Neon Labs - Maintenance

Solana Program Security Audit

Prepared by: **Halborn**

Date of Engagement: **May 31st, 2022 - June 6th, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	5/31/2022	Przemyslaw Swiatowiec
0.2	Draft Review	6/10/2022	Gabi Urrutia
1.0	Remediation Plan	06/15/2022	Przemyslaw Swiatowiec
1.1	Remediation Plan Review	6/15/2022	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
Przemyslaw Swiatowiec	Halborn	Przemyslaw.Swiatowiec@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Neon Labs engaged Halborn to conduct a security audit on their governance maintenance program, beginning on May 31st, 2022 and ending on June 6th, 2022 .

Neon EVM is a tool that allows for Ethereum-like transactions to be processed on Solana, taking full advantage of the functionality native to Solana, including parallel execution of transactions. As such, Neon EVM allows dApps to operate with the low gas fees, high transaction speed, and high throughput of Solana, as well as offering access to the growing Solana market.

The security assessment was scoped to the programs provided in the maintenance GitHub repository. Commit hashes and further details can be found in the Scope section of this report.

The Maintenance program provides functionality to delegate the process of program upgrading to other users (delegates). The program can be upgraded with code that was previously white-listed only.

1.2 AUDIT SUMMARY

The team at Halborn was provided 2 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 smart-contract hacking skills, and deep knowledge of multiple blockchain protocols.

The purpose of this audit is to:

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

In summary, Halborn identified improvements to reduce the likelihood and

impact of multiple risks, which has beed successfully addressed by Neon Labs . The main ones were the following:
- validation of program_data account in CloseMaintenance instruction was fixed,
- limit on setting delegates and code hashes were introduced.

1.3 TEST APPROACH & METHODOLOGY

Halborn performed a combination of manual review of the 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 logic, process, 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.
- Solana program manual code review and walkthrough to identify any logic issue.
- 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 cluster simulation (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. Neon Labs

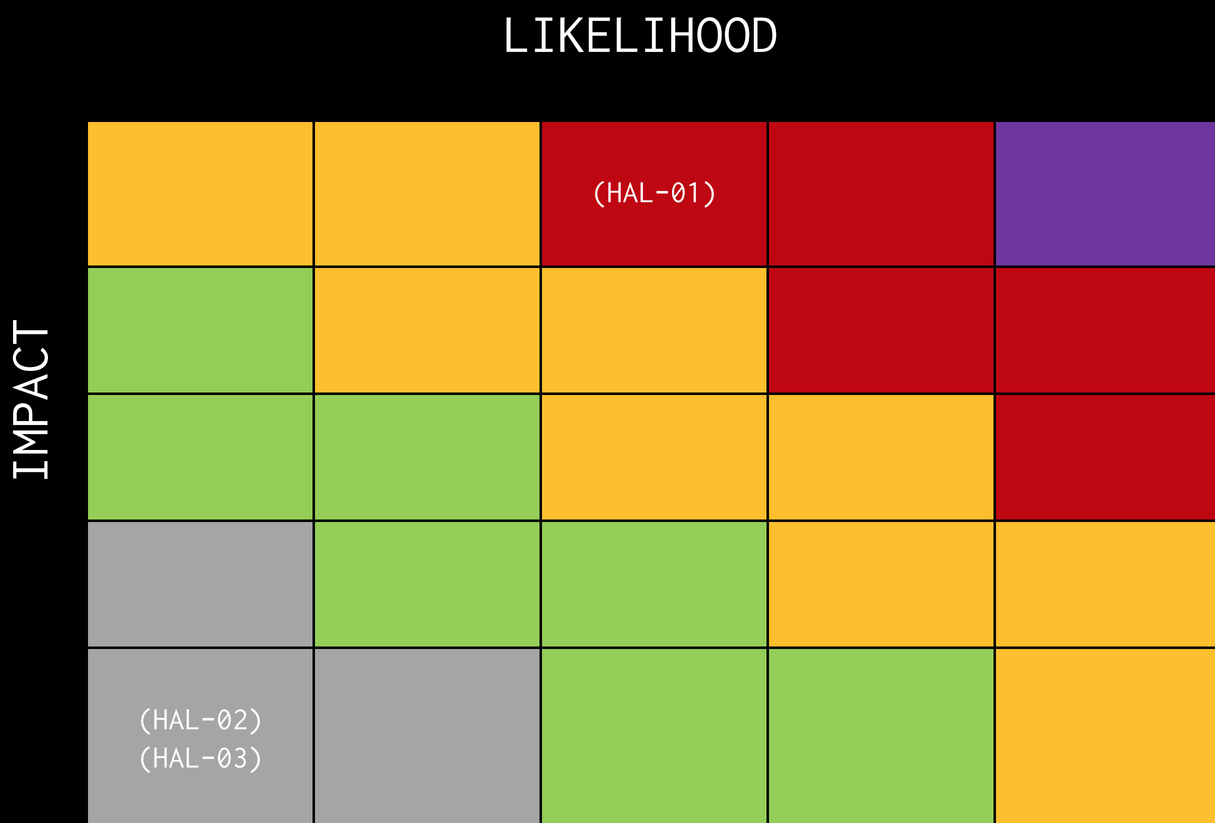
- Repository: maintenance
- Commit ID: d14eb160397fd5eb5baf9865ca1cf57a3e5e52d9
- Programs in scope:

    1. Maintenance (maintenance/program)

Out-of-scope: External libraries and financial related attacks.

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	1	0	0	2



SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
(HAL-01) PROGRAM HIJACK DUE TO POSSIBILITY OF CLOSING ACTIVE MAINTENANCE ACCOUNT	High	SOLVED - 06/13/2022
(HAL-02) NO LIMIT ON SETTING DELEGATES AND CODE HASHES	Informational	SOLVED - 06/13/2022
(HAL-03) INCORRECT ERROR MESSAGE	Informational	SOLVED - 06/13/2022

FINDINGS & TECH DETAILS

3.1 (HAL-01) PROGRAM HIJACK DUE TO POSSIBILITY OF CLOSING ACTIVE MAINTENANCE ACCOUNT - HIGH

Description:

In the CloseMaintenance instruction, program_data is deserialized and used to get the current authority of the maintained program. The maintenance account can close only if it is not set as an authority of the maintained program.

However, as there is no validation for the program_data account in the CloseMaintenance instruction, it's possible to supply program_data for another program, not the one actually maintained. It could result in disposing of an active maintenance account. After disposal, any user could recreate and modify the maintenance account and set custom maintenance authority. It can create a risk of program hijacking, as the recreated maintenance account would still be an authority for the maintained program.

Code Location:

Listing 1: maintenance/program/src/processor.rs (Line 304)

```
304 let maintenance_record_info = next_account_info(account_info_iter)?; // 0
305 let program_data_info = next_account_info(account_info_iter)?; // 1
306 let authority_info = next_account_info(account_info_iter)?; // 2
```

Listing 2: maintenance/program/src/processor.rs (Lines 324,326-335)

```
322
323 let upgradeable_loader_state: UpgradeableLoaderState =
324     bincode::deserialize(&program_data_info.data.borrow()).map_err(|_| ProgramError::from(MaintenanceError::AuthorityDeserializationError) )?;
```

```rust

let program_authority: Pubkey =
    match upgradeable_loader_state {
        UpgradeableLoaderState::ProgramData { slot: _, upgrade_authority_address } =>
            upgrade_authority_address.ok_or(ProgramError::from(MaintenanceError::AuthorityDeserializationError))?,
        _ =>
            return Err(ProgramError::from(MaintenanceError::AuthorityDeserializationError)),
    };
// msg!("MAINTENANCE-INSTRUCTION: CLOSE MAINTENANCE program authority {:?}", program_authority);

if *maintenance_record_info.key == program_authority {
    return Err(MaintenanceError::RecordMustBeInactive.into());
}

```

Risk Level:

Likelihood - 3
Impact - 5

Recommendation:

Consider introducing check for program_data account in CloseMaintenance . Another option could be to modify the CloseMaintenance instruction to derive the program_data account address instead of accepting this account address as a handler parameter.

Remediation Plan:

SOLVED: The issue was fixed by introducing the program_data account validation in the CloseMaintenance instruction. In the fixed version, program_data's PDA is derived in the processing function in commit (3469f98546719ed777703a65329b464029a82927).

3.2 (HAL-02) NO LIMIT ON SETTING DELEGATES AND CODE HASHES - INFORMATIONAL

Description:

The SetDelegates and SetCodeHashes instructions are used to set delegates that can upgrade the maintained program and define the correct hash of updated code.

No limits for code hashes and delegates are introduced in the program code. The number of code hashes and delegates is restricted by account size. Maintenance account can store 20 items. Due to Solana's instruction size limitation, it's possible to supply maximum 18 delegates / code hashes in one transaction. If user supply 18 delegates, maintenance account can store only 2 codes hashes. Going over this limit returns BorshIo error, which is not very verbose.

Code Location:

Listing 3: maintenance/program/src/state.rs (Line 33)

```
33 impl AccountMaxSize for MaintenanceRecord {
34     fn get_max_size(&self) -> Option<usize> {
35         Some(649)    // for delegate size = 10, hashes size = 10
36     }
37 }
```

Listing 4: maintenance/program/src/processor.rs (Lines 120,137,148,165)

```
117 pub fn process_set_delegates(
118     program_id: &Pubkey,
119     accounts: &[AccountInfo],
120     delegate: Vec<Pubkey>,
121 ) -> ProgramResult {
122     let account_info_iter = &mut accounts.iter();
```

```rust

    let maintenance_record_info = next_account_info(account_info_iter)?; // 0
    let authority_info = next_account_info(account_info_iter)?; // 1

    if !authority_info.is_signer {
        return Err(MaintenanceError::MissingRequiredSigner.into());
    }

    let mut maintenance_record = get_account_data::<MaintenanceRecord>(program_id, maintenance_record_info)?;

    if *authority_info.key != maintenance_record.authority {
        return Err(MaintenanceError::WrongAuthority.into());
    }

    maintenance_record.delegate = delegate;

    maintenance_record.serialize(&mut *maintenance_record_info.data.borrow_mut())?;

    Ok(())
}

/// Processes Set Code Hashes instruction
pub fn process_set_code_hashes(
    program_id: &Pubkey,
    accounts: &[AccountInfo],
    hashes: Vec<Hash>,
) -> ProgramResult {
    let account_info_iter = &mut accounts.iter();

    let maintenance_record_info = next_account_info(account_info_iter)?; // 0
    let authority_info = next_account_info(account_info_iter)?; // 1

    if !authority_info.is_signer {
        return Err(MaintenanceError::MissingRequiredSigner.into());
    }

```

```rust
    let mut maintenance_record = get_account_data::<MaintenanceRecord>(program_id, maintenance_record_info)?;

    if *authority_info.key != maintenance_record.authority {
        return Err(MaintenanceError::WrongAuthority.into());
    }

    maintenance_record.hashes = hashes;

    maintenance_record.serialize(&mut *maintenance_record_info.data.borrow_mut())?;

    Ok(())
}
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

Consider adding check for maximum amount of delegates and code hashes.

Remediation Plan:

SOLVED: The issue was fixed by restricting the maximum number of allowed delegates and hashes in commit (3469f98546719ed777703a65329b464029a82927).

3.3 (HAL-03) INCORRECT ERROR MESSAGE - INFORMATIONAL

Description:

Incorrect error message was identified in the CloseMaintenance instruction handler. A proper error message should inform that spill accounts cannot be the subjects of closing.

The lack of proper error messages affects the usability of the program, since it is more difficult for users to understand the reason for the error and solve transaction problems.

Code Location:

Listing 5: maintenance/program/src/processor.rs (Line 314)

```
313 if maintenance_record_info.key == spill_info.key {
314     return Err(ProgramError::InvalidAccountData);
315 }
316
```

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to implement more verbose and specific error message.

Remediation Plan:

SOLVED: The issue was fixed by introducing a correct error message in CloseMaintenance instruction in commit (3469f98546719ed777703a65329b464029a82927).

AUTOMATED TESTING

4.1 AUTOMATED ANALYSIS

Description:

Halborn used automated security scanners to assist with detection of well-known security issues and vulnerabilities. Among the tools used was cargo audit, a security scanner for vulnerabilities reported to the RustSec 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.

id	package	categories
RUSTSEC-2020-0036	failure	unmaintained

4.2 AUTOMATED VULNERABILITY SCANNING

Description:

Halborn used automated security scanners to assist with 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 and sent the compiled results to the analyzers to locate any vulnerabilities.

Results:

Two untrustful accounts were found by Soteria scanner.

1. Untrustful upgrade_buffer_info account in Upgrade instruction.

Untrustful Account No.1 (Critical)

The account is missing owner check:

program/src/processor.rs:184

```
 178|
 179|    let bpf_loader_program_info = next_account_info(account_info_iter)?; // 0
 180|    let sysvar_rent_program_info = next_account_info(account_info_iter)?; // 1
 181|    let sysvar_clock_program_info = next_account_info(account_info_iter)?; // 2
 182|    let maintained_program_info = next_account_info(account_info_iter)?; // 3
 183|    let maintained_program_data_info = next_account_info(account_info_iter)?; // 4
>184|    let upgrade_buffer_info = next_account_info(account_info_iter)?; // 5
 185|    let maintenance_record_info = next_account_info(account_info_iter)?; // 6
 186|    let authority_info = next_account_info(account_info_iter)?; // 7
 187|    let spill_info = next_account_info(account_info_iter)?; // 8
 188|
 189|    if !authority_info.is_signer {
 190|        return Err(MaintenanceError::MissingRequiredSigner.into());
```

```
sol.process_upgrade [program/src/processor.rs:60]
```

Detailed Explanation

This finding was marked as false positive, as later in instruction processing flow, this account is validated by bpf_loader_upgradeable upgrade instruction.

2. Untrustful `program_data` account in `CloseMaintenance` instruction.

Untrustful Account No.2 (Critical)

The account is missing owner check:

program/src/processor.rs:305

```
 299|    program_id: &Pubkey,
 300|    accounts: &[AccountInfo],
 301|) -> ProgramResult {
 302|    let account_info_iter = &mut accounts.iter();
 303|
 304|    let maintenance_record_info = next_account_info(account_info_iter)?; // 0
>305|    let program_data_info = next_account_info(account_info_iter)?; // 1
 306|    let authority_info = next_account_info(account_info_iter)?; // 2
 307|    let spill_info = next_account_info(account_info_iter)?; // 3
 308|
 309|    if !authority_info.is_signer {
 310|        return Err(MaintenanceError::MissingRequiredSigner.into());
 311|    }
```

```
sol.process_close_maintenance [program/src/processor.rs:68]
```

Detailed Explanation

The finding was reported with HAL-01 vulnerability in the previous chapter.

4.3 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:

Please see the next 5 pages.

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        0/0          0/0    0/0     0/0      ❓ maintenance 0.1.0
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 1.0.137
0/0        0/0          0/0    0/0     0/0      ❓ │       └── serde_derive 1.0.137
0/0        12/12        0/0    0/0     3/3      ☢️ │           ├── proc-macro2 1.0.39
0/0        4/4          0/0    0/0     0/0      ☢️ │           │   └── unicode-ident 1.0.0
0/0        0/0          0/0    0/0     0/0      ❓ │           ├── quote 1.0.18
0/0        12/12        0/0    0/0     3/3      ☢️ │           │   └── proc-macro2 1.0.39
0/0        50/50        3/3    0/0     2/2      ☢️ │           └── syn 1.0.95
0/0        12/12        0/0    0/0     3/3      ☢️ │               ├── proc-macro2 1.0.39
0/0        0/0          0/0    0/0     0/0      ❓ │               ├── quote 1.0.18
0/0        4/4          0/0    0/0     0/0      ☢️ │               └── unicode-ident 1.0.0
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      ❓ │   │   ├── borsh-derive-internal 0.9.3
0/0        12/12        0/0    0/0     3/3      ☢️ │   │   │   ├── proc-macro2 1.0.39
0/0        0/0          0/0    0/0     0/0      ❓ │   │   │   ├── quote 1.0.18
0/0        50/50        3/3    0/0     2/2      ☢️ │   │   │   └── syn 1.0.95
0/0        0/0          0/0    0/0     0/0      ❓ │   │   ├── borsh-schema-derive-internal 0.9.3
0/0        12/12        0/0    0/0     3/3      ☢️ │   │   │   ├── proc-macro2 1.0.39
0/0        0/0          0/0    0/0     0/0      ❓ │   │   │   ├── quote 1.0.18
0/0        50/50        3/3    0/0     2/2      ☢️ │   │   │   └── syn 1.0.95
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 0.5.9
0/0        5/5          0/0    0/0     0/0      ☢️ │   │   │       └── serde 1.0.137
0/0        12/12        0/0    0/0     3/3      ☢️ │   │   ├── proc-macro2 1.0.39
0/0        50/50        3/3    0/0     2/2      ☢️ │   │   └── syn 1.0.95
2/2        1082/1198    19/22  1/1     51/58    ☢️ │   └── hashbrown 0.11.2
0/0        26/30        0/0    0/0     0/0      ☢️ │       ├── ahash 0.7.6
2/4        56/173       1/1    0/0     3/3      ☢️ │       │   ├── getrandom 0.2.6
0/0        0/0          0/0    0/0     0/0      ❓ │       │   │   ├── cfg-if 1.0.0
0/21       12/368       0/2    0/0     2/40     ☢️ │       │   │   └── libc 0.2.126
1/1        36/117       2/6    0/0     1/3      ☢️ │       │   ├── once_cell 1.12.0
0/16       0/1341       0/0    0/0     0/56     ❓ │       │   │   └── parking_lot_core 0.9.3
0/0        0/0          0/0    0/0     0/0      ❓ │       │   │       ├── cfg-if 1.0.0
0/21       12/368       0/2    0/0     2/40     ☢️ │       │   │       ├── libc 0.2.126
0/1        0/392        0/7    0/1     0/13     ❓ │       │   │       └── smallvec 1.8.0
0/0        5/5          0/0    0/0     0/0      ☢️ │       │   │           └── serde 1.0.137
0/0        5/5          0/0    0/0     0/0      ☢️ │       │   └── serde 1.0.137
4/6        389/1102     3/9    1/1     12/25    ☢️ │       ├── bumpalo 3.9.1
6/6        663/663      5/5    0/0     3/3      ☢️ │       ├── rayon 1.5.3
0/0        451/451      6/6    0/0     6/6      ☢️ │       │   ├── crossbeam-deque 0.8.1
```

AUTOMATED TESTING

```
0/0    451/451   6/6    0/0  6/6    ☢  │       │   ├── crossbeam-deque 0.8.1
0/0    0/0       0/0    0/0  0/0    ?  │       │   │   ├── cfg-if 1.0.0
3/3    418/430   9/9    0/0  26/26  ☢  │       │   │   ├── crossbeam-epoch 0.9.8
0/0    0/0       0/0    0/0  0/0    ?  │       │   │   │   ├── cfg-if 1.0.0
4/4    85/85     14/14  0/0  2/2    ☢  │       │   │   │   ├── crossbeam-utils 0.8.8
0/0    0/0       0/0    0/0  0/0    ?  │       │   │   │   │   ├── cfg-if 1.0.0
0/0    7/7       1/1    0/0  0/0    ☢  │       │   │   │   │   └── lazy_static 1.4.0
0/0    7/7       1/1    0/0  0/0    ☢  │       │   │   │   ├── lazy_static 1.4.0
0/0    0/0       0/0    0/0  0/0    ?  │       │   │   │   ├── memoffset 0.6.5
0/0    18/18     1/1    0/0  0/0    ☢  │       │   │   │   └── scopeguard 1.1.0
4/4    85/85     14/14  0/0  2/2    ☢  │       │   │   └── crossbeam-utils 0.8.8
0/0    0/0       0/0    0/0  0/0    ?  │       │   ├── either 1.6.1
0/0    5/5       0/0    0/0  0/0    ☢  │       │   │   └── serde 1.0.137
5/5    485/488   2/2    0/0  20/20  ☢  │       │   └── rayon-core 1.9.3
2/2    498/507   6/7    0/0  12/14  ☢  │       │       ├── crossbeam-channel 0.5.4
0/0    0/0       0/0    0/0  0/0    ?  │       │       │   ├── cfg-if 1.0.0
4/4    85/85     14/14  0/0  2/2    ☢  │       │       │   └── crossbeam-utils 0.8.8
0/0    451/451   6/6    0/0  6/6    ☢  │       │       ├── crossbeam-deque 0.8.1
4/4    85/85     14/14  0/0  2/2    ☢  │       │       ├── crossbeam-utils 0.8.8
0/0    72/72     0/0    0/0  0/0    ☢  │       │       └── num_cpus 1.13.1
0/21   12/368    0/2    0/0  2/40   ☢  │       │           └── libc 0.2.126
0/0    5/5       0/0    0/0  0/0    ☢  │       └── serde 1.0.137
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 1.0.137
0/0    0/0       0/0    0/0  0/0    ?  ├── num-derive 0.3.3
0/0    12/12     0/0    0/0  3/3    ☢  │   ├── proc-macro2 1.0.39
0/0    0/0       0/0    0/0  0/0    ?  │   ├── quote 1.0.18
0/0    50/50     3/3    0/0  2/2    ☢  │   └── syn 1.0.95
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 1.0.137
0/0    0/0       0/0    0/0  0/0    ?  ├── serde_derive 1.0.137
3/3    423/423   1/1    0/0  2/2    ☢  ├── solana-program 1.10.21
0/0    0/0       0/0    0/0  0/0    🔒 │   ├── base64 0.13.0
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
10/78  71/3973   0/0    0/0  0/0    ☢  │   ├── blake3 1.3.1
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 1.0.137
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    ?  │   │   ├── constant_time_eq 0.1.5
0/0    0/0       0/0    0/0  0/0    🔒 │   │   ├── digest 0.10.3
0/0    16/16     0/0    0/0  0/0    ☢  │   │   │   ├── block-buffer 0.10.2
1/1    292/292   20/20  8/8  5/5    ☢  │   │   │   │   └── generic-array 0.14.5
0/0    5/5       0/0    0/0  0/0    ☢  │   │   │   │       ├── serde 1.0.137
0/0    0/0       0/0    0/0  0/0    🔒 │   │   │   │       └── typenum 1.15.0
0/0    0/0       0/0    0/0  0/0    🔒 │   │   │   ├── crypto-common 0.1.3
1/1    292/292   20/20  8/8  5/5    ☢  │   │   │   │   ├── generic-array 0.14.5
0/0    15/15     0/0    0/0  0/0    ☢  │   │   │   │   ├── rand_core 0.6.3
2/4    56/173    1/1    0/0  3/3    ☢  │   │   │   │   │   ├── getrandom 0.2.6
0/0    5/5       0/0    0/0  0/0    ☢  │   │   │   │   │   └── serde 1.0.137
0/0    0/0       0/0    0/0  0/0    🔒 │   │   │   │   └── typenum 1.15.0
0/0    3/3       0/0    0/0  0/0    ☢  │   │   │   └── subtle 2.4.1
6/6    663/663   5/5    0/0  3/3    ☢  │   │   └── rayon 1.5.3
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
```

AUTOMATED TESTING

```
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
8/8      202/202  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
1/1      292/292  20/20  8/8   5/5    ☢    │   │       │   └── generic-array 0.14.5
0/0      0/0      0/0    0/0   0/0    ?    │   │       ├── cfg-if 1.0.0
0/1      0/14     0/0    0/0   0/0    ?    │   │       ├── cpufeatures 0.2.2
0/0      0/0      0/0    0/0   0/0    🔒   │   │       ├── digest 0.9.0
1/1      292/292  20/20  8/8   5/5    ☢    │   │       │   └── generic-array 0.14.5
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 1.0.137
18/18    370/382  325/326 7/7  0/0    ☢    │   ├── bytemuck 1.9.1
0/0      0/0      0/0    0/0   0/0    ?    │   │   └── bytemuck_derive 1.1.0
0/0      12/12    0/0    0/0   3/3    ☢    │   │       ├── proc-macro2 1.0.39
0/0      0/0      0/0    0/0   0/0    ?    │   │       ├── quote 1.0.18
0/0      50/50    3/3    0/0   2/2    ☢    │   │       └── syn 1.0.95
0/2      0/857    0/0    0/0   0/0    ?    │   ├── curve25519-dalek 3.2.1
0/1      176/193  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
2/4      50/150   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
0/21     12/368   0/2    0/0   2/40   ☢    │   │   │   │   ├── libc 0.2.126
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 1.0.137
0/0      5/5      0/0    0/0   0/0    ☢    │   │   │   └── serde 1.0.137
0/0      5/5      0/0    0/0   0/0    ☢    │   │   ├── serde 1.0.137
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 0.10.3
0/0      0/0      0/0    0/0   0/0    ?    │   │   └── either 1.6.1
0/0      0/72     0/3    0/1   0/3    ?    │   ├── itertools 0.10.3
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
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      292/292  20/20  8/8   5/5    ☢    │   │   │   ├── generic-array 0.14.5
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      292/292  20/20  8/8   5/5    ☢    │   │   │       │   ├── generic-array 0.14.5
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
2/4      50/150   1/1    0/0   3/3    ☢    │   │   │   ├── getrandom 0.1.16
0/21     12/368   0/2    0/0   2/40   ☢    │   │   │   │   ├── libc 0.2.126
1/1      16/18    1/1    0/0   0/0    ☢    │   │   │   │   ├── log 0.4.17
```

```
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
2/2    636/712   0/0    0/0  17/25  ☢️  │   │   │   │   ├── ppv-lite86 0.2.16
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 1.0.137
8/8    202/202   0/0    0/0  0/0    ☢️  │   │   ├── sha2 0.9.9
0/0    0/0       0/0    0/0  0/0    🔒  │   │   └── typenum 1.15.0
1/1    16/18     1/1    0/0  0/0    ☢️  │   ├── log 0.4.17
0/0    0/0       0/0    0/0  0/0    ?   │   ├── 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
0/1    0/1       0/0    0/0  0/0    ?   │   ├── rustversion 1.0.6
0/0    5/5       0/0    0/0  0/0    ☢️  │   ├── serde 1.0.137
0/0    16/16     0/0    0/0  0/0    ☢️  │   ├── serde_bytes 0.11.6
0/0    5/5       0/0    0/0  0/0    ☢️  │   │   └── serde 1.0.137
0/0    0/0       0/0    0/0  0/0    ?   │   ├── serde_derive 1.0.137
8/8    196/196   0/0    0/0  0/0    ☢️  │   ├── sha2 0.10.2
0/0    0/0       0/0    0/0  0/0    ?   │   │   ├── cfg-if 1.0.0
0/1    0/14      0/0    0/0  0/0    ?   │   │   ├── cpufeatures 0.2.2
0/0    0/0       0/0    0/0  0/0    🔒  │   │   └── digest 0.10.3
0/0    0/0       0/0    0/0  0/0    🔒  │   ├── sha3 0.10.1
0/0    0/0       0/0    0/0  0/0    🔒  │   │   ├── digest 0.10.3
0/0    0/0       0/0    0/0  0/0    ?   │   │   └── keccak 0.1.2
0/0    0/0       0/0    0/0  0/0    ?   │   ├── solana-frozen-abi 1.10.21
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    292/292   20/20  8/8  5/5    ☢️  │   │   ├── generic-array 0.14.5
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 1.15.0
0/0    15/15     0/0    0/0  0/0    ☢️  │   │   │   ├── rand_core 0.6.3
0/0    0/0       0/0    0/0  0/0    ?   │   │   │   ├── rand_xoshiro 0.6.0
0/0    15/15     0/0    0/0  0/0    ☢️  │   │   │   │   ├── rand_core 0.6.3
0/0    5/5       0/0    0/0  0/0    ☢️  │   │   │   │   └── serde 1.0.137
6/6    663/663   5/5    0/0  3/3    ☢️  │   │   │   ├── rayon 1.5.3
0/0    5/5       0/0    0/0  0/0    ☢️  │   │   │   ├── serde 1.0.137
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 1.15.0
0/0    0/0       0/0    0/0  0/0    🔒  │   │   │   └── typenum 1.15.0
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    147/282   4/6    0/0  7/7    ☢️  │   │   ├── memmap2 0.5.3
0/21   12/368    0/2    0/0  2/40   ☢️  │   │   │   └── libc 0.2.126
0/0    5/5       0/0    0/0  0/0    ☢️  │   │   ├── serde 1.0.137
0/0    16/16     0/0    0/0  0/0    ☢️  │   │   ├── serde_bytes 0.11.6
0/0    0/0       0/0    0/0  0/0    ?   │   │   ├── serde_derive 1.0.137
8/8    196/196   0/0    0/0  0/0    ☢️  │   │   ├── sha2 0.10.2
0/0    0/0       0/0    0/0  0/0    ?   │   │   ├── solana-frozen-abi-macro 1.10.21
0/0    12/12     0/0    0/0  3/3    ☢️  │   │   │   ├── proc-macro2 1.0.39
```

AUTOMATED TESTING

AUTOMATED TESTING

```
0/0     12/12     0/0     0/0  3/3    ☢  │   │   │   ├── proc-macro2 1.0.39
0/0     0/0       0/0     0/0  0/0    ?  │   │   │   ├── quote 1.0.18
0/0     50/50     3/3     0/0  2/2    ☢  │   │   │   └── syn 1.0.95
0/0     0/0       0/0     0/0  0/0    ?  │   │   └── thiserror 1.0.31
0/0     0/0       0/0     0/0  0/0    ?  │   │       └── thiserror-impl 1.0.31
0/0     12/12     0/0     0/0  3/3    ☢  │   │           ├── proc-macro2 1.0.39
0/0     0/0       0/0     0/0  0/0    ?  │   │           ├── quote 1.0.18
0/0     50/50     3/3     0/0  2/2    ☢  │   │           └── syn 1.0.95
0/0     0/0       0/0     0/0  0/0    ?  │   ├── solana-frozen-abi-macro 1.10.21
0/0     0/0       0/0     0/0  0/0    ?  │   ├── solana-sdk-macro 1.10.21
0/0     1/1       0/0     0/0  0/0    ☢  │   │   ├── bs58 0.4.0
0/0     12/12     0/0     0/0  3/3    ☢  │   │   ├── proc-macro2 1.0.39
0/0     0/0       0/0     0/0  0/0    ?  │   │   ├── quote 1.0.18
0/1     0/1       0/0     0/0  0/0    ?  │   │   ├── rustversion 1.0.6
0/0     50/50     3/3     0/0  2/2    ☢  │   │   └── syn 1.0.95
0/0     0/0       0/0     0/0  0/0    ?  │   ├── thiserror 1.0.31
12/14   432/496   16/16   2/2  9/9    ☢  │   └── wasm-bindgen 0.2.80
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 1.0.137
0/1     0/0       0/1     0/0  0/1    ?  │       └── wasm-bindgen-macro 0.2.80
0/0     0/0       0/0     0/0  0/0    ?  │           ├── quote 1.0.18
0/0     0/0       0/0     0/0  0/0    ?  │           └── wasm-bindgen-macro-support 0.2.80
0/0     12/12     0/0     0/0  3/3    ☢  │               ├── proc-macro2 1.0.39
0/0     0/0       0/0     0/0  0/0    ?  │               ├── quote 1.0.18
0/0     50/50     3/3     0/0  2/2    ☢  │               ├── syn 1.0.95
0/0     0/0       0/0     0/0  0/0    ?  │               ├── wasm-bindgen-backend 0.2.80
4/6     389/1102  3/9     1/1  12/25  ☢  │               │   ├── bumpalo 3.9.1
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     12/12     0/0     0/0  3/3    ☢  │               │   ├── proc-macro2 1.0.39
0/0     0/0       0/0     0/0  0/0    ?  │               │   ├── quote 1.0.18
0/0     50/50     3/3     0/0  2/2    ☢  │               │   ├── syn 1.0.95
0/0     0/0       0/0     0/0  0/0    ?  │               │   └── wasm-bindgen-shared 0.2.80
0/0     0/0       0/0     0/0  0/0    ?  │               └── wasm-bindgen-shared 0.2.80
0/0     0/0       0/0     0/0  0/0    ?  ├── spl-governance-tools 0.1.2
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     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
0/0     5/5       0/0     0/0  0/0    ☢  │   ├── serde 1.0.137
0/0     0/0       0/0     0/0  0/0    ?  │   ├── serde_derive 1.0.137
3/3     423/423   1/1     0/0  2/2    ☢  │   ├── solana-program 1.10.21
0/0     0/0       0/0     0/0  0/0    ?  │   ├── spl-token 3.3.0
0/0     0/0       0/0     0/0  0/0    ?  │   │   ├── arrayref 0.3.6
18/18   370/382   325/326 7/7   0/0    ☢  │   │   ├── bytemuck 1.9.1
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 1.1.3
0/0     0/0       0/0     0/0  0/0    ?  │   │   │       │   ├── thiserror 1.0.31
0/0     0/0       0/0     0/0  0/0    🔒 │   │   │       │   └── toml 0.5.9
0/0     12/12     0/0     0/0  3/3    ☢  │   │   │       ├── proc-macro2 1.0.39
0/0     0/0       0/0     0/0  0/0    ?  │   │   │       ├── quote 1.0.18
0/0     50/50     3/3     0/0  2/2    ☢  │   │   │       └── syn 1.0.95
```

THANK YOU FOR CHOOSING

HALBORN