VRust

Security Assessment

O2Lab VRust Team

11/04/2022 19:29:02

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_metaplex
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	37
Major	0
Medium	0
Minor	0
Informational	0
Discussion	0

Findings

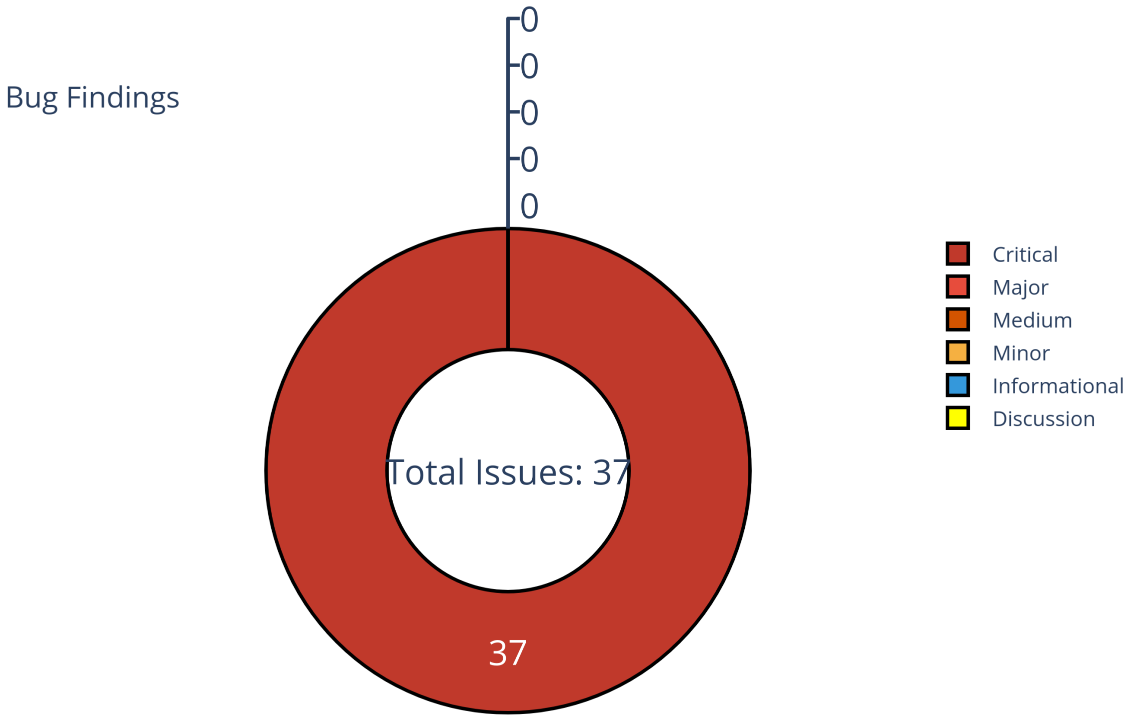

Figure 1: Findings

Finding Statistic

Category	Count
IntegerFlow	10
MissingKeyCheck	13
TypeConfusion	14

ID	Category	Severity	Status
0	IntegerFlow	Critical	UnResolved
1	IntegerFlow	Critical	UnResolved
2	IntegerFlow	Critical	UnResolved
3	IntegerFlow	Critical	UnResolved
4	IntegerFlow	Critical	UnResolved
5	IntegerFlow	Critical	UnResolved
6	IntegerFlow	Critical	UnResolved
7	IntegerFlow	Critical	UnResolved
8	IntegerFlow	Critical	UnResolved
9	IntegerFlow	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	MissingKeyCheck	Critical	UnResolved
18	MissingKeyCheck	Critical	UnResolved
19	MissingKeyCheck	Critical	UnResolved

ID	Category	Severity	Status
20	MissingKeyCheck	Critical	UnResolved
21	MissingKeyCheck	Critical	UnResolved
22	MissingKeyCheck	Critical	UnResolved
23	TypeConfusion	Critical	GitHub Link to be added.
24	TypeConfusion	Critical	GitHub Link to be added.
25	TypeConfusion	Critical	GitHub Link to be added.
26	TypeConfusion	Critical	GitHub Link to be added.
27	TypeConfusion	Critical	GitHub Link to be added.
28	TypeConfusion	Critical	GitHub Link to be added.
29	TypeConfusion	Critical	GitHub Link to be added.
30	TypeConfusion	Critical	GitHub Link to be added.
31	TypeConfusion	Critical	GitHub Link to be added.
32	TypeConfusion	Critical	GitHub Link to be added.
33	TypeConfusion	Critical	GitHub Link to be added.
34	TypeConfusion	Critical	GitHub Link to be added.
35	TypeConfusion	Critical	GitHub Link to be added.
36	TypeConfusion	Critical	GitHub Link to be added.

Issue: 0: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

metaplex/program/src/processor/set_store_index.rs:169:49: 169:61

```
offset_u - 1

```

- Code Context

Vulnerability at Line: 169

```
      return Err(MetaplexError::ExpectedAboveAuctionCacheToBeProvided.into());
    }
  }

  if offset_u > 0 {
    let below_key = &indexer.auction_caches[offset_u - 1];
    // special case where you're at top of stack, there is no above
    let cache_used_for_below = if offset_u == indexer.auction_caches.len() - 1 {
      &above_cache
    } else {

```

- Call Stack

```
fn entrypoint::process_instruction(){// metaplex/program/src/entrypoint.rs:14:1: 25:2 }
  fn processor::process_instruction(){// metaplex/program/src/processor.rs:50:1: 169:2 }
    fn processor::set_store_index::process_set_store_index(){// metaplex/program/src/processor/set_store_index.rs:22:1: 217:2 }

```

- description:
- link:
- alleviation:

Issue: 1: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

metaplex/program/src/state.rs:1538:35: 1540:59

```
7 - order
            .checked_rem(8)
            .ok_or(MetaplexError::NumericalOverflowError)?

```

- Code Context

Vulnerability at Line: 1538

```
        let u8_position = order
            .checked_div(8)
            .ok_or(MetaplexError::NumericalOverflowError)?
            .checked_add(offset)
            .ok_or(MetaplexError::NumericalOverflowError)?;
        let position_from_right = 7 - order
            .checked_rem(8)
            .ok_or(MetaplexError::NumericalOverflowError)?;
        let mask = u8::pow(2, position_from_right as u32);


```

- Call Stack

```
fn entrypoint::process_instruction(){// metaplex/program/src/entrypoint.rs:14:1: 25:2 }
    fn processor::process_instruction(){// metaplex/program/src/processor.rs:50:1: 169:2 }
        fn processor::redeem_participation_bid::process_redeem_participation_bid(){// metaplex/program/src/processor/redeem_participation_bid.rs:235:1: 447:2 }
```

```
            fn utils::common_redeem_checks(){// metaplex/program/src/utils.rs:509:1: 681:2 }
                    fn state::BidRedemptionTicket::check_ticket(){// metaplex/program/src/state.rs:1476:5: 1520:6 }
                        fn state::BidRedemptionTicket::get_index_and_mask(){// metaplex/program/src/state.rs:1522:5: 1544:6 }

```

- description:
- link:
- alleviation:

Issue: 2: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

metaplex/program/src/utils.rs:742:17: 742:59

```
742 1 + 9 + 32 + 1 + token_type_count as usize
743
```

- Code Context

Vulnerability at Line: 742

```
737                 *program_id,
738                 &bid_redemption_info,
739                 &rent_info,
740                 &system_info,
741                 &payer_info,
742                 1 + 9 + 32 + 1 + token_type_count as usize,
743                 redemption_seeds,
744             )?;
745         }
746
747
```

- Call Stack

```
fn entrypoint::process_instruction(){// metaplex/program/src/entrypoint.rs:14:1: 25:2 }
    fn processor::process_instruction(){// metaplex/program/src/processor.rs:50:1: 169:2 }
        fn processor::redeem_participation_bid::process_redeem_participation_bid(){// metaplex/program/src/processor/redeem_participation_bid.rs:235:1: 447:2 }
            fn utils::common_redeem_finish(){// metaplex/program/src/utils.rs:702:1: 776:2 }
```

```

```

- description:
- link:
- alleviation:

Issue: 3: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

metaplex/program/src/state.rs:1460:13: 1460:48

```
offset += self.amount_type as usize

```

- Code Context

Vulnerability at Line: 1460

```
        data[2] = self.length_type as u8;
        *array_mut_ref![data, 3, 4] = (self.amount_ranges.len() as
↪   u32).to_le_bytes();
        let mut offset: usize = 7;
        for range in &self.amount_ranges {
            write_amount_type(&mut data, self.amount_type, offset, range);
            offset += self.amount_type as usize;
            write_length_type(&mut data, self.length_type, offset, range);
            offset += self.length_type as usize;
        }
    }

```

- Call Stack

```
fn entrypoint::process_instruction(){//
↪   metaplex/program/src/entrypoint.rs:14:1: 25:2 }
    fn processor::process_instruction(){//
    ↪   metaplex/program/src/processor.rs:50:1: 169:2 }
        fn proces-
        ↪   sor::validate_safety_deposit_box_v2::process_validate_safety_deposit_box_v2(
        ↪   meta-
        ↪   plex/program/src/processor/validate_safety_deposit_box_v2.rs:383:1:
        ↪   533:2 }
            fn state::AuctionWinnerTokenTypeTracker::save(){//
            ↪   metaplex/program/src/state.rs:1451:5: 1464:6 }
```

- description:
- link:
- alleviation:

Issue: 4: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

/home/yifei/.cargo/registry/src/github.com-1ecc6299db9ec823/arrayref-0.3.6/src/lib.rs:290:43:290:49

```
290 offset
291
```

- Code Context

– Function Definition:

```
887 fn write_length_type(
888     data: &mut RefMut<&mut [u8]>,
889     length_type: TupleNumericType,
890     offset: usize,
891     range: &AmountRange,
892 )
893
```

Vulnerability at Line: 890

```
887 fn write_length_type(
888     data: &mut RefMut<&mut [u8]>,
889     length_type: TupleNumericType,
890     offset: usize,
891     range: &AmountRange,
892 ) {
893     match length_type {
894         TupleNumericType::U8 => data[offset] = range.1 as u8,
895
```

Other Use Case for Variable: offset

```
        TupleNumericType::U8 => data[offset] = range.1 as u8,
```

```
        TupleNumericType::U16 => *array_mut_ref![data, offset, 2] = (range.1 as u16).to_le_bytes(),
```

```
        TupleNumericType::U32 => *array_mut_ref![data, offset, 4] = (range.1 as u32).to_le_bytes(),
```

```
        TupleNumericType::U64 => *array_mut_ref![data, offset, 8] = range.1.to_le_bytes(),
```

- Call Stack

```
fn entrypoint::process_instruction(){// metaplex/program/src/entrypoint.rs:14:1: 25:2 }
    fn processor::process_instruction(){// metaplex/program/src/processor.rs:50:1: 169:2 }
        fn processor::validate_safety_deposit_box_v2::process_validate_safety_deposit_box_v2(
        metaplex/program/src/processor/validate_safety_deposit_box_v2.rs:383:1: 533:2 }
            fn state::AuctionWinnerTokenTypeTracker::save(){// metaplex/program/src/state.rs:1451:5: 1464:6 }
                fn state::write_length_type(){// metaplex/program/src/state.rs:887:1: 900:2 }

```

- description:
- link:
- alleviation:

Issue: 5: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

metaplex/program/src/state.rs:906:15: 906:70

```
(self.amount_type as usize + self.length_type as usize)

