



CAIRO SECURITY
CLAN

ENDUR - LIQUID STAKING TOKEN (LST)

SECURITY ASSESMENT REPORT

NOVEMBER 2024

Prepared for
ENDUR



Contents

1	About Cairo Security Clan	2
2	Disclaimer	2
3	Executive Summary	3
4	Summary of Audit	4
4.1	Scoped Files	4
4.2	Issues	4
5	Risk Classification	5
6	Issues by Severity Levels	6
6.1	High	6
6.1.1	exit_delegation_pool_action() Can Be Called Directly, Bypassing unstake_action() Function	6
6.1.2	Attacker Could Perform DOS Attack on Withdrawal Queue by Donating 1 Wei	7
6.2	Medium	8
6.2.1	Function _unclaimed_rewards() Does Not Subtract Commission of Delegation Pool	8
6.3	Low	9
6.3.1	_unclaimed_rewards() Might Use Outdated global_index for Calculation	9
6.3.2	_process_claim() Could Revert Due to Breaking the Gas Limit	10
6.3.3	Function adjust_mint() Should Be Called Before _convert_to_assets()	11
6.4	Informational	12
6.4.1	info.delegator_index is Not Used and Has No Impact in the Function update_delegator_info()	12
6.5	Best Practices	13
6.5.1	Unnecessary Storage Usage	13
7	Test Evaluation	14
7.1	Compilation Output	14
7.2	Tests Output	15



1 About Cairo Security Clan

Cairo Security Clan is a leading force in the realm of blockchain security, dedicated to fortifying the foundations of the digital age. As pioneers in the field, we specialize in conducting meticulous smart contract security audits, ensuring the integrity and reliability of decentralized applications built on blockchain technology.

At Cairo Security Clan, we boast a multidisciplinary team of seasoned professionals proficient in blockchain security, cryptography, and software engineering. With a firm commitment to excellence, our experts delve into every aspect of the Web3 ecosystem, from foundational layer protocols to application-layer development. Our comprehensive suite of services encompasses smart contract audits, formal verification, and real-time monitoring, offering unparalleled protection against potential vulnerabilities.

Our team comprises industry veterans and scholars with extensive academic backgrounds and practical experience. Armed with advanced methodologies and cutting-edge tools, we scrutinize and analyze complex smart contracts with precision and rigor. Our track record speaks volumes, with a plethora of published research papers and citations, demonstrating our unwavering dedication to advancing the field of blockchain security.

At Cairo Security Clan, we prioritize collaboration and transparency, fostering meaningful partnerships with our clients. We believe in a customer-oriented approach, engaging stakeholders at every stage of the auditing process. By maintaining open lines of communication and soliciting client feedback, we ensure that our solutions are tailored to meet the unique needs and objectives of each project.

Beyond our core services, Cairo Security Clan is committed to driving innovation and shaping the future of blockchain technology. As active contributors to the ecosystem, we participate in the development of emerging technologies such as Starknet, leveraging our expertise to build robust infrastructure and tools. Through strategic guidance and support, we empower our partners to navigate the complexities of the blockchain landscape with confidence and clarity.

In summary, Cairo Security Clan stands at the forefront of blockchain security, blending technical prowess with a client-centric ethos to deliver unparalleled protection and peace of mind in an ever-evolving digital landscape. Join us in safeguarding the future of decentralized finance and digital assets with confidence and conviction.

2 Disclaimer

Disclaimer Limitations of this Audit:

This report is based solely on the materials and documentation provided by you to Cairo Security Clan for the specific purpose of conducting the security review outlined in the [Summary of Audit](#) and [Scoped Files](#). The findings presented here may not be exhaustive and may not identify all potential vulnerabilities. Cairo Security Clan provides this review and report on an "as-is" and "as-available" basis. You acknowledge that your use of this report, including any associated services, products, protocols, platforms, content, and materials, occurs entirely at your own risk.

Inherent Risks of Blockchain Technology:

Blockchain technology remains in its developmental stage and is inherently susceptible to unknown risks and vulnerabilities. This review is specifically focused on the smart contract code and does not extend to the compiler layer, programming language elements beyond the reviewed code, or other potential security risks outside the code itself.

Report Purpose and Reliance:

This report should not be construed as an endorsement of any specific project or team, nor does it guarantee the absolute security of the audited smart contracts. No third party should rely on this report for any purpose, including making investment or purchasing decisions.

Liability Disclaimer:

To the fullest extent permitted by law, Cairo Security Clan disclaims all liability associated with this report, its contents, and any related services and products arising from your use. This includes, but is not limited to, implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

Third-Party Products and Services:

Cairo Security Clan does not warrant, endorse, guarantee, or assume responsibility for any products or services advertised by third parties within this report, nor for any open-source or third-party software, code, libraries, materials, or information linked to, referenced by, or accessible through this report, its content, and related services and products. This includes any hyperlinked websites, websites or applications appearing on advertisements, and Cairo Security Clan will not be responsible for monitoring any transactions between you and third-party providers. It is recommended that you exercise due diligence and caution when considering any third-party products or services, just as you would with any purchase or service through any medium.

Disclaimer of Advice:

FOR THE AVOIDANCE OF DOUBT, THIS REPORT, ITS CONTENT, ACCESS, AND/OR USE, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHOULD NOT BE CONSIDERED OR RELIED UPON AS FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER PROFESSIONAL ADVICE.



3 Executive Summary

This document presents the security review performed by **Cairo Security Clan** on the Liquid Staking Token (LST) developed by **Endur**. Learn more from [docs](#).

Endur.Fi is a platform that introduces xSTRK, a tradeable Liquid Staking Token (LST) on Starknet. Developed through a collaboration between **STRKFarm** and **Karnot**, Endur allows STRK token holders to stake their assets while maintaining liquidity.

The audit was performed using

- manual analysis of the codebase,
- automated analysis tools,
- simulation of the smart contract,
- analysis of edge test cases

8 points of attention, where 0 is classified as Critical, 2 are classified as High, 1 is classified as Medium, 3 are classified as Low, 1 is classified as Informational and 1 is classified as Best Practices. The issues are summarized in Fig. 1.

This document is organized as follows. Section 1 About Cairo Security Clan. Section 2 Disclaimer. Section 3 Executive Summary. Section 4 Summary of Audit. Section 5 Risk Classification. Section 6 Issues by Severity Levels. Section 7 Test Evaluation.

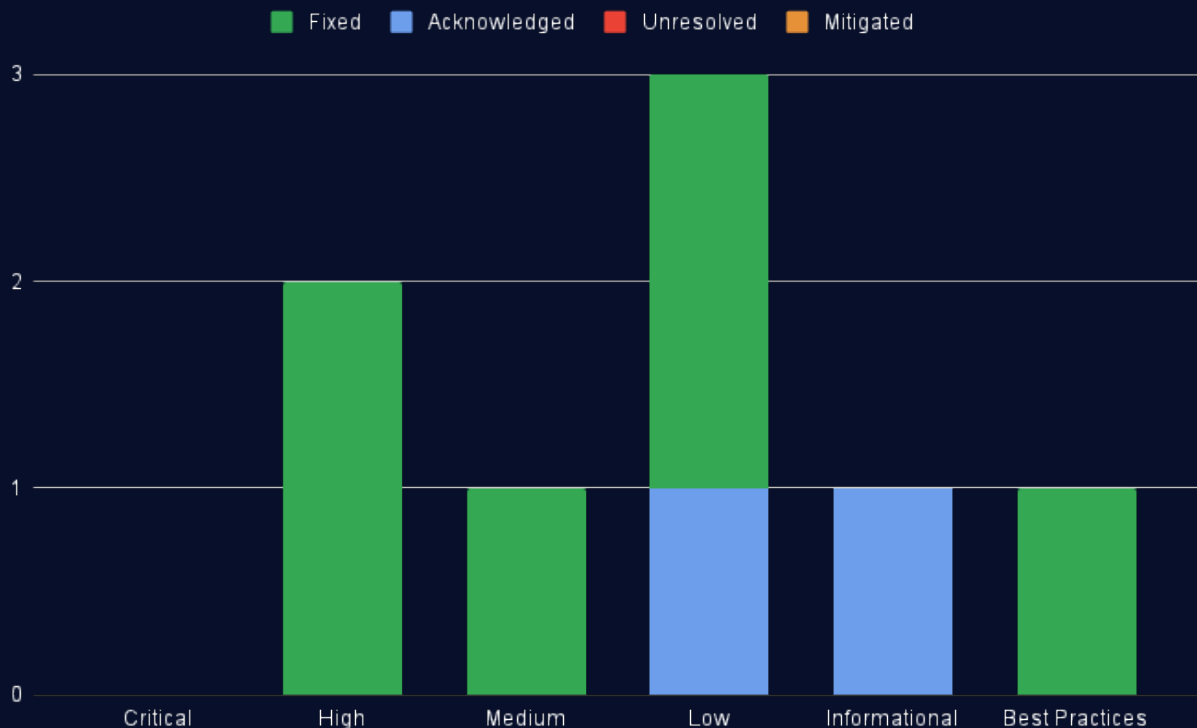


Fig 1: Distribution of issues: Critical (0), High (2), Medium (1), Low (3), Informational (1), Best Practices (1). Distribution of status: Fixed (6), Acknowledged (2), Mitigated (0), Unresolved (0).



