VRust

Security Assessment

O2Lab VRust Team

11/04/2022 19:56:16

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_auction
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	13
Major	0
Medium	0
Minor	0
Informational	0
Discussion	0

Findings

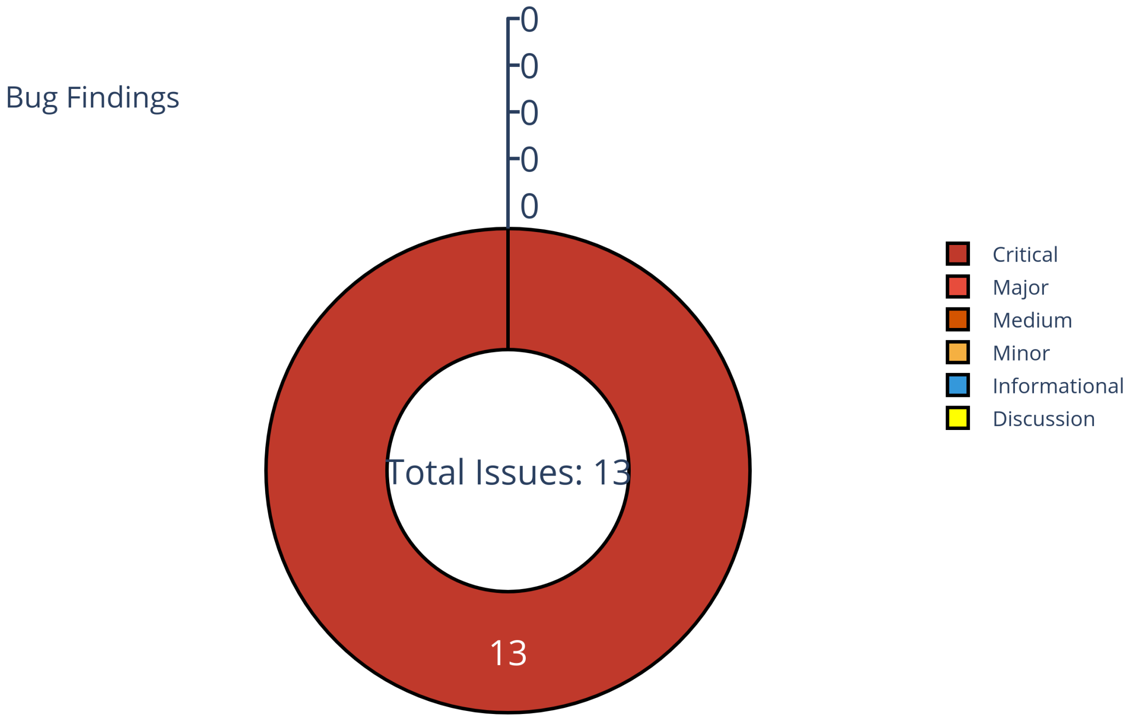

Figure 1: Findings

Finding Statistic

Category	Count
IntegerFlow	4
MissingKeyCheck	1
CrossProgramInvocation	4
TypeConfusion	4

ID	Category	Severity	Status
0	IntegerFlow	Critical	UnResolved
1	IntegerFlow	Critical	UnResolved
2	IntegerFlow	Critical	UnResolved
3	IntegerFlow	Critical	UnResolved
4	MissingKeyCheck	Critical	UnResolved
5	CrossProgramInvocation	Critical	UnResolved
6	CrossProgramInvocation	Critical	UnResolved
7	CrossProgramInvocation	Critical	UnResolved
8	CrossProgramInvocation	Critical	UnResolved
9	TypeConfusion	Critical	GitHub Link to be added.
10	TypeConfusion	Critical	GitHub Link to be added.
11	TypeConfusion	Critical	GitHub Link to be added.
12	TypeConfusion	Critical	GitHub Link to be added.