```

- Code Context

Vulnerability at Line: 906

```
pub fn created_size(&self) -> usize {
        return BASE_SAFETY_CONFIG_SIZE
            + (self.amount_type as usize + self.length_type as usize) *
 ↪  self.amount_ranges.len();
    }

```

- Call Stack

```
fn entrypoint::process_instruction(){//
 ↪  metaplex/program/src/entrypoint.rs:14:1: 25:2 }
    fn processor::process_instruction(){//
     ↪  metaplex/program/src/processor.rs:50:1: 169:2 }
        fn proces-
         ↪  sor::validate_safety_deposit_box_v2::process_validate_safety_deposit_box_v2(
         ↪  meta-
         ↪  plex/program/src/processor/validate_safety_deposit_box_v2.rs:383:1:
         ↪  533:2 }
            fn proces-
             ↪  sor::validate_safety_deposit_box_v2::make_safety_deposit_config(){//
             ↪  meta-
             ↪  plex/program/src/processor/validate_safety_deposit_box_v2.rs:27:1:
             ↪  67:2 }
                fn state::SafetyDepositConfig::created_size(){//
                 ↪  metaplex/program/src/state.rs:904:5: 907:6 }
```

```

```

- description:
- link:
- alleviation:

Issue: 6: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

metaplex/program/src/state.rs:1182:13: 1182:48

```
1182 offset += self.amount_type as usize
1183
```

- Code Context

Vulnerability at Line: 1182

```
1177         *array_mut_ref![data, AMOUNT_RANGE_SIZE_POSITION, 4] =
1178             (self.amount_ranges.len() as u32).to_le_bytes();
1179         let mut offset: usize = AMOUNT_RANGE_FIRST_EL_POSITION;
1180         for range in &self.amount_ranges {
1181             write_amount_type(&mut data, self.amount_type, offset, range);
1182             offset += self.amount_type as usize;
1183             write_length_type(&mut data, self.length_type, offset, range);
1184             offset += self.length_type as usize;
1185         }
1186
1187
```

- Call Stack

```
fn entrypoint::process_instruction(){//
↪  metaplex/program/src/entrypoint.rs:14:1: 25:2 }
    fn processor::process_instruction(){//
    ↪  metaplex/program/src/processor.rs:50:1: 169:2 }
        fn proces-
        ↪  sor::validate_safety_deposit_box_v2::process_validate_safety_deposit_box_v2(
        ↪  meta-
        ↪  plex/program/src/processor/validate_safety_deposit_box_v2.rs:383:1:
        ↪  533:2 }
            fn proces-
            ↪  sor::validate_safety_deposit_box_v2::make_safety_deposit_config(){//
            ↪  meta-
            ↪  plex/program/src/processor/validate_safety_deposit_box_v2.rs:27:1:
            ↪  67:2 }
```

```
                    fn state::SafetyDepositConfig::create(){// metaplex/program/src/state.rs:1164:5: 1227:6 }

```

- description:
- link:
- alleviation:

Issue: 7: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

metaplex/program/src/state.rs:1278:15: 1278:92

```
(self.amount_type as usize + self.length_type as usize) * range_size as usize

```

- Code Context

Vulnerability at Line: 1278

```
pub fn created_size(&self, range_size: u64) -> usize {
        return BASE_TRACKER_SIZE
            + (self.amount_type as usize + self.length_type as usize) * range_size as usize;
    }

```

- Call Stack

```
fn entrypoint::process_instruction(){// metaplex/program/src/entrypoint.rs:14:1: 25:2 }
    fn processor::process_instruction(){// metaplex/program/src/processor.rs:50:1: 169:2 }
        fn processor::init_auction_manager_v2::process_init_auction_manager_v2(){// metaplex/program/src/processor/init_auction_manager_v2.rs:97:1: 196:2 }
            fn state::AuctionWinnerTokenTypeTracker::created_size(){// metaplex/program/src/state.rs:1276:5: 1279:6 }

```

- description:

- link:
- alleviation:

Issue: 8: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

/mnt/data/yifei/open/vrust/examples2/metaplex-program-library/auction/program/src/processor.rs:650:33: 650:51

```
bids.len() - index

```

- Code Context

Vulnerability at Line: 650

```
pub fn amount(&self, index: usize) -> u64 {
        match self {
            BidState::EnglishAuction { bids, max } => {
                if index >= 0 as usize && index < bids.len() {
                    return bids[bids.len() - index - 1].1;
                } else {
                    return 0;
                }
            }

```

- Call Stack

```
fn entrypoint::process_instruction(){// metaplex/program/src/entrypoint.rs:14:1: 25:2 }
    fn processor::process_instruction(){// metaplex/program/src/processor.rs:50:1: 169:2 }
        fn processor::empty_payment_account::process_empty_payment_account(){// metaplex/program/src/processor/empty_payment_account.rs:212:1: 458:2 }
            fn processor::empty_payment_account::calculate_owed_amount(){// metaplex/program/src/processor/empty_payment_account.rs:75:1: 210:2 }
```

```
                    fn mpl_auction::processor::BidState::amount(){///mnt/data/yifei/open/vrust/examples2/metaplex-program-library/auction/program/src/processor.rs:646:5: 657:6 }

```

- description:
- link:
- alleviation:

Issue: 9: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

metaplex/program/src/processor/empty_payment_account.rs:126:17: 126:64

```
126 (10000 - metadata.data.seller_fee_basis_points)
127
```

- Code Context

– Function Definition:

```
75  fn calculate_owed_amount(
76      auction_token_tracker_info: Option<&AccountInfo>,
77      safety_deposit_config_info: Option<&AccountInfo>,
78      auction_manager: &Box<dyn AuctionManager>,
79      auction: &AuctionData,
80      metadata: &Metadata,
81      winning_config_index: &Option<u8>,
82      winning_config_item_index: &Option<u8>,
83      creator_index: &Option<u8>,
84  ) -> Result<u64, ProgramError>
85
```

Vulnerability at Line: 126

```
121             }
122         }
123         None => {
124             if primary_sale_happened {
125                 // during secondary sale, auctioneer gets whats left after
                     ↪  artists get their cut
126                 (10000 - metadata.data.seller_fee_basis_points) as u128
127             } else {
128                 // during primary sale, auctioneer (creator index not
                     ↪  provided)
```

```
                    // get none of the proceeds
                    0u128

```

- Call Stack

```
fn entrypoint::process_instruction(){//
↪   metaplex/program/src/entrypoint.rs:14:1: 25:2 }
    fn processor::process_instruction(){//
    ↪   metaplex/program/src/processor.rs:50:1: 169:2 }
        fn proces-
        ↪   sor::empty_payment_account::process_empty_payment_account(){//
        ↪   metaplex/program/src/processor/empty_payment_account.rs:212:1:
        ↪   458:2 }
            fn processor::empty_payment_account::calculate_owed_amount(){//
            ↪   meta-
            ↪   plex/program/src/processor/empty_payment_account.rs:75:1:
            ↪   210:2 }

```

- description:
- link:
- alleviation:

Issue: 10: 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

```
self.lamports.borrow()

```

- Code Context

Vulnerability at Line: 66

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

```

- Call Stack

```
fn entrypoint::process_instruction(){//
↪   metaplex/program/src/entrypoint.rs:14:1: 25:2 }
    fn processor::process_instruction(){//
    ↪   metaplex/program/src/processor.rs:50:1: 169:2 }
        fn processor::set_store::process_set_store_v2(){//
        ↪   metaplex/program/src/processor/set_store.rs:122:1: 193:2 }
            fn processor::set_store::set_store_logic(){//
            ↪   metaplex/program/src/processor/set_store.rs:19:1: 88:2 }
                    fn utils::create_or_allocate_account_raw(){//
                    ↪   metaplex/program/src/utils.rs:212:1: 257:2 }
                        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
                        ↪   }

```

- description:
- link:
- alleviation:

Issue: 11: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

metaplex/program/src/processor/set_store.rs:86:27: 86:55

```
store_info.data.borrow_mut()

```

- Code Context

Vulnerability at Line: 86

```
        store.token_metadata_program = *token_metadata_program_info.key;
    }
    if store.auction_program == solana_program::system_program::id() {
        store.auction_program = *auction_program_info.key;
    }
    store.serialize(&mut *store_info.data.borrow_mut())?;
    Ok(())
}

```

- Call Stack

```
fn entrypoint::process_instruction(){//
↪  metaplex/program/src/entrypoint.rs:14:1: 25:2 }
    fn processor::process_instruction(){//
    ↪  metaplex/program/src/processor.rs:50:1: 169:2 }
        fn processor::set_store::process_set_store_v2(){//
        ↪  metaplex/program/src/processor/set_store.rs:122:1: 193:2 }
            fn processor::set_store::set_store_logic(){//
            ↪  metaplex/program/src/processor/set_store.rs:19:1: 88:2 }

```

- description:

- link:
- alleviation:

Issue: 12: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

metaplex/program/src/state.rs:213:19: 213:40

```
213 account.data.borrow()
214
```

- Code Context

Vulnerability at Line: 213

```
212 pub fn get_auction_manager(account: &AccountInfo) -> Result<Box<dyn AuctionManager>, ProgramError> {
213     let version = account.data.borrow()[0];
214 
215     // For some reason when converting Key to u8 here, it becomes unreachable. Use direct constant instead.
216     match version {
217         7 => return Ok(Box::new(AuctionManagerV1::from_account_info(account)?)),
218
```

- Call Stack

```
fn entrypoint::process_instruction(){// metaplex/program/src/entrypoint.rs:14:1: 25:2 }
    fn processor::process_instruction(){// metaplex/program/src/processor.rs:50:1: 169:2 }
        fn processor::end_auction::process_end_auction(){// metaplex/program/src/processor/end_auction.rs:43:1: 118:2 }
            fn state::get_auction_manager(){// metaplex/program/src/state.rs:212:1: 221:2 }

```

- description:

- link:
- alleviation:

Issue: 13: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

metaplex/program/src/utils.rs:540:22: 540:46

```
540 store_info.data.borrow()
541
```

- Code Context

Vulnerability at Line: 540

```
535     } = args;
536
537     let rent = &Rent::from_account_info(&rent_info)?;
538
539     let mut auction_manager: Box<dyn AuctionManager> =
     ↪  get_auction_manager(auction_manager_info)?;
540     let store_data = store_info.data.borrow();
541     let cancelled: bool;
542
543     let auction_program = Pubkey::new_from_array(*array_ref![store_data, 2,
     ↪  32]);
544     let token_vault_program =
     ↪  Pubkey::new_from_array(*array_ref![store_data, 34, 32]);
545
```

- Call Stack

```
1 fn entrypoint::process_instruction(){//
  ↪ metaplex/program/src/entrypoint.rs:14:1: 25:2 }
2     fn processor::process_instruction(){//
      ↪ metaplex/program/src/processor.rs:50:1: 169:2 }
3         fn proces-
          ↪ sor::redeem_participation_bid::process_redeem_participation_bid(){//
          ↪ meta-
          ↪ plex/program/src/processor/redeem_participation_bid.rs:235:1:
          ↪ 447:2 }
```

```
            fn utils::common_redeem_checks(){// metaplex/program/src/utils.rs:509:1: 681:2 }

```

- description:
- link:
- alleviation:

Issue: 14: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

/mnt/data/yifei/open/vrust/examples2/metaplex-program-library/auction/program/src/processor.rs:191:20: 191:35

```
191 a.data.borrow()
192
```

- Code Context

– Function Definition:

```
190 fn find_bid_state_beginning(a: &AccountInfo) -> usize
191
```

Vulnerability at Line: 191

