Solana Labs - SPL Stake Pool

Solana Program Security Audit

Prepared by: **Halborn**

Date of Engagement: **December 19th, 2022 - January 25th, 2023**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	01/24/2022	Michael Smith
0.2	Document Updates	01/25/2023	Michael Smith
0.3	Document Updates	01/25/2023	Michael Smith
0.4	Draft Review	01/25/2023	Isabel Burruezo
0.5	Draft Review	01/25/2023	Piotr Cielas
0.6	Draft Review	01/25/2023	Gabi Urrutia
1.0	Remediation Plan	02/08/2023	Michael Smith
1.1	Remediation Plan Review	02/09/2023	Isabel Burruezo
1.2	Remediation Plan Review	02/09/2023	Piotr Cielas
1.3	Remediation Plan Review	02/09/2023	Gabi Urrutia

CONTACTS

CONTACT	COMPANY	EMAIL
Rob Behnke	Halborn	Rob.Behnke@halborn.com
Steven Walbroehl	Halborn	Steven.Walbroehl@halborn.com
Gabi Urrutia	Halborn	Gabi.Urrutia@halborn.com
Piotr Cielas	Halborn	Piotr.Cielas@halborn.com
Isabel Burruezo	Halborn	Isabel.Burruezo@halborn.com
Michael Smith	Halborn	michael.smith@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Solana Labs engaged Halborn to conduct a security audit on their Solana programs, beginning on December 19th, 2022 and ending on January 25th, 2023 . The security assessment was scoped to the programs provided in the Stake Pool GitHub repository. Commit hashes and further details can be found in the Scope section of this report.

The SPL Stake Pool program lets SOL holders deposit SOL into a pool where it can be staked by a designated Staker across multiple validators. This allows the depositors to earn staking rewards without having to manage stakes.

1.2 AUDIT SUMMARY

The team at Halborn was provided five weeks for the engagement and assigned one full-time security engineer to audit the security of the programs in scope. The security engineer is a blockchain and Solana program security expert with advanced penetration testing and Solana program hacking skills, and deep knowledge of multiple blockchain protocols.

The purpose of this audit is to:

- Ensure that Solana program functions operate as intended
- Identify potential security issues within the programs

In summary, Halborn identified some improvements to reduce the likelihood and impact of multiple risks, which has been mostly addressed by Solana Labs . The main ones are the following:

- Fees can be updated before new epoch

Solana Labs solved this finding, new fee’s will take effect two epochs after they are changed.

- Possible rust panics due to unsafe unwrap usage

Solana Labs solved this finding, unnecessary unwraps were removed to optimize compute usage.

- Missing cargo overflow checks

Solana Labs acknowledged this finding, most of the crate uses clippy to prevent developers from using integer arithmetic. In stake-pool/program/src/big_vec.rs checked math isn't used as it introduces too much compute usage.

1.3 TEST APPROACH & METHODOLOGY

Halborn performed a combination of a manual review of the source code and automated security testing to balance efficiency, timeliness, practicality, and accuracy in regard to the scope of the program audit. While manual testing is recommended to uncover flaws in business logic, processes, and implementation; automated testing techniques help enhance coverage of programs and can quickly identify items that do not follow security best practices.

The following phases and associated tools were used throughout the term of the audit:

- Research into the architecture, purpose, and use of the platform.
- Manual program source code review to identify business logic issues.
- Mapping out possible attack vectors
- Thorough assessment of safety and usage of critical Rust variables and functions in scope that could lead to arithmetic vulnerabilities.
- Finding unsafe Rust code usage (cargo-geiger)

- Scanning dependencies for known vulnerabilities (cargo audit).
- Local runtime testing (solana-test-framework)
- Scanning for common Solana vulnerabilities (soteria)

RISK METHODOLOGY:

Vulnerabilities or issues observed by Halborn are ranked based on the risk assessment methodology by measuring the **LIKELIHOOD** of a security incident and the **IMPACT** should an incident occur. This framework works for communicating the characteristics and impacts of technology vulnerabilities. The quantitative model ensures repeatable and accurate measurement while enabling users to see the underlying vulnerability characteristics that were used to generate the Risk scores. For every vulnerability, a risk level will be calculated on a scale of 5 to 1 with 5 being the highest likelihood or impact.

RISK SCALE - LIKELIHOOD

5 - Almost certain an incident will occur.
4 - High probability of an incident occurring.
3 - Potential of a security incident in the long term.
2 - Low probability of an incident occurring.
1 - Very unlikely issue will cause an incident.

RISK SCALE - IMPACT

5 - May cause devastating and unrecoverable impact or loss.
4 - May cause a significant level of impact or loss.
3 - May cause a partial impact or loss to many.
2 - May cause temporary impact or loss.
1 - May cause minimal or un-noticeable impact.

The risk level is then calculated using a sum of these two values, creating a value of 10 to 1 with 10 being the highest level of security risk.

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL

10 - CRITICAL

9 - 8 - HIGH

7 - 6 - MEDIUM

5 - 4 - LOW

3 - 1 - VERY LOW AND INFORMATIONAL

1.4 SCOPE

Code repositories:

1. Project Name

- Repository: XYZ
- Commit ID: eba709b9317f8c7b8b197045161cb744241f0bff
- Programs in scope:

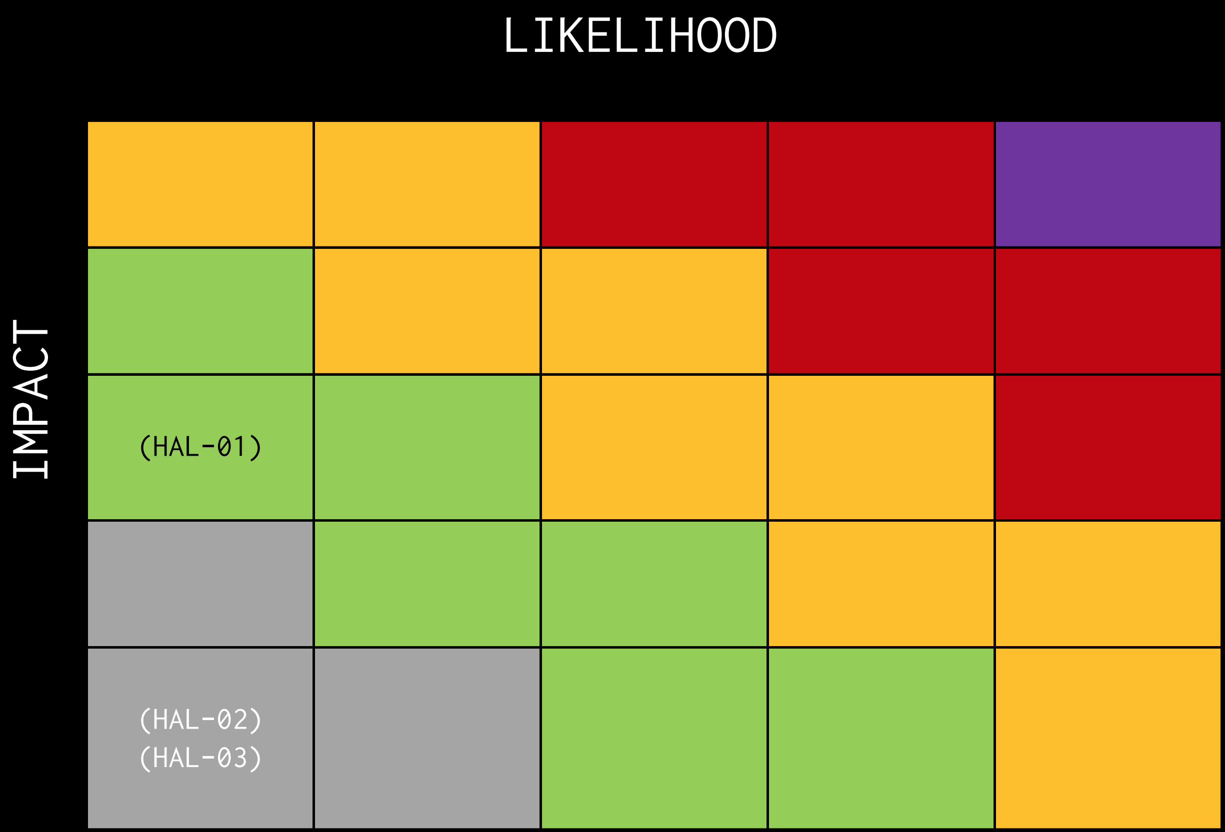
    1. (/stake-pool/program)

Out-of-scope:

- third-party libraries and dependencies
- financial-related attacks

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	0	0	1	2



SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL01 - FEES CAN BE UPDATED BEFORE NEW EPOCH	Low	SOLVED - 01/26/2023
HAL02 - POSSIBLE RUST PANICS DUE TO UNSAFE UNWRAP USAGE	Informational	SOLVED - 01/27/2023
HAL03 - MISSING CARGO OVERFLOW CHECKS	Informational	ACKNOWLEDGED

FINDINGS & TECH DETAILS

3.1 (HAL-01) FEES CAN BE UPDATED BEFORE NEW EPOCH - LOW

Description:

Managers charge users various fees for managing the stake pool, such as an epoch_fee charged for earned staking rewards, stake_withdrawal and sol_withdrawal when users withdraw their stake/SOL from the pool. The stake-pool program prevents malicious managers from rug pulling users and increasing fee's during an epoch by ensuring any fee changes take effect on the next epoch. A clever manager can wait until the last slot in the epoch and then raise fees.

This leaves unsuspecting users no time to withdraw funds before new fees are applied to the stake pool.

- epoch_fee is the most impacted by this as it can be raised to 100%, allowing the manager to take all new staking rewards.
- stake_withdrawal and sol_withdrawal is impacted to a lesser extent as the program limits how much a manager can increase new fee's compared to old fee's. In a worst-case scenario, the withdrawal fee can be larger than the return the user received from participating in the stake pool, causing the user to lose funds from their original deposit.

Code Location:

Listing 1: stake-pool/program/src/processor.rs (Line 2445)

```
2445 fn process_update_stake_pool_balance(
2446     program_id: &Pubkey,
2447     accounts: &[AccountInfo],
2448 ) -> ProgramResult {
2449     let account_info_iter = &mut accounts.iter();
2450     let stake_pool_info = next_account_info(account_info_iter)?;
2451     let withdraw_info = next_account_info(account_info_iter)?;
2452     let validator_list_info = next_account_info(account_info_iter)
```

```
↳ ?;
    let reserve_stake_info = next_account_info(account_info_iter)
↳ ?;
    let manager_fee_info = next_account_info(account_info_iter)?;
    let pool_mint_info = next_account_info(account_info_iter)?;
    let token_program_info = next_account_info(account_info_iter)
↳ ?;
    let clock = Clock::get()?;

    check_account_owner(stake_pool_info, program_id)?;
    let mut stake_pool = try_from_slice_unchecked::<StakePool>(&
↳ stake_pool_info.data.borrow())?;
    if !stake_pool.is_valid() {
        return Err(StakePoolError::InvalidState.into());
    }
    stake_pool.check_mint(pool_mint_info)?;
    stake_pool.check_authority_withdraw(withdraw_info.key,
↳ program_id, stake_pool_info.key)?;
    stake_pool.check_reserve_stake(reserve_stake_info)?;
    if stake_pool.manager_fee_account != *manager_fee_info.key {
        return Err(StakePoolError::InvalidFeeAccount.into());
    }

    if *validator_list_info.key != stake_pool.validator_list {
        return Err(StakePoolError::InvalidValidatorStakeList.into
↳ ());
    }
    if stake_pool.token_program_id != *token_program_info.key {
        return Err(ProgramError::IncorrectProgramId);
    }

    check_account_owner(validator_list_info, program_id)?;
    let mut validator_list_data = validator_list_info.data.
↳ borrow_mut();
    let (header, validator_list) =
        ValidatorListHeader::deserialize_vec(&mut
↳ validator_list_data)?;
    if !header.is_valid() {
        return Err(StakePoolError::InvalidState.into());
    }

    let previous_lamports = stake_pool.total_lamports;
    let previous_pool_token_supply = stake_pool.pool_token_supply;
    let reserve_stake = try_from_slice_unchecked::<stake::state::
```

```
StakeState>(
        &reserve_stake_info.data.borrow(),
    )?;
    let mut total_lamports = if let stake::state::StakeState::Initialized(meta) = reserve_stake
    {
        reserve_stake_info
            .lamports()
            .checked_sub(minimum_reserve_lamports(&meta))
            .ok_or(StakePoolError::CalculationFailure)?
    } else {
        msg!("Reserve stake account in unknown state, aborting");
        return Err(StakePoolError::WrongStakeState.into());
    };
    for validator_stake_record in validator_list.iter::<ValidatorStakeInfo>() {
        if validator_stake_record.last_update_epoch < clock.epoch {
            return Err(StakePoolError::StakeListOutOfDate.into());
        }
        total_lamports = total_lamports
            .checked_add(validator_stake_record.stake_lamports())
            .ok_or(StakePoolError::CalculationFailure)?;
    }

    msg!("total_lamports {}", total_lamports);
    msg!("previous_lamports {}", previous_lamports);

    let reward_lamports = total_lamports.saturating_sub(previous_lamports);
    msg!("reward_lamports {}", reward_lamports);

    // If the manager fee info is invalid, they don't deserve to receive the fee.
    let fee = if stake_pool.check_manager_fee_info(manager_fee_info).is_ok() {
        msg!("yes fee");
        stake_pool
            .calc_epoch_fee_amount(reward_lamports)
            .ok_or(StakePoolError::CalculationFailure)?
    } else {
        msg!("no fee");
        0
    };
```

```
    msg!("fee is: {:?}", fee);

    if fee > 0 {
        Self::token_mint_to(
            stake_pool_info.key,
            token_program_info.clone(),
            pool_mint_info.clone(),
            manager_fee_info.clone(),
            withdraw_info.clone(),
            AUTHORITY_WITHDRAW,
            stake_pool.stake_withdraw_bump_seed,
            fee,
        )?;
    }

    if stake_pool.last_update_epoch < clock.epoch {
        if let Some(fee) = stake_pool.next_epoch_fee {
            stake_pool.epoch_fee = fee;
            stake_pool.next_epoch_fee = None;
        }
        if let Some(fee) = stake_pool.next_stake_withdrawal_fee {
            stake_pool.stake_withdrawal_fee = fee;
            stake_pool.next_stake_withdrawal_fee = None;
        }
        if let Some(fee) = stake_pool.next_sol_withdrawal_fee {
            stake_pool.sol_withdrawal_fee = fee;
            stake_pool.next_sol_withdrawal_fee = None;
        }
        stake_pool.last_update_epoch = clock.epoch;
        stake_pool.last_epoch_total_lamports = previous_lamports;
        stake_pool.last_epoch_pool_token_supply = previous_pool_token_supply;
    }
    stake_pool.total_lamports = total_lamports;
```

Listing 2: stake-pool/program/src/processor.rs (Line 3649)

```
fn process_set_fee(
        program_id: &Pubkey,
        accounts: &[AccountInfo],
        fee: FeeType,
    ) -> ProgramResult {
        let account_info_iter = &mut accounts.iter();
```

```
        let stake_pool_info = next_account_info(account_info_iter)?;
        let manager_info = next_account_info(account_info_iter)?;
        let clock = Clock::get()?;

        check_account_owner(stake_pool_info, program_id)?;
        let mut stake_pool = try_from_slice_unchecked::<StakePool>(&stake_pool_info.data.borrow())?;
        if !stake_pool.is_valid() {
            return Err(StakePoolError::InvalidState.into());
        }
        stake_pool.check_manager(manager_info)?;

        if fee.can_only_change_next_epoch() && stake_pool.last_update_epoch < clock.epoch {
            return Err(StakePoolError::StakeListAndPoolOutOfDate.into());
        }

        fee.check_too_high()?;
        stake_pool.update_fee(&fee)?;
        stake_pool.serialize(&mut *stake_pool_info.data.borrow_mut())?;
        Ok(())
    }
```

Listing 3: stake-pool/program/src/state.rs (Line 826)

```
pub fn check_withdrawal(&self, old_withdrawal_fee: &Fee) -> Result<(), StakePoolError> {
        // If the previous withdrawal fee was 0, we allow the fee to be set to a
        // maximum of (WITHDRAWAL_BASELINE_FEE * MAX_WITHDRAWAL_FEE_INCREASE)
        let (old_num, old_denom) =
            if old_withdrawal_fee.denominator == 0 || old_withdrawal_fee.numerator == 0 {
                (
                    WITHDRAWAL_BASELINE_FEE.numerator,
                    WITHDRAWAL_BASELINE_FEE.denominator,
                )
            } else {
                (old_withdrawal_fee.numerator, old_withdrawal_fee.denominator)
```

```
            };

        // Check that new_fee / old_fee <= MAX_WITHDRAWAL_FEE_INCREASE
        // Program fails if provided numerator or denominator is too large, resulting in overflow
        if (old_num as u128)
            .checked_mul(self.denominator as u128)
            .map(|x| x.checked_mul(MAX_WITHDRAWAL_FEE_INCREASE.numerator as u128))
            .ok_or(StakePoolError::CalculationFailure)?
            < (self.numerator as u128)
                .checked_mul(old_denom as u128)
                .map(|x| x.checked_mul(MAX_WITHDRAWAL_FEE_INCREASE.denominator as u128))
                .ok_or(StakePoolError::CalculationFailure)?
        {
            msg!(
                "Fee increase exceeds maximum allowed, proposed increase factor ({} / {})",
                self.numerator.saturating_mul(old_denom),
                old_num.saturating_mul(self.denominator),
            );
            return Err(StakePoolError::FeeIncreaseTooHigh);
        }
        Ok(())
    }
```

Risk Level:

Likelihood - 1
Impact - 3

Recommendation:

Similar to issue 29346 fee updates should be limited to the first half of an epoch or increase the delay to two epochs, so users get at least one epoch to withdraw their funds.

Remediation Plan:

SOLVED The Solana Labs team resolved this issue in commit:
- dc7e16cdae85defd53189d3882fef43fdd25df05

Going forward, new fee’s will take effect two epochs after they are changed.

3.2 (HAL-02) POSSIBLE RUST PANICS DUE TO UNSAFE UNWRAP USAGE - INFORMATIONAL

Description:

The use of helper methods in Rust, such as unwrap, is allowed in dev and testing environment because those methods are supposed to throw an error (also known as panic!) when called on Option::None or a Result which is not Ok. However, keeping unwrap functions in production environment is considered bad practice because they may lead to program crashes, which are usually accompanied by insufficient or misleading error messages.

Code Location:

Note: only unwraps introduced by the changes in scope are listed, justified usages such as in tests were excluded.

Listing 4

```
./stake-pool/program/src/lib.rs:120   seed.as_ref().map(|s| s.
↳ as_slice()).unwrap_or(&[]),
./stake-pool/program/src/processor.rs:1145  let mut
↳ validator_stake_info = maybe_validator_stake_info.unwrap();
./stake-pool/program/src/processor.rs:1319   let mut
↳ validator_stake_info = maybe_validator_stake_info.unwrap();
./stake-pool/program/src/processor.rs:1428   let
↳ stake_history_info = maybe_stake_history_info.unwrap();
./stake-pool/program/src/processor.rs:1576   let mut
↳ validator_stake_info = maybe_validator_stake_info.unwrap();
./stake-pool/program/src/processor.rs:1897   let mut
↳ validator_stake_info = maybe_validator_stake_info.unwrap();
./stake-pool/program/src/processor.rs:2000   let mut
↳ validator_stake_info = maybe_validator_stake_info.unwrap();
./stake-pool/program/src/processor.rs:2212, 2213   let
↳ validator_stake_info = validator_stakes.first().unwrap(); let
↳ transient_stake_info = validator_stakes.last().unwrap();
./stake-pool/program/src/state.rs:667: .unwrap()
```

```
./stake-pool/program/src/state.rs:683: u64::try_from_slice(&data
↳ [0..8]).unwrap() > *lamports
./stake-pool/program/src/state.rs:689: u64::try_from_slice(&data
↳ [8..16]).unwrap() > *lamports
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended not to use the unwrap function in the production environment because its use causes panic! and may crash any affected module, program or in the worst case the runtime without verbose error messages. Crashing the system will result in a loss of availability and, in some cases, even private information stored in the state. Some alternatives are possible, such as propagating the error with ? instead of unwrapping, or using the error-chain crate for errors.

Remediation Plan:

SOLVED The Solana Labs team resolved this issue in commits:
- bd7a2cbb2338095e011388ae890400db30cd4aef
- 254c087861e6ba94cbe53197088b92a8f4539df7

Unnecessary unwraps were removed to optimize compute usage.

3.3 (HAL-03) MISSING CARGO OVERFLOW CHECKS - INFORMATIONAL

Description:

It was observed that there is no overflow-checks=true in Cargo.toml. By default, overflow checks are disabled in optimized release builds. Hence, if there is an overflow on release builds, it will be silenced, leading to unexpected behavior of an application. Even if checked arithmetic is used through checked_*, it is recommended to have that check in Cargo.toml.

Code Location:

- stake-pool/Cargo.toml

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to add overflow-checks=true under your release profile in Cargo.toml.

Remediation Plan:

ACKNOWLEDGED: The Solana Labs Team acknowledged this finding, most of the crate uses clippy to prevent developers from using integer arithmetic. In stake-pool/program/src/big_vec.rs checked math is not used as it introduces too much compute usage.

MANUAL TESTING

In the manual testing phase, the following scenarios were simulated. The scenarios listed below were selected based on the severity of the vulnerabilities Halborn was testing the program for.

4.1 SET DEACTIVATED VALIDATOR AS PREFERRED VALIDATOR

Description:

Tests were done to:

1) Set a deactivated validator to the preferred_deposit_validator_vote_address or preferred_withdraw_validator_vote_address
2) Deactivate an preferred_deposit_validator_vote_address or preferred_withdraw_validator_vote_address

Results:

The program prevents both scenarios, we believe successfully doing either of the above would result in a denial of service or in a worst-case scenario a loss of funds.

MANUAL TESTING

4.2 DENIAL OF SERVICE

Description:

Prior to commit 9aac29c250f1f5408f112720e0ff0c89fe7bf9c8 at the start of an epoch a malicious user can deposit the minimum stake delegation to a validator and then increase the stake on the validator. This prevented SOL deposits from being activated, tests were done to ensure that increase_additional_validator_stake and decrease_additional_valdator-stake resolve this issue. This was tested in several scenarios to ensure users could always receive their funds either from SOL or Stake.

Results:

All deposits and withdrawals were successful as per the order of priority in the stake pool operation manual

4.3 INCREASE AND DECREASE VALIDATOR STAKE

Description:

The stake-pool operation manual states that it is impossible to increase and decrease a validators stake until an update instruction is performed. With the recent addition of the increase_additional_validator_stake and decrease_additional_valdator-stake tests were done to try to bypass this limitation using the new instructions.

Results:

We were unable to bypass the restriction and were required to update the validator_list_balance and stake_pool_balance before decreasing the validator stake.

4.4 REGRESSION TESTING

Recently reported Critical and High vulnerabilities were reviewed and re-tested to confirm that no new vulnerabilities were introduced and that the previous fixes resolved the issue.

Results:

No code vulnerabilities were identified.

AUTOMATED TESTING

5.1 AUTOMATED VULNERABILITY SCANNING

Description:

Halborn used automated security scanners to assist with the detection of well-known security issues, and to identify low-hanging fruits on the targets for this engagement. Among the tools used was Soteria, a security analysis service for Solana programs. Soteria performed a scan on all the programs in scope and sent the compiled results to analyzers to locate well-known vulnerabilities.

Results:

```
=============This account may be UNTRUSTFUL!================
Found a potential vulnerability at line 712, column 34 in src/processor.rs
The account info is not trustful:

 706|        let account_info_iter = &mut accounts.iter();
 707|        let stake_pool_info = next_account_info(account_info_iter)?;
 708|        let manager_info = next_account_info(account_info_iter)?;
 709|        let staker_info = next_account_info(account_info_iter)?;
 710|        let withdraw_authority_info = next_account_info(account_info_iter)?;
 711|        let validator_list_info = next_account_info(account_info_iter)?;
>712|        let reserve_stake_info = next_account_info(account_info_iter)?;
 713|        let pool_mint_info = next_account_info(account_info_iter)?;
 714|        let manager_fee_info = next_account_info(account_info_iter)?;
 715|        let token_program_info = next_account_info(account_info_iter)?;
 716|
 717|        let rent = Rent::get()?;
 718|
>>>Stack Trace:
>>>spl_stake_pool::processor::Processor::process::h9fa102d36ee60ceb [src/entrypoint.rs:19]
>>>   spl_stake_pool::processor::Processor::process_initialize::ha25dd3c653c05abb [src/processor.rs:3737]

 - ✔ [00m:00s] Building Static Happens-Before Graph
 - ✔ [00m:00s] Detecting Vulnerabilities
detected 3 untrustful accounts in total.
detected 7 unsafe math operations in total.