Issue: 0: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

auction/program/src/processor.rs:529:26: 529:42

```
529 (100 + gap_tick)
530
```

- Code Context

– Function Definition:

```
521 fn assert_valid_gap_insertion(
522         gap_tick: u8,
523         beaten_bid: &Bid,
524         beating_bid: &Bid,
525     ) -> ProgramResult
526
```

Vulnerability at Line: 529

```
524         beating_bid: &Bid,
525     ) -> ProgramResult {
526         // Use u128 to avoid potential overflow due to temporary mult of
             ↪  100x since
527         // we haven't divided yet.
528         let mut minimum_bid_amount: u128 = (beaten_bid.1 as u128)
529             .checked_mul((100 + gap_tick) as u128)
530             .ok_or(AuctionError::NumericalOverflowError)?;
531         minimum_bid_amount = minimum_bid_amount
532             .checked_div(100u128)
533             .ok_or(AuctionError::NumericalOverflowError)?;
534
```

- Call Stack

```
fn entrypoint::entrypoint(){// /home/yifei/.cargo/registry/src/github.com-1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10 }
    fn entrypoint::process_instruction(){// auction/program/src/entrypoint.rs:12:1: 23:2 }
        fn processor::process_instruction(){// auction/program/src/processor.rs:30:1: 48:2 }
            fn processor::place_bid::place_bid(){// auction/program/src/processor/place_bid.rs:125:1: 341:2 }
                    fn processor::AuctionData::place_bid(){// auction/program/src/processor.rs:386:5: 423:6 }
                        fn processor::BidState::place_bid(){// auction/program/src/processor.rs:545:5: 629:6 }
                            fn processor::BidState::assert_valid_gap_insertion(){// auction/program/src/processor.rs:521:5: 541:6 }

```

- description:
- link:
- alleviation:

Issue: 1: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

auction/program/src/processor.rs:499:24: 499:36

```
2 * real_max

```

- Code Context

Vulnerability at Line: 499

```
pub fn max_array_size_for(n: usize) -> usize {
        let mut real_max = n;
        if real_max < 8 {
            real_max = 8;
        } else {
            real_max = 2 * real_max
        }
        real_max
    }

```

- Call Stack

```
fn entrypoint::entrypoint(){// /home/yifei/.cargo/registry/src/github.com-1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10 }
    fn entrypoint::process_instruction(){// auction/program/src/entrypoint.rs:12:1: 23:2 }
        fn processor::process_instruction(){// auction/program/src/processor.rs:30:1: 48:2 }
            fn processor::place_bid::place_bid(){// auction/program/src/processor/place_bid.rs:125:1: 341:2 }
                    fn processor::AuctionData::place_bid(){// auction/program/src/processor.rs:386:5: 423:6 }
```

```
                        fn processor::BidState::place_bid(){//auction/program/src/processor.rs:545:5: 629:6 }
                            fn processor::BidState::max_array_size_for(){//auction/program/src/processor.rs:494:5: 502:6 }

```

- description:
- link:
- alleviation:

Issue: 2: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

auction/program/src/processor.rs:718:44: 718:61

```
718 bids.len() - *max
719
```

- Code Context

Vulnerability at Line: 718

```
713 pub fn lowest_winning_bid_is_instant_bid_price(&self, instant_sale_amount: u64) -> bool {
714         match self {
715             // In a capped auction, track the limited number of winners.
716             BidState::EnglishAuction { bids, max } => {
717                 // bids.len() - max = index of the last winner bid
718                 bids.len() >= *max && bids[bids.len() - *max].1 >= instant_sale_amount
719             }
720             _ => false,
721         }
722     }
723
```

- Call Stack

```
fn entrypoint::entrypoint(){// /home/yifei/.cargo/registry/src/github.com-1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10 }
    fn entrypoint::process_instruction(){// auction/program/src/entrypoint.rs:12:1: 23:2 }
        fn processor::process_instruction(){// auction/program/src/processor.rs:30:1: 48:2 }
            fn processor::place_bid::place_bid(){// auction/program/src/processor/place_bid.rs:125:1: 341:2 }
```

```
          fn processor::AuctionData::place_bid(){//
          ↪  auction/program/src/processor.rs:386:5: 423:6 }
            fn proces-
            ↪  sor::AuctionData::consider_instant_bid(){//
            ↪  auction/program/src/processor.rs:373:5: 384:6 }
              fn proces-
              ↪  sor::BidState::lowest_winning_bid_is_instant_bid_price()
              ↪  auction/program/src/processor.rs:713:5:
              ↪  722:6 }

