Quantstamp Security Assessment Certificate

October 22nd 2021 — Quantstamp Verified

Solana Stake Pool

This security assessment was prepared by Quantstamp, the leader in blockchain security

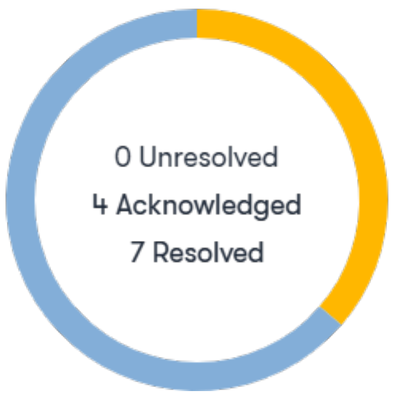
Executive Summary

Type	Stake Pool on Solana
Auditors	Poming Lee, Research Engineer Joseph Xu, Technical R&D Advisor Jake Goh Si Yuan, Senior Security Researcher
Timeline	2021-09-06 through 2021-10-22
Languages	Rust
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	Introduction to stake-pool program
Documentation Quality	Medium
Test Quality	Medium

Source Code

Repository	Commit
solana-program-library	99914c9
solana-program-library	3b48fa0

Total Issues	**11** (7 Resolved)
High Risk Issues	0 (0 Resolved)
Medium Risk Issues	**3** (3 Resolved)
Low Risk Issues	**5** (3 Resolved)
Informational Risk Issues	**3** (1 Resolved)
Undetermined Risk Issues	0 (0 Resolved)



High Risk	The issue puts a large number of users' sensitive information at risk, or is reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
Medium Risk	The issue puts a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or is reasonably likely to lead to moderate financial impact.
Low Risk	The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low-impact in view of the client's business circumstances.
Informational	The issue does not post an immediate risk, but is relevant to security best practices or Defence in Depth.
Undetermined	The impact of the issue is uncertain.

Unresolved	Acknowledged the existence of the risk, and decided to accept it without engaging in special efforts to control it.
Acknowledged	The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings).
Resolved	Adjusted program implementation, requirements or constraints to eliminate the risk.
Mitigated	Implemented actions to minimize the impact or likelihood of the risk.

Summary of Findings

The project is mostly well-written, though code comments could be improved. During the audit, we found 11 potential issues of various levels of severity: 3 medium-severity, 5 low-severity issues, as well as 3 informational-severity issues. We also made 8 best practices recommendations.
We highly recommend addressing the findings before going live.
Disclaimer: Please note that only the on-chain program in the `stake-pool` folder in the repository is audited.
2021-10-22: during this reaudit, the admin team has either brought all the status of findings into fixed or acknowledged.

ID	Description	Severity	Status
QSP-1	Inconsistency in pool token minting between `DepositSol` and `DepositStake` instructions	Medium	Fixed
QSP-2	Unsafe external program call to `token_programs`	Medium	Fixed
QSP-3	`DecreaseValidatorStake` may lead to insufficient amount of SOL in the validator stake account	Medium	Fixed
QSP-4	`MINIMUM_ACTIVE_STAKE` is missing from all validator stake information	Low	Fixed
QSP-5	Initial values not enforced on certain stake pool fields	Low	Fixed
QSP-6	Contradictory information on SOL transferred while merging stake accounts	Low	Fixed
QSP-7	Reliance on cloned `stake_program` code	Low	Acknowledged
QSP-8	New staker could be zero address	Low	Acknowledged
QSP-9	Unused stake pool lockup information	Informational	Acknowledged
QSP-10	Susceptibility to overflow	Informational	Fixed
QSP-11	Unmaintained crates are used	Informational	Acknowledged

Quantstamp Audit Breakdown

Quantstamp's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.

Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow / underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Methodology

The Quantstamp auditing process follows a routine series of steps:

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

Toolset

The notes below outline the setup and steps performed in the process of this audit.

Setup

Tool Setup:

- Rust Audit v0.15.0

- Rust-Clippy Latest

Steps taken to run the tools:

Rust Audit:

1. `cargo install cargo-audit`
2. `cargo audit`

Rust-Clippy:

1. `rustup component add clippy`
2. `cargo clippy`

Findings

QSP-1 Inconsistency in pool token minting between `DepositSol` and `DepositStake` instructions

Severity: *Medium Risk*

Status: Fixed

File(s) affected: `stake-pool\program\src\processor.rs`

Description: Users can receive stake pool tokens in two ways: by depositing SOL directly into the pool's reserve account using the `DepositSol` instruction, or by merging the stake account using the `DepositStake` instruction. However, if the source account has undelegated SOL (e.g., stake all of the SOL except the rent-exempt amount => receive additional SOL but the stake program's re-delegate has not been called yet), `DepositStake` will not mint additional stake pool tokens even though these extra SOL are ultimately transferred to the reserve account as if using the `DepositSol` instruction.
The relevant calculation can be found in `L1882-L1899` of `stake-pool\program\src\processor.rs`:

```
        let all_deposit_lamports = post_all_validator_lamports
            .checked_sub(pre_all_validator_lamports)
            .ok_or(StakePoolError::CalculationFailure)?;
        let stake_deposit_lamports = post_validator_stake
            .delegation
            .stake
            .checked_sub(validator_stake.delegation.stake)
            .ok_or(StakePoolError::CalculationFailure)?;
        let additional_lamports = all_deposit_lamports
            .checked_sub(stake_deposit_lamports)
            .ok_or(StakePoolError::CalculationFailure)?;
        let credited_additional_lamports = additional_lamports.min(unactivated_stake_rent);
        let credited_deposit_lamports =
            stake_deposit_lamports.saturating_add(credited_additional_lamports);

        let new_pool_tokens = stake_pool
            .calc_pool_tokens_for_deposit(credited_deposit_lamports)
            .ok_or(StakePoolError::CalculationFailure)?;
```

The amount of new pool tokens generated are calculated based on the variable `credited_deposit_lamports`. This amount is the delta in the validator stake pre- and post-merge (the `stake_deposit_lamports` variable) plus any extras (the `credited_additional_lamports`), which should include any extra SOL that was present but not yet staked in the source stake account.
However, `credited_additional_lamports` does not represent the extra SOL that were transferred from the source stake account (the variable `additional_lamports`), because the variable currently takes the minimum of the extra SOL that are transferred and the rent exempt amount (or zero). Thus, using `DepositStake` alone would lead to a user to miss out on the pool token from the extra SOL on the source stake account. On the other hand, the user can receive more pool tokens by first depositing the extra SOL to the reserve account using the `DepositSol` instruction and then merging with the `DepositStake` instructions.

