VRust

Security Assessment

O2Lab VRust Team

11/04/2022 19:59:04

Contents

Summary

This report has been prepared for O2Lab VRust Team to discover issues and vulnerabilities in the source code of the O2Lab VRust Team project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques. The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

Project Summary

Project Name	O2Lab VRust Team
Platform	Ethereum
Language	Solana
Crate	mpl_nft_packs
GitHub Location	https://github.com/parasol-aser/vrust
sha256	Unknown

Audit Summary

Delivery Date	11/04/2022
Audit Methodology	Static Analysis
Key Components	

Vulnerability Summary

Vulnerability Level	Total
Critical	20
Major	0
Medium	0
Minor	0
Informational	0
Discussion	0

Findings

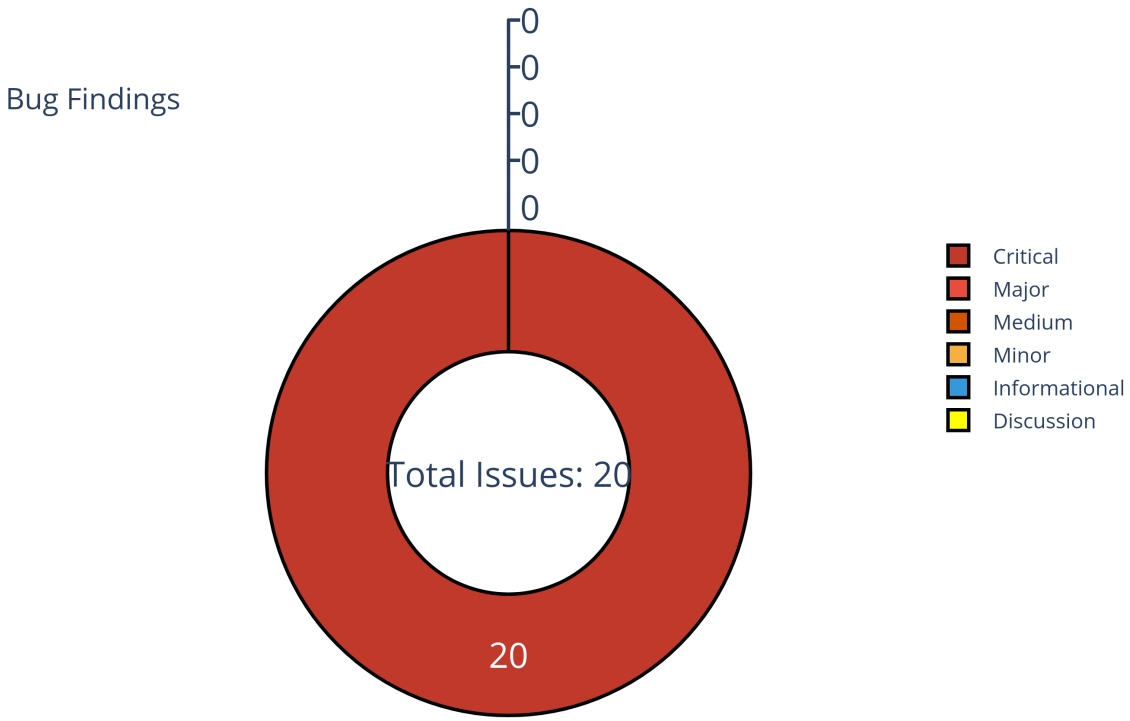

Figure 1: Findings

Finding Statistic

Category	Count
IntegerFlow	2
Precision	1
MissingKeyCheck	14
TypeConfusion	3

ID	Category	Severity	Status
0	IntegerFlow	Critical	UnResolved
1	IntegerFlow	Critical	UnResolved
2	Precision	Critical	UnResolved
3	MissingKeyCheck	Critical	UnResolved
4	MissingKeyCheck	Critical	UnResolved
5	MissingKeyCheck	Critical	UnResolved
6	MissingKeyCheck	Critical	UnResolved
7	MissingKeyCheck	Critical	UnResolved
8	MissingKeyCheck	Critical	UnResolved
9	MissingKeyCheck	Critical	UnResolved
10	MissingKeyCheck	Critical	UnResolved
11	MissingKeyCheck	Critical	UnResolved
12	MissingKeyCheck	Critical	UnResolved
13	MissingKeyCheck	Critical	UnResolved
14	MissingKeyCheck	Critical	UnResolved
15	MissingKeyCheck	Critical	UnResolved
16	MissingKeyCheck	Critical	UnResolved
17	TypeConfusion	Critical	GitHub Link to be added.
18	TypeConfusion	Critical	GitHub Link to be added.

ID	Category	Severity	Status
19	TypeConfusion	Critical	GitHub Link to be added.

Issue: 0: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

nft-packs/program/src/state/pack_config.rs:121:24: 121:53

```
121 rand as f64 / u16::MAX as f64
122
```

- Code Context

Vulnerability at Line: 121

```
116         let selected = self.weights.last().unwrap();
117         let mut bound = if weight_sum == 0 {
118             let max = rand / self.weights.len() as u16;
119             rand.clamp(0, max) as u32
120         } else {
121             let rndp = rand as f64 / u16::MAX as f64;
122             (rndp * weight_sum as f64).round().to_u32().unwrap()
123         };
124         for i in self.weights.iter() {
125             bound = match bound.error_sub(i.1) {
126
```

- Call Stack

```
1 fn entrypoint::process_instruction(){//
  ↪  nft-packs/program/src/entrypoint.rs:12:1: 23:2 }
