info></code>, could be replaced with:</p> <pre data-bbox="448 456 1145 667">#[account(address = token_store.mint_authority @ OFTError::InvalidMintAuthority)] pub mint_authority: AccountInfo<'info>,</pre>
Recommendation	Consider making the fixes mentioned above.
Resolution	<div data-bbox="456 775 485 801"></div> ACKNOWLEDGED



2.2 errors.rs

errors defines custom error types for the OFT program.

2.2.1 Issues & Recommendations

Issue #06	Typographical issues
Severity	 INFORMATIONAL
Description	<p>The following errors are defined but not used and should be removed:</p> <ul style="list-style-type: none">- TransferBlocked- InvalidExtraAccountMetaList- TransferNotAllowed
Recommendation	Consider making the fixes mentioned above.
Resolution	 RESOLVED

2.3 lib.rs

lib declares all program instructions and defines the program entry point.

2.3.1 Privileged Functions

- `init[UpgradeAuthority]`
- `set_admin[DefaultAdmin]`
- `set_alt[DefaultAdmin]`
- `set_peer_config[DefaultAdmin]`
- `register_token[TokenRegistryAdmin]`
- `deregister_token[TokenRegistryAdmin]`
- `sync_mint_authority[TokenRegistryAdmin]`
- `configure_rate_limiter[RateLimiterAdmin]`
- `close_rate_limiter[DefaultAdmin]`
- `set_global_paused[Pauser/Unpauser]`
- `set_id_paused[Pauser/Unpauser]`

2.3.2 Issues & Recommendations

No issues found.

2.4 **events.rs**

events declares all events emitted by the program for indexing transfers and configuration changes.

2.4.1 **Issues & Recommendations**

No issues found.

2.5 `types.rs`

`types` provides wrapper types enabling Anchor to work with SPL Token 2022 accounts.

2.5.1 Issues & Recommendations

No issues found.



2.6 msg_codec.rs

msg_codec.rs encodes and decodes OFT message format containing token ID, recipient, amount, and optional compose data.

2.6.1 Issues & Recommendations

No issues found.



2.7 `compose_msg_codec.rs`

`compose_msg_codec` encodes and decodes compose messages for executing complex operations after token transfers.

2.7.1 Issues & Recommendations

No issues found.



2.8 state/mod.rs

mod re-exports all state account structures.

2.8.1 Issues & Recommendations

No issues found.



2.9 oft.rs

oft defines the global OFT configuration including admin roles, pause state, and ALT address.

2.9.1 Issues & Recommendations

No issues found.



2.10 token.rs

token stores per-token configuration including mint address, decimals conversion, and pause state.

2.10.1 Issues & Recommendations

No issues found.

2.11 `peer_config.rs`

`peer_config` maintains peer addresses and enforced options for each remote chain.

2.11.1 Issues & Recommendations

No issues found.



2.12 rate_limiter.rs

rate_limiter implements token bucket rate limiting for inbound and outbound transfers.

2.12.1 Issues & Recommendations

No issues found.



2.13 instructions/mod.rs

`mod.rs` re-exports all instruction modules and shared helper functions.

2.13.1 Issues & Recommendations

No issues found.



2.14 send.rs

send handles outgoing transfers by burning tokens and sending LayerZero messages.

2.14.1 Issues & Recommendations

No issues found.



2.15 lz_receive_type_info.rs

lz_receive_type_info returns version and minimal accounts for executor transaction construction.

2.15.1 Issues & Recommendations

No issues found.



2.16 lz_receive_types_v2.rs

lz_receive_types_v2 generates complete execution plans with all required accounts and ALTs.

2.16.1 Issues & Recommendations

No issues found.



2.17 `quote_send.rs`

`quote_send` provides gas estimates and validates parameters for send operations.

2.17.1 Issues & Recommendations



No issues found.



2.18 quote_ofc.rs

quote_ofc returns transfer limits, fees, and receipt information for potential operations.

2.18.1 Issues & Recommendations

Issue #07	quote_offt does not enforce requested amount within rate limits
Severity	 LOW SEVERITY
Description	<p>quote_offt returns a receipt for the caller's requested amount even if that amount exceeds the current rate limit. It also returns a max_amount_ld derived from the rate limiter, but does not enforce it.</p> <p>An integrator who misreads the API and only checks the receipt could try to send more than allowed, leading to failed send transactions.</p> <p>Suppose the current outbound sendable limit is 100 tokens. A client calls quote_offt with amount_ld = 1000, min_amount_ld = 900. The debit_view returns amount_sent_ld = ~amount_ld and amount_received_ld ≥ min_amount_ld.</p> <p>quote_offt returns:</p> <pre>oft_receipt.amount_received_ld = ~1000</pre> <pre>oft_limits.max_amount_ld = 100 (from rate limiter)</pre> <p>If the integrator is careless and looks only at oft_receipt, they may allow the user to proceed with amount_ld = 1000.</p> <p>However, it is still worth mentioning that actual send calls rate_limiter.outflow(amount) will revert with OFTError::RateLimitExceeded for amount > sendable, so it is not possible to bypass rate limiting via quote_offt.</p>
Recommendation	<p>If the intent is "quote this specific amount and fail if it violates limits" add the same check as in quote_send:</p> <pre>require!(amount_received_ld <= sendable, OFTError::RateLimitExceeded);</pre>
Resolution	 ACKNOWLEDGED

2.19 instructions/privileged/mod.rs

mod re-exports all privileged instruction modules.