```

- description:
- link:
- alleviation:

Issue: 3: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

auction/program/src/processor/create_auction.rs:97:13: 97:68

```
97  mem::size_of::<Bid>() * BidState::max_array_size_for(n)
98
```

- Code Context

Vulnerability at Line: 97

```
92          return Err(AuctionError::InvalidAuctionAccount.into());
93      }
94      // The data must be large enough to hold at least the number of
        ↪  winners.
95      let auction_size = match args.winners {
96          WinnerLimit::Capped(n) => {
97              mem::size_of::<Bid>() * BidState::max_array_size_for(n) +
                ↪  BASE_AUCTION_DATA_SIZE
98          }
99          WinnerLimit::Unlimited(_) => BASE_AUCTION_DATA_SIZE,
100     };
101
102
```

- Call Stack

```
1  fn entrypoint::entrypoint(){// /home/yifei/.cargo/registry/src/github.com-
   ↪  1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10
   ↪  }
2      fn entrypoint::process_instruction(){//
       ↪  auction/program/src/entrypoint.rs:12:1: 23:2 }
3          fn processor::process_instruction(){//
           ↪  auction/program/src/processor.rs:30:1: 48:2 }
```

```
            fn processor::create_auction::create_auction(){// auction/program/src/processor/create_auction.rs:72:1: 181:2 }

```

- description:
- link:
- alleviation:

Issue: 4: 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::entrypoint(){// /home/yifei/.cargo/registry/src/github.com-1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10 }
    fn entrypoint::process_instruction(){// auction/program/src/entrypoint.rs:12:1: 23:2 }
        fn processor::process_instruction(){// auction/program/src/processor.rs:30:1: 48:2 }
            fn processor::create_auction_v2::create_auction_v2(){// auction/program/src/processor/create_auction_v2.rs:77:1: 99:2 }
                    fn processor::create_auction::create_auction(){// auction/program/src/processor/create_auction.rs:72:1: 181:2 }
                        fn utils::create_or_allocate_account_raw(){// auction/program/src/utils.rs:90:1: 133:2 }
```

```
                                      fn
                                      ↪   solana_program::account_info::AccountInfo::<'a>::lamport
                                      ↪   /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: 5: CrossProgramInvocation

Category	Severity	Status
CrossProgramInvocation	Critical	UnResolved

- Location

```
auction/program/src/utils.rs
```

- Code Context

```rust
pub fn spl_token_create_account<'a>(params: TokenCreateAccount<'_, '_>) -> ProgramResult {
    let TokenCreateAccount {
        payer,
        mint,
        account,
        authority,
        authority_seeds,
        token_program,
        system_program,
        rent,
    } = params;
    let acct = &account.key.clone();

    create_or_allocate_account_raw(
        *token_program.key,
        &account,
        &rent,
        &system_program,
        &payer,
        spl_token::state::Account::LEN,
        authority_seeds,
    )?;
    msg!("Created account {}", acct);
    invoke_signed(
        &spl_token::instruction::initialize_account(
            &spl_token::id(),
            acct,
            mint.key,
            authority.key,
        )?,
        &[
            account,
            authority,
            mint,
            token_program,
            system_program,
            rent,
        ],
        &[authority_seeds],
    )?;

    Ok(())
}

```

- Call Stack

```
fn entrypoint::entrypoint(){// /home/yifei/.cargo/registry/src/github.com-1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10 }
    fn entrypoint::process_instruction(){// auction/program/src/entrypoint.rs:12:1: 23:2 }
        fn processor::process_instruction(){// auction/program/src/processor.rs:30:1: 48:2 }
            fn processor::place_bid::place_bid(){// auction/program/src/processor/place_bid.rs:125:1: 341:2 }
                    fn utils::spl_token_create_account(){// auction/program/src/utils.rs:216:1: 258:2 }