2     fn processor::Processor::process_instruction(){//
      ↪  nft-packs/program/src/processor.rs:42:5: 110:6 }
3         fn processor::request_card_to_redeem::request_card_for_redeem(){//
          ↪  nft-packs/program/src/processor/request_card_to_redeem.rs:32:1:
          ↪  213:2 }
4             fn state::pack_config::PackConfig::select_weighted_random(){//
              ↪  nft-packs/program/src/state/pack_config.rs:111:5: 134:6 }
5
```

- description:
- link:
- alleviation:

Issue: 1: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

nft-packs/program/src/state/pack_set.rs:126:9: 126:68

```
126 self.pack_vouchers += self.pack_vouchers.error_increment()?
127
```

- Code Context

Vulnerability at Line: 126

```
125 pub fn add_pack_voucher(&mut self) -> Result<(), ProgramError> {
126         self.pack_vouchers += self.pack_vouchers.error_increment()?;
127         Ok(())
128     }
129
```

- Call Stack

```
1 fn entrypoint::process_instruction(){//
  ↪  nft-packs/program/src/entrypoint.rs:12:1: 23:2 }
2     fn processor::Processor::process_instruction(){//
      ↪  nft-packs/program/src/processor.rs:42:5: 110:6 }
3         fn processor::add_voucher_to_pack::add_voucher_to_pack(){//
          ↪  nft-packs/program/src/processor/add_voucher_to_pack.rs:27:1:
          ↪  128:2 }
4             fn state::pack_set::PackSet::add_pack_voucher(){//
              ↪  nft-packs/program/src/state/pack_set.rs:125:5: 128:6 }
5
```

- description:
- link:
- alleviation:

Issue: 2: Precision

Category	Severity	Status
Precision	Critical	UnResolved

- Location

```
nft-packs/program/src/state/pack_config.rs
```

- Code Context

```
111 pub fn select_weighted_random(
112         &mut self,
113         rand: u16,
114         weight_sum: u64,
115     ) -> Result<(u32, u32, u32), ProgramError> {
116         let selected = self.weights.last().unwrap();
117         let mut bound = if weight_sum == 0 {
118             let max = rand / self.weights.len() as u16;
119             rand.clamp(0, max) as u32
120         } else {
121             let rndp = rand as f64 / u16::MAX as f64;
122             (rndp * weight_sum as f64).round().to_u32().unwrap()
123         };
124         for i in self.weights.iter() {
125             bound = match bound.error_sub(i.1) {
126                 Ok(num) => num,
127                 Err(_) => 0,
128             };
129             if bound <= 0 {
130                 return Ok(i.clone());
131             }
132         }
133         return Ok(selected.clone());
134     }
135
```