Recommendation: The audit team is not able to verify the intended implementation through documentation at this time. However, based on the understanding that the merge instruction in the stake pool seems to be transferring all of the SOL from the source account to the destination account including the rent exempt amount, it appears that `L1893` is unnecessary and `credited_deposit_lamports` should simply equal `additional_lamports`. Alternatively, `L1893` might be modified to `let credited_additional_lamports = additional_lamports.checked_sub(unactivated_stake_rent).ok_or(StakePoolError::CalculationFailure);` if the rent exempt amount should not be given credit.

QSP-2 Unsafe external program call to `token_program`s

Severity: *Medium Risk*

Status: Fixed

File(s) affected: `stake-pool\program\src\processor.rs`

Description: The `token_program` that is used for the management of the pool token can be arbitrarily assigned by the pool creator. Therefore, the underlying logic of the critical `token_program` component cannot be guaranteed, and may lead to unexpected or even malicious behavior when called upon.

Recommendation: Restrict the `token_program` used for pool token management to the one deployed by the Solana Foundation, similar to the validation done for `stake_program_info` in other functions such as `process_add_validator_to_pool()`.

QSP-3 `DecreaseValidatorStake` may lead to insufficient amount of SOL in the validator stake account

Severity: *Medium Risk*

Status: Fixed

File(s) affected: `stake-pool\program\src\processor.rs`

Description: The function `process_decrease_validator_stake()` only checks for the amount of SOL to be split into the transient stake account and does not check for the remaining amount of SOL in the validator stake account. This can cause the validator stake account to have SOL below the rent-exempt amount, or the rent-exempt amount plus the minimum active stake amount.

Recommendation: Add a check to ensure that at least `lib::minimum_stake_lamports()` amount of SOL will remain in the validator stake account after `DecreaseValidatorStake` (similar to the checks in `process_withdraw_stake()`).

QSP-4 `MINIMUM_ACTIVE_STAKE` is missing from all validator stake information

Severity: *Low Risk*

Status: Fixed

File(s) affected: `stake-pool\program\src\processor.rs, stake-pool\program\src\state.rs`,

Description: The implementation in `processor::process_add_validator_to_pool()` requires at least rent-exempt amount plus the `MINIMUM_ACTIVE_STAKE`. However, the `ValidatorStakeInfo` that is pushed to the validation list has the field `active_stake_lamports: 0` (`L800`).

Recommendation: Assign the value `active_stake_lamports: MINIMUM_ACTIVE_STAKE` to the new validator stake information. It may also be useful to add a comment to the `ValidatorStakeInfo` struct definition indicating the fact that the field `active_stake_lamports` does not include the rent-exempt amount of SOL.

Update: 2021-10-21: `MINIMUM_ACTIVE_STAKE` is not part of the `active_stake_lamports`. Comments are added to address this.

QSP-5 Initial values not enforced on certain stake pool fields

Severity: *Low Risk*

Status: Fixed

File(s) affected: `stake-pool\program\src\processor.rs`

Description: Stake pools must first be initialized, but not all stake pool fields have values assigned within the function `process_initialize()`. Specifically, `stake_pool.pool_token_supply`, `stake_pool.lockup`, `stake_pool.sol_deposit_fee`, and `stake_pool.sol_referral_fee` are assumed to be the default values (which may not be the case in reality).

Exploit Scenario: A malicious stake pool operator may deploy a stake pool program that has been modified with inappropriate values assigned to these fields prior to initialization (particularly `stake_pool.sol_deposit_fee` and `stake_pool.sol_referral_fee`), taking advantage of the implicit user assumption that these fields will be assigned a default values through the derived `Default` trait.

Recommendation: Default values should be assigned to every stake pool field to ensure proper initialization.

QSP-6 Contradictory information on SOL transferred while merging stake accounts

Severity: *Low Risk*

Status: Fixed

File(s) affected: `stake-pool\program\src\processor.rs, stake-pool\program\src\stake_program.rs`

Description: There is contradicting information on what happens when stake accounts are merged. The stake program code and the comment in `L108` of `stake_program.rs` indicate that all SOL from the source account, including the rent-exempt amount will be transferred to the destination account on merge. However, the implementation in `processor::process_deposit_stake()` at `L1821-L1827` (included below) seems to assume that the rent-exempt reserve may not be transferred to the destination account on merge.

```
        // If the stake account is mergeable (full-activated), `meta.rent_exempt_reserve`
        // will not be merged into `stake.delegation.stake`
        let unactivated_stake_rent = if stake.delegation.activation_epoch < clock.epoch {
            meta.rent_exempt_reserve
        } else {
            0
        };
```

Recommendation: Double check and revise the comments, documentation, or implementation to be consistent with the intended behavior.

Update: 2021-10-21: The code comments are intended and correct because when one merges an active stake into another one, all of the SOL is transferred, but it is not all added to the delegated stake.