```
190 fn find_bid_state_beginning(a: &AccountInfo) -> usize {
191         let data = a.data.borrow();
192         let mut bid_state_beginning = 32 + 32;
193
194         for i in 0..4 {
195             // One for each unix timestamp
196
```

- Call Stack

```
fn entrypoint::process_instruction(){//
↪   metaplex/program/src/entrypoint.rs:14:1: 25:2 }
    fn processor::process_instruction(){//
    ↪   metaplex/program/src/processor.rs:50:1: 169:2 }
        fn proces-
        ↪   sor::redeem_participation_bid::process_redeem_participation_bid(){//
        ↪   meta-
        ↪   plex/program/src/processor/redeem_participation_bid.rs:235:1:
        ↪   447:2 }
```

```
            fn utils::common_redeem_checks(){// metaplex/program/src/utils.rs:509:1: 681:2 }
                fn utils::calculate_win_index(){// metaplex/program/src/utils.rs:422:1: 471:2 }
                    fn mpl_auction::processor::AuctionData::get_is_winner(){// /mnt/data/yifei/open/vrust/examples2/metaplex-program-library/auction/program/src/processor.rs:221:5: 252:6 }
                        fn mpl_auction::processor::AuctionData::find_bid_state_begi
                        /mnt/data/yifei/open/vrust/examples2/metaplex-program-library/auction/program/src/processor.rs:190:5: 207:6 }

```

- description:
- link:
- alleviation:

Issue: 15: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

/mnt/data/yifei/open/vrust/examples2/metaplex-program-library/auction/program/src/processor.rs:223:20: 223:35

```
a.data.borrow()

```

- Code Context

Vulnerability at Line: 223

```
pub fn get_is_winner(a: &AccountInfo, key: &Pubkey) -> Option<usize> {
        let bid_state_beginning = AuctionData::find_bid_state_beginning(a);
        let data = a.data.borrow();
        let as_bytes = key.to_bytes();
        let (bid_state_beginning, num_elements, max) =
         ↪  AuctionData::get_vec_info(a);
        for idx in 0..std::cmp::min(num_elements, max) {
            match AuctionData::get_winner_at_inner(

```

Other Use Case for Variable: a.data.borrow()

```
                &a.data.borrow(),
```

- Call Stack

```
fn entrypoint::process_instruction(){//
↪  metaplex/program/src/entrypoint.rs:14:1: 25:2 }
    fn processor::process_instruction(){//
    ↪  metaplex/program/src/processor.rs:50:1: 169:2 }
        fn proces-
         ↪  sor::redeem_participation_bid::process_redeem_participation_bid(){//
         ↪  meta-
         ↪  plex/program/src/processor/redeem_participation_bid.rs:235:1:
         ↪  447:2 }
```

```
            fn utils::common_redeem_checks(){// metaplex/program/src/utils.rs:509:1: 681:2 }
                    fn utils::calculate_win_index(){// metaplex/program/src/utils.rs:422:1: 471:2 }
                        fn mpl_auction::processor::AuctionData::get_is_winner(){// /mnt/data/yifei/open/vrust/examples2/metaplex-program-library/auction/program/src/processor.rs:221:5: 252:6 }

```

- description:
- link:
- alleviation:

Issue: 16: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

/mnt/data/yifei/open/vrust/examples2/metaplex-program-library/auction/program/src/processor.rs:211:20: 211:35

```
211 a.data.borrow()
212
```

- Code Context

– Function Definition:

```
209 fn get_vec_info(a: &AccountInfo) -> (usize, usize, usize)
210
```

Vulnerability at Line: 211

```
209 fn get_vec_info(a: &AccountInfo) -> (usize, usize, usize) {
210         let bid_state_beginning = AuctionData::find_bid_state_beginning(a);
211         let data = a.data.borrow();
212
213         let num_elements_data = array_ref![data, bid_state_beginning - 4, 4];
214         let num_elements = u32::from_le_bytes(*num_elements_data) as usize;
215         let max_data = array_ref![data, bid_state_beginning + BID_LENGTH * num_elements, 8];
216
```

- Call Stack

```
1 fn entrypoint::process_instruction(){// metaplex/program/src/entrypoint.rs:14:1: 25:2 }
2     fn processor::process_instruction(){// metaplex/program/src/processor.rs:50:1: 169:2 }
```

```
        fn proces-
         ↪  sor::redeem_participation_bid::process_redeem_participation_bid(){//
         ↪  meta-
         ↪  plex/program/src/processor/redeem_participation_bid.rs:235:1:
         ↪  447:2 }
            fn utils::common_redeem_checks(){//
             ↪  metaplex/program/src/utils.rs:509:1: 681:2 }
                    fn utils::calculate_win_index(){//
                     ↪  metaplex/program/src/utils.rs:422:1: 471:2 }
                        fn
                         ↪  mpl_auction::processor::AuctionData::get_is_winner(){//
                         ↪  /mnt/data/yifei/open/vrust/examples2/metaplex-
                         ↪  program-
                         ↪  library/auction/program/src/processor.rs:221:5:
                         ↪  252:6 }
                            fn
                             ↪  mpl_auction::processor::AuctionData::get_vec_info(){//
                             ↪  /mnt/data/yifei/open/vrust/examples2/metaplex-
                             ↪  program-
                             ↪  library/auction/program/src/processor.rs:209:5:
                             ↪  219:6 }

```

- description:
- link:
- alleviation:

Issue: 17: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

/mnt/data/yifei/open/vrust/examples2/metaplex-program-library/auction/program/src/processor.rs:257:14: 257:29

```
257 a.data.borrow()
258
```

- Code Context

Vulnerability at Line: 257

```
254 pub fn get_winner_at(a: &AccountInfo, idx: usize) -> Option<Pubkey> {
255         let (bid_state_beginning, num_elements, max) =
            ↪  AuctionData::get_vec_info(a);
256         match AuctionData::get_winner_at_inner(
257             &a.data.borrow(),
258             idx,
259             bid_state_beginning,
260             num_elements,
261             max,
262
```

- Call Stack

```
1 fn entrypoint::process_instruction(){//
  ↪  metaplex/program/src/entrypoint.rs:14:1: 25:2 }
2     fn processor::process_instruction(){//
      ↪  metaplex/program/src/processor.rs:50:1: 169:2 }
3         fn proces-
          ↪  sor::redeem_participation_bid::process_redeem_participation_bid(){//
          ↪  meta-
          ↪  plex/program/src/processor/redeem_participation_bid.rs:235:1:
          ↪  447:2 }
4             fn utils::common_redeem_checks(){//
              ↪  metaplex/program/src/utils.rs:509:1: 681:2 }
```

```
                        fn utils::calculate_win_index(){//
                         ↪  metaplex/program/src/utils.rs:422:1: 471:2 }
                            fn
                             ↪  mpl_auction::processor::AuctionData::get_winner_at(){//
                             ↪  /mnt/data/yifei/open/vrust/examples2/metaplex-
                             ↪  program-
                             ↪  library/auction/program/src/processor.rs:254:5:
                             ↪  266:6 }

```

- description:
- link:
- alleviation:

Issue: 18: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

metaplex/program/src/state.rs:910:20: 910:35

```
a.data.borrow()

```

- Code Context

Vulnerability at Line: 910

```
pub fn get_order(a: &AccountInfo) -> u64 {
        let data = a.data.borrow();
        return u64::from_le_bytes(*array_ref![data, ORDER_POSITION, 8]);
    }

```

- Call Stack

```
fn entrypoint::process_instruction(){//
↪   metaplex/program/src/entrypoint.rs:14:1: 25:2 }
    fn processor::process_instruction(){//
    ↪   metaplex/program/src/processor.rs:50:1: 169:2 }
        fn proces-
        ↪   sor::redeem_participation_bid::process_redeem_participation_bid(){//
        ↪   meta-
        ↪   plex/program/src/processor/redeem_participation_bid.rs:235:1:
        ↪   447:2 }
            fn utils::common_redeem_checks(){//
            ↪   metaplex/program/src/utils.rs:509:1: 681:2 }
                    fn state::BidRedemptionTicket::check_ticket(){//
                    ↪   metaplex/program/src/state.rs:1476:5: 1520:6 }
                        fn state::SafetyDepositConfig::get_order(){//
                        ↪   metaplex/program/src/state.rs:909:5: 912:6 }

```

- description:
- link:
- alleviation:

Issue: 19: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

/mnt/data/yifei/open/vrust/examples2/metaplex-program-library/auction/program/src/processor.rs:168:20:168:35

```
a.data.borrow()

```

- Code Context

Vulnerability at Line: 168

```
pub fn get_token_mint(a: &AccountInfo) -> Pubkey {
        let data = a.data.borrow();
        let token_mint_data = array_ref![data, 32, 32];
        Pubkey::new_from_array(*token_mint_data)
    }

```

- Call Stack

```
fn entrypoint::process_instruction(){//
 ↪  metaplex/program/src/entrypoint.rs:14:1: 25:2 }
    fn processor::process_instruction(){//
     ↪  metaplex/program/src/processor.rs:50:1: 169:2 }
        fn proces-
         ↪  sor::redeem_participation_bid::process_redeem_participation_bid(){//
         ↪  meta-
         ↪  plex/program/src/processor/redeem_participation_bid.rs:235:1:
         ↪  447:2 }
            fn mpl_auction::processor::AuctionData::get_token_mint(){//
             ↪  /mnt/data/yifei/open/vrust/examples2/metaplex-program-
             ↪  library/auction/program/src/processor.rs:167:5: 171:6
             ↪  }

```

- description:
- link:
- alleviation:

Issue: 20: 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:102:9: 103:26

```
102 self.data
103             .try_borrow()
104
```

- Code Context

Vulnerability at Line: 102

```
101 pub fn try_borrow_data(&self) -> Result<Ref<&mut [u8]>, ProgramError> {
102         self.data
103             .try_borrow()
104             .map_err(|_| ProgramError::AccountBorrowFailed)
105     }
106
```

- Call Stack

```
1 fn entrypoint::process_instruction(){//
  ↪  metaplex/program/src/entrypoint.rs:14:1: 25:2 }
2     fn processor::process_instruction(){//
      ↪  metaplex/program/src/processor.rs:50:1: 169:2 }
3         fn proces-
          ↪  sor::redeem_participation_bid::process_redeem_participation_bid(){//
          ↪  meta-
          ↪  plex/program/src/processor/redeem_participation_bid.rs:235:1:
          ↪  447:2 }
4             fn
              ↪  mpl_token_metadata::utils::get_supply_off_master_edition(){//
              ↪  /home/yifei/.cargo/registry/src/github.com-
              ↪  1ecc6299db9ec823/mpl-token-metadata-
              ↪  1.1.0/src/utils.rs:483:1: 491:2
              ↪  }
```

```
                    fn
                    ↪  solana_program::account_info::AccountInfo::<'a>::try_borrow_data
                    ↪  /home/yifei/.cargo/registry/src/github.com-
                    ↪  1ecc6299db9ec823/solana-program-
                    ↪  1.9.5/src/account_info.rs:101:5: 105:6
                    ↪  }

```

- description:
- link:
- alleviation:

Issue: 21: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

metaplex/program/src/processor/redeem_printing_v2_bid.rs:107:25: 107:60

```
107 metadata_account_info.data.borrow()
108
```

- Code Context

Vulnerability at Line: 107