4 Summary of Audit

Audit Type	Security Review
Scarb Version	2.8.4
Final Report	22/11/2024
Repository	Endur-fi/lst
Initial Commit Hash	471693ff16d4db3272e721dc976e5d933e192782
Final Commit Hash	d607247e307c88e01deb1bae084bd2e62c67d47f
Documentation	Website documentation
Test Suite Assessment	High

4.1 Scoped Files

	Contracts
1	src/delegator/delegator.cairo
2	src/delegator/interface.cairo
3	src/interfaces/delegation_pool.cairo
4	src/interfaces/staking.cairo
5	src/lib.cairo
6	src/lst/erc4626.cairo
7	src/lst/interface.cairo
8	src/lst/lst.cairo
9	src/utills/common.cairo
10	src/utills/constants.cairo
11	src/utills/erc20helper.cairo
12	src/utills/errors.cairo
13	src/utills/math.cairo
14	src/withdrawal_queue/interface.cairo
15	src/withdrawal_queue/withdrawal_queue.cairo

4.2 Issues

	Findings	Severity	Update
1	exit_delegation_pool_action() can be called directly, bypassing unstake_action() function	High	Fixed
2	Attacker could perform DOS attack on withdrawal queue from taking funds by donating 1 wei	High	Fixed
3	Function _unclaimed_rewards() does not subtract commission of delegation pool	Medium	Fixed
4	_unclaimed_rewards() might use outdated global_index for calculation	Low	Acknowledged
5	_process_claim() could revert due to breaking the gas limit	Low	Fixed
6	Function adjust_mint() should be called before _convert_to_assets()	Low	Fixed
7	info.delegator_index is not used and has no impact in the function update_delegator_info()	Informational	Acknowledged
8	Unnecessary storage usage	Best Practices	Fixed



5 Risk Classification

The risk rating methodology used by **Cairo Security Clan** follows the principles established by the **CVSS risk rating methodology**. The severity of each finding is determined by two factors: **Likelihood** and **Impact**.

Likelihood measures how likely an attacker will uncover and exploit the finding. This factor will be one of the following values:

- a) **High**: The issue is trivial to exploit and has no specific conditions that need to be met;
- b) **Medium**: The issue is moderately complex and may have some conditions that need to be met;
- c) **Low**: The issue is very complex and requires very specific conditions to be met.

When defining the likelihood of a finding, other factors are also considered. These can include but are not limited to Motive, opportunity, exploit accessibility, ease of discovery, and ease of exploit.

Impact is a measure of the damage that may be caused if an attacker exploits the finding. This factor will be one of the following values:

- a) **High**: The issue can cause significant damage such as loss of funds or the protocol entering an unrecoverable state;
- b) **Medium**: The issue can cause moderate damage such as impacts that only affect a small group of users or only a particular part of the protocol;
- c) **Low**: The issue can cause little to no damage such as bugs that are easily recoverable or cause unexpected interactions that cause minor inconveniences.

When defining the impact of a finding other factors are also considered. These can include but are not limited to Data/state integrity, loss of availability, financial loss, and reputation damage. After defining the likelihood and impact of an issue, the severity can be determined according to the table below.

		Likelihood		
		High	Medium	Low
Impact	High	Critical	High	Medium
	Medium	High	Medium	Low
	Low	Medium	Low	Info/Best Practices

To address issues that do not fit a High/Medium/Low severity, **Cairo Security Clan** also uses three more finding severities: **Informational**, **Best Practices** and **Gas**

- a) **Informational** findings do not pose any risk to the application, but they carry some information that the audit team intends to formally pass to the client;
- b) **Best Practice** findings are used when some piece of code does not conform with smart contract development best practices;
- b) **Gas** findings are used when some piece of code uses more gas than it should be or have some functions that can be removed to save gas.



6 Issues by Severity Levels

6.1 High

6.1.1 `exit_delegation_pool_action()` Can Be Called Directly, Bypassing `unstake_action()` Function

File(s): `src/delegator/delegator.cairo`

Description: The unstaking process in the delegator contract consists of two steps. First, the owner calls the function `start_unstake_intent()`, which triggers the `delegation_pool.exit_delegation_pool_intent()` function to initiate an unstaking delay. After the delay period, any user can invoke the `unstake_action()` function (permissionless) to finalize the unstaking process by calling `delegation_pool.exit_delegation_pool_action()`.

```

1 fn unstake_action(ref self: ContractState) {
2     self._general_validations();
3     assert(self.is_unstake_in_progress.read(), Errors::UNSTAKE_NOT_IN_PROGRESS);
4
5     // ensures rewards are claimed and re-staked
6     // else they will come here in to this contract
7     // on exit from delegation pool
8     self.lst.read().claim_rewards();
9
10    let amount = self
11        .delegation_pool
12        .read()
13        .exit_delegation_pool_action(get_contract_address()); // @audit this function can be called directly on
        pool
14
15    // approve tokens
16    erc20helper::approve(
17        self.token.read(), self.withdraw_queue.read().contract_address, amount.into()
18    );
19
20    self.withdraw_queue.read().take_funds(amount.into());
21    self.is_unstake_in_progress.write(false);
22
23    self.emit(UnstakeAction { amount: amount });
24 }

```

However, the function `exit_delegation_pool_action()` can be called by anyone for any address. An attacker could potentially call this function directly, bypassing the `unstake_action()` function, which would finalize the unstaking process and transfer the tokens to the delegator contract.

This attack vector would prevent the `take_funds()` function from being executed, meaning the funds would be transferred to the delegator contract but not properly pulled to the withdrawal queue. As a result, users in the withdrawal queue would be unable to withdraw their funds. While the funds wouldn't be locked (since the `_stake()` function uses the entire contract balance to stake), the internal accounting would be corrupted.

Recommendation(s): Consider implementing additional logic to properly handle tokens that are unstaked via direct calls to `exit_delegation_pool_action()`.

Status: Fixed

Update from the client: Fixed in `c82774fb4`



6.1.2 Attacker Could Perform DOS Attack on Withdrawal Queue by Donating 1 Wei

File(s): [src/withdrawal_queue/withdrawal_queue.cairo](#)

Description: The withdrawal queue mechanism in the system is designed to process unstaking in two steps:

1. The LST contract calls the function `reduce_total_queue_amount()` to notify the withdrawal queue of an unstaking event, causing the queue to increase the value of `intransit_amount`.
2. The `take_funds()` function is called to actually transfer the tokens to the withdrawal queue, thereby reducing `intransit_amount`.

```
1 fn take_funds(ref self: ContractState, amount: u256) {  
2     let caller = get_caller_address();  
3     // should be ok permissionless i guess  
4     let unprocessed = self.unprocessed_withdraw_queue_amount.read();  
5     let total_pending_funds = self.intransit_amount.read();  
6     assert(total_pending_funds >= amount, Errors::EXCESS_FUNDS); // @audit Attacker can donate 1 wei and force  
7     // the original unstaking flow to fail  
8     // ...  
9 }
```

The core issue is that the `take_funds()` function is permissionless, meaning any address can call it and donate an arbitrary amount of funds (e.g., 1 wei). By donating even a minimal amount, an attacker can artificially reduce the value of `intransit_amount`, triggering a situation where the condition `total_pending_funds >= amount` fails when the original unstaking operation attempts to process the withdrawal.

Recommendation(s): Consider adding access control to the function `take_funds()` to prevent unauthorized addresses from calling it and modifying state variables.

Status: Fixed

Update from the client: Fixed in [ee20d1a2](#)



6.2 Medium

6.2.1 Function `_unclaimed_rewards()` Does Not Subtract Commission of Delegation Pool

File(s): `src/lst/lst.cairo`

Description: In the delegation pool, a commission is deducted from the earned rewards, as seen in the function `update_rewards`. This function calculates the rewards, subtracts the commission, and updates the unclaimed rewards for the pool member:

```
1      self: @ContractState, ref pool_member_info: InternalPoolMemberInfo, updated_index: Index
2  ) {
3      let interest: Index = updated_index - pool_member_info.index;
4      pool_member_info.index = updated_index;
5      let rewards_including_commission = compute_rewards_rounded_down(
6          amount: pool_member_info.amount, :interest
7      );
8      let commission_amount = compute_commission_amount_rounded_up(
9          :rewards_including_commission, commission: pool_member_info.commission
10     );
11     let rewards = rewards_including_commission - commission_amount;
12     pool_member_info.unclaimed_rewards += rewards;
13     pool_member_info.commission = self.commission.read();
14 }
```

However, the `_unclaimed_rewards()` function in the `lst` contract does not account for this commission when calculating the unclaimed rewards. As a result, the function returns a value that is higher than it should be, which affects the output of the `total_assets()` function and the share calculation for the ERC4626 vault.

Recommendation(s): Consider adjusting the `_unclaimed_rewards()` function to subtract the commission from the calculated unclaimed rewards.

Status: Fixed

Update from the client: Fixed in `60486a5`



6.3 Low

6.3.1 `_unclaimed_rewards()` Might Use Outdated `global_index` for Calculation

File(s): `src/lst/lst.cairo`

Description: In the staking contract, the reward calculation for each user is tracked by an index. Subtracting the current `global_index` from the last recorded index gives the amount of rewards earned per staked token. The function `_unclaimed_rewards()` currently reads the `global_index` directly from the staking contract's storage:

```
1 fn _unclaimed_rewards(self: @ContractState) -> u256 {
2     let config = self.config.read();
3     let staking_config = config.staker.contract_parameters(); // @audit global_index might be outdated without
4     update_global_index_if_needed()
5     let global_index = staking_config.global_index;
6     let last_index = self.staker_index.read();
7
8     // Note: contracts is sn staking contracts dependency
9     let rewards: u256 = contracts::utils::compute_rewards_rounded_up
10     self.staked_amount.read().try_into().unwrap(), global_index - last_index
11 )
12     .into();
13
14     // adjust for reward fee
15     let fee = math::get_fee_by_deduction(rewards, config.rewards_fee_bps);
16
17     rewards - fee
18 }
```

However, this value could be outdated if the `update_global_index_if_needed()` function is not called beforehand. As a result, the calculation of unclaimed rewards may be inaccurate, which can impact other accounting calculations and potentially lead to incorrect reward distributions.

Recommendation(s): Consider re-implementing the logic of `update_global_index_if_needed()` to ensure the `global_index` is always up-to-date before it is used for reward calculation.

Status: Acknowledged

Update from the client: As per Starknet, they shall update the index every 30 minutes. Similarly, we can also run a backend system to trigger an index update every 30 minutes.