QSP-7 Reliance on cloned `stake_program` code

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `stake-pool\program\src\instruction.rs, stake-pool\program\src\lib.rs, stake-pool\program\src\state.rs, stake-pool\program\src\processor.rs, stake-pool\program\src\stake_program.rs`

Description: The `spl-stake-pool` crate brings `stake_program` into the scope from the local crate's `stake_program.rs`, which is a cloned subset of code from the `solana-program::stake` module. If the `solana-program::stake` module gets updated within the core Solana Program, the two implementations may become out of sync, leading to incompatibilities between different stake pool programs or potentially unforeseen problems with the stake pool program in general.

Recommendation: Use `solana-program::stake` instead of the local `stake_program.rs`.

Update: 2021-10-21: The admin team stated that this will be solved later. Interested readers could read Issue-1865 for more details.

QSP-8 New staker could be zero address

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `stake-pool\program\src\processor.rs`

Description: There is a method `process_set_staker` to add a new staker, which is signed and executed by the current staker. The method requires an input of the public key of the incoming staker. The public key of the incoming staker is not validated against nonsensical or zero addresses, which means that it is entirely possible that the staker inputs an invalid public key, leading to the loss of some stake pool functions.

Recommendation: The most optimal solution would be to ensure that the new staker signs the transaction as well, otherwise, some validation against the zero address would mitigate this.

Update: 2021-10-21: The admin team stated that "set_staker can also be signed by the manager, so even if there's a 0 address, there's no way to brick the pool."

QSP-9 Unused stake pool lockup information

Severity: Informational

Status: Acknowledged

File(s) affected: `stake-pool\program\src\processor.rs`

Description: Adding a validator to the stake pool through `AddValidatorToPool` instruction will create a new stake account for this validator. However, the lockup information for this new stake account defaults to `stake_program::Lockup::default()` as opposed to the information in the `stake_pool.lockup` field for each stake pool.

Recommendation: Double check if this is intended. If `stake_program::Lockup::default()` is ultimately used, one might consider eliminating the `stake_pool.lockup` field since it does not seem to be used anywhere.

Update: 2021-10-21: The admin team stated that they will keep this field as it would be used in the future. Interested readers could read PR-1948 for more details.

QSP-10 Susceptibility to overflow

Severity: Informational

Status: Fixed

File(s) affected: `stake-pool\program\src\big_vec.rs`

Description: The following check `if skip + len > vec_len as usize` in `L93` of `big_vec.rs` may be susceptible to overflow.

Recommendation: Use `checked_add()` instead of the `+` operator.

QSP-11 Unmaintained crates are used

Severity: Informational

Status: Acknowledged

Description: The result generated from cargo-audit based on the `Cargo.lock` of the repository root shows that some of the crates used by `stake-pool` are either deprecated or unmaintained, shown as follows:

```
Crate:         failure
Version:       0.1.8
Warning:       unmaintained
Title:         failure is officially deprecated/unmaintained
Date:          2020-05-02
ID:            RUSTSEC-2020-0036
URL:           https://rustsec.org/advisories/RUSTSEC-2020-0036

Crate:         net2
Version:       0.2.37
Warning:       unmaintained
Title:         `net2` crate has been deprecated; use `socket2` instead
Date:          2020-05-01
ID:            RUSTSEC-2020-0016
URL:           https://rustsec.org/advisories/RUSTSEC-2020-0016
```

Please refer to the `cargo-audit` tool result for more details.

Recommendation: It is recommended to use maintained crates and update them to their latest version that contain patches for most of the known bugs or vulnerabilities.

Update: 2021-10-21: The admin team stated that the upstream dependencies of these crates are `solana-sdk` and `solana-client`, which the on-chain program does not depend on.