```

- description:
- link:
- alleviation:

Issue: 6: CrossProgramInvocation

Category	Severity	Status
CrossProgramInvocation	Critical	UnResolved

- Location

```
auction/program/src/utils.rs
```

- Code Context

```rust
pub fn spl_token_transfer(params: TokenTransferParams<'_, '_>) -> ProgramResult {
    let TokenTransferParams {
        source,
        destination,
        authority,
        token_program,
        amount,
        authority_signer_seeds,
    } = params;

    let result = invoke_signed(
        &spl_token::instruction::transfer(
            token_program.key,
            source.key,
            destination.key,
            authority.key,
            &[],
            amount,
        )?,
        &[source, destination, authority, token_program],
        &[authority_signer_seeds],
    );

    result.map_err(|_| AuctionError::TokenTransferFailed.into())
}

```

- Call Stack

```
fn entrypoint::entrypoint(){// /home/yifei/.cargo/registry/src/github.com-1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10 }
    fn entrypoint::process_instruction(){// auction/program/src/entrypoint.rs:12:1: 23:2 }
        fn processor::process_instruction(){// auction/program/src/processor.rs:30:1: 48:2 }
            fn processor::place_bid::place_bid(){// auction/program/src/processor/place_bid.rs:125:1: 341:2 }
                    fn utils::spl_token_transfer(){// auction/program/src/utils.rs:152:1: 176:2 }

```

- description:
- link:
- alleviation:

Issue: 7: CrossProgramInvocation

Category	Severity	Status
CrossProgramInvocation	Critical	UnResolved

- Location

```
auction/program/src/utils.rs
```

- Code Context

```rust
pub fn spl_token_transfer(params: TokenTransferParams<'_, '_>) -> ProgramResult {
    let TokenTransferParams {
        source,
        destination,
        authority,
        token_program,
        amount,
        authority_signer_seeds,
    } = params;

    let result = invoke_signed(
        &spl_token::instruction::transfer(
            token_program.key,
            source.key,
            destination.key,
            authority.key,
            &[],
            amount,
        )?,
        &[source, destination, authority, token_program],
        &[authority_signer_seeds],
    );

    result.map_err(|_| AuctionError::TokenTransferFailed.into())
}
```

- Call Stack

```
fn entrypoint::entrypoint(){// /home/yifei/.cargo/registry/src/github.com-1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10 }
    fn entrypoint::process_instruction(){// auction/program/src/entrypoint.rs:12:1: 23:2 }
        fn processor::process_instruction(){// auction/program/src/processor.rs:30:1: 48:2 }
            fn processor::claim_bid::claim_bid(){// auction/program/src/processor/claim_bid.rs:87:1: 214:2 }
                    fn utils::spl_token_transfer(){// auction/program/src/utils.rs:152:1: 176:2 }

```

- description:
- link:
- alleviation:

Issue: 8: CrossProgramInvocation

Category	Severity	Status
CrossProgramInvocation	Critical	UnResolved

- Location

```
auction/program/src/utils.rs
```

- Code Context

```rust
pub fn spl_token_transfer(params: TokenTransferParams<'_, '_>) -> ProgramResult {
    let TokenTransferParams {
        source,
        destination,
        authority,
        token_program,
        amount,
        authority_signer_seeds,
    } = params;

    let result = invoke_signed(
        &spl_token::instruction::transfer(
            token_program.key,
            source.key,
            destination.key,
            authority.key,
            &[],
            amount,
        )?,
        &[source, destination, authority, token_program],
        &[authority_signer_seeds],
    );

    result.map_err(|_| AuctionError::TokenTransferFailed.into())
}
```

- Call Stack

```
fn entrypoint::entrypoint(){// /home/yifei/.cargo/registry/src/github.com-1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10 }
    fn entrypoint::process_instruction(){// auction/program/src/entrypoint.rs:12:1: 23:2 }
        fn processor::process_instruction(){// auction/program/src/processor.rs:30:1: 48:2 }
            fn processor::cancel_bid::cancel_bid(){// auction/program/src/processor/cancel_bid.rs:93:1: 253:2 }
                    fn utils::spl_token_transfer(){// auction/program/src/utils.rs:152:1: 176:2 }