6.3.2 _process_claim() Could Revert Due to Breaking the Gas Limit

File(s): `src/withdrawal_queue/withdrawal_queue.cairo`

Description: The function `_process_claim()` handles user withdrawal claims. Users can claim a withdrawal request if it is the oldest unclaimed request, or if the current timestamp has passed the claim time.

```
15     let request = self.requests.read(request_id);
16     assert(!request.isClaimed, Errors::ALREADY_CLAIMED);
17
18     let last_fill_id = self.last_fill_id.read();
19     let condition1 = (last_fill_id + 1) == request_id;
20     let condition2 = request.claimTime <= get_block_timestamp();
21     assert(condition1 || condition2, Errors::TOO_EARLY_TO_CLAIM);
22
23     // mark request as claimed
24     let request = WithdrawRequest { isClaimed: true, ..request };
25     self.requests.write(request_id, request);
26
27     if (condition1) {
28         // loop for next unclaimed request
29         let mut next_id = request_id + 1;
30         loop {
31             let request = self.requests.read(next_id);
32             if (!request.isClaimed) {
33                 self.last_fill_id.write(next_id - 1); // @audit claim ahead of `last_fill_id` could make this
34                 loop run out of gas
35                 break;
36             }
37             next_id += 1;
38         }
39         // ...
40     }
```

When the request is claimed under the first condition, the function enters a loop to find the next unclaimed request. However, this loop is unbounded, meaning there is no way to stop it if the next unclaimed request cannot be found after too many iterations. As a result, the function could run out of gas, causing it to revert. This would prevent the withdrawal request from being processed and leave it stuck in the queue.

Although the function `claim_withdrawal()` is public, allowing anyone to help other users claim their withdrawals, callers must still pay the gas fees. Malicious actors could exploit this by spamming the system with thousands of 1-wei withdrawal requests, forcing the admin or other users to spend a significant amount of gas to prevent `_process_claim()` from running out of gas.

Recommendation(s): Consider reviewing the current mechanism in the `_process_claim()` function. A suggestion is allowing users to specifically choose to claim a request under the second condition (i.e., when the claim time has passed) to ensure that withdrawal requests are not left stuck in the queue. Another suggestion is removing claiming under the first condition and forcing all withdrawal requests to go through the waiting period.

Status: Fixed

Update from the client: Fixed in `f86c8ad2`



6.3.3 Function `adjust_mint()` Should Be Called Before `_convert_to_assets()`

File(s): `src/lst/erc4626.cairo`

Description: In the ERC4626 standard, each action (like minting or redeeming) has a corresponding preview function. These preview functions estimate the amount of assets or shares a user will receive based on their input and consider fees through the use of "adjust" functions. The "adjust" functions are designed to modify the input value (shares or assets) to account for fees.

```
1  fn adjust_deposit(self: @ComponentState<TContractState>, assets: u256) -> u256 {
2      assets
3  }
4
5  fn adjust_mint(self: @ComponentState<TContractState>, shares: u256) -> u256 {
6      shares
7  }
8
9  fn adjust_withdraw(self: @ComponentState<TContractState>, assets: u256) -> u256 {
10     assets
11 }
12
13 fn adjust_redeem(self: @ComponentState<TContractState>, shares: u256) -> u256 {
14     shares
15 }
16 }
17
18 fn preview_mint(self: @ComponentState<TContractState>, shares: u256) -> u256 {
19     let raw_amount = self._convert_to_assets(shares, Rounding::Ceil);
20     Fee::adjust_mint(self, raw_amount) // @audit should be called before calling _convert_to_assets()
21 }
22
23 fn preview_redeem(self: @ComponentState<TContractState>, shares: u256) -> u256 {
24     let raw_amount = self._convert_to_assets(shares, Rounding::Floor);
25     Fee::adjust_redeem(self, raw_amount) // @audit should be called before _convert_to_assets()
26 }
```

Currently, in the `preview_mint()` and `preview_redeem()` functions, the adjustment functions are applied after converting shares to assets using `_convert_to_assets()`. However, the "adjust" functions are intended to be applied on the original input (shares for `adjust_mint()` and shares for `adjust_redeem()`) as shown in the naming of input parameters. Applying them after the conversion to assets could lead to unintended results, as the adjustments may not be applied to the intended value.

Recommendation(s): Consider calling function `adjust_mint()` before `_convert_to_assets()`. Or, if the variable naming in `adjust_mint()` or `adjust_redeem()` is incorrect and these functions were meant to take assets as input, the function signatures and their calls should be adjusted accordingly.

Status: Fixed

Update from the client: Fixed in [28d9d77](#)



6.4 Informational

6.4.1 info.delegator_index is Not Used and Has No Impact in the Function update_delegator_info()

File(s): [src/lst/lst.cairo](#)

Description: The function `update_delegator_info()` allows the caller to pass in a delegator address and a `DelegatorInfo` struct. The `DelegatorInfo` struct contains a `delegator_index` field, but this field is never used in the function and does not impact the execution of the function in any way.

```
41     ref self: ContractState, delegator: ContractAddress, info: DelegatorInfo
42 ) { // @audit info.delegator_index is not used and have no impact in this function
43     self.common.assert_only_owner();
44     assert(delegator.is_non_zero(), Errors::INVALID_DELEGATOR);
45     let delegator_index = self.delegator_info.read(delegator).delegator_index;
46     assert(delegator_index != 0, Errors::DELEGATOR_NOT_FOUND);
47
48     // assert no staked funds if deactivating
49     if (!info.is_active) {
50         let member_info = IDelegatorDispatcher { contract_address: delegator }
51             .get_pool_member_info();
52         assert(member_info.amount == 0, Errors::REMOVE_DELEGATOR_NOT_ALLOWED);
53     }
54
55     self._update_delegator_info(delegator, info);
56 }
```

In the function `_update_delegator_info()`, the `info.delegator_index` is not used and will be overwritten before being written to storage.

Recommendation(s): Consider allowing the caller to pass only the `is_active` variable instead of the entire `DelegatorInfo` struct to the function `update_delegator_info()`.

Status: Acknowledged

Update from the client: This design choice prioritizes simplicity. In the future, when additional properties are added to `DelegatorInfo`, they can all be set through a single input. Since this is an admin-only function, the additional index won't introduce confusion. Furthermore, sufficient checks are implemented in the code to ensure this variable does not disrupt any logic.



6.5 Best Practices

6.5.1 Unnecessary Storage Usage

File(s): `src/withdrawal_queue/withdrawal_queue.cairo`

Description: The Withdrawal Queue contract acts like the ERC721 interface; in storage, it holds the next request IDs at the `last_request_id` property. However, this property type is `u256`. Since it is increased in each request, reaching the `u128` limit is not feasible. The current implementation uses two storage slots because the `u256` struct has two `u128` properties. This causes unnecessary storage usage.

Recommendation(s): Consider using `u128` data type for `last_request_id`.

Status: Fixed

Update from the client: Fixed in commit `8a59fd`.



7 Test Evaluation

7.1 Compilation Output

```

1  scarb build
2      Updating git repository github.com/starkware-libs/starknet-staking
3      Updating git repository github.com/foundry-rs/starknet-foundry
4  Downloading openzeppelin v0.18.0
5  Downloading openzeppelin_security v0.18.0
6  Downloading openzeppelin_finance v0.18.0
7  Downloading openzeppelin_merkle_tree v0.18.0
8  Downloading openzeppelin_presets v0.18.0
9  Downloading openzeppelin_token v0.18.0
10 Downloading openzeppelin_governance v0.18.0
11 Downloading openzeppelin_account v0.18.0
12 Downloading openzeppelin_access v0.18.0
13 Downloading openzeppelin_upgrades v0.18.0
14 Downloading openzeppelin_utils v0.18.0
15 Downloading openzeppelin_introspection v0.18.0
16     Updating crates.io index
17     Updating git repository `github.com/xJonathanLEI/starknet-rs`
18     Updating git submodule `github.com/OpenZeppelin/cairo-contracts`
19 Downloading crates ...
20 Downloaded tinyvec_macros v0.1.1
21 Downloaded autocfg v1.3.0
22 Downloaded num-traits v0.2.19
23 Downloaded getrandom v0.2.15
24 Downloaded cairo-lang-proc-macros v2.8.2
25 Downloaded futures-macro v0.3.30
26 Downloaded cairo-lang-compiler v2.8.2
27 Downloaded cairo-lang-macro-attributes v0.1.0
28 Downloaded cached_proc_macro_types v0.1.1
29 Downloaded smallvec v1.13.2
30 Downloaded parking_lot_core v0.9.10
31 Downloaded rawpointer v0.2.1
32 Downloaded schemars_derive v0.8.21
33 Downloaded heck v0.4.1
34 Downloaded cairo-lang-sierra v2.8.2
35 Downloaded salsa20 v0.10.2
36 Downloaded mio v1.0.2
37 Downloaded clap_lex v0.7.2
38 Downloaded which v5.0.0
39 Downloaded strsim v0.10.0
40 Downloaded promptly v0.3.1
41 Downloaded crossbeam-deque v0.8.5
42 Downloaded memoffset v0.6.5
43 Downloaded hermit-abi v0.3.9
44 Downloaded native-tls v0.2.12
45 Downloaded tempfile v3.13.0
46 Downloaded windows-targets v0.48.5
47 Downloaded colored v2.1.0
48 Compiling ark-secp256r1 v0.4.0
49 Compiling ark-secp256k1 v0.4.0
50 Compiling linkme-impl v0.3.28
51 Compiling similar v2.6.0
52 Compiling normalize-line-endings v0.3.0
53 Compiling unicode-width v0.1.13
54 Compiling console v0.15.8
55 Compiling snapbox v0.4.17
56 Compiling linkme v0.3.28
57 Compiling cairo-lang-runner v2.8.2
58 Compiling starknet v0.11.0 (github.com/xJonathanLEI/starknet-rs?rev=660a732#660a7323)
59 Compiling cairo-lang-macro-attributes v0.1.0
60 Compiling cairo-lang-macro-stable v1.0.0
61 Compiling cairo-lang-macro v0.1.0
62 Compiling shared v0.1.0 (/home/runner/.cache/scarb/registry/git/checkouts/starknet-foundry-v6206fshfutuo/3817
    c903/crates/shared)