- Call Stack

```
fn entrypoint::process_instruction(){//
↪  nft-packs/program/src/entrypoint.rs:12:1: 23:2 }
    fn processor::Processor::process_instruction(){//
    ↪  nft-packs/program/src/processor.rs:42:5: 110:6 }
        fn processor::request_card_to_redeem::request_card_for_redeem(){//
        ↪  nft-packs/program/src/processor/request_card_to_redeem.rs:32:1:
        ↪  213:2 }
            fn state::pack_config::PackConfig::select_weighted_random(){//
            ↪  nft-packs/program/src/state/pack_config.rs:111:5: 134:6 }

```

- description:
- link:
- alleviation:

Issue: 3: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

nft-packs/program/src/processor/delete_pack_config.rs:21:37: 21:71

```
21 pack_set_account.data.borrow_mut()
22
```

- Code Context

Vulnerability at Line: 21

```
16     let refunder_account = next_account_info(account_info_iter)?;
17     let authority_account = next_account_info(account_info_iter)?;
18
19     assert_signer(&authority_account)?;
20
21     let pack_set = PackSet::unpack(&pack_set_account.data.borrow_mut())?;
22
23     assert_account_key(authority_account, &pack_set.authority)?;
24
25     pack_set.assert_ended()?;
26
```

- Call Stack

```
1  fn entrypoint::process_instruction(){//
   ↪  nft-packs/program/src/entrypoint.rs:12:1: 23:2 }
2      fn processor::Processor::process_instruction(){//
       ↪  nft-packs/program/src/processor.rs:42:5: 110:6 }
3          fn processor::delete_pack_config::delete_pack_config(){//
           ↪  nft-packs/program/src/processor/delete_pack_config.rs:12:1:
           ↪  38:2 }
4
```

- description:

- link:
- alleviation:

Issue: 4: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

nft-packs/program/src/processor/clean_up.rs:22:41: 22:72

```
22 pack_set_info.data.borrow_mut()
23
```

- Code Context

Vulnerability at Line: 22

```
17 pub fn clean_up(program_id: &Pubkey, accounts: &[AccountInfo]) ->
   ↪  ProgramResult {
18     let account_info_iter = &mut accounts.iter();
19     let pack_set_info = next_account_info(account_info_iter)?;
20     let pack_config_info = next_account_info(account_info_iter)?;
21
22     let mut pack_set = PackSet::unpack(&pack_set_info.data.borrow_mut())?;
23
24     if pack_set.pack_state == PackSetState::NotActivated {
25         return Err(NFTPacksError::WrongPackState.into());
26     }
27
```

Other Use Case for Variable: pack_set_info.data.borrow_mut()

```
47             PackSet::pack(pack_set, *pack_set_info.data.borrow_mut())?;
```

- Call Stack

```
1 fn entrypoint::process_instruction(){//
  ↪  nft-packs/program/src/entrypoint.rs:12:1: 23:2 }
2     fn processor::Processor::process_instruction(){//
      ↪  nft-packs/program/src/processor.rs:42:5: 110:6 }
```

```
3         fn processor::clean_up::clean_up(){//
          ↪  nft-packs/program/src/processor/clean_up.rs:17:1: 61:2 }
4
```

- description:
- link:
- alleviation:

Issue: 5: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

nft-packs/program/src/processor/request_card_to_redeem.rs:78:37: 78:67

```
78 pack_set_account.data.borrow()
79
```