Automated Analyses

Rust Audit

```
    Fetching advisory database from `https://github.com/RustSec/advisory-db.git`
      Loaded 356 security advisories (from /root/.cargo/advisory-db)
    Updating crates.io index
    Scanning Cargo.lock for vulnerabilities (438 crate dependencies)
Crate:         failure
Version:       0.1.8
Warning:       unmaintained
Title:         failure is officially deprecated/unmaintained
Date:          2020-05-02
ID:            RUSTSEC-2020-0036
URL:           https://rustsec.org/advisories/RUSTSEC-2020-0036
Dependency tree:
failure 0.1.8
├── ed25519-dalek-bip32 0.1.1
│   └── solana-sdk 1.7.11
│       ├── test-client 0.1.0
│       ├── spl-token-swap 2.1.0
│       │   ├── test-client 0.1.0
│       │   └── spl-token-swap-fuzz 0.0.1
│       ├── spl-token-lending-cli 0.1.0
│       ├── spl-token-lending 0.1.0
│       │   └── spl-token-lending-cli 0.1.0
│       ├── spl-token-cli 2.0.14
│       ├── spl-token 3.2.0
│       │   ├── test-client 0.1.0
│       │   ├── spl-token-swap-fuzz 0.0.1
│       │   ├── spl-token-swap 2.1.0
│       │   ├── spl-token-lending-cli 0.1.0
```

```
|       |   ├── spl-token-lending 0.1.0
|       |   ├── spl-token-cli 2.0.14
|       |   ├── spl-stake-pool-cli 0.5.0
|       |   ├── spl-stake-pool 0.5.0
|       |   |   └── spl-stake-pool-cli 0.5.0
|       |   ├── spl-governance 1.1.0
|       |   ├── spl-feature-proposal 1.0.0
|       |   |   └── spl-feature-proposal-cli 1.2.0
|       |   ├── spl-binary-oracle-pair 0.1.0
|       |   ├── spl-associated-token-account 1.0.3
|       |   |   ├── spl-token-cli 2.0.14
|       |   |   └── spl-stake-pool-cli 0.5.0
|       |   └── binary-option 0.1.0
|       ├── spl-stake-pool-cli 0.5.0
|       ├── spl-stake-pool 0.5.0
|       ├── spl-shared-memory 2.0.6
|       ├── spl-record 0.1.0
|       ├── spl-name-service 0.1.1
|       ├── spl-memo 3.0.1
|       |   ├── test-client 0.1.0
|       |   └── spl-token-cli 2.0.14
|       ├── spl-math 0.1.0
|       |   ├── spl-token-swap-fuzz 0.0.1
|       |   ├── spl-token-swap 2.1.0
|       |   └── spl-stake-pool 0.5.0
|       ├── spl-governance 1.1.0
|       |   └── spl-stake-pool 0.5.0
|       ├── spl-governance 1.1.0
|       ├── spl-feature-proposal-cli 1.2.0
|       ├── spl-feature-proposal 1.0.0
|       ├── spl-example-transfer-lamports 1.0.0
|       ├── spl-example-sysvar 1.0.0
|       ├── spl-example-logging 1.0.0
|       ├── spl-example-custom-heap 1.0.0
|       ├── spl-example-cross-program-invocation 1.0.0
|       ├── spl-binary-oracle-pair 0.1.0
|       ├── spl-associated-token-account 1.0.3
|       ├── solana-vote-program 1.7.11
|       |   ├── spl-stake-pool 0.5.0
|       |   ├── solana-transaction-status 1.7.11
|       |   |   ├── spl-token-cli 2.0.14
|       |   |   ├── solana-client 1.7.11
|       |   |   |   ├── spl-token-lending-cli 0.1.0
|       |   |   |   ├── spl-token-cli 2.0.14
|       |   |   |   ├── spl-stake-pool-cli 0.5.0
|       |   |   |   ├── spl-feature-proposal-cli 1.2.0
|       |   |   |   └── solana-cli-output 1.7.11
|       |   |   |       └── spl-token-cli 2.0.14
|       |   |   └── solana-cli-output 1.7.11
|       |   ├── solana-stake-program 1.7.11
|       |   |   └── solana-runtime 1.7.11
|       |   |       ├── solana-transaction-status 1.7.11
|       |   |       ├── solana-program-test 1.7.11
|       |   |       |   ├── spl-token-lending 0.1.0
|       |   |       |   ├── spl-stake-pool 0.5.0
|       |   |       |   ├── spl-shared-memory 2.0.6
|       |   |       |   ├── spl-record 0.1.0
|       |   |       |   ├── spl-name-service 0.1.1
|       |   |       |   ├── spl-memo 3.0.1
|       |   |       |   ├── spl-math 0.1.0
|       |   |       |   ├── spl-governance 1.1.0
|       |   |       |   ├── spl-feature-proposal 1.0.0
|       |   |       |   ├── spl-example-transfer-lamports 1.0.0
|       |   |       |   ├── spl-example-sysvar 1.0.0
|       |   |       |   ├── spl-example-logging 1.0.0
|       |   |       |   ├── spl-example-custom-heap 1.0.0
|       |   |       |   ├── spl-example-cross-program-invocation 1.0.0
|       |   |       |   ├── spl-binary-oracle-pair 0.1.0
|       |   |       |   └── spl-associated-token-account 1.0.3
|       |   |       ├── solana-bpf-loader-program 1.7.11
|       |   |       |   └── solana-program-test 1.7.11
|       |   |       └── solana-banks-server 1.7.11
|       |   |           └── solana-program-test 1.7.11
|       |   ├── solana-runtime 1.7.11
|       |   ├── solana-program-test 1.7.11
|       |   ├── solana-client 1.7.11
|       |   ├── solana-cli-output 1.7.11
|       |   └── solana-account-decoder 1.7.11
|       |       ├── spl-token-cli 2.0.14
|       |       ├── spl-stake-pool-cli 0.5.0
|       |       ├── solana-transaction-status 1.7.11
|       |       ├── solana-client 1.7.11
|       |       └── solana-cli-output 1.7.11
|       ├── solana-version 1.7.11
|       |   ├── solana-net-utils 1.7.11
|       |   |   └── solana-client 1.7.11
|       |   ├── solana-faucet 1.7.11
|       |   |   └── solana-client 1.7.11
|       |   └── solana-client 1.7.11
|       ├── solana-transaction-status 1.7.11
|       ├── solana-stake-program 1.7.11
|       ├── solana-secp256k1-program 1.7.11
|       |   └── solana-runtime 1.7.11
|       ├── solana-runtime 1.7.11
|       ├── solana-remote-wallet 1.7.11
|       |   ├── spl-token-cli 2.0.14
|       |   ├── spl-stake-pool-cli 0.5.0
|       |   └── solana-clap-utils 1.7.11
|       |       ├── spl-token-lending-cli 0.1.0
|       |       ├── spl-token-cli 2.0.14
|       |       ├── spl-stake-pool-cli 0.5.0
|       |       ├── spl-feature-proposal-cli 1.2.0
|       |       ├── solana-net-utils 1.7.11
|       |       ├── solana-faucet 1.7.11
|       |       ├── solana-client 1.7.11
|       |       └── solana-cli-output 1.7.11
|       ├── solana-program-test 1.7.11
|       ├── solana-net-utils 1.7.11
|       ├── solana-metrics 1.7.11
|       |   ├── solana-vote-program 1.7.11
|       |   ├── solana-stake-program 1.7.11
|       |   ├── solana-runtime 1.7.11
|       |   ├── solana-measure 1.7.11
```

```
|       |   |   ├── solana-runtime 1.7.11
|       |   |   └── solana-bpf-loader-program 1.7.11
|       |   ├── solana-faucet 1.7.11
|       |   └── solana-banks-server 1.7.11
|       ├── solana-measure 1.7.11
|       ├── solana-faucet 1.7.11
|       ├── solana-config-program 1.7.11
|       |   ├── solana-stake-program 1.7.11
|       |   ├── solana-runtime 1.7.11
|       |   └── solana-account-decoder 1.7.11
|       ├── solana-client 1.7.11
|       ├── solana-cli-output 1.7.11
|       ├── solana-clap-utils 1.7.11
|       ├── solana-bpf-loader-program 1.7.11
|       ├── solana-banks-server 1.7.11
|       ├── solana-banks-interface 1.7.11
|       |   ├── solana-banks-server 1.7.11
|       |   └── solana-banks-client 1.7.11
|       |       └── solana-program-test 1.7.11
|       ├── solana-banks-client 1.7.11
|       └── solana-account-decoder 1.7.11
└── derivation-path 0.1.3
    ├── solana-sdk 1.7.11
    └── ed25519-dalek-bip32 0.1.1

Crate:         memmap
Version:       0.7.0
Warning:       unmaintained
Title:         memmap is unmaintained
Date:          2020-12-02
ID:            RUSTSEC-2020-0077
URL:           https://rustsec.org/advisories/RUSTSEC-2020-0077
Dependency tree:
memmap 0.7.0
└── honggfuzz 0.5.54
    └── spl-token-swap-fuzz 0.0.1

Crate:         net2
Version:       0.2.37
Warning:       unmaintained
Title:         `net2` crate has been deprecated; use `socket2` instead
Date:          2020-05-01
ID:            RUSTSEC-2020-0016
URL:           https://rustsec.org/advisories/RUSTSEC-2020-0016
Dependency tree:
net2 0.2.37
└── solana-client 1.7.11
    ├── spl-token-lending-cli 0.1.0
    ├── spl-token-cli 2.0.14
    ├── spl-stake-pool-cli 0.5.0
    ├── spl-feature-proposal-cli 1.2.0
    └── solana-cli-output 1.7.11
        └── spl-token-cli 2.0.14

warning: 3 allowed warnings found
```