```



```
63   Compiling snforge-scarb-plugin v0.32.0 (/home/runner/.cache/scarb/registry/git/checkouts/starknet-foundry-
    v6206fshfutuo/3817c903/crates/snforge-scarb-plugin)
64   Finished `release` profile [optimized] target(s) in 10m 05s
65   Compiling lst v0.1.0 (/home/runner/work/031-KARNOT-LST/031-KARNOT-LST/contractsafteraudit/Scarb.toml)
66 warn: external contracts not found for selectors: `contracts_commons::erc20_mock::DualCaseERC20Mock`
67   Finished `dev` profile target(s) in 11 minutes
```

7.2 Tests Output

```
1 snforge test --max-n-steps 4000000000
2
3   Updating git repository github.com/starkware-libs/starknet-staking
4   Updating git repository github.com/foundry-rs/starknet-foundry
5 Downloading openzeppelin v0.18.0
6 Downloading openzeppelin_security v0.18.0
7 Downloading openzeppelin_finance v0.18.0
8 Downloading openzeppelin_merkle_tree v0.18.0
9 Downloading openzeppelin_presets v0.18.0
10 Downloading openzeppelin_token v0.18.0
11 Downloading openzeppelin_governance v0.18.0
12 Downloading openzeppelin_account v0.18.0
13 Downloading openzeppelin_access v0.18.0
14 Downloading openzeppelin_upgrades v0.18.0
15 Downloading openzeppelin_utils v0.18.0
16 Downloading openzeppelin_introspection v0.18.0
17   Updating crates.io index
18   Updating git repository `github.com/xJonathanLEI/starknet-rs`
19   Updating git submodule `github.com/OpenZeppelin/cairo-contracts`
20 Downloading crates ...
21   Downloaded percent-encoding v2.3.1
22   Downloaded autocfg v1.3.0
23   Downloaded quote v1.0.37
24   Downloaded ryu v1.0.18
25   Downloaded futures-sink v0.3.30
26   Downloaded slab v0.4.9
27   Downloaded windows-targets v0.52.6
28   Downloaded minimal-lexical v0.2.1
29   Downloaded derivative v2.2.0
30   Downloaded anyhow v1.0.89
31   Downloaded crossbeam-utils v0.8.20
32   Downloaded cairo-lang-project v2.8.2
33   Downloaded project-root v0.2.2
34   Downloaded futures-executor v0.3.30
35   Downloaded deadpool-runtime v0.1.4
36   Downloaded sct v0.7.1
37   Downloaded native-tls v0.2.12
38   Downloaded dirs-sys-next v0.1.2
39   Downloaded libredox v0.1.3
40   Downloaded ring v0.17.8
41   Downloaded rustls-pemfile v1.0.4
42   Downloaded num-complex v0.2.4
43   Downloaded rawpointer v0.2.1
44   Downloaded itertools v0.11.0
45   Downloaded good_lp v1.8.1
46   Downloaded string_cache v0.8.7
47   Downloaded num-traits v0.1.43
48   Downloaded const-hex v1.13.1
49   Downloaded xshell-macros v0.2.6
50   Downloaded petgraph v0.6.5
51   Downloaded xshell v0.2.6
52   Downloaded cairo-lang-syntax-codegen v2.8.2
53   Downloaded clap v4.5.18
54   Downloaded portable-atomic v1.7.0
55   Downloaded doc-comment v0.3.3
56   Downloaded instant v0.1.13
57   Downloaded nu-ansi-term v0.46.0
58   Downloaded crossbeam-epoch v0.9.18
59   Downloaded globset v0.4.15
60   Downloaded starknet_api v0.13.0-rc.1
```




Cairo Security Clan

```
61 Downloaded globwalk v0.9.1
62 Downloaded clap_derive v4.5.18
63 Downloaded indicatif v0.17.8
64 Downloaded overload v0.1.1
65 Downloaded windows-sys v0.59.0
66 Downloaded heck v0.5.0
67 Downloaded windows_aarch64_gnullvm v0.48.5
68 Downloaded windows_x86_64_gnullvm v0.48.5
69 Downloaded yansi v0.5.1
70 Downloaded pretty_assertions v1.4.0
71 Downloaded linux_raw_sys v0.4.14
72 Downloaded ignore v0.4.23
73 Downloaded relative-path v1.9.3
74 Downloaded id-arena v2.2.1
75 Downloaded genco v0.17.9
76 Downloaded windows_i686_gnu v0.48.5
77 Downloaded windows_i686_msvc v0.48.5
78 Downloaded windows_x86_64_gnu v0.48.5
79 Downloaded windows_x86_64_msvc v0.48.5
80 Downloaded cairo-lang-test-utils v2.8.2
81 Downloaded cairo-lang-semantic v2.8.2
82 Downloaded bstr v1.10.0
83 Downloaded number_prefix v0.4.0
84 Downloaded camino v1.1.9
85 Downloaded crossbeam-deque v0.8.5
86 Downloaded ndarray v0.13.1
87 Downloaded windows_aarch64_msvc v0.48.5
88 Downloaded scarb-metadata v1.12.0
89 Downloaded windows-targets v0.48.5
90 Downloaded winapi-i686-pc-windows-gnu v0.4.0
91 Downloaded colored v2.1.0
92 Downloaded diff v0.1.13
93 Downloaded toml v0.8.19
94 Downloaded assert_matches v1.5.0
95 Downloaded windows-sys v0.52.0
96 Downloaded winapi-x86_64-pc-windows-gnu v0.4.0
97 Downloaded cairo-lang-parser v2.8.2
98 Downloaded diffliib v0.4.0
99 Downloaded diffy v0.3.0
100 Downloaded cairo-lang-formatter v2.8.2
101 Downloaded cairo-lang-sierra-generator v2.8.2
102 Downloaded cairo-lang-casm v2.8.2
103 Downloaded cairo-lang-sierra-ap-change v2.8.2
104 Downloaded matrixmultiply v0.2.4
105 Downloaded fs_extra v1.3.0
106 Downloaded rtoolbox v0.0.2
107 Downloaded strsim v0.10.0
108 Downloaded num-rational v0.4.2
109 Downloaded strum v0.24.1
110 Downloaded strum_macros v0.24.3
111 Downloaded sprs v0.7.1
112 Downloaded bit-set v0.5.3
113 Downloaded ena v0.14.3
114 Downloaded strum_macros v0.25.3
115 Downloaded axum-core v0.4.5
116 Downloaded pkcs8 v0.10.2
117 Downloaded p256 v0.13.2
118 Downloaded rand_xorshift v0.3.0
119 Downloaded unarray v0.1.4
120 Downloaded rpassword v7.3.1
121 Downloaded darling_core v0.14.4
122 Downloaded darling_macro v0.14.4
123 Downloaded darling v0.14.4
124 Downloaded cached_proc_macro v0.17.0
125 Downloaded strum v0.25.0
126 Downloaded convert_case v0.4.0
127 Downloaded derive_more v0.99.18
128 Downloaded security-framework-sys v2.11.1
129 Downloaded serdect v0.2.0
130 Downloaded flatten-serde-json v0.1.0
```



```
131 Downloaded fs4 v0.7.0
132 Downloaded cached_proc_macro_types v0.1.1
133 Downloaded cached v0.44.0
134 Downloaded blockifier v0.8.0-rc.3
135 Downloaded ctor v0.2.8
136 Downloaded starknet-curve v0.3.0
137 Downloaded starknet-crypto v0.5.2
138 Downloaded minilp v0.2.2
139 Downloaded ascii-canvas v3.0.0
140 Downloaded bit-vec v0.6.3
141 Downloaded fixedbitset v0.4.2
142 Downloaded pico-args v0.5.0
143 Downloaded new_debug_unreachable v1.0.6
144 Downloaded siphasher v0.3.11
145 Downloaded phf_shared v0.10.0
146 Downloaded precomputed-hash v0.1.1
147 Downloaded sec1 v0.7.3
148 Downloaded elliptic-curve v0.13.8
149 Downloaded primeorder v0.13.6
150 Downloaded base16ct v0.2.0
151 Downloaded proptest v1.5.0
152 Downloaded phf_generator v0.11.2
153 Downloaded phf_macros v0.11.2
154 Downloaded phf_shared v0.11.2
155 Downloaded phf v0.11.2
156 Downloaded schannel v0.1.24
157 Downloaded fastrand v2.1.1
158 Downloaded winapi v0.3.9
159 Downloaded cairo-lang-eq-solver v2.8.2
160 Downloaded cairo-lang-sierra-gas v2.8.2
161 Downloaded const-fnv1a-hash v1.1.0
162 Downloaded core-foundation v0.9.4
163 Downloaded security-framework v2.11.1
164 Downloaded errno v0.3.9
165 Downloaded web-sys v0.3.70
166 Downloaded openssl v0.10.66
167 Downloaded rustix v0.38.37
168 Downloaded tokio v1.40.0
169 Downloaded webpki v0.22.4
170 Downloaded auto_impl v1.2.0
171 Downloaded cc v1.1.18
172 Downloaded redox_users v0.4.6
173 Downloaded mime v0.3.17
174 Downloaded serde_urlencoded v0.7.1
175 Downloaded vcpkg v0.2.15
176 Downloaded webpki-roots v0.22.6
177 Downloaded async-trait v0.1.83
178 Downloaded iana-time-zone-haiku v0.1.2
179 Downloaded windows-core v0.52.0
180 Downloaded dirs-next v2.0.0
181 Downloaded foreign-types-shared v0.1.1
182 Downloaded foreign-types v0.3.2
183 Downloaded openssl-macros v0.1.1
184 Downloaded openssl-probe v0.1.5
185 Downloaded pkg-config v0.3.30
186 Downloaded openssl-sys v0.9.103
187 Downloaded tempfile v3.13.0
188 Downloaded rustls v0.21.12
189 Downloaded gimli v0.31.0
190 Downloaded redox_syscall v0.5.3
191 Downloaded winreg v0.10.1
192 Downloaded windows-sys v0.48.0
193 Downloaded reqwest v0.11.18
194 Downloaded tokio-native-tls v0.3.1
195 Downloaded rustls-webpki v0.101.7
196 Downloaded addr2line v0.24.1
197 Downloaded rustc-demangle v0.1.24
198 Downloaded mio v1.0.2
199 Downloaded lock_api v0.4.12
200 Downloaded bitflags v2.6.0
```