- Code Context

Vulnerability at Line: 78

```
73     let store = Store::from_account_info(store_account)?;
74
75     assert_owned_by(edition_data_account, &store.token_metadata_program)?;
76     assert_signer(&user_wallet_account)?;
77
78     let pack_set = PackSet::unpack(&pack_set_account.data.borrow())?;
79     assert_account_key(store_account, &pack_set.store)?;
80     assert_account_key(randomness_oracle_account,
    ↪  &pack_set.random_oracle)?;
81
82     let proving_process_seeds = &[
83
```

- Call Stack

```
1  fn entrypoint::process_instruction(){//
    ↪  nft-packs/program/src/entrypoint.rs:12:1: 23:2 }
2      fn processor::Processor::process_instruction(){//
        ↪  nft-packs/program/src/processor.rs:42:5: 110:6 }
3          fn processor::request_card_to_redeem::request_card_for_redeem(){//
            ↪  nft-packs/program/src/processor/request_card_to_redeem.rs:32:1:
            ↪  213:2 }
4
```

- description:
- link:
- alleviation:

Issue: 6: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

nft-packs/program/src/processor/request_card_to_redeem.rs:249:42: 249:72

```
249 account_info.data.borrow_mut()
250
```

- Code Context

Vulnerability at Line: 249

```
244     pack_set: &Pubkey,
245     signers_seeds: &[&[u8]],
246     bump_seed: u8,
247     rent: &Rent,
248 ) -> Result<ProvingProcess, ProgramError> {
249     let unpack = ProvingProcess::unpack(&account_info.data.borrow_mut());
250
251     let proving_process = match unpack {
252         Ok(data) => Ok(data),
253         Err(_) => {
254
```

Other Use Case for Variable: account_info.data.borrow_mut()

```
271             let mut data = ProvingPro-
                ↪  cess::unpack_unchecked(&account_info.data.borrow_mut())?;
```

- Call Stack

```
1 fn entrypoint::process_instruction(){//
  ↪  nft-packs/program/src/entrypoint.rs:12:1: 23:2 }
2     fn processor::Processor::process_instruction(){//
       ↪  nft-packs/program/src/processor.rs:42:5: 110:6 }
```

```
3         fn processor::request_card_to_redeem::request_card_for_redeem(){//
          ↪  nft-packs/program/src/processor/request_card_to_redeem.rs:32:1:
          ↪  213:2 }
4             fn proces-
               ↪  sor::request_card_to_redeem::get_proving_process_data(){//
               ↪  nft-
               ↪  packs/program/src/processor/request_card_to_redeem.rs:238:1:
               ↪  285:2 }
5
```

- description:
- link:
- alleviation:

Issue: 7: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

nft-packs/program/src/processor/edit_pack.rs:29:41: 29:75

```
29 pack_set_account.data.borrow_mut()
30
```

- Code Context

Vulnerability at Line: 29

```
24     let pack_set_account = next_account_info(account_info_iter)?;
25     let authority_account = next_account_info(account_info_iter)?;
26
27     assert_signer(&authority_account)?;
28
29     let mut pack_set =
       ↪  PackSet::unpack(&pack_set_account.data.borrow_mut())?;
30
31     assert_account_key(authority_account, &pack_set.authority)?;
32
33     pack_set.assert_able_to_edit()?;
34
```

Other Use Case for Variable: pack_set_account.data.borrow_mut()

```
39     PackSet::pack(pack_set, *pack_set_account.data.borrow_mut())?;
```

- Call Stack

```
1  fn entrypoint::process_instruction(){//
   ↪  nft-packs/program/src/entrypoint.rs:12:1: 23:2 }
2      fn processor::Processor::process_instruction(){//
       ↪  nft-packs/program/src/processor.rs:42:5: 110:6 }
```

```
3       fn processor::edit_pack::edit_pack(){//
        ↪  nft-packs/program/src/processor/edit_pack.rs:18:1: 42:2 }
4
```