2.19.1 Issues & Recommendations

No issues found.



2.20 init.rs



`init.rs` initializes the OFT program with admin roles and registers with LayerZero endpoint.



2.20.1 Privileged Functions

- `init[UpgradeAuthority]`



2.20.2 Issues & Recommendations

Issue #08	init allows default_admin to be zero address
Severity	 INFORMATIONAL
Description	<p>During init, it is possible to set default_admin = Pubkey::default(). After that, no one can ever call set_admin again, because set_admin requires the caller to be the default_admin.</p> <p>This can permanently lock admin role rotation and OApp delegate changes, and cannot be recovered without a program upgrade.</p>
Recommendation	We recommend adding a non-zero check in Init.apply. <code>require!(params.default_admin != Pubkey::default(), OFTErrors::InvalidAdmin);</code>
Resolution	 ACKNOWLEDGED

Issue #09	Spelling error of PBF_LOADER_UPGRADEABLE_PROGRAM_ID
Severity	 INFORMATIONAL
Description	PBF_LOADER_UPGRADEABLE_PROGRAM_ID is misspelled. It should be BPF_LOADER_UPGRADEABLE_PROGRAM_ID to indicate the Berkeley Packet Filter.
Recommendation	Update the variable to be BPF_LOADER_UPGRADEABLE_PROGRAM_ID.
Resolution	 ACKNOWLEDGED



2.21 set_admin.rs

set_admin updates admin role assignments for program governance.

2.21.1 Privileged Functions

- `set_admin[DefaultAdmin]`

2.21.2 Issues & Recommendations

Issue #10	Missing two-step ownership transfer
Severity	 INFORMATIONAL
Description	<p>The DefaultAdmin in the protocol has the highest privilege, allowing them to configure the protocol and override other lower-privilege admins.</p> <p>The current method of updating the admin is simply setting the address directly in <code>set_admin.rs:25</code>.</p> <p>This is problematic because if the admin accidentally inputs an address that no one owns, the protocol will be forced to operate without an admin.</p>
Recommendation	Consider implementing a two-step ownership transfer: the admin transfers ownership to a pending admin, who must explicitly accept it in another transaction.
Resolution	 ACKNOWLEDGED

2.22 set_alt.rs

set_alt configures Address Lookup Tables for transaction compression.

2.22.1 Privileged Functions

- set_alt[DefaultAdmin]

2.22.2 Issues & Recommendations

No issues found.



2.23 set_peer_config.rs

set_peer_config manages peer addresses and enforced options for remote chains.

2.23.1 Privileged Functions

- set_peer_config[DefaultAdmin]

2.23.2 Issues & Recommendations

No issues found.



2.24 register_token.rs

register_token registers new tokens for cross-chain bridging with validation.

2.24.1 Privileged Functions

- register_token[TokenRegistryAdmin]

2.24.2 Issues & Recommendations

Issue #11	Incorrect comment
Severity	INFORMATIONAL
Description	In register_token.rs:35, there is a comment that states "Initialize the oft_store". However, this is incorrect because the RegisterToken instruction does not initialize the oft_store account.
Recommendation	Consider removing or updating the incorrect comment.
Resolution	ACKNOWLEDGED

2.25 sync_mint_authority.rs

sync_mint_authority updates stored mint authority to match on-chain mint accounts.

2.25.1 Privileged Functions

- sync_mint_authority[TokenRegistryAdmin]

2.25.2 Issues & Recommendations

No issues found.





2.26 `configure_rate_limiter.rs`

`configure_rate_limiter` sets rate limit parameters for token and chain combinations.

2.26.1 Privileged Functions

- `configure_rate_limiter[RateLimiterAdmin]`

2.26.2 Issues & Recommendations

Issue #12	The rate limiter account bump only needs to be set when the account is not initialized
Severity	 INFORMATIONAL
Description	<p>At <code>configure_rate_limiter.rs:45</code>, the rate limiter account's bump is set to <code>ctx.bumps.rate_limiter</code> every time <code>ConfigureRateLimiter</code> is being called, even after the account is already initialized.</p> <p>This approach is unnecessary because the bumps only need to be set once during the rate limiter account creation.</p>
Recommendation	<p>Modify the implementation so that <code>rate_limiter.bump = ctx.bumps.rate_limiter</code> is only called inside the if statement in line 47. This works because the newly created rate limiter account will have the initialized field set to false by default.</p>
Resolution	 ACKNOWLEDGED

2.27 close_rate_limiter.rs

close_rate_limiter closes rate limiter accounts and returns rent to admin.

2.27.1 Privileged Functions

- close_rate_limiter[DefaultAdmin]

2.27.2 Issues & Recommendations

No issues found.



2.28 `set_global_paused.rs`

`set_global_paused` pauses or unpauses all bridge operations globally.

2.28.1 Privileged Functions

- `set_global_paused[Pauser/Unpauser]`

2.28.2 Issues & Recommendations

No issues found.



2.29 `set_id_paused.rs`

`set_id_paused` pauses or unpauses bridging for specific tokens.

2.29.1 Privileged Functions

- `set_id_paused[Pauser/Unpauser]`

2.29.2 Issues & Recommendations

No issues found.





PALADIN
BLOCKCHAIN SECURITY