```

- description:
- link:
- alleviation:

Issue: 9: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

auction/program/src/processor/cancel_bid.rs:38:1: 40:2

```
pub struct CancelBidArgs {
    pub resource: Pubkey,
}
auction/program/src/processor/claim_bid.rs:33:1: 35:2
    pub struct ClaimBidArgs {
    pub resource: Pubkey,
}
auction/program/src/processor/start_auction.rs:45:1: 48:2
    pub struct StartAuctionArgs {
    /// The resource being auctioned. See AuctionData.
    pub resource: Pubkey,
}
auction/program/src/processor/end_auction.rs:28:1: 34:2
    pub struct EndAuctionArgs {
    /// The resource being auctioned. See AuctionData.
    pub resource: Pubkey,
    /// If the auction was blinded, a revealing price must be specified to release the auction
    /// winnings.
    pub reveal: Option<Revealer>,
}
auction/program/src/processor.rs:461:1: 461:37
    pub struct Bid(pub Pubkey, pub u64);
auction/program/src/processor.rs:765:1: 774:2
    pub struct BidderPot {
    /// Points at actual pot that is a token account
    pub bidder_pot: Pubkey,
    /// Originating bidder account
    pub bidder_act: Pubkey,
    /// Auction account
    pub auction_act: Pubkey,
    /// emptied or not
    pub emptied: bool,
}
auction/program/src/processor.rs:737:1: 749:2
    pub struct BidderMetadata {
    // Relationship with the bidder who's metadata this covers.
    pub bidder_pubkey: Pubkey,
    // Relationship with the auction this bid was placed on.
    pub auction_pubkey: Pubkey,
    // Amount that the user bid.
    pub last_bid: u64,
    // Tracks the last time this user bid.
    pub last_bid_timestamp: UnixTimestamp,
    // Whether the last bid the user made was cancelled. This should also be enough to know if the
    // user is a winner, as if cancelled it implies previous bids were also cancelled.
    pub cancelled: bool,
}
auction/program/src/processor.rs:72:1: 96:2
    pub struct AuctionData {
    /// Pubkey of the authority with permission to modify this auction.
    pub authority: Pubkey,
    /// Pubkey of the resource being bid on.
    /// TODO try to bring this back some day. Had to remove this due to a stack access violation bug
    /// interactin that happens in metaplex during redemptions due to some low level rust error
    /// that happens when AuctionData has too many fields. This field was the least used.
    ///pub resource: Pubkey,
    /// Token mint for the SPL token being used to bid
    pub token_mint: Pubkey,
    /// The time the last bid was placed, used to keep track of auction timing.
    pub last_bid: Option<UnixTimestamp>,
    /// Slot time the auction was officially ended by.
    pub ended_at: Option<UnixTimestamp>,
    /// End time is the cut-off point that the auction is forced to end by.
    pub end_auction_at: Option<UnixTimestamp>,
    /// Gap time is the amount of time in slots after the previous bid at which the auction ends.
    pub end_auction_gap: Option<UnixTimestamp>,
    /// Minimum price for any bid to meet.
    pub price_floor: PriceFloor,
    /// The state the auction is in, whether it has started or ended.
    pub state: AuctionState,
    /// Auction Bids, each user may have one bid open at a time.
    pub bid_state: BidState,
}

```

- Call Stack

```
UnResolved
```

- description:
- link:
- alleviation:

Issue: 10: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

auction/program/src/processor/claim_bid.rs:33:1: 35:2

```
pub struct ClaimBidArgs {
    pub resource: Pubkey,
}
auction/program/src/processor/start_auction.rs:45:1: 48:2
    pub struct StartAuctionArgs {
    /// The resource being auctioned. See AuctionData.
    pub resource: Pubkey,
}
auction/program/src/processor/end_auction.rs:28:1: 34:2
    pub struct EndAuctionArgs {
    /// The resource being auctioned. See AuctionData.
    pub resource: Pubkey,
    /// If the auction was blinded, a revealing price must be specified to
     ↪ release the auction
    /// winnings.
    pub reveal: Option<Revealer>,
}
auction/program/src/processor.rs:461:1: 461:37
    pub struct Bid(pub Pubkey, pub u64);
auction/program/src/processor.rs:765:1: 774:2
    pub struct BidderPot {
    /// Points at actual pot that is a token account
    pub bidder_pot: Pubkey,
    /// Originating bidder account
    pub bidder_act: Pubkey,
    /// Auction account
    pub auction_act: Pubkey,
    /// emptied or not
    pub emptied: bool,
}
auction/program/src/processor.rs:737:1: 749:2
```

```
    pub struct BidderMetadata {
    // Relationship with the bidder who's metadata this covers.
    pub bidder_pubkey: Pubkey,
    // Relationship with the auction this bid was placed on.
    pub auction_pubkey: Pubkey,
    // Amount that the user bid.
    pub last_bid: u64,
    // Tracks the last time this user bid.
    pub last_bid_timestamp: UnixTimestamp,
    // Whether the last bid the user made was cancelled. This should also
     ↪  be enough to know if the
    // user is a winner, as if cancelled it implies previous bids were also
     ↪  cancelled.
    pub cancelled: bool,
}
auction/program/src/processor.rs:72:1: 96:2
    pub struct AuctionData {
    /// Pubkey of the authority with permission to modify this auction.
    pub authority: Pubkey,
    /// Pubkey of the resource being bid on.
    /// TODO try to bring this back some day. Had to remove this due to a
     ↪  stack access violation bug
    /// interactin that happens in metaplex during redemptions due to some
     ↪  low level rust error
    /// that happens when AuctionData has too many fields. This field was
     ↪  the least used.
    ///pub resource: Pubkey,
    /// Token mint for the SPL token being used to bid
    pub token_mint: Pubkey,
    /// The time the last bid was placed, used to keep track of auction
     ↪  timing.
    pub last_bid: Option<UnixTimestamp>,
    /// Slot time the auction was officially ended by.
    pub ended_at: Option<UnixTimestamp>,
    /// End time is the cut-off point that the auction is forced to end by.
    pub end_auction_at: Option<UnixTimestamp>,
    /// Gap time is the amount of time in slots after the previous bid at
     ↪  which the auction ends.
    pub end_auction_gap: Option<UnixTimestamp>,
    /// Minimum price for any bid to meet.
    pub price_floor: PriceFloor,
    /// The state the auction is in, whether it has started or ended.
    pub state: AuctionState,
    /// Auction Bids, each user may have one bid open at a time.
    pub bid_state: BidState,
}

```

- Call Stack

```
UnResolved
```

- description:
- link:
- alleviation:

Issue: 11: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

auction/program/src/processor/start_auction.rs:45:1: 48:2

```
pub struct StartAuctionArgs {
    /// The resource being auctioned. See AuctionData.
    pub resource: Pubkey,
}
auction/program/src/processor/end_auction.rs:28:1: 34:2
    pub struct EndAuctionArgs {
    /// The resource being auctioned. See AuctionData.
    pub resource: Pubkey,
    /// If the auction was blinded, a revealing price must be specified to
     ↪  release the auction
    /// winnings.
    pub reveal: Option<Revealer>,
}
auction/program/src/processor.rs:461:1: 461:37
    pub struct Bid(pub Pubkey, pub u64);
auction/program/src/processor.rs:765:1: 774:2
    pub struct BidderPot {
    /// Points at actual pot that is a token account
    pub bidder_pot: Pubkey,
    /// Originating bidder account
    pub bidder_act: Pubkey,
    /// Auction account
    pub auction_act: Pubkey,
    /// emptied or not
    pub emptied: bool,
}
auction/program/src/processor.rs:737:1: 749:2
    pub struct BidderMetadata {
    // Relationship with the bidder who's metadata this covers.
    pub bidder_pubkey: Pubkey,
    // Relationship with the auction this bid was placed on.
    pub auction_pubkey: Pubkey,
    // Amount that the user bid.
    pub last_bid: u64,
    // Tracks the last time this user bid.
    pub last_bid_timestamp: UnixTimestamp,
    // Whether the last bid the user made was cancelled. This should also
     ↪  be enough to know if the
    // user is a winner, as if cancelled it implies previous bids were also
     ↪  cancelled.
    pub cancelled: bool,
}
auction/program/src/processor.rs:72:1: 96:2
    pub struct AuctionData {
    /// Pubkey of the authority with permission to modify this auction.
    pub authority: Pubkey,
    /// Pubkey of the resource being bid on.
    /// TODO try to bring this back some day. Had to remove this due to a
     ↪  stack access violation bug
    /// interactin that happens in metaplex during redemptions due to some
     ↪  low level rust error
    /// that happens when AuctionData has too many fields. This field was
     ↪  the least used.
    ///pub resource: Pubkey,
    /// Token mint for the SPL token being used to bid
    pub token_mint: Pubkey,
    /// The time the last bid was placed, used to keep track of auction
     ↪  timing.
    pub last_bid: Option<UnixTimestamp>,
    /// Slot time the auction was officially ended by.
    pub ended_at: Option<UnixTimestamp>,
    /// End time is the cut-off point that the auction is forced to end by.
    pub end_auction_at: Option<UnixTimestamp>,
    /// Gap time is the amount of time in slots after the previous bid at
     ↪  which the auction ends.
    pub end_auction_gap: Option<UnixTimestamp>,
    /// Minimum price for any bid to meet.
    pub price_floor: PriceFloor,
    /// The state the auction is in, whether it has started or ended.
    pub state: AuctionState,
    /// Auction Bids, each user may have one bid open at a time.
    pub bid_state: BidState,
}
```

- Call Stack

```
UnResolved
```

- description:
- link:
- alleviation:

Issue: 12: TypeConfusion

Category	Severity	Status
TypeConfusion	Critical	GitHub Link to be added.

- Location

auction/program/src/processor/create_auction.rs:28:1: 47:2

```rust
pub struct CreateAuctionArgs {
    /// How many winners are allowed for this auction. See AuctionData.
    pub winners: WinnerLimit,
    /// End time is the cut-off point that the auction is forced to end by. See AuctionData.
    pub end_auction_at: Option<UnixTimestamp>,
    /// Gap time is how much time after the previous bid where the auction ends. See AuctionData.
    pub end_auction_gap: Option<UnixTimestamp>,
    /// Token mint for the SPL token used for bidding.
    pub token_mint: Pubkey,
    /// Authority
    pub authority: Pubkey,
    /// The resource being auctioned. See AuctionData.
    pub resource: Pubkey,
    /// Set a price floor.
    pub price_floor: PriceFloor,
    /// Add a tick size increment
    pub tick_size: Option<u64>,
    /// Add a minimum percentage increase each bid must meet.
    pub gap_tick_size_percentage: Option<u8>,
}
auction/program/src/processor/create_auction_v2.rs:29:1: 52:2
    pub struct CreateAuctionArgsV2 {
    /// How many winners are allowed for this auction. See AuctionData.
    pub winners: WinnerLimit,
    /// End time is the cut-off point that the auction is forced to end by. See AuctionData.
    pub end_auction_at: Option<UnixTimestamp>,
    /// Gap time is how much time after the previous bid where the auction ends. See AuctionData.
    pub end_auction_gap: Option<UnixTimestamp>,
    /// Token mint for the SPL token used for bidding.
    pub token_mint: Pubkey,
    /// Authority
    pub authority: Pubkey,
    /// The resource being auctioned. See AuctionData.
    pub resource: Pubkey,
    /// Set a price floor.
    pub price_floor: PriceFloor,
    /// Add a tick size increment
    pub tick_size: Option<u64>,
    /// Add a minimum percentage increase each bid must meet.
    pub gap_tick_size_percentage: Option<u8>,
    /// Add a instant sale price.
    pub instant_sale_price: Option<u64>,
    /// Auction name
    pub name: Option<AuctionName>,
}

```

- 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.