- description:
- link:
- alleviation:

Issue: 8: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

nft-packs/program/src/processor/delete_pack.rs:20:37: 20:71

```
20 pack_set_account.data.borrow_mut()
21
```

- Code Context

Vulnerability at Line: 20

```
15     let authority_account = next_account_info(account_info_iter)?;
16     let refunder_account = next_account_info(account_info_iter)?;
17
18     assert_signer(&authority_account)?;
19
20     let pack_set = PackSet::unpack(&pack_set_account.data.borrow_mut())?;
21
22     assert_account_key(authority_account, &pack_set.authority)?;
23
24     pack_set.assert_ended()?;
25
```

- Call Stack

```
1 fn entrypoint::process_instruction(){//
  ↪  nft-packs/program/src/entrypoint.rs:12:1: 23:2 }
2     fn processor::Processor::process_instruction(){//
       ↪  nft-packs/program/src/processor.rs:42:5: 110:6 }
3         fn processor::delete_pack::delete_pack(){//
           ↪  nft-packs/program/src/processor/delete_pack.rs:12:1: 33:2 }
4
```

- description:

- link:
- alleviation:

Issue: 9: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

nft-packs/program/src/processor/change_authority.rs:24:41: 24:75

```
24 pack_set_account.data.borrow_mut()
25
```

- Code Context

Vulnerability at Line: 24

```
19     let authority_account = next_account_info(account_info_iter)?;
20     let new_authority_account = next_account_info(account_info_iter)?;
21
22     assert_signer(&authority_account)?;
23
24     let mut pack_set =
       ↪  PackSet::unpack(&pack_set_account.data.borrow_mut())?;
25     assert_account_key(authority_account, &pack_set.authority)?;
26
27     if pack_set.pack_state == PackSetState::Activated {
28         return Err(NFTPacksError::WrongPackState.into());
29
```

Other Use Case for Variable: pack_set_account.data.borrow_mut()

```
33     PackSet::pack(pack_set, *pack_set_account.data.borrow_mut())?;
```

- Call Stack

```
1  fn entrypoint::process_instruction(){//
   ↪  nft-packs/program/src/entrypoint.rs:12:1: 23:2 }
2      fn processor::Processor::process_instruction(){//
       ↪  nft-packs/program/src/processor.rs:42:5: 110:6 }
```

```
3         fn processor::change_authority::transfer_authority(){//
          ↪  nft-packs/program/src/processor/change_authority.rs:16:1: 36:2
          ↪  }
4
```

- description:
- link:
- alleviation:

Issue: 10: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

nft-packs/program/src/processor/claim_pack.rs:55:55: 55:96

```
55 proving_process_account.data.borrow_mut()
56
```

- Code Context

Vulnerability at Line: 55

```
50     assert_owned_by(randomness_oracle_account,
   ↪   &randomness_oracle_program::id())?;
51
52     assert_signer(&user_wallet_account)?;
53
54     let pack_set = PackSet::unpack(&pack_set_account.data.borrow())?;
55     let mut proving_process = ProvingPro-
       ↪   cess::unpack(&proving_process_account.data.borrow_mut())?;
56     let ClaimPackArgs { index } = args;
57
58     assert_account_key(user_wallet_account, &proving_process.wallet_key)?;
59     assert_account_key(pack_set_account, &proving_process.pack_set)?;
60
```

Other Use Case for Variable: proving_process_account.data.borrow_mut()

```
120     ProvingProcess::pack(proving_process,
        ↪   *proving_process_account.data.borrow_mut())?;
```

- Call Stack

```
fn entrypoint::process_instruction(){//
↪  nft-packs/program/src/entrypoint.rs:12:1: 23:2 }
    fn processor::Processor::process_instruction(){//
    ↪  nft-packs/program/src/processor.rs:42:5: 110:6 }
        fn processor::claim_pack::claim_pack(){//
        ↪  nft-packs/program/src/processor/claim_pack.rs:21:1: 124:2 }