```
102     rent_info: &AccountInfo<'a>,
103     system_info: &AccountInfo<'a>,
104     master_edition_account_info: &AccountInfo<'a>,
105     expected_redemptions: u64,
106 ) -> Result<u64, ProgramError> {
107     let metadata_data = metadata_account_info.data.borrow();
108     let metadata_mint = Pubkey::new_from_array(*array_ref![metadata_data, 33, 32]);
109
110     let bump = assert_derivation(
111         program_id,
112
```

- Call Stack

```
fn entrypoint::process_instruction(){// metaplex/program/src/entrypoint.rs:14:1: 25:2 }
    fn processor::process_instruction(){// metaplex/program/src/processor.rs:50:1: 169:2 }
        fn processor::redeem_participation_bid::process_redeem_participation_bid(){// metaplex/program/src/processor/redeem_participation_bid.rs:235:1: 447:2 }
            fn processor::redeem_participation_bid::v2_validation(){// metaplex/program/src/processor/redeem_participation_bid.rs:65:1: 129:2 }
```

```
                    fn proces-
                    ↪  sor::redeem_printing_v2_bid::create_or_update_prize_tracking(){/
                    ↪  meta-
                    ↪  plex/program/src/processor/redeem_printing_v2_bid.rs:96:1:
                    ↪  165:2 }
```

- description:
- link:
- alleviation:

Issue: 22: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

/mnt/data/yifei/open/vrust/examples2/metaplex-program-library/token-vault/program/src/state.rs:71:16: 71:31

```
a.data.borrow()

```

- Code Context

Vulnerability at Line: 71

```
pub fn get_token_type_count(a: &AccountInfo) -> u8 {
        return a.data.borrow()[194];
    }

```

- Call Stack

```
fn entrypoint::process_instruction(){// metaplex/program/src/entrypoint.rs:14:1: 25:2 }
    fn processor::process_instruction(){// metaplex/program/src/processor.rs:50:1: 169:2 }
        fn processor::redeem_participation_bid::process_redeem_participation_bid(){// metaplex/program/src/processor/redeem_participation_bid.rs:235:1: 447:2 }
            fn utils::common_redeem_finish(){// metaplex/program/src/utils.rs:702:1: 776:2 }
                fn mpl_token_vault::state::Vault::get_token_type_count(){// /mnt/data/yifei/open/vrust/examples2/metaplex-program-library/token-vault/program/src/state.rs:70:5: 72:6 }
```

```

```

- description:
- link:
- alleviation:

Issue: 23: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

metaplex/program/src/processor/redeem_participation_bid.rs:25:1: 27:2

```
struct LegacyAccounts<'a> {
    pub participation_printing_holding_account_info: &'a AccountInfo<'a>,
}
metaplex/program/src/state.rs:558:1: 566:2
    pub struct ParticipationStateV2 {
    /// We have this variable below to keep track in the case of the participation NFTs, whose
    /// income will trickle in over time, how much the artists have in the escrow account and
    /// how much would/should be owed to them if they try to claim it relative to the winning bids.
    /// It's  abit tougher than a straightforward bid which has a price attached to it, because
    /// there are many bids of differing amounts (in the case of GivenForBidPrice) and they dont all
    /// come in at one time, so this little ledger here keeps track.
    pub collected_to_accept_payment: u64,
}
metaplex/program/src/instruction.rs:52:1: 55:2
    pub struct RedeemPrintingV2BidArgs {
    pub edition_offset: u64,
    pub win_index: u64,
}
metaplex/program/src/instruction.rs:82:1: 85:2
    pub struct SetStoreIndexArgs {
    pub page: u64,
    pub offset: u64,
}
metaplex/program/src/state.rs:805:1: 805:42
    pub struct AmountRange(pub u64, pub u64);
metaplex/program/src/state.rs:118:1: 122:2
```

```
    pub struct CommonWinningIndexReturn {
    pub amount: u64,
    pub winning_config_type: WinningConfigType,
    pub winning_config_item_index: Option<usize>,
}
metaplex/program/src/state.rs:124:1: 128:2
    pub struct PrintingV2CalculationCheckReturn {
    pub expected_redemptions: u64,
    pub winning_config_type: WinningConfigType,
    pub winning_config_item_index: Option<usize>,
}
metaplex/program/src/state.rs:849:1: 853:2
    pub struct AmountCumulativeReturn {
    pub amount: u64,
    pub cumulative_amount: u64,
    pub total_amount: u64,
}
metaplex/program/src/state.rs:101:1: 106:2
    pub struct CommonWinningIndexChecks<'a> {
    pub safety_deposit_info: &'a AccountInfo<'a>,
    pub winning_index: usize,
    pub auction_manager_v1_ignore_claim: bool,
    pub safety_deposit_config_info: Option<&'a AccountInfo<'a>>,
}
metaplex/program/src/deprecated_state.rs:557:1: 575:2
    pub struct ParticipationStateV1 {
    /// We have this variable below to keep track in the case of the
     ↪ participation NFTs, whose
    /// income will trickle in over time, how much the artists have in the
     ↪ escrow account and
    /// how much would/should be owed to them if they try to claim it
     ↪ relative to the winning bids.
    /// It's  abit tougher than a straightforward bid which has a price
     ↪ attached to it, because
    /// there are many bids of differing amounts (in the case of
     ↪ GivenForBidPrice) and they dont all
    /// come in at one time, so this little ledger here keeps track.
    pub collected_to_accept_payment: u64,

    /// Record of primary sale or not at time of auction creation, set
     ↪ during validation step
    pub primary_sale_happened: bool,
    pub validated: bool,

    /// NOTE: DEPRECATED.
    /// An account for printing authorization tokens that are made with the one time use token
    /// after the auction ends. Provided during validation step.
    pub printing_authorization_token_account: Option<Pubkey>,
}
metaplex/program/src/state.rs:108:1: 116:2
    pub struct PrintingV2CalculationChecks<'a> {
    pub safety_deposit_info: &'a AccountInfo<'a>,
    pub winning_index: usize,
    pub auction_manager_v1_ignore_claim: bool,
    pub safety_deposit_config_info: Option<&'a AccountInfo<'a>>,
    pub short_circuit_total: bool,
    pub edition_offset: u64,
    pub winners: usize,
}
metaplex/program/src/processor/redeem_participation_bid.rs:29:1: 39:2
    struct V2Accounts<'a> {
    pub prize_tracking_ticket_info: &'a AccountInfo<'a>,
    pub new_metadata_account_info: &'a AccountInfo<'a>,
    pub new_edition_account_info: &'a AccountInfo<'a>,
    pub master_edition_account_info: &'a AccountInfo<'a>,
    pub mint_info: &'a AccountInfo<'a>,
    pub edition_marker_info: &'a AccountInfo<'a>,
    pub mint_authority_info: &'a AccountInfo<'a>,
    pub metadata_account_info: &'a AccountInfo<'a>,
    pub auction_extended_info: &'a AccountInfo<'a>,
}
metaplex/program/src/utils.rs:683:1: 700:2
    pub struct CommonRedeemFinishArgs<'a> {
    pub program_id: &'a Pubkey,
    pub auction_manager: Box<dyn AuctionManager>,
    pub auction_manager_info: &'a AccountInfo<'a>,
    pub bidder_metadata_info: &'a AccountInfo<'a>,
    pub rent_info: &'a AccountInfo<'a>,
    pub system_info: &'a AccountInfo<'a>,
    pub payer_info: &'a AccountInfo<'a>,
    pub bid_redemption_info: &'a AccountInfo<'a>,
    pub vault_info: &'a AccountInfo<'a>,
    pub safety_deposit_config_info: Option<&'a AccountInfo<'a>>,
    pub winning_index: Option<usize>,
    pub redemption_bump_seed: u8,
    pub bid_redeemed: bool,
    pub participation_redeemed: bool,
    pub winning_item_index: Option<usize>,
    pub overwrite_win_index: Option<usize>,
}
metaplex/program/src/processor/validate_safety_deposit_box_v2.rs:185:1: 203:2
    pub struct SupplyLogicCheckArgs<'a, 'b> {
    pub program_id: &'a Pubkey,
    pub auction_manager_info: &'a AccountInfo<'a>,
    pub metadata_info: &'a AccountInfo<'a>,
    pub edition_info: &'a AccountInfo<'a>,
    pub metadata_authority_info: &'a AccountInfo<'a>,
    pub original_authority_lookup_info: &'a AccountInfo<'a>,
    pub rent_info: &'a AccountInfo<'a>,
    pub system_info: &'a AccountInfo<'a>,
    pub payer_info: &'a AccountInfo<'a>,
    pub token_metadata_program_info: &'a AccountInfo<'a>,
    pub safety_deposit_token_store_info: &'a AccountInfo<'a>,
    pub auction_manager: &'b dyn AuctionManager,
    pub winning_config_type: &'b WinningConfigType,
    pub metadata: &'b Metadata,
    pub safety_deposit: &'b SafetyDepositBox,
    pub store: &'b Store,
    pub total_amount_requested: u64,
}
metaplex/program/src/processor/validate_safety_deposit_box_v2.rs:69:1: 89:2
    pub struct CommonCheckArgs<'a, 'b> {
    pub program_id: &'a Pubkey,
    pub auction_manager_info: &'a AccountInfo<'a>,
    pub metadata_info: &'a AccountInfo<'a>,
    pub original_authority_lookup_info: &'a AccountInfo<'a>,
    pub whitelisted_creator_info: &'a AccountInfo<'a>,
    pub safety_deposit_info: &'a AccountInfo<'a>,
    pub safety_deposit_token_store_info: &'a AccountInfo<'a>,
    pub edition_info: &'a AccountInfo<'a>,
    pub vault_info: &'a AccountInfo<'a>,
    pub mint_info: &'a AccountInfo<'a>,
    pub token_metadata_program_info: &'a AccountInfo<'a>,
    pub auction_manager_store_info: &'a AccountInfo<'a>,
    pub authority_info: &'a AccountInfo<'a>,
    pub store: &'b Store,
    pub auction_manager: &'b dyn AuctionManager,
    pub metadata: &'b Metadata,
    pub safety_deposit: &'b SafetyDepositBox,
    pub vault: &'b Vault,
    pub winning_config_type: &'b WinningConfigType,
}
metaplex/program/src/utils.rs:389:1: 420:2
    pub struct CommonRedeemCheckArgs<'a> {
    pub program_id: &'a Pubkey,
    pub auction_manager_info: &'a AccountInfo<'a>,
    pub safety_deposit_token_store_info: &'a AccountInfo<'a>,
    pub destination_info: &'a AccountInfo<'a>,
    pub bid_redemption_info: &'a AccountInfo<'a>,
    pub safety_deposit_info: &'a AccountInfo<'a>,
    pub vault_info: &'a AccountInfo<'a>,
    pub auction_info: &'a AccountInfo<'a>,
    pub auction_extended_info: Option<&'a AccountInfo<'a>>,
    pub bidder_metadata_info: &'a AccountInfo<'a>,
    pub bidder_info: &'a AccountInfo<'a>,
    pub token_program_info: &'a AccountInfo<'a>,
    pub token_vault_program_info: &'a AccountInfo<'a>,
    pub token_metadata_program_info: &'a AccountInfo<'a>,
    pub store_info: &'a AccountInfo<'a>,
    pub rent_info: &'a AccountInfo<'a>,
    pub safety_deposit_config_info: Option<&'a AccountInfo<'a>>,
    pub is_participation: bool,
    // If this is being called by the auctioneer to pull prizes out they overwrite the win index
    // they would normally get if they themselves bid for whatever win index they choose.
    pub overwrite_win_index: Option<usize>,
    // In newer endpoints, to conserve CPU and make way for 10,000 person auctions,
    // client must specify win index and then we simply check if the address matches for O(1) lookup vs O(n)
    // scan. This is an option so older actions which rely on the O(n) lookup because we can't change their call structure
    // can continue to work.
    pub user_provided_win_index: Option<Option<usize>>,
    pub assert_bidder_signer: bool,
    // For printing v2, the edition pda is what essentially forms a
    ↪  backstop for bad bidders. We do not need this additional
    // check which isn't accurate anyway when one winning config item has
    ↪  an amount > 1.
    pub ignore_bid_redeemed_item_check: bool,
}

```

- Call Stack