--------The summary of potential vulnerabilities in spl_stake_pool.ll--------

         3 untrustful account issues
         7 unsafe arithmetic issues
```

5.2 AUTOMATED ANALYSIS

Description:

Halborn used automated security scanners to assist with the detection of well-known security issues and vulnerabilities. Among the tools used was cargo-audit, a security scanner for vulnerabilities reported to the Rust-Sec Advisory Database. All vulnerabilities published in https://crates.io are stored in a repository named The RustSec Advisory Database. cargo audit is a human-readable version of the advisory database which performs a scanning on Cargo.lock. Security Detections are only in scope. All vulnerabilities shown here were already disclosed in the above report. However, to better assist the developers maintaining this code, the auditors are including the output with the dependencies tree, and this is included in the cargo audit output to better know the dependencies affected by unmaintained and vulnerable crates.

Results:

ID	package	Short Description
RUSTSEC-2020-0071	time	Potential segfault in the time crate
RUSTSEC-2021-0139	ansi term	Unmaintained
RUSTSEC-2020-0016	net2	Unmaintained

5.3 UNSAFE RUST CODE DETECTION

Description:

Halborn used automated security scanners to assist with the detection of well-known security issues and vulnerabilities. Among the tools used was cargo-geiger, a security tool that lists statistics related to the usage of unsafe Rust code in a core Rust codebase and all its dependencies.

Results:

AUTOMATED TESTING

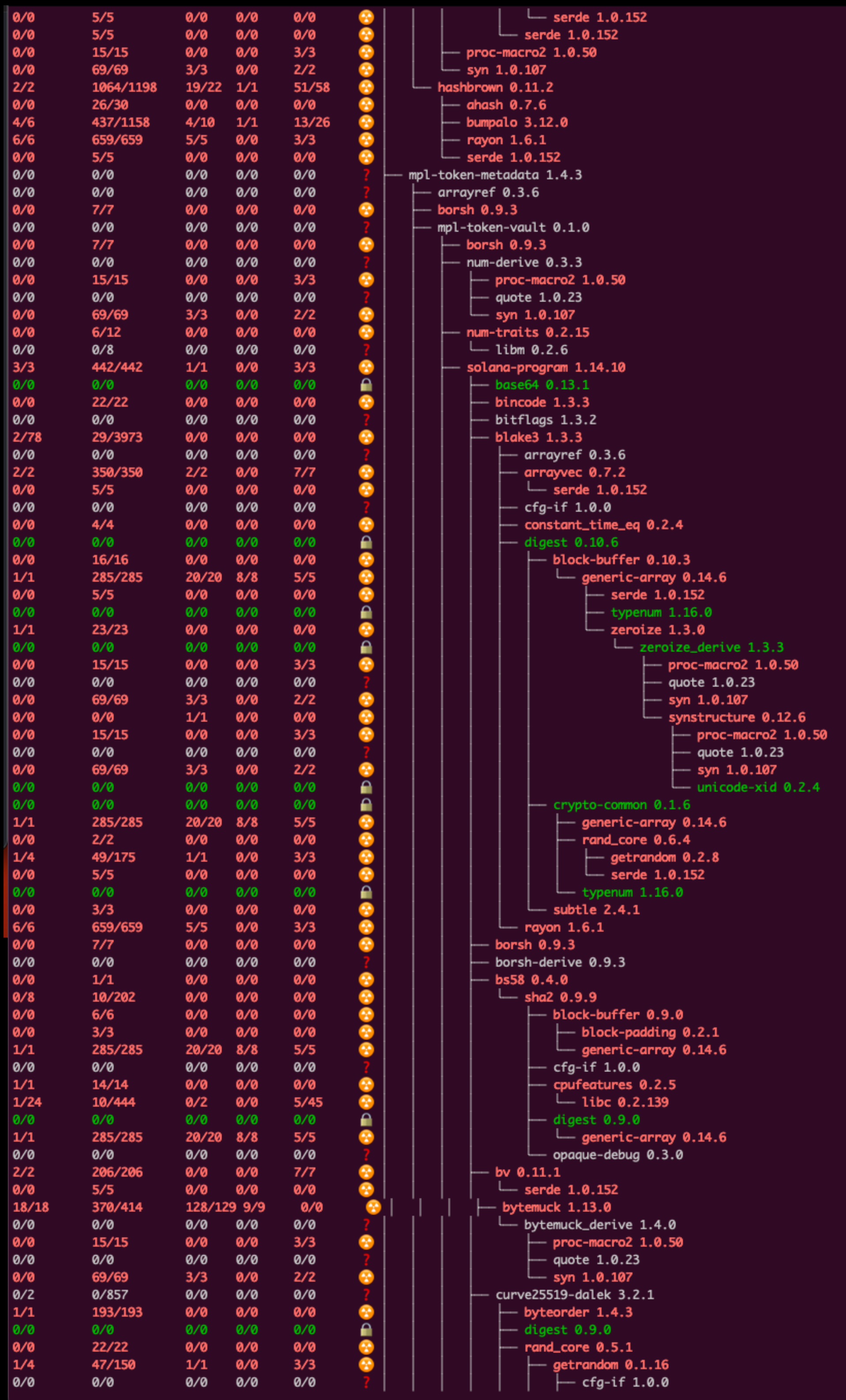
```
Metric output format: x/y
    x = unsafe code used by the build
    y = total unsafe code found in the crate

Symbols:
    🔒  = No `unsafe` usage found, declares #![forbid(unsafe_code)]
    ?   = No `unsafe` usage found, missing #![forbid(unsafe_code)]
    ☢   = `unsafe` usage found

Functions  Expressions  Impls  Traits  Methods  Dependency

0/0        61/61        0/0    0/0     0/0      ☢  spl-stake-pool 0.7.0
0/0        0/0          0/0    0/0     0/0      ?  ├── arrayref 0.3.6
0/0        22/22        0/0    0/0     0/0      ☢  ├── bincode 1.3.3
0/0        5/5          0/0    0/0     0/0      ☢  │   └── serde 1.0.152
0/0        0/0          0/0    0/0     0/0      ?  │       └── serde_derive 1.0.152
0/0        15/15        0/0    0/0     3/3      ☢  │           ├── proc-macro2 1.0.50
0/0        4/4          0/0    0/0     0/0      ☢  │           │   └── unicode-ident 1.0.6
0/0        0/0          0/0    0/0     0/0      ?  │           ├── quote 1.0.23
0/0        15/15        0/0    0/0     3/3      ☢  │           │   └── proc-macro2 1.0.50
0/0        69/69        3/3    0/0     2/2      ☢  │           └── syn 1.0.107
0/0        15/15        0/0    0/0     3/3      ☢  │               ├── proc-macro2 1.0.50
0/0        0/0          0/0    0/0     0/0      ?  │               ├── quote 1.0.23
0/0        4/4          0/0    0/0     0/0      ☢  │               └── unicode-ident 1.0.6
0/0        7/7          0/0    0/0     0/0      ☢  ├── borsh 0.9.3
0/0        0/0          0/0    0/0     0/0      ?  │   ├── borsh-derive 0.9.3
0/0        0/0          0/0    0/0     0/0      ?  │   │   ├── borsh-derive-internal 0.9.3
0/0        15/15        0/0    0/0     3/3      ☢  │   │   │   ├── proc-macro2 1.0.50
0/0        0/0          0/0    0/0     0/0      ?  │   │   │   ├── quote 1.0.23
0/0        69/69        3/3    0/0     2/2      ☢  │   │   │   └── syn 1.0.107
0/0        0/0          0/0    0/0     0/0      ?  │   │   ├── borsh-schema-derive-internal 0.9.3
0/0        15/15        0/0    0/0     3/3      ☢  │   │   │   ├── proc-macro2 1.0.50
0/0        0/0          0/0    0/0     0/0      ?  │   │   │   ├── quote 1.0.23
0/0        69/69        3/3    0/0     2/2      ☢  │   │   │   └── syn 1.0.107
0/0        0/0          0/0    0/0     0/0      ?  │   │   ├── proc-macro-crate 0.1.5
0/0        0/0          0/0    0/0     0/0      🔒 │   │   │   └── toml 0.5.11
0/0        41/46        1/1    0/0     0/0      ☢  │   │   │       ├── indexmap 1.9.2
1/1        1223/1367    21/24  1/1     62/69    ☢  │   │   │       │   ├── hashbrown 0.12.3
0/0        26/30        0/0    0/0     0/0      ☢  │   │   │       │   │   ├── ahash 0.7.6
1/4        49/175       1/1    0/0     3/3      ☢  │   │   │       │   │   │   ├── getrandom 0.2.8
0/0        0/0          0/0    0/0     0/0      ?  │   │   │       │   │   │   │   ├── cfg-if 1.0.0
1/24       10/444       0/2    0/0     5/45     ☢  │   │   │       │   │   │   │   └── libc 0.2.139
1/1        92/138       5/9    0/0     2/4      ☢  │   │   │       │   │   │   ├── once_cell 1.17.0
0/16       0/1327       0/0    0/0     0/56     ?  │   │   │       │   │   │   │   └── parking_lot_core 0.9.6
0/0        0/0          0/0    0/0     0/0      ?  │   │   │       │   │   │   │       ├── cfg-if 1.0.0
1/24       10/444       0/2    0/0     5/45     ☢  │   │   │       │   │   │   │       ├── libc 0.2.139
0/2        0/73         0/5    0/1     0/1      ?  │   │   │       │   │   │   │       ├── petgraph 0.6.2
0/0        0/70         0/0    0/0     0/0      ?  │   │   │       │   │   │   │       │   ├── fixedbitset 0.4.2
0/0        5/5          0/0    0/0     0/0      ☢  │   │   │       │   │   │   │       │   │   └── serde 1.0.152
0/0        41/46        1/1    0/0     0/0      ☢  │   │   │       │   │   │   │       │   ├── indexmap 1.9.2
0/0        5/5          0/0    0/0     0/0      ☢  │   │   │       │   │   │   │       │   ├── serde 1.0.152
0/0        0/0          0/0    0/0     0/0      ?  │   │   │       │   │   │   │       │   └── serde_derive 1.0.152
0/1        0/399        0/7    0/1     0/13     ?  │   │   │       │   │   │   │       └── smallvec 1.10.0
0/0        5/5          0/0    0/0     0/0      ☢  │   │   │       │   │   │   │           └── serde 1.0.152
0/0        5/5          0/0    0/0     0/0      ☢  │   │   │       │   │   │   └── serde 1.0.152
4/6        437/1158     4/10   1/1     13/26    ☢  │   │   │       │   │   ├── bumpalo 3.12.0
6/6        659/659      5/5    0/0     3/3      ☢  │   │   │       │   │   ├── rayon 1.6.1
0/0        14/14        0/0    0/0     0/0      ☢  │   │   │       │   │   │   ├── either 1.8.0
0/0        5/5          0/0    0/0     0/0      ☢  │   │   │       │   │   │   │   └── serde 1.0.152
7/7        657/660      5/5    0/0     33/33    ☢  │   │   │       │   │   │   └── rayon-core 1.10.2
2/2        485/494      6/7    0/0     12/14    ☢  │   │   │       │   │   │       ├── crossbeam-channel 0.5.6
0/0        0/0          0/0    0/0     0/0      ?  │   │   │       │   │   │       │   ├── cfg-if 1.0.0
4/4        94/94        16/16  0/0     3/3      ☢  │   │   │       │   │   │       │   └── crossbeam-utils 0.8.14
0/0        0/0          0/0    0/0     0/0      ?  │   │   │       │   │   │       │       └── cfg-if 1.0.0
0/0        453/453      6/6    0/0     6/6      ☢  │   │   │       │   │   │       ├── crossbeam-deque 0.8.2
0/0        0/0          0/0    0/0     0/0      ?  │   │   │       │   │   │       │   ├── cfg-if 1.0.0
3/3        448/460      11/11  0/0     29/29    ☢  │   │   │       │   │   │       │   ├── crossbeam-epoch 0.9.13
0/0        0/0          0/0    0/0     0/0      ?  │   │   │       │   │   │       │   │   ├── cfg-if 1.0.0
4/4        94/94        16/16  0/0     3/3      ☢  │   │   │       │   │   │       │   │   ├── crossbeam-utils 0.8.14
0/0        0/0          0/0    0/0     0/0      ?  │   │   │       │   │   │       │   │   ├── memoffset 0.7.1
0/0        18/18        1/1    0/0     0/0      ☢  │   │   │       │   │   │       │   │   └── scopeguard 1.1.0
4/4        94/94        16/16  0/0     3/3      ☢  │   │   │       │   │   │       │   └── crossbeam-utils 0.8.14
4/4        94/94        16/16  0/0     3/3      ☢  │   │   │       │   │   │       ├── crossbeam-utils 0.8.14
0/0        65/72        0/0    0/0     0/0      ☢  │   │   │       │   │   │       └── num_cpus 1.15.0
1/24       10/444       0/2    0/0     5/45     ☢  │   │   │       │   │   │           └── libc 0.2.139
0/0        5/5          0/0    0/0     0/0      ☢  │   │   │       │   │   └── serde 1.0.152
6/6        659/659      5/5    0/0     3/3      ☢  │   │   │       │   ├── rayon 1.6.1
0/0        5/5          0/0    0/0     0/0      ☢  │   │   │       │   └── serde 1.0.152
```

AUTOMATED TESTING