```

- description:
- link:
- alleviation:

Issue: 11: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

nft-packs/program/src/processor/close_pack.rs:27:41: 27:75

```
27 pack_set_account.data.borrow_mut()
28
```

- Code Context

Vulnerability at Line: 27

```
22     let clock_info = next_account_info(account_info_iter)?;
23     let clock = Clock::from_account_info(clock_info)?;
24
25     assert_signer(&authority_account)?;
26
27     let mut pack_set =
       ↪  PackSet::unpack(&pack_set_account.data.borrow_mut())?;
28     assert_account_key(authority_account, &pack_set.authority)?;
29
30     if let Some(end_date) = pack_set.redeem_end_date {
31         if (clock.unix_timestamp as u64) < end_date {
32
```

Other Use Case for Variable: pack_set_account.data.borrow_mut()

```
42     PackSet::pack(pack_set, *pack_set_account.data.borrow_mut())?;
```

- Call Stack

```
1  fn entrypoint::process_instruction(){//
   ↪  nft-packs/program/src/entrypoint.rs:12:1: 23:2 }
2      fn processor::Processor::process_instruction(){//
       ↪  nft-packs/program/src/processor.rs:42:5: 110:6 }
```

```
3       fn processor::close_pack::close_pack(){//
        ↪  nft-packs/program/src/processor/close_pack.rs:18:1: 45:2 }
4
```

- description:
- link:
- alleviation:

Issue: 12: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

nft-packs/program/src/processor/deactivate.rs:22:41: 22:75

```
22 pack_set_account.data.borrow_mut()
23
```

- Code Context

Vulnerability at Line: 22

```
17     let pack_set_account = next_account_info(account_info_iter)?;
18     let authority_account = next_account_info(account_info_iter)?;
19
20     assert_signer(&authority_account)?;
21
22     let mut pack_set =
       ↪  PackSet::unpack(&pack_set_account.data.borrow_mut())?;
23     assert_account_key(authority_account, &pack_set.authority)?;
24
25     pack_set.assert_activated()?;
26
27
```

Other Use Case for Variable: pack_set_account.data.borrow_mut()

```
29     PackSet::pack(pack_set, *pack_set_account.data.borrow_mut())?;
```

- Call Stack

```
1  fn entrypoint::process_instruction(){//
   ↪  nft-packs/program/src/entrypoint.rs:12:1: 23:2 }
2      fn processor::Processor::process_instruction(){//
       ↪  nft-packs/program/src/processor.rs:42:5: 110:6 }
```

```
3         fn processor::deactivate::deactivate_pack(){//
          ↪  nft-packs/program/src/processor/deactivate.rs:15:1: 32:2 }
4
```

- description:
- link:
- alleviation:

Issue: 13: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

nft-packs/program/src/processor/activate.rs:23:41: 23:75

```
23 pack_set_account.data.borrow_mut()
24
```

- Code Context

Vulnerability at Line: 23

```
18     let pack_set_account = next_account_info(account_info_iter)?;
19     let authority_account = next_account_info(account_info_iter)?;
20
21     assert_signer(&authority_account)?;
22
23     let mut pack_set =
       ↪  PackSet::unpack(&pack_set_account.data.borrow_mut())?;
24     assert_account_key(authority_account, &pack_set.authority)?;
25
26     if pack_set.pack_cards == 0 || pack_set.pack_vouchers == 0 {
27         return Err(NFTPacksError::PackSetNotConfigured.into());
28
```

Other Use Case for Variable: pack_set_account.data.borrow_mut()

```
38     PackSet::pack(pack_set, *pack_set_account.data.borrow_mut())?;
```