The security finding that are related to the current project has been added as a finding in the finding section.

Rust-Clippy

No findings.

Adherence to Specification

1. [fixed] The comment in `StakePoolInstruction::AddValidatorToPool` indicates that at least the rent-exempt amount plus `1` lamport is needed for the stake account. However, the implementation in `processor::process_add_validator_to_pool()` requires at least the rent-exempt amount plus the `MINIMUM_ACTIVE_STAKE`.

2. [fixed] The comment in `StakePoolInstruction::RemoveValidatorFromPool` indicates that the operation will succeed if and only if the stake account has the rent-exempt amount plus `1` lamport. However, the implementation in `processor::process_remove_validator_from_pool()` requires exactly the rent-exempt amount plus the `MINIMUM_ACTIVE_STAKE` for the operation to succeed.

3. [fixed] The comment in `StakePoolInstruction::DecreaseValidatorStake` indicates that at least the rent-exempt amount plus `1` lamport needs to be moved to the transient account. However, the implementation in `processor::process_decrease_validator_stake()` only requires the rent-exempt amount to be moved to the transient account.

4. [fixed] The comment in `StakePoolInstruction::IncreaseValidatorStake` indicates that at least the rent-exempt amount plus `1` lamport needs to be moved to the transient account. However, the implementation in `processor::process_increase_validator_stake()` requires at least the rent-exempt amount plus the `MINIMUM_ACTIVE_STAKE` amount to be moved to the transient account.

5. [fixed] The comment in `StakePoolInstruction::WithdrawStake(u64)` indicates that at least the rent-exempt amount plus `1` lamport is needed in the stake account after withdrawal. However, the implementation in `processor::process_withdraw_stake()` indicates that at least the rent-exempt amount plus the `MINIMUM_ACTIVE_STAKE` amount is needed in the stake account after withdrawal.

Code Documentation

1. [fixed] `stake-pool\program\src\processor.rs`: `L168`: "Issue a stake_deactivate instruction." => "Issue a stake_delegate instruction."
2. [fixed] `stake-pool\program\src\instruction.rs`: `L305`: should be `[s]` instead of `[]`. That is, the comment for `StakePoolInstruction::SetManager` indicates the new manager account to be read-only with no signature. However, the implementation in `instruction::set_manager()` requires signature by the new manager account.
3. `stake-pool\program\src\instruction.rs`: `L331`: the position of this account mentioned in the comment is incorrect. It should be the last instead of the 2nd account given to this instruction. To be more specific, the sequence of accounts in the comments for `StakePoolInstruction::DepositSol(u64)` lists the stake pool SOL deposit authority in index `1`. However, the implementation in `instruction::deposit_sol_with_authority()` has the stake pool withdraw authority in index `1` and adds the

deposit authority to as the last element.

4. [fixed] `stake-pool\program\src\instruction.rs`: `L112`: has an incomplete comment in "Must be" .

Adherence to Best Practices

1. [fixed] There are multiple `FIXME` in `stake-pool\program\src\stake_program.rs` that are required to be fixed.

2. `stake-pool\program\src\state.rs`: `L144`: consider using "and" operator instead of "or" operator to prevent unexpected results from happening.

3. Consider adding sanity checks to the function `process_increase_validator_stake` to make sure `validator_stake_info.transient_stake_lamports == 0` before performing actual operations to the storage data.

4. [fixed] Consider a sanity check that makes sure the length of the input variable `validator_stake_accounts` can be divided by 2, to prevent unexpected conditions from happening.

5. On `L2396` in `stake-pool\program\src\processor.rs`: should send `return Err(StakePoolError::WrongStakeState.into())` instead of `return Err(StakePoolError::ValidatorNotFound.into())`.

6. [fixed] `stake-pool\program\src\processor.rs`: `L1929` and `L2094`: `if pool_tokens_user > 0` are redundant because the lines of `L1925` and `L2083` already ensured that the value would be greater than 0, especially since `pool_tokens_user` is an unsigned integer.

