VRust

Security Assessment

O2Lab VRust Team

07/11/2024 08:26:14

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_candy_machine
GitHub Location	https://github.com/parasol-aser/vrust
sha256	Unknown

Audit Summary

Delivery Date	07/11/2024
Audit Methodology	Static Analysis
Key Components	

Vulnerability Summary

Vulnerability Level	Total
Critical	10
Major	0
Medium	0
Minor	0
Informational	0
Discussion	0

Findings

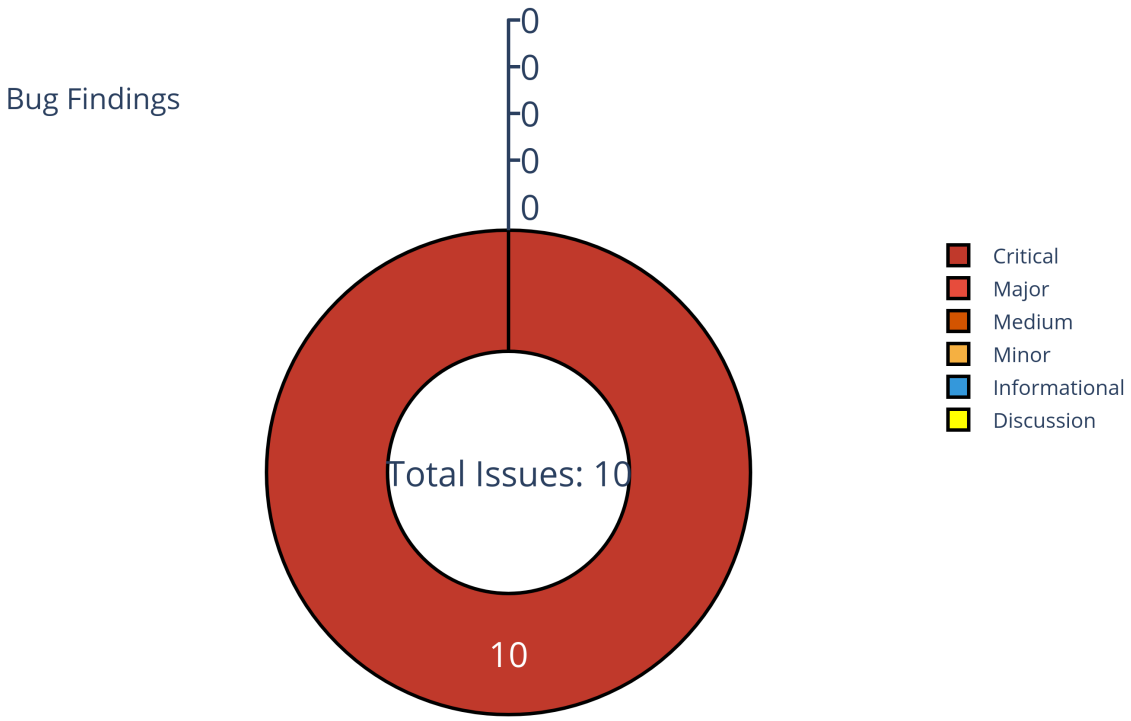

Figure 1: Findings

Finding Statistic

Category	Count
IntegerFlow	6
MissingKeyCheck	4

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	MissingKeyCheck	Critical	UnResolved
7	MissingKeyCheck	Critical	UnResolved
8	MissingKeyCheck	Critical	UnResolved
9	MissingKeyCheck	Critical	UnResolved

Issue: 0: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

src/lib.rs:657:15: 657:65

```
657 (data.items_available as usize) * CONFIG_LINE_SIZE
658
```

- Code Context

– Function Definition:

```
651 fn get_space_for_candy(data: CandyMachineData) ->
    ↪  core::result::Result<usize, ProgramError>
652
```

Vulnerability at Line: 657

```
652     let num = if data.hidden_settings.is_some() {
653         CONFIG_ARRAY_START
654     } else {
655         CONFIG_ARRAY_START
656             + 4
657             + (data.items_available as usize) * CONFIG_LINE_SIZE
658             + 8
659             + 2 * ((data
660             .items_available
661             .checked_div(8)
662
```