- Call Stack

```
1  fn entrypoint::process_instruction(){//
   ↪  nft-packs/program/src/entrypoint.rs:12:1: 23:2 }
2      fn processor::Processor::process_instruction(){//
       ↪  nft-packs/program/src/processor.rs:42:5: 110:6 }
```

```
3         fn processor::activate::activate_pack(){//
          ↪  nft-packs/program/src/processor/activate.rs:16:1: 41:2 }
4
```

- description:
- link:
- alleviation:

Issue: 14: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

/home/yifei/.cargo/registry/src/github.com-1ecc6299db9ec823/mpl-token-metadata-1.1.0/src/utils.rs:139:43:139:69

```
139 account_info.data.borrow()
140
```

- Code Context

Vulnerability at Line: 139

```
136 pub fn assert_initialized<T: Pack + IsInitialized>(
137     account_info: &AccountInfo,
138 ) -> Result<T, ProgramError> {
139     let account: T = T::unpack_unchecked(&account_info.data.borrow())?;
140     if !account.is_initialized() {
141         Err(MetadataError::Uninitialized.into())
142     } else {
143         Ok(account)
144
```

- Call Stack

```
1 fn entrypoint::process_instruction(){//
  ↪  nft-packs/program/src/entrypoint.rs:12:1: 23:2 }
2     fn processor::Processor::process_instruction(){//
       ↪  nft-packs/program/src/processor.rs:42:5: 110:6 }
3         fn processor::add_voucher_to_pack::add_voucher_to_pack(){//
           ↪  nft-packs/program/src/processor/add_voucher_to_pack.rs:27:1:
           ↪  128:2 }
4             fn mpl_token_metadata::utils::assert_initialized(){//
               ↪  /home/yifei/.cargo/registry/src/github.com-
               ↪  1ecc6299db9ec823/mpl-token-metadata-
               ↪  1.1.0/src/utils.rs:136:1: 145:2
               ↪  }
```

5

- description:
- link:
- alleviation:

Issue: 15: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

/home/yifei/.cargo/registry/src/github.com-1ecc6299db9ec823/solana-program-1.9.5/src/account_info.rs:66:11: 66:33

```
66 self.lamports.borrow()
67
```

- Code Context

Vulnerability at Line: 66

```
65 pub fn lamports(&self) -> u64 {
66         **self.lamports.borrow()
67     }
68
```

- Call Stack

```
1 fn entrypoint::process_instruction(){//
  ↪  nft-packs/program/src/entrypoint.rs:12:1: 23:2 }
2     fn processor::Processor::process_instruction(){//
      ↪  nft-packs/program/src/processor.rs:42:5: 110:6 }
3         fn processor::init_pack::init_pack(){//
          ↪  nft-packs/program/src/processor/init_pack.rs:20:1: 104:2 }
4             fn utils::assert_rent_exempt(){//
              ↪  nft-packs/program/src/utils.rs:60:1: 66:2 }
5                     fn
                      ↪  solana_program::account_info::AccountInfo::<'a>::lamports(){//
                      ↪  /home/yifei/.cargo/registry/src/github.com-
                      ↪  1ecc6299db9ec823/solana-program-
                      ↪  1.9.5/src/account_info.rs:65:5: 67:6
                      ↪  }
6
```

- description:
- link:
- alleviation:

Issue: 16: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

nft-packs/program/src/processor/init_pack.rs:46:51: 46:85

```
46 pack_set_account.data.borrow_mut()
47
```

- Code Context

Vulnerability at Line: 46

```
41         store_account,
42         whitelisted_creator_account,
43         authority_account,
44     )?;
45
46     let mut pack_set =
       ↪  PackSet::unpack_unchecked(&pack_set_account.data.borrow_mut())?;
47
48     // make sure that random oracle account is already initialized
49     if random_oracle_account.data.borrow()[0]
50         != randomness_oracle_program::state::AccountType::RandomnessOracle
   ↪   as u8
51
```

Other Use Case for Variable: pack_set_account.data.borrow_mut()

```
101     PackSet::pack(pack_set, *pack_set_account.data.borrow_mut())?;
```