Cairo Security Clan

```
201 Downloaded smallvec v1.13.2
202 Downloaded parking_lot v0.12.3
203 Downloaded signal-hook-registry v1.4.2
204 Downloaded socket2 v0.5.7
205 Downloaded object v0.36.4
206 Downloaded backtrace v0.3.74
207 Downloaded bytes v1.7.1
208 Downloaded hermit-abi v0.3.9
209 Downloaded scopeguard v1.2.0
210 Downloaded parking_lot_core v0.9.10
211 Downloaded pin-project-lite v0.2.14
212 Downloaded tokio-macros v2.4.0
213 Downloaded tokio-rustls v0.24.1
214 Downloaded tower-service v0.3.3
215 Downloaded wasm-bindgen-futures v0.4.43
216 Downloaded untrusted v0.9.0
217 Downloaded scarb-ui v0.1.5
218 Downloaded serde_path_to_error v0.1.16
219 Downloaded option-ext v0.2.0
220 Downloaded dirs-sys v0.4.1
221 Downloaded dirs v5.0.1
222 Downloaded shellexpand v3.1.0
223 Downloaded test-case-core v3.3.1
224 Downloaded test-case-macros v3.3.1
225 Downloaded test-case v3.3.1
226 Downloaded assert-json-diff v2.0.2
227 Downloaded rayon v1.10.0
228 Downloaded cairo-lang-compiler v2.8.2
229 Downloaded cairo-lang-lowering v2.8.2
230 Downloaded cairo-lang-plugins v2.8.2
231 Downloaded cairo-lang-proc-macros v2.8.2
232 Downloaded cairo-lang-diagnostics v2.8.2
233 Downloaded genco-macros v0.17.9
234 Downloaded rustversion v1.0.17
235 Downloaded term v0.7.0
236 Downloaded same-file v1.0.6
237 Downloaded winapi-util v0.1.9
238 Downloaded walkdir v2.5.0
239 Downloaded lalrpop v0.20.2
240 Downloaded lalrpop-util v0.20.2
241 Downloaded cairo-lang-sierra v2.8.2
242 Downloaded cairo-lang-sierra-type-size v2.8.2
243 Downloaded cairo-lang-sierra-to-casm v2.8.2
244 Downloaded unicode-segmentation v1.11.0
245 Downloaded convert_case v0.6.0
246 Downloaded cairo-lang-starknet-classes v2.8.2
247 Downloaded cairo-lang-debug v2.8.2
248 Downloaded tracing-core v0.1.32
249 Downloaded tracing v0.1.40
250 Downloaded try-lock v0.2.5
251 Downloaded want v0.3.1
252 Downloaded hyper v0.14.30
253 Downloaded hyper-tls v0.5.0
254 Downloaded ipnet v2.10.0
255 Downloaded matchit v0.7.3
256 Downloaded sync_wrapper v1.0.1
257 Downloaded sync_wrapper v0.1.2
258 Downloaded tower v0.5.1
259 Downloaded axum v0.7.7
260 Downloaded include_dir_macros v0.7.4
261 Downloaded include_dir v0.7.4
262 Downloaded cairo-lang-test-plugin v2.8.2
263 Downloaded bimap v0.6.3
264 Downloaded derive_more-impl v1.0.0
265 Downloaded derive_more v1.0.0
266 Downloaded cairo-annotations v0.1.0
267 Downloaded fs2 v0.4.3
268 Downloaded glob v0.3.1
269 Downloaded k256 v0.13.4
270 Downloaded signature v2.2.0
```



Cairo Security Clan

```
271 Downloaded ecdsa v0.16.9
272 Downloaded ff v0.13.0
273 Downloaded group v0.13.0
274 Downloaded pem-rfc7468 v0.7.0
275 Downloaded der v0.7.9
276 Downloaded spki v0.7.3
277 Downloaded cairo-lang-starknet v2.8.2
278 Downloaded schemars_derive v0.8.21
279 Downloaded h2 v0.3.26
280 Downloaded httpdate v1.0.3
281 Downloaded aes v0.8.4
282 Downloaded password-hash v0.4.2
283 Downloaded pbkdf2 v0.11.0
284 Downloaded uuid v0.8.2
285 Downloaded num_cpus v1.16.0
286 Downloaded deadpool v0.10.0
287 Downloaded http-body-util v0.1.2
288 Downloaded tower v0.4.13
289 Downloaded hyper-util v0.1.8
290 Downloaded wiremock v0.6.2
291 Downloaded data-encoding v2.6.0
292 Downloaded xxhash-rust v0.8.12
293 Downloaded scarb-stable-hash v1.0.0
294 Downloaded cairo-lang-macro-attributes v0.1.0
295 Downloaded cairo-lang-macro-stable v1.0.0
296 Downloaded linkme-impl v0.3.28
297 Downloaded linkme v0.3.28
298 Downloaded cairo-lang-macro v0.1.0
299 Downloaded ark-secp256k1 v0.4.0
300 Downloaded ark-poly v0.4.2
301 Downloaded hashbrown v0.13.2
302 Downloaded rust-analyzer-salsa-macros v0.17.0-pre.6
303 Downloaded triomphe v0.1.13
304 Downloaded const_format_proc_macros v0.2.33
305 Downloaded const_format v0.2.33
306 Downloaded serde_derive_internals v0.29.1
307 Downloaded schemars v0.8.21
308 Downloaded http v0.2.12
309 Downloaded http-body v0.4.6
310 Downloaded httparse v1.9.4
311 Downloaded tracing-attributes v0.1.27
312 Downloaded valuable v0.1.0
313 Downloaded ctr v0.9.2
314 Downloaded base64ct v1.6.0
315 Downloaded inout v0.1.3
316 Downloaded cipher v0.4.4
317 Downloaded salsa20 v0.10.2
318 Downloaded scrypt v0.10.0
319 Downloaded eth-keystore v0.5.0
320 Downloaded base64 v0.21.7
321 Downloaded crc32fast v1.4.2
322 Downloaded Adler2 v2.0.0
323 Downloaded miniz_oxide v0.8.0
324 Downloaded flate2 v1.0.34
325 Downloaded serde_json_pythonic v0.1.2
326 Downloaded android-tzdata v0.1.1
327 Downloaded android_system_properties v0.1.5
328 Downloaded core-foundation-sys v0.8.7
329 Downloaded jobserver v0.1.32
330 Downloaded clipboard-win v4.5.0
331 Downloaded fd-lock v3.0.13
332 Downloaded bitflags v1.3.2
333 Downloaded memoffset v0.6.5
334 Downloaded nix v0.23.2
335 Downloaded endian-type v0.1.2
336 Downloaded rustyline v9.1.2
337 Downloaded promptly v0.3.1
338 Downloaded predicates-core v1.0.8
339 Downloaded termtree v0.4.1
340 Downloaded predicates-tree v1.0.11
```