```
UnResolved
```

- description:
- link:
- alleviation:

Issue: 24: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

metaplex/program/src/deprecated_state.rs:600:1: 604:2

```
pub struct WinningConfig {
    // For now these are just array-of-array proxies but wanted to make them first class
    // structs in case we want to attach other top level metadata someday.
    pub items: Vec<WinningConfigItem>,
}
metaplex/program/src/instruction.rs:75:1: 79:2
    pub struct EndAuctionArgs {
    /// If the auction was blinded, a revealing price must be specified to release the auction
    /// winnings.
    pub reveal: Option<(u64, u64)>,
}
metaplex/program/src/deprecated_state.rs:545:1: 553:2
    pub struct AuctionManagerSettingsV1 {
    /// The safety deposit box index in the vault containing the winning items, in order of place
    /// The same index can appear multiple times if that index contains n tokens for n appearances (this will be checked)
    pub winning_configs: Vec<WinningConfig>,

    /// The participation config is separated because it is structurally a bit different,
    /// having different options and also because it has no real "winning place" in the array.
    pub participation_config: Option<ParticipationConfigV1>,
}
metaplex/program/src/deprecated_state.rs:608:1: 612:2
    pub struct WinningConfigState {
    pub items: Vec<WinningConfigStateItem>,
    /// Ticked to true when money is pushed to accept_payment account from auction bidding pot
    pub money_pushed_to_accept_payment: bool,
}

```

- Call Stack

```
UnResolved
```

- description:
- link:
- alleviation:

Issue: 25: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

metaplex/program/src/instruction.rs:13:1: 15:2

```
pub struct SetStoreArgs {
    pub public: bool,
}
metaplex/program/src/instruction.rs:24:1: 26:2
    pub struct SetWhitelistedCreatorArgs {
    pub activated: bool,
}
metaplex/program/src/deprecated_state.rs:624:1: 629:2
    pub struct WinningConfigStateItem {
    /// Record of primary sale or not at time of auction creation, set during validation step
    pub primary_sale_happened: bool,
    /// Ticked to true when a prize is claimed by person who won it
    pub claimed: bool,
}
metaplex/program/src/instruction.rs:18:1: 21:2
    pub struct SetStoreV2Args {
    pub public: bool,
    pub settings_uri: Option<String>,
}
metaplex/program/src/instruction.rs:46:1: 49:2
    pub struct RedeemUnusedWinningConfigItemsAsAuctioneerArgs {
    pub winning_config_item_index: u8,
    pub proxy_call: ProxyCallAddress,
}
metaplex/program/src/deprecated_state.rs:616:1: 620:2
    pub struct WinningConfigItem {
    pub safety_deposit_box_index: u8,
    pub amount: u8,
    pub winning_config_type: WinningConfigType,
}
```

```
metaplex/program/src/utils.rs:380:1: 387:2
    pub struct CommonRedeemReturn {
    pub redemption_bump_seed: u8,
    pub auction_manager: Box<dyn AuctionManager>,
    pub cancelled: bool,
    pub rent: Rent,
    pub win_index: Option<usize>,
    pub token_metadata_program: Pubkey,
}

```

- Call Stack

```
UnResolved
```

- description:
- link:
- alleviation:

Issue: 26: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

metaplex/program/src/instruction.rs:24:1: 26:2

```
pub struct SetWhitelistedCreatorArgs {
    pub activated: bool,
}
metaplex/program/src/deprecated_state.rs:624:1: 629:2
    pub struct WinningConfigStateItem {
    /// Record of primary sale or not at time of auction creation, set during validation step
    pub primary_sale_happened: bool,
    /// Ticked to true when a prize is claimed by person who won it
    pub claimed: bool,
}
metaplex/program/src/instruction.rs:18:1: 21:2
    pub struct SetStoreV2Args {
    pub public: bool,
    pub settings_uri: Option<String>,
}
metaplex/program/src/instruction.rs:46:1: 49:2
    pub struct RedeemUnusedWinningConfigItemsAsAuctioneerArgs {
    pub winning_config_item_index: u8,
    pub proxy_call: ProxyCallAddress,
}
metaplex/program/src/deprecated_state.rs:616:1: 620:2
    pub struct WinningConfigItem {
    pub safety_deposit_box_index: u8,
    pub amount: u8,
    pub winning_config_type: WinningConfigType,
}
metaplex/program/src/utils.rs:380:1: 387:2
    pub struct CommonRedeemReturn {
    pub redemption_bump_seed: u8,
    pub auction_manager: Box<dyn AuctionManager>,
    pub cancelled: bool,
    pub rent: Rent,
    pub win_index: Option<usize>,
    pub token_metadata_program: Pubkey,
}

```

- Call Stack

```
UnResolved
```

- description:
- link:
- alleviation:

Issue: 27: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

metaplex/program/src/instruction.rs:75:1: 79:2

```
pub struct EndAuctionArgs {
    /// If the auction was blinded, a revealing price must be specified to release the auction
    /// winnings.
    pub reveal: Option<(u64, u64)>,
}
metaplex/program/src/deprecated_state.rs:545:1: 553:2
    pub struct AuctionManagerSettingsV1 {
    /// The safety deposit box index in the vault containing the winning items, in order of place
    /// The same index can appear multiple times if that index contains n tokens for n appearances (this will be checked)
    pub winning_configs: Vec<WinningConfig>,

    /// The participation config is separated because it is structurally a bit different,
    /// having different options and also because it has no real "winning place" in the array.
    pub participation_config: Option<ParticipationConfigV1>,
}
metaplex/program/src/deprecated_state.rs:608:1: 612:2
    pub struct WinningConfigState {
    pub items: Vec<WinningConfigStateItem>,
    /// Ticked to true when money is pushed to accept_payment account from auction bidding pot
    pub money_pushed_to_accept_payment: bool,
}

```

- Call Stack

```
UnResolved
```

- description:
- link:
- alleviation:

Issue: 28: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

metaplex/program/src/state.rs:558:1: 566:2

```
pub struct ParticipationStateV2 {
    /// We have this variable below to keep track in the case of the participation NFTs, whose
    /// income will trickle in over time, how much the artists have in the escrow account and
    /// how much would/should be owed to them if they try to claim it relative to the winning bids.
    /// It's  abit tougher than a straightforward bid which has a price attached to it, because
    /// there are many bids of differing amounts (in the case of GivenForBidPrice) and they dont all
    /// come in at one time, so this little ledger here keeps track.
    pub collected_to_accept_payment: u64,
}
metaplex/program/src/instruction.rs:52:1: 55:2
    pub struct RedeemPrintingV2BidArgs {
    pub edition_offset: u64,
    pub win_index: u64,
}
metaplex/program/src/instruction.rs:82:1: 85:2
    pub struct SetStoreIndexArgs {
    pub page: u64,
    pub offset: u64,
}
metaplex/program/src/state.rs:805:1: 805:42
    pub struct AmountRange(pub u64, pub u64);
metaplex/program/src/state.rs:118:1: 122:2
    pub struct CommonWinningIndexReturn {
    pub amount: u64,
    pub winning_config_type: WinningConfigType,
    pub winning_config_item_index: Option<usize>,
}
metaplex/program/src/state.rs:124:1: 128:2
    pub struct PrintingV2CalculationCheckReturn {
    pub expected_redemptions: u64,
    pub winning_config_type: WinningConfigType,
    pub winning_config_item_index: Option<usize>,
}
metaplex/program/src/state.rs:849:1: 853:2
    pub struct AmountCumulativeReturn {
    pub amount: u64,
    pub cumulative_amount: u64,
    pub total_amount: u64,
}
metaplex/program/src/state.rs:101:1: 106:2
    pub struct CommonWinningIndexChecks<'a> {
    pub safety_deposit_info: &'a AccountInfo<'a>,
    pub winning_index: usize,
    pub auction_manager_v1_ignore_claim: bool,
    pub safety_deposit_config_info: Option<&'a AccountInfo<'a>>,
}
metaplex/program/src/deprecated_state.rs:557:1: 575:2
    pub struct ParticipationStateV1 {
    /// We have this variable below to keep track in the case of the participation NFTs, whose
    /// income will trickle in over time, how much the artists have in the escrow account and
    /// how much would/should be owed to them if they try to claim it relative to the winning bids.
    /// It's  abit tougher than a straightforward bid which has a price attached to it, because
    /// there are many bids of differing amounts (in the case of GivenForBidPrice) and they dont all
    /// come in at one time, so this little ledger here keeps track.
    pub collected_to_accept_payment: u64,

    /// Record of primary sale or not at time of auction creation, set during validation step
    pub primary_sale_happened: bool,