- Call Stack

```
fn entrypoint(){// /home/biagio/.cargo/registry/src/github.com-
↪  1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10
↪  }
    fn entry(){// src/lib.rs:37:1: 37:11 }
        fn dispatch(){// src/lib.rs:37:1: 37:11 }
            fn __private::__global::initialize_candy_machine(){//
            ↪  src/lib.rs:37:1: 37:11 }
                    fn <InitializeCandyMachine<'info> as
                    ↪  anchor_lang::Accounts<'info>>::try_accounts(){//
                    ↪  src/lib.rs:670:10: 670:18 }
                        fn get_space_for_candy(){// src/lib.rs:651:1: 667:2
                        ↪  }

```

- description:
- link:
- alleviation:

Issue: 1: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

src/lib.rs:608:15: 608:79

```
608 (candy_machine.data.items_available as usize) * CONFIG_LINE_SIZE
609
```

- Code Context

Vulnerability at Line: 608

```
603             data[i] = new_data[i];
604         }
605
606         let vec_start = CONFIG_ARRAY_START
607             + 4
608             + (candy_machine.data.items_available as usize) *
                ↪  CONFIG_LINE_SIZE;
609         let as_bytes = (candy_machine
610             .data
611             .items_available
612             .checked_div(8)
613
```

- Call Stack

```
1 fn entrypoint(){// /home/biagio/.cargo/registry/src/github.com-
  ↪  1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10
  ↪  }
2     fn entry(){// src/lib.rs:37:1: 37:11 }
3         fn dispatch(){// src/lib.rs:37:1: 37:11 }
4             fn __private::__global::initialize_candy_machine(){//
               ↪  src/lib.rs:37:1: 37:11 }
5                     fn nft_candy_machine_v2::initialize_candy_machine(){//
                       ↪  src/lib.rs:551:5: 620:6 }
6
```

- description:
- link:
- alleviation:

Issue: 2: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

src/lib.rs:497:49: 497:84

```
(index as usize) * CONFIG_LINE_SIZE

```

- Code Context

Vulnerability at Line: 497

```

        let as_vec = fixed_config_lines.try_to_vec()?;
        // remove unneeded u32 because we're just gonna edit the u32 at the front
        let serialized: &[u8] = &as_vec.as_slice()[4..];

        let position = CONFIG_ARRAY_START + 4 + (index as usize) * CONFIG_LINE_SIZE;

        let array_slice: &mut [u8] =
            &mut data[position..position + fixed_config_lines.len() * CONFIG_LINE_SIZE];


```

- Call Stack

```
fn entrypoint(){// /home/biagio/.cargo/registry/src/github.com-1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10 }
    fn entry(){// src/lib.rs:37:1: 37:11 }
        fn dispatch(){// src/lib.rs:37:1: 37:11 }
            fn __private::__global::add_config_lines(){// src/lib.rs:37:1: 37:11 }
```

```
fn nft_candy_machine_v2::add_config_lines(){//
↪  src/lib.rs:460:5: 549:6 }
```

- description:
- link:
- alleviation:

Issue: 3: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

src/lib.rs:134:24: 134:53

```
134 (expire_time - EXPIRE_OFFSET)
135
```

- Code Context

Vulnerability at Line: 134

```
129                     msg!(
130                         "Comparing token expire time {} and go_live_date
                             ↪  {}",
131                         expire_time,
132                         val
133                     );
134                     if (expire_time - EXPIRE_OFFSET) < val {
135                         if let Some(ws) =
                             ↪  &candy_machine.data.whitelist_mint_settings {
136                             // when dealing with whitelist, the expire_time
                                 ↪  can be
137                             // before the go_live_date only if presale
                                 ↪  enabled
138                             if !ws.presale {
139
```

- Call Stack

```
1 fn entrypoint(){// /home/biagio/.cargo/registry/src/github.com-
  ↪  1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10
  ↪  }
2     fn entry(){// src/lib.rs:37:1: 37:11 }
3         fn dispatch(){// src/lib.rs:37:1: 37:11 }
```

```
4               fn __private::__global::mint_nft(){// src/lib.rs:37:1: 37:11 }
5                       fn nft_candy_machine_v2::mint_nft(){// src/lib.rs:42:5:
                         ↪  428:6 }
6
```