Cairo Security Clan

```
341 Downloaded assert_fs v1.1.2
342 Downloaded home v0.5.9
343 Downloaded which v5.0.0
344 Downloaded clap_lex v0.7.2
345 Downloaded clap_builder v4.5.18
346 Downloaded futures v0.3.30
347 Downloaded path-clean v1.0.1
348 Downloaded cairo-lang-filesystem v2.8.2
349 Downloaded oorandom v11.1.4
350 Downloaded heck v0.4.1
351 Downloaded rustc-hash v1.1.0
352 Downloaded stable_deref_trait v1.2.0
353 Downloaded rust-analyzer-salsa v0.17.0-pre.6
354 Downloaded unescaper v0.1.5
355 Downloaded cairo-lang-syntax v2.8.2
356 Downloaded unicode-xid v0.2.5
357 Downloaded indent v0.1.1
358 Downloaded indoc v2.0.5
359 Downloaded dyn-clone v1.0.17
360 Downloaded base64 v0.22.1
361 Downloaded shlex v1.3.0
362 Downloaded nibble_vec v0.1.0
363 Downloaded radix_trie v0.2.1
364 Downloaded http v1.1.0
365 Downloaded encoding_rs v0.8.34
366 Downloaded windows_i686_gnu v0.52.6
367 Downloaded windows_i686_msvc v0.52.6
368 Downloaded windows_x86_64_gnu v0.52.6
369 Downloaded error-code v2.3.1
370 Downloaded str-buf v1.0.6
371 Downloaded predicates v3.1.2
372 Downloaded atomic-waker v1.1.2
373 Downloaded h2 v0.4.6
374 Downloaded http-body v1.0.1
375 Downloaded hyper v1.4.1
376 Downloaded pin-project-internal v1.1.5
377 Downloaded pin-project v1.1.5
378 Downloaded tower-layer v0.3.3
379 Downloaded ark-ec v0.4.2
380 Downloaded ark-secp256r1 v0.4.0
381 Downloaded cairo-lang-utils v2.8.2
382 Downloaded bincode v2.0.0-rc.3
383 Downloaded lru v0.12.4
384 Downloaded num-modular v0.5.1
385 Downloaded num-prime v0.4.4
386 Downloaded rust_decimal v1.36.0
387 Downloaded starknet-crypto-codegen v0.3.3
388 Downloaded starknet-curve v0.4.2
389 Downloaded ark-ff-asm v0.4.2
390 Downloaded ark-ff-macros v0.4.2
391 Downloaded starknet-ff v0.3.7
392 Downloaded starknet-crypto v0.6.2
393 Downloaded thiserror-impl-no-std v2.0.2
394 Downloaded thiserror-no-std v2.0.2
395 Downloaded bzip2-sys v0.1.11+1.0.8
396 Downloaded zstd-sys v2.0.13+zstd.1.5.6
397 Downloaded cairo-vm v1.0.1
398 Downloaded cairo-lang-runner v2.8.2
399 Downloaded regex-automata v0.4.8
400 Downloaded normalize-line-endings v0.3.0
401 Downloaded snapbox-macros v0.3.10
402 Downloaded snapbox v0.4.17
403 Downloaded ethbloom v0.13.0
404 Downloaded winnow v0.6.18
405 Downloaded parity-scale-codec-derive v3.6.12
406 Downloaded parity-scale-codec v3.6.12
407 Downloaded windows_aarch64_msvc v0.52.6
408 Downloaded windows_x86_64_msvc v0.52.6
409 Downloaded serde_with_macros v3.9.0
410 Downloaded serde_with v3.9.0
```



```
411 Downloaded sha3 v0.10.8
412 Downloaded lambdaworks-crypto v0.7.0
413 Downloaded lambdaworks-math v0.7.0
414 Downloaded rayon-core v1.12.1
415 Downloaded cairo-lang-defs v2.8.2
416 Downloaded ark-serialize v0.4.2
417 Downloaded ark-std v0.4.0
418 Downloaded ark-ff v0.4.2
419 Downloaded zstd-safe v5.0.2+zstd.1.5.2
420 Downloaded encode_unicode v0.3.6
421 Downloaded unicode-width v0.1.13
422 Downloaded impl-trait-for-tuples v0.2.2
423 Downloaded impl-serde v0.4.0
424 Downloaded primitive-types v0.12.2
425 Downloaded openssl-src v300.3.2+3.3.2
426 Downloaded ethereum-types v0.14.1
427 Downloaded windows_aarch64_gnullvm v0.52.6
428 Downloaded windows_x86_64_gnullvm v0.52.6
429 Downloaded rfc6979 v0.4.0
430 Downloaded libc v0.2.158
431 Downloaded sha1 v0.10.6
432 Downloaded zstd v0.11.2+zstd.1.5.2
433 Downloaded zip v0.6.6
434 Downloaded either v1.13.0
435 Downloaded itertools v0.12.1
436 Downloaded regex-syntax v0.8.5
437 Downloaded anstyle-parse v0.2.5
438 Downloaded anstyle v1.0.8
439 Downloaded tiny-keccak v2.0.2
440 Downloaded arrayvec v0.7.6
441 Downloaded radium v0.7.0
442 Downloaded tap v1.0.1
443 Downloaded toml_datetime v0.6.8
444 Downloaded syn v1.0.109
445 Downloaded rlp v0.5.2
446 Downloaded static_assertions v1.1.0
447 Downloaded uint v0.9.5
448 Downloaded windows_i686_gnullvm v0.52.6
449 Downloaded hashbrown v0.12.3
450 Downloaded hashbrown v0.14.5
451 Downloaded regex v1.11.0
452 Downloaded bitvec v1.0.1
453 Downloaded strsim v0.11.1
454 Downloaded time-macros v0.2.18
455 Downloaded generic-array v0.14.7
456 Downloaded typenum v1.17.0
457 Downloaded subtle v2.6.1
458 Downloaded lazy_static v1.5.0
459 Downloaded byteorder v1.5.0
460 Downloaded js-sys v0.3.70
461 Downloaded wasi v0.11.0+wasi-snapshot-preview1
462 Downloaded wasm-bindgen-backend v0.2.93
463 Downloaded wasm-bindgen-macro-support v0.2.93
464 Downloaded wasm-bindgen v0.2.93
465 Downloaded getrandom v0.2.15
466 Downloaded num-integer v0.1.46
467 Downloaded num-traits v0.2.19
468 Downloaded proc-macro2 v1.0.86
469 Downloaded unicode-ident v1.0.12
470 Downloaded unicode-normalization v0.1.23
471 Downloaded idna v0.5.0
472 Downloaded itertools v0.10.5
473 Downloaded rustc_version v0.4.1
474 Downloaded bzip2 v0.4.4
475 Downloaded console v0.15.8
476 Downloaded aho-corasick v1.1.3
477 Downloaded similar v2.6.0
478 Downloaded anstyle-wincon v3.0.4
479 Downloaded colorchoice v1.0.2
480 Downloaded wyz v0.5.1
```



```
481 Downloaded byte-slice-cast v1.2.2
482 Downloaded serde_spanned v0.6.7
483 Downloaded toml_edit v0.22.22
484 Downloaded proc-macro-crate v3.2.0
485 Downloaded impl-codec v0.6.0
486 Downloaded rustc-hex v2.1.0
487 Downloaded impl-rlp v0.3.0
488 Downloaded crunchy v0.2.2
489 Downloaded hyper-rustls v0.24.2
490 Downloaded tokio-util v0.7.12
491 Downloaded nom v7.1.3
492 Downloaded ark-serialize-derive v0.4.2
493 Downloaded paste v1.0.15
494 Downloaded constant_time_eq v0.1.5
495 Downloaded semver v1.0.23
496 Downloaded anstyle-query v1.1.1
497 Downloaded is_terminal_polyfill v1.70.1
498 Downloaded utf8parse v0.2.2
499 Downloaded anstream v0.6.15
500 Downloaded fixed-hash v0.8.0
501 Downloaded funty v2.0.0
502 Downloaded futures-channel v0.3.30
503 Downloaded futures-io v0.3.30
504 Downloaded futures-macro v0.3.30
505 Downloaded futures-task v0.3.30
506 Downloaded pin-utils v0.1.0
507 Downloaded futures-util v0.3.30
508 Downloaded chrono v0.4.38
509 Downloaded indexmap v1.9.3
510 Downloaded ahash v0.8.11
511 Downloaded allocator-api2 v0.2.18
512 Downloaded indexmap v2.5.0
513 Downloaded darling_core v0.20.10
514 Downloaded time v0.3.36
515 Downloaded hmac v0.12.1
516 Downloaded thiserror v1.0.64
517 Downloaded serde_json v1.0.128
518 Downloaded equivalent v1.0.1
519 Downloaded fnv v1.0.7
520 Downloaded ident_case v1.0.1
521 Downloaded darling_macro v0.20.10
522 Downloaded darling v0.20.10
523 Downloaded deranged v0.3.11
524 Downloaded time-core v0.1.2
525 Downloaded hex v0.4.3
526 Downloaded thiserror-impl v1.0.64
527 Downloaded block-buffer v0.10.4
528 Downloaded zeroize_derive v1.4.2
529 Downloaded zeroize v1.8.1
530 Downloaded crypto-common v0.1.6
531 Downloaded digest v0.10.7
532 Downloaded cpufeatures v0.2.14
533 Downloaded keccak v0.1.5
534 Downloaded itoa v1.0.11
535 Downloaded zerocopy-derive v0.7.35
536 Downloaded cfg-if v1.0.0
537 Downloaded bumpalo v3.16.0
538 Downloaded once_cell v1.19.0
539 Downloaded syn v2.0.79
540 Downloaded futures-core v0.3.30
541 Downloaded iana-time-zone v0.1.60
542 Downloaded log v0.4.22
543 Downloaded wasm-bindgen-shared v0.2.93
544 Downloaded wasm-bindgen-macro v0.2.93
545 Downloaded libm v0.2.8
546 Downloaded serde_derive v1.0.210
547 Downloaded serde v1.0.210
548 Downloaded unicode-bidi v0.3.15
549 Downloaded tinyvec_macros v0.1.1
550 Downloaded tinyvec v1.8.0
```