7. [fixed] `stake-pool\program\src\processor.rs`: `L1757` performs the verification of the `stake_program` through `*stake_program_info.key != stake_program::id()` instead of using the `check_stake_program` helper function, which breaks the consistency that the rest of the codebase follows.

8. [fixed] `stake-pool\program\src\processor.rs`: `L1767` and `L2031` contain dead code of the form `// Self::check_stake_activation(stake_info, clock, stake_history)?;`

Test Results

Test Suite Results

All tests passed.

```
running 20 tests
test big_vec::tests::find ... ok
test big_vec::tests::deserialize_mut_slice ... ok
test big_vec::tests::find_mut ... ok
test big_vec::tests::push ... ok
test big_vec::tests::retain ... ok
test stake_program::test::borsh_deserialization_live_data ... ok
test stake_program::test_id ... ok
test stake_program::test::bincode_vs_borsh ... ok
test state::test::approximate_apr_calculation ... ok
test state::test::divide_by_zero_fee ... ok
test state::test::deposit_and_withdraw ... ok
test state::test::specific_fee_calculation ... ok
test state::test::fee_calculation ... ok
test state::test::state_packing ... ok
test state::test::validator_list_active_stake ... ok
test state::test::validator_list_deserialize_mut_slice ... ok
test state::test::validator_list_iter ... ok
test state::test::zero_withdraw_calculation ... ok
test test_id ... ok
test state::test::stake_list_size_calculation ... ok

test result: ok. 20 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 7.11s

     Running tests/decrease.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/decrease-b8f7988a3634fca5)

running 8 tests
test fail_big_overdraw ... ok
test fail_decrease_twice ... ok
test fail_overdraw ... ok
test fail_with_small_lamport_amount ... ok
test fail_with_wrong_validator_list ... ok
test fail_with_unknown_validator ... ok
test fail_with_wrong_withdraw_authority ... ok
test success ... ok

test result: ok. 8 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 10.81s

     Running tests/deposit.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/deposit-849d95c2331d3cbd)

running 16 tests
test fail_with_out_of_dated_pool_balances ... ok
test fail_with_uninitialized_validator_list ... ok
test fail_with_unknown_validator ... ok
test fail_with_invalid_referrer ... ok
test fail_with_wrong_mint_for_receiver_acc ... ok
test fail_with_wrong_preferred_deposit ... ok
test fail_with_wrong_stake_program_id ... ok
test fail_with_wrong_token_program_id ... ok
test fail_with_wrong_validator_list_account ... ok
test fail_with_wrong_withdraw_authority ... ok
test fail_without_stake_deposit_authority_signature ... ok
test success ... ok
test success_with_extra_stake_lamports ... ok
test success_with_preferred_deposit ... ok
test success_with_referral_fee ... ok
test success_with_stake_deposit_authority ... ok

test result: ok. 16 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 19.59s

     Running tests/deposit_sol.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/deposit_sol-3df5a2e2559b2e14)

running 8 tests
test fail_with_invalid_referrer ... ok
```

```
test fail_with_wrong_mint_for_receiver_acc ... ok
test fail_with_wrong_token_program_id ... ok
test fail_with_wrong_withdraw_authority ... ok
test success ... ok
test fail_without_sol_deposit_authority_signature ... ok
test success_with_referral_fee ... ok
test success_with_sol_deposit_authority ... ok

test result: ok. 8 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 7.18s

     Running tests/huge_pool.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/huge_pool-b9708a27ac1064b3)

running 6 tests
test deposit_stake ... ok
test add_validator_to_pool ... ok
test remove_validator_from_pool ... ok
test set_preferred ... ok
test update ... ok
test withdraw ... ok

test result: ok. 6 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 15.21s

     Running tests/increase.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/increase-222cbc6166d2f23b)

running 7 tests
test fail_overdraw_reserve ... ok
test fail_increase_twice ... ok
test fail_with_small_lamport_amount ... ok
test fail_with_unknown_validator ... ok
test fail_with_wrong_validator_list ... ok
test fail_with_wrong_withdraw_authority ... ok
test success ... ok

test result: ok. 7 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 9.66s

     Running tests/initialize.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/initialize-502180976e6291d7)

running 18 tests
test fail_double_initialize ... ok
test fail_with_already_initialized_validator_list ... ok
test fail_with_fee_owned_by_wrong_token_program_id ... ok
test fail_with_freeze_authority ... ok
test fail_with_bad_reserve ... ok
test fail_with_high_withdrawal_fee ... ok
test fail_with_high_fee ... ok
test fail_with_not_rent_exempt_pool ... ok
test fail_with_not_rent_exempt_validator_list ... ok
test fail_with_wrong_fee_account ... ok
test fail_with_pre_minted_pool_tokens ... ok
test fail_with_wrong_max_validators ... ok
test fail_with_wrong_mint_authority ... ok
test fail_with_wrong_token_program_id ... ok
test fail_with_wrong_withdraw_authority ... ok
test fail_without_manager_signature ... ok
test success ... ok
test success_with_required_stake_deposit_authority ... ok

test result: ok. 18 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 12.35s

     Running tests/set_deposit_fee.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/set_deposit_fee-e33156d0210df9fe)

running 7 tests
test fail_sol_high_deposit_fee ... ok
test fail_sol_wrong_manager ... ok
test fail_stake_high_deposit_fee ... ok
test fail_stake_wrong_manager ... ok
test success_sol ... ok
test success_stake ... ok
test success_stake_increase_fee_from_0 ... ok

test result: ok. 7 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 5.43s

     Running tests/set_epoch_fee.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/set_epoch_fee-7d0d64d3c2b3ee36)

running 4 tests
test fail_high_fee ... ok
test fail_not_updated ... ok
test fail_wrong_manager ... ok
test success ... ok

test result: ok. 4 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 3.35s

     Running tests/set_funding_authority.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/set_funding_authority-c78e51c464f370dd)

running 5 tests
test fail_without_signature ... ok
test fail_wrong_manager ... ok
test success_set_stake_deposit_authority ... ok
test success_set_sol_deposit_authority ... ok
test success_set_withdraw_authority ... ok

test result: ok. 5 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 4.69s

     Running tests/set_manager.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/set_manager-03f7f7e18d2a661c)