    pub validated: bool,

    /// NOTE: DEPRECATED.
    /// An account for printing authorization tokens that are made with the one time use token
    /// after the auction ends. Provided during validation step.
    pub printing_authorization_token_account: Option<Pubkey>,
}
metaplex/program/src/state.rs:108:1: 116:2
    pub struct PrintingV2CalculationChecks<'a> {
    pub safety_deposit_info: &'a AccountInfo<'a>,
    pub winning_index: usize,
    pub auction_manager_v1_ignore_claim: bool,
    pub safety_deposit_config_info: Option<&'a AccountInfo<'a>>,
    pub short_circuit_total: bool,
    pub edition_offset: u64,
    pub winners: usize,
}
metaplex/program/src/processor/redeem_participation_bid.rs:29:1: 39:2
    struct V2Accounts<'a> {
    pub prize_tracking_ticket_info: &'a AccountInfo<'a>,
    pub new_metadata_account_info: &'a AccountInfo<'a>,
    pub new_edition_account_info: &'a AccountInfo<'a>,
    pub master_edition_account_info: &'a AccountInfo<'a>,
    pub mint_info: &'a AccountInfo<'a>,
    pub edition_marker_info: &'a AccountInfo<'a>,
    pub mint_authority_info: &'a AccountInfo<'a>,
    pub metadata_account_info: &'a AccountInfo<'a>,
    pub auction_extended_info: &'a AccountInfo<'a>,
}
metaplex/program/src/utils.rs:683:1: 700:2
    pub struct CommonRedeemFinishArgs<'a> {
    pub program_id: &'a Pubkey,
    pub auction_manager: Box<dyn AuctionManager>,
    pub auction_manager_info: &'a AccountInfo<'a>,
    pub bidder_metadata_info: &'a AccountInfo<'a>,
    pub rent_info: &'a AccountInfo<'a>,
    pub system_info: &'a AccountInfo<'a>,
    pub payer_info: &'a AccountInfo<'a>,
    pub bid_redemption_info: &'a AccountInfo<'a>,
    pub vault_info: &'a AccountInfo<'a>,
    pub safety_deposit_config_info: Option<&'a AccountInfo<'a>>,
    pub winning_index: Option<usize>,
    pub redemption_bump_seed: u8,
    pub bid_redeemed: bool,
    pub participation_redeemed: bool,
    pub winning_item_index: Option<usize>,
    pub overwrite_win_index: Option<usize>,
}
metaplex/program/src/processor/validate_safety_deposit_box_v2.rs:185:1: 203:2
    pub struct SupplyLogicCheckArgs<'a, 'b> {
    pub program_id: &'a Pubkey,
    pub auction_manager_info: &'a AccountInfo<'a>,
    pub metadata_info: &'a AccountInfo<'a>,
    pub edition_info: &'a AccountInfo<'a>,
    pub metadata_authority_info: &'a AccountInfo<'a>,
    pub original_authority_lookup_info: &'a AccountInfo<'a>,
    pub rent_info: &'a AccountInfo<'a>,
    pub system_info: &'a AccountInfo<'a>,
    pub payer_info: &'a AccountInfo<'a>,
    pub token_metadata_program_info: &'a AccountInfo<'a>,
    pub safety_deposit_token_store_info: &'a AccountInfo<'a>,
    pub auction_manager: &'b dyn AuctionManager,
    pub winning_config_type: &'b WinningConfigType,
    pub metadata: &'b Metadata,
    pub safety_deposit: &'b SafetyDepositBox,
    pub store: &'b Store,
    pub total_amount_requested: u64,
}
metaplex/program/src/processor/validate_safety_deposit_box_v2.rs:69:1: 89:2
    pub struct CommonCheckArgs<'a, 'b> {
    pub program_id: &'a Pubkey,
    pub auction_manager_info: &'a AccountInfo<'a>,
    pub metadata_info: &'a AccountInfo<'a>,
    pub original_authority_lookup_info: &'a AccountInfo<'a>,
    pub whitelisted_creator_info: &'a AccountInfo<'a>,
    pub safety_deposit_info: &'a AccountInfo<'a>,
    pub safety_deposit_token_store_info: &'a AccountInfo<'a>,
    pub edition_info: &'a AccountInfo<'a>,
    pub vault_info: &'a AccountInfo<'a>,
    pub mint_info: &'a AccountInfo<'a>,
    pub token_metadata_program_info: &'a AccountInfo<'a>,
    pub auction_manager_store_info: &'a AccountInfo<'a>,
    pub authority_info: &'a AccountInfo<'a>,
    pub store: &'b Store,
    pub auction_manager: &'b dyn AuctionManager,
    pub metadata: &'b Metadata,
    pub safety_deposit: &'b SafetyDepositBox,
    pub vault: &'b Vault,
    pub winning_config_type: &'b WinningConfigType,
}
metaplex/program/src/utils.rs:389:1: 420:2
    pub struct CommonRedeemCheckArgs<'a> {
    pub program_id: &'a Pubkey,
    pub auction_manager_info: &'a AccountInfo<'a>,
    pub safety_deposit_token_store_info: &'a AccountInfo<'a>,
    pub destination_info: &'a AccountInfo<'a>,
    pub bid_redemption_info: &'a AccountInfo<'a>,
    pub safety_deposit_info: &'a AccountInfo<'a>,
    pub vault_info: &'a AccountInfo<'a>,
    pub auction_info: &'a AccountInfo<'a>,
    pub auction_extended_info: Option<&'a AccountInfo<'a>>,
    pub bidder_metadata_info: &'a AccountInfo<'a>,
    pub bidder_info: &'a AccountInfo<'a>,
    pub token_program_info: &'a AccountInfo<'a>,
    pub token_vault_program_info: &'a AccountInfo<'a>,
    pub token_metadata_program_info: &'a AccountInfo<'a>,
    pub store_info: &'a AccountInfo<'a>,
    pub rent_info: &'a AccountInfo<'a>,
    pub safety_deposit_config_info: Option<&'a AccountInfo<'a>>,
    pub is_participation: bool,
    // If this is being called by the auctioneer to pull prizes out they
     ↪  overwrite the win index
    // they would normally get if they themselves bid for whatever win
     ↪  index they choose.
    pub overwrite_win_index: Option<usize>,
    // In newer endpoints, to conserve CPU and make way for 10,000 person
     ↪  auctions,
    // client must specify win index and then we simply check if the
     ↪  address matches for O(1) lookup vs O(n)
    // scan. This is an option so older actions which rely on the O(n)
     ↪  lookup because we can't change their call structure
    // can continue to work.
    pub user_provided_win_index: Option<Option<usize>>,
    pub assert_bidder_signer: bool,
    // For printing v2, the edition pda is what essentially forms a
     ↪  backstop for bad bidders. We do not need this additional
    // check which isn't accurate anyway when one winning config item has
     ↪  an amount > 1.
    pub ignore_bid_redeemed_item_check: bool,
}

```

- Call Stack

```
UnResolved
```

- description:
- link:
- alleviation:

Issue: 29: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

metaplex/program/src/state.rs:1469:1: 1473:2

```
pub struct BidRedemptionTicket {
    // With BidRedemptionTicket is easier to hide it's legacy V1/V2 behind an internal facade,
    // since all of it's values are read directly off the array.
    pub key: Key,
}
metaplex/program/src/deprecated_state.rs:634:1: 637:2
    pub struct SafetyDepositValidationTicket {
    pub key: Key,
    pub address: Pubkey,
}
metaplex/program/src/state.rs:643:1: 646:2
    pub struct OriginalAuthorityLookup {
    pub key: Key,
    pub original_authority: Pubkey,
}
metaplex/program/src/state.rs:745:1: 748:2
    pub struct StoreConfig {
    pub key: Key,
    pub settings_uri: Option<String>,
}
metaplex/program/src/instruction.rs:63:1: 73:2
    pub struct InitAuctionManagerV2Args {
    pub amount_type: TupleNumericType,
    pub length_type: TupleNumericType,
    // how many ranges you can store in the AuctionWinnerTokenTypeTracker. For a limited edition single, you really
    // only need 1, for more complex auctions you may need more. Feel free to scale this
    // with the complexity of your auctions - this thing stores a range of how many unique token types
    // each range of people gets in the most efficient compressed way possible, but if you don't
    // give a high enough list length, while you may save space, you may also blow out your struct size while performing
    // validation and have a failed auction.
    pub max_ranges: u64,
}
metaplex/program/src/state.rs:570:1: 585:2
    pub struct ParticipationConfigV2 {
    /// Setups:
    /// 1. Winners get participation + not charged extra
    /// 2. Winners dont get participation prize
    pub winner_constraint: WinningConstraint,

    /// Setups:
    /// 1. Losers get prize for free
    /// 2. Losers get prize but pay fixed price
    /// 3. Losers get prize but pay bid price
    pub non_winning_constraint: NonWinningConstraint,

    /// Setting this field disconnects the participation prizes price from the bid. Any bid you submit, regardless
    /// of amount, charges you the same fixed price.
    pub fixed_price: Option<u64>,
}
metaplex/program/src/state.rs:662:1: 666:2
    pub struct PayoutTicket {
    pub key: Key,
    pub recipient: Pubkey,
    pub amount_paid: u64,
}
metaplex/program/src/state.rs:763:1: 767:2
    pub struct WhitelistedCreator {
    pub key: Key,
    pub address: Pubkey,
    pub activated: bool,
}
metaplex/program/src/deprecated_state.rs:533:1: 541:2
    pub struct AuctionManagerStateV1 {
    pub status: AuctionManagerStatus,
    /// When all configs are validated the auction is started and auction manager moves to Running
    pub winning_config_items_validated: u8,

    pub winning_config_states: Vec<WinningConfigState>,

    pub participation_state: Option<ParticipationStateV1>,
}
metaplex/program/src/deprecated_state.rs:579:1: 596:2
    pub struct ParticipationConfigV1 {
    /// Setups:
    /// 1. Winners get participation + not charged extra
    /// 2. Winners dont get participation prize
    pub winner_constraint: WinningConstraint,

    /// Setups:
    /// 1. Losers get prize for free
    /// 2. Losers get prize but pay fixed price
    /// 3. Losers get prize but pay bid price
    pub non_winning_constraint: NonWinningConstraint,

    /// The safety deposit box index in the vault containing the template
     ↪ for the participation prize
    pub safety_deposit_box_index: u8,
    /// Setting this field disconnects the participation prizes price from
     ↪ the bid. Any bid you submit, regardless
    /// of amount, charges you the same fixed price.
    pub fixed_price: Option<u64>,
}
metaplex/program/src/state.rs:546:1: 554:2
    pub struct AuctionManagerStateV2 {
    pub status: AuctionManagerStatus,
    /// When all configs are validated the auction is started and auction
     ↪ manager moves to Running
    pub safety_config_items_validated: u64,
    /// how many bids have been pushed to accept payment
    pub bids_pushed_to_accept_payment: u64,

    pub has_participation: bool,
}
metaplex/program/src/state.rs:682:1: 687:2
    pub struct StoreIndexer {
    pub key: Key,
    pub store: Pubkey,
    pub page: u64,
    pub auction_caches: Vec<Pubkey>,
}
metaplex/program/src/state.rs:1267:1: 1273:2
    pub struct AuctionWinnerTokenTypeTracker {
    pub key: Key,
    pub amount_type: TupleNumericType,
    pub length_type: TupleNumericType,
    /// Tuple is (amount of editions or tokens given to people in this
     ↪ range, length of range)
    pub amount_ranges: Vec<AmountRange>,
}
metaplex/program/src/state.rs:783:1: 789:2
    pub struct PrizeTrackingTicket {
    pub key: Key,
    pub metadata: Pubkey,
    pub supply_snapshot: u64,
    pub expected_redemptions: u64,
    pub redemptions: u64,
}
metaplex/program/src/state.rs:726:1: 733:2
    pub struct Store {
    pub key: Key,
    pub public: bool,
    pub auction_program: Pubkey,
    pub token_vault_program: Pubkey,
    pub token_metadata_program: Pubkey,
    pub token_program: Pubkey,
}
metaplex/program/src/state.rs:224:1: 238:2
    pub struct AuctionManagerV2 {
    pub key: Key,

    pub store: Pubkey,

    pub authority: Pubkey,

    pub auction: Pubkey,

    pub vault: Pubkey,

    pub accept_payment: Pubkey,

    pub state: AuctionManagerStateV2,
}
metaplex/program/src/state.rs:703:1: 711:2
    pub struct AuctionCache {
    pub key: Key,
    pub store: Pubkey,
    pub timestamp: i64,
    pub metadata: Vec<Pubkey>,
    pub auction: Pubkey,
    pub vault: Pubkey,
    pub auction_manager: Pubkey,
}
metaplex/program/src/deprecated_state.rs:57:1: 76:2
    pub struct AuctionManagerV1 {
    pub key: Key,

    pub store: Pubkey,

    pub authority: Pubkey,

    pub auction: Pubkey,

    pub vault: Pubkey,

    pub accept_payment: Pubkey,

    pub state: AuctionManagerStateV1,

    pub settings: AuctionManagerSettingsV1,

    /// True if this is only winning configs of one item each, used for optimization in saving.
    pub straight_shot_optimization: bool,
}
metaplex/program/src/state.rs:831:1: 847:2
    pub struct SafetyDepositConfig {
    pub key: Key,
    /// reverse lookup
    pub auction_manager: Pubkey,
    // only 255 safety deposits on vault right now but soon this will likely expand.
    /// safety deposit order
    pub order: u64,
    pub winning_config_type: WinningConfigType,
    pub amount_type: TupleNumericType,
    pub length_type: TupleNumericType,
    /// Tuple is (amount of editions or tokens given to people in this range, length of range)
    pub amount_ranges: Vec<AmountRange>,
    /// if winning config type is "Participation" then you use this to parameterize it.
    pub participation_config: Option<ParticipationConfigV2>,
    /// if winning config type is "Participation" then you use this to keep track of it.
    pub participation_state: Option<ParticipationStateV2>,
}

```

- Call Stack