- description:
- link:
- alleviation:

Issue: 4: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

src/lib.rs:886:30: 892:51

```
886 CONFIG_ARRAY_START
887         + 4
888         + (items_available) * CONFIG_LINE_SIZE
889         + 4
890         + items_available
891         .checked_div(8)
892         .ok_or(ErrorCode::NumericalOverflowError)?
893
```

- Code Context

Vulnerability at Line: 886

```
881     pos: bool,
882 ) -> core::result::Result<(usize, bool), ProgramError> {
883     let mut index_to_use = index;
884     let mut taken = 1;
885     let mut found = false;
886     let bit_mask_vec_start = CONFIG_ARRAY_START
887         + 4
888         + (items_available) * CONFIG_LINE_SIZE
889         + 4
890         + items_available
891
```

- Call Stack

```
1 fn entrypoint(){// /home/biagio/.cargo/registry/src/github.com-
  ↪ 1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10
  ↪ }
2     fn entry(){// src/lib.rs:37:1: 37:11 }
```

```
        fn dispatch(){// src/lib.rs:37:1: 37:11 }
            fn __private::__global::mint_nft(){// src/lib.rs:37:1: 37:11 }
                    fn nft_candy_machine_v2::mint_nft(){// src/lib.rs:42:5:
                    ↪  428:6 }
                        fn get_config_line(){// src/lib.rs:957:1: 1021:2 }
                            fn get_good_index(){// src/lib.rs:877:1: 955:2
                            ↪  }

```

- description:
- link:
- alleviation:

Issue: 5: IntegerFlow

Category	Severity	Status
IntegerFlow	Critical	UnResolved

- Location

src/lib.rs:988:37: 988:70

```
988 index_to_use * (CONFIG_LINE_SIZE)
989
```

- Code Context

Vulnerability at Line: 988

```
983
984     msg!(
985         "Index actually ends up due to used bools {:?}",
986         index_to_use
987     );
988     if arr[CONFIG_ARRAY_START + 4 + index_to_use * (CONFIG_LINE_SIZE)] == 1
        ↪  {
989         return Err(ErrorCode::CannotFindUsableConfigLine.into());
990     }
991
992     let data_array = &mut arr[CONFIG_ARRAY_START + 4 + index_to_use *
        ↪  (CONFIG_LINE_SIZE)
993
```

Other Use Case for Variable: index_to_use * (CONFIG_LINE_SIZE)

```
992     let data_array = &mut arr[CONFIG_ARRAY_START + 4 + index_to_use *
        ↪  (CONFIG_LINE_SIZE)
```

- Call Stack

```
fn entrypoint(){// /home/biagio/.cargo/registry/src/github.com-
↪  1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10
↪  }
    fn entry(){// src/lib.rs:37:1: 37:11 }
        fn dispatch(){// src/lib.rs:37:1: 37:11 }
            fn __private::__global::mint_nft(){// src/lib.rs:37:1: 37:11 }
                    fn nft_candy_machine_v2::mint_nft(){// src/lib.rs:42:5:
                    ↪  428:6 }
                        fn get_config_line(){// src/lib.rs:957:1: 1021:2 }

```

- description:
- link:
- alleviation:

Issue: 6: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

src/lib.rs:642:11: 642:42

```
642 authority.lamports.borrow_mut()
643
```

- Code Context

Vulnerability at Line: 642

```
637         let pay = &ctx.accounts.candy_machine.to_account_info();
638         let snapshot: u64 = pay.lamports();
639
640         **pay.lamports.borrow_mut() = 0;
641
642         **authority.lamports.borrow_mut() = authority
643             .lamports()
644             .checked_add(snapshot)
645             .ok_or(ErrorCode::NumericalOverflowError)?;
646
647
```

- Call Stack

```
1 fn entrypoint(){// /home/biagio/.cargo/registry/src/github.com-
  ↪  1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10
  ↪  }
2     fn entry(){// src/lib.rs:37:1: 37:11 }
3         fn dispatch(){// src/lib.rs:37:1: 37:11 }
4             fn __private::__global::withdraw_funds(){// src/lib.rs:37:1:
               ↪  37:11 }
5                     fn nft_candy_machine_v2::withdraw_funds(){//
                       ↪  src/lib.rs:635:5: 648:6 }
6
```