running 5 tests
test test_set_manager ... ok
test test_set_manager_by_malicious ... ok
test test_set_manager_without_existing_signature ... ok
test test_set_manager_with_wrong_mint_for_pool_fee_acc ... ok
test test_set_manager_without_new_signature ... ok

test result: ok. 5 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 4.80s

     Running tests/set_preferred.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/set_preferred-e5548ed7720902db)

running 6 tests
test fail_ready_for_removal ... ok
test fail_not_present_validator ... ok
test fail_wrong_staker ... ok
test success_deposit ... ok
```

```
test success_unset ... ok
test success_withdraw ... ok

test result: ok. 6 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 5.40s

     Running tests/set_referral_fee.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/set_referral_fee-58c6dba1a6c947dc)

running 7 tests
test fail_sol_high_referral_fee ... ok
test fail_sol_wrong_manager ... ok
test fail_stake_high_referral_fee ... ok
test fail_stake_wrong_manager ... ok
test success_sol ... ok
test success_stake ... ok
test success_stake_increase_fee_from_0 ... ok

test result: ok. 7 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 5.72s

     Running tests/set_staker.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/set_staker-60bf2449a08868e4)

running 4 tests
test fail_set_staker_without_signature ... ok
test fail_wrong_manager ... ok
test success_set_staker_as_manager ... ok
test success_set_staker_as_staker ... ok

test result: ok. 4 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 3.10s

     Running tests/set_withdrawal_fee.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/set_withdrawal_fee-f1962d3e9d72b2d0)

running 10 tests
test fail_high_sol_fee_increase ... ok
test fail_high_sol_fee_increase_from_0 ... ok
test fail_high_stake_fee_increase_from_0 ... ok
test fail_high_stake_fee_increase ... ok
test fail_high_withdrawal_fee ... ok
test fail_not_updated ... ok
test fail_wrong_manager ... ok
test success ... ok
test success_fee_cannot_increase_more_than_once ... ok
test success_increase_fee_from_0 ... ok

test result: ok. 10 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 8.37s

     Running tests/update_stake_pool_balance.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/update_stake_pool_balance-
a31b5de38b303cc2)

running 7 tests
test fail_with_wrong_reserve ... ok
test fail_with_wrong_pool_fee_account ... ok
test fail_with_wrong_validator_list ... ok
test success ... ok
test test_update_stake_pool_balance_with_out_of_dated_validators_balances ... ok
test test_update_stake_pool_balance_with_uninitialized_validator_list ... ok
test success_ignoring_extra_lamports ... ok

test result: ok. 7 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 8.79s

     Running tests/update_validator_list_balance.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/update_validator_list_balance-
2aac03c3eb4edec2)

running 8 tests
test fail_with_wrong_stake_state ... ok
test fail_with_uninitialized_validator_list ... ok
test merge_into_reserve ... ok
test merge_into_validator_stake ... ok
test merge_transient_stake_after_remove ... ok
test success ... ok
test success_ignoring_hijacked_transient_stake ... ok
test success_with_burned_tokens ... ok

test result: ok. 8 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 26.31s

     Running tests/vsa_add.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/vsa_add-921aba0d5830ff0b)

running 12 tests
test fail_double_add ... ok
test fail_add_too_many_validator_stake_accounts ... ok
test fail_on_incorrectly_derived_stake_account ... ok
test fail_with_uninitialized_validator_list_account ... ok
test fail_with_unupdated_stake_pool ... ok
test fail_on_non_vote_account ... ok
test fail_with_wrong_stake_program_id ... ok
test fail_with_wrong_system_program_id ... ok
test fail_with_wrong_validator_list_account ... ok
test fail_without_signature ... ok
test fail_wrong_staker ... ok
test success ... ok

test result: ok. 12 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 8.33s

     Running tests/vsa_remove.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/vsa_remove-c2810cd8e496e129)

running 13 tests
test fail_no_signature ... ok
test fail_double_remove ... ok
test fail_not_updated_stake_pool ... ok
test fail_not_at_minimum ... ok
test fail_with_uninitialized_validator_list_account ... ok
test fail_with_activating_transient_stake ... ok
test fail_with_wrong_stake_program_id ... ok
test fail_with_wrong_validator_list_account ... ok
test fail_wrong_staker ... ok
test success ... ok
test success_resets_preferred_validator ... ok
test success_with_deactivating_transient_stake ... ok
test success_with_hijacked_transient_account ... ok

test result: ok. 13 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 12.67s

     Running tests/withdraw.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/withdraw-7c2e071a827dcef4)
```

```
running 16 tests
test fail_double_withdraw_to_the_same_account ... ok
test fail_overdraw_validator ... ok
test fail_with_low_delegation ... ok
test fail_with_unknown_validator ... ok
test fail_with_wrong_stake_program ... ok
test fail_with_wrong_preferred_withdraw ... ok
test fail_with_wrong_token_program_id ... ok
test fail_with_wrong_validator_list ... ok
test fail_with_wrong_withdraw_authority ... ok
test fail_without_token_approval ... ok
test success ... ok
test success_with_closed_manager_fee_account ... ok
test success_with_preferred_validator ... ok
test success_with_reserve ... ok
test success_withdraw_all_fee_tokens ... ok
test success_withdraw_from_transient ... ok

test result: ok. 16 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 28.62s

     Running tests/withdraw_sol.rs (/tmp/20211021/solana-program-library-3b48fa09d38d1b66ffb4fef186b606f1bc4fdb31/target/debug/deps/withdraw_sol-9cd93ee14df86c0d)

running 5 tests
test fail_with_wrong_withdraw_authority ... ok
test fail_overdraw_reserve ... ok
test fail_without_sol_withdraw_authority_signature ... ok
test success ... ok
test success_with_sol_withdraw_authority ... ok

test result: ok. 5 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 5.57s

   Doc-tests spl-stake-pool

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

Code Coverage

The coverage score could not be obtained for Solana programs at the current moment.

Appendix

File Signatures

The following are the SHA-256 hashes of the reviewed files. A file with a different SHA-256 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different SHA-256 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review.

Contracts