```
UnResolved
```

- description:
- link:
- alleviation:

Issue: 30: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

metaplex/program/src/deprecated_state.rs:624:1: 629:2

```
pub struct WinningConfigStateItem {
    /// Record of primary sale or not at time of auction creation, set
     ↪  during validation step
    pub primary_sale_happened: bool,
    /// Ticked to true when a prize is claimed by person who won it
    pub claimed: bool,
}
metaplex/program/src/instruction.rs:46:1: 49:2
    pub struct RedeemUnusedWinningConfigItemsAsAuctioneerArgs {
    pub winning_config_item_index: u8,
    pub proxy_call: ProxyCallAddress,
}
metaplex/program/src/deprecated_state.rs:616:1: 620:2
    pub struct WinningConfigItem {
    pub safety_deposit_box_index: u8,
    pub amount: u8,
    pub winning_config_type: WinningConfigType,
}

```

- Call Stack

```
UnResolved
```

- description:
- link:
- alleviation:

Issue: 31: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

metaplex/program/src/deprecated_state.rs:634:1: 637:2

```
pub struct SafetyDepositValidationTicket {
    pub key: Key,
    pub address: Pubkey,
}
metaplex/program/src/state.rs:643:1: 646:2
    pub struct OriginalAuthorityLookup {
    pub key: Key,
    pub original_authority: Pubkey,
}
metaplex/program/src/state.rs:662:1: 666:2
    pub struct PayoutTicket {
    pub key: Key,
    pub recipient: Pubkey,
    pub amount_paid: u64,
}
metaplex/program/src/state.rs:763:1: 767:2
    pub struct WhitelistedCreator {
    pub key: Key,
    pub address: Pubkey,
    pub activated: bool,
}
metaplex/program/src/state.rs:682:1: 687:2
    pub struct StoreIndexer {
    pub key: Key,
    pub store: Pubkey,
    pub page: u64,
    pub auction_caches: Vec<Pubkey>,
}
metaplex/program/src/state.rs:783:1: 789:2
    pub struct PrizeTrackingTicket {
    pub key: Key,
    pub metadata: Pubkey,
    pub supply_snapshot: u64,
    pub expected_redemptions: u64,
    pub redemptions: u64,
}
metaplex/program/src/state.rs:224:1: 238:2
    pub struct AuctionManagerV2 {
    pub key: Key,

    pub store: Pubkey,

    pub authority: Pubkey,

    pub auction: Pubkey,

    pub vault: Pubkey,

    pub accept_payment: Pubkey,

    pub state: AuctionManagerStateV2,
}
metaplex/program/src/state.rs:703:1: 711:2
    pub struct AuctionCache {
    pub key: Key,
    pub store: Pubkey,
    pub timestamp: i64,
    pub metadata: Vec<Pubkey>,
    pub auction: Pubkey,
    pub vault: Pubkey,
    pub auction_manager: Pubkey,
}
metaplex/program/src/deprecated_state.rs:57:1: 76:2
    pub struct AuctionManagerV1 {
    pub key: Key,

    pub store: Pubkey,

    pub authority: Pubkey,

    pub auction: Pubkey,

    pub vault: Pubkey,

    pub accept_payment: Pubkey,

    pub state: AuctionManagerStateV1,

    pub settings: AuctionManagerSettingsV1,

    /// True if this is only winning configs of one item each, used for optimization in saving.
    pub straight_shot_optimization: bool,
}
metaplex/program/src/state.rs:831:1: 847:2
    pub struct SafetyDepositConfig {
    pub key: Key,
    /// reverse lookup
    pub auction_manager: Pubkey,
    // only 255 safety deposits on vault right now but soon this will likely expand.
    /// safety deposit order
    pub order: u64,
    pub winning_config_type: WinningConfigType,
    pub amount_type: TupleNumericType,
    pub length_type: TupleNumericType,
    /// Tuple is (amount of editions or tokens given to people in this range, length of range)
    pub amount_ranges: Vec<AmountRange>,
    /// if winning config type is "Participation" then you use this to parameterize it.
    pub participation_config: Option<ParticipationConfigV2>,
    /// if winning config type is "Participation" then you use this to keep track of it.
    pub participation_state: Option<ParticipationStateV2>,
}

```

- Call Stack

```
UnResolved
```

- description:
- link:

- alleviation:

Issue: 32: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

metaplex/program/src/instruction.rs:46:1: 49:2

```
pub struct RedeemUnusedWinningConfigItemsAsAuctioneerArgs {
    pub winning_config_item_index: u8,
    pub proxy_call: ProxyCallAddress,
}
metaplex/program/src/deprecated_state.rs:616:1: 620:2
    pub struct WinningConfigItem {
    pub safety_deposit_box_index: u8,
    pub amount: u8,
    pub winning_config_type: WinningConfigType,
}

```

- Call Stack

```
UnResolved
```

- description:
- link:
- alleviation:

Issue: 33: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

metaplex/program/src/instruction.rs:52:1: 55:2

```
pub struct RedeemPrintingV2BidArgs {
    pub edition_offset: u64,
    pub win_index: u64,
}
metaplex/program/src/instruction.rs:82:1: 85:2
    pub struct SetStoreIndexArgs {
    pub page: u64,
    pub offset: u64,
}
metaplex/program/src/state.rs:805:1: 805:42
    pub struct AmountRange(pub u64, pub u64);
metaplex/program/src/state.rs:849:1: 853:2
    pub struct AmountCumulativeReturn {
    pub amount: u64,
    pub cumulative_amount: u64,
    pub total_amount: u64,
}
metaplex/program/src/state.rs:101:1: 106:2
    pub struct CommonWinningIndexChecks<'a> {
    pub safety_deposit_info: &'a AccountInfo<'a>,
    pub winning_index: usize,
    pub auction_manager_v1_ignore_claim: bool,
    pub safety_deposit_config_info: Option<&'a AccountInfo<'a>>,
}
metaplex/program/src/state.rs:108:1: 116:2
    pub struct PrintingV2CalculationChecks<'a> {
    pub safety_deposit_info: &'a AccountInfo<'a>,
    pub winning_index: usize,
    pub auction_manager_v1_ignore_claim: bool,
    pub safety_deposit_config_info: Option<&'a AccountInfo<'a>>,
    pub short_circuit_total: bool,
    pub edition_offset: u64,
    pub winners: usize,
}
metaplex/program/src/processor/redeem_participation_bid.rs:29:1: 39:2
    struct V2Accounts<'a> {
    pub prize_tracking_ticket_info: &'a AccountInfo<'a>,
    pub new_metadata_account_info: &'a AccountInfo<'a>,
    pub new_edition_account_info: &'a AccountInfo<'a>,
    pub master_edition_account_info: &'a AccountInfo<'a>,
    pub mint_info: &'a AccountInfo<'a>,
    pub edition_marker_info: &'a AccountInfo<'a>,
    pub mint_authority_info: &'a AccountInfo<'a>,
    pub metadata_account_info: &'a AccountInfo<'a>,
    pub auction_extended_info: &'a AccountInfo<'a>,
}
metaplex/program/src/processor/validate_safety_deposit_box_v2.rs:185:1: 203:2
    pub struct SupplyLogicCheckArgs<'a, 'b> {
    pub program_id: &'a Pubkey,
    pub auction_manager_info: &'a AccountInfo<'a>,
    pub metadata_info: &'a AccountInfo<'a>,
    pub edition_info: &'a AccountInfo<'a>,
    pub metadata_authority_info: &'a AccountInfo<'a>,
    pub original_authority_lookup_info: &'a AccountInfo<'a>,
    pub rent_info: &'a AccountInfo<'a>,
    pub system_info: &'a AccountInfo<'a>,
    pub payer_info: &'a AccountInfo<'a>,
    pub token_metadata_program_info: &'a AccountInfo<'a>,
    pub safety_deposit_token_store_info: &'a AccountInfo<'a>,
    pub auction_manager: &'b dyn AuctionManager,
    pub winning_config_type: &'b WinningConfigType,
    pub metadata: &'b Metadata,
    pub safety_deposit: &'b SafetyDepositBox,
    pub store: &'b Store,
    pub total_amount_requested: u64,
}
metaplex/program/src/processor/validate_safety_deposit_box_v2.rs:69:1: 89:2
    pub struct CommonCheckArgs<'a, 'b> {
    pub program_id: &'a Pubkey,
    pub auction_manager_info: &'a AccountInfo<'a>,
    pub metadata_info: &'a AccountInfo<'a>,
    pub original_authority_lookup_info: &'a AccountInfo<'a>,
    pub whitelisted_creator_info: &'a AccountInfo<'a>,
    pub safety_deposit_info: &'a AccountInfo<'a>,
    pub safety_deposit_token_store_info: &'a AccountInfo<'a>,
    pub edition_info: &'a AccountInfo<'a>,
    pub vault_info: &'a AccountInfo<'a>,
    pub mint_info: &'a AccountInfo<'a>,
    pub token_metadata_program_info: &'a AccountInfo<'a>,
    pub auction_manager_store_info: &'a AccountInfo<'a>,
    pub authority_info: &'a AccountInfo<'a>,
    pub store: &'b Store,
    pub auction_manager: &'b dyn AuctionManager,
    pub metadata: &'b Metadata,
    pub safety_deposit: &'b SafetyDepositBox,
    pub vault: &'b Vault,
    pub winning_config_type: &'b WinningConfigType,
}
metaplex/program/src/utils.rs:389:1: 420:2
    pub struct CommonRedeemCheckArgs<'a> {
    pub program_id: &'a Pubkey,
    pub auction_manager_info: &'a AccountInfo<'a>,
    pub safety_deposit_token_store_info: &'a AccountInfo<'a>,
    pub destination_info: &'a AccountInfo<'a>,
    pub bid_redemption_info: &'a AccountInfo<'a>,
    pub safety_deposit_info: &'a AccountInfo<'a>,
    pub vault_info: &'a AccountInfo<'a>,
    pub auction_info: &'a AccountInfo<'a>,
    pub auction_extended_info: Option<&'a AccountInfo<'a>>,
    pub bidder_metadata_info: &'a AccountInfo<'a>,
    pub bidder_info: &'a AccountInfo<'a>,
    pub token_program_info: &'a AccountInfo<'a>,
    pub token_vault_program_info: &'a AccountInfo<'a>,
    pub token_metadata_program_info: &'a AccountInfo<'a>,
    pub store_info: &'a AccountInfo<'a>,
    pub rent_info: &'a AccountInfo<'a>,
    pub safety_deposit_config_info: Option<&'a AccountInfo<'a>>,
    pub is_participation: bool,
    // If this is being called by the auctioneer to pull prizes out they
     ↪ overwrite the win index
    // they would normally get if they themselves bid for whatever win
     ↪ index they choose.
    pub overwrite_win_index: Option<usize>,
    // In newer endpoints, to conserve CPU and make way for 10,000 person
     ↪ auctions,
    // client must specify win index and then we simply check if the address matches for O(1) lookup vs O(n)
    // scan. This is an option so older actions which rely on the O(n) lookup because we can't change their call structure
    // can continue to work.
    pub user_provided_win_index: Option<Option<usize>>,
    pub assert_bidder_signer: bool,
    // For printing v2, the edition pda is what essentially forms a backstop for bad bidders. We do not need this additional
    // check which isn't accurate anyway when one winning config item has an amount > 1.
    pub ignore_bid_redeemed_item_check: bool,
}

```

- Call Stack