- description:
- link:
- alleviation:

Issue: 7: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

src/utils.rs:18:43: 18:69

```
account_info.data.borrow()

```

- Code Context

Vulnerability at Line: 18

```
pub fn assert_initialized<T: Pack + IsInitialized>(
    account_info: &AccountInfo,
) -> Result<T, ProgramError> {
    let account: T = T::unpack_unchecked(&account_info.data.borrow())?;
    if !account.is_initialized() {
        Err(ErrorCode::Uninitialized.into())
    } else {
        Ok(account)

```

- Call Stack

```
fn entrypoint(){// /home/biagio/.cargo/registry/src/github.com-
↪  1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10
↪  }
    fn entry(){// src/lib.rs:37:1: 37:11 }
        fn dispatch(){// src/lib.rs:37:1: 37:11 }
            fn __private::__global::initialize_candy_machine(){//
            ↪  src/lib.rs:37:1: 37:11 }
                    fn nft_candy_machine_v2::initialize_candy_machine(){//
                    ↪  src/lib.rs:551:5: 620:6 }
                        fn utils::assert_initialized(){//
                        ↪  src/utils.rs:15:1: 24:2 }

```

- description:
- link:
- alleviation:

Issue: 8: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

src/lib.rs:600:24: 600:63

```
600 candy_machine_account.data.borrow_mut()
601
```

- Code Context

Vulnerability at Line: 600

```
595             return Err(ErrorCode::TooManyCreators.into());
596         }
597
598         let mut new_data =
            ↪ CandyMachine::discriminator().try_to_vec().unwrap();
599         new_data.append(&mut candy_machine.try_to_vec().unwrap());
600         let mut data = candy_machine_account.data.borrow_mut();
601         // god forgive me couldnt think of better way to deal with this
602         for i in 0..new_data.len() {
603             data[i] = new_data[i];
604         }
605
```

- Call Stack

```
1 fn entrypoint(){// /home/biagio/.cargo/registry/src/github.com-
  ↪ 1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10
  ↪ }
2     fn entry(){// src/lib.rs:37:1: 37:11 }
3         fn dispatch(){// src/lib.rs:37:1: 37:11 }
4             fn __private::__global::initialize_candy_machine(){//
               ↪ src/lib.rs:37:1: 37:11 }
5                     fn nft_candy_machine_v2::initialize_candy_machine(){//
                       ↪ src/lib.rs:551:5: 620:6 }
6
```

- description:
- link:
- alleviation:

Issue: 9: MissingKeyCheck

Category	Severity	Status
MissingKeyCheck	Critical	UnResolved

- Location

src/lib.rs:90:16: 90:48

```
90 gateway_token_info.data.borrow()
91
```

- Code Context

Vulnerability at Line: 90

```
85                 return Err(ErrorCode::GatewayTokenMissing.into());
86             }
87             let gateway_token_info =
               ↪  &ctx.remaining_accounts[remaining_accounts_counter];
88             let gateway_token =
               ↪  ::solana_gateway::borsh::try_from_slice_incomplete::<
89                 ::solana_gateway::state::GatewayToken,
90             >(*gateway_token_info.data.borrow())?;
91             // stores the expire_time before the verification, since the
               ↪  verification
92             // will update the expire_time of the token and we won't be
               ↪  able to
93             // calculate the creation time
94             let expire_time = gateway_token
95
```

- Call Stack

```
1 fn entrypoint(){// /home/biagio/.cargo/registry/src/github.com-
  ↪  1ecc6299db9ec823/solana-program-1.9.5/src/entrypoint.rs:120:9: 127:10
  ↪  }
2     fn entry(){// src/lib.rs:37:1: 37:11 }
3         fn dispatch(){// src/lib.rs:37:1: 37:11 }
```

```
4               fn __private::__global::mint_nft(){// src/lib.rs:37:1: 37:11 }
5                       fn nft_candy_machine_v2::mint_nft(){// src/lib.rs:42:5:
                         ↪  428:6 }
6
```

- 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.