```
2e0a5f61fd7397bfe99f1cdf6fa271df1be2f2fbf86f007853b94e1a72997dce ./src/big_vec.rs

a89c683e01ecfe8ceec214fef6a2aa0e591ea7c4efc4e5dd71d9fb928f4e41ff ./src/entrypoint.rs

cf44da3ab9faa77198193ebf5c958d64bb95e146ab0f237e886793a801fa98b8 ./src/error.rs

2fd2fc9a6bca1a5d99887e428e42ae5affb93ee7b5cba9e2de6f1ec2b9e5b1a7 ./src/instruction.rs

7fed30d12ac8b3abfa1884cc24197ff78e16884d22932a9b2c34c705927c88c7 ./src/lib.rs

1ecfcf784c1a96611afa1c52ff73412b25c5ab63f392bfe64811650ccfba3b89 ./src/processor.rs

305d80417a631b5b8fd080f4b97aad2d161ab4d218c08736cf6941dc9a71d864 ./src/stake_program.rs

b23b3a48b695bbd0f3bb90d1e46cac084162f0a3749a822a63b374ba991f5f54 ./src/state.rs
```

Tests

```
2347b0904fc52e5365ed2ca66d5e00ac275bc39281b4bbc275a882755e40359a ./tests/decrease.rs

ea3baac0267cdf1ea717fcf2b615f83e151c45ccf7e813ddb7cba4d0fc467f58 ./tests/deposit.rs

cc177e4e9d0031b8050fb63fe2f605577b5e0f2377442cec6b9fa93dff5d00e6 ./tests/deposit_sol.rs

f87c30b94205cea54f9fa0176888f713ad681ff18b77193abacec2650b8bb1ea ./tests/huge_pool.rs

0d67dcec36a0d27bd3aac14795bf016cb4c3672d0311d4f991b518df2b3f448b ./tests/increase.rs

903c5c6b04333f86dcd2c1ee3d8378723c55c289776a4afc1c5702a4b317a39a ./tests/initialize.rs

8a007d95a3a4f9aadefbdfd202901f3aa582a3c7a91b9d7bbe0f7120b6f72863 ./tests/set_deposit_fee.rs

1bd50f75cbb5ff13c40be7d49328e561e7a2028b2e14cd5096e6635b0ef0f259 ./tests/set_epoch_fee.rs

2351582c032e00321d8f954c61fc12f61a7d5fe13f40a6f7cfa74f0035929ce2 ./tests/set_funding_authority.rs

3ba0b562b58491018dccd52e4c6ad8559845b788b4566e14d2c9cf586f08526c ./tests/set_manager.rs

0eb75c04ded7bb99093e8d4a07f3c98ffe6abb8a2aafe83f18d3f9e406476cfc ./tests/set_preferred.rs

db94cc143f8e932921721d769a68148bb23bb143e65633d2e3af9573c7250c8f ./tests/set_referral_fee.rs

db6b32e83ad6fbadd0aad6c9587cc6c6036ac404be0a112e2fdeab7e07bda205 ./tests/set_staker.rs

68fc2aef0f9737c8ac4f15d06db07767259c385c84859d219fe5e265992a9e8d ./tests/set_withdrawal_fee.rs

d5eaea3b297b4f676671cf18efa04179c74578713453234a739c1364948ab398 ./tests/update_stake_pool_balance.rs

94fafd3d23ba7fdfd256fe4674a23875a5d7f3631e0662f951a976836be27b51 ./tests/update_validator_list_balance.rs

8b39b3b5ba8327ef835783876aab0fd7972e4cd35c7df046100d4fa5b4b5f631 ./tests/vsa_add.rs

30146e70dbe0aaa3bf29b59706647c506bf5dea633db142a37b83b3a22700096 ./tests/vsa_remove.rs

27fc49902d34667a5e43e7df9f8cdd4a5f263b27e27836951f155bf14e6f56e5 ./tests/withdraw.rs

d346bbec23d90c5f25bc1a2a80956bf3060ecdd61e04f61ff37789f44143c9a4 ./tests/withdraw_sol.rs

3d7f991f83b50db046798557c30306bdcc4c4d9dbdc2d249085e65247048ebc3 ./tests/helpers/mod.rs
```

Changelog

- 2021-10-15 - Initial report
- 2021-10-22 - Final report

About Quantstamp

Quantstamp is a Y Combinator-backed company that helps to secure blockchain platforms at scale using computer-aided reasoning tools, with a mission to help boost the adoption of this exponentially growing technology.

With over 1000 Google scholar citations and numerous published papers, Quantstamp's team has decades of combined experience in formal verification, static analysis, and software verification. Quantstamp has also developed a protocol to help smart contract developers and projects worldwide to perform cost-effective smart contract security scans.

To date, Quantstamp has protected $5B in digital asset risk from hackers and assisted dozens of blockchain projects globally through its white glove security assessment services. As an evangelist of the blockchain ecosystem, Quantstamp assists core infrastructure projects and leading community initiatives such as the Ethereum Community Fund to expedite the adoption of blockchain technology.

Quantstamp's collaborations with leading academic institutions such as the National University of Singapore and MIT (Massachusetts Institute of Technology) reflect our commitment to research, development, and enabling world-class blockchain security.