```
UnResolved
```

- description:
- link:
- alleviation:

Issue: 34: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

metaplex/program/src/instruction.rs:82:1: 85:2

```
pub struct SetStoreIndexArgs {
    pub page: u64,
    pub offset: u64,
}
metaplex/program/src/state.rs:805:1: 805:42
    pub struct AmountRange(pub u64, pub u64);
metaplex/program/src/state.rs:849:1: 853:2
    pub struct AmountCumulativeReturn {
    pub amount: u64,
    pub cumulative_amount: u64,
    pub total_amount: u64,
}
metaplex/program/src/state.rs:101:1: 106:2
    pub struct CommonWinningIndexChecks<'a> {
    pub safety_deposit_info: &'a AccountInfo<'a>,
    pub winning_index: usize,
    pub auction_manager_v1_ignore_claim: bool,
    pub safety_deposit_config_info: Option<&'a AccountInfo<'a>>,
}
metaplex/program/src/state.rs:108:1: 116:2
    pub struct PrintingV2CalculationChecks<'a> {
    pub safety_deposit_info: &'a AccountInfo<'a>,
    pub winning_index: usize,
    pub auction_manager_v1_ignore_claim: bool,
    pub safety_deposit_config_info: Option<&'a AccountInfo<'a>>,
    pub short_circuit_total: bool,
    pub edition_offset: u64,
    pub winners: usize,
}
metaplex/program/src/processor/redeem_participation_bid.rs:29:1: 39:2
    struct V2Accounts<'a> {
```

```
    pub prize_tracking_ticket_info: &'a AccountInfo<'a>,
    pub new_metadata_account_info: &'a AccountInfo<'a>,
    pub new_edition_account_info: &'a AccountInfo<'a>,
    pub master_edition_account_info: &'a AccountInfo<'a>,
    pub mint_info: &'a AccountInfo<'a>,
    pub edition_marker_info: &'a AccountInfo<'a>,
    pub mint_authority_info: &'a AccountInfo<'a>,
    pub metadata_account_info: &'a AccountInfo<'a>,
    pub auction_extended_info: &'a AccountInfo<'a>,
}
metaplex/program/src/processor/validate_safety_deposit_box_v2.rs:185:1:
 ↪  203:2
    pub struct SupplyLogicCheckArgs<'a, 'b> {
    pub program_id: &'a Pubkey,
    pub auction_manager_info: &'a AccountInfo<'a>,
    pub metadata_info: &'a AccountInfo<'a>,
    pub edition_info: &'a AccountInfo<'a>,
    pub metadata_authority_info: &'a AccountInfo<'a>,
    pub original_authority_lookup_info: &'a AccountInfo<'a>,
    pub rent_info: &'a AccountInfo<'a>,
    pub system_info: &'a AccountInfo<'a>,
    pub payer_info: &'a AccountInfo<'a>,
    pub token_metadata_program_info: &'a AccountInfo<'a>,
    pub safety_deposit_token_store_info: &'a AccountInfo<'a>,
    pub auction_manager: &'b dyn AuctionManager,
    pub winning_config_type: &'b WinningConfigType,
    pub metadata: &'b Metadata,
    pub safety_deposit: &'b SafetyDepositBox,
    pub store: &'b Store,
    pub total_amount_requested: u64,
}
metaplex/program/src/processor/validate_safety_deposit_box_v2.rs:69:1: 89:2
    pub struct CommonCheckArgs<'a, 'b> {
    pub program_id: &'a Pubkey,
    pub auction_manager_info: &'a AccountInfo<'a>,
    pub metadata_info: &'a AccountInfo<'a>,
    pub original_authority_lookup_info: &'a AccountInfo<'a>,
    pub whitelisted_creator_info: &'a AccountInfo<'a>,
    pub safety_deposit_info: &'a AccountInfo<'a>,
    pub safety_deposit_token_store_info: &'a AccountInfo<'a>,
    pub edition_info: &'a AccountInfo<'a>,
    pub vault_info: &'a AccountInfo<'a>,
    pub mint_info: &'a AccountInfo<'a>,
    pub token_metadata_program_info: &'a AccountInfo<'a>,
    pub auction_manager_store_info: &'a AccountInfo<'a>,
    pub authority_info: &'a AccountInfo<'a>,
    pub store: &'b Store,
    pub auction_manager: &'b dyn AuctionManager,
    pub metadata: &'b Metadata,
    pub safety_deposit: &'b SafetyDepositBox,
    pub vault: &'b Vault,
    pub winning_config_type: &'b WinningConfigType,
}
metaplex/program/src/utils.rs:389:1: 420:2
    pub struct CommonRedeemCheckArgs<'a> {
    pub program_id: &'a Pubkey,
    pub auction_manager_info: &'a AccountInfo<'a>,
    pub safety_deposit_token_store_info: &'a AccountInfo<'a>,
    pub destination_info: &'a AccountInfo<'a>,
    pub bid_redemption_info: &'a AccountInfo<'a>,
    pub safety_deposit_info: &'a AccountInfo<'a>,
    pub vault_info: &'a AccountInfo<'a>,
    pub auction_info: &'a AccountInfo<'a>,
    pub auction_extended_info: Option<&'a AccountInfo<'a>>,
    pub bidder_metadata_info: &'a AccountInfo<'a>,
    pub bidder_info: &'a AccountInfo<'a>,
    pub token_program_info: &'a AccountInfo<'a>,
    pub token_vault_program_info: &'a AccountInfo<'a>,
    pub token_metadata_program_info: &'a AccountInfo<'a>,
    pub store_info: &'a AccountInfo<'a>,
    pub rent_info: &'a AccountInfo<'a>,
    pub safety_deposit_config_info: Option<&'a AccountInfo<'a>>,
    pub is_participation: bool,
    // If this is being called by the auctioneer to pull prizes out they overwrite the win index
    // they would normally get if they themselves bid for whatever win index they choose.
    pub overwrite_win_index: Option<usize>,
    // In newer endpoints, to conserve CPU and make way for 10,000 person auctions,
    // client must specify win index and then we simply check if the address matches for O(1) lookup vs O(n)
    // scan. This is an option so older actions which rely on the O(n) lookup because we can't change their call structure
    // can continue to work.
    pub user_provided_win_index: Option<Option<usize>>,
    pub assert_bidder_signer: bool,
    // For printing v2, the edition pda is what essentially forms a
     ↪  backstop for bad bidders. We do not need this additional
    // check which isn't accurate anyway when one winning config item has
     ↪  an amount > 1.
    pub ignore_bid_redeemed_item_check: bool,
}

```

- Call Stack

```
UnResolved
```

- description:
- link:
- alleviation:

Issue: 35: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

metaplex/program/src/state.rs:643:1: 646:2

```
pub struct OriginalAuthorityLookup {
    pub key: Key,
    pub original_authority: Pubkey,
}
metaplex/program/src/state.rs:662:1: 666:2
    pub struct PayoutTicket {
    pub key: Key,
    pub recipient: Pubkey,
    pub amount_paid: u64,
}
metaplex/program/src/state.rs:763:1: 767:2
    pub struct WhitelistedCreator {
    pub key: Key,
    pub address: Pubkey,
    pub activated: bool,
}
metaplex/program/src/state.rs:682:1: 687:2
    pub struct StoreIndexer {
    pub key: Key,
    pub store: Pubkey,
    pub page: u64,
    pub auction_caches: Vec<Pubkey>,
}
metaplex/program/src/state.rs:783:1: 789:2
    pub struct PrizeTrackingTicket {
    pub key: Key,
    pub metadata: Pubkey,
    pub supply_snapshot: u64,
    pub expected_redemptions: u64,
    pub redemptions: u64,
}
```

```
metaplex/program/src/state.rs:224:1: 238:2
    pub struct AuctionManagerV2 {
    pub key: Key,

    pub store: Pubkey,

    pub authority: Pubkey,

    pub auction: Pubkey,

    pub vault: Pubkey,

    pub accept_payment: Pubkey,

    pub state: AuctionManagerStateV2,
}
metaplex/program/src/state.rs:703:1: 711:2
    pub struct AuctionCache {
    pub key: Key,
    pub store: Pubkey,
    pub timestamp: i64,
    pub metadata: Vec<Pubkey>,
    pub auction: Pubkey,
    pub vault: Pubkey,
    pub auction_manager: Pubkey,
}
metaplex/program/src/deprecated_state.rs:57:1: 76:2
    pub struct AuctionManagerV1 {
    pub key: Key,

    pub store: Pubkey,

    pub authority: Pubkey,

    pub auction: Pubkey,

    pub vault: Pubkey,

    pub accept_payment: Pubkey,

    pub state: AuctionManagerStateV1,

    pub settings: AuctionManagerSettingsV1,

    /// True if this is only winning configs of one item each, used for
     ↪ optimization in saving.
    pub straight_shot_optimization: bool,
}
metaplex/program/src/state.rs:831:1: 847:2
    pub struct SafetyDepositConfig {
    pub key: Key,
    /// reverse lookup
    pub auction_manager: Pubkey,
    // only 255 safety deposits on vault right now but soon this will
     ↪ likely expand.
    /// safety deposit order
    pub order: u64,
    pub winning_config_type: WinningConfigType,
    pub amount_type: TupleNumericType,
    pub length_type: TupleNumericType,
    /// Tuple is (amount of editions or tokens given to people in this
     ↪ range, length of range)
    pub amount_ranges: Vec<AmountRange>,
    /// if winning config type is "Participation" then you use this to
     ↪ parameterize it.
    pub participation_config: Option<ParticipationConfigV2>,
    /// if winning config type is "Participation" then you use this to keep
     ↪ track of it.
    pub participation_state: Option<ParticipationStateV2>,
}

```

- Call Stack

```
UnResolved
```

- description:
- link:
- alleviation:

Issue: 36: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

metaplex/program/src/state.rs:662:1: 666:2

```
pub struct PayoutTicket {
    pub key: Key,
    pub recipient: Pubkey,
    pub amount_paid: u64,
}
metaplex/program/src/state.rs:682:1: 687:2
    pub struct StoreIndexer {
    pub key: Key,
    pub store: Pubkey,
    pub page: u64,
    pub auction_caches: Vec<Pubkey>,
}
metaplex/program/src/state.rs:783:1: 789:2
    pub struct PrizeTrackingTicket {
    pub key: Key,
    pub metadata: Pubkey,
    pub supply_snapshot: u64,
    pub expected_redemptions: u64,
    pub redemptions: u64,
}
metaplex/program/src/state.rs:703:1: 711:2
    pub struct AuctionCache {
    pub key: Key,
    pub store: Pubkey,
    pub timestamp: i64,
    pub metadata: Vec<Pubkey>,
    pub auction: Pubkey,
    pub vault: Pubkey,
    pub auction_manager: Pubkey,
}
metaplex/program/src/state.rs:831:1: 847:2
```

```
    pub struct SafetyDepositConfig {
    pub key: Key,
    /// reverse lookup
    pub auction_manager: Pubkey,
    // only 255 safety deposits on vault right now but soon this will likely expand.
    /// safety deposit order
    pub order: u64,
    pub winning_config_type: WinningConfigType,
    pub amount_type: TupleNumericType,
    pub length_type: TupleNumericType,
    /// Tuple is (amount of editions or tokens given to people in this range, length of range)
    pub amount_ranges: Vec<AmountRange>,
    /// if winning config type is "Participation" then you use this to parameterize it.
    pub participation_config: Option<ParticipationConfigV2>,
    /// if winning config type is "Participation" then you use this to keep track of it.
    pub participation_state: Option<ParticipationStateV2>,
}

```

- 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.