Cairo Security Clan

```
551 Downloaded url v2.5.2
552 Downloaded powerfmt v0.2.0
553 Downloaded num-conv v0.1.0
554 Downloaded crypto-bigint v0.5.5
555 Downloaded sha2 v0.10.8
556 Downloaded const-oid v0.9.6
557 Downloaded version_check v0.9.5
558 Downloaded memchr v2.7.4
559 Downloaded starknet-types-core v0.1.6
560 Downloaded spin v0.9.8
561 Downloaded zerocopy v0.7.35
562 Downloaded ppv-lite86 v0.2.20
563 Downloaded rand_chacha v0.3.1
564 Downloaded rand_core v0.6.4
565 Downloaded rand v0.8.5
566 Downloaded num-bigint v0.4.6
567 Downloaded smol_str v0.2.2
568 Downloaded form_urlencoded v1.2.1
569 Compiling snforge_scarb_plugin v0.32.0 (git+github.com/foundry-rs/starknet-foundry?tag=v0.32.0#3817
    c903b640201c72e743b9bbe70a97149828a2)
570 Compiling proc-macro2 v1.0.86
571 Compiling unicode-ident v1.0.12
572 Compiling version_check v0.9.5
573 Compiling libc v0.2.158
574 Compiling autocfg v1.3.0
575 Compiling quote v1.0.37
576 Compiling syn v2.0.79
577 Compiling cfg-if v1.0.0
578 Compiling serde v1.0.210
579 Compiling typenum v1.17.0
580 Compiling generic-array v0.14.7
581 Compiling byteorder v1.5.0
582 Compiling getrandom v0.2.15
583 Compiling rand_core v0.6.4
584 Compiling ahash v0.8.11
585 Compiling num-traits v0.2.19
586 Compiling serde_derive v1.0.210
587 Compiling zerocopy-derive v0.7.35
588 Compiling zerocopy v0.7.35
589 Compiling ppv-lite86 v0.2.20
590 Compiling rand_chacha v0.3.1
591 Compiling rand v0.8.5
592 Compiling num-integer v0.1.46
593 Compiling once_cell v1.19.0
594 Compiling num-bigint v0.4.6
595 Compiling crypto-common v0.1.6
596 Compiling itoa v1.0.11
597 Compiling memchr v2.7.4
598 Compiling subtle v2.6.1
599 Compiling allocator-api2 v0.2.18
600 Compiling hashbrown v0.14.5
601 Compiling block-buffer v0.10.4
602 Compiling digest v0.10.7
603 Compiling spin v0.9.8
604 Compiling ryu v1.0.18
605 Compiling either v1.13.0
606 Compiling equivalent v1.0.1
607 Compiling indexmap v2.5.0
608 Compiling serde_json v1.0.128
609 Compiling pin-project-lite v0.2.14
610 Compiling lock_api v0.4.12
611 Compiling syn v1.0.109
612 Compiling parking_lot_core v0.9.10
613 Compiling tracing-core v0.1.32
614 Compiling tracing-attributes v0.1.27
615 Compiling tracing v0.1.40
616 Compiling jobserver v0.1.32
617 Compiling shlex v1.3.0
618 Compiling cc v1.1.18
619 Compiling lazy_static v1.5.0
```




```
620 Compiling cpufeatures v0.2.14
621 Compiling zeroize_derive v1.4.2
622 Compiling indexmap v1.9.3
623 Compiling thiserror v1.0.64
624 Compiling thiserror-impl v1.0.64
625 Compiling serde_derive_internals v0.29.1
626 Compiling hashbrown v0.12.3
627 Compiling schemars v0.8.21
628 Compiling schemars_derive v0.8.21
629 Compiling sha2 v0.10.8
630 Compiling smallvec v1.13.2
631 Compiling dyn-clone v1.0.17
632 Compiling itertools v0.12.1
633 Compiling log v0.4.22
634 Compiling cairo-lang-utils v2.8.2
635 Compiling scopeguard v1.2.0
636 Compiling derivative v2.2.0
637 Compiling heck v0.4.1
638 Compiling keccak v0.1.5
639 Compiling stable_deref_trait v1.2.0
640 Compiling triomphe v0.1.13
641 Compiling sha3 v0.10.8
642 Compiling rust-analyzer-salsa-macros v0.17.0-pre.6
643 Compiling parking_lot v0.12.3
644 Compiling rustc-hash v1.1.0
645 Compiling oorandom v11.1.4
646 Compiling rust-analyzer-salsa v0.17.0-pre.6
647 Compiling smol_str v0.2.2
648 Compiling lambdaworks-math v0.7.0
649 Compiling lambdaworks-crypto v0.7.0
650 Compiling starknet-types-core v0.1.6
651 Compiling aho-corasick v1.1.3
652 Compiling regex-syntax v0.8.5
653 Compiling regex-automata v0.4.8
654 Compiling zeroize v1.8.1
655 Compiling bytes v1.7.1
656 Compiling fnv v1.0.7
657 Compiling anyhow v1.0.89
658 Compiling semver v1.0.23
659 Compiling dirs-sys-next v0.1.2
660 Compiling crunchy v0.2.2
661 Compiling paste v1.0.15
662 Compiling hex v0.4.3
663 Compiling dirs-next v2.0.0
664 Compiling cairo-lang-debug v2.8.2
665 Compiling ring v0.17.8
666 Compiling hmac v0.12.1
667 Compiling siphasher v0.3.11
668 Compiling crossbeam-utils v0.8.20
669 Compiling tiny-keccak v2.0.2
670 Compiling genco-macros v0.17.9
671 Compiling unicode-xid v0.2.5
672 Compiling path-clean v1.0.1
673 Compiling cairo-lang-file-system v2.8.2
674 Compiling phf_shared v0.10.0
675 Compiling term v0.7.0
676 Compiling ark-ff-macros v0.4.2
677 Compiling ark-ff-asm v0.4.2
678 Compiling new_debug_unreachable v1.0.6
679 Compiling same-file v1.0.6
680 Compiling precomputed-hash v0.1.1
681 Compiling ident_case v1.0.1
682 Compiling bit-vec v0.6.3
683 Compiling fixedbitset v0.4.2
684 Compiling strsim v0.11.1
685 Compiling darling_core v0.20.10
686 Compiling petgraph v0.6.5
687 Compiling bit-set v0.5.3
688 Compiling ena v0.14.3
689 Compiling string_cache v0.8.7
```



```
690 Compiling walkdir v2.5.0
691 Compiling ascii-canvas v3.0.0
692 Compiling regex v1.11.0
693 Compiling lalrpop-util v0.20.2
694 Compiling itertools v0.11.0
695 Compiling inout v0.1.3
696 Compiling num-complex v0.2.4
697 Compiling xshell-macros v0.2.6
698 Compiling pico-args v0.5.0
699 Compiling pkg-config v0.3.30
700 Compiling untrusted v0.9.0
701 Compiling relative-path v1.9.3
702 Compiling genco v0.17.9
703 Compiling lalrpop v0.20.2
704 Compiling xshell v0.2.6
705 Compiling cipher v0.4.4
706 Compiling crossbeam-epoch v0.9.18
707 Compiling darling_macro v0.20.10
708 Compiling crypto-bigint v0.5.5
709 Compiling unescaper v0.1.5
710 Compiling rawpointer v0.2.1
711 Compiling serde_json_pythonic v0.1.2
712 Compiling ndarray v0.13.1
713 Compiling matrixmultiply v0.2.4
714 Compiling cairo-lang-sierra v2.8.2
715 Compiling cairo-lang-syntax v2.8.2
716 Compiling darling v0.20.10
717 Compiling crossbeam-deque v0.8.5
718 Compiling cairo-lang-syntax-codegen v2.8.2
719 Compiling cairo-lang-diagnostics v2.8.2
720 Compiling ark-serialize-derive v0.4.2
721 Compiling colored v2.1.0
722 Compiling ark-std v0.4.0
723 Compiling socket2 v0.5.7
724 Compiling mio v1.0.2
725 Compiling unicode-segmentation v1.11.0
726 Compiling futures-core v0.3.30
727 Compiling Adler2 v2.0.0
728 Compiling base64ct v1.6.0
729 Compiling password-hash v0.4.2
730 Compiling miniz_oxide v0.8.0
731 Compiling convert_case v0.6.0
732 Compiling tokio v1.40.0
733 Compiling ark-serialize v0.4.2
734 Compiling cairo-lang-parser v2.8.2
735 Compiling serde_with_macros v3.9.0
736 Compiling rfc6979 v0.4.0
737 Compiling itertools v0.10.5
738 Compiling bstr v1.10.0
739 Compiling num-traits v0.1.43
740 Compiling crc32fast v1.4.2
741 Compiling slab v0.4.9
742 Compiling indoc v2.0.5
743 Compiling const-fnv1a-hash v1.1.0
744 Compiling overload v0.1.1
745 Compiling rustc-hex v2.1.0
746 Compiling nu-ansi-term v0.46.0
747 Compiling globset v0.4.15
748 Compiling flate2 v1.0.34
749 Compiling sprs v0.7.1
750 Compiling ark-ff v0.4.2
751 Compiling pbkdf2 v0.11.0
752 Compiling http v0.2.12
753 Compiling serde_spanned v0.6.7
754 Compiling toml_datetime v0.6.8
755 Compiling futures-sink v0.3.30
756 Compiling futures-task v0.3.30
757 Compiling winnow v0.6.18
758 Compiling static_assertions v1.1.0
759 Compiling rustls v0.21.12
```



```
760 Compiling httparse v1.9.4
761 Compiling pin-utils v0.1.0
762 Compiling tinyvec_macros v0.1.1
763 Compiling tinyvec v1.8.0
764 Compiling futures-util v0.3.30
765 Compiling toml_edit v0.22.22
766 Compiling tokio-util v0.7.12
767 Compiling minilp v0.2.2
768 Compiling ignore v0.4.23
769 Compiling diffy v0.3.0
770 Compiling rlp v0.5.2
771 Compiling aes v0.8.4
772 Compiling sct v0.7.1
773 Compiling rustls-webpki v0.101.7
774 Compiling zstd-sys v2.0.13+zstd.1.5.6
775 Compiling yansi v0.5.1
776 Compiling percent-encoding v2.3.1
777 Compiling try-lock v0.2.5
778 Compiling base64 v0.21.7
779 Compiling diff v0.1.13
780 Compiling pretty_assertions v1.4.0
781 Compiling want v0.3.1
782 Compiling form_urlencoded v1.2.1
783 Compiling impl-rlp v0.3.0
784 Compiling cairo-lang-formatter v2.8.2
785 Compiling h2 v0.3.26
786 Compiling good_lp v1.8.1
787 Compiling toml v0.8.19
788 Compiling unicode-normalization v0.1.23
789 Compiling fixed-hash v0.8.0
790 Compiling http-body v0.4.6
791 Compiling cairo-lang-defs v2.8.2
792 Compiling futures-channel v0.3.30
793 Compiling starknet-curve v0.5.0 (github.com/xJonathanLEI/starknet-rs?rev=660a732#660a7323)
794 Compiling impl-serde v0.4.0
795 Compiling tower-service v0.3.3
796 Compiling httpdate v1.0.3
797 Compiling indent v0.1.1
798 Compiling unicode-bidi v0.3.15
799 Compiling idna v0.5.0
800 Compiling cairo-lang-plugins v2.8.2
801 Compiling hyper v0.14.30
802 Compiling starknet-crypto v0.7.1 (github.com/xJonathanLEI/starknet-rs?rev=660a732#660a7323)