```
0/0    5/5        0/0    0/0   0/0    ☢️  │   │   │       │   └── serde 1.0.152
0/0    5/5        0/0    0/0   0/0    ☢️  │   │   │       └── serde 1.0.152
0/0    15/15      0/0    0/0   3/3    ☢️  │   │   ├── proc-macro2 1.0.50
0/0    69/69      3/3    0/0   2/2    ☢️  │   │   └── syn 1.0.107
2/2    1064/1198  19/22  1/1   51/58  ☢️  │   └── hashbrown 0.11.2
0/0    26/30      0/0    0/0   0/0    ☢️  │       ├── ahash 0.7.6
4/6    437/1158   4/10   1/1   13/26  ☢️  │       ├── bumpalo 3.12.0
6/6    659/659    5/5    0/0   3/3    ☢️  │       ├── rayon 1.6.1
0/0    5/5        0/0    0/0   0/0    ☢️  │       └── serde 1.0.152
0/0    0/0        0/0    0/0   0/0    ❓  ├── mpl-token-metadata 1.4.3
0/0    0/0        0/0    0/0   0/0    ❓  │   ├── arrayref 0.3.6
0/0    7/7        0/0    0/0   0/0    ☢️  │   ├── borsh 0.9.3
0/0    0/0        0/0    0/0   0/0    ❓  │   ├── mpl-token-vault 0.1.0
0/0    7/7        0/0    0/0   0/0    ☢️  │   │   ├── borsh 0.9.3
0/0    0/0        0/0    0/0   0/0    ❓  │   │   ├── num-derive 0.3.3
0/0    15/15      0/0    0/0   3/3    ☢️  │   │   │   ├── proc-macro2 1.0.50
0/0    0/0        0/0    0/0   0/0    ❓  │   │   │   ├── quote 1.0.23
0/0    69/69      3/3    0/0   2/2    ☢️  │   │   │   └── syn 1.0.107
0/0    6/12       0/0    0/0   0/0    ☢️  │   │   ├── num-traits 0.2.15
0/0    0/8        0/0    0/0   0/0    ❓  │   │   │   └── libm 0.2.6
3/3    442/442    1/1    0/0   3/3    ☢️  │   │   ├── solana-program 1.14.10
0/0    0/0        0/0    0/0   0/0    🔒  │   │   │   ├── base64 0.13.1
0/0    22/22      0/0    0/0   0/0    ☢️  │   │   │   ├── bincode 1.3.3
0/0    0/0        0/0    0/0   0/0    ❓  │   │   │   ├── bitflags 1.3.2
2/78   29/3973    0/0    0/0   0/0    ☢️  │   │   │   ├── blake3 1.3.3
0/0    0/0        0/0    0/0   0/0    ❓  │   │   │   │   ├── arrayref 0.3.6
2/2    350/350    2/2    0/0   7/7    ☢️  │   │   │   │   ├── arrayvec 0.7.2
0/0    5/5        0/0    0/0   0/0    ☢️  │   │   │   │   │   └── serde 1.0.152
0/0    0/0        0/0    0/0   0/0    ❓  │   │   │   │   ├── cfg-if 1.0.0
0/0    4/4        0/0    0/0   0/0    ☢️  │   │   │   │   ├── constant_time_eq 0.2.4
0/0    0/0        0/0    0/0   0/0    🔒  │   │   │   │   ├── digest 0.10.6
0/0    16/16      0/0    0/0   0/0    ☢️  │   │   │   │   │   ├── block-buffer 0.10.3
1/1    285/285    20/20  8/8   5/5    ☢️  │   │   │   │   │   │   └── generic-array 0.14.6
0/0    5/5        0/0    0/0   0/0    ☢️  │   │   │   │   │   │       ├── serde 1.0.152
0/0    0/0        0/0    0/0   0/0    🔒  │   │   │   │   │   │       ├── typenum 1.16.0
1/1    23/23      0/0    0/0   0/0    ☢️  │   │   │   │   │   │       └── zeroize 1.3.0
0/0    0/0        0/0    0/0   0/0    🔒  │   │   │   │   │   │           └── zeroize_derive 1.3.3
0/0    15/15      0/0    0/0   3/3    ☢️  │   │   │   │   │   │               ├── proc-macro2 1.0.50
0/0    0/0        0/0    0/0   0/0    ❓  │   │   │   │   │   │               ├── quote 1.0.23
0/0    69/69      3/3    0/0   2/2    ☢️  │   │   │   │   │   │               ├── syn 1.0.107
0/0    0/0        1/1    0/0   0/0    ☢️  │   │   │   │   │   │               └── synstructure 0.12.6
0/0    15/15      0/0    0/0   3/3    ☢️  │   │   │   │   │   │                   ├── proc-macro2 1.0.50
0/0    0/0        0/0    0/0   0/0    ❓  │   │   │   │   │   │                   ├── quote 1.0.23
0/0    69/69      3/3    0/0   2/2    ☢️  │   │   │   │   │   │                   ├── syn 1.0.107
0/0    0/0        0/0    0/0   0/0    🔒  │   │   │   │   │   │                   └── unicode-xid 0.2.4
0/0    0/0        0/0    0/0   0/0    🔒  │   │   │   │   │   ├── crypto-common 0.1.6
1/1    285/285    20/20  8/8   5/5    ☢️  │   │   │   │   │   │   ├── generic-array 0.14.6
0/0    2/2        0/0    0/0   0/0    ☢️  │   │   │   │   │   │   ├── rand_core 0.6.4
1/4    49/175     1/1    0/0   3/3    ☢️  │   │   │   │   │   │   │   ├── getrandom 0.2.8
0/0    5/5        0/0    0/0   0/0    ☢️  │   │   │   │   │   │   │   └── serde 1.0.152
0/0    0/0        0/0    0/0   0/0    🔒  │   │   │   │   │   │   └── typenum 1.16.0
0/0    3/3        0/0    0/0   0/0    ☢️  │   │   │   │   │   └── subtle 2.4.1
6/6    659/659    5/5    0/0   3/3    ☢️  │   │   │   │   └── rayon 1.6.1
0/0    7/7        0/0    0/0   0/0    ☢️  │   │   │   ├── borsh 0.9.3
0/0    0/0        0/0    0/0   0/0    ❓  │   │   │   ├── borsh-derive 0.9.3
0/0    1/1        0/0    0/0   0/0    ☢️  │   │   │   ├── bs58 0.4.0
0/8    10/202     0/0    0/0   0/0    ☢️  │   │   │   │   └── sha2 0.9.9
0/0    6/6        0/0    0/0   0/0    ☢️  │   │   │   │       ├── block-buffer 0.9.0
0/0    3/3        0/0    0/0   0/0    ☢️  │   │   │   │       │   ├── block-padding 0.2.1
1/1    285/285    20/20  8/8   5/5    ☢️  │   │   │   │       │   └── generic-array 0.14.6
0/0    0/0        0/0    0/0   0/0    ❓  │   │   │   │       ├── cfg-if 1.0.0
1/1    14/14      0/0    0/0   0/0    ☢️  │   │   │   │       ├── cpufeatures 0.2.5
1/24   10/444     0/2    0/0   5/45   ☢️  │   │   │   │       │   └── libc 0.2.139
0/0    0/0        0/0    0/0   0/0    🔒  │   │   │   │       ├── digest 0.9.0
1/1    285/285    20/20  8/8   5/5    ☢️  │   │   │   │       │   └── generic-array 0.14.6
0/0    0/0        0/0    0/0   0/0    ❓  │   │   │   │       └── opaque-debug 0.3.0
2/2    206/206    0/0    0/0   7/7    ☢️  │   │   │   ├── bv 0.11.1
0/0    5/5        0/0    0/0   0/0    ☢️  │   │   │   │   └── serde 1.0.152
18/18  370/414    128/129 9/9  0/0    ☢️  │   │   │   ├── bytemuck 1.13.0
0/0    0/0        0/0    0/0   0/0    ❓  │   │   │   │   └── bytemuck_derive 1.4.0
0/0    15/15      0/0    0/0   3/3    ☢️  │   │   │   │       ├── proc-macro2 1.0.50
0/0    0/0        0/0    0/0   0/0    ❓  │   │   │   │       ├── quote 1.0.23
0/0    69/69      3/3    0/0   2/2    ☢️  │   │   │   │       └── syn 1.0.107
0/2    0/857      0/0    0/0   0/0    ❓  │   │   │   ├── curve25519-dalek 3.2.1
1/1    193/193    0/0    0/0   0/0    ☢️  │   │   │   │   ├── byteorder 1.4.3
0/0    0/0        0/0    0/0   0/0    🔒  │   │   │   │   ├── digest 0.9.0
0/0    22/22      0/0    0/0   0/0    ☢️  │   │   │   │   ├── rand_core 0.5.1
1/4    47/150     1/1    0/0   3/3    ☢️  │   │   │   │   │   ├── getrandom 0.1.16
0/0    0/0        0/0    0/0   0/0    ❓  │   │   │   │   │   │   ├── cfg-if 1.0.0
```

AUTOMATED TESTING

