



Magic Eden – Open Creator Protocol Solana Program Security Audit

Prepared by: Halborn

Date of Engagement: November 23rd, 2022 – December 7th, 2022

Visit: Halborn.com

DOCUMENT REVISION HISTORY	4
CONTACTS	5
1 EXECUTIVE OVERVIEW	6
1.1 INTRODUCTION	7
1.2 AUDIT SUMMARY	7
1.3 TEST APPROACH & METHODOLOGY	8
RISK METHODOLOGY	9
1.4 SCOPE	11
2 ASSESSMENT SUMMARY & FINDINGS OVERVIEW	12
3 FINDINGS & TECH DETAILS	13
3.1 (HAL-01) POLICY VALIDATION PANIC - LOW	15
Description	15
Code Location	15
Risk Level	15
Recommendation	15
Remediation Plan	16
3.2 (HAL-02) HARDCODED MANAGEMENT ADDRESS - LOW	17
Description	17
Code Location	17
Risk Level	18
Recommendation	18
Remediation Plan	18
3.3 (HAL-03) INTEGER UNDERFLOW - INFORMATIONAL	19
Description	19

Code Location	19
Risk Level	19
Recommendation	20
Remediation Plan	20
3.4 (HAL-04) LACK OF TEST COVERAGE - INFORMATIONAL	21
Description	21
Code Location	21
Risk Level	21
Recommendation	21
Remediation Plan	21
3.5 (HAL-05) POSSIBLE RUST PANICS DUE TO UNSAFE UNWRAP USAGE - INFORMATIONAL	22
Description	22
Code Location	22
Risk Level	24
Recommendation	24
Remediation Plan	24
3.6 (HAL-06) MISSING CARGO OVERFLOW CHECKS - INFORMATIONAL	26
Description	26
Code Location	26
Risk Level	26
Recommendation	26
Remediation Plan	26
4 MANUAL TESTING	27
4.1 POLICY BYPASS	28
Description	28

Results	28
4.2 TOKEN LOCKED DUE TO UNEXPECTED BEHAVIOR	29
Description	29
Results	29
4.3 INTENDED FLOW BYPASS	30
Description	30
Results	30
5 AUTOMATED TESTING	32
5.1 AUTOMATED VULNERABILITY SCANNING	33
Description	33
Results	33
5.2 AUTOMATED ANALYSIS	34
Description	34
Results	34
5.3 UNSAFE RUST CODE DETECTION	35
Description	35
Results	36

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	12/01/2022	Guillermo Alvarez
0.2	Document Updates	12/06/2022	Guillermo Alvarez
0.3	Final Draft	12/07/2022	Guillermo Alvarez
0.4	Draft Review	12/06/2022	Isabel Burrueto
0.5	Draft Review	12/07/2022	Piotr Cielas
0.6	Draft Review	12/07/2022	Gabi Urrutia
1.0	Remediation Plan	12/31/2022	Guillermo Alvarez
1.1	Remediation Plan Review	02/01/2023	Isabel Burrueto
1.2	Remediation Plan Review	02/01/2023	Piotr Cielas
1.3	Remediation Plan Review	02/01/2023	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 Burrueto	Halborn	Isabel.Burrueto@halborn.com
Guillermo Alvarez	Halborn	Guillermo.Alvarez@halborn.com
Luca Distefano	Halborn	Luca.Distefano@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Open Creator Protocol (OCP) is an open protocol for creators to build utilities and the policy engine for their tokens. The policy engine can be configured to support the following features:

- Protected royalties: the protocol allows creators to ban market-places that have not protected royalties on their collection.
- Freeze trading until mint is done: Creators can use Open Creator Protocol to limit trading until after mint is complete
- Create rules for NFT transfers: Creators can gamify transferability for collections, including completely non-transferable tokens.

Magic Eden engaged [Halborn](#) to conduct a security audit on their Solana programs beginning on November 23rd, 2022 and ending on December 7th, 2022 . The security assessment was scoped to the programs provided in the [Open Creator Protocol](#) GitHub repository. Commit hashes and further details can be found in the Scope section of this report.

1.2 AUDIT SUMMARY

The team at Halborn was provided 2 weeks for the engagement and assigned 1 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:

- Identify potential security issues within the programs

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

- The Policy Managed Authority has full control over all OCP policies, even those that are created and managed by another users. In case this address is compromised, all defined policies can be modified by an attacker. It is recommended to either remove the managed authority or add multisig support.
- Invalid or malformed policies lead to program crashes. Proper error handling and propagation should be in place to ensure that state is not affected by panics.

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

EXECUTIVE OVERVIEW

9 - 8 - HIGH
7 - 6 - MEDIUM
5 - 4 - LOW
3 - 1 - VERY LOW AND INFORMATIONAL

1.4 SCOPE

Code repositories:

1. Open Creator Protocol

- Repository: [Open Creator Protocol](#)
- Commit ID: [0ad875da0b7081bb88f5c9f897d7bed561bfc1bb](#)
- Programs in scope:
 1. ([programs/open_creator_protocol/](#))

Out-of-scope:

- Community Managed Token (CMT) Program
- Dynamic Royalties Feature
- Third-party libraries and dependencies
- Financial-related attacks

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	0	0	2	4

LIKELIHOOD

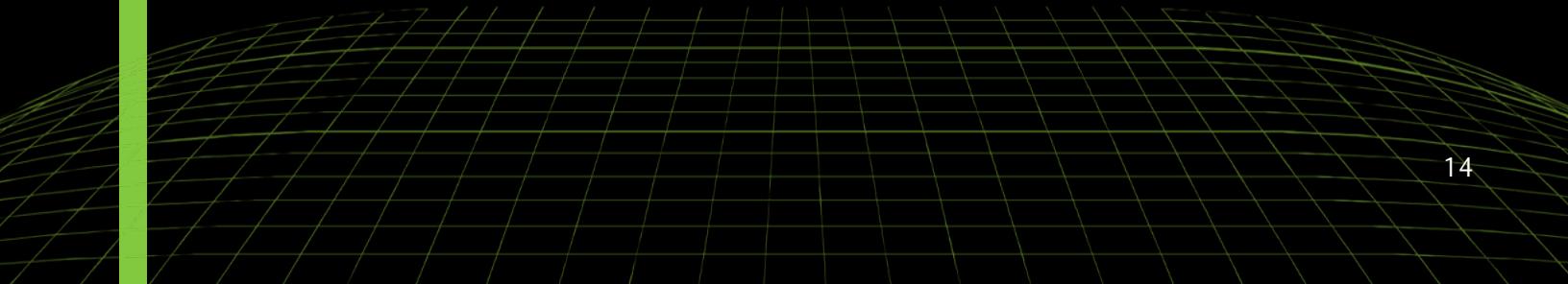


EXECUTIVE OVERVIEW

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL-01 - POLICY VALIDATION PANIC	Low	SOLVED - 11/29/2022
HAL-02 - HARDCODED MANAGEMENT ADDRESS	Low	FUTURE RELEASE
HAL-03 - INTEGER UNDERFLOW	Informational	ACKNOWLEDGED
HAL-04 - LACK OF TEST COVERAGE	Informational	SOLVED - 12/03/2022
HAL-05 - POSSIBLE RUST PANICS DUE TO UNSAFE UNWRAP USAGE	Informational	ACKNOWLEDGED
HAL-06 - MISSING CARGO OVERFLOW CHECKS	Informational	SOLVED - 12/30/2022



FINDINGS & TECH DETAILS



3.1 (HAL-01) POLICY VALIDATION PANIC - LOW

Description:

OCP policies are JSON objects that are validated with the `Policy::valid()` function against a `json_rules_engine_fork::rule` rule. Since the function returns either a `Rule` or an `Error`, submitting invalid JSON policies causes a panic when the program tries to unwrap an Error.

Code Location:

Listing 1: `programs/open_creator_protocol/src/state.rs` (Line 55)

```
50 pub fn valid(&self) -> Result<()> {
51     if self.json_rule.len() > Policy::JSON_RULE_MAX_LEN {
52         return Err(OCPErrorCode::InvalidPolicyCreation.into());
53     }
54     // make sure the rule is valid
55     serde_json::from_str::<Rule>(&self.json_rule).unwrap();
56     Ok(())
57 }
```

Risk Level:

Likelihood - 3

Impact - 2

Recommendation:

It is recommended not to use the `unwrap` function in the production environment because its use causes `panic!` and may crash the program without detailed error messages. Crashing the system will result in a loss of availability and, in some cases, even loss of private information stored in the state. Some alternatives are possible, like propagating the error with `?` instead of unwrapping it, or using the `error-chain` crate

FINDINGS & TECH DETAILS

for errors.

Remediation Plan:

SOLVED: The Magic Eden team fixed the issue in commit `b77c8b7`. The `unwrap` function that was causing the `panic` was replaced with `expect`

3.2 (HAL-02) HARDCODED MANAGEMENT ADDRESS - LOW

Description:

The `MANAGED_AUTHORITY` address, which can modify any existing policy, even those that were created by a different authority, is hardcoded in `state.rs`. In case this address is compromised, all existing policies could be modified, and the program owner would have to redeploy the program to update it.

Code Location:

Listing 2: programs/open_creator_protocol/src/state.rs (Line 47)

```

44 impl Policy {
45     pub const LEN: usize = Policy::JSON_RULE_MAX_LEN + 200 /* with
↳ padding */;
46     pub const SEED: &'static str = "policy";
47     pub const MANAGED_AUTHORITY: &'static str = "
↳ RULERZZDGsXqd9TeJu5ikLfbXzBFpoDPT8N3FHRhq1T";
48     pub const JSON_RULE_MAX_LEN: usize = 1000;
49     [...]

```

**Listing 3: programs/open_creator_protocol/src/instructions/policy/update_-
policy.rs (Lines 20,25,26)**

```

16 #[account(
17     // only policy.authority or MANAGED_AUTHORITY can update the
↳ policy
18     constraint = (
19         authority.key() == policy.authority ||
20         authority.key().to_string() == Policy::MANAGED_AUTHORITY
21     ) @ OCPErrorCode::InvalidAuthority,
22
23     // only MANAGED_AUTHORITY can set the arg.authority to be
↳ MANAGED_AUTHORITY
24     constraint = (
25         arg.authority.to_string() != Policy::MANAGED_AUTHORITY ||

```

```
26     authority.key().to_string() == Policy::MANAGED_AUTHORITY  
27     ) @ OCPErrorCode::InvalidAuthority,  
28 ]  
29 authority: Signer<'info>,
```

Risk Level:

Likelihood - 1

Impact - 3

Recommendation:

Consider removing the hardcoded **MANAGED_AUTHORITY**, adding multisig support or making administrative addresses modifiable, and implement a function to update these addresses in case they are compromised.

Remediation Plan:

PENDING: The [Magic Eden team](#) stated that the address will be replaced with a multi-sig wallet of squads.so.

3.3 (HAL-03) INTEGER UNDERFLOW - INFORMATIONAL

Description:

An underflow or overflow happens when an arithmetic operation attempts to create a numeric value that is outside the range that can be represented by a given number of bits. If not, Rust specifies that the resulting value wrap as two's complement, resulting in a value either too low or too high under the circumstances.

Code Location:

Listing 4: programs/open_creator_protocol/src/action_ctx.rs (Lines 207, 208)

```
198     MintStateCtx {  
199         version: mint_state.version,  
200         policy: mint_state.policy.to_string(),  
201         locked_by: mint_state.locked_by.map(|x| x.to_string()),  
202         last_approved_at: mint_state.last_approved_at,  
203         last_transferred_at: mint_state.last_transferred_at,  
204         transferred_count: mint_state.transferred_count,  
205  
206         derived_cooldown: min(  
207             max(0, now - mint_state.last_approved_at),  
208             max(0, now - mint_state.last_transferred_at),  
209         ),  
210         derived_datetime: now.into(),  
211     }  
212 }
```

Risk Level:

Likelihood - 1

Impact - 1

Recommendation:

Consider using `checked_sub` or `saturating_sub` arithmetic operations instead of regular arithmetic operators to handle overflows gracefully.

Remediation Plan:

ACKNOWLEDGED: The Magic Eden team added a clamp function to `derived_cooldown` in commit [d797a93](#) and reviewed and acknowledged this finding indicates that if it underflows the value will be 0.

3.4 (HAL-04) LACK OF TEST COVERAGE - INFORMATIONAL

Description:

Verifying the code using automated tests (unit tests or functional tests) is a good practice to ensure that all lines of code are working correctly. It was observed that only a few tests were defined to initialize and update a policy, but there were no tests for the NFT proxy instructions set.

Code Location:

- tests/

Risk Level:

Likelihood - 1

Impact - 1

Recommendation:

It is recommended to implement as many test cases as possible to cover all scenarios in the program flow.

Remediation Plan:

SOLVED: The [Magic Eden team](#) fixed the issue in commit [3aeabbf](#) by adding tests for all entry points.

3.5 (HAL-05) POSSIBLE RUST PANICS DUE TO UNSAFE UNWRAP USAGE - INFORMATIONAL

Description:

The use of helper methods in Rust, such as `unwrap`, is allowed in the development and test environment because those methods are supposed to throw an error (also known as `panic!`) when `Option::None` is called or a `Result` that is not `Ok`. However, keeping `unwrap` functions in the production environment is considered bad practice because they can cause program crashes, often accompanied by insufficient or misleading error messages.

Code Location:

Listing 5: `programs/open_creator_protocol/src/state.rs` (Lines 26,30)

```

21 impl MintState {
22     pub const LEN: usize = 200;
23     pub const SEED: &'static str = "mint_state";
24
25     pub fn record_transfer(&mut self) {
26         self.last_transferred_at = Clock::get().unwrap().
↳ unix_timestamp;
27         self.transferred_count = self.transferred_count.
↳ checked_add(1).unwrap_or(u32::MAX);
28     }
29     pub fn record_approve(&mut self) {
30         self.last_approved_at = Clock::get().unwrap().
↳ unix_timestamp;
31     }
32 }
```

Listing 6: `programs/open_creator_protocol/src/state.rs` (Line 55)

```

50 pub fn valid(&self) -> Result<()> {
51     if self.json_rule.len() > Policy::JSON_RULE_MAX_LEN {
```

```

52         return Err(OCPErrorCode::InvalidPolicyCreation.into());
53     }
54     // make sure the rule is valid
55     serde_json::from_str::<Rule>(&self.json_rule).unwrap();
56     Ok(())
57 }
```

Listing 7: programs/open_creator_protocol/src/state.rs (Lines 68,69)

```

63 pub fn matches(&self, ctx: &ActionCtx) -> Result<()> {
64     if self.json_rule.is_empty() {
65         return Ok(());
66     }
67
68     let rule: Rule = serde_json::from_str::<Rule>(&self.json_rule)
69     .unwrap();
70     let fact: &Value = &serde_json::to_value::<&ActionCtx>(ctx).
71     unwrap();
72     let result = rule.check_value(fact);
73     if result.condition_result.status != Status::Met {
74         msg!("Policy does not match: {}", result.condition_result.
75             name);
76         msg!("fact: {}", fact);
77         msg!("json_rule: {}", self.json_rule);
78         return Err(OCPErrorCode::InvalidPolicyEvaluation.into());
79     }
80     Ok(())
81 }
```

Listing 8: programs/open_creator_protocol/src/instructions/nft_proxy/wrap.rs (Line 78)

```

72 pub fn handler<'info>(ctx: Context<'_, '_ , '_ , 'info , WrapCtx<'_
73     info>>) -> Result<()> {
74     let action_ctx: ActionCtx = ctx.accounts.into();
75     let policy = &ctx.accounts.policy;
76     ctx.accounts.policy.matches(&action_ctx)?;
77
78     let mint_state = &mut ctx.accounts.mint_state;
79     mint_state.bump = [*ctx.bumps.get("mint_state").unwrap()];
80     mint_state.policy = policy.key();
81     mint_state.mint = ctx.accounts.mint.key();
```

```
81     mint_state.version = 0;
```

Listing 9: programs/open_creator_protocol/src/instructions/policy/init_policy.rs (Line 29)

```
26 pub fn handler(ctx: Context<InitPolicyCtx>, arg: InitPolicyArg) ->
27     Result<()> {
28     let policy = &mut ctx.accounts.policy;
29     policy.bump = [*ctx.bumps.get("policy").unwrap()];
30     policy.uuid = arg.uuid;
31     policy.authority = ctx.accounts.authority.key();
32     policy.json_rule = arg.json_rule;
33     policy.valid()
34 }
35
```

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 detailed error messages. Crashing the system will result in a loss of availability and, in some cases, even loss of private information stored in the state. Some alternatives are possible, such as propagating the error with `?` instead of unwrapping it, or using the `error-chain` crate for errors.

Remediation Plan:

ACKNOWLEDGED: The Magic Eden team reviewed and fixed the issues associated with the policy in commit `b77c8b7` and acknowledged the remaining ones stating that it is supposed to panic and stop the program execution

FINDINGS & TECH DETAILS

immediately.

3.6 (HAL-06) MISSING CARGO OVERFLOW CHECKS - INFORMATIONAL

Description:

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

Code Location:

- `programs/open_creator_protocol/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:

SOLVED: The Magic Eden team fixed the issue in commit `20b0d15`. Release profile updated to include `overflow-checks=true`.

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 POLICY BYPASS

Description:

In `open_creator_protocol`, `policy` accounts are used to enforce specific rules when an action is performed. This is a key feature of the OCP; thus, multiple tests were performed to ensure the following:

- Only the authority or `MANAGED_AUTHORITY` can update the policy.
- The `policy` is validated correctly.
- Policies are enforced, and you cannot perform any unintended actions, such as transfer a token if a policy denies it.
- The policy validation process of an action cannot be bypassed.

Results:

Some observations about policy validation are documented as `HAL-02` in the previous section of the report.

```
2022-12-07T16:36:53.480Z000127 INFO solana_runtime::bank] accounts hash slot: 0 stats: BankHashStats { num_updated_accounts: 15, num_removed_accounts: 0, num_lamports_stored: 400000541362095, total_data_len: 761744, num_executable_accounts: 31 }
[2022-12-07T16:36:53.48104080127 DEBUG solana_runtime::accounts_db] no recycle stores max: 0 min: 1844674407379951615 len: 0 locking: 28744, 1844674407379951615 avail: 0
[2022-12-07T16:36:53.48104080127 DEBUG solana_runtime::accounts_db] lock: 0 unlock: 0
[2022-12-07T16:36:53.4837736730612 DEBUG solana_runtime::bank::executor_cache] Executor Cache Stats - Hit: 9, Misses: 0, Evictions: 0, Insertions: 0, Deletions: 0, One-Hit-Wonders: 0
[2022-12-07T16:36:53.4847229112 DEBUG solana_runtime::bank] processing transactions:
[2022-12-07T16:36:53.4850443709312 DEBUG solana_runtime::message_processor::stable_log] Program cokpWUdA2zXWV3QHwAdyoyz0wJ3hRdvTvbo9E invoke []
[2022-12-07T16:36:53.4850443709312 DEBUG solana_runtime::message_processor::stable_log] Program cokpWUdA2zXWV3QHwAdyoyz0wJ3hRdvTvbo9E invoke []
[2022-12-07T16:36:53.4851729192132 DEBUG solana_runtime::message_processor::stable_log] Program log: Instruction: UpdatePolicy
[2022-12-07T16:36:53.4851729192132 DEBUG solana_runtime::message_processor::stable_log] Program log: Instruction: Success
[2022-12-07T16:36:53.485674371132 DEBUG solana_runtime::message_processor::stable_log] Program cokpWUdA2zXWV3QHwAdyoyz0wJ3hRdvTvbo9E success
[2022-12-07T16:36:53.485674371132 DEBUG solana_runtime::message_processor::stable_log] Program cokpWUdA2zXWV3QHwAdyoyz0wJ3hRdvTvbo9E success
[2022-12-07T16:36:53.4860217332 DEBUG solana_runtime::bank] checkSbus load: 144us execute: 757us txs:1 len:1
[2022-12-07T16:36:53.4860217332 DEBUG solana_runtime::bank] bank unlock accounts
[2022-12-07T16:36:53.4860626732 DEBUG solana_runtime::counts] bank unlock accounts
running 0 tests
test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

Payer pubkey: 2yWhr4c58puUpdw4G0RmCnSu9p7t8tE6vYHVUAhEYB
Active pubkey: 3M6oL51B5f0013gqNj000L5gPvMqM3c1kkKKRq
UUID: 4d4eV1tcvIq7yW0K0svkxE01Zhnxuaas1wCj2LwM
PDA 8Zm2Abay8BqN8zXyQ0EBx19fkj3SBHdMANjre7rf1
JSON rule: {"conditions": {"and": [{"field": "action", "operator": "string_not_equal", "value": "test"}, {"events": []}]}}

policy {
    version: 0,
    bump: 252,
    ...
    rules: [
        {"rule_id": 4d4eV1tcvIq7yW0K0svkxE01Zhnxuaas1wCj2LwM, "authority": 9m0mOna12q4p0Gm00BLsLa3FvWNTW3c1kkKKRq, "json_rule": {"conditions": {"and": [{"field": "action", "operator": "string_not_equal", "value": "test"}, {"events": []}]}}, "update_policy_bypass": ...}
    ]
}
test update_policy_bypass ... ok
test result: ok. 1 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.12s

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

4.2 TOKEN LOCKED DUE TO UNEXPECTED BEHAVIOR

Description:

Several edge cases were simulated to exclude all possibilities of users or the protocol being locked out of their funds due to unexpected program behavior.

Results:

No code vulnerabilities identified.

```

Program ComputeBudget11111111111111111111111111111111 invoke [1]
Program ComputeBudget11111111111111111111111111111111 success
Program ocp4vWUzA2z2XMYJ3QhM9vWdyoyoQwAFJhRdVTbvo9E invoke [1]
Program log: Instruction: Unlock
Program ocp4vWUzA2z2XMYJ3QhM9vWdyoyoQwAFJhRdVTbvo9E consumed 210102 of 1400000 compute units
Program ocp4vWUzA2z2XMYJ3QhM9vWdyoyoQwAFJhRdVTbvo9E success
Program ocp4vWUzA2z2XMYJ3QhM9vWdyoyoQwAFJhRdVTbvo9E invoke [1]
Program log: Instruction: InitAccount
Program CMTQqjzH6Anr9XcPvt73EfdtjWkJPzH7H6DtvhHcyzV invoke [2]
Program log: ManagedTokenInstruction::InitializeAccount
Program ATOKENGPvbdGVxr1b2hvZbsiqW5xWH25efTNsLJA8knL invoke [3]
Program log: Create
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA invoke [4]
Program log: Instruction: GetAccountDataSize
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA consumed 1622 of 947228 compute units
Program return: TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA pQAAAAAAA=
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA success
Program 11111111111111111111111111111111 invoke [4]
Program 11111111111111111111111111111111 success
Program log: Initialize the associated token account
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA invoke [4]
Program log: Instruction: InitializeImmutableOwner
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA consumed 1405 of 940738 compute units
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA success
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA invoke [4]
Program log: Instruction: InitializeAccount3
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA consumed 4241 of 936854 compute units
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA success
Program ATOKENGPvbdGVxr1b2hvZbsiqW5xWH25efTNsLJA8knL consumed 20394 of 952703 compute units
Program ATOKENGPvbdGVxr1b2hvZbsiqW5xWH25efTNsLJA8knL success
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA invoke [3]
Program log: Instruction: FreezeAccount
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA consumed 4265 of 924343 compute units
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA success
Program CMTQqjzH6Anr9XcPvt73EfdtjWkJPzH7H6DtvhHcyzV consumed 40811 of 960305 compute units
Program CMTQqjzH6Anr9XcPvt73EfdtjWkJPzH7H6DtvhHcyzV success
Program ocp4vWUzA2z2XMYJ3QhM9vWdyoyoQwAFJhRdVTbvo9E consumed 272140 of 1189898 compute units
Program ocp4vWUzA2z2XMYJ3QhM9vWdyoyoQwAFJhRdVTbvo9E success
Program ocp4vWUzA2z2XMYJ3QhM9vWdyoyoQwAFJhRdVTbvo9E invoke [1]
Program log: Instruction: Transfer
Program CMTQqjzH6Anr9XcPvt73EfdtjWkJPzH7H6DtvhHcyzV invoke [2]
Program log: ManagedTokenInstruction::Transfer
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA invoke [3]
```

4.3 INTENDED FLOW BYPASS

Description:

The `open_creator_protocol` program provides several instructions allowing specific operations. Several test cases were simulated to ensure that the intended workflow cannot be bypassed or altered unexpectedly. The following were tested:

- The ownership chain is enforced in all instructions.
- Only authorized accounts can perform specific operations on specific accounts.
- The signature requirements are in place.
- Unchecked Accounts are validated by the Community Managed Token program via CPI.

Results:

No code vulnerabilities were identified.

```

Program log: Instruction: Wrap
Program 111111111111111111111111111111111111 invoke [2]
Program 111111111111111111111111111111111111 success
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA invoke [2]
Program log: Instruction: SetAuthority
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA consumed 3167 of 1131024 compute units
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA success
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA invoke [2]
Program log: Instruction: SetAuthority
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA consumed 3007 of 1123758 compute units
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA success
Program ocp4vWUzA2z2XMYJ3QhM9vWdyoyoQwAFJhRdVTbvo9E consumed 246453 of 1365570 compute units
Program ocp4vWUzA2z2XMYJ3QhM9vWdyoyoQwAFJhRdVTbvo9E success
Program ocp4vWUzA2z2XMYJ3QhM9vWdyoyoQwAFJhRdVTbvo9E invoke [1]
Program log: Instruction: InitAccount
Program CMTQqjzH6Anr9XcPVt73EDTjWkJPzH7H6DtvhHcyzV invoke [2]
Program log: ManagedTokenInstruction::InitializeAccount
Program ATokenGPvbdGVxr1b2hvZbsiqW5xWH25efTNsLJA8knL invoke [3]
Program log: Create
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA invoke [4]
Program log: Instruction: GetAccountDataSize
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA consumed 1622 of 821979 compute units
Program return: TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA pQAAAAAAA=
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA success
Program 111111111111111111111111111111111111 invoke [4]
Program 111111111111111111111111111111111111 success
Program log: Initialize the associated token account
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA invoke [4]
Program log: Instruction: InitializeImmutableOwner
Program log: Please upgrade to SPL Token 2022 for immutable owner support
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA consumed 1405 of 815489 compute units
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA success
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA invoke [4]
Program log: Instruction: InitializeAccount3
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA consumed 4241 of 811607 compute units
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA success
Program ATOKENGPvbdGVxr1b2hvZbsiqW5xWH25efTNsLJA8knL consumed 20293 of 827376 compute units
Program ATOKENGPvbdGVxr1b2hvZbsiqW5xWH25efTNsLJA8knL success
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA invoke [3]
Program log: Instruction: FreezeAccount
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA consumed 4265 of 802117 compute units
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA success
Program CMTQqjzH6Anr9XcPVt73EDTjWkJPzH7H6DtvhHcyzV consumed 37574 of 834863 compute units
Program CMTQqjzH6Anr9XcPVt73EDTjWkJPzH7H6DtvhHcyzV success
Program ocp4vWUzA2z2XMYJ3QhM9vWdyoyoQwAFJhRdVTbvo9E consumed 323576 of 1119117 compute units
Program ocp4vWUzA2z2XMYJ3QhM9vWdyoyoQwAFJhRdVTbvo9E success
Program ocp4vWUzA2z2XMYJ3QhM9vWdyoyoQwAFJhRdVTbvo9E invoke [1]
Program log: Instruction: MintTo
Program CMTQqjzH6Anr9XcPVt73EDTjWkJPzH7H6DtvhHcyzV invoke [2]
Program log: ManagedTokenInstruction::MintTo
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA invoke [3]
Program log: Instruction: ThawAccount
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA consumed 4267 of 507219 compute units
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA success
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA invoke [3]
Program log: Instruction: MintTo
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA consumed 4538 of 500126 compute units
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA success
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA invoke [3]
Program log: Instruction: FreezeAccount
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA consumed 4265 of 492750 compute units
Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA success
Program CMTQqjzH6Anr9XcPVt73EDTjWkJPzH7H6DtvhHcyzV consumed 25272 of 513307 compute units
Program CMTQqjzH6Anr9XcPVt73EDTjWkJPzH7H6DtvhHcyzV success
Program ocp4vWUzA2z2XMYJ3QhM9vWdyoyoQwAFJhRdVTbvo9E consumed 309521 of 795541 compute units
Program ocp4vWUzA2z2XMYJ3QhM9vWdyoyoQwAFJhRdVTbvo9E success

```

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:

Soteria reported two issues related to unsafe math, which are already covered in [HAL-03](#).

```
=====This arithmetic operation may be UNSAFE=====
Found a potential vulnerability at line 225, column 24 in programs/mtoken/src/state.rs
The div operation may result in overflows:

219|     policy: mint_state.policy.to_string(),
220|     locked_by: mint_state.locked_by.map(|x| x.to_string()),
221|     last_approved_at: mint_state.last_approved_at,
222|     last_transferred_at: mint_state.last_transferred_at,
223|
224|     derived cooldown: min(
225|         max(0, now - mint_state.last_approved_at),
226|         max(0, now - mint_state.last_transferred_at),
227|     ),
228|     derived_now: now,
229| }
230}
231}

>>>Stack Trace:
>>>mtoken::try_entry::h5af7617a3e70f656 [programs/mtoken/src/lib.rs:12]
>>>mtoken::dispatch::ha512842334ac47be [programs/mtoken/src/lib.rs:12]
>>> mtoken::__private::__global::mint_to::h445f23186e6663e8 [programs/mtoken/src/lib.rs:12]
>>> mtoken::mtoken::mint_to::h46d63d2970e3d45e [programs/mtoken/src/lib.rs:12]
>>> mtoken::instructions::nft_proxy::mint_to::handler::he9e13487f345a64b [programs/mtoken/src/lib.rs:63]
>>> _SLT$T$u20$as$u20$core..convert..IntoSLTSUGT$SGT$::int_to::he2c016b825b15586 [programs/mtoken/src/instructions/nft_proxy/mint_to.rs:68]
>>> mtoken::instructions::nft_proxy::mint_to:::SLTimpl$u20$core..convert..FromSLT$SRF$mut$u20$mtoken..instructions..nft_proxy..mint_to..MintToCtx$GT$u20
$for$u20$Smtoken..state..ActionCtx$GT$::from::hba2552f687bbcd4b [/home/runner/work/bpf-tools/bpf-tools/out/rust/library/core/src/convert/mod.rs:556]
>>> _SLT$T$u20$as$u20$core..convert..IntoSLTSUGT$SGT$::into::h78ef5c645c5de900 [programs/mtoken/src/instructions/nft_proxy/mint_to.rs:62]
>>> _SLT$T$u20$as$u20$core..convert..FromSLT$SGT$::from::h19519253eb978a07 [/home/runner/work/bpf-tools/bpf-tools/out/rust/library/core/src/convert/mod.rs:556]

=====This arithmetic operation may be UNSAFE=====
Found a potential vulnerability at line 226, column 24 in programs/mtoken/src/state.rs
The div operation may result in overflows:

220|     locked_by: mint_state.locked_by.map(|x| x.to_string()),
221|     last_approved_at: mint_state.last_approved_at,
222|     last_transferred_at: mint_state.last_transferred_at,
223|
224|     derived cooldown: min(
225|         max(0, now - mint_state.last_approved_at),
226|         max(0, now - mint_state.last_transferred_at),
227|     ),
228|     derived_now: now,
229| }
230}
231}

>>>Stack Trace:
>>>mtoken::try_entry::h5af7617a3e70f656 [programs/mtoken/src/lib.rs:12]
>>> mtoken::dispatch::ha512842334ac47be [programs/mtoken/src/lib.rs:12]
>>> mtoken::__private::__global::mint_to::h445f23186e6663e8 [programs/mtoken/src/lib.rs:12]
>>> mtoken::mtoken::mint_to::h46d63d2970e3d45e [programs/mtoken/src/lib.rs:12]
>>> mtoken::instructions::nft_proxy::mint_to::handler::he9e13487f345a64b [programs/mtoken/src/lib.rs:63]
>>> _SLT$T$u20$as$u20$core..convert..IntoSLTSUGT$SGT$::int_to::he2c016b825b15586 [programs/mtoken/src/instructions/nft_proxy/mint_to.rs:68]
>>> mtoken::instructions::nft_proxy::mint_to:::SLTimpl$u20$core..convert..FromSLT$SRF$mut$u20$mtoken..instructions..nft_proxy..mint_to..MintToCtx$GT$u20
$for$u20$Smtoken..state..ActionCtx$GT$::from::hba2552f687bbcd4b [/home/runner/work/bpf-tools/bpf-tools/out/rust/library/core/src/convert/mod.rs:556]
>>> _SLT$T$u20$as$u20$core..convert..IntoSLTSUGT$SGT$::into::h78ef5c645c5de900 [programs/mtoken/src/instructions/nft_proxy/mint_to.rs:62]
>>> _SLT$T$u20$as$u20$core..convert..FromSLT$SGT$::from::h19519253eb978a07 [/home/runner/work/bpf-tools/bpf-tools/out/rust/library/core/src/convert/mod.rs:556]

- ✓ [0m:03s] Building Static Happens-Before Graph
- ✓ [0m:00s] Detecting Vulnerabilities
detected 0 untrustful accounts in total.
detected 2 unsafe math operations in total.

-----The summary of potential vulnerabilities in all.ll-----
2 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 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.

Results:

No vulnerabilities were identified.

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:

Symbols:						
	Functions	Expressions	Impls	Traits	Methods	Dependency
	0/0	0/0	0/0	0/0	?	open_creator_protocol 0.1.2
	0/0	0/0	0/0	0/0	?	anchor-lang 0.25.0
	0/0	0/0	0/0	0/0	?	anchor-attribute-access-control 0.25.0
	0/0	8/8	0/0	0/0	?	anchor-syn 0.25.0
15/18	453/460	3/3	0/0	12/12	?	anyhow 1.0.66
0/0	1/1	0/0	0/0	0/0	?	bs58 0.3.1
0/0	0/0	0/0	0/0	0/0	?	heck 0.3.3
0/0	0/0	0/0	0/0	0/0	?	unicode-segmentation 1.10.0
0/0	0/0	0/0	0/0	0/0	?	proc-macro2 1.0.47
0/0	15/15	0/0	0/0	3/3	?	unicode-ident 1.0.5
0/0	15/15	0/0	0/0	0/0	?	proc-macro2-diagnostics 0.9.1
0/0	0/0	0/0	0/0	0/0	?	proc-macro2 1.0.47
0/0	0/0	0/0	0/0	0/0	?	quote 1.0.21
0/0	15/15	0/0	0/0	3/3	?	proc-macro2 1.0.47
0/0	69/69	3/3	0/0	2/2	?	syn 1.0.104
0/0	15/15	0/0	0/0	3/3	?	proc-macro2 1.0.47
0/0	0/0	0/0	0/0	0/0	?	quote 1.0.21
0/0	4/4	0/0	0/0	0/0	?	unicode-ident 1.0.5
3/3	34/34	0/0	0/0	0/0	?	yansi 0.5.1
0/0	0/0	0/0	0/0	0/0	?	quote 1.0.21
0/0	5/5	0/0	0/0	0/0	?	serde 1.0.148
0/0	0/0	0/0	0/0	0/0	?	serde_derive 1.0.148
0/0	15/15	0/0	0/0	3/3	?	proc-macro2 1.0.47
0/0	0/0	0/0	0/0	0/0	?	quote 1.0.21
0/0	69/69	3/3	0/0	2/2	?	syn 1.0.104
0/0	4/7	0/0	0/0	0/0	?	serde.json 1.0.89
0/0	0/46	0/1	0/0	0/0	?	indexmap 1.9.2
0/1	0/1367	0/24	0/1	0/69	?	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
1/21	10/368	0/2	0/0	5/40	?	libc 0.2.137
1/1	76/122	4/8	0/0	2/4	?	once_cell 1.16.0
0/16	0/1323	0/0	0/0	0/56	?	parking_lot_core 0.9.4
0/0	0/0	0/0	0/0	0/0	?	cfg-if 1.0.0
1/21	10/368	0/2	0/0	5/40	?	libc 0.2.137
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 1.0.148
0/0	5/5	0/0	0/0	0/0	?	bumpalo 3.11.1
4/6	437/1158	4/10	1/1	13/26	?	rayon 1.6.0
6/6	659/659	5/5	0/0	3/3	?	crossbeam-deque 0.8.2
0/0	453/453	6/6	0/0	6/6	?	cfg-if 1.0.0
0/0	0/0	0/0	0/0	0/0	?	crossbeam-epoch 0.9.13
3/3	448/460	11/11	0/0	29/29	?	cfg-if 1.0.0
0/0	0/0	0/0	0/0	0/0	?	crossbeam-utils 0.8.14
4/4	94/94	16/16	0/0	3/3	?	memoffset 0.7.1
0/0	0/0	0/0	0/0	0/0	?	scopeguard 1.1.0
0/0	18/18	1/1	0/0	0/0	?	crossbeam-utils 0.8.14
4/4	94/94	16/16	0/0	3/3	?	either 1.8.0
0/0	14/14	0/0	0/0	0/0	?	serde 1.0.148
0/0	5/5	0/0	0/0	0/0	?	rayon-core 1.10.1
7/7	524/527	4/4	0/0	23/23	?	crossbeam-channel 0.5.6
2/2	485/494	6/7	0/0	12/14	?	

2/2	485/494	6/7	0/0	12/14	?
0/0	0/0	0/0	0/0	0/0	crossbeam-channel 0.5.6
4/4	94/94	16/16	0/0	3/3	cfg-if 1.0.0
0/0	453/453	6/6	0/0	6/6	crossbeam-deque 0.8.2
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 1.14.0
1/21	10/368	0/2	0/0	5/40	libc 0.2.137
0/0	5/5	0/0	0/0	0/0	serde 1.0.148
6/6	659/659	5/5	0/0	3/3	rayon 1.6.0
0/0	5/5	0/0	0/0	0/0	serde 1.0.148
0/0	7/7	0/0	0/0	0/0	itoa 1.0.4
7/9	587/723	0/0	0/0	2/2	ryu 1.0.11
0/0	5/5	0/0	0/0	0/0	serde 1.0.148
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	5/5	0/0	0/0	0/0	serde 1.0.148
0/0	0/0	0/0	0/0	0/0	typenum 1.15.0
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 1.3.2
0/0	15/15	0/0	0/0	3/3	proc-macro2 1.0.47
0/0	0/0	0/0	0/0	0/0	quote 1.0.21
0/0	69/69	3/3	0/0	2/2	syn 1.0.104
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 1.0.47
0/0	0/0	0/0	0/0	0/0	quote 1.0.21
0/0	69/69	3/3	0/0	2/2	syn 1.0.104
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	cfg-if 1.0.0
1/1	14/14	0/0	0/0	0/0	cpufeatures 0.2.5
1/21	10/368	0/2	0/0	5/40	libc 0.2.137
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
0/0	69/69	3/3	0/0	2/2	syn 1.0.104
0/0	0/0	0/0	0/0	0/0	thiserror 1.0.37
0/0	0/0	0/0	0/0	0/0	thiserror-impl 1.0.37
0/0	15/15	0/0	0/0	3/3	proc-macro2 1.0.47
0/0	0/0	0/0	0/0	0/0	quote 1.0.21
0/0	69/69	3/3	0/0	2/2	syn 1.0.104
15/18	453/460	3/3	0/0	12/12	anyhow 1.0.66
0/0	15/15	0/0	0/0	3/3	proc-macro2 1.0.47
0/0	0/0	0/0	0/0	0/0	quote 1.0.21
0/0	34/34	1/2	0/0	2/2	regex 1.7.0
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
1/21	10/368	0/2	0/0	5/40	memchr 2.5.0
2/37	361/2144	0/0	0/0	4/21	regex-syntax 0.6.28
0/0	0/0	0/0	0/0	0/0	syn 1.0.104
0/0	69/69	3/3	0/0	2/2	anchor-attribute-account 0.25.0
0/0	0/0	0/0	0/0	0/0	anchor-syn 0.25.0
0/0	8/8	0/0	0/0	0/0	anyhow 1.0.66
15/18	453/460	3/3	0/0	12/12	bs58 0.4.0
0/0	1/1	0/0	0/0	0/0	sha2 0.9.9
0/8	10/202	0/0	0/0	0/0	proc-macro2 1.0.47
0/0	15/15	0/0	0/0	3/3	quote 1.0.21
0/0	0/0	0/0	0/0	0/0	rustversion 1.0.9
0/1	0/1	0/0	0/0	0/0	syn 1.0.104
0/0	69/69	3/3	0/0	2/2	anchor-attribute-constant 0.25.0
0/0	0/0	0/0	0/0	0/0	anchor-syn 0.25.0
0/0	8/8	0/0	0/0	0/0	proc-macro2 1.0.47
0/0	15/15	0/0	0/0	3/3	syn 1.0.104
0/0	69/69	3/3	0/0	2/2	anchor-attribute-error 0.25.0
0/0	0/0	0/0	0/0	0/0	anchor-syn 0.25.0
0/0	8/8	0/0	0/0	0/0	proc-macro2 1.0.47
0/0	15/15	0/0	0/0	3/3	syn 1.0.104

AUTOMATED TESTING

0/0	8/8	0/0	0/0	0/0	?		anchor-syn 0.25.0 proc-macro2 1.0.47 quote 1.0.21 syn 1.0.104
0/0	15/15	0/0	0/0	3/3	?		anchor-attribute-event 0.25.0 anchor-syn 0.25.0 anyhow 1.0.66 proc-macro2 1.0.47 quote 1.0.21 syn 1.0.104
0/0	0/0	0/0	0/0	0/0	?		anchor-attribute-interface 0.25.0 anchor-syn 0.25.0 anyhow 1.0.66 heck 0.3.3 proc-macro2 1.0.47 quote 1.0.21 syn 1.0.104
0/0	69/69	3/3	0/0	2/2	?		anchor-attribute-program 0.25.0 anchor-syn 0.25.0 anyhow 1.0.66 proc-macro2 1.0.47 quote 1.0.21 syn 1.0.104
0/0	0/0	0/0	0/0	0/0	?		anchor-attribute-state 0.25.0 anchor-syn 0.25.0 anyhow 1.0.66 proc-macro2 1.0.47 quote 1.0.21 syn 1.0.104
0/0	8/8	0/0	0/0	0/0	?		anchor-derive-accounts 0.25.0 anchor-syn 0.25.0 anyhow 1.0.66 proc-macro2 1.0.47 quote 1.0.21 syn 1.0.104
15/18	453/460	3/3	0/0	12/12	?		arrayref 0.3.6 base64 0.13.1 bincode 1.3.3 serde 1.0.148
0/0	15/15	0/0	0/0	3/3	?		borsh 0.9.3 borsh-derive 0.9.3 borsh-derive-internal 0.9.3 proc-macro2 1.0.47 quote 1.0.21 syn 1.0.104
0/0	0/0	0/0	0/0	0/0	?		borsh-schema-derive-internal 0.9.3 proc-macro2 1.0.47 quote 1.0.21 syn 1.0.104
0/0	69/69	3/3	0/0	2/2	?		proc-macro-crate 0.1.5 toml 0.5.9 indexmap 1.9.2 serde 1.0.148
0/0	0/0	0/0	0/0	0/0	?		proc-macro2 1.0.47 syn 1.0.104
0/0	7/7	0/0	0/0	0/0	?		hashbrown 0.11.2 ahash 0.7.6 bumpalo 3.11.1 rayon 1.6.0 serde 1.0.148
0/0	0/0	0/0	0/0	0/0	?		bytemuck 1.12.3 bytemuck_derive 1.3.0 proc-macro2 1.0.47 quote 1.0.21 syn 1.0.104
0/0	15/15	0/0	0/0	3/3	?		solana-program 1.10.34 base64 0.13.1
0/0	0/0	0/0	0/0	0/0	?		
2/2	1064/1198	19/22	1/1	51/58	?		
0/0	26/30	0/0	0/0	0/0	?		
4/6	437/1158	4/10	1/1	13/26	?		
6/6	659/659	5/5	0/0	3/3	?		
0/0	5/5	0/0	0/0	0/0	?		
18/18	370/415	339/340	9/9	0/0	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	15/15	0/0	0/0	3/3	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	69/69	3/3	0/0	2/2	?		
3/3	426/426	1/1	0/0	3/3	?		
0/0	0/0	0/0	0/0	0/0	?		

0/0	0/0	0/0	0/0	0/0		
0/0	22/22	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0		
2/78	29/3973	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0		
2/2	350/350	2/2	0/0	7/7	?	
0/0	5/5	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	4/4	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	16/16	0/0	0/0	0/0	?	
1/1	285/285	20/20	8/8	5/5	?	
0/0	0/0	0/0	0/0	0/0	?	
1/1	285/285	20/20	8/8	5/5	?	
0/0	2/2	0/0	0/0	0/0	?	
1/4	49/175	1/1	0/0	3/3	?	
0/0	5/5	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	3/3	0/0	0/0	0/0	?	
6/6	659/659	5/5	0/0	3/3	?	
0/0	7/7	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	1/1	0/0	0/0	0/0	?	
2/2	206/206	0/0	0/0	7/7	?	
0/0	5/5	0/0	0/0	0/0	?	
18/18	370/415	339/340	9/9	0/0	?	
0/2	0/857	0/0	0/0	0/0	?	
1/1	193/193	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	22/22	0/0	0/0	0/0	?	
1/4	47/150	1/1	0/0	3/3	?	
0/0	0/0	0/0	0/0	0/0	?	
1/21	10/368	0/2	0/0	5/40	?	
1/1	16/18	1/1	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	5/5	0/0	0/0	0/0	?	
0/0	5/5	0/0	0/0	0/0	?	
0/0	3/3	0/0	0/0	0/0	?	
1/1	23/23	0/0	0/0	0/0	?	
0/0	0/72	0/3	0/1	0/3	?	
0/0	14/14	0/0	0/0	0/0	?	
0/0	0/72	0/3	0/1	0/3	?	
0/0	7/7	1/1	0/0	0/0	?	
0/0	0/49	0/6	0/0	0/3	?	
0/0	4/4	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
1/1	285/285	20/20	8/8	5/5	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
1/1	285/285	20/20	8/8	5/5	?	
0/0	3/3	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	7/7	1/1	0/0	0/0	?	
0/0	33/33	0/0	0/0	2/2	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	3/3	0/0	0/0	0/0	?	
0/0	15/15	0/0	0/0	0/0	?	
1/4	47/150	1/1	0/0	3/3	?	
1/21	10/368	0/2	0/0	5/40	?	
1/1	16/18	1/1	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	

AUTOMATED TESTING

0/0	0/0	0/0	0/0	0/0	?	
0/2	165/712	0/0	0/0	16/25	?	
0/0	22/22	0/0	0/0	0/0	?	
0/0	22/22	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	22/22	0/0	0/0	0/0	?	
0/0	5/5	0/0	0/0	0/0	?	
0/0	5/5	0/0	0/0	0/0	?	
0/8	10/202	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
1/1	16/18	1/1	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	15/15	0/0	0/0	3/3	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	69/69	3/3	0/0	2/2	?	
0/0	6/12	0/0	0/0	0/0	?	
0/0	15/15	0/0	0/0	0/0	?	
0/1	0/1	0/0	0/0	0/0	?	
0/0	5/5	0/0	0/0	0/0	?	
0/0	16/16	0/0	0/0	0/0	?	
0/0	5/5	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/8	4/196	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
1/1	14/14	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/1	1/2	0/0	0/0	0/0	?	
1/1	14/14	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	1/1	0/0	0/0	0/0	?	
2/2	206/206	0/0	0/0	7/7	?	
1/1	285/285	20/20	8/8	5/5	?	
1/1	122/122	2/2	0/0	4/4	?	
0/0	100/100	0/0	0/0	9/9	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	2/2	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	2/2	0/0	0/0	0/0	?	
0/0	5/5	0/0	0/0	0/0	?	
6/6	659/659	5/5	0/0	3/3	?	
0/0	5/5	0/0	0/0	0/0	?	
0/1	323/643	0/0	0/0	20/39	?	
0/0	100/100	0/0	0/0	9/9	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	7/7	1/1	0/0	0/0	?	
1/1	16/18	1/1	0/0	0/0	?	
0/0	161/293	4/6	0/0	7/7	?	
1/21	19/368	0/2	0/0	5/40	?	
0/0	0/0	0/18	0/2	0/0	?	
0/0	5/5	0/0	0/0	0/0	?	
0/0	16/16	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/8	4/196	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	15/15	0/0	0/0	3/3	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	69/69	3/3	0/0	2/2	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	1/1	0/0	0/0	0/0	?	
0/0	15/15	0/0	0/0	3/3	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	69/69	3/3	0/0	2/2	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/0	0/0	0/0	0/0	0/0	?	
0/1	0/1	0/0	0/0	0/0	?	

0/0	0/0	0/0	0/0	0/0	?		
0/0	0/71	0/0	0/0	0/0	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	0/0	0/0	0/0	0/0	?		
1/21	10/368	0/2	0/0	5/40	?		
0/0	0/79	0/0	0/0	0/0	?		
0/25	0/1834	0/102	0/1	0/69	?		
0/0	0/13	0/2	0/0	0/0	?		
0/0	0/36	0/0	0/0	0/0	?		
0/25	0/1834	0/102	0/1	0/69	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	5/5	0/0	0/0	0/0	?		
1/1	16/18	1/1	0/0	0/0	?		
0/0	0/2	0/0	0/0	0/0	?		
0/0	0/36	0/0	0/0	0/0	?		
1/1	76/122	4/8	0/0	2/4	?		
0/0	0/3	0/0	0/0	0/0	?		
0/0	0/165	0/0	0/0	0/2	?		
0/0	0/5	0/0	0/0	0/0	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	5/5	0/0	0/0	0/0	?		
0/0	4/7	0/0	0/0	0/0	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	0/2	0/0	0/0	0/0	?		
0/0	0/3	0/0	0/0	0/0	?		
0/0	7/7	0/0	0/0	0/0	?		
7/9	587/723	0/0	0/0	2/2	?		
0/0	5/5	0/0	0/0	0/0	?		
0/25	0/1834	0/102	0/1	0/69	?		
0/0	0/13	0/2	0/0	0/0	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	0/51	0/2	0/0	0/1	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	5/5	0/0	0/0	0/0	?		
0/0	0/20	0/0	0/0	0/0	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	5/5	0/0	0/0	0/0	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	0/3	0/0	0/0	0/0	?		
0/0	5/5	0/0	0/0	0/0	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	5/5	0/0	0/0	0/0	?		
0/0	15/15	0/0	0/0	3/3	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	69/69	3/3	0/0	2/2	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	5/5	0/0	0/0	0/0	?		
0/0	4/7	0/0	0/0	0/0	?		
3/3	426/426	1/1	0/0	3/3	?		
0/0	0/0	0/0	0/0	0/0	?		
0/0	0/0	0/0	0/0	0/0	?		
127/560 10023/38424 469/901 21/58 285/861							

THANK YOU FOR CHOOSING
 HALBORN