803 Compiling cairo-lang-proc-macros v2.8.2
804 Compiling cairo-lang-eq-solver v2.8.2
805 Compiling cairo-lang-test-utils v2.8.2
806 Compiling tokio-rustls v0.24.1
807 Compiling uint v0.9.5
808 Compiling cairo-lang-sierra-type-size v2.8.2
809 Compiling serde_with v3.9.0
810 Compiling salsa20 v0.10.2
811 Compiling webpki v0.22.4
812 Compiling bzip2-sys v0.1.11+1.0.8
813 Compiling zstd-safe v5.0.2+zstd.1.5.2
814 Compiling id-arena v2.2.1
815 Compiling rayon-core v1.12.1
816 Compiling radium v0.7.0
817 Compiling starknet-core v0.11.1 (github.com/xJonathanLEI/starknet-rs?rev=660a732#660a7323)
818 Compiling cairo-lang-semantic v2.8.2
819 Compiling webpki-roots v0.22.6
820 Compiling scrypt v0.10.0
821 Compiling primitive-types v0.12.2
822 Compiling ethbloom v0.13.0
823 Compiling hyper-rustls v0.24.2
824 Compiling starknet-ff v0.3.7
825 Compiling url v2.5.2
826 Compiling serde_urlencoded v0.7.1
827 Compiling rustls-pemfile v1.0.4
828 Compiling ctr v0.9.2
829 Compiling hashbrown v0.13.2
```



```
830 Compiling uuid v0.8.2
831 Compiling async-trait v0.1.83
832 Compiling auto_impl v1.2.0
833 Compiling encoding_rs v0.8.34
834 Compiling tap v1.0.1
835 Compiling ipnet v2.10.0
836 Compiling powerfmt v0.2.0
837 Compiling mime v0.3.17
838 Compiling reqwest v0.11.18
839 Compiling deranged v0.3.11
840 Compiling wyz v0.5.1
841 Compiling eth-keystore v0.5.0
842 Compiling ark-poly v0.4.2
843 Compiling starknet-curve v0.4.2
844 Compiling ethereum-types v0.14.1
845 Compiling cairo-lang-lowering v2.8.2
846 Compiling cairo-lang-sierra-gas v2.8.2
847 Compiling cairo-lang-sierra-ap-change v2.8.2
848 Compiling cairo-lang-casm v2.8.2
849 Compiling rust_decimal v1.36.0
850 Compiling funty v2.0.0
851 Compiling assert_matches v1.5.0
852 Compiling num-conv v0.1.0
853 Compiling time-core v0.1.2
854 Compiling utf8parse v0.2.2
855 Compiling anstyle-parse v0.2.5
856 Compiling time v0.3.36
857 Compiling cairo-lang-sierra-to-casm v2.8.2
858 Compiling bitvec v1.0.1
859 Compiling rayon v1.10.0
860 Compiling cairo-lang-sierra-generator v2.8.2
861 Compiling bzip2 v0.4.4
862 Compiling starknet-providers v0.11.0 (github.com/xJonathanLEI/starknet-rs?rev=660a732#660a7323)
863 Compiling starknet-crypto-codegen v0.3.3
864 Compiling zstd v0.11.2+zstd.1.5.2
865 Compiling ark-ec v0.4.2
866 Compiling starknet-signers v0.9.0 (github.com/xJonathanLEI/starknet-rs?rev=660a732#660a7323)
867 Compiling cairo-lang-project v2.8.2
868 Compiling const_format_proc_macros v0.2.33
869 Compiling thiserror-impl-no-std v2.0.2
870 Compiling sha1 v0.10.6
871 Compiling lru v0.12.4
872 Compiling num-modular v0.5.1
873 Compiling anstyle-query v1.1.1
874 Compiling minimal-lexical v0.2.1
875 Compiling is_terminal_polyfill v1.70.1
876 Compiling anstyle v1.0.8
877 Compiling colorchoice v1.0.2
878 Compiling arrayvec v0.7.6
879 Compiling constant_time_eq v0.1.5
880 Compiling zip v0.6.6
881 Compiling anstream v0.6.15
882 Compiling nom v7.1.3
883 Compiling num-prime v0.4.4
884 Compiling thiserror-no-std v2.0.2
885 Compiling const_format v0.2.33
886 Compiling starknet-accounts v0.10.0 (github.com/xJonathanLEI/starknet-rs?rev=660a732#660a7323)
887 Compiling cairo-lang-compiler v2.8.2
888 Compiling starknet-crypto v0.6.2
889 Compiling cairo-lang-starknet-classes v2.8.2
890 Compiling bincode v2.0.0-rc.3
891 Compiling xxhash-rust v0.8.12
892 Compiling data-encoding v2.6.0
893 Compiling scarb-stable-hash v1.0.0
894 Compiling cairo-vm v1.0.1
895 Compiling cairo-lang-starknet v2.8.2
896 Compiling starknet-contract v0.10.0 (github.com/xJonathanLEI/starknet-rs?rev=660a732#660a7323)
897 Compiling snapbox-macros v0.3.10
898 Compiling starknet-macros v0.2.0 (github.com/xJonathanLEI/starknet-rs?rev=660a732#660a7323)
899 Compiling ark-secp256r1 v0.4.0
```



```

900   Compiling ark-secp256k1 v0.4.0
901   Compiling linkme-impl v0.3.28
902   Compiling unicode-width v0.1.13
903   Compiling similar v2.6.0
904   Compiling normalize-line-endings v0.3.0
905   Compiling snapbox v0.4.17
906   Compiling console v0.15.8
907   Compiling linkme v0.3.28
908   Compiling cairo-lang-runner v2.8.2
909   Compiling starknet v0.11.0 (github.com/xJonathanLEI/starknet-rs?rev=660a732#660a7323)
910   Compiling cairo-lang-macro-attributes v0.1.0
911   Compiling cairo-lang-macro-stable v1.0.0
912   Compiling cairo-lang-macro v0.1.0
913   Compiling shared v0.1.0 (/home/runner/.cache/scarb/registry/git/checkouts/starknet-foundry-v6206fshfutuo/3817
c903/crates/shared)
914   Compiling snforge-scarb-plugin v0.32.0 (/home/runner/.cache/scarb/registry/git/checkouts/starknet-foundry-
v6206fshfutuo/3817c903/crates/snforge-scarb-plugin)
915   Finished `release` profile [optimized] target(s) in 9m 52s
916   Compiling test(lst_unittest) lst v0.1.0 (/home/runner/work/031-KARNOT-LST/031-KARNOT-LST/contractsafteraudit/
Scarb.toml)
917   Finished `dev` profile target(s) in 11 minutes
918   Collected 31 test(s) from lst package
919   Running 31 test(s) from src/
920   [PASS] lst::delegator::tests::test_delegator::test_delegator::claim_rewards_contract_paused (gas: ~9341)
921   [PASS] lst::delegator::tests::test_delegator::test_delegator::claim_rewards_ok (gas: ~9983)
922   [PASS] lst::delegator::tests::test_delegator::test_delegator::claim_rewards_only_lst_can_call (gas: ~9262)
923   [PASS] lst::delegator::tests::test_delegator::test_delegator::send_to_withdraw_queue_zero_amount_ok (gas: ~9280)
924   [PASS] lst::delegator::tests::test_delegator::test_delegator::stake_contract_paused (gas: ~9338)
925   [PASS] lst::delegator::tests::test_delegator::test_delegator::stake_ok (gas: ~9867)
926   [PASS] lst::delegator::tests::test_delegator::test_delegator::stake_only_lst_can_call (gas: ~9259)
927   [PASS] lst::delegator::tests::test_delegator::test_delegator::start_delegation_contract_paused (gas: ~9415)
928   [PASS] lst::delegator::tests::test_delegator::test_delegator::start_delegation_ok (gas: ~9675)
929   [PASS] lst::delegator::tests::test_delegator::test_delegator::start_delegation_only_lst_can_call (gas: ~9258)
930   [PASS] lst::delegator::tests::test_delegator::test_delegator::start_unstake_intent_contract_paused (gas: ~9334)
931   [PASS] lst::delegator::tests::test_delegator::test_delegator::start_unstake_intent_not_owner (gas: ~9257)
932   [PASS] lst::delegator::tests::test_delegator::test_delegator::start_unstake_intent_intent_in_progress (gas:
~11877)
933   [PASS] lst::delegator::tests::test_delegator::test_delegator::test_delegator_reinitialize_should_fail (gas:
~9263)
934   [PASS] lst::delegator::tests::test_delegator::test_delegator::start_unstake_intent_ok (gas: ~11913)
935   [PASS] lst::delegator::tests::test_delegator::test_delegator::unstake_action_contract_paused (gas: ~9337)
936   [PASS] lst::delegator::tests::test_delegator::test_delegator::unstake_action_ok (gas: ~39726)
937   [PASS] lst::lst::tests::test_lst_config::test_lst_additional::test_add_delegator_non_admin (gas: ~1557)
938   [PASS] lst::lst::tests::test_lst_config::test_lst_additional::test_lst_delegator_funcs (gas: ~2565)
939   [PASS] lst::lst::tests::test_lst_config::test_lst_additional::test_reinitialize_should_fail (gas: ~1554)
940   [PASS] lst::lst::tests::test_lst_config::test_lst_additional::test_set_config_non_admin (gas: ~1568)
941   [PASS] lst::lst::tests::test_lst_config::test_lst_additional::test_update_delegator_info_non_admin (gas: ~1557)
942   [PASS] lst::lst::tests::test_lst_deposit::test_lst_deposit::test_constructor (gas: ~1604)
943   [PASS] lst::lst::tests::test_integration::test_integration::test_integration (gas: ~47865)
944   [PASS] lst::lst::tests::test_lst_deposit::test_lst_deposit::test_deposit_no_fee (gas: ~2027)
945   [PASS] lst::lst::tests::test_lst_deposit::test_lst_deposit::test_deposit_with_fee (gas: ~2448)
946   [PASS] lst::lst::tests::test_lst_mint::test_lst_mint::test_mint_with_fees (gas: ~2374)
947   [PASS] lst::lst::tests::test_lst_mint::test_lst_mint::test_mint_without_fees (gas: ~1959)
948   [PASS] lst::withdrawal_queue::tests::test_withdrawal_queue::test_withdrawal_queue::
test_request_withdrawal_invalid_caller (gas: ~2650)
949   [PASS] lst::withdrawal_queue::tests::test_withdrawal_queue::test_withdrawal_queue::test_full_withdrawal_queue (
gas: ~5479)
950   [PASS] lst::withdrawal_queue::tests::test_withdrawal_queue::test_withdrawal_queue::test_wq_constructor (gas:
~3265)
951   Tests: 31 passed, 0 failed, 0 skipped, 0 ignored, 0 filtered out

```