```
0/0    0/0      0/0   0/0  0/0    ?   │   │   │   │   │   │   ├── cfg-if 1.0.0
1/24   10/444   0/2   0/0  5/45   ☢   │   │   │   │   │   │   ├── libc 0.2.139
1/1    16/18    1/1   0/0  0/0    ☢   │   │   │   │   │   │   └── log 0.4.17
0/0    0/0      0/0   0/0  0/0    ?   │   │   │   │   │   │       ├── cfg-if 1.0.0
0/0    5/5      0/0   0/0  0/0    ☢   │   │   │   │   │   │       └── serde 1.0.152
0/0    5/5      0/0   0/0  0/0    ☢   │   │   │   │   │   └── serde 1.0.152
0/0    5/5      0/0   0/0  0/0    ☢   │   │   │   │   ├── serde 1.0.152
0/0    3/3      0/0   0/0  0/0    ☢   │   │   │   │   ├── subtle 2.4.1
1/1    23/23    0/0   0/0  0/0    ☢   │   │   │   │   └── zeroize 1.3.0
0/0    0/72     0/3   0/1  0/3    ?   │   │   │   ├── itertools 0.10.5
0/0    14/14    0/0   0/0  0/0    ☢   │   │   │   │   └── either 1.8.0
0/0    0/72     0/3   0/1  0/3    ?   │   │   │   ├── itertools 0.10.5
0/0    7/7      1/1   0/0  0/0    ☢   │   │   │   ├── lazy_static 1.4.0
0/0    0/49     0/6   0/0  0/3    ?   │   │   │   │   └── spin 0.5.2
1/24   10/444   0/2   0/0  5/45   ☢   │   │   │   ├── libc 0.2.139
0/0    4/4      0/0   0/0  0/0    ☢   │   │   │   ├── libsecp256k1 0.6.0
0/0    0/0      0/0   0/0  0/0    ?   │   │   │   │   ├── arrayref 0.3.6
0/0    0/0      0/0   0/0  0/0    🔒   │   │   │   │   ├── base64 0.12.3
0/0    0/0      0/0   0/0  0/0    🔒   │   │   │   │   ├── digest 0.9.0
0/0    0/0      0/0   0/0  0/0    ?   │   │   │   │   ├── hmac-drbg 0.3.0
0/0    0/0      0/0   0/0  0/0    🔒   │   │   │   │   │   ├── digest 0.9.0
1/1    285/285  20/20 8/8  5/5    ☢   │   │   │   │   │   ├── generic-array 0.14.6
0/0    0/0      0/0   0/0  0/0    🔒   │   │   │   │   │   └── hmac 0.8.1
0/0    0/0      0/0   0/0  0/0    🔒   │   │   │   │   │       ├── crypto-mac 0.8.0
1/1    285/285  20/20 8/8  5/5    ☢   │   │   │   │   │       │   ├── generic-array 0.14.6
0/0    3/3      0/0   0/0  0/0    ☢   │   │   │   │   │       │   └── subtle 2.4.1
0/0    0/0      0/0   0/0  0/0    🔒   │   │   │   │   │       └── digest 0.9.0
0/0    7/7      1/1   0/0  0/0    ☢   │   │   │   │   ├── lazy_static 1.4.0
0/0    33/33    0/0   0/0  2/2    ☢   │   │   │   │   ├── libsecp256k1-core 0.2.2
0/0    0/0      0/0   0/0  0/0    ?   │   │   │   │   │   ├── crunchy 0.2.2
0/0    0/0      0/0   0/0  0/0    🔒   │   │   │   │   │   ├── digest 0.9.0
0/0    3/3      0/0   0/0  0/0    ☢   │   │   │   │   │   └── subtle 2.4.1
0/0    15/15    0/0   0/0  0/0    ☢   │   │   │   │   ├── rand 0.7.3
1/4    47/150   1/1   0/0  3/3    ☢   │   │   │   │   │   ├── getrandom 0.1.16
1/24   10/444   0/2   0/0  5/45   ☢   │   │   │   │   │   ├── libc 0.2.139
1/1    16/18    1/1   0/0  0/0    ☢   │   │   │   │   │   ├── log 0.4.17
0/0    0/0      0/0   0/0  0/0    ?   │   │   │   │   │   ├── rand_chacha 0.2.2
0/2    165/712  0/0   0/0  16/25  ☢   │   │   │   │   │   │   ├── ppv-lite86 0.2.17
0/0    22/22    0/0   0/0  0/0    ☢   │   │   │   │   │   │   └── rand_core 0.5.1
0/0    22/22    0/0   0/0  0/0    ☢   │   │   │   │   │   └── rand_core 0.5.1
0/0    5/5      0/0   0/0  0/0    ☢   │   │   │   │   ├── serde 1.0.152
0/8    10/202   0/0   0/0  0/0    ☢   │   │   │   │   ├── sha2 0.9.9
0/0    0/0      0/0   0/0  0/0    🔒   │   │   │   │   └── typenum 1.16.0
1/1    16/18    1/1   0/0  0/0    ☢   │   │   │   ├── log 0.4.17
0/0    0/0      0/0   0/0  0/0    ?   │   │   │   ├── memoffset 0.6.5
0/0    0/0      0/0   0/0  0/0    ?   │   │   │   ├── num-derive 0.3.3
0/0    6/12     0/0   0/0  0/0    ☢   │   │   │   ├── num-traits 0.2.15
0/0    15/15    0/0   0/0  0/0    ☢   │   │   │   ├── rand 0.7.3
0/0    0/0      0/0   0/0  0/0    ?   │   │   │   ├── rand_chacha 0.2.2
0/1    0/1      0/0   0/0  0/0    ?   │   │   │   ├── rustversion 1.0.11
0/0    5/5      0/0   0/0  0/0    ☢   │   │   │   ├── serde 1.0.152
0/0    16/16    0/0   0/0  0/0    ☢   │   │   │   ├── serde_bytes 0.11.8
0/0    5/5      0/0   0/0  0/0    ☢   │   │   │   │   └── serde 1.0.152
0/0    0/0      0/0   0/0  0/0    ?   │   │   │   ├── serde_derive 1.0.152
0/0    4/7      0/0   0/0  0/0    ☢   │   │   │   ├── serde_json 1.0.91
0/0    41/46    1/1   0/0  0/0    ☢   │   │   │   │   ├── indexmap 1.9.2
0/0    7/7      0/0   0/0  0/0    ☢   │   │   │   │   ├── itoa 1.0.5
7/9    579/715  0/0   0/0  2/2    ☢   │   │   │   │   ├── ryu 1.0.12
0/0    5/5      0/0   0/0  0/0    ☢   │   │   │   │   └── serde 1.0.152
0/8    4/196    0/0   0/0  0/0    ☢   │   │   │   ├── sha2 0.10.6
0/0    0/0      0/0   0/0  0/0    ?   │   │   │   │   ├── cfg-if 1.0.0
1/1    14/14    0/0   0/0  0/0    ☢   │   │   │   │   ├── cpufeatures 0.2.5
0/0    0/0      0/0   0/0  0/0    🔒   │   │   │   │   └── digest 0.10.6
0/0    0/0      0/0   0/0  0/0    🔒   │   │   │   ├── sha3 0.10.6
0/0    0/0      0/0   0/0  0/0    🔒   │   │   │   │   ├── digest 0.10.6
0/1    1/2      0/0   0/0  0/0    ☢   │   │   │   │   └── keccak 0.1.3
1/1    14/14    0/0   0/0  0/0    ☢   │   │   │   │       └── cpufeatures 0.2.5
0/0    0/0      0/0   0/0  0/0    ?   │   │   │   ├── solana-frozen-abi 1.14.10
0/0    26/30    0/0   0/0  0/0    ☢   │   │   │   │   ├── ahash 0.7.6
2/78   29/3973  0/0   0/0  0/0    ☢   │   │   │   │   ├── blake3 1.3.3
0/0    6/6      0/0   0/0  0/0    ☢   │   │   │   │   ├── block-buffer 0.9.0
0/0    1/1      0/0   0/0  0/0    ☢   │   │   │   │   ├── bs58 0.4.0
2/2    206/206  0/0   0/0  7/7    ☢   │   │   │   │   ├── bv 0.11.1
1/1    193/193  0/0   0/0  0/0    ☢   │   │   │   │   ├── byteorder 1.4.3
1/1    29/201   0/2   0/0  0/4    ☢   │   │   │   │   ├── cc 1.0.78
0/0    198/311  0/2   0/0  4/6    ☢   │   │   │   │   │   └── jobserver 0.1.25
1/24   10/444   0/2   0/0  5/45   ☢   │   │   │   │   │       └── libc 0.2.139
0/0    14/14    0/0   0/0  0/0    ☢   │   │   │   │   ├── either 1.8.0
1/1    285/285  20/20 8/8  5/5    ☢   │   │   │   │   ├── generic-array 0.14.6
```

AUTOMATED TESTING

```
1/1   285/285    20/20  8/8   5/5    ☢  │   │   │   │   ├── generic-array 0.14.6
1/4   47/150     1/1    0/0   3/3    ☢  │   │   │   │   ├── getrandom 0.1.16
1/1   1223/1367  21/24  1/1   62/69  ☢  │   │   │   │   ├── hashbrown 0.12.3
1/1   122/122    2/2    0/0   4/4    ☢  │   │   │   │   ├── im 15.1.0
0/0   100/100    0/0    0/0   9/9    ☢  │   │   │   │   │   ├── bitmaps 2.1.0
0/0   0/0        0/0    0/0   0/0    🔒 │   │   │   │   │   │   └── typenum 1.16.0
0/0   0/9        0/0    0/0   0/0    ❓ │   │   │   │   │   ├── proptest 1.0.0
0/0   0/9        0/0    0/0   0/0    ❓ │   │   │   │   │   │   ├── bit-set 0.5.3
0/0   0/4        0/0    0/0   0/2    ❓ │   │   │   │   │   │   │   └── bit-vec 0.6.3
0/0   5/5        0/0    0/0   0/0    ☢  │   │   │   │   │   │   │       └── serde 1.0.152
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   │   │   │   ├── bitflags 1.3.2
1/1   193/193    0/0    0/0   0/0    ☢  │   │   │   │   │   │   ├── byteorder 1.4.3
0/0   7/7        1/1    0/0   0/0    ☢  │   │   │   │   │   │   ├── lazy_static 1.4.0
0/0   6/12       0/0    0/0   0/0    ☢  │   │   │   │   │   │   ├── num-traits 0.2.15
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   │   │   │   ├── quick-error 2.0.1
0/0   0/32       0/0    0/0   0/0    ❓ │   │   │   │   │   │   ├── rand 0.8.5
1/24  10/444     0/2    0/0   5/45   ☢  │   │   │   │   │   │   │   ├── libc 0.2.139
1/1   16/18      1/1    0/0   0/0    ☢  │   │   │   │   │   │   │   ├── log 0.4.17
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   │   │   │   │   ├── rand_chacha 0.3.1
0/2   165/712    0/0    0/0   16/25  ☢  │   │   │   │   │   │   │   │   ├── ppv-lite86 0.2.17
0/0   2/2        0/0    0/0   0/0    ☢  │   │   │   │   │   │   │   │   ├── rand_core 0.6.4
0/0   5/5        0/0    0/0   0/0    ☢  │   │   │   │   │   │   │   │   └── serde 1.0.152
0/0   2/2        0/0    0/0   0/0    ☢  │   │   │   │   │   │   │   ├── rand_core 0.6.4
0/0   5/5        0/0    0/0   0/0    ☢  │   │   │   │   │   │   │   └── serde 1.0.152
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   │   │   │   ├── rand_chacha 0.3.1
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   │   │   │   ├── rand_xorshift 0.3.0
0/0   2/2        0/0    0/0   0/0    ☢  │   │   │   │   │   │   │   ├── rand_core 0.6.4
0/0   5/5        0/0    0/0   0/0    ☢  │   │   │   │   │   │   │   └── serde 1.0.152
0/0   0/0        0/0    0/0   0/0    🔒 │   │   │   │   │   │   ├── regex-syntax 0.6.28
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   │   │   │   ├── rusty-fork 0.3.0
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   │   │   │   │   ├── fnv 1.0.7
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   │   │   │   │   ├── quick-error 1.2.3
0/0   0/71       0/0    0/0   0/0    ❓ │   │   │   │   │   │   │   ├── tempfile 3.3.0
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   │   │   │   │   │   ├── cfg-if 1.0.0
0/0   0/0        0/0    0/0   0/0    🔒 │   │   │   │   │   │   │   │   ├── fastrand 1.8.0
1/24  10/444     0/2    0/0   5/45   ☢  │   │   │   │   │   │   │   │   ├── libc 0.2.139
0/0   0/79       0/0    0/0   0/0    ❓ │   │   │   │   │   │   │   │   └── remove_dir_all 0.5.3
0/0   0/66       0/0    0/0   0/0    ❓ │   │   │   │   │   │   │   └── wait-timeout 0.2.0
1/24  10/444     0/2    0/0   5/45   ☢  │   │   │   │   │   │   │       └── libc 0.2.139
0/0   0/71       0/0    0/0   0/0    ❓ │   │   │   │   │   │   └── tempfile 3.3.0
0/0   2/2        0/0    0/0   0/0    ☢  │   │   │   │   │   ├── rand_core 0.6.4
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   │   │   ├── rand_xoshiro 0.6.0
0/0   2/2        0/0    0/0   0/0    ☢  │   │   │   │   │   │   ├── rand_core 0.6.4
0/0   5/5        0/0    0/0   0/0    ☢  │   │   │   │   │   │   └── serde 1.0.152
6/6   659/659    5/5    0/0   3/3    ☢  │   │   │   │   │   ├── rayon 1.6.1
0/0   5/5        0/0    0/0   0/0    ☢  │   │   │   │   │   ├── serde 1.0.152
0/1   323/643    0/0    0/0   20/39  ☢  │   │   │   │   │   ├── sized-chunks 0.6.5
0/0   100/100    0/0    0/0   9/9    ☢  │   │   │   │   │   │   ├── bitmaps 2.1.0
0/0   0/0        0/0    0/0   0/0    🔒 │   │   │   │   │   │   └── typenum 1.16.0
0/0   0/0        0/0    0/0   0/0    🔒 │   │   │   │   │   └── typenum 1.16.0
0/0   7/7        1/1    0/0   0/0    ☢  │   │   │   │   ├── lazy_static 1.4.0
1/1   16/18      1/1    0/0   0/0    ☢  │   │   │   │   ├── log 0.4.17
0/0   161/293    4/6    0/0   7/7    ☢  │   │   │   │   ├── memmap2 0.5.8
1/24  10/444     0/2    0/0   5/45   ☢  │   │   │   │   │   └── libc 0.2.139
1/1   92/138     5/9    0/0   2/4    ☢  │   │   │   │   ├── once_cell 1.17.0
1/1   92/138     5/9    0/0   2/4    ☢  │   │   │   │   ├── once_cell 1.17.0
0/0   2/2        0/0    0/0   0/0    ☢  │   │   │   │   ├── rand_core 0.6.4
0/0   5/5        0/0    0/0   0/0    ☢  │   │   │   │   ├── serde 1.0.152
0/0   16/16      0/0    0/0   0/0    ☢  │   │   │   │   ├── serde_bytes 0.11.8
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   │   ├── serde_derive 1.0.152
0/0   4/7        0/0    0/0   0/0    ☢  │   │   │   │   ├── serde_json 1.0.91
0/8   4/196      0/0    0/0   0/0    ☢  │   │   │   │   ├── sha2 0.10.6
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   │   ├── solana-frozen-abi-macro 1.14.10
0/0   15/15      0/0    0/0   3/3    ☢  │   │   │   │   │   ├── proc-macro2 1.0.50
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   │   │   ├── quote 1.0.23
0/0   69/69      3/3    0/0   2/2    ☢  │   │   │   │   │   └── syn 1.0.107
0/0   3/3        0/0    0/0   0/0    ☢  │   │   │   │   ├── subtle 2.4.1
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   │   └── thiserror 1.0.38
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   │       └── thiserror-impl 1.0.38
0/0   15/15      0/0    0/0   3/3    ☢  │   │   │   │           ├── proc-macro2 1.0.50
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   │           ├── quote 1.0.23
0/0   69/69      3/3    0/0   2/2    ☢  │   │   │   │           └── syn 1.0.107
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   ├── solana-frozen-abi-macro 1.14.10
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   ├── solana-sdk-macro 1.14.10
0/0   1/1        0/0    0/0   0/0    ☢  │   │   │   │   ├── bs58 0.4.0
0/0   15/15      0/0    0/0   3/3    ☢  │   │   │   │   ├── proc-macro2 1.0.50
0/0   0/0        0/0    0/0   0/0    ❓ │   │   │   │   ├── quote 1.0.23
0/1   0/1        0/0    0/0   0/0    ❓ │   │   │   │   ├── rustversion 1.0.11
0/0   69/69      3/3    0/0   2/2    ☢  │   │   │   │   └── syn 1.0.107
```

AUTOMATED TESTING

```
0/0     69/69     3/3     0/0  2/2    ☢  │   │   │   │   └── syn 1.0.107
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   ├── thiserror 1.0.38
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   ├── tiny-bip39 0.8.2
15/18   453/460   3/3     0/0  12/12  ☢  │   │   │   │   ├── anyhow 1.0.68
0/0     0/0       0/0     0/0  0/0    🔒 │   │   │   │   ├── hmac 0.8.1
1/1     92/138    5/9     0/0  2/4    ☢  │   │   │   │   ├── once_cell 1.17.0
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   │   ├── pbkdf2 0.4.0
0/0     0/0       0/0     0/0  0/0    🔒 │   │   │   │   │   ├── base64 0.12.3
0/0     0/0       0/0     0/0  0/0    🔒 │   │   │   │   │   ├── crypto-mac 0.8.0
0/0     0/0       0/0     0/0  0/0    🔒 │   │   │   │   │   ├── hmac 0.8.1
0/0     15/15     0/0     0/0  0/0    ☢  │   │   │   │   │   ├── rand 0.7.3
0/0     22/22     0/0     0/0  0/0    ☢  │   │   │   │   │   ├── rand_core 0.5.1
6/6     659/659   5/5     0/0  3/3    ☢  │   │   │   │   │   ├── rayon 1.6.1
0/8     10/202    0/0     0/0  0/0    ☢  │   │   │   │   │   ├── sha2 0.9.9
0/0     3/3       0/0     0/0  0/0    ☢  │   │   │   │   │   └── subtle 2.4.1
0/0     15/15     0/0     0/0  0/0    ☢  │   │   │   │   ├── rand 0.7.3
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   │   ├── rustc-hash 1.1.0
0/8     10/202    0/0     0/0  0/0    ☢  │   │   │   │   ├── sha2 0.9.9
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   │   ├── thiserror 1.0.38
0/0     20/20     0/0     0/0  0/0    ☢  │   │   │   │   ├── unicode-normalization 0.1.22
0/0     0/0       0/0     0/0  0/0    🔒 │   │   │   │   │   └── tinyvec 1.6.0
0/0     5/5       0/0     0/0  0/0    ☢  │   │   │   │   │       ├── serde 1.0.152
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   │   │       └── tinyvec_macros 0.1.0
1/1     23/23     0/0     0/0  0/0    ☢  │   │   │   │   └── zeroize 1.3.0
12/14   438/502   13/13   2/2  10/10  ☢  │   │   │   ├── wasm-bindgen 0.2.83
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   │   ├── cfg-if 1.0.0
0/0     5/5       0/0     0/0  0/0    ☢  │   │   │   │   ├── serde 1.0.152
0/0     4/7       0/0     0/0  0/0    ☢  │   │   │   │   ├── serde_json 1.0.91
0/0     0/0       0/1     0/0  0/0    ?  │   │   │   │   └── wasm-bindgen-macro 0.2.83
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   │       ├── quote 1.0.23
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   │       └── wasm-bindgen-macro-support 0.2.83
0/0     15/15     0/0     0/0  3/3    ☢  │   │   │   │           ├── proc-macro2 1.0.50
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   │           ├── quote 1.0.23
0/0     69/69     3/3     0/0  2/2    ☢  │   │   │   │           ├── syn 1.0.107
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   │           ├── wasm-bindgen-backend 0.2.83
4/6     437/1158  4/10    1/1  13/26  ☢  │   │   │   │           │   ├── bumpalo 3.12.0
1/1     16/18     1/1     0/0  0/0    ☢  │   │   │   │           │   ├── log 0.4.17
1/1     92/138    5/9     0/0  2/4    ☢  │   │   │   │           │   ├── once_cell 1.17.0
0/0     15/15     0/0     0/0  3/3    ☢  │   │   │   │           │   ├── proc-macro2 1.0.50
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   │           │   ├── quote 1.0.23
0/0     69/69     3/3     0/0  2/2    ☢  │   │   │   │           │   ├── syn 1.0.107
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   │           │   └── wasm-bindgen-shared 0.2.83
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   │           └── wasm-bindgen-shared 0.2.83
1/1     23/23     0/0     0/0  0/0    ☢  │   │   │   └── zeroize 1.3.0
0/0     0/0       0/0     0/0  0/0    ?  │   │   ├── spl-token 3.5.0
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   ├── arrayref 0.3.6
18/18   370/414   128/129 9/9  0/0    ☢  │   │   │   ├── bytemuck 1.13.0
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   ├── num-derive 0.3.3
0/0     6/12      0/0     0/0  0/0    ☢  │   │   │   ├── num-traits 0.2.15
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   ├── num_enum 0.5.7
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   │   └── num_enum_derive 0.5.7
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   │       ├── proc-macro-crate 1.3.0
1/1     92/138    5/9     0/0  2/4    ☢  │   │   │   │       │   ├── once_cell 1.17.0
1/1     24/24     0/0     0/0  0/0    ☢  │   │   │   │       │   └── toml_edit 0.18.0
0/0     41/46     1/1     0/0  0/0    ☢  │   │   │   │       │       ├── indexmap 1.9.2
0/0     40/40     0/0     0/0  0/0    ☢  │   │   │   │       │       ├── nom8 0.2.0
2/37    361/2144  0/0     0/0  4/21   ☢  │   │   │   │       │       │   └── memchr 2.5.0
1/24    10/444    0/2     0/0  5/45   ☢  │   │   │   │       │       │       └── libc 0.2.139
0/0     5/5       0/0     0/0  0/0    ☢  │   │   │   │       │       ├── serde 1.0.152
0/0     0/0       0/0     0/0  0/0    🔒 │   │   │   │       │       └── toml_datetime 0.5.1
0/0     5/5       0/0     0/0  0/0    ☢  │   │   │   │       │           └── serde 1.0.152
0/0     15/15     0/0     0/0  3/3    ☢  │   │   │   │       ├── proc-macro2 1.0.50
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   │       ├── quote 1.0.23
0/0     69/69     3/3     0/0  2/2    ☢  │   │   │   │       └── syn 1.0.107
3/3     442/442   1/1     0/0  3/3    ☢  │   │   │   ├── solana-program 1.14.10
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   └── thiserror 1.0.38
0/0     0/0       0/0     0/0  0/0    ?  │   │   └── thiserror 1.0.38
0/0     0/0       0/0     0/0  0/0    ?  │   ├── num-derive 0.3.3
0/0     6/12      0/0     0/0  0/0    ☢  │   ├── num-traits 0.2.15
0/0     5/5       0/0     0/0  0/0    ☢  │   ├── serde 1.0.152
0/0     0/0       0/0     0/0  0/0    ?  │   ├── shank 0.0.12
0/0     0/0       0/0     0/0  0/0    ?  │   │   └── shank_macro 0.0.12
0/0     15/15     0/0     0/0  3/3    ☢  │   │       ├── proc-macro2 1.0.50
0/0     0/0       0/0     0/0  0/0    ?  │   │       ├── quote 1.0.23
0/0     0/0       0/0     0/0  0/0    ?  │   │       ├── shank_macro_impl 0.0.12
15/18   453/460   3/3     0/0  12/12  ☢  │   │       │   ├── anyhow 1.0.68
0/0     15/15     0/0     0/0  3/3    ☢  │   │       │   ├── proc-macro2 1.0.50
0/0     0/0       0/0     0/0  0/0    ?  │   │       │   ├── quote 1.0.23
0/0     5/5       0/0     0/0  0/0    ☢  │   │       │   ├── serde 1.0.152
```

AUTOMATED TESTING

```
0/0    5/5      0/0     0/0  0/0  ☢ │   │   │   │   ├── serde 1.0.152
0/0    69/69    3/3     0/0  2/2  ☢ │   │   │   │   └── syn 1.0.107
0/0    0/0      0/0     0/0  0/0  ? │   │   │   ├── shank_render 0.0.1
0/0    15/15    0/0     0/0  3/3  ☢ │   │   │   │   ├── proc-macro2 1.0.50
0/0    0/0      0/0     0/0  0/0  ? │   │   │   │   ├── quote 1.0.23
0/0    0/0      0/0     0/0  0/0  ? │   │   │   │   └── shank_macro_impl 0.0.12
0/0    69/69    3/3     0/0  2/2  ☢ │   │   │   └── syn 1.0.107
3/3    442/442  1/1     0/0  3/3  ☢ │   ├── solana-program 1.14.10
0/0    0/0      0/0     0/0  0/0  🔒 │   ├── spl-associated-token-account 1.1.2
0/0    0/0      0/0     0/0  0/0  ? │   │   ├── assert_matches 1.5.0
0/0    7/7      0/0     0/0  0/0  ☢ │   │   ├── borsh 0.9.3
0/0    0/0      0/0     0/0  0/0  ? │   │   ├── num-derive 0.3.3
0/0    6/12     0/0     0/0  0/0  ☢ │   │   ├── num-traits 0.2.15
3/3    442/442  1/1     0/0  3/3  ☢ │   │   ├── solana-program 1.14.10
0/0    0/0      0/0     0/0  0/0  ? │   │   ├── spl-token 3.5.0
0/0    0/0      0/0     0/0  0/0  ? │   │   ├── spl-token-2022 0.5.0
0/0    0/0      0/0     0/0  0/0  ? │   │   │   ├── arrayref 0.3.6
18/18  370/414  128/129 9/9  0/0  ☢ │   │   │   ├── bytemuck 1.13.0
0/0    0/0      0/0     0/0  0/0  ? │   │   │   ├── num-derive 0.3.3
0/0    6/12     0/0     0/0  0/0  ☢ │   │   │   ├── num-traits 0.2.15
0/0    0/0      0/0     0/0  0/0  ? │   │   │   ├── num_enum 0.5.7
0/0    5/5      0/0     0/0  0/0  ☢ │   │   │   ├── serde 1.0.152
3/3    442/442  1/1     0/0  3/3  ☢ │   │   │   ├── solana-program 1.14.10
0/0    70/70    18/18   0/0  0/0  ☢ │   │   │   ├── solana-zk-token-sdk 1.14.10
0/0    0/0      0/0     0/0  0/0  ? │   │   │   │   ├── aes-gcm-siv 0.10.3
0/0    0/0      0/0     0/0  0/0  🔒 │   │   │   │   │   ├── aead 0.4.3
1/1    285/285  20/20   8/8  5/5  ☢ │   │   │   │   │   │   ├── generic-array 0.14.6
0/0    2/2      0/0     0/0  0/0  ☢ │   │   │   │   │   │   └── rand_core 0.6.4
0/28   0/1060   0/0     0/0  0/0  ? │   │   │   │   │   ├── aes 0.7.5
0/0    0/0      0/0     0/0  0/0  ? │   │   │   │   │   │   ├── cfg-if 1.0.0
0/0    0/0      0/0     0/0  0/0  🔒 │   │   │   │   │   │   ├── cipher 0.3.0
1/1    285/285  20/20   8/8  5/5  ☢ │   │   │   │   │   │   │   └── generic-array 0.14.6
1/1    14/14    0/0     0/0  0/0  ☢ │   │   │   │   │   │   ├── cpufeatures 0.2.5
0/0    3/3      0/0     0/0  0/0  ☢ │   │   │   │   │   │   ├── ctr 0.8.0
0/0    0/0      0/0     0/0  0/0  🔒 │   │   │   │   │   │   │   └── cipher 0.3.0
0/0    0/0      0/0     0/0  0/0  ? │   │   │   │   │   │   └── opaque-debug 0.3.0
0/0    0/0      0/0     0/0  0/0  🔒 │   │   │   │   │   ├── cipher 0.3.0
0/0    3/3      0/0     0/0  0/0  ☢ │   │   │   │   │   ├── ctr 0.8.0
0/3    0/137    0/0     0/0  0/2  ? │   │   │   │   │   ├── polyval 0.5.3
0/0    0/0      0/0     0/0  0/0  ? │   │   │   │   │   │   ├── cfg-if 1.0.0
1/1    14/14    0/0     0/0  0/0  ☢ │   │   │   │   │   │   ├── cpufeatures 0.2.5
0/0    0/0      0/0     0/0  0/0  ? │   │   │   │   │   │   ├── opaque-debug 0.3.0
0/0    0/0      0/0     0/0  0/0  🔒 │   │   │   │   │   │   ├── universal-hash 0.4.1
1/1    285/285  20/20   8/8  5/5  ☢ │   │   │   │   │   │   │   ├── generic-array 0.14.6
0/0    3/3      0/0     0/0  0/0  ☢ │   │   │   │   │   │   │   └── subtle 2.4.1
1/1    23/23    0/0     0/0  0/0  ☢ │   │   │   │   │   │   └── zeroize 1.3.0
0/0    3/3      0/0     0/0  0/0  ☢ │   │   │   │   │   ├── subtle 2.4.1
1/1    23/23    0/0     0/0  0/0  ☢ │   │   │   │   │   └── zeroize 1.3.0
0/0    0/0      0/0     0/0  0/0  ? │   │   │   │   ├── arrayref 0.3.6
0/0    0/0      0/0     0/0  0/0  🔒 │   │   │   │   ├── base64 0.13.1
0/0    22/22    0/0     0/0  0/0  ☢ │   │   │   │   ├── bincode 1.3.3
18/18  370/414  128/129 9/9  0/0  ☢ │   │   │   │   ├── bytemuck 1.13.0
1/1    193/193  0/0     0/0  0/0  ☢ │   │   │   │   ├── byteorder 1.4.3
0/0    7/7      0/0     0/0  0/0  ☢ │   │   │   │   ├── cipher 0.4.3
0/0    0/0      0/0     0/0  0/0  🔒 │   │   │   │   │   ├── crypto-common 0.1.6
0/0    126/126  0/0     0/0  3/3  ☢ │   │   │   │   │   └── inout 0.1.3
1/1    285/285  20/20   8/8  5/5  ☢ │   │   │   │   │       └── generic-array 0.14.6
0/2    0/857    0/0     0/0  0/0  ? │   │   │   │   ├── curve25519-dalek 3.2.1
1/4    47/150   1/1     0/0  3/3  ☢ │   │   │   │   ├── getrandom 0.1.16
0/0    0/72     0/3     0/1  0/3  ? │   │   │   │   ├── itertools 0.10.5
0/0    7/7      1/1     0/0  0/0  ☢ │   │   │   │   ├── lazy_static 1.4.0
0/0    5/5      0/0     0/0  0/0  ☢ │   │   │   │   ├── merlin 3.0.0
1/1    193/193  0/0     0/0  0/0  ☢ │   │   │   │   │   ├── byteorder 1.4.3
0/1    1/2      0/0     0/0  0/0  ☢ │   │   │   │   │   ├── keccak 0.1.3
0/0    2/2      0/0     0/0  0/0  ☢ │   │   │   │   │   ├── rand_core 0.6.4
1/1    23/23    0/0     0/0  0/0  ☢ │   │   │   │   │   └── zeroize 1.3.0
0/0    0/0      0/0     0/0  0/0  ? │   │   │   │   ├── num-derive 0.3.3
0/0    6/12     0/0     0/0  0/0  ☢ │   │   │   │   ├── num-traits 0.2.15
0/0    15/15    0/0     0/0  0/0  ☢ │   │   │   │   ├── rand 0.7.3
0/0    5/5      0/0     0/0  0/0  ☢ │   │   │   │   ├── serde 1.0.152
0/0    4/7      0/0     0/0  0/0  ☢ │   │   │   │   ├── serde_json 1.0.91
0/0    14/14    0/0     0/0  0/0  ☢ │   │   │   │   ├── sha3 0.9.1
0/0    6/6      0/0     0/0  0/0  ☢ │   │   │   │   │   ├── block-buffer 0.9.0
0/0    0/0      0/0     0/0  0/0  🔒 │   │   │   │   │   ├── digest 0.9.0
0/1    1/2      0/0     0/0  0/0  ☢ │   │   │   │   │   ├── keccak 0.1.3
0/0    0/0      0/0     0/0  0/0  ? │   │   │   │   │   └── opaque-debug 0.3.0
3/3    442/442  1/1     0/0  3/3  ☢ │   │   │   │   ├── solana-program 1.14.10
0/0    3/3      0/0     0/0  0/0  ☢ │   │   │   │   ├── solana-sdk 1.14.10
0/0    0/0      0/0     0/0  0/0  ? │   │   │   │   │   ├── assert_matches 1.5.0
```

AUTOMATED TESTING

```
0/0    0/0      0/0     0/0  0/0  ?  │ │ │ │ │ ├── assert_matches 1.5.0
0/0    0/0      0/0     0/0  0/0  🔒 │ │ │ │ │ ├── base64 0.13.1
0/0    22/22    0/0     0/0  0/0  ☢  │ │ │ │ │ ├── bincode 1.3.3
0/0    0/0      0/0     0/0  0/0  ?  │ │ │ │ │ ├── bitflags 1.3.2
0/0    7/7      0/0     0/0  0/0  ☢  │ │ │ │ │ ├── borsh 0.9.3
0/0    1/1      0/0     0/0  0/0  ☢  │ │ │ │ │ ├── bs58 0.4.0
18/18  370/414  128/129 9/9  0/0  ☢  │ │ │ │ │ ├── bytemuck 1.13.0
1/1    193/193  0/0     0/0  0/0  ☢  │ │ │ │ │ ├── byteorder 1.4.3
0/0    2/48     2/2     0/0  0/0  ☢  │ │ │ │ │ ├── chrono 0.4.23
0/2    21/147   0/0     0/0  1/1  ☢  │ │ │ │ │ │   ├── iana-time-zone 0.1.53
0/0    3/3      0/0     0/0  2/2  ☢  │ │ │ │ │ │   │   └── core-foundation-sys 0.8.3
0/0    0/0      0/0     0/0  0/0  ?  │ │ │ │ │ │   ├── num-integer 0.1.45
0/0    6/12     0/0     0/0  0/0  ☢  │ │ │ │ │ │   │   └── num-traits 0.2.15
0/0    6/12     0/0     0/0  0/0  ☢  │ │ │ │ │ │   ├── num-traits 0.2.15
0/0    5/5      0/0     0/0  0/0  ☢  │ │ │ │ │ │   ├── serde 1.0.152
1/1    221/221  0/0     0/0  0/0  ☢  │ │ │ │ │ │   └── time 0.1.45
1/24   10/444   0/2     0/0  5/45 ☢  │ │ │ │ │ │       └── libc 0.2.139
0/2    0/857    0/0     0/0  0/0  ?  │ │ │ │ │ ├── curve25519-dalek 3.2.1
0/0    0/0      0/0     0/0  0/0  ?  │ │ │ │ │ ├── derivation-path 0.2.0
0/0    0/0      0/0     0/0  0/0  🔒 │ │ │ │ │ ├── digest 0.10.6
0/0    0/0      0/0     0/0  0/0  🔒 │ │ │ │ │ ├── ed25519-dalek 1.0.1
0/2    0/857    0/0     0/0  0/0  ?  │ │ │ │ │ │   ├── curve25519-dalek 3.2.1
0/0    0/0      0/0     0/0  0/0  🔒 │ │ │ │ │ │   ├── ed25519 1.5.3
0/0    5/5      0/0     0/0  0/0  ☢  │ │ │ │ │ │   │   ├── serde 1.0.152
0/0    16/16    0/0     0/0  0/0  ☢  │ │ │ │ │ │   │   ├── serde_bytes 0.11.8
0/0    0/0      0/0     0/0  0/0  🔒 │ │ │ │ │ │   │   ├── signature 1.6.4
0/0    0/0      0/0     0/0  0/0  🔒 │ │ │ │ │ │   │   │   ├── digest 0.10.6
0/0    2/2      0/0     0/0  0/0  ☢  │ │ │ │ │ │   │   │   └── rand_core 0.6.4
1/1    23/23    0/0     0/0  0/0  ☢  │ │ │ │ │ │   │   └── zeroize 1.3.0
0/0    15/15    0/0     0/0  0/0  ☢  │ │ │ │ │ │   ├── rand 0.7.3
0/0    22/22    0/0     0/0  0/0  ☢  │ │ │ │ │ │   ├── rand_core 0.5.1
0/0    5/5      0/0     0/0  0/0  ☢  │ │ │ │ │ │   ├── serde 1.0.152
0/0    16/16    0/0     0/0  0/0  ☢  │ │ │ │ │ │   ├── serde_bytes 0.11.8
0/8    10/202   0/0     0/0  0/0  ☢  │ │ │ │ │ │   ├── sha2 0.9.9
1/1    23/23    0/0     0/0  0/0  ☢  │ │ │ │ │ │   └── zeroize 1.3.0
0/0    0/0      0/0     0/0  0/0  ?  │ │ │ │ │ ├── ed25519-dalek-bip32 0.2.0
0/0    0/0      0/0     0/0  0/0  ?  │ │ │ │ │ │   ├── derivation-path 0.2.0
0/0    0/0      0/0     0/0  0/0  🔒 │ │ │ │ │ │   ├── ed25519-dalek 1.0.1
0/0    0/0      0/0     0/0  0/0  🔒 │ │ │ │ │ │   ├── hmac 0.12.1
0/0    0/0      0/0     0/0  0/0  🔒 │ │ │ │ │ │   │   └── digest 0.10.6
0/8    4/196    0/0     0/0  0/0  ☢  │ │ │ │ │ │   └── sha2 0.10.6
1/1    285/285  20/20   8/8  5/5  ☢  │ │ │ │ │ ├── generic-array 0.14.6
0/0    0/0      0/0     0/0  0/0  🔒 │ │ │ │ │ ├── hmac 0.12.1
0/0    0/72     0/3     0/1  0/3  ?  │ │ │ │ │ ├── itertools 0.10.5
0/0    7/7      1/1     0/0  0/0  ☢  │ │ │ │ │ ├── lazy_static 1.4.0
0/0    4/4      0/0     0/0  0/0  ☢  │ │ │ │ │ ├── libsecp256k1 0.6.0
1/1    16/18    1/1     0/0  0/0  ☢  │ │ │ │ │ ├── log 0.4.17
0/0    161/293  4/6     0/0  7/7  ☢  │ │ │ │ │ ├── memmap2 0.5.8
0/0    0/0      0/0     0/0  0/0  ?  │ │ │ │ │ ├── num-derive 0.3.3
0/0    6/12     0/0     0/0  0/0  ☢  │ │ │ │ │ ├── num-traits 0.2.15
0/0    0/0      0/0     0/0  0/0  ?  │ │ │ │ │ ├── pbkdf2 0.11.0
0/0    0/0      0/0     0/0  0/0  🔒 │ │ │ │ │ │   ├── digest 0.10.6
0/0    0/0      0/0     0/0  0/0  🔒 │ │ │ │ │ │   ├── hmac 0.12.1
6/6    659/659  5/5     0/0  3/3  ☢  │ │ │ │ │ │   ├── rayon 1.6.1
0/1    0/83     0/0     0/0  0/0  ?  │ │ │ │ │ │   ├── sha-1 0.10.1
0/0    0/0      0/0     0/0  0/0  ?  │ │ │ │ │ │   │   ├── cfg-if 1.0.0
1/1    14/14    0/0     0/0  0/0  ☢  │ │ │ │ │ │   │   ├── cpufeatures 0.2.5
0/0    0/0      0/0     0/0  0/0  🔒 │ │ │ │ │ │   │   └── digest 0.10.6
0/8    4/196    0/0     0/0  0/0  ☢  │ │ │ │ │ │   └── sha2 0.10.6
0/0    0/0      0/0     0/0  0/0  ?  │ │ │ │ │ ├── qstring 0.7.2
0/0    3/3      0/0     0/0  0/0  ☢  │ │ │ │ │ │   └── percent-encoding 2.2.0
0/0    15/15    0/0     0/0  0/0  ☢  │ │ │ │ │ ├── rand 0.7.3
0/0    0/0      0/0     0/0  0/0  ?  │ │ │ │ │ ├── rand_chacha 0.2.2
0/1    0/1      0/0     0/0  0/0  ?  │ │ │ │ │ ├── rustversion 1.0.11
0/0    5/5      0/0     0/0  0/0  ☢  │ │ │ │ │ ├── serde 1.0.152
0/0    16/16    0/0     0/0  0/0  ☢  │ │ │ │ │ ├── serde_bytes 0.11.8
0/0    0/0      0/0     0/0  0/0  ?  │ │ │ │ │ ├── serde_derive 1.0.152
0/0    4/7      0/0     0/0  0/0  ☢  │ │ │ │ │ ├── serde_json 1.0.91
0/8    4/196    0/0     0/0  0/0  ☢  │ │ │ │ │ ├── sha2 0.10.6
0/0    0/0      0/0     0/0  0/0  🔒 │ │ │ │ │ ├── sha3 0.10.6
0/0    0/0      0/0     0/0  0/0  ?  │ │ │ │ │ ├── solana-frozen-abi 1.14.10
0/0    0/0      0/0     0/0  0/0  ?  │ │ │ │ │ ├── solana-frozen-abi-macro 1.14.10
0/0    0/0      0/0     0/0  0/0  ?  │ │ │ │ │ ├── solana-logger 1.14.10
0/0    0/0      0/0     0/0  0/0  ?  │ │ │ │ │ │   ├── env_logger 0.9.3
2/2    45/45    0/0     0/0  0/0  ☢  │ │ │ │ │ │   │   ├── atty 0.2.14
1/24   10/444   0/2     0/0  5/45 ☢  │ │ │ │ │ │   │   │   └── libc 0.2.139
0/0    0/0      0/0     0/0  0/0  🔒 │ │ │ │ │ │   │   ├── humantime 2.1.0
1/1    16/18    1/1     0/0  0/0  ☢  │ │ │ │ │ │   │   ├── log 0.4.17
0/0    34/34    1/2     0/0  2/2  ☢  │ │ │ │ │ │   │   ├── regex 1.7.1
```

```
0/0     34/34        1/2     0/0    2/2     ☢  │   │   │   │   │   │   │   ├── regex 1.7.1
0/19    33/678       0/0     0/0    1/22    ☢  │   │   │   │   │   │   │   │   ├── aho-corasick 0.7.20
2/37    361/2144     0/0     0/0    4/21    ☢  │   │   │   │   │   │   │   │   │   └── memchr 2.5.0
2/37    361/2144     0/0     0/0    4/21    ☢  │   │   │   │   │   │   │   │   ├── memchr 2.5.0
0/0     0/0          0/0     0/0    0/0     🔒  │   │   │   │   │   │   │   │   └── regex-syntax 0.6.28
0/0     0/0          0/0     0/0    0/0     ❓  │   │   │   │   │   │   │   └── termcolor 1.2.0
0/0     7/7          1/1     0/0    0/0     ☢  │   │   │   │   │   │   ├── lazy_static 1.4.0
1/1     16/18        1/1     0/0    0/0     ☢  │   │   │   │   │   │   └── log 0.4.17
3/3     442/442      1/1     0/0    3/3     ☢  │   │   │   │   │   ├── solana-program 1.14.10
0/0     0/0          0/0     0/0    0/0     ❓  │   │   │   │   │   ├── solana-sdk-macro 1.14.10
0/0     0/0          0/0     0/0    0/0     ❓  │   │   │   │   │   ├── thiserror 1.0.38
1/1     97/97        0/0     0/0    1/1     ☢  │   │   │   │   │   ├── uriparse 0.6.4
0/0     0/0          0/0     0/0    0/0     ❓  │   │   │   │   │   │   ├── fnv 1.0.7
0/0     7/7          1/1     0/0    0/0     ☢  │   │   │   │   │   │   ├── lazy_static 1.4.0
0/0     5/5          0/0     0/0    0/0     ☢  │   │   │   │   │   │   └── serde 1.0.152
12/14   438/502      13/13   2/2    10/10   ☢  │   │   │   │   │   └── wasm-bindgen 0.2.83
0/0     3/3          0/0     0/0    0/0     ☢  │   │   │   │   ├── subtle 2.4.1
0/0     0/0          0/0     0/0    0/0     ❓  │   │   │   │   ├── thiserror 1.0.38
1/1     23/23        0/0     0/0    0/0     ☢  │   │   │   │   └── zeroize 1.3.0
0/0     0/0          0/0     0/0    0/0     ❓  │   │   │   ├── spl-memo 3.0.1
3/3     442/442      1/1     0/0    3/3     ☢  │   │   │   │   └── solana-program 1.14.10
0/0     0/0          0/0     0/0    0/0     ❓  │   │   │   ├── spl-token 3.5.0
0/0     0/0          0/0     0/0    0/0     ❓  │   │   │   └── thiserror 1.0.38
0/0     0/0          0/0     0/0    0/0     ❓  │   │   └── thiserror 1.0.38
0/0     0/0          0/0     0/0    0/0     ❓  │   ├── spl-token 3.5.0
0/0     0/0          0/0     0/0    0/0     ❓  │   └── thiserror 1.0.38
0/0     0/0          0/0     0/0    0/0     ❓  ├── num-derive 0.3.3
0/0     6/12         0/0     0/0    0/0     ☢  ├── num-traits 0.2.15
0/0     0/0          0/0     0/0    0/0     ❓  ├── num_enum 0.5.7
0/0     5/5          0/0     0/0    0/0     ☢  ├── serde 1.0.152
0/0     0/0          0/0     0/0    0/0     ❓  ├── serde_derive 1.0.152
3/3     442/442      1/1     0/0    3/3     ☢  ├── solana-program 1.14.10
0/0     0/0          0/0     0/0    0/0     🔒  ├── spl-math 0.1.0
0/0     7/7          0/0     0/0    0/0     ☢  │   ├── borsh 0.9.3
0/0     0/0          0/0     0/0    0/0     ❓  │   ├── borsh-derive 0.9.3
0/0     0/0          0/0     0/0    0/0     ❓  │   ├── num-derive 0.3.3
0/0     6/12         0/0     0/0    0/0     ☢  │   ├── num-traits 0.2.15
3/3     442/442      1/1     0/0    3/3     ☢  │   ├── solana-program 1.14.10
0/0     0/0          0/0     0/0    0/0     ❓  │   ├── thiserror 1.0.38
0/0     0/0          0/0     0/0    0/0     ❓  │   └── uint 0.9.5
1/1     193/193      0/0     0/0    0/0     ☢  │       ├── byteorder 1.4.3
0/0     0/0          0/0     0/0    0/0     ❓  │       ├── crunchy 0.2.2
0/0     0/0          0/0     0/0    0/0     ❓  │       ├── hex 0.4.3
0/0     5/5          0/0     0/0    0/0     ☢  │       │   └── serde 1.0.152
0/0     0/0          0/0     0/0    0/0     ❓  │       └── static_assertions 1.1.0
0/0     0/0          0/0     0/0    0/0     ❓  ├── spl-token-2022 0.5.0
0/0     0/0          0/0     0/0    0/0     ❓  └── thiserror 1.0.38

108/352      11857/26486  302/351 22/25   352/575

error: Found 18 warnings
```

THANK YOU FOR CHOOSING

HALBORN