- Call Stack

```
1  fn entrypoint::process_instruction(){//
   ↪  nft-packs/program/src/entrypoint.rs:12:1: 23:2 }
2      fn processor::Processor::process_instruction(){//
       ↪  nft-packs/program/src/processor.rs:42:5: 110:6 }
```

```
3       fn processor::init_pack::init_pack(){//
        ↪  nft-packs/program/src/processor/init_pack.rs:20:1: 104:2 }
4
```

- description:
- link:
- alleviation:

Issue: 17: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

nft-packs/program/src/instruction.rs:66:1: 69:2

```
66 pub struct ClaimPackArgs {
67     /// Card index
68     pub index: u32,
69 }
70 nft-packs/program/src/instruction.rs:74:1: 77:2
71     pub struct RequestCardToRedeemArgs {
72     /// Voucher index
73     pub index: u32,
74 }
75 nft-packs/program/src/instruction.rs:18:1: 25:2
76     pub struct AddCardToPackArgs {
77     /// How many editions of this card will exists in pack
78     pub max_supply: u32,
79     /// Probability value, required only if PackSet distribution type ==
       ↪  Fixed
80     pub weight: u16,
81     /// Index
82     pub index: u32,
83 }
84
```

- Call Stack

```
1 UnResolved
```

- description:
- link:
- alleviation:

Issue: 18: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

nft-packs/program/src/instruction.rs:74:1: 77:2

```
74 pub struct RequestCardToRedeemArgs {
75     /// Voucher index
76     pub index: u32,
77 }
78 nft-packs/program/src/instruction.rs:18:1: 25:2
79     pub struct AddCardToPackArgs {
80     /// How many editions of this card will exists in pack
81     pub max_supply: u32,
82     /// Probability value, required only if PackSet distribution type ==
       ↪  Fixed
83     pub weight: u16,
84     /// Index
85     pub index: u32,
86 }
87
```

- Call Stack

```
1 UnResolved
```

- description:
- link:
- alleviation:

Issue: 19: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

nft-packs/program/src/state/pack_voucher.rs:15:1: 24:2

```
15 pub struct PackVoucher {
16     /// Account type - PackVoucher
17     pub account_type: AccountType,
18     /// Pack set
19     pub pack_set: Pubkey,
20     /// Master edition account
21     pub master: Pubkey,
22     /// Metadata account
23     pub metadata: Pubkey,
24 }
25 nft-packs/program/src/state/pack_card.rs:16:1: 31:2
26     pub struct PackCard {
27     /// Account type - PackCard
28     pub account_type: AccountType,
29     /// Pack set
30     pub pack_set: Pubkey,
31     /// Master edition account
32     pub master: Pubkey,
33     /// Metadata account
34     pub metadata: Pubkey,
35     /// Program token account which holds MasterEdition token
36     pub token_account: Pubkey,
37     /// How many instances(editions) of this card exists in this pack
38     pub max_supply: u32,
39     /// Fixed probability, should be filled if PackSet distribution_type is
       ↪  "fixed"
40     pub weight: u16,
41 }
42
```

- Call Stack

```
UnResolved
```

- description:
- link:
- alleviation:

Appendix

Copied from https://leaderboard.certik.io/projects/aave

Finding Categories

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

Mathematical Operations

Mathematical Operation findings relate to mishandling of math formulas, such as overflows, incorrect operations etc.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block.timestamp works.

Language Specific

Language Specific findings are issues that would only arise within Solidity, i.e. incorrect usage of private or delete.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

Checksum Calculation Method

The “Checksum” field in the “Audit Scope” section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux “sha256sum” command against the target file.

Disclaimer

Copied from https://leaderboard.certik.io/projects/aave

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you (“Customer” or the “Company”) in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK’s prior written consent in each instance.

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology. Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK’s position is that each company and individual are responsible for their own due diligence and continuous security